

# Il magico mondo della console $\frac{1}{2}$

## Introduzione alla shell e ai principali comandi unix

Francesco Versaci

`webmonster -apud- apf.it`

*MonteLUG*

Montebelluna Linux User Group



# Licenza d'utilizzo

Copyright © 2006, Francesco Versaci.

Questo documento viene rilasciato secondo i termini della licenza Creative Commons (<http://creativecommons.org>).

L'utente è libero di:

***distribuire, comunicare al pubblico, rappresentare o esporre in pubblico la presente opera***

alle seguenti condizioni:

**Attribuzione** Deve riconoscere la paternità dell'opera all'autore originario.

**Non commerciale** Non può utilizzare quest'opera per scopi commerciali.

**No opere derivate** Non può alterare, trasformare o sviluppare quest'opera.

In occasione di ogni atto di riutilizzazione o distribuzione, deve chiarire agli altri i termini della licenza di quest'opera.

Se ottiene il permesso dal titolare del diritto d'autore, è possibile rinunciare a ciascuna di queste condizioni. Le utilizzazioni libere e gli altri diritti non sono in nessun modo limitati da quanto sopra. Questo è un riassunto in lingua corrente dei concetti chiave della licenza completa (codice legale), reperibile sul sito Internet

<http://creativecommons.org/licenses/by-nc-nd/2.0/legalcode>



# Sommario

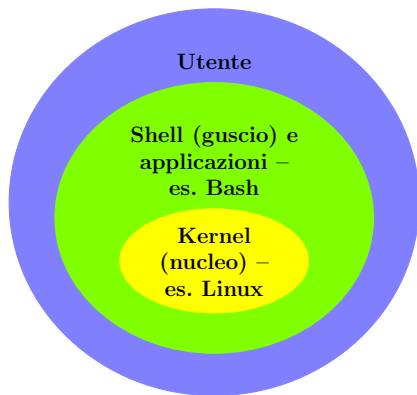
- 1 La shell Bash
  - Generalità
  - Gestione base dei processi
- 2 Comandi
  - Comandi base
  - Gestione avanzata dei processi



# Cos'è una shell?

## A che serve?

- Interagire col kernel
- Gestire file e processi in modo potente e veloce
- Lanciare applicazioni
- Automatizzare operazioni frequenti e/o complesse



Schema a cipolla di un sistema operativo



# Bash – La shell GNU

# BASH

## Caratteristiche principali

- Segue lo standard POSIX
- Disponibile per molti sistemi operativi
- Programmabile, personalizzabile

## Login utente

```
spongebob login: cesco
Password:
cesco@spongebob:~$ pwd
/home/cesco
cesco@spongebob:~$ logout
spongebob login:
```

## Login root

```
spongebob login: root
Password:
spongebob:~# pwd
/root
spongebob:~# logout
spongebob login:
```



# Comandi

## Sintassi generale

```
$ nome-comando <opzioni> <argomenti>
```

## Spiegazione

**opzioni** modificano l'esecuzione del programma

- --nome-opzione
- -o

**argomenti** bersagli del comando, di solito nomi di file

## Esempi

```
$ uname  
Linux  
$ uname --machine  
i686  
$ uname -m  
i686  
$ wc -l /etc/fstab  
12 /etc/fstab
```



# Espansioni in linea di comando

## Sostituzione comando

```
$ dpkg -S `which import`  
imagemagick: /usr/bin/import
```

## Caratteri *jolly*

- \* qualsiasi stringa (nulla inclusa)
- ? carattere qualsiasi
- [...] carattere in intervallo
- [^...] carattere non in intervallo

## Espansione delle graffe

```
$ touch {g,p,tr}ino  
$ ls  
gino pino trino
```

## Espansione di percorso

```
$ ls ?ino  
gino pino  
$ ls *ino  
gino pino trino  
$ ls [a-m]ino  
gino  
$ ls t*i*  
trino
```



# Quoting

## Caratteri usati

- backslash** per caratteri singoli
- virgolette** permettono di indicare nomi con spazi, ma non bloccano le sostituzioni di comando
- apici** bloccano anche le sostituzioni comando

## Esempi

```
$ touch "sale e pepe"
$ touch olio\ e\ aceto
olio e aceto sale e pepe
$ echo "$(ls)"
olio e aceto
sale e pepe
$ echo '$(ls)'
$(ls)
```





# Alias

## Uso degli alias

- Per velocizzare la digitazione di comandi frequenti
- Possono aggiungere opzioni di default ai comandi
- Possono essere aggirati col quoting

## Esempi

```
$ alias lc='ls --color=auto'  
$ alias ll='ls -l'  
$ alias mutt='mutt -y'  
$ mutt #==> mutt -y  
$ 'mutt' #==> mutt  
$ unalias ll
```



# Variabili d'ambiente

## A che servono?

- Per impostare opzioni per la shell
- Per impostare opzioni predefinite per alcuni programmi

## Esempio

```
$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/
games
$ export PATH=~ /mybin:$PATH
$ echo $PATH
/home/cesco/mybin:/usr/local/bin:/usr/bin:/bin:
/usr/bin/X11:/usr/games
```



# File di configurazione

## ~/ .bashrc

- Eseguito da shell non di login
- Quello globale è  
/etc/bash.bashrc

## Esempio

```
alias ls='ls --color=auto'  
alias ll='ls -ctrl'  
alias rm='rm -i'  
. /etc/bash_completion
```

## ~/ .bash\_profile

- Eseguito da shell di login (no terminali)
- Quello globale è  
/etc/profile

## Esempio

```
. ~/.bashrc  
PATH=~/.bin:"${PATH}"  
umask 077
```



# History

## Archivio

Nel file `~/.bash_history` vengono memorizzati gli ultimi `$HISTSIZE` comandi.

## Combinazioni da tastiera

**frecce su/giù** scorrono l'history  
**CTRL-R** consente la ricerca all'indietro

## Esempio

```
$ less /etc/fstab
$ ...
$ ...
$ <CTRL-R>
(reverse-i-search) 'fs': less /etc/fstab
```



# Esecuzioni in background

## Diverse esecuzioni

**foreground** esecuzione in primo piano, blocca la shell

**background** esecuzione in sfondo, lascia libera la shell; si invoca aggiungendo il carattere & alla fine del comando

## Esempio

```
$ firefox &  
[1] 7984  
$ ...  
$ <chiudiamo firefox>  
$ ...  
[1]+  Done      firefox
```



# Cambiamento della modalità di esecuzione

## Combinazioni da tastiera

Mentre un processo è in esecuzione in foreground si può alterarne l'esecuzione con

**CTRL-\** si chiede al processo di uscire

**CTRL-C** viene ucciso il processo

**CTRL-Z** il processo viene sospeso e restituita la shell

## Riattivazione

Per riattivare un processo:

**bg** viene mandato in background

**fg** viene mandato in foreground

## Esempio

```
$ mc <CTRL-Z>
[1]+  Stopped          mc
$ vim <CTRL-Z>
[2]+  Stopped          vim
$ fg 1 # si riavvia mc
```

# Canali di flusso dei processi

## I 3 canali principali

- `stdin` canale di input
- `stdout` canale di output
- `stderr` canale per messaggi d'errore

## Redirezione dei canali

- `< nome-file` reindirizza input
- `> nome-file` reindirizza output
- `2> nome-file` reindirizza stderr
- `>> nome-file` aggiunge al file se già esistente

## Esempio

```
$ echo "6*7" > f.txt
$ echo "10+3" >> f.txt
$ cat f.txt
6*7
10+3
$ bc < f.txt
42
13
$ cat /etc/shadow > p 2>&1
$ # = cat /etc/shadow &> p
$ cat p
cat: /etc/shadow: Permissi
on denied
```



# Pipeline e sequenze

## Pipeline

Si può anche collegare lo *stdout* di un processo con lo *stdin* di un altro, utilizzando il carattere `|` e formando una pipeline.

## Esempio

```
$ ps -e | grep bash
7191 pts/3 00:00:00 bash
8582 pts/5 00:00:00 bash
8732 pts/5 00:00:00 bash
```

## Sequenze

Per combinare comandi:

```
com1 ; com2 com2 è
                    eseguito dopo
                    com1
```

```
com1 && com2 idem, ma
                    solo se com2 non
                    ha dato errori
```

## Esempi

```
$ sleep 20m; sudo halt

$ ./configure && make
```



## echo e touch

## echo

- Stampa una stringa nello *stdout*
- Utile negli script e nelle sequenze di comandi

## Esempio

```
$ make && echo "Fatto..."
```

## touch

- Crea nuovi file vuoti
- Cambia data di accesso o modifica di file esistenti

## Esempi

```
$ touch main.cc ; make  
  
# mount -o remount,ro /  
# touch me  
touch: cannot touch `me':  
Read-only file system  
# fsck /
```



## cat e less

## cat

- Visualizza uno o piú file sullo *stdout*
- Utile per unire piú file (vedi `split`)
- Visualizza su schermo file piccoli
- Può essere usato anche per scrivere piccoli file

## Esempi

```
$ cat /etc/fstab
$ cat >> TODO.txt
```

## less

- Visualizzatore di file
  - Usato dal `man`
- `/` Ricerca nel file
- `less -i` Abilita ricerca case-insensitive

## Esempio

```
$ dmesg | less
```



# head e tail

## head

Visualizza le prime righe di un file

```
head -n n
```

 Mostra le prime n righe

## Esempio

```
$ find . | sort | head -3  
.  
./ .abook  
./ .abook/addressbook
```

## tail

- Visualizza le ultime righe di un file
- Utile per seguire file di log

```
tail -n n
```

 Mostra le ultime n righe

```
tail -f
```

 Aspetta nuovi caratteri dal file

## Esempio

```
# tail -f /var/log/messages  
$ tail -f wget-log
```



# help, man e info

## help

Aiuto sui comandi interni `bash`

## man

Aiuto per programmi e comandi

`man -L en` Richiede il manuale in lingua inglese

## Esempi

```
$ help for
$ man -L en vim
$ gunzip -c bash.1.gz | gro
ff -man > ~/bash-man.ps
```

## info

- Manuali in ipertesto
- Spesso piú completi
- Non molto usato

`<INVIO>` Segue il link  
`l` (L minusc.) Torna indietro  
`/` Ricerca

## Esempi

```
$ info wget
$ info find
```



## pwd e cd

## head

Stampa la directory corrente

```
pwd -P stampa
directory fisica,
ignorando i
symlinks
```

## Esempio

```
$ pwd
/home/cesco/mp3
$ pwd -P
/store/audio/mp3
```

## cd

Cambia directory

```
cd torna alla home
cd - torna alla directory
precedente
```

## Esempio

```
$ pwd
/home/cesco/mp3
$ cd /etc
$ pwd
/etc
$ cd -
/home/cesco/mp3
```



## ls

## Sintassi...

Visualizza contenuto directory e informazioni file

- `ls -l` visualizza anche informazioni
- `ls -R` mostra ricorsivamente sottodirectory
- `ls -a` mostra anche file nascosti (`.nomefile`)
- `ls -t` ordina secondo orario
- `ls -c` ordina secondo orario di modifica
- `ls -r` inverte ordine

## ...sintassi

- `ls -d` mostra directory senza entrarci
- `ls -h` human-readable, dimensioni leggibili (es. 9,5M)
- `ls --color` usa colori

## Esempi

```
$ touch {p,g,l}ino
$ mkdir dir{1,2,3}
$ ls -d */
dir1/ dir2/ dir3/
$ ls -ctrl ~/downloads
```

## cp e mv

## cp

Serve per copiare file o directory

- `cp -R` copia directory ricorsivamente
- `cp -p` preserva attributi
- `cp -v` elenca file durante la copia
- `cp -u` copia solo file piú recenti

## mv

Sposta e/o rinomina file

- `mv -v` elenca file durante lo spostamento
- `mv -i` chiede conferma prima di una sovrascrittura

## Esempi

```
$ cp -v /media/usdisk/*.jpg ~/foto  
$ mv vecchio-nome.jpg nuovo-nome.jpg
```



## rm e ln

## rm

Cancella file e directory

- `rm -R` cancella ricorsivamente le sottodirectory
- `rm -i` chiede conferma a ogni file
- `rm -f` forza la rimozione

## Esempio

```
$ rm -R .[^.]*
```

## ln

Crea link fra file, alias per raggiungerli da diversi percorsi

- `ln -s` crea link simbolico (il piú usato)
- `ln -f` cancella un eventuale link già esistente

## Esempio

```
$ ln -s /store/mp3 ~/mp3
```





## df e du

## df

Mostra spazio libero dei dispositivi

```
df -h human-readable
```

```
df -T mostra tipo dei  
filesystem
```

## du

Mostra spazio usato da file

```
du -h human-readable
```

```
du -s non mostra file  
ricorsivamente
```

## Esempi

```
$ df -Th
Filesystem      Type Dimens. Usati Disp. Uso% Montato su
/dev/hda5       ext3   5,5G   2,1G   3,2G   40% /
/dev/hda6       ext3   3,7G   328M   3,4G    9% /home
/dev/hda8       ext3   27G    2,1G   24G    9% /store
tmpfs           tmpfs  118M    0     118M   0% /dev/shm
$ du -sh ~
264M    /home/cesco
```



## ps e top

ps

Visualizza i processi attivi

```
ps -u nome
```

 mostra processi  
di nome

```
ps -e
```

 mostra tutti i  
processi

```
ps ax
```

 mostra tutti i  
processi con linea  
di comando

## Esempio

```
$ ps -u cesco
```

top

Programma grafico per gestire i  
processi attivi

```
? aiuto
```

```
s scegli tempi di  
aggiornamento
```

```
F ordina
```

```
u filtra per utente
```

```
k manda segnale
```

```
r cambia priorità
```

## Esempio

```
$ top
```

## su e nice

su

Apri una shell come altro utente (*root* di default)

su - Apri una shell di login

## Esempi

```
$ su -  
Password:  
# su - postgres  
$ psql
```

nice

Avvia un comando con una certa priorità

**priorità** numero da -20 (max. priorità) a 20 (minimo)

**permessi** solo *root* può avviare a priorità negative

nice -n Avvia con priorità n

renice n PID Cambia priorità

## Esempio

```
$ nice -15 make  
# nice --10 xcdroast
```



# kill e killall

## kill

Manda un segnale a un processo

```
kill -s PID Invia segnale s
```

## killall

Manda un segnale a tutti i processi con lo stesso nome

```
killall -s nome Invia  
segnale s
```


## Segnali

- 15 Termina (default)
- 3 Quit
- 9 Uccidi
- 19 Ferma
- 18 Riattiva

## Esempio

```
$ ps -e | grep firefox  
4888 ? 00:06:29 firefox  
$ kill -15 4888  
$ killall -15 firefox
```

# Lectures recommended for deepening...

 **Brian Fox e Chet Ramey**  
*Manuale bash*

`man bash`

 **Autori Vari**

*Tutte le pagine man* 😊

```
find /usr/share/man -name '*. [1-9].gz'  
-exec man -l {} \;
```

