

# *Guida all'uso dei comandi nella shell - comandi Unix piu' diffusi*

Autore: Stefano Pardini

<http://www.viareggiolinux.org>

Linux User Group: ACROS <http://www.lug-acros.org>

## **Premessa**

La shell è sicuramente una delle parti più importanti di un sistema Unix e si può definire un programma che implementa l'interfaccia a riga di comando e che esegue i comandi digitati dall'utente. Attraverso la shell è possibile concatenare una serie di comandi riuscendo a realizzare compiti anche complessi con estrema velocità e flessibilità. La shell è uno strumento molto potente e l'unico modo per imparare ad usarla è utilizzarla. Vi renderete conto che non potete fare a meno di imparare a conoscere ed usare la shell per gestire al meglio il vostro sistema GNU/Linux. Esistono molte shell Unix diverse, ma molte delle loro funzioni dipendono dalla shell Bourne, o /bin/sh. GNU/Linux utilizza una versione migliorata della shell Bourne, chiamata bash. Bash è la shell predefinita per la maggior parte dei sistemi GNU/Linux. Durante l'installazione di un sistema GNU/Linux solitamente si imposta una password di root (superuser) e si crea un utente comune per poter lavorare senza dover accedere al sistema come root. A volte lavorando con la shell sarete obbligati a diventare root per poter eseguire certi comandi, ad esempio se avrete la necessita' di modificare un file di configurazione del vostro sistema GNU/Linux. Diamoci da fare ed iniziamo subito a richiamare una finestra della shell. La finestra della shell è una finestra del terminale che a sua volta può essere definita una interfaccia per l'utente, cioè un programma che accetta i comandi digitati e disegna i caratteri sullo schermo. In pratica il terminale è un intermediario fra la shell e l'utente. Apriamo quindi un terminale quale konsole e cominciamo a vedere quali sono i comandi base più diffusi:

## **cat**

il comando cat è uno dei piu' semplici, questo comando stampa il contenuto di un file ed esce, cat è ottimo per iniziare a lavorare, provate a digitare:

cat nomefile e vedete cosa succede...

cat vi restituirà il contenuto del file, se provate a digitare solo cat senza alcun file di input e battete invio, noterete che non tornerete al prompt della shell perché cat è ancora in esecuzione. Se digitate ora qualsiasi cosa e premete invio, cat ripeterà la riga precedente finché non uscirete premendo control-D;

## **cat /proc/cpuinfo**

è un'altro esempio di cosa si può fare con cat, questa riga di comando vi mostra la CPU in uso sulla vostra macchina con informazioni dettagliate;

## **ls**

il comando ls elenca il contenuto di una directory e può contenere diverse opzioni, se digitate ls -l ottenete un elenco dettagliato, se digitate ls -f ottenete informazioni sul tipo di file, se digitate ls -a visualizzerete anche i files nascosti, ls -i vi mostrerà l'inode associato al file;

## **cp**

il comando cp serve per copiare un file, ad esempio:

cp nomefile1 nomefile2;

mentre cp nomefile1 dir copia il file nomefile1 nella directory dir;

## **cp -r**

copia le directory ricorsivamente;

### **cp -a**

si utilizza per copiare i file mantenendo intatti i permessi;

### **mv**

il comando mv si usa per rinominare o spostare un file, ad esempio:

mv nomefile1 nomefile2 rinomina nomefile1;

mv nomefile1 dir sposta nomefile1 in dir;

### **rm**

il comando rm cancella un file definitivamente e non potrà essere recuperabile quindi

okkio!!!!!!!!!!!!!!!!!!!!!!.

Con l'opzione -i vi sarà chiesta conferma prima dell'eliminazione del file;

### **cd**

il comando cd serve per cambiare directory, ad esempio:

cd test (supponendo che abbiate normalmente aperto un terminale all'interno della vostra home e che abbiate una directory test sempre nella vostra "home");

### **cd**

da solo vi porta alla vostra home;

### **cd ..**

vi riporta alla directory precedente (okkio allo spazio!);

### **du -h**

il comando du -h visualizza la quantità di spazio usata nella directory e sottodirectory in "kb" e "mb";

### **df**

il comando df mostra le partizioni montate;

### **top**

il comando top mostra i processi in uso dalla cpu (per interrompere Ctrl+c);

### **ps**

mostra una selezione dei processi attivi;

### **ps aux**

il comando ps aux Visualizza i processi e demoni in corso con i relativi pid;

### **kill**

il comando kill uccide un processo;

### **kill -9**

il comando kill -9 forza la chiusura del processo;

### **killall**

il comando killall uccide tutti i processi;

### **mkdir**

il comando mkdir crea una nuova directory, ad esempio:

mkdir test;

### **rmdir**

il comando rmdir elimina la directory dir supponendo che essa sia vuota, se non lo è potete comunque cancellarla con rm -rf dir;

### **pwd**

il comando pwd vi mostra dove siete... provate per vedere cosa vi riporta il terminale;

### **diff**

il comando diff vi mostra le differenze fra due file, ad esempio:

```
diff nomefile1 nomefile2;
```

### **file**

se non siete certi del formato di un file provate a scrivere:

```
file nomefile e vedete cosa vi riporta il terminale;
```

### **less**

il comando less è molto usato per visualizzare un file e spostarsi all'interno del testo con le frecce “su” e “giù” della tastiera, comodo anche se ad esempio non volete aprire un editor di testo, esiste anche un'altro comando che è more ma less è più potente e più utilizzato, provate a digitare:

```
less nomefile e premete invio per scorrere il testo del file;
```

### **ln -s**

il comando ln -s si usa per creare un link simbolico;

### **df**

il comando df mostra le partizioni montate;

### **free**

il comando free mostra la memoria ram e swap;

### **find**

il comando find si usa per cercare un file, ad esempio:

```
find nomefile dir;
```

### **head**

il comando head visualizza rapidamente una porzione di un file, ad esempio:

```
head /home/stefano/nomefile vi mostra le prime righe di questo file;
```

### **tail**

il comando tail visualizza anch'esso una porzione di file, ad esempio:

```
tail /home/stefano/nomefile vi mostra le ultime righe di questo file;
```

### **touch**

il comando touch crea un file, ad esempio:

```
touch nomefile crea nomefile;
```

### **grep**

il comando grep è molto utile per cercare ad esempio una voce all'interno di un file ed è comodissimo quando si deve lavorare su più files simultaneamente, ad esempio per cercare la parola root all'interno di tutti i file che sono in una determinata directory provate a scrivere:

```
grep root /dir/*;
```

### **whoami**

mostra l'utente associato all'ID corente;

### **uname -a**

il comando uname

-a vi mostra la versione di GNU/Linux in uso sul vostro sistema;

### **dmesg**

Il comando dmesg ripropone i messaggi visualizzati dal kernel durante l'avvio del sistema.

### **shutdown -r now**

il comando shutdown -r now riavvia il vostro computer;

### **reboot**

come sopra;

### **halt**

il comando halt spegne il vostro computer;

### **fdformat**

il comando fdformat /dev/fd0 formatta il floppy disk;

/dev sta per device (cioè dispositivo che in questo caso è il floppy), fd0 è appunto il floppy disk visto da GNU/Linux;

### **eject**

eject /dev/device espelle il cd-dvd dal lettore ottico;

### **cal**

il comando cal visualizza il calendario;

### **lspci**

il comando lspci vi rende informazioni utili sull'hardware installato;

### **zip**

il comandi zip comprime un file ad esempio:

zip test.zip test.txt crea un file zippato a partire dal file test.txt, unzip test.zip estrae il file.

### **lpr**

il comando lpr manda in stampa un file;

### **man**

man è il paginatore dei manuali del sistema. Di solito ognuno degli argomenti pagina dati a man è il nome di un programma, di un'utility o di una funzione, la pagina di manuale associata con ognuno di questi argomenti viene poi trovata e mostrata;

### **exit**

il comando exit chiude il terminale (si usa anche per uscire da root e ritornare utente normale all'interno di un terminale aperto, senza dover aprire un'altra scheda o un'altro terminale).

Questa è solo una panoramica dei comandi base che utilizzerete e che vi permetteranno di lavorare col vostro computer tramite il terminale, ricordate che la shell è molto potente, vi può facilitare il lavoro ma potete anche fare danni se non la utilizzate con la consapevolezza di quello che state facendo nel

momento in cui digitate un certo comando. Cercate di prendere confidenza coi comandi e non eseguite operazioni potenzialmente pericolose come la cancellazione di file di sistema e di configurazione, fate sempre un backup dei vostri file prima di effettuare delle modifiche, almeno fino a che non avrete acquisito una certa padronanza del vostro sistema GNU/Linux.

**Enjoy with Gnu/Linux...**



Questo/a opera è pubblicata sotto una [Licenza Creative Commons](#).