

Avvio del sistema e conclusione



Interazione con il kernel	3191
Avvio e conclusione del sistema «normale»	3194

os16 ha due modalità di funzionamento: si può interagire direttamente con il kernel, oppure si può avviare il processo **'init'** e procedere con l'organizzazione consueta di un sistema Unix tradizionale. La modalità di colloquio diretto con il kernel è servita per consentire lo sviluppo di os16 e potrebbe essere utile per motivi di studio. Va osservato che durante l'interazione diretta con il kernel si dispone di una sola console, mentre quando si avvia **'init'** si possono avere delle console virtuali in base alla configurazione.

Interazione con il kernel

L'avvio di os16 passa, in ogni caso, per una prima fase di colloquio con il kernel. Si ottiene un menù e si possono premere semplicemente dei tasti, secondo l'elenco previsto, per ottenere delle azioni molto semplici. In questa fase il disco da cui risulta avviato il kernel è già innestato ed è prevista la possibilità di avviare tre programmi: **'/bin/aaa'**, **'/bin/bbb'** e **'/bin/ccc'**. In tal modo, si ha la possibilità di avviare qualcosa, a titolo diagnostico, prima dello stesso **'init'** (**'/bin/init'**).



Figura u151.1. Aspetto di os16 in funzione, con il menù in evidenza.

```
os16 build 20AA.MM.GG HH:MM:SS ram 639 Kibyte
.------.
| [h]      show this menu                               |
| [p]      process status and memory map              |
| [1]..[9] kill process 1 to 9                        |
| [A]..[F] kill process 10 to 15                     |
| [l]      send SIGCHLD to process 1                 |
| [a]..[c] run programs '/bin/aaa' to '/bin/ccc' in parallel |
| [f]      system file status                         |
| [n], [N] list of active inodes                     |
| [z]      print root file system zone map (read left to right) |
| [m], [M] mount/umount '/dev/dsk1' at '/usr/'      |
| [x]      exit interaction with kernel and start '/bin/init' |
| [q]      quit kernel                               |
|-----|
```

Le funzioni principali disponibili in questa modalità diagnostica sono riassunte nella tabella successiva:

Tasto	Risultato
[h]	Mostra il menù di funzioni disponibili.
[1]	Invia il segnale ' SIGKILL ' al processo numero uno.
[2]...[9]	Invia il segnale ' SIGTERM ' al processo con il numero corrispondente.
[A]...[F]	Invia il segnale ' SIGTERM ' al processo con il numero da 10 a 15.
[a], [b], [c]	Avvia il programma '/bin/aaa', '/bin/bbb' o '/bin/ccc'.
[f]	Mostra l'elenco dei file aperti nel sistema.
[m], [M]	Innesta o stacca il secondo dischetto dalla directory '/usr/'.
[n], [N]	Mostra l'elenco degli inode aperti: l'elenco è composto da due parti.

Tasto	Risultato
[l]	Invia il segnale ' SIGCHLD ' al processo numero uno.
[p]	Mostra la situazione dei processi e altre informazioni.
[x]	Termina il ciclo e successivamente si passa all'avvio di '/bin/init'.
[q]	Ferma il sistema.

Figura u151.3. Aspetto di os16 in funzione mentre visualizza anche la tabella dei processi avviati (tasto [p]).

```

ababaaabababaaababababbaabababaaababap
pp  p pg
id id rp  tty  uid euid suid usage s iaddr isiz daddr dsiz sp  name
0  0  0 0000  0  0  0 00.35 R 10500 eb7c 00500 0000 ffc8 os16 kernel
0  1  0 0000  0  0  0 00.33 r 2f100 0600 2f700 aa00 a8e8 /bin/ccc
0  2  0 0000  0  0  0 00.01 r 1f100 0600 84300 aa00 a8e8 /bin/ccc
2  3  0 0000 10 10 10 00.01 r 1f100 0600 44b00 aa00 a8e8 /bin/ccc
0  4  0 0000  0  0  0 00.17 r 21600 0600 3a100 aa00 a8e8 /bin/ccc
4  5  0 0000 10 10 10 00.02 r 21c00 2400 6f100 a900 a86c /bin/aaa
4  6  0 0000 11 11 11 00.02 s 24000 2500 59d00 a900 a8b6 /bin/bbb
0  7  0 0000  0  0  0 00.13 r 26500 0600 64600 aa00 a8e8 /bin/ccc
7  8  0 0000 10 10 10 00.02 r 26b00 2400 8ee00 a900 981e /bin/aaa
7  9  0 0000 11 11 11 00.02 s 2bf00 2500 79a00 a900 a8b6 /bin/bbb
CS=1050 DS=0050 SS=0050 ES=0050 BP=ffe4 SP=ffe4 heap_min=878c etext=eb7c edata=1
b3c ebss=4c34 ksp=ffc8 clock=0000084b, time elapsed=00:01:57
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffe000000000001ffffffff0007fffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff800
0000000000000000000000000000000000000000000000000000000000000000007
ffffffffffffffffffffffffffffffffffffffff7ffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffbfffffffffffffffffffffffffffffffffffffffffffffe0000000000000000000000f

abbaaaaabababaaababababbaabbaabababbaabbbbbbbbbbbbbbb

```

Premendo [x], il ciclo termina e il kernel avvia '/bin/init', ma prima di farlo occorre che non ci siano altri processi in funzione, perché 'init' deve assumere il ruolo di processo 1, ovvero il primo

dopo il kernel.

Figura u151.4. Aspetto di os16 in funzione con il menù in evidenza, dopo aver premuto il tasto [x] per avviare 'init'.

```
os16 build 20AA.MM.GG HH:MM:SS ram 639 Kibyte
.-----
| [h]      show this menu                               |
| [p]      process status and memory map               |
| [1]..[9] kill process 1 to 9                         |
| [A]..[F] kill process 10 to 15                      |
| [l]      send SIGCHLD to process 1                   |
| [a]..[c] run programs '/bin/aaa' to '/bin/ccc' in parallel |
| [f]      system file status                          |
| [n], [N] list of active inodes                       |
| [z]      print root file system zone map (read left to right) |
| [m], [M] mount/umount '/dev/dsk1' at '/usr/'        |
| [x]      exit interaction with kernel and start '/bin/init' |
| [q]      quit kernel                                 |
'-----
init
os16: a basic os. [Ctrl q], [Ctrl r], [Ctrl s], [Ctrl t] to change
console.
This is terminal /dev/console0
Log in as "root" or "user" with password "ciao" :-)
login:
```

Avvio e conclusione del sistema «normale»

«

Se non si intende operare direttamente con il kernel, come descritto nella sezione precedente, con la pressione del tasto [x] si avvia 'init'.

Il programma 'init' legge il file '/etc/inittab' e sulla base del suo contenuto, avvia uno o più processi 'getty', per la gestione dei vari terminali disponibili (si tratta comunque soltanto di console virtuali).

Il programma **'getty'** apre il terminale che gli viene indicato come opzione della chiamata (da **'init'** che lo determina in base al contenuto di **'/etc/inittab'**), facendo in modo che sia associato al descrittore zero (standard input). Quindi, dopo aver visualizzato il contenuto del file **'/etc/issue'**, mostra un proprio messaggio e avvia il programma **'login'**.

Il programma **'login'** prende il posto di **'getty'** che così scompare dall'elenco dei processi. **'login'** procede chiedendo all'utente di identificarsi, utilizzando il file **'/etc/passwd'** per verificare le credenziali di accesso. Se l'identificazione ha successo, viene avviata la shell definita nel file **'/etc/passwd'** per l'utente, in modo da sostituirsi al programma **'login'**, il quale scompare a sua volta dall'elenco dei processi.

Attraverso la shell è possibile interagire con il sistema operativo, secondo la modalità «normale», nei limiti delle possibilità di os16. Quando la shell termina di funzionare, **'init'** riavvia **'getty'**.

Per cambiare console virtuale si possono usare le combinazioni **[Ctrl q]**, **[Ctrl r]**, **[Ctrl s]** e **[Ctrl t]**, ma bisogna considerare che dipende dalla configurazione del file **'/etc/inittab'** se effettivamente vengono attivate tutte queste console.

Per concludere l'attività del sistema, basta concludere il funzionamento delle varie sessioni di lavoro (la shell finisce di funzionare con il comando interno **'exit'**) e non c'è bisogno di altro; pertanto, non è previsto l'uso di comandi come **'halt'** o **'shutdown'** e, d'altro canto, le operazioni di scrittura nel file system sono sincrone, in modo tale da non richiedere accorgimenti particolari per la chiusura delle attività.

Riquadro u151.5. Coerenza del file system.

os16 è destinato a essere usato attraverso un emulatore, dove gli unici due dischi previsti (dischetti da 1 440 Kibyte) sono semplicemente dei file. Pertanto, la manutenzione del file system, avviene all'esterno di os16, di norma con un sistema GNU/Linux. Tuttavia, occorre tenere presente che quando si usa il sistema os16, tramite l'emulatore, i file-immagine dei dischi di os16 non devono essere innestati nel sistema operativo ospitante, perché altrimenti le operazioni di scrittura eseguite da os16 potrebbero essere annullate, anche solo parzialmente, dalla gestione del sistema ospitante, rendendo il file system incoerente.