

## Organizzazione generale del filesystem con particolare riferimento a Linux

### Premessa: il filesystem e i nomi dei file

Un *filesystem* e' si puo' definire un modo di organizzare dei dati all'interno di un supporto in grado di contenerli: l'elemento di base e' il *file*, che si puo' definire un insieme di dati riuniti sotto uno specifico nome: un file puo' essere un documento, un programma, un'immagine o qualsiasi altra cosa. Per identificarlo e' convenzionale utilizzare l'accoppiata '*nome*' piu' '*estensione*' separate da un punto: il nome dovrebbe dare un'idea del contenuto del file, mentre l'estensione ci dovrebbe dire il 'tipo' dei dati che compongono il file. Cio' non e' piu' necessario ma rimane buona regola continuare a rispettare questa convenzione per evitare confusioni: 'dimissioni.txt' ci fa gia' capire il contenuto e il tipo di dati del file, mentre 'x32.g' no.

Alcune estensioni possono indicare file creati secondo uno standard o usando determinati programmi: possiamo avere '.txt', '.doc', '.rtf', '.wri' per i documenti di testo, '.jpg', '.gif', '.bmp' per le immagini, '.php', '.c', '.cpp', '.vba' che rappresentano il "codici" di programmazione, '.au' o '.mp3' saranno file audio e cosi' via.

Un gruppo a parte e' rappresentato dai file "eseguibili", cioe' da quei file che rappresentano un programma che puo' essere eseguito dalla macchina: in questo caso, per ambienti "dos-like" (come windows) abbiamo le estensioni '.com', '.exe', '.bat' mentre per gli ambienti "unix-like" queste estensioni non sono necessarie e servono solo come indicazione, ad esempio '.sh' (per shell) o '.bin' (binary). In effetti, l'*eseguibilita'* di un file negli ambienti Unix e' regolata dai cosiddetti *permessi* i quali determinano anche l'*usabilita'* (in senso di permesso di *lettura* e *scrittura*) del file da parte dei *proprietari* (che possono essere individui singoli (*user*) o gruppi (*group*)) Per maggiori informazioni, si cerchi in rete "permessi file linux" oppure si leggano le pagine del manuale: 'man chmod' e 'man chown'.

Per identificare univocamente un file occorre specificare, oltre a nome ed estensione, anche il "percorso" (in inglese "path") completo che serve per "arrivare" ad esso. La struttura piu' comune di un filesystem rassomiglia ad un albero, che parte da un tronco (directory o "cartella") principale per diramarsi in piu' rami (sottodirectory o sottocartelle) in modo da organizzare i file in maniera ordinata.

Ogni ramo puo' avere un solo file con quel nome + estensione ma nulla vieta che rami diversi possano avere degli omonimi (vedere dopo). La directory principale del filesystem unix-like e' detta '*root*' (da radice) ed e' identificata con la barra (*slash*: '/'), la stessa che viene usata in internet) mentre nei sistemi dos-like si usa la barra rovesciata (*backslash*: '\'). Per convenzione, nel DOS il supporto fisico che contiene un filesystem viene indicato da una lettera seguita dal duepunti (':'): partendo dalle prime lettere e seguendo ragioni storiche. Cosi' 'a:' sara' il primo floppy disk, 'b:' il secondo floppy (che attualmente non c'e' quasi mai); 'c:' il primo disco fisso e le lettere successive, che possono indicare altri supporti come hard disk secondari, CDROM, memoria (penna) USB e cosi' via. Questi nomi vengono riorganizzati, rinominando anche le periferiche gia' presenti, ad ogni aggiunta di nuove periferiche: questo genera diversi problemi per la corretta identificazione delle periferiche stesse, che possono cambiare "nome" inserendo o togliendo nuovi supporti (ad esempio, comandi come "e:\programmi\flipper.exe" possono non funzionare se abbiamo appena connesso un nuovo hard disk esterno che ha preso, automaticamente, proprio il nome 'e:'). Negli ambienti come linux, invece, un supporto verra' "montato" come directory (detta "*mount point*") all'interno del filesystem principale ('root') in maniera non sempre automatica ma sicuramente piu' flessibile e sicura, anche se richiede una certa attenzione.

### Organizzazione del filesystem

Una buona organizzazione dei dati e' indispensabile per un lavoro efficiente e conviene perdere un po' di tempo per organizzare il proprio filesystem in maniera corretta piuttosto che ammassare file di qualunque tipo alla rinfusa dentro una directory qualunque. Facciamo solo alcuni esempi:

1) *Due file "omonimi" che identificano persone diverse (a meno di non essere molto fortunati):*

```
c:\mieidocumenti\foto\fidanzata\monica1.jpg
a:\artistifamosi\bellucci\monica1.jpg
```

la stessa cosa, piu' neutra, per ambienti Unix (notare come il "mount-point" del secondo esempio e' probabilmente una penna USB):

```
/docs/personali/fotogite/roma2005/fontanaditrevi.jpg
/mnt/pennausb/scanner/cartoline/fontanaditrevi.jpg
```

2) *Un esempio di filesystem ben organizzato (Unix):*

```
/documenti/foto/personali/gite/roma2005/riassunto.txt
/documenti/lavoro/redditi/2002/riassunto.txt
```

/documenti/racconti/guerraepace/riassunto.txt

3) Esempio caotico (ambiente DOS) degli stessi documenti del punto (2):

```
c:\riassuom.txt
c:\riassdich1.rdt
c:\tmp3\riassgp
```

4) Altra differenza tra DOS e Unix:

```
c:\tmp\prova.rtf
c:\tmp\Prova.RTF
```

(siamo in ambienti DOS) sono lo stesso file. Sotto Unix invece:

```
/tmp/prova.rtf
/tmp/Prova.RTF
```

i due file possono benissimo coesistere nella stessa directory perche' Unix e' "case-sensitive", cioe' "A" e' diverso da "a". Per chi e' nuovo di Linux puo' spiazzare, basta prenderci un attimo la mano.

Per maggiori informazioni sul filesystem di Linux e la gestione delle directory, dei file e dei "mount point" consultare le pagine man di: mount, cd, mkdir, rmdir, umount, fstab, cp, mv eccetera.

### Breve panoramica delle directory Linux

L'albero delle directory (il 'filesystem') parte dalla directory radice (root, '/') e si estende attraverso tutto cio' che e' montato e raggiungibile dal computer. Per convenzione vi sono alcune directory "fisse", quasi sempre le stesse per ogni distribuzione Linux (ma non e' del tutto vero). Sapere cosa contengono le directory principali puo' essere utile:

#### **/boot**

Contiene l'immagine del kernel (di solito 'vmlinuz...' ma puo' anche esser chiamato 'bzImage', 'zImage' o anche in altri modi) ed altri file che servono per effettuare il "boot" (cioe' l'avvio, detto "bootstrap") della macchina. Possono esser presenti piu' kernel (differenti configurazioni dello stesso kernel oppure versioni diverse): un programma apposito ("bootloader", ad esempio LILO e GRUB) si occupera' all'accensione del computer di caricare il kernel adatto. Conviene non toccarla se non si sa che cosa fare...

#### **/bin (binaries)**

Contiene i programmi di uso comune: i comandi generici usabili da tutti gli utenti risiedono quasi tutti qui. Nota: per sapere dove risiede un comando si puo' usare il comando 'which'. Digitando 'which mkdir' ad esempio restituira':  
/bin/mkdir

#### **/sbin (system binaries)**

Anche questa contiene file "binari" (cosa che di solito identifica gli eseguibili, cioe' i programmi); si tratta di quelli riservati al "superuser" (cioe' all'amministratore di sistema, di solito detto "root") e non utilizzabili da tutti gli utenti. Di solito sono molti di piu' di quelli contenuti nella directory /bin e possono fare molti danni...

#### **/dev (devices)**

Contiene i link a tutti i dispositivi di sistema. Ogni file qui presente identifica una periferica fisica (floppy, har disk, dispositivo usb, porta seriale, porta audio...) e deve essere presente affinche' il sistema possa accedervi. Ovviamente si tratta di un filesystem virtuale (il file NON E' la periferica, e' soltanto un modo per accedervi) ed e' gestito in maniera automatica dal sistema operativo (attraverso il demone devfsd nei kernel piu' recenti). Agendo sui permessi dei file a questo livello, si puo' ad esempio regolare l'accesso alle periferiche ai singoli utenti: ad esempio posso riservare l'accesso della scheda audio (cioe' la possibilita' di sentire i suoni) al singolo superuser oppure permettere ai soli utenti autorizzati la scrittura su un determinato disco. Trattandosi di file, sia pure virtuali, posso effettuare operazioni di scrittura da e per la periferica in maniera molto semplice. Ad esempio il comando:

```
cat /dev/fd0 > /dev/sda1
```

esegue una copia completa del floppy su di un dispositivo inserito nella prima porta USB (la solita penna dell'esempio precedente). Attenzione: non si tratta di una copia di dati, ma di tutto il disco. Allo stesso modo:

```
cat /myaudio/versobuffo.au > /dev/audio
```

mi invia il file audio versobuffo sulla scheda audio (in pratica dovrei sentire il verso, se ho i permessi per farlo).

### **/etc (et cetera)**

E' la directory che contiene i vari file di configurazione del sistema, validi cioe' per tutti gli utenti, che regolano il comportamento di quasi tutti i programmi presenti sul computer. Si tratta dunque di un directory da studiare con attenzione ed e' conveniente prenderci dimestichezza prima di modificare le cose.

### **/etc/rcN.d (N e' un numero, di solito da 0 a 7)**

Sono piu' directory particolari nelle quali son contenuti dei file (di solito si tratta di file eseguibili o di link ad essi, mentre i file "veri" si trovano in '/etc/init.d') che vengono avviati all'accensione della macchina. Ogni directory rappresenta una diversa configurazione del sistema per la quale alcuni servizi devono essere avviati e altri no (ad esempio, /etc/rc5.d avvia il sistema grafico, /etc/rc1.d la modalita' singolo utente per la manutenzione del sistema, /etc/rc0.d e' la sessione di spegnimento del computer, /etc/rc2.d la modalita' "normale" non grafica e cosi' via). Per maggiori informazioni, si veda 'man init' e 'man runlevel'.

### **/etc/X11**

Contiene invece tutti i file di configurazione di X (l'ambiente grafico): risoluzione del monitor, tipo di tastiera, caratteri visualizzabili in X ed altro sono regolabili agendo nei file di questa directory (esistono comunque molte utility anche grafiche che facilitano il compito)

### **/home**

La casa degli utenti: qui ogni utente ha la sua directory (ad esempio /home/gianfranco) all'interno della quale trovera' i file di configurazione personalizzati, eventuali dati e quant'altro gli piacera' di metterci. Notare che un utente puo' vedere i nomi degli altri utenti, ma non potra' mai entrare nelle loro cartelle perche' non ne ha i permessi: solo root (il superuser) ha il potere assoluto sul sistema. Se si vogliono condividere documenti con altre persone, quindi, e' conveniente creare una directory apposita (ad esempio '/documenti'), renderla accessibile a tutti e piazzare li' i dati "comuni". Essendo una directory particolare, il comando 'cd' usato da solo porta sempre nella directory home. Per una spiegazione dei permessi si veda 'man chmod' e 'man chown'.

### **/mnt (mount)**

Directory di servizio: di solito i punti di mount (cioe' i collegamenti ai dispositivi connessi al sistema) risiedono qui, ma non e' per niente obbligatorio. Quando ad esempio un utente monta un floppy, e' buona norma creare una directory '/mnt/fd' o similare e montarlo qui. Ma niente vieta di montare un floppy su '/ghianda/w3eL9iT/bluzorro', anche se la cosa e' un tantinello confusionaria.

### **/root**

La casa del superuser, il dio supremo del sistema. La struttura e' del tutto simile a quella degli utenti, ma appartiene all'amministratore.

### **/lib (libraries)**

Le librerie e i moduli di sistema, condivisi da tutti. Si tratta di raccolte di funzioni che e' bene non toccare se non si e' abbastanza preparati

### **/opt**

Di solito contiene programmi particolari: piu' spesso vi si trovano gli "ambienti" grafici come gnome e KDE e i loro applicativi.

### **/proc (processes)**

Filesystem virtuale dei dati della tabella dei processi. Ogni directory qui contenuta rappresenta un processo (ad esempio /proc/1/ e' il processo Init) e contiene dei file virtuali che ne rappresentano lo stato. A questa directory accedono i programmi che gestiscono i processi. Per l'utente normale, la directory puo' essere utile perche' contiene le informazioni di tutto il computer: ad esempio '/proc/cpuinfo' fornisce tutte le informazioni sul processore, '/proc/iomem/' su tutte le

porte di IO (Input/Output) e cosi' via.

### **/usr (user)**

Contiene tutto cio' che riguarda gli utenti della macchina: file di documentazione (/usr/doc), librerie (/usr/lib), programmi specifici (/usr/bin e /usr/sbin), i "sorgenti" del kernel e di alcuni programmi (/usr/src) e cosi' via. In particolare, cio' che e' condiviso da tutti sta in (/usr/share) mentre applicativi e dati dedicati agli utenti possono risiedere anche in /usr/local e relative sottodirectory. NOTA: il fatto che programmi e dati possano trovarsi in piu' directory differenti (/sbin, /bin, /usr/bin, /usr/local/bin per i programmi) e' un vantaggio perche' permette di smistare i programmi a seconda dell'uso che ne verra' fatto, e di mantenerne alcuni quando si aggiornera' il sistema. Si pensi alla directory /bin: se tutti gli eseguibili fossero li', ad un aggiornamento della distribuzione si dovrebbero ritrovare i programmi inseriti dagli utenti, toglierli di li', far aggiornare la directory e poi piazzarci di nuovo. Se invece l'utente accorto piazza i suoi programmini (realizzati in proprio o presi da altri non importa) in /usr/local/bin, un eventuale aggiornamento del sistema risulta molto piu' semplice. L'unica accortezza e' di far si' che il sistema "sappia" dove trovare i vari programmi, e questo si fa aggiornando la variabile d'ambiente 'PATH'.

Dentro la directory /usr, a seconda delle distribuzioni possiamo avere una directory chiamata /usr/share/doc o /usr/doc all'interno della quale son contenuti tutti i documenti relativi ai pacchetti installati, spesso con le istruzioni d'uso e di installazione. Dare un'occhiata in queste directory e' sempre una cosa intelligente.

### **/var (variables)**

Ci stanno tutte le "code": ad esempio i vari servizi (demoni) piazzano in questa directory cio' che deve essere processato. La posta e le code di stampa si trovano solitamente in questa directory, mentre

### **/var/log**

contiene i file di LOG, cioe' dei messaggi che i vari programmi inviano per comunicare regolarita' o irregolarita' nel funzionamento. A titolo di prova, si dia da console il comando

```
tail -f /var/log/messages
```

e dopo si inserisca una penna USB nella relativa chiavetta. Probabilmente si riuscirà a vedere i messaggi del kernel che riconosce la chiavetta (il programma tail in questione si interrompe premendo Ctrl-C (Control e contemporaneamente il tasto "C")). Questa directory e' molto utile per ottenere informazioni importanti quando si cerca di risolvere qualche problema.

### **/tmp**

Ovviamente per i file temporanei. Attenzione a non lasciarci niente di importante dentro: in molte distribuzioni viene periodicamente cancellata.

### **/lost+found**

File cancellati o persi. Una sorta di cestino, insomma.

## **Conclusioni**

Uno dei maggiori problemi che ho trovato quando cercavo di imparare Linux e' stato quello di capire come fosse organizzato il sistema e dove andare a cercare quello che mi serviva. Non ho mai trovato un *howto* che spiegasse, perlomeno non in poche parole, come erano organizzate tutte le directory che vedevo e cosa potevo trovarci dentro. Se trovate imprecisioni o avete qualcosa da dire, potete scrivermi una email a [gbertozzi@tiscali.it](mailto:gbertozzi@tiscali.it). Spero di aver fatto cosa utile per qualcuno e se non fosse, pazienza. Almeno e' servito a me. :-)

## **Licenza**

GPL.

--

*Gianfranco Bertozzi*