

Linux dalla *console* – 1

Shell e Bash, flussi, filtri, gestione file e cartelle,
filesystems

Emiliano Vavassori

8 marzo 2017

Spazio Polaresco, Bergamo

Bergamo Linux Users Group

Sommario

Definizioni

Primi passi

Flussi

Filtri

bash: funzioni utili

Gestione file e cartelle

Permessi

Filesystem

Definizioni

Console, terminale, shell, bash...

Console Wikipedia rimanda a «Riga di comando»

Riga di comando Interfaccia a riga di comando: una tipologia di interfaccia utente di tipo testuale tra utente ed elaboratore

CLI *Command Line Interface*

Virtual Terminal Terminale virtuale

Terminale Dispositivo hardware per inserire dati di input ad un computer e riceverli in output; anche l'emulatore di terminale

Shell invece...

Interprete dei comandi; parte di un sistema operativo che permette agli utenti di impartire **comandi** ed avviare altri programmi.

...e quindi bash?

bash (*Bourne Again Shell*, 1989) è solamente una delle shell a disposizione, ma ce ne sono molte altre:

- **sh**: *Bourne Shell*, da Stephen Bourne (AT&T), 1970
- **csh**: *C Shell*, più simile al linguaggio C, tardi '70
- **tcsh**: *TENEX C Shell*. Shell di Mac OS X fino a 10.3
- **ksh**: *Korn Shell*, compatibile con **sh**, da David Korn, 1983
- **rc**: *Run commands*, nata per scripting e per Plan 9
- **busybox**: non è solo una shell, Bruce Perens, 1996
- **zsh**: *Z Shell*, Paul Falstad, 1990

Utilizzeremo comunque **bash** perché è quella più diffusa.

Primi passi

Demo: console ed emulatore di terminale

Prima di farsi male

```
$ man <comando>
```

```
$ info <comando>
```

```
$ apropos <qualcosa>
```

```
$ <comando> -h
```

```
$ <comando> --help
```


Attenti!

```
$ <comando> -a -b -c <parametro> -d<parametro> \  
-fg <argomento1> <argomento2>
```

```
$ <comando> --opzione-lunga \  
--opzione-con-parametro=<parametro> \  
--altraopzione <parametro> <argomento1> \  
<argomento2>
```

Primi comandi e interazione di base

```
$ ls
```

```
$ echo 'Ciao mondo!'
```

```
$ echo "Ciao mondo!" > ciaomondo.txt
```

```
$ cat ciaomondo.txt
```

Flussi

- `STDIN`, *standard input*
- `STDOUT`, *standard output*
- `STDERR`, *standard error*

- > Redirezione di STDOUT
- >> Appendere STDOUT alla destinazione
- tee** Copia STDOUT e lo redirige
 - < Redirezione di STDIN
 - << Appendere STDIN alla destinazione
 - 2> Redirezione di STDERR
 - &1** Puntatore a STDOUT
 - | *Pipe*, redirige lo STDOUT della sorgente sullo STDIN della destinazione

Lavorare con i flussi: esempi

```
$ patch < nomefile.txt > output.txt 2> /dev/null
```

```
$ ls -lAh | grep txt | tee lista.txt
```

```
$ comando_molto_verboso.sh > /dev/null 2>&1
```

Codici d'uscita

Il *codice d'uscita* o *valore di ritorno* di un eseguibile è un valore che dice se c'è stato un errore durante l'esecuzione del programma precedente oppure no.

```
$ ./script.sh; echo $?
```

Il valore *0* significa che il programma è stato eseguito senza alcun errore. Qualsiasi valore uguale o superiore a *1* è un codice d'uscita valido per segnalare un errore.

Filtri

Filtri: peculiarità di *nix

```
$ date --iso | tr -d '-'
```

```
$ ps aux | grep python | cut -d' ' -f 2
```

```
$ ls -lAh | grep root | cut -d' ' -f 1
```

bash: funzioni utili

History e richiamo comandi

```
$ history
```

Freccia su, Freccia giù: comando precedente, successivo

```
$ !!
```

```
$ !date
```

```
$ ^hosst^hosts
```

Ctrl+A, Ctrl+E: inizio, fine riga

Alt+B, Alt+F (Ctrl+Freccia Destra, Ctrl+Freccia Sinistra): inizio,
fine parola

Modifica sulla riga di comando

Alt+T: inverti posizione parola

Ctrl+T: inverti posizione caratteri

Ctrl+K: taglia fino alla fine della riga

Alt+U, Alt+L: tutto maiuscolo, tutto minuscolo parola

Tab: tasto di autocompletamento

bash-completion: pacchetto che abilita
l'autocompletamento su alcuni comandi ben noti

Gestione file e cartelle

Tutto in *nix è un file...

```
$ ls /dev
```

```
$ cat /proc/cpuinfo
```


Creazione di file

```
$ touch file.txt
```

```
$ echo "ciccio" > file.txt
```

```
$ cat > file.txt  
Ctrl+D
```

```
$ $EDITOR file.txt
```

Visualizzazione dei file

```
$ cat file.txt
```

```
$ more file.txt
```

```
$ less file.txt
```

```
$ $EDITOR file.txt
```

Creazione ed eliminazione delle directory

```
$ mkdir directory
```

```
$ mkdir -p dir/con/molte/subdirectory
```

```
$ rmdir directory
```

```
$ rm -rf dir/
```

Permessi

Teoria dei permessi

Soggetti dei permessi:

Proprietario *owner*, utente che ha la proprietà di un file

Gruppo proprietario *group*, gruppo di utenti proprietari

Altri *others* Tutti gli altri utenti del sistema

Possibili permessi:

read, r, 4 Permesso di lettura

write, w, 2 Permesso di scrittura (anche creazione e cancellazione)

execute, x, 1 Permesso di esecuzione

E poi ci sono le ACL (*Access Control Lists*)...

Rappresentazione dei permessi

```
$ ls -lAh
drwxr-xr-x. 2 root root 4096 27 gen 20.32 mibs
-rw-r--r--. 1 root root 293 27 gen 20.32
README.txt
-rwxr-xr-x. 1 root root 146 27 gen 20.32
snmpgetnext.sh
```

Cambiare i proprietari

```
$ chown user:group file.txt
```

```
$ chgrp users file.txt
```

Cambiare i permessi

Rappresentazione logica

```
$ chmod u+rw,g-w+r file.txt
```

```
$ chmod a+x script.sh
```

Rappresentazione ottale:

```
$ chmod 640 file.txt
```

```
$ chmod 755 script.sh
```


Cenno alle ACL

```
$ getfacl vagrant/  
# file: vagrant/  
# owner: root  
# group: www-data  
user::rwx  
group::r-x  
group:users:rwx  
mask::rwx  
other::r-x  
default:user::rwx  
default:group::r-x  
default:group:users:rwx #effective:r-x  
default:mask::r-x  
default:other::r-x
```

Filesystem

FHS (*Filesystem Hierarchy Standard*) è uno standard che definisce, all'interno dei sistemi *Unix-like*, la struttura e le directory principali dei *filesystems*.

È importante conoscerlo per potersi muovere nelle directory di sistema.

Directory principali

- bin** Binari utente essenziali
- boot** File statici per l'avvio
- dev** File dei dispositivi
- etc** Configurazioni di sistema (specifiche per host)
- lib** Librerie condivise essenziali e moduli del kernel
- media** Punto di montaggio dispositivi rimovibili
- mnt** Punto di montaggio temporaneo dei filesystem
- opt** Pacchetti software aggiuntivi
- sbin** Binari di sistema essenziali
- srv** Dati forniti dai servizi di sistema
- tmp** File temporanei
- usr** Gerarchia secondaria
- var** Dati variabili

- bin** Buona parte dei programmi utente
- include** *Header* dei programmi C
- lib** Librerie per i programmi utente
- local** Gerarchia locale
- sbin** Binari di sistema non essenziali
- share** Dati indipendenti dall'architettura

Casi particolari: `var`

- cache** Dati di *cache* delle applicazioni
- lib** Informazioni di stato variabili
- local** Dati variabili per `/usr/local`
- lock** File di blocco
- log** File di log
- opt** Dati variabili per `/opt`
- run** Dati rilevanti per l'esecuzione dei processi
- spool** Dati di *spool* delle applicazioni
- tmp** File temporanei da conservare fra reboot

`/proc` Informazioni sui processi e sul kernel

`/sys` Contiene la maggior parte delle informazioni sui dispositivi