



Primo corso su GNU/Linux e informatica libera

Il Filesystem

Relatori:

Giorgio “Bodo” Bodini
Roberto “Oliverino” Azzini

Cosa si intende per filesystem:
con il termine filesystem ci si riferisce a due concetti diversi:

- 1) metodo di organizzazione dei dati sui supporti fisici (floppy, cdrom,hd,rete: ext2,ext3, reiserfs...). Decisione iniziale dell'amministratore del sistema in base alle caratteristiche che meglio si adattano all'uso del sistema stesso. Poi diviene trasparente all'utente durante l'uso quotidiano
- 2) modo del SO di organizzare al suo interno e di mostrare all'utente i file e le directory (cosa trovo dove?). Uso quotidiano dell'utente: aprire, salvare, spostare file e directory.

1) iniziamo col prendere in considerazione la prima accezione del termine filesystem

Cenni sui concetti e il funzionamento di memorie di massa (tipo, geometria), partizioni (primarie, estese) e MBR (bootloader in MBR o all'inizio della partizione di boot – attiva o no), file e directory:

Nel computer abbiamo due tipi di memoria:

- la RAM velocissima e volatile usata dal processore per le sue operazioni al volo. Costituita da circuiti integrati. Quando spegnamo la macchina il suo contenuto è perso.
- l'altro tipo è la memoria di massa, molto più lenta ma duratura. Essa è costituita da nastri dischi magnetici o ottici, hard disk e ultimamente anche da RAM non volatili (pen drive USB)

Gli hard disk sono la memoria di massa più comune. Sono costituiti da uno o più piatti (dischi) di materiale trasparente ai campi magnetici (alluminio, plastica, vetro) ricoperti su una o entrambe le facce da ossido di ferro (sì, lo stesso del nastro delle audiocassette...). Questi piatti sono fatti ruotare a velocità (angolare) costante (vedi harddisk da 7500, 10000 RPM).

Le operazioni di registrazione e lettura dei dati sono effettuate da *testine* che possono scorrere dal bordo verso il centro del piatto e viceversa su *tracce* circolari concentriche. Le

tracce magnetiche sono definite dalle testine stesse durante l'operazione di inizializzazione (o formattazione a basso livello) del disco (l'operazione di solito è effettuata dal produttore ma è ripetibile dall'utente con i comandi appropriati).

Le tracce a loro volta sono divise in *settori* di uguale dimensione (angolare) ognuno contraddistinto da un codice di identificazione. I settori sono in pratica i contenitori dei dati. Se i piatti sono più di uno questi sono posti verticalmente uno sopra l'altro sullo stesso asse e ruotano in maniera solidale.

La registrazione e la lettura avvengono attraverso un “pettine” di testine collegate tra loro. L'insieme delle tracce dei vari piatti poste alla stessa distanza dal centro è detta *cilindro*.

Quando le testine sono poste in un dato punto della loro corsa ed il disco sta girando, leggono un cilindro (o meglio, la sequenza dei settori contenuti in un cilindro).

Per accedere ad uno specifico dato dovrò quindi determinare: su quale cilindro posizionare il pettine delle testine, da quale testina leggere, quale settore della traccia selezionare: per questo la geometria di un disco è definita dai parametri *cilindri*, *testine*, *settori* (*cylinders*, *heads*, *sectors*).

Il BIOS deve conoscere (in automatico o per inserimento manuale) questa geometria per poter gestire il disco.

Un disco può essere suddiviso in più parti che verranno trattate dal sistema operativo come dischi separati. Queste suddivisioni sono dette *partizioni*.

In origine le partizioni possibili all'interno di un disco erano al massimo solo quattro (le cosiddette *partizioni primarie*). Per superare questo limite è stato introdotto il concetto di *partizione estesa*. Questa al suo interno può contenere una serie di partizioni virtuali.

Nelle memorie di massa la quantità di dati può essere enorme, dunque va organizzata in strutture che prendono il nome di file e directory.

Un file può avere dimensioni che quasi sempre superano quelle dei settori del disco, quindi viene diviso in parti che verranno collocate nei settori spesso anche in modo non contiguo. Esistono molti modi diversi di collocare queste strutture sui dischi. Ogni modo diverso è un *tipo di filesystem*.

I tipi di filesystem differiscono tra di loro per dimensioni minime e massime, prestazioni, propensione alla velocità piuttosto che all'affidabilità, tipo di supporto utilizzabile...

Ogni partizione ha un solo tipo di filesystem

Ogni disco può avere diverse partizioni con filesystem diversi

“Creare un filesystem” significa formattare (!!! spiegare concetto di formattazione!!!) una partizione (o un intero disco) con un determinato metodo di organizzazione dei file...

principali tipi di Filesystem:

- **minix**: (vecchio) filesystem usato dal sistema operativo omonimo, fu il primo FS usato su GNU/Linux
- **ext**: (vecchio) Extended Filesystem, estensione del FS minix
- **ext2**: (quasi vecchio) Second Extended Filesystem, fino a ieri default per GNU/Linux. Resta il migliore in fatto di velocità e uso della CPU.
- **ext3**: (Journaled) Terza versione di extended filesystem, con funzionalità di journaling e supporto ACL
- **reiserfs**: (Journaled) un “campione” su sistemi dove vengono trattati molti file di piccole dimensioni
- **xfs**: filesystem standard di SGI, da paura!, massimo 18.000 petabyte, supporta journal, quote, ACL
- **jfs**: Journaled File System, utilizzato dai server IBM, massimo 4 petabyte

- **nfs**: Network Filesystem, filesystem di rete per l'accesso a risorse condivise da computer remoti
- **swap**: filesystem per le partizioni di swap di Linux (la riserva per la memoria RAM)
- **iso9660**: il filesystem usato per i cdrom (con estensioni RockRidge (UNIX), Joliet (Windows), ElTorito (bootabili))
- **udf**: Universal Disk Format, filesystem a pacchetti per cd riscrivibili
- **smb**: filesystem per l'accesso a risorse condivise in rete da computer windows attraverso il protocollo SMB (Server Message Block): un protocollo per condividere file, stampanti, porte seriali e comunicazioni (named pipes e altro) tra computer. GNU/Linux accede a queste risorse attraverso la famosa suite di programmi SAMBA
- **fat**: (poi detto fat12): filesystem di MS-DOS (nomi file limitati a 8+3 caratteri) prende il suo nome da File Allocation Table, una struttura su cui è basato, massimo 16MB (!)
- **fat16**: filesystem di MS-DOS e Win3.1 (8+3 caratteri), usa 16 bit per memorizzare il numero dei cluster del disco, da cui il nome; massimo 2GB
- **vfat**: variante di fat16 che ha introdotto il supporto per i nomi lunghi; filesystem di Win95. Tutte le partizioni "fat" create sotto Win95 o successivi sono in realtà partizioni vfat
- **fat32**: filesystem da Win95 SR2 in poi, usa 28 bit per memorizzare il numero dei cluster del disco (non 32... 4 bit sono definiti "reserved"); massimo 2TB
- **ntfs**: New Technology Filesystem, è il file system proprietario di WinNT 2000 e XP, supporta ACL, compressione, criptazione. Sotto GNU/Linux è più sicuro accedere a questo FS in sola lettura, massimo 2TB
- **hfs**: Hierarchical File System, usato nei Mac moderni
- **hpfs**: High Performance File System, utilizzato da IBM OS/2
- **ufs/ffs**: filesystem degli Unix *BSD
- **ofs/ffs**: Old File System/Fast File System, AmigaOS
- ... mille altri: AFS, GFS, CryptoFS, ...

alcuni SO possono utilizzare solo uno o due FS, Linux tutti!

cosa è il journaling:

- cache della scrittura su disco
- giornale delle modifiche al fs
- i dati sono al sicuro anche in caso di crash del sistema e il ripristino è veloce

2) modo del SO di organizzare al suo interno e di mostrare all'utente i file e le directory (cosa trovo dove?).

L'organizzazione del filesystem di GNU/Linux è piuttosto diversa da quella di Windows alla quale probabilmente siete abituati... A prima vista sembra criptica: non ci sono quelle belle cartelle di nome "Programmi" o "Documenti"... Dobbiamo ricordare che UNIX è stato scritto da hacker per gli hacker, per cui "meno lettere possibile, ma con molto significato!". Se notate nessuna directory ha un nome più lungo di quattro lettere e quando si lavora al terminale di un server senza interfaccia grafica, meno si scrive e più si è veloci...

```

/
/root
/bin
/sbin
/boot
/dev
/etc
/home

```

/lib
/mnt
/opt
/proc
/tmp
/var
/usr

la disposizione di directory e file di Unix (e quindi di GNU/Linux) segue lo standard FHS (Filesystem Hierarchy Standard) [con qualche piccola eccezione...(ad es: /proc)]

FHS è una raccolta di requisiti e linee guida per la disposizione di file e directory nei sistemi operativi Unix-like.

Il suo scopo è quello di garantire l'interoperabilità di applicazioni, strumenti di amministrazione, ambienti di sviluppo e scripts, così come una maggiore uniformità nella documentazione di questi strumenti.

Mandrake 9.2, che avete installato sulle vostre macchine negli scorsi incontri, si basa sulla versione 2.2 dell'FHS, ora già superato dalla v2.3...

Unix a differenza di Win ha un unico albero nel quale vengono innestati i rami di ogni periferica che viene “montata”. In Win ogni “disco” (anche se “astratto” [di rete]) invece ha un suo albero separato che parte da \ (root – radice). Infatti Win deriva dal DOS – Disk Operating System, sistema operativo basato sui dischi.

Attenzione a / (slash) e \ (backslash)... (Internet [e non solo...] parla UNIX, è per questo che gli indirizzi URL usano le slash dritte!!!!)

Ogni ramo (o anche sottoramo!!!) può essere un filesystem diverso (periferica diversa o partizione diversa) che si dice “montato” sull'albero!!!

In caso di installazioni toste si usa mettere alcune directory su dischi o partizioni separate (/boot, /home, /var, /usr, /usr/local, /tmp, /var) per mantenere intatti certi dati in caso di guasti hardware o upgrade del SO e mantenere il più leggero possibile il filesystem di base che deve partire nudo in caso di guai...

Il punto in cui si innesta nell'albero un filesystem separato si chiama mountpoint (punto di innesto).

/ (root – la radice) (attenzione!!! non confondere con la directory /root). È il punto di origine di tutto il filesystem. Non deve contenere file (se non il kernel in casi particolari) ma solo le directory stabilite dallo standard FHS

/bin (binaries - binari) contiene i programmi eseguibili fondamentali per la gestione del sistema che possono essere usati sia dall'amministratore (root) che dagli utenti (in /sbin che vedremo dopo vanno i binari di uso esclusivo del superutente). I comandi che impareremo ad usare più tardi sono tutti contenuti in questa directory!

/boot (boot – avvio) contiene tutto ciò che serve al processo di avvio (bootstrap) del sistema: per esempio il kernel (nocciolo del sistema operativo) stesso.

/dev (device – dispositivi) contiene i file speciali di dispositivo, uno per ogni dispositivo fisico del computer. In pratica è come se si trattasse di “driver” dei dispositivi. (mostrare #cat /dev/mouse come esempio) In Unix ogni cosa è un file... (accennare a /dev/zero, /dev/null...)

/etc (-) contiene i dati di configurazione del sistema, specifici del computer locale. Parlare di fstab (poi lo useremo), mtab (idem), exports, printcap, resolve.conf, services, X11/XF86Config...

/home (home – personale) contiene i dati personali di tutti gli utenti del sistema (tranne root che poi vedremo), suddivisi in una directory per ogni utente che ha solitamente come nome lo username dell'utente. Spessissimo si tratta di un filesystem secondario (un altro disco o partizione o condivisione di rete) montato in questo punto (per ragioni di sicurezza e di scalabilità)

/initrd Fuori standard – usata solo da Mandrake – in uso solo durante l'avvio del sistema, poi praticamente vuota.

/lib (libraries – librerie) contiene i file di libreria condivisi necessari per l'avvio del sistema e per i programmi di uso generale che si possono trovare in /bin e /sbin. Le librerie che riguardano solo programmi collocati in /usr non vanno qui... Prevede anche la directory /lib/modules che contiene i moduli caricabili dal kernel (spiegare brevemente kernel monolitico / kernel modulare)

/mnt (mount – innesto) punto di innesto standard per filesystem montati temporaneamente. Normalmente dovrebbe essere vuota e riservata ai montaggi temporanei (come specificato da FHS v2.3) ma finora è stato uso comune (e anche in Mandrake 9.2 è così) montarvi permanentemente i filesystem ospiti /mnt/floppy, /mnt/cdrom, /mnt/removable (zip, usb)... Più tardi vedremo come montare e smontare i filesystem ospiti

/opt (optional applications – applicazioni opzionali) Qui dovrebbero andare applicazioni opzionali ed altri software, ognuno in una sua sottodirectory. I file di configurazione di questi pacchetti dovrebbero poi andare in /etc/opt e i file variabili in /var/opt. Peccato che in Mandrake /opt sia desolatamente vuota in favore di un po' di casino aggiuntivo nella directory /usr !

/proc (processes – processi) è una directory vuota che viene usata dal SO per montare il filesystem virtuale omonimo (solo in GNU/Linux – non fa parte dell'HFS). I file e le directory contenute in questo filesystem virtuale sono indispensabili ai programmi che hanno la necessità di accedere alle informazioni sul sistema. (fare qualche esempio da terminale...)

/root home del superutente (root) – contiene i dati (documenti) e le impostazioni dell'utente amministratore del sistema. Ci sono molti validi motivi per tenerla separata dalle home degli utenti comuni: impedire più facilmente accessi indesiderati, poterla mantenere nel filesystem principale anche se le home degli utenti sono su un filesystem separato in modo da essere in grado di accedervi anche quando il sistema viene avviato in condizioni di emergenza e non si possono montare altri filesystem

/sbin (system binaries – binari di sistema) utility per l'amministrazione del sistema essenziali per avvio, restore, riparazione del sistema in aggiunta a quelli contenuti in /bin

/tmp (temporary – temporanei) Contiene i file temporanei dei programmi. Potrebbe addirittura essere collocata in un disco virtuale basato su memoria volatile (RAMdisk)

/usr la directory /usr è molto importante: in pratica è la seconda maggior sezione del filesystem. Si compone di una struttura molto articolata. /usr contiene solo dati condivisibili e statici (ogni file di contenuto variabile o che contenga opzioni specifiche per la macchina locale dovrebbe stare in una delle directory che hanno questo scopo - /etc o /var). In pratica /usr replica l'albero principale per contenere tutti i programmi non essenziali per l'amministrazione del sistema (ovvero tutti quelli che usiamo di più se non facciamo il sysadmin di mestiere!!!) Vediamo alcune delle sottodirectory che contiene:

/usr/bin

/usr/etc

/usr/games strano che non siano ritenuti essenziali per il sistema!!!

/usr/include (...)

/usr/lib

/usr/local gerarchia in cui vengono installati i programmi locali (che non possono essere eseguiti dalle macchine in rete con noi). L'albero che si dirama da questa directory non viene sovrascritto in caso di upgrade (aggiornamento) del sistema

/bin

/doc

/etc

/games

...

/usr/sbin

/usr/share contiene i dati che possono essere condivisi tra macchine con piattaforme hardware differenti (i386, alph, PPC...) (sono molti più di quelli che pensiamo possano esistere – l'interoperabilità è un miraggio solo nelle pubblicità dei SO proprietari...). Contiene anche le manpages in /usr/share/man (i comandi sono uguali e hanno le stesse opzioni su tutte le piattaforme. Se vi sedete davanti a un IBM S/370 o ad una macchina Sparc con Linux a bordo, sarete perfettamente in grado di usarla come il vostro portatolino!!!!)

/usr/src contiene il codice sorgente dei programmi da compilare e in /usr/src/linux il codice sorgente del sistema operativo stesso (prima o poi vorrete ricompilarlo anche voi...)

/usr/X11R6 contiene la gerarchia riguardante il sistema grafico di Unix ed i suoi programmi

/var contiene le directory e i file di contenuto variabile (cioè un po' di tutto di quello che non può stare in /usr perchè /usr deve poter funzionare anche in sola lettura. Tra gli altri contiene:

/var/local dati variabili relativi alla struttura /usr/local

/var/log contiene i log dei programmi e di sistema (importante controllarli spesso)

/var/spool contiene le code di stampa e delle e-mail ancora da consegnare agli utenti

/var/www contiene le pagine web se avete installato apache o un altro server web

/var/ftp

All'interno di / (root) FHS v2.2 richiede che siano obbligatoriamente presenti (o soft-linkate) le seguenti directory: /bin /boot /dev /etc /lib /mnt /opt /sbin /tmp /usr /var. Sono richieste solo se i relativi sottosistemi sono installati: /home /lib<qual> (condivise per sistemi diversi) /root.

FHS v2.3 ha apportato alcune variazioni che alcune distribuzioni hanno già implementato (ad es. Suse): /media (punto di mount per filesystem ospitati costantemente – floppy, cdrom, rimovibili), /srv, ...

... e poi c'è sempre montata (ma non si vede) almeno una partizione di **swap** ovvero di memoria virtuale che si aggiunge alla RAM in caso di necessità. Nei sistemi Intel x86 si possono montare fino a 16 partizioni di swap contemporaneamente! In caso di problemi alla partizione che la ospita swap può essere disattivata e riattivata con i comandi #swapon e #swapoff

comandi per gestire i file:

ora cominciamo ad apprendere come usare il sistema per navigare ed agire nel filesystem, con qualche esempio pratico:

APRIRE LA SHELL

(mostrare icona per aprire il terminale)

(nominare xterm, shell, bash, prompt \$ o #...)

\$ls (parlare dei file nascosti [.qualcosa])

\$cp

\$mv

\$rm

\$touch

\$ln

comandi per gestire directory

.

..

~

pwd

cd (parlare di percorsi assoluti e relativi)

tree

mkdir

rmdir

comandi per cercare file e directory

file

locate

updatedb

find

comandi per gestire filesystem

(spiegare #su)

#mount

#umount

#mkfs

#du

#df

#fsck

#fdisk

/etc/fstab

/etc/mstab

è sempre una buona idea consultare man!

Fonti:

- Daniele Giacomini: Appunti di informatica libera
- Rusty Russel, Daniel Quintan: Filesystem Hierarchy Standard v2.2
- Rusty Russel, Daniel Quintan, Christopher Yeoh: Filesystem Hierarchy Standard v2.3
- Domenico Pingitore: Il filesystem di Linux: struttura e contenuto (in Linux Magazine n° 38 Febbraio 2004)

- Palin e Lusc (LugCR): Seminario filesystem e permessi per il LinuxDay 2003
- Jon “Maddog” Hall e Paul G. Sery: RedHat Linux for dummies- IDG Books
- Moshe Bar: Linux file system – McGraw Hill
- Claudio Panichi: Hardening(2): rendere sicuro l'accesso al filesystem (in Linux&c numero 37 Febbraio 2004)
- autori vari: man pages