

Il File System Linux

Samuele Cacchiarelli

30-09-2004

Sommario

Il contenuto di tale documento è una sintesi di quanto esposto durante la prima lezione del CORSO LINUX organizzata dal GLM. Esso pertanto non ha la pretesa di essere esaustivo, costituisce semmai una sorta di linea guida, sotto forma di appunti, fornendo nel contempo gli adeguati riferimenti per un eventuale approfondimento dell'argomento. La bibliografia da cui il documento è tratto è indicata nelle note.

Parte I

Organizzazione del File System Linux

“Per i sistemi Unix, un filesystem è un dispositivo (ad esempio un disco fisso, un floppy, o un CD-ROM) formattato in modo da memorizzare dei file. I filesystem si trovano sui dischi fissi, sui CD-ROM e su ogni altro dispositivo di memorizzazione e consente l'esecuzione di accessi diretti (un nastro dunque che per sua natura, consente solo accessi sequenziali, non può contenere un filesystem)¹”

Cosa contiene dunque un filesystem

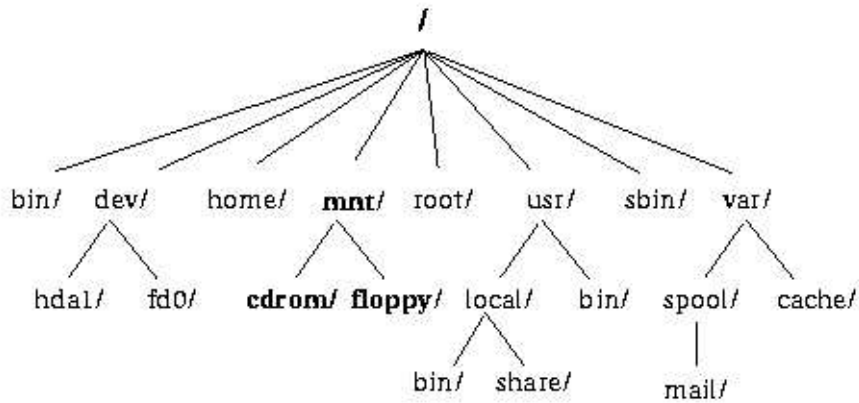
File, Directory, Device ... ovvero *File*.

In un filesystem Linux infatti, possiamo considerare gli oggetti che esso contiene come semplici file.

In realtà il filesystem è molto di più che un semplice contenitore logico di oggetti (file), ma ai fini della comprensione del suo ruolo, tale definizione è più che sufficiente.

¹LINUX LA GUIDA di M. Welsh, M. K. Dalheimer e L. Kaufman, edito dalla O'Reilly APOGEO2000

Figura 1: Albero delle directory



Più in generale il filesystem può essere pensato come comprendente quattro componenti principali²: Nota

- Un *namespace* - un modo di dare un nome agli oggetti e organizzarli in una gerarchia.
- Un *API* - un set di chiamate di sistema per navigare e manipolare gli oggetti.
- Un *security model* - uno schema per proteggere, nascondere e condividere gli oggetti.
- Una implementazione software che leghi il modello logico all'attuale hardware.

Come si presenta un filesystem Linux

Il filesystem Linux si presenta come una singola gerarchia unificata³ che inizia dalla directory / (*radice*) e si dirama in diversi file e sottodirectory seguendo una struttura ad albero.

La sequenza delle directory che devono essere attraversate per raggiungere un determinato file costituisce il *path* o percorso di quel file.

Si definisce *pathname* il nome del file completo del proprio *path* (ad esempio */usr/local/bin/xterm*).

²LINUX ADMINISTRATION HANDBOOK di E. Nemeth, G. Snyder, T. R. Hein, edito da Prentice Hall PTR 2002

³Il sistema a gerarchia singola differisce da quello adottato Windows, che invece conserva un sistema di *namespace* basato sui dischi (*disk-specific namespaces*).

Albero delle directory e filesystem

Sebbene il filesystem sia concepibile come un insieme ordinato di file raggruppati in diverse directory (e sottodirectory), non va confuso con l'*albero delle directory* di un sistema Linux.

Quest'ultimo costituisce l'insieme complessivo dei file e delle directory su cui si sviluppa un intero sistema Linux e può essere (e spesso lo è) formato da più *filesystem*.

A differenza di un sistema Windows una installazione di un sistema Linux può ad esempio venire distribuita su più partizioni di uno stesso disco o addirittura su più dischi. Ciò comporta che l'albero complessivo delle directory sarà formato da più filesystem.

Tornando quindi alla nostra analogia, possiamo pensare al *file system* come ad un blocco o ad un ramo dell'albero delle directory, il cui contenuto è fisicamente situato in uno specifico dispositivo di memorizzazione, che può essere la partizione di un Hard Disk, un floppy, un cdrom o addirittura la stessa memoria RAM.

Come è possibile unire più filesystem in un unico albero delle directory

Il meccanismo che permette ad un sistema Linux di legare più filesystem in un unico albero di directory si chiama *mounting*.

Il *mounting* permette di agganciare un filesystem ad una directory di un altro filesystem. In tal modo il contenuto del filesystem *montato* sarà visibile a partire dalla directory su cui è stato *montato*. Quest'ultima directory viene chiamata *punto di mounting*.

Un tipico esempio di tale operazione, è quando si vuole accedere ad un dispositivo esterno come un cdrom o un floppy. I dati presenti in un cdrom o un floppy sono essi stessi organizzati in un filesystem. Per poterli visualizzare in un sistema Linux occorrerà preliminarmente montare il loro file di dispositivo⁴ su un determinato punto di mount e quindi entrare in tale directory.

Il comando utilizzato per tale operazione è *mount*.

Ad esempio:

```
mount -t auto /dev/hdb /mnt/cdrom
cd /mnt/cdrom
ls
```

Organizzazione dell'albero delle directory

Un sistema operativo Linux presenta una serie di directory più o meno standard che sono riportate di seguito

⁴Per i file di dispositivo, vedi a pag. 5

Pathname	Contenuto
/bin	Comandi base
/boot	Kernel e file necessari per il caricamento dello stesso
/dev	File di dispositivo per terminali, dischi, modem, ecc.
/etc	File di configurazione
/home	Home directory degli utenti
/lib	Librerie e parti del compilatore C
/opt	Programmi opzionali
/root	Home directory del superuser root
/sbin	Comandi per booting, riparazione e recupero del sistema
/tmp	File temporanei
/usr	Gerarchia di file e comandi secondari
/var	dati, file di cache, file di log, file di spool

Tipi di File

In un filesystem Linux, possiamo individuare 7 tipi di file⁵:

1. File Regolari
2. Directory
3. File di dispositivi a caratteri
4. File di dispositivi a blocchi
5. Sockets
6. File FIFO
7. Link simbolici

Il comando per riconoscere di che tipo è un determinato file è *ls*

```
ls -ld nomefile
drwxr-x--- 27 sam user 4096 Sep 30 20:45 nomefile
```

la prima lettera dell'output indica il tipo di file.

Ogni tipo di file è generato da uno o più comandi specifici, ma tutti possono essere rimossi con il comando *rm* .

File regolari

“Sono semplicemente un 'insieme di byte'. Linux non impone alcuna struttura riguardo al suo contenuto. File di testo, file di dati, programmi eseguibili e librerie condivise, sono tutti memorizzati come file regolari. “

⁵Vedi nota 2

Directory

Sono particolari file che contengono i riferimenti verso altri file. Contengono inoltre:

- il riferimento alla directory corrente (indicata con il “.”)
- il riferimento alla directory precedente (indicata con i “..”)
- il nome dei file in esso indicati.

I nomi dei file non sono infatti memorizzati nei file stessi ma nella directory che li contiene. Il nome di un file è stabilito dunque dalla indicazione e dal riferimento ad esso contenuta nella directory.

Tale condizione fa sì che nei sistemi linux sia possibile assegnare anche due nomi diversi ad uno stesso file. Non avremo in tal caso due copie di uno stesso file ma un unico file indicato con due nomi diversi.

Si parla in questo caso di *hard links* e vengono creati con il comando *ln* . Gli *hard links* non costituiscono pertanto un tipo diverso di file, ma solo un modo di dare più nomi ad uno stesso file.

Nota:
hard links

Il comando per creare una directory è *mkdir* .

File dei dispositivi a caratteri e a blocchi

“I file dei dispositivi consentono ai programmi degli utenti di accedere [...] ai dispositivi hardware presenti nel sistema. [...] Quando si accede ad un file di questo tipo, il kernel riconosce la richiesta di I/O e la passa al driver appropriato per il dispositivo.”⁶

I file di dispositivo sono dunque speciali file che fungono da punti di incontro per far comunicare i driver del dispositivo hardware con il dispositivo stesso. In un certo senso, rappresentano per il sistema operativo il dispositivo stesso (un disco fisso, un lettore CD-ROM ecc.). Tali file si trovano nella specifica directory */dev* . Normalmente vengono creati automaticamente dal sistema, oppure è possibile crearli con il comando *mknod* . Nuovi sistemi come *devfs* o *udev*, tuttavia, hanno reso questa operazione non necessaria.

A seconda della modalità di accesso consentita dal dispositivo hardware, si fanno riferimento a due tipi di file di dispositivo: a caratteri e a blocchi.

L'accesso in lettura e scrittura su di un dispositivo a caratteri (stampanti, modem, mouse, tastiera ecc.) avviene byte per byte; l'accesso su di un dispositivo a blocchi (dischi fissi, floppy ecc.) avviene per blocchi di byte.

⁶Rif. LINUX LA GUIDA - pag. 199

Sockets

Sono connessioni/collegamenti tra processi che consentono loro di comunicare in maniera corretta. In linux esistono diversi tipi di sockets, molti dei quali implicano l'uso della rete.

FIFO

Analogamente ai sockets, sono dei particolari file che permettono la comunicazione fra due processi attivi in uno stesso computer.

Vengono creati con il comando *mkfifo* .

Link simbolici

Mentre un *hard link* costituisce un riferimento diretto ad un file, un link simbolico, detto anche *soft link*, costituisce un riferimento diretto ad un *nome* o ad un *percorso* di un file o directory.

Vengono creati con il comando *ln -s* .

Permessi e attributi di un file

In un sistema Linux ogni file porta con se una serie di informazioni riguardanti:

- l'indicazione del tipo di file
- il nome dell'utente proprietario del file
- il nome del gruppo proprietario del file
- i permessi sul file assegnati al proprietario
- i permessi sul file assegnati al gruppo
- i permessi sul file assegnati a tutti gli altri
- altre indicazioni quali, la data di creazione o dell'ultima modifica, la dimensione e ulteriori attributi.

Il comando per visualizzare queste informazioni è *ls -ld*

```
ls -ld fileesempio
drwxr-x--- 27 sam user 4096 Sep 30 20:45 fileesempio
```

Linux nel definire i permessi su un file fa riferimento a tre categorie di utenti:

- l'utente proprietario del file

- gli utenti appartenenti al gruppo proprietario del file
- tutti gli altri utenti.

Per ognuna di queste tre categorie di utenti vengono indicati tre tipi di permessi:

- permesso di lettura (indicato con la lettera *r*)
- permesso di scrittura (indicato con la lettera *w*)
- permesso di esecuzione (indicato con la lettera *x*)

Il tipo di file è indicato dal primo carattere dell'output del comando `ls -ld` e può essere una delle seguenti indicate in tabella

Tipo di file	Simbolo
File regolare	-
Directory	d
File di dispositivo a caratteri	c
File di dispositivo a blocchi	b
Socket	s
FIFO (o named pipe)	p
Link simbolico	l

I successivi nove caratteri dell'output del comando `ls -ld` vanno letti a gruppi di tre e indicano i permessi per le tre categorie di utenti. Nell'esempio precedente (di pagina 6) vediamo che il file *fileesempio* è una directory (d), che l'utente proprietario ha i permessi di lettura (r) di scrittura (w) e di esecuzione (x) indicati da *rw*x, che gli utenti appartenenti al gruppo proprietario hanno il permesso di lettura (r) , non hanno quello di scrittura (-), e hanno il permesso di esecuzione (x) indicati da *r-x*. Gli altri utenti invece non hanno alcun permesso sulla directory (—).

Per modificare i permessi di un file si usa il comando *chmod*.

Ad esempio per concedere i permessi di lettura e scrittura a tutti gli altri utenti (*others*):

```
chmod o+rw fileesempio
```

Per modificare il proprietario o il gruppo proprietario del file si usa invece il comando *chown*.

Ad esempio il comando:

```
chown rik.glm fileesempio
```

cambia il proprietario di *filesempio* in *rik* e il gruppo di appartenenza in *glm*⁷.

Continuando con l'esempio di pagina 6, si può notare che subito dopo i permessi è presente un numero (27). Questo indica il numero di *hard links* presenti nel sistema riferiti alla stessa directory, ovvero indica che vi sono 27 riferimenti nel sistema che puntano alla directory *filesempio*.

Come è possibile? È molto facile per una directory avere molti *hard links* che puntano alla stessa. Facciamo un esempio.

Come detto a pagina 5, una directory contiene al suo interno il riferimento alla directory stessa, identificata dal comando *ls -ld* con un "." . Il punto è dunque un altro nome con cui si fa riferimento alla stessa directory. Da ciò si evince che una directory avrà sempre almeno due *hard links*: 1) il suo nome (contenuto nella directory parente) e 2) il "." (contenuto nella directory stessa).

Abbiamo anche detto che una directory contiene anche il riferimento alla directory precedente, identificata dal comando *ls -ld* con un ".." . Questo significa che qualora una directory contenga una o più directory al suo interno, vi saranno ulteriori *hard links* che punteranno alla stessa.

Le ulteriori informazioni date dal comando *ls -ld* sono:

- il nome dell'utente proprietario (sam)
- il nome del gruppo proprietario (users)
- la dimensione del file di directory (4096)
- la data e l'ora di creazione o di ultima modifica del file stesso (Sep 30 20:45)
- il nome del file (filesempio)

Parte II

Gestione del File System Linux

Unità dischi e partizioni sotto Linux

In Linux esiste un modo particolare per chiamare i vari dispositivi quali floppy dischi o partizioni degli stessi. Mentre in Windows questi vengono identificati da lettere (*A:* o *B:* per il primo e secondo floppy, *C:* per il primo disco o meglio la prima partizione dello stesso, *D:* per la seconda o se

⁷Solo l'utente che è proprietario del file e appartenente al gruppo del file può cambiare il gruppo di lavoro dello stesso. L'utente root non ha limitazioni.

Nota:
*numero di
hard links*

mancante il primo lettore cd-rom/dvd o masterizzatore ecc...) secondo una logica alquanto “bizarra”, in Linux i dispositivi vengono identificati dai *file di dispositivo*. Questi identificano in maniera più chiara e precisa tutti i dispositivi presenti nel sistema.

Vediamo quali sono i più comuni:

Dispositivo	file di dispositivo
Primo floppy (A:)	/dev/fd0
Secondo floppy (B:)	/dev/fd1
Primo disco fisso del primo canale IDE (intero disco)	/dev/hd a
Secondo disco fisso del secondo canale IDE (intero disco)	/dev/hd b
Primo disco fisso del secondo canale IDE (intero disco)	/dev/hd c
Secondo disco fisso del secondo canale IDE (intero disco)	/dev/hd d
Prima partizione <i>primaria</i> del primo disco	/dev/hda1
Seconda partizione <i>primaria</i> del primo disco	/dev/hda2
Terza partizione <i>primaria</i> del primo disco	/dev/hda3
Quarta partizione <i>primaria</i> del primo disco	/dev/hda4
Prima partizione <i>LOGICA</i> del primo disco	/dev/hda5
Seconda partizione <i>LOGICA</i> del primo disco	/dev/hda6
Primo disco SCSI (intero disco)	/dev/ <i>sda</i>
Secondo disco SCSI (intero disco)	/dev/ <i>sdb</i>
Prima partizione <i>primaria</i> del primo disco SCSI	/dev/sda1
Prima partizione <i>LOGICA</i> del primo disco SCSI	/dev/sda5
Seconda partizione <i>LOGICA</i> del secondo disco SCSI	/dev/sdb6

I filesystem Linux

Al momento Linux supporta una enorme varietà di filesystem . Ogni tipo di file system ha propri attributi e limitazioni; un noto esempio è il filesystem MS-DOS, che può utilizzare nomi di file lunghi al massimo 8 caratteri con l'estensione di tre. Nei sistemi Linux come per gli altri sistemi operativi tale filesystem oramai è usato solo per la memorizzazione o la lettura di dati su Floppy Disk o su vecchi dischi.

I filesystem maggiormente utilizzati sono invece i seguenti:

ext2 filesystem nativo

ext3 un evoluzione dell'ext2, con supporto al journaling

reiserfs anch'esso con supporto al journaling

iso9660 il filesystem presente sui supporti cdrom

Mounting dei filesystem

Come anticipato, per accedere ai dati di un determinato filesystem, è necessario che questo sia *montato* su qualche directory. Alcuni di questi vengono montati automaticamente in fase di *boot* del sistema. Altri possono essere montati (e smontati) successivamente.

Il comando per fare questa operazione è *mount*.

```
mount -t auto /dev/hdb /mnt/cdrom
```

Il comando indicato ad esempio monta il dispositivo CD-ROM rappresentato da */dev/hdb* sulla directory */mnt/cdrom*. Una volta montata sarà possibile entrare nella directory */mnt/cdrom* e visualizzare il contenuto del cdrom.

Per smontare un filesystem si usa il comando *umount*.

```
umount /mnt/cdrom
```

Affinchè il comando vada a buon fine è necessario che il dispositivo non risulti occupato (in altre parole non deve esserci un programma che stia leggendo o scrivendo dati all'interno del filesystem da smontare).

Quando si monta un dispositivo come un CD-ROM, un Floppy, un unità ZIP si prende in considerazione il file di dispositivo rappresentante l'intero disco (*/dev/hdb*) in quanto lo stesso non contiene partizioni.

Al contrario quando si monta un disco fisso o una chiave usb non si fa riferimento all'intero disco, ma ad una specifica partizione in esso contenuta⁸. Ad esempio:

```
mount -t ext3 /dev/hda7 /mnt/DATI
```

Questo comando monta la terza partizione LOGICA del primo disco, sul punto di mount */mnt/DATI*. Con l'opzione *-t ext3* si specifica espressamente che la partizione che si accinge a montare contiene un file system formattato in ext3. In alternativa poteva usare l'opzione *-t auto*, per far sì che il sistema riconoscesse in automatico il tipo di filesystem da montare.

Mounting automatico dei dispositivi

Un sistema operativo Linux può essere distribuito su più filesystem. Ad esempio potremmo avere una situazione in cui la directory degli utenti */home*, la directory */tmp* e la directory */var* siano collocate su partizioni diverse da quella in cui risiede il sistema principale. In tale circostanza avremo un albero di directory distribuito su ben quattro filesystem:

⁸I dischi fissi infatti richiedono di essere partizionati, prima ancora di venire formattati, per poter essere utilizzati come dispositivi di memorizzazione.

- uno per la partizione di root (quella che contiene la directory principale “/” detta anche radice)
- uno per la directory /home
- uno per la directory /tmp
- uno per la directory /var

In una situazione del genere è necessario che le quattro partizioni e di conseguenza i quattro filesystem vengano montati automaticamente dal sistema in fase di boot. Oltre a queste inoltre il sistema Linux monta normalmente in fase di boot anche una particolare partizione definita di *swap* (quest’ultima è una partizione dedicata che Linux usa per gestire la cosiddetta memoria Virtuale). Può montare, inoltre, altri “particolari filesystem” come ad esempio il filesystem *proc* o il filesystem *sys*, che non risiedono affatto in una partizione di un disco ma bensì rimangono localizzati in una zona della memoria fisica (RAM) del computer.

L’automatismo con cui avviene il montaggio di tali filesystem viene stabilito da un apposito file di configurazione: */etc/fstab* , il cui contenuto può riflettere una struttura come la seguente:

<i>#dispositivo</i>	<i>mountpoint</i>	<i>tipo filesystem</i>	<i>opzioni</i>	<i>opzioni</i>
/dev/hda1	/	ext3	defaults	1 1
/dev/hda2	/home	reiserfs	defaults	1 2
/dev/hda5	/tmp	ext2	defaults	0 2
/dev/hda6	/var	ext3	defaults	1 1
/dev/hda3	none	swap	defaults	0 0
none	/proc	proc	defaults	0 0

nello stesso file vengono inoltre solitamente indicati i punti di mount di default per i dispositivi rimovibili (cdrom e floppy) e di eventuali altre partizioni:

/dev/hdb	/mnt/cdrom	iso9660	noauto,rw,user	0 0
/dev/fd0	/mnt/floppy	auto	noauto,user	0 0
/dev/hda7	/mnt/DATI	vfat	noauto,user	0 2

con questa ultima configurazione si permette agli utenti (purchè dispongano comunque dei diritti di accesso ai dispositivi) di montare e smontare all’occorrenza il cdrom ed il floppy. In questo caso sarà per loro sufficiente dare i comandi:

```
mount /mnt/cdrom
mount /mnt/floppy
```

Salvo esplicita indicazione delle opzioni *user* o *users* nel file `/etc/fstab`, ogni operazione di mount è riservata all'utente root, anche qualora gli utenti dispongano dei diritti di accesso al dispositivo da montare o smontare.