

# preLINUX

HD, MBR, sistemi dual boot e boot loader

2014.01



## Indice

Premessa.....	2
Memorie cancellabili e permanenti.....	2
BIOS e POST.....	3
Dischi magnetici e SSD.....	3
Hard Disk, interfacce per collegamenti, device Linux.....	6
Partizioni.....	7
Master Boot Record.....	9
Il boot loader GRUB.....	10
Riferimenti.....	12



## Premessa

Questi appunti trattano le conoscenze richieste, a corredo, per l'installazione di un sistema dual boot in cui due o più sistemi operativi coesistono e possono essere avviati a scelta, e della fase di avvio di un sistema di elaborazione con architettura Intel, intendendo con tale termine quei personal computer, discendenti dai PC IBM e IBM-compatibili, nati alla fine degli anni 70 e costruiti originariamente attorno al microprocessore Intel 8086/8088. Anche se i microprocessori montati negli attuali personal differiscono in maniera notevole dai loro predecessori, o si parla di microprocessori commercializzati con altri marchi (per esempio AMD), tuttavia buona parte dell'hardware che ruota attorno al microprocessore, per motivi di compatibilità, continua a funzionare a partire dagli stessi principi.

La prima parte del nome scelto per questi appunti (*pre*) vuole indicare il fatto che le informazioni trattate sono utili per poter installare un sistema Linux in un elaboratore con dual boot, in modo cosciente e consapevole. Tali informazioni intendono fornire conoscenze da acquisire, possibilmente, *prima*, non solo di installare un sistema Linux su una macchina, ma anche per un uso consapevole del computer stesso. I programmi di installazione delle distribuzioni, in generale, riescono a prendere tutte le decisioni che servono. Tuttavia una certa conoscenza di quello che accade, aiuta nella comprensione e consapevolezza delle varie scelte effettuate e inoltre può mettere al riparo da qualche disastro che comporta perdita di dati.

È richiesta, per poter seguire lo svolgimento dei vari argomenti trattati, la conoscenza minima di cosa è e come funziona, in generale, un computer.

## Memorie cancellabili e permanenti

Il computer è una macchina programmabile: per poter assolvere un compito qualsiasi è necessaria la presenza di un programma che istruisca il processore su quali parti dell'hardware fare intervenire e quando. In definitiva senza la presenza di un programma, il computer non può nemmeno essere avviato e questo comporterebbe, per esempio, la necessità di avere sempre presente, all'accensione della macchina, un programma che possa, quanto meno, inizializzare la macchina e permettere di utilizzarla. Si osservi, però, che la presenza di un programma in una memoria non cancellabile se, da una parte, permette di avviare la macchina e utilizzarla, dall'altra fa perdere il senso di macchina programmabile: non sarebbe più possibile utilizzare la *stessa macchina* per effettuare operazioni diverse. Per disporre, infatti, di tale opportunità è necessario avere a disposizione una memoria facilmente cancellabile, nella quale si può scrivere un programma diverso ogni volta che serve.

Si aggiunga, a quanto osservato precedentemente, la necessità di aggiornamento del software. Se il software normalmente fosse conservato in memorie non cancellabili, anche una semplice modifica dello stesso porterebbe, quanto meno, alla sostituzione dei circuiti in cui è registrato il programma da aggiornare. Il software dovrà quindi risiedere in memorie di massa, che permettono facilmente aggiornamenti e modifiche, e da queste verrà ricopiato in memoria quando serve. La maggior parte della memoria presente nel computer è, quindi, di tipo RAM, facilmente cancellabile.

Anche il sistema operativo, al pari degli altri software, è normalmente conservato in memorie di massa dalle quali verrà caricato in memoria centrale all'accensione della macchina. A questo punto viene spontanea una osservazione: se è vero, come è vero, che il sistema operativo ha il compito di mettere a disposizione e gestire le risorse dell'hardware, come è possibile caricare dalla memoria di massa il sistema operativo, se non c'è un sistema operativo che permette l'accesso all'hard disk, la

lettura di qualcosa e il trasferimento in memoria centrale? Chiaramente se tutta la memoria di un computer è di tipo cancellabile, il problema non ha soluzione. Occorre una memoria di tipo permanente, la ROM, nella quale è registrato un piccolo programma che viene avviato all'accensione della macchina e che riesce a caricare, da un'opportuna zona di un hard disk particolare, il sistema operativo cui cede il controllo.

## BIOS e POST

Il system BIOS (Basic Input/Output System) è un insieme di servizi per l'accesso ai dispositivi, rilevati e presenti nel computer (dischi, video, stampante, tastiera), registrati nella ROM. I servizi messi a disposizione riguardano operazioni elementari con i dispositivi del tipo: lettura/scrittura di un settore su disco/dischetto, spostamento testina lettura/scrittura, invio di un byte alla stampante, settaggio del modo video ecc... Questi servizi sono l'interfaccia a più basso livello verso i dispositivi. Nel ROM BIOS è conservato anche il programma di boot che è quello lanciato in automatico all'accensione della macchina.

La prima fase della *sequenza di boot* che porta all'avvio della macchina è l'esecuzione del POST (Power-On Self Test), un programma di diagnostica che effettua un check della macchina (scheda video, memoria) per controllare se tutto è presente e funziona in maniera corretta. Per avere una percezione di questa fase basta fare attenzione ai messaggi che appaiono su video immediatamente dopo l'accensione. Si tenga presente, tuttavia, che questa è una fase estremamente veloce e, spesso, se ne nota la presenza solo quando ci sono problemi. Se qualche parte dei dispositivi non dovesse funzionare correttamente in base alle informazioni contenute nel BIOS, il sistema emette delle segnalazioni sonore che sono codificate in modo da permettere di individuare la parte hardware che ha fatto riscontrare errori. Spesso la segnalazione sonora è accompagnata da un opportuno messaggio. Se vengono riscontrati errori, subito dopo la segnalazione sonora si ha il blocco del processo di avvio e la conseguente terminazione della sequenza di boot.

Successivamente viene eseguito il ROM BIOS degli altri dispositivi (hard disk principalmente). I risultati vengono mostrati su video in una schermata riassuntiva che mostra informazioni sul tipo di processore, la sua velocità, la quantità di memoria installata, i dispositivi installati.

Successivamente, se tutto è andato per il verso giusto, inizia la ricerca del drive di boot, cioè il drive che contiene un supporto in cui sono registrate le informazioni per il caricamento, in memoria centrale, del sistema operativo. È possibile, agendo sul settaggio dei parametri configurati nel BIOS, modificare la sequenza di ricerca del drive di boot. Durante il POST utilizzando il tasto `F2` o `CanC`, in dipendenza del BIOS presente nella macchina, è possibile accedere alle opzioni utilizzate durante il boot. Per esempio, potrebbe essere settato l'avvio dall'hard disk, cioè la macchina è predisposta per cercare nel HD il programma di avvio senza necessità di cercare altrove. In questo caso, modificando il valore in modo da permettere l'avvio da CD-ROM, uscendo dalla schermata, salvando la nuova impostazione e riavviando la macchina si ha la possibilità di avviare la macchina con un CD-ROM opportunamente predisposto. In generale c'è anche la possibilità di avviare la macchina utilizzando una chiavetta USB.

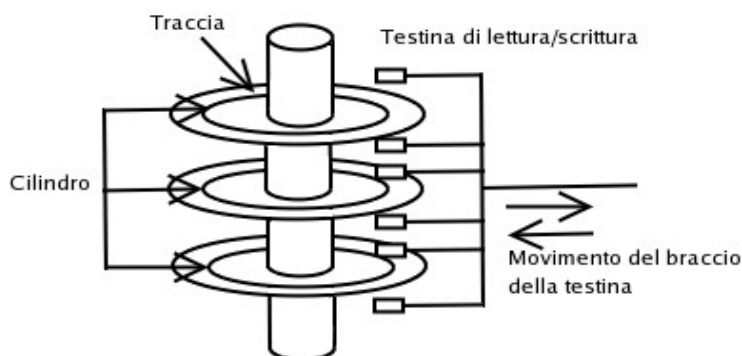
## Dischi magnetici e SSD

Fra le memorie di massa, presenti in un computer, particolare importanza assumono gli hard disk che, grazie principalmente alle loro caratteristiche peculiari quali alta velocità e capacità, sono i candidati più accreditati, nella categoria delle memorie di massa, per essere utilizzati in modo

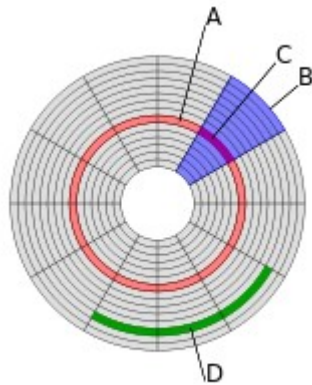
intensivo nell'attività quotidiana, vuoi per conservare dati sui quali operare frequentemente, vuoi per conservare lo stesso sistema operativo.



Un HD è composto da uno o più dischi di alluminio (fra l'altro è questo il motivo per cui è stato chiamato hard disk, in contrapposizione agli, ormai scomparsi, floppy disk in cui il disco non era di alluminio), rivestiti di sostanza magnetizzabile, in rotazione. La testina di lettura/scrittura (una per ogni superficie), è sospesa sopra le superficie, sostenuta da un cuscinio d'aria generato dalla rotazione stessa. In scrittura quando la corrente arriva alla testina, viene magnetizzata la superficie sotto la testina e le particelle si allineano in un verso o in un altro in relazione alla corrente ricevuta (positiva/negativa). In lettura quando la parte magnetizzata passa sotto la testina viene indotta una corrente positiva o negativa a seconda dell'orientamento delle particelle e si può rileggere quanto conservato.



Una sequenza di bit scritta, mentre il disco compie una rotazione completa, si chiama **traccia**. Una traccia non è un *solco* nel HD, ma un anello di sostanza magnetizzata, distinto dal successivo da un piccolo spazio. Ogni traccia è divisa in **settori** di lunghezza fissa (512 byte) preceduti, ognuno, da una parte non utilizzata che serve alla testina per sincronizzarsi prima della lettura/scrittura (intersector **gap**). I dati sono seguiti da una serie di informazioni utilizzate per il controllo/correzione degli errori. Il braccio della testina si muove radialmente rispetto alla pila di dischi, permettendo così il raggiungimento di qualsiasi parte del disco. L'insieme delle tracce che si trovano, in un certo momento, sotto le testine di lettura/scrittura che si muovono in modo solidale, viene chiamato **cilindro**.

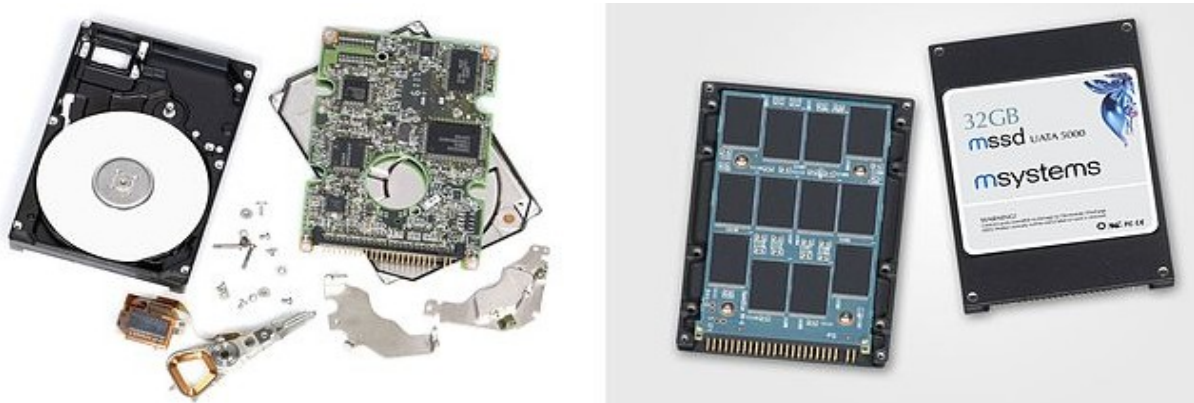


- A) Traccia
- B) Settore
- C) Traccia del settore
- D) Cluster

Prima di poter utilizzare un disco è necessario avviare la *formattazione* fisica, o *a basso livello*, dell'unità. È durante tale formattazione che viene effettuata una mappatura del disco, creando le tracce, i settori e le informazioni che servono per rintracciare i dati conservati nel disco, che quindi consentono l'accesso al disco stesso. La presenza di spazi vuoti fra le zone mappate porta come conseguenza che la capacità di stoccaggio teorica di un HD è diversa dalla capacità formattata che è quella effettivamente utilizzabile. Quest'ultima è inferiore di circa il 15% della capacità non formattata. Per motivi di efficienza i settori vengono raggruppati in cluster che rappresentano l'unità di allocazione: un file su memoria di massa occuperà almeno un cluster.

Per *formattazione* logica, o *ad alto livello*, si intende invece la generazione da parte del Sistema Operativo delle tabelle di allocazione dello spazio, necessarie per gestire la conservazione dei file sul supporto.

Recentemente è stata introdotta una nuova tipologia di memoria di massa basata, al posto delle unità magnetizzabili, sui semiconduttori: le unità a stato solido **SSD** (Solid State Drive):



In realtà, come si può notare dall'immagine di confronto, negli SSD non c'è la presenza di alcun disco, il nome di *disco a stato solido*, termine spesso utilizzato, deriva dal fatto che questo tipo di unità assume le funzioni di un HD tradizionale. In raffronto ad un HD tradizionale offre alcuni vantaggi: rumorosità assente non essendoci parti meccaniche in rotazione, tempi di accesso ridotti, minore produzione di calore, minore possibilità di rottura e maggiore resistenza agli urti. Queste ultime caratteristiche ne hanno fatto l'unità di archiviazione predefinita nei dispositivi mobili. La giovinezza della tecnologia evidenzia, in confronto agli HD, alcuni svantaggi: maggiore costo e minore durata dovuta alla quantità limitata di operazioni di scrittura consentite. La maturazione della tecnologia porterà sicuramente all'annullamento dei suddetti svantaggi.

In un sistema Linux è possibile conoscere il modo in cui l'hard disk (qualunque sia la tecnologia con cui è costruito) viene visto dal sistema, ovvero quella che viene chiamata la *geometria del disco*:

```
# sfdisk -l

Disk /dev/sda: 14593 cylinders, 255 heads, 63 sectors/track
Units = cylinders of 8225280 bytes, blocks of 1024 bytes, counting from 0
...
```

utilizzando il comando `sfdisk` (possibile per l'utente *root* e quindi o si utilizza, come nell'esempio, l'account di root evidenziato dal prompt #, o si utilizza il comando `sudo` per eseguirlo, come: `sudo sfdisk -l`) con il parametro `-l`, è possibile vedere la cosiddetta geometria CHS (Cylinders, Heads, Sectors): nel caso dell'esempio 14593, 255, 63. Come evidenziato, nel manuale in linea, alla pagina del comando “*Questa non è necessariamente la geometria fisica del disco*” ma quella utilizzata per l'accesso al disco.

## Hard Disk, interfacce per collegamenti, device Linux

Un dispositivo qualunque collegato al computer (disco, stampante ...) è sempre composto da una parte meccanica e da una parte elettronica che ne controlla il funzionamento. La parte elettronica, chiamata appunto controller, è residente in genere in una scheda a cui è collegato, per mezzo di cavi, il dispositivo meccanico. I primi hard disk erano chiamati *hardcards* ad indicare il fatto che sia la parte meccanica che quella elettronica risiedevano in una scheda inserita in uno slot di espansione della scheda madre.

Successivamente, per motivi diversi non ultimo il fatto che un HD su scheda è soggetto a vibrazioni che lo possono danneggiare, il controller viene integrato e assemblato assieme al disco e il tutto può trovare alloggiamento in uno spazio, ricavato nel case del computer, per i drive. Il collegamento con la scheda madre è assicurato da un cavo che congiunge l'unità direttamente con il BUS: è quella tecnologia nota come interfaccia IDE/ATA (Integrated Drive Electronics).



Nella scheda madre dei computer trovano posto due connettori IDE/ATA (AT Attachment chiamati così perché la prima volta furono usati nel computer IBM PC/AT), uno per il canale primario (*primary*) ed uno per il canale secondario (*secondary*). Ad ognuno di questi connettori è possibile collegare, per mezzo di uno speciale cavo ad 80 conduttori, due dispositivi (hard disk, lettore CD-ROM/DVD, masterizzatore). Un dispositivo è montato come *master*, l'altro come *slave* ad indicare l'ordine di scansione dei dispositivi collegati.

Una configurazione abbastanza comune potrebbe prevedere, per esempio, l'hard disk su cui è

conservato il sistema operativo configurato come primary master e il canale secondario riservato per unità ottiche (lettore CD-ROM/DVD).

In Linux ad ogni dispositivo è associato uno speciale file, chiamato *device* per mezzo del quale si ha accesso al dispositivo stesso. Si tratta di file contenuti nella directory `/dev`. I dispositivi IDE sono riconosciuti con il nome `hd`, per cui si avrà:

<i>Device</i>	<i>Dispositivo associato</i>
<code>/dev/hda<math>\times</math></code>	Primary master
<code>/dev/hdb<math>\times</math></code>	Primary slave
<code>/dev/hdc<math>\times</math></code>	Secondary master
<code>/dev/hdd<math>\times</math></code>	Secondary slave

La  $\times$  finale, presente in tutti i device elencati, è un numero associato alle partizioni in cui è diviso il dispositivo.

Nei computer più moderni i connettori di tipo IDE sono stati sostituiti dai connettori SATA (Serial ATA, Serial Advanced Technology Attachment).



La connessione è affidata ad un cavo meno ingombrante (15 contatti) rispetto a quello utilizzato per la connessione IDE/ATA (nell'immagine il confronto fra i due cavi). Oltre che per il minor ingombro il collegamento Serial ATA è preferito per la maggior velocità di trasferimento dati e la possibilità di collegamento *a caldo* (a computer acceso). In Linux ai dispositivi di questo tipo sono associati device riconosciuti con il nome `sd` e, quindi, si avrà `/dev/sda`, `/dev/sdb` ecc...

## Partizioni

Una memoria di massa può essere suddivisa in uno o più dischi logici chiamati partizioni ognuna delle quali può contenere un Sistema Operativo diverso. Il primo hard disk presente nei personal computer aveva una capacità di 10Mb e, chiaramente, non c'era alcuna necessità di suddividerlo in più parti. Anche in questi casi è necessario tuttavia dichiarare che si vuole utilizzare una sola partizione che comprenda tutto l'hard disk. In pratica definire una partizione vuol dire stabilire i confini, in termini di cilindro iniziale e finale, che delimitano l'area che il sistema operativo può utilizzare per conservare i dati, anzi, per essere più precisi, le coordinate CHS di inizio e fine.

```
# sfdisk -l
```

```
Disk /dev/sda: 14593 cylinders, 255 heads, 63 sectors/track
Units = cylinders of 8225280 bytes, blocks of 1024 bytes, counting from 0
```

```

Device Boot Start      End    #cyls   #blocks  Id System
/dev/sda1  *          0+    7582    7583-   60910416  c  W95 FAT32 (LBA)
/dev/sda2             7583+  14592    7010-   56307794+  5  Extended
/dev/sda5             7583+   9163-   1580-   12691349+  83  Linux
/dev/sda6             9163+   9408    246-   1975963+  82  Linux swap / Solaris
```

```
/dev/sda7      9409+  14592    5184-  41640448+  83  Linux
```

I simboli + e - associati ai cilindri che delimitano le partizioni, indicano che si tratta di valori approssimati. Una stima sulla posizione esatta di inizio e fine partizione andrebbe effettuata, specificando come unità di misura il settore al posto del cilindro, aggiungendo `-u S` alla riga di comando.

Osservando l'output completo del comando `sfdisk` si può notare che il disco di esempio (`/dev/sda`), è diviso in diverse partizioni. Per esempio l'ultima partizione `/dev/hda7` comincia dal cilindro 9409 e termina al cilindro 14592.

L'esigenza di suddividere un disco in più partizioni è la conseguenza di diversi fattori. Solo per citarne alcuni:

- ➔ A differenza dei primi dischi montati sui computer, gli attuali hanno capacità molto elevate. La suddivisione in più partizioni permette un raggruppamento più razionale dei dati conservati.
- ➔ La suddivisione di un disco in, per esempio, due partizioni una per il sistema operativo e una per i dati, permette, per esempio, di aggiornare il sistema operativo, anche riformattando la partizione, senza perdere i dati che risiedono in una partizione diversa non interessata dalle modifiche.
- ➔ La necessità o curiosità di provare più sistemi operativi porta a destinare, ad ogni sistema operativo installato, una o più partizioni distinte. Ogni sistema operativo infatti, per esempio, organizza lo spazio disponibile in modo diverso da qualsiasi altro.

Ogni sistema operativo rende disponibile un proprio strumento per generare le partizioni. Le informazioni relative alle partizioni in cui è stato suddiviso un disco vengono scritte nel primo settore del disco. Il settore 1 del cilindro 0 della testina 0, di 512 byte, chiamato **MBR** (Master Boot Record) contiene, fra l'altro, la **tabella delle partizioni** in cui sono conservate, appunto, tali informazioni.

Il modo in cui sono gestite le partizioni è rimasto praticamente invariato dai primi personal computer. Nella tabella delle partizioni c'è posto per la descrizione di soltanto quattro partizioni. Tale numero, esagerato ai tempi della nascita dei primi personal (si è già accennato al fatto che il primo disco aveva una capacità di 10Mb), si è rivelato insufficiente in rapporto alle capienze dei dischi attuali e alle esigenze di partizionamento di un sistema multi-boot. Per problemi legati alla compatibilità con l'hardware e il software precedente la limitazione delle quattro partizioni si è aggirata in questo modo:

- ➔ In un disco possono esserci solo quattro **partizioni primarie** di cui solo una può essere attiva. La partizione attiva è la partizione da cui viene effettuato il boot.
- ➔ Una delle partizioni ammissibili può essere una **partizione estesa**. Questo è un sotterfugio per aggirare il limite delle quattro partizioni senza modificare la tabella delle partizioni. La partizione estesa è, sostanzialmente, un *contenitore* di partizioni.
- ➔ Nella partizione estesa possono essere generate tante **partizioni logiche** quante ne servono. In realtà un limite al numero delle partizioni logiche possibili esiste (24) ma è talmente elevato da non costituire un vero e proprio limite.

Nella tabella delle partizioni è presente la descrizione delle partizioni primarie, quando invece si



tratta di partizione estesa, la descrizione, contiene l'indicazione di dove si trova su disco la tabella della partizione estesa che, a sua volta, contiene la descrizione della prima partizione logica e l'indicazione di dove si trova la prossima tabella della partizione estesa con la descrizione della seconda partizione logica e così via.

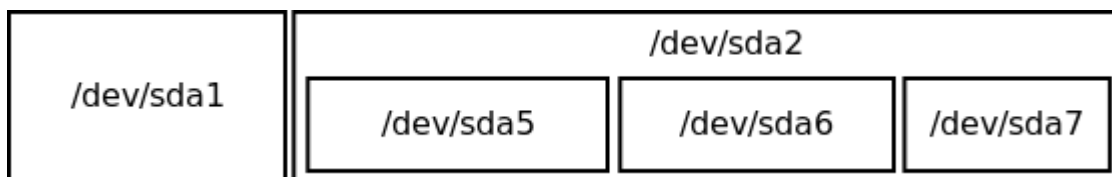
L'esempio di output di `sfdisk` trattato in precedenza, per quanto riguarda le partizioni presenti

```

...
  Device Boot Start      End  #cyls   #blocks  Id System
 /dev/sda1  *        0+    7582   7583- 55793745  c W95 FAT32 (LBA)
 /dev/sda2          7583+  14592   7010- 56307794+  5 Extended
 /dev/sda5          7583+   9163-  1580- 12691349+  83 Linux
 /dev/sda6          9163+   9408    246-  1975963+  82 Linux swap / Solaris
 /dev/sda7          9409+  14592   5184- 41640448+  83 Linux

```

può essere riassunto, per maggiore chiarezza, graficamente



- ➔ `/dev/sda1` è una partizione primaria, la presenza di `*` nella colonna Boot indica, inoltre, che si tratta della partizione attiva.
- ➔ `/dev/sda2` è una partizione estesa che va dal cilindro 7583 al cilindro 14592 (terza riga riportata dell'esempio)
- ➔ `/dev/sda5`, `/dev/sda6` e `/dev/sda7` sono partizioni logiche contenute nella `/dev/sda2`. Infatti la `/dev/sda5` comincia dal cilindro 7583 e la `/dev/sda7` termina nel cilindro 14592 che sono, appunto, i confini della `/dev/sda2`.

Si sarà notato che la numerazione delle partizioni logiche comincia da 5 (`/dev/sda5` è la prima) nonostante non esistano le partizioni 3 e 4. In realtà la cosa è meno strana di quello che sembra: basta pensare al fatto che ci possono essere 4 partizioni primarie di cui una, che potrebbe essere la quarta, estesa. Nella partizione estesa possono esserci partizioni logiche la cui numerazione comincia, quindi, da 5.

## Master Boot Record

Il primo settore di un disco, i primi 512 byte, sono estremamente importanti: la tabella delle partizioni, che permette l'accesso al disco, è registrata in questo settore. Oltre alla tabella delle partizioni nel MBR è registrato il Master Boot Code: un piccolo programma che viene eseguito dal BIOS per avviare la fase di boot. Questo piccolo programma eventualmente è in grado di trasferire l'esecuzione al programma di boot registrato nel settore iniziale della partizione usata per l'avvio.

Chiaramente il meccanismo evidenziato permette, anche in presenza di più partizioni, l'avvio di un solo sistema operativo. Se nel computer sono installati più sistemi operativi il minimo che si richiede è avere la possibilità di scegliere quale sistema avviare fra quelli installati nel disco. Per provvedere a questa necessità sono stati sviluppati i boot-loaders. Un boot-loader è un programma che all'avvio della macchina mostra un menù dei sistemi operativi installati e, in rapporto alla scelta effettuata dall'utente, cede il controllo al sistema scelto. Il boot loader installato da Linux è GRUB (GRand Unified Bootloader).

In ragione dell'importanza del MBR è opportuno effettuare copie di backup in modo da ripristinarlo in caso di malfunzionamenti:

```
# dd if=/dev/sda of=mbr.out count=1 bs=512
```

Il comando `dd` (Disk Dump) permette di ottenere una copia esatta, byte per byte, di una parte di un dispositivo (dumping del dispositivo). Il parametro `if` (Input File) specifica il file da dove copiare (il file `/dev/sda` associato all'HD), `of` (Output File) rappresenta il file destinazione, dove copiare (in questo caso un file che sarà salvato nella directory corrente e che avrà nome `mbr.out`). I successivi parametri indicano la quantità di byte da copiare: 1 blocco (`count=1`) di 512 byte (`bs`, Block Size), giusto la dimensione del MBR.

```
# sfdisk -d /dev/sda > sda.out
```

Il parametro `-d` di `sfdisk` permette di effettuare un dumping della tabella delle partizioni quando esistono partizioni logiche e le informazioni relative non sono presenti nel MBR. L'output potrà essere successivamente utilizzato da `sfdisk` per il ripristino.

I due piccoli file generati possono essere conservati in un unità di backup e, in caso di necessità, la situazione precedente può essere ripristinata utilizzando gli stessi comandi:

```
# dd if=mbr.out of=/dev/sda
# sfdisk < sda.out
```

Non è inutile sottolineare il significato delle ultime due operazioni descritte: si sta sovrascrivendo una parte importante del disco di avvio e si sta manipolando la tabella delle partizioni. È necessario porre la massima attenzione quando si usano comandi di tale importanza: eventuali errori commessi avrebbero, come conseguenza, effetti che comprometterebbero la funzionalità del disco.

Tutti i comandi specificati, come evidenziato dal prompt `#`, vanno impartiti con i diritti di `root`.

## Il boot loader GRUB

GRUB in parte si sostituisce al Master Boot Code, in parte si installa subito dopo il MBR e prima della prima partizione. L'operazione viene effettuata in automatico quando si installa un sistema Linux. Qualche sistema operativo non-Linux cancella questa parte di disco e, quando ciò capita, è necessario reinstallare GRUB (`grub-install /dev/sda`). La stessa riga di comando va eseguita qualora si tratti di riparare un sistema danneggiato o di sostituire file di configurazione persi.

GRUB presenta, all'avvio della macchina e in caso di presenza di più Sistemi Operativi (se ne esiste uno solo questo viene avviato e non viene presentato alcun menù), un menù da cui si può scegliere quale sistema avviare fra quelli installati nel HD. GRUB si basa su alcuni file di configurazione dislocati in varie directory attraverso i quali se ne possono controllare le azioni.

```
$ less /boot/grub/grub.cfg

#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#
...
```

Il file principale che contiene le informazioni visualizzate nel menù di avvio è `/boot/grub/grub.cfg` ma, come riportato anche nelle prime righe del file, si tratta di un file generato, per esempio, per mezzo del comando `update-grub` (forma semplificata del comando suggerito `grub-mkconfig`) in conseguenza al contenuto di altri file.

Un primo file con le impostazioni generali è `/etc/default/grub`:

```
$ less /etc/default/grub

# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
...
GRUB_TIMEOUT=10
...
```

Le opzioni che possono interessare maggiormente sono:

- ➔ `GRUB_DEFAULT` che gestisce il sistema da avviare per default se l'utente non sposta il cursore per la selezione. Il valore 0 indica la prima voce visualizzata nel menù. Se si vuole modificare il sistema avviato per default, si può specificare la nuova riga (si tenga conto del fatto che il conteggio delle righe comincia da 0).
- ➔ `GRUB_TIMEOUT` che gestisce il tempo di attesa dell'input dell'utente prima che venga avviato il sistema di default. Nell'esempio è impostato a 10 secondi. È ammesso anche il valore 0 (nessun tempo di attesa) e -1 (attesa indefinita fino all'azione dell'utente).

Le scelte del menù di avvio sono ricavate dai file presenti in `/etc/grub.d`:

```
$ ls /etc/grub.d/
00_header          10_linux          20_memtest86+    30_uefi-firmware  41_custom
05_debian_theme    20_linux_xen      30_os-prober     40_custom         README
```

Il file `README` spiega il significato di tali file:

```
$ less /etc/grub.d/README
All executable files in this directory are processed in shell expansion order.

00_*: Reserved for 00_header.
10_*: Native boot entries.
20_*: Third party apps (e.g. memtest86+).

The number namespace in-between is configurable by system installer and/or
administrator. For example, you can add an entry to boot another OS as
01_otheros, 11_otheros, etc, depending on the position you want it to occupy in
the menu; and then adjust the default setting via /etc/default/grub.
```

I file sono script elaborati nell'ordine del numero contenuto nel nome e se si vuole modificare l'ordine inserendo la voce di un sistema personalizzato si può usare come numero d'ordine 01, 11 ecc... anche se il modo più semplice di modificare il sistema da avviare per default è di intervenire in `/etc/default/grub`.

Dopo eventuali modifiche del contenuto dei file suddetti è necessario rigenerare il file `/boot/grub/grub.cfg`, con il comando `update-grub`, in modo che sia sincronizzarlo con le

modifiche effettuate.

## Riferimenti

Risorse consultate per la realizzazione di queste note e disponibili per approfondimenti degli argomenti trattati:

- ➔ Auto-documentazione di Linux: `man sfdisk, info grub`
- ➔ <http://it.wikipedia.org> da cui sono state tratte pure le foto usate in questo documento
- ➔ <https://help.ubuntu.com/community/Grub2> documentazione sulla configurazione di GRUB
- ➔ *Architettura dei computer* di Andrew S. Tanenbaum



### **Creative Commons Public License**

#### **Attribuzione-NonCommerciale-CondividiAlloStessoModo 3.0 Italia**

Tu sei libero:

di distribuire, comunicare al pubblico, rappresentare o esporre in pubblico l'opera,  
di creare opere derivate

Alle seguenti condizioni:

- \* **Attribuzione.** Devi riconoscere la paternità dell'opera all'autore originario.
- \* **Non commerciale.** Non puoi utilizzare quest'opera per scopi commerciali.
- \* **Condividi sotto la stessa licenza.** Se alteri, trasformi o sviluppi quest'opera, puoi distribuire l'opera risultante solo per mezzo di una licenza identica a questa.

In occasione di ogni atto di riutilizzazione o distribuzione,  
devi chiarire agli altri i termini della licenza di quest'opera.

Se ottieni il permesso dal titolare del diritto d'autore,  
è possibile rinunciare a ciascuna di queste condizioni.

Le tue utilizzazioni libere e gli altri diritti  
non sono in nessun modo limitati da quanto sopra.

Questo è un riassunto in lingua corrente dei concetti chiave della licenza completa  
(codice legale) che è disponibile alla pagina web:

<http://creativecommons.org/licenses/by-nc-sa/3.0/it/legalcode>