

L'hard disk

L'amministrazione dei dischi puo' essere riassunta nelle seguenti operazioni di base:

- La formattazione
- Il partizionamento
- La creazione del filesystem
- Il mount

L'hard disk

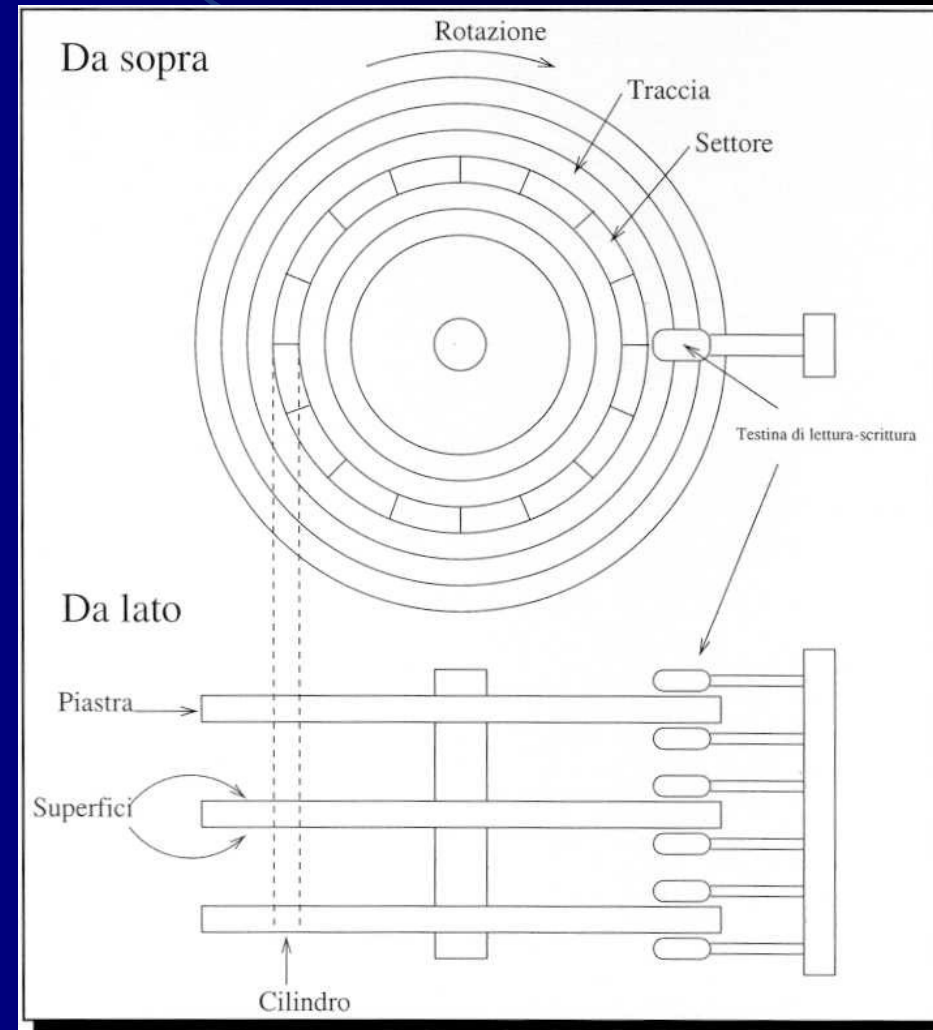
Le superfici sono divise in anelli concentrici, chiamati tracce, e a loro volta divise in settori.

Per indirizzare un punto preciso dell'hard disk si potrebbe dire "piastra 3, traccia 5, settore 7".

Di solito il numero dei settori è lo stesso per tutte le tracce, ma in qualche hard disk sono presenti più settori nelle tracce più esterne:

Tipicamente, un settore contiene 512 byte di dati.

L'insieme delle tracce sovrapposte forma un cilindro



Dischi IDE e SCSI

Esistono vari tipi di dispositivi per gestire i dischi

Ad esempio, i dischi IDE e SCSI si differenziano per i controller e i bus utilizzati per la gestione e il trasferimento dei dati

Adattatori SCSI (Small Computer System Interface)

SCSI ha avuto il suo principio nel mondo dei Macintosh Apple.

Gli adattatori SCSI permettono di concatenare assieme sette diversi tipi di periferiche (dischi fissi, scanner, stampanti e così via), tutti controllati da una scheda adattatrice.

Un adattatore SCSI-2, per esempio, fornisce 8 diversi indirizzi per le periferiche, uno dei quali viene usato dallo stesso adattatore.

La flessibilità della SCSI va anche oltre. Dei bridge controller sofisticati possono collegare sette periferiche per ciascun indirizzo di periferica SCSI.

Ciò significa che a un singolo adattatore SCSI si possono collegare fino a 49 periferiche.

Gli elementi a sfavore che hanno condizionato molto la sua diffusione sono dovuti ai costi e alla difficoltà d'uso.

Adattatori IDE (Integrated Drive Electronics)

- Originalmente utilizzati dai dischi fissi del PC IBM AT, il primo personal computer a 16 bit.
- Il significato della sigla è molto appropriato. Queste unità infatti collocano la maggior parte dell'elettronica del controller sul supporto del disco e non su una scheda separata.
- Ne risulta un modello che riduce i costi di fabbricazione, e ha contribuito a farne uno standard affermato nei PC.
- Anche lo standard IDE ha avuto vari miglioramenti col tempo. Una delle versioni più recenti e diffuse viene detta EIDE (Enhanced Integrated Drive Electronics).

Adattatori EIDE

- EIDE permette di avere un controller primario e uno secondario, a cui di possono collegare fino a 4 periferiche.

La formidabile diffusione che questo tipo di controller ha avuto è dovuto al fatto che l'IDE era in linea con le specifiche BIOS che erano allora presenti nella maggior parte dei sistemi.

In pratica, la maggioranza dei sistemi sono dotati di controller IDE integrati sulla scheda madre, che non richiedono nessuna installazione. Inoltre risultano molto meno cari dei controller SCSI.

Possiamo individuare gli svantaggi, oltre che nel limitato numero di periferiche supportate, anche nella velocità di trasferimento dei dati che nonostante tutti i vari miglioramenti, rimane al di sotto dello standard SCSI.

Device file

Dato che Linux spesso non conosce la geometria reale di un disco, i suoi filesystem non provano neanche a mantenere i file all'interno di un singolo cilindro

Provano invece ad assegnare ai file settori numerati sequenzialmente, cosa che dà quasi sempre una performance simile.

Ciascun hard disk è rappresentato da un file di device separato.

Notare che i file di device per gli hard disk danno accesso al disco intero, senza considerare le partizioni e di solito vengono usati solo per accedere al loro master boot record.

La formattazione

La formattazione (a basso livello) è il processo con cui si segnano elettricamente e permanentemente sul mezzo magnetico le tracce e i settori.

La terminologia a questo proposito è piuttosto confusa: in MS-DOS, la parola formattazione viene usata anche per il processo di creazione di un filesystem (ad alto livello); qui i due processi sono spesso combinati, specialmente per i floppy.

In ambiente UNIX, i due processi vengono chiamati formattazione e creazione di un filesystem.

Per i dischi la formattazione viene fatta in realtà durante la costruzione

La formattazione

Durante la formattazione si possono incontrare dei punti danneggiati sul disco, che vengono chiamati *blocchi danneggiati* o *settori danneggiati*.

Molti controller di dischi individuano automaticamente i blocchi danneggiati, e tentano di recuperarli usando al loro posto un blocco speciale, integro, riservato per questo scopo, che è invisibile al sistema operativo.

Se il numero di blocchi danneggiati diventa troppo grande, anche su dischi di questo tipo si può avere perdita di dati

Il partizionamento

Un singolo hard disk puo'essere diviso fino ad un massimo di quattro partizioni di tipo primarie.

Per ottenere maggiore flessibilita' di partizionamento e' stata creata la partizione di tipo estesa.

La partizione di tipo estesa puo' essere creata in sostituzione di una qualunque delle quattro partizioni primarie e a sua volta puo' contenere al massimo quattro partizioni logiche (in realta' con i recenti fdisk e' possibile creare piu' di quattro partizioni logiche).

Le informazioni sul partizionamento di un hard disk si trovano nel suo primo settore (cioè, il primo settore della prima traccia della prima superficie del disco).

Il primo settore si chiama *master boot record* (MBR) del disco: è il settore che il BIOS legge quando la macchina viene accesa.

Il partizionamento

Il master boot record contiene un piccolo programma che

- legge la tabella delle partizioni,
- controlla quale partizione è attiva (cioè quale è contrassegnata avviabile) e
- legge il primo settore di quella partizione, chiamato *boot sector* (settore di avvio) della partizione

Il boot sector contiene un altro programmino che legge la prima parte del sistema operativo contenuto in quella partizione (sempre che sia avviabile) e lo avvia.

Il partizionamento

Le tabelle delle partizioni (quelle nell'MBR, e quelle per le partizioni estese) contengono un byte per partizione che identifica il tipo di partizione.

In questo modo si cerca di identificare il sistema operativo che usa la partizione, o il modo in cui viene usata.

Lo scopo è evitare che due sistemi operativi usino accidentalmente la stessa partizione.

Non c'è nessuna agenzia di standardizzazione che specifica cosa significhi ciascun valore di questo byte, ma alcuni valori comunemente accettati sono riportati nel comando fdisk stesso.

Il partizionamento

Ci sono molti programmi per creare e rimuovere partizioni. La maggior parte dei sistemi operativi ne ha uno proprio, ed è una buona idea usare quello del sistema che si usa, in caso faccia qualcosa di insolito che gli altri non fanno. In Linux si usa fdisk.

La tabella delle partizioni si puo' visualizzare come segue

```
> fdisk -l /dev/hda
```

```
Disk /dev/hda: 15 heads, 57 sectors, 790 cylinders Units = cylinders of
855 * 512 bytes
```

| Device | Boot | Begin | Start | End | Blocks | Id | System |
|-----------|------|-------|-------|-----|---------|----|--------------|
| /dev/hda1 | | 1 | 1 | 24 | 10231+ | 82 | Linux swap |
| /dev/hda2 | | 25 | 25 | 48 | 10260 | 83 | Linux native |
| /dev/hda3 | | 49 | 49 | 408 | 153900 | 83 | Linux native |
| /dev/hda4 | | 409 | 409 | 790 | 163305 | 5 | Extended |
| /dev/hda5 | | 409 | 409 | 744 | 143611+ | 83 | Linux native |
| /dev/hda6 | | 745 | 745 | 790 | 19636+ | 83 | Linux native |

Creazione del file system

Un filesystem comprende i metodi e le strutture di dati usate da un sistema operativo per tenere traccia dei file su un hard disk o su una sua partizione, cioè il modo in cui i file sono organizzati sui dischi.

La maggior parte dei programmi operano su un filesystem, e quindi non funzionano su una partizione che non ne contiene uno (o ne contiene uno del tipo sbagliato).

Prima che si possa usare un disco o una partizione come filesystem, deve essere inizializzato, e bisogna preparare le strutture di dati per l'archiviazione. Questo processo si chiama creazione di un filesystem.

Creazione del file system

La maggior parte dei tipi di filesystem UNIX hanno una struttura generale simile, anche se i dettagli esatti cambiano abbastanza. I concetti centrali sono quelli di superblocco ed inode.

Il *superblocco* contiene delle informazioni sull'intero filesystem, come la sua dimensione (l'informazione esatta dipende dal filesystem).

Come visto a teoria l'inode contiene il numero dei diversi blocchi dati, che vengono usati per immagazzinare i dati del file.

Tipi di filesystem

Linux supporta diversi tipi di filesystem:

- minix

Il più vecchio, e si presume il più affidabile, ma piuttosto limitato (mancano alcuni time stamp, e i nomi di file sono al massimo di 30 caratteri) e di capacità ristrette (al massimo 64~MB per filesystem).

- xia

Una versione modificata del filesystem minix, che alza i limiti sulla lunghezza dei nomi di file e sulla dimensione dei filesystem, ma non introduce nuove caratteristiche. Non è molto conosciuto, ma si dice funzioni molto bene.

La creazione del file system

- ext2
Il più completo dei filesystem nativi di Linux, e al momento anche il più usato. E' disegnato per essere compatibile in avanti, in modo che nuove versioni del codice del filesystem non abbiano bisogno di rifare i filesystem esistenti.
- ext
Una versione più vecchia dell'ext2 che non era compatibile in avanti.

La creazione del file system

Oltre a questi vengono supportati filesystem di altri sistemi operativi, per rendere più semplice lo scambio di file:

- msdos

Compatibilità con i filesystem FAT di MS-DOS (e OS/2 e Windows NT).

- umsdos

Estende il driver del filesystem msdos sotto Linux, in modo da avere i nomi di file lunghi, i permessi, i link e i file di device, e da assegnare ciascun file ad un utente. In questo modo è possibile usare un normale filesystem msdos come se fosse uno Linux, eliminando la necessità di avere una partizione separata per Linux.

La creazione del file system

- **nfs**
Un filesystem di rete che permette di condividere un filesystem tra vari computer per avere da essi un accesso più facile ai file.
- **hpfs**
Il filesystem di OS/2.
- **sysv**
Il filesystem SystemV/386, Coherent e Xenix.
- **iso9660**
Il filesystem standard per i CD-ROM: viene supportata automaticamente l'estensione Rock Ridge allo standard per i CD-ROM, che permette di avere i nomi dei file lunghi.

Creazione del file system

I filesystem vengono creati, cioè inizializzati, con il comando ***mkfs***.

In realtà esiste un programma separato per ciascun tipo di filesystem, e ***mkfs*** è solo un'interfaccia che avvia il programma adatto a seconda del tipo di filesystem desiderato. Il tipo viene selezionato con l'opzione ***-t tipofs***.

La struttura del filesystem

- Esiste uno standard dei filesystem Linux, FSSTD versione 1.2 che tenta di impostare uno standard per l'organizzazione dell'albero delle directory nei sistemi Linux.
- Uno standard del genere ha il vantaggio di rendere più agevole la scrittura o il porting del software per Linux, e amministrare le macchine Linux, poiché i file si troveranno nel posto designato.
- Non c'è nessuna autorità che impone a nessuno di uniformarsi allo standard, ma questo ha il supporto della maggior parte, se non di tutte, le distribuzioni Linux.

Non è una buona idea rompere con lo standard FSSTND se non per ragioni molto particolari. Il FSSTND tenta di seguire la tradizione Unix e le tendenze più recenti, rendendo i sistemi Linux familiari per chi ha esperienza con altri sistemi Unix, e viceversa.

La struttura del filesystem

- Il filesystem *root* è specifico per ciascuna macchina (generalmente viene immagazzinato su un disco locale, anche se può trattarsi di un ramdisk o di un disco in rete) e contiene i file necessari per avviare il sistema e per portarlo ad uno stato tale che possa montare gli altri filesystem.
- Il contenuto del filesystem di root sarà quindi sufficiente per la modalità a singolo utente. Conterrà anche gli strumenti per recuperare un filesystem danneggiato o copiare dai backup i file perduti. Il filesystem root dovrebbe generalmente essere piccolo, dato che contiene file estremamente critici, e un filesystem piccolo che viene modificato poco ha migliori possibilità di non venire corrotto.
- Un filesystem di root corrotto in genere significa che diventa impossibile avviare il sistema tranne che con misure eccezionali (ad esempio da un floppy), quindi è meglio non rischiare.
-

La struttura del filesystem

- La directory principale in genere non contiene nessun file, tranne forse l'immagine standard di avvio per il sistema, che di solito si chiama `/vmlinuz`. Tutti gli altri file sono in sottodirectory del filesystem root

La struttura del filesystem

- /usr
- Il filesystem /usr contiene tutti i comandi, le librerie, le pagine man e altri file che non vengono modificati durante le normali operazioni.
- Nessun file in /usr dovrebbe essere specifico per nessuna macchina data, né dovrebbe essere modificato durante il normale uso.
- Questo permette che il file venga condiviso in rete, cosa che può portare ad un risparmio economico dato che permette di risparmiare in spazio disco e può rendere l'amministrazione molto più semplice (basta modificare solo /usr principale quando si aggiorna un'applicazione, e non c'è bisogno di farlo separatamente su ciascuna macchina).
- Anche se il filesystem si trova su un disco locale, può essere montato con accesso a sola lettura, per diminuire le possibilità di corruzione durante un crash.

La struttura del filesystem

- `/usr/local`
- Il filesystem `/usr` è spesso grande, dato che vi sono installati tutti i programmi.
- Tutti i file in `/usr` vengono di solito da una distribuzione di Linux; i programmi installati in locale e il resto vanno sotto `/usr/local`; in questo modo è possibile aggiornare il sistema a una nuova versione della distribuzione, o addirittura ad una distribuzione completamente nuova, senza dover reinstallare tutti i programmi da capo.

La struttura del filesystem

- /var
- Il filesystem var contiene dei file che cambiano, come le directory di spool (per la posta, le news, le stampanti eccetera), i file di log, le pagine man formattate e i file temporanei..
- contiene i dati che vengono modificati quando il sistema lavora normalmente, è specifico per ciascun sistema, cioè non viene condiviso in rete con altri computer.
-

La struttura del filesystem

- /home
- Il filesystem /home contiene le home directory degli utenti, cioè tutti i dati sul sistema.

Separare le home directory su un albero o su un filesystem separato rende molto più semplici i backup: le altre parti in genere non hanno bisogno di backup, o almeno non frequentemente (cambiano poco nel tempo).

Una /home grande potrebbe dover essere separata in vari filesystem, cosa che richiede l'aggiunta di sottolivelli, ad esempio /home/students e /home/staff.

La struttura del filesystem

- /bin

Contiene i comandi necessari durante il boot del sistema che possono anche essere usati dagli utenti normali dopo il boot.

- /sbin

Come /bin, ma i comandi non sono intesi per gli utenti normali, anche se questi li possono usare se necessario e se hanno i permessi.

- /root

La home directory dell'utente root.

La struttura del filesystem

- /lib

Le librerie condivise necessarie ai programmi.

Molto importante e' /lib/modules che contiene i moduli del kernel caricabili, specialmente quelli che sono necessari per avviare il sistema quando lo si sta recuperando da un problema (ad esempio i driver di rete e dei filesystem).

- /tmp

I file temporanei. I programmi che vengono avviati dopo il boot dovrebbero usare /var/tmp, non /tmp, dato che il primo si trova probabilmente in un disco con più spazio.

La struttura del filesystem

- /boot

I file usati dal boot manager, come LILO. Le immagini del kernel spesso vengono tenute qui invece che nella directory root. Se ci sono diverse immagini di kernel, la directory può facilmente crescere parecchio, e spesso può essere meglio tenerla in un filesystem separato.

- /mnt

Il punto di mount dove l'amministratore di sistema può montare temporaneamente delle directory. /mnt può essere diviso in sottodirectory (esempio /mnt/dosa può essere il floppy che usa un filesystem MS-DOS, e /mnt/exta lo stesso con un filesystem ext2)

La struttura del filesystem

- /dev

La directory dev contiene degli speciali file di device, uno per ciascun dispositivo. I file di device vengono chiamati usando delle speciali convenzioni, che sono descritte nell'elenco dei dispositivi

I file di device vengono creati durante l'installazione, ma possono essere creati anche in seguito usando lo script chiamato MAKEDEV.

La struttura del filesystem

- /proc

Il filesystem /proc contiene un filesystem virtuale:

In realtà non è per niente un filesystem, anche se gli somiglia. /proc permette di avere facile accesso ad alcune strutture di dati del kernel, come la lista dei processi (da cui il nome).

Fa apparire queste strutture di dati come un filesystem, e quel filesystem può essere manipolato con i normali strumenti di manipolazione di file.

Montaggio di un filesystem

Prima di poter utilizzare un filesystem, bisogna montarlo (mount)

Dato che tutti i file in UNIX si trovano in un singolo albero di directory, l'operazione di mount farà sembrare che il nuovo filesystem sia contenuto in una sottodirectory esistente di qualche filesystem già montato. Il comando *mount* accetta due argomenti:

- il file di device corrispondente al disco o alla partizione che contiene il filesystem,
- e la directory sotto cui montarlo.

Esempio mount

```
> mount /dev/hda2 /home  
> mount /dev/hda3 /usr
```

Dopo questi comandi il contenuto dei due filesystem sembra essere in /home e /usr.

Si può dire che ``/dev/hda2 è montato su /home'', e la stessa cosa per /usr.

Per vedere i due filesystem basta guardare nella directory su cui è stato montato, come se fosse una qualsiasi altra directory. Notate la differenza tra il file di device /dev/hda2 e la directory su cui viene montato, /home. Il file di device dà accesso ai contenuti fisici del disco, la directory su cui viene montato dà accesso ai file che vi si trovano. La directory su cui viene montato un filesystem si chiama punto di mount.

Il mount

- Abbiamo visto che Linux supporta molti tipi di filesystem;

mount cerca di indovinare il tipo di quello che si sta montando. Si può anche usare l'opzione *-t tipofs* per specificare direttamente il tipo: talvolta è necessaria, dato che l'euristica usata da `cmdmount` non sempre funziona. Ad esempio per montare un floppy MS-DOS si usa il seguente comando:

> *mount -t msdos /dev/fd0 /floppy*

La directory di mount può anche non essere vuota, ma deve esistere.

L'opzione *-r* consente un mount in sola lettura. E' indispensabile usare i mount a sola lettura per i mezzi non scrivibili, come i CD-ROM.

Il mount

I filesystem che devono essere montati automaticamente al momento dell'avvio vengono specificati nel file

`/etc/fstab`

Quando non serve più che un filesystem sia montato, può essere smontato usando `umount`.

`umount` accetta un solo argomento, o il file di device o il punto di mount. Ad esempio, per smontare le directory dell'esempio precedente, si possono usare i comandi

- > `umount /dev/hda2`
- > `umount /usr`