

# Installazione e avvio di un sistema GNU/Linux



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

[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	<p>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.</p> <p>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'.</p> <p>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.</p>
Ubuntu	<p>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.</p>

## 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 1024-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**’,<sup>1</sup> 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<sub>16</sub> 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'`:<sup>2</sup>

```
# 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'`:<sup>3</sup>

```
# 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
<code>root=<i>dispositivo</i></code>	Permette di indicare un dispositivo differente da quello predefinito per innestare il file system principale.
<code>ro</code>	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.
<code>rw</code>	Permette di definire un accesso iniziale al file system principale in lettura e scrittura.

Tabella 6.31. Memoria.

Sintassi	Descrizione
<code>mem=<i>dimensione</i></code>	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.

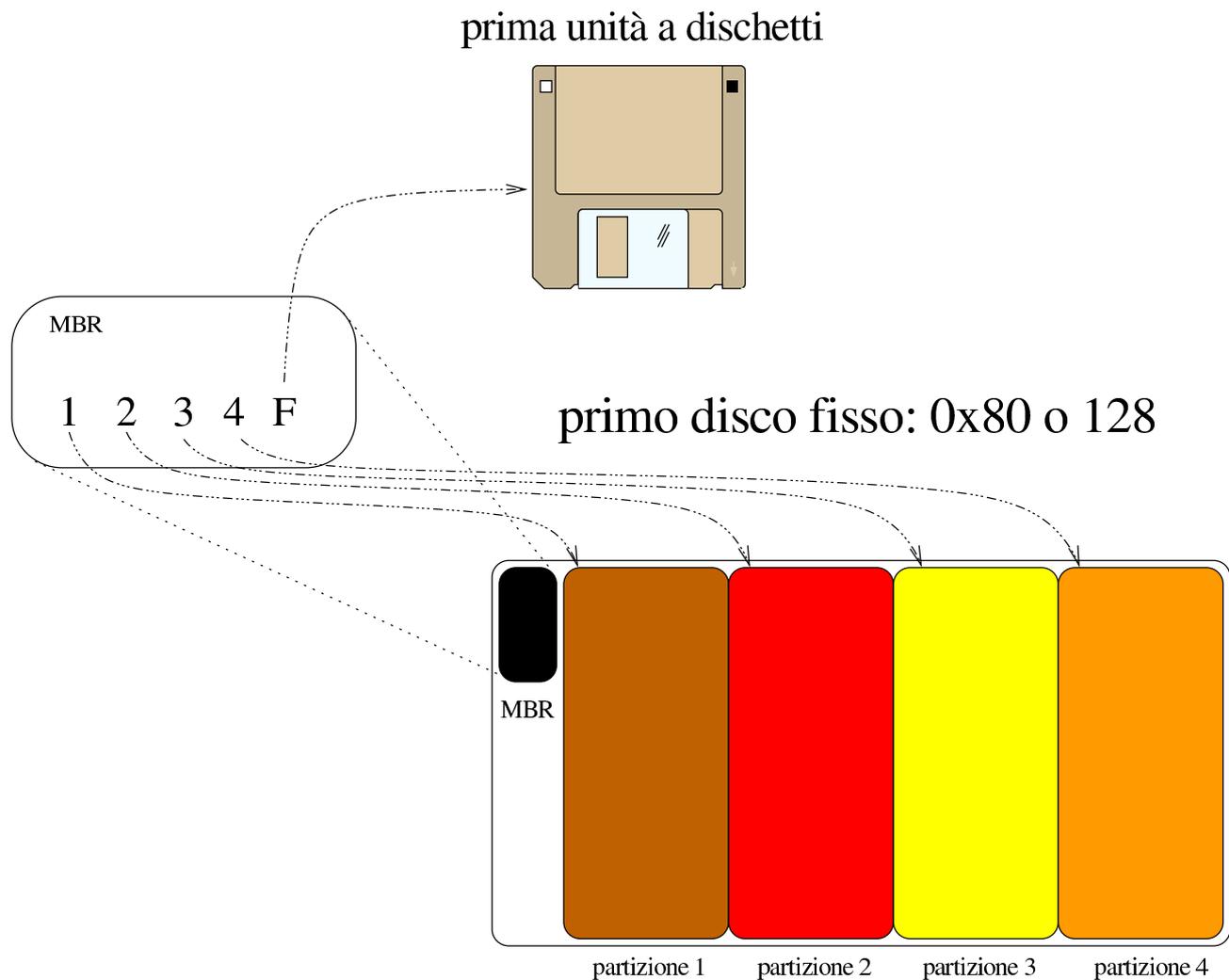
Sintassi	Descrizione
<code>init=<i>programma_iniziale</i></code>	Permette di definire il nome, completo di percorso, del programma che deve svolgere le funzioni di «processo iniziale» (Init). Il kernel provvede da solo a cercare '/sbin/init' e in alternativa '/etc/init'. Come ultima risorsa tenta di 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
<pre>reserve=<i>indirizzo_i_o</i>, ↵ ↵ <i>estensione</i> ↵ ↵ [ , <i>indirizzo_i/o</i> , ↵ ↵ <i>estensione</i> ] ...</pre>	<p>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.</p>

## 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<sup>4</sup>, 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 rinunciarebbe 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 <i>drive</i>  --drive <i>drive</i>	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
<pre>-i [s] [k] [a] [n] --interrupt [s] [k] [a] [n]</pre>	<p>Consente di controllare l'apparizione dell'invito: la lettera «s» fa sì che la pressione di un tasto [<i>Maiuscole</i>] o di un tasto [<i>Ctrl</i>] 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>
<pre>-p [1] [2] [3] [4] [F] [D] --partition [1] [2] [3] [4] [F] [D]</pre>	<p>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».</p>
<pre>-t <i>n</i> --timeout <i>n</i></pre>	<p>Attende <i>n</i>/18 secondi prima di avviare la partizione o l'unità a dischetti fissata con l'opzione '-p'.</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.

<i>ms-sys(1)</i> <sup>5</sup>	<p><a href="http://ms-sys.sourceforge.net">http://ms-sys.sourceforge.net</a></p> <p>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.</p>
-------------------------------	--

## 6.10 SYSLINUX

«

**SYSLINUX**<sup>6</sup> 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=mementestp.dsk
```

In questo caso, tra i file di avvio si trova anche ‘`mementestp.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=nlxrd.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

&lt;&lt;

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 predisporre 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 server 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 server 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/>

<sup>1</sup> **util-linux: fdisk** GNU GPL

<sup>2</sup> **util-linux: disk-utils** GNU GPL

<sup>3</sup> **e2fsprogs** GNU GPL

<sup>4</sup> **MBR** GNU GPL

<sup>5</sup> **MS-SYS** GNU GPL

<sup>6</sup> **SYSLINUX** GNU GPL