

Sistemi Operativi 2

Davide Pucci
Marco Panunzio

2016

Indice

1	Introduzione alla Shell	2
1.1	Cenni	2
1.2	Comandi essenziali	2
1.3	Utenti	3
1.4	FileSystem	4
1.5	File	6
1.6	Permessi	6

Capitolo 1

Introduzione alla Shell

1.1 Cenni

La shell bash contiene una *history*. Attraverso di essa è possibile visitare tutti i comandi digitati. Con i tasti cursore è possibile invece navigare nei comandi. Digitando *CTRL+R* è possibile eseguire una sorta di query tra i comandi listati nella history. La Shell essenzialmente attende che l'utente digiti un comando (per questo viene spesso chiamata *prompt*). Il simbolo *~* indica la *home directory*.

1.2 Comandi essenziali

- **man [sezione] comando:** dà informazioni complete su un comando, includendone la sintassi, la descrizione, gli esempi e una larga documentazione.
- **cd path:** varia la *current working directory* attualmente in uso nella shell. Nel caso in cui path è *..*, la *pwd* varia nella directory precedente alla attuale. Il path *.* indica la directory attuale.
- **pwd:** indica la directory sulla quale quali si è posizionati all'interno della shell (*current working directory*).
- **ls [opzioni] [path]:** mostra il contenuto di una certa directory *path* (se non specificata, sottintende che sia la *pwd*). Tra le opzioni più importanti:
 - *-a*: visualizza i file nascosti (in Unix convenzionalmente i file nascosti iniziano con *.*).
 - *-R*: applica il comando ricorsivamente.
- **mkdir [opzioni] [path]:** crea una directory *path*. Tra le opzioni più importanti:

- *-p*: crea tutte le super-directory necessarie alla creazione della directory che si intende creare.
- **touch file**: crea il file *file*.
- **tree [opzioni] [path]**: mostra l'albero delle directory contenute in *path* (se non specificata, sottintende che sia la *pwd*).
- **mount [opzioni] [partizione puntodimount]**: se non viene passato alcun parametro, stampa la mappa di mount attualmente in uso. Se passati *partizione* e *puntodimount*, monta la partizione *partizione* sulla path *puntodimount*, rilevando automaticamente il filesystem utilizzato. Tra le opzioni più importanti:
 - *-t tipofilesystem*: specifica il filesystem (*ext4*, *ext3*, *ntfs*, *hfs*, *nfs*, e così via ...) da utilizzare nel mount della *partizione* specificata.
- **cat [file] [>/>>outfile]**: stampa a video il contenuto del *file*. Se non viene passato in input il *file*, aspetta l'arrivo di uno *stdin* (*standard input*) e lo stampa, fino a che non viene segnalato l'*EOF* (*end-of-file*). Può far uso di opzioni particolari:
 - *>*: reindirizza lo *stdout* (*standard output*) rimpiazzandolo al contenuto del file *outfile*.
 - *>>*: reindirizza lo *stdout* aggiungendolo in coda al file *outfile*.
- **stat file**: stampa le diverse informazioni dell'*inode* relativo al file *file*.
- **chmod [opzioni] permessi file**: modifica i permessi sul file *file*, applicando quelli specificati in *permessi* (attraverso combinazioni alfabetiche o ottali). Tra le opzioni più importanti:
 - *-R*: applica il cambiamento dei permessi ricorsivamente (nel caso in cui *file* sia una directory).

1.3 Utenti

Dopo l'installazione di un OS è sempre necessario configurare un utente. Ogni utente appartiene sempre almeno ad un gruppo. Per ottenere i gruppi di cui fa parte l'utente che utilizza la shell, si utilizza il comando:

```
groups
```

Invece, per ottenere i gruppi di cui fa parte l'utente generico *utente*:

```
groups utente
```

Un gruppo importante è *sudo*. *sudo* è un comando che accetta comandi da elevare a privilegiati. Per cambiare utente da shell occorre eseguire il comando:

```
su -l utente
```

Per aggiungere un utente ad un gruppo, occorre utilizzare il comando:

```
useradd utente gruppo
```

Un file importante in questo ambito è */etc/passwd*, che contiene diverse informazioni sull'utente separate da : o , . Le informazioni sono:

- Username
- Password (cifrata e gestita in un altro file - */etc/shadow*)
- *UID* (*User ID*)
- *GID* (*Group ID*)
- Path della home directory
- Nome della shell associata all'utente

Analogamente, il file */etc/groups* contiene le informazioni relative ai gruppi:

- Nome gruppo
- *GID*
- *UID* - separati da , - degli utenti membri del gruppo

1.4 FileSystem

Tutti i file e le directory sono contenuti direttamente o indirettamente nella directory di root, con la struttura di un albero. Le foglie di questo albero sono:

- Directory vuote
- File

All'interno di una directory non ci possono essere elementi con lo stesso nome (la differenza di *case* rende i nomi diversi). Il *path assoluto*, quindi una cosa fatta così indica il percorso totale per raggiungere il dato elemento all'interno dell'albero del filesystem (sostanzialmente una sequenza di directory separate da /). Quello *relativo* invece è la serie delle sole directory - separate da / - necessarie per arrivare all'elemento desiderato a partire dalla *current working directory*. Il Filesystem di Linux è unico, contenuto interamente in / (Windows si divide in volumi). Ciononostante, può contenere elementi eterogenei:

- Dischi fisici
- Filesystem virtuali
- Filesystem di rete

- Filesystem in memoria principale (*RAM*)

Questo è possibile tramite il comando *mount*. Per esempio, */proc* contiene informazioni su tutti i processi attualmente in esecuzione, attraverso i loro *PID*. Questa path è virtuale - non fisica - e viene montata automaticamente dal kernel all'avvio dell'OS. Ciascuna di queste directory contiene un file *status* che contiene informazioni sensibili:

- Nome del processo
- *PID* del processo
- *PID* del processo padre
- Così via ...

Solitamente il disco principale, sul quale viene installato l'OS è montato in */*. Nei casi in cui ci sia un solo disco, è possibile partizionarlo in più parti, montandole ciascuna in un punto di mount differente. Partizionare un disco può molto spesso voler dire eliminare definitivamente i dati precedentemente allocati. Esistono diversi programmi per gestire il partizionamento dei dischi (come *parted*, *gparted*, *fdisk*, e così via ...). Tra i file importanti in questo ambito:

- */etc/mtab*: equivalente dell'output del comando *mount* (mostra i punti di mount attualmente in uso)
- */etc/fstab*: specifica le partizioni (o i dischi) da montare all'avvio dell'OS, con annesso il relativo punto di mount e filesystem.

Lo schema generico del filesystem è il seguente:

Dimensione	Spiegazione	Montata
<i>/boot</i>	Kernel e file di boot	NO
<i>/bin</i>	Binari (programmi eseguibili) di base	NO
<i>/dev</i>	Devices (periferiche) hardware e virtuali	boot
<i>/etc</i>	File di configurazione di sistema	NO
<i>/proc</i>	Dati e statistiche dei processi e parametri del kernel	boot
<i>/sys</i>	Informazioni e statistiche di device di sistema	boot
<i>/media</i>	Mountpoint per device di I/O (es: CD, DVD, USB pen)	quando necessario
<i>/mnt</i>	(come <i>/media</i>)	quando necessario
<i>/sbin</i>	Binari di sistema	NO
<i>/var</i>	File variabili (log le, code di stampa, mail ...)	NO
<i>/tmp</i>	File temporanei	NO
<i>/lib</i>	Librerie	NO

1.5 File

Ad ogni file del filesystem è associato un numero identificativo che ne indica l'*inode*, una struttura dati contenente informazioni specifiche al file stesso:

Attributo	Spiegazione
Type	Tipo del file
User ID	ID del proprietario
Group ID	ID del gruppo associato
Mode	Permessi di accesso per il proprietario, il gruppo e tutti gli altri
Size	Dimensione in byte del file
Timestamps	<ul style="list-style-type: none">• ctime (inode changing time: cambiamento di un attributo)• mtime (content modification time: solo scrittura)• atime (content access time: anche lettura)
Link count	numero di hard links
Data pointers	Puntatore alla lista di blocchi che compongono il file

Ciascuna di queste informazioni possono essere visualizzate tramite apposite opzioni del comando *ls*.

1.6 Permessi

Ad ogni *inode* è associato un sistema che ne gestisce i permessi. I permessi sono di *lettura*, *scrittura* ed *esecuzione*. Nell'applicazione dell'impostazione dei permessi si può specificare il permesso tramite la lettera alfabetica (*r*, *w* o *x*) o il valore ottale (*r* vale 4, *w* vale 2 e *x* vale 1: le varie combinazioni della somma dei vari valori compone i permessi associati al file) associati ad esso. Per modificare i permessi di un file si usa il comando *chmod*.

Permesso	Ottale	Significato
—	0	Non si può fare niente
-x	1	Si può settare come cwd; si può anche "attraversare", se già se ne conosce il contenuto (ad esempio, si può leggere un file o una directory al suo interno, se i permessi di questi ultimi contengono la lettura)
-w-	2	Non si può fare niente (per fare veramente modi che, occorrono i permessi di esecuzione)
-wx	3	Come il permesso 7, ma non si può listare il contenuto (con o senza attributi)
r-	4	Si può solo listarne il contenuto, ma senza vedere gli attributi delle/directories contenute (l'unica cosa che si può sapere è se si tratta di file o di directory); non può essere "attraversata"
r-x	5	Si può leggere (attributi compresi), settare come cwd ed attraversare; non è possibile cancellare o aggiungere file/directory
rw-	6	Come il permesso 4 (write senza execute è inutile)
rwX	7	Si può fare tutto: listare contenuto (attributi compresi), aggiungere le directory, cancellare le contenute in essa (anche senza avere il permesso di scrittura sul file! correggibile con lo sticky bit, vedere più avanti), cancellare directory contenute in essa (ma occorrono tutti i permessi su tali directory)