

Corsi Linux 2018

Practical BASH

Niccolò Izzo

izzo.niccolo@gmail.com



Queste slides le trovate su

slides.poul.org/2018/corsi-linux-avanzati/PracticalBash



Gli Hacker Usano il Terminale

- È più veloce
- È universale
- È flessibile
- È automatizzabile
- È **più fico**

Cos'è una Shell?

Interfaccia testuale dotata di 3 **canali standard**:

- Testo in uscita: `stdout`
- Testo in ingresso: `stdin`
- Messaggi di errore: `stderr`

Esempi

- Bash → linux & macOS
- zsh → linux & macOS
- cmd.exe → Windows
- PowerShell → Windows

Bash

- Acronimo di bourne again shell
- Creata da Chet Ramey nel 1989
- È la shell oggi più diffusa
- Dispone di un linguaggio di scripting

zsh

- Sviluppata da Peter Stephenson nel 1990
- Tante belle cose per lavorare meglio:
 - Autocompletamento programmabile
 - Temi e Plugin
- Esperienza magnifica con **oh-my-zsh**

Usare una Shell

comando [OPZIONI] [ARGOMENTI]

- Tutto ciò che è tra parentesi quadre è facoltativo
- Le opzioni si passano con uno o due trattini (- o -)

Esempio:

```
hack_ip --stealth -p 3210 127.0.0.1
```

Gotta go Faster

- *TAB* ↔: Autocompletamento
- *CTRL+C*: Interrompi il processo
- *SHIFT+CTRL+C*: Copia
- *SHIFT+CTRL+V*: Incolla

DON'T PANIC

man

C'è e ci sarà quasi sempre per spiegarvi i comandi complessi:

- Per cercare in una man page: /parolachiave e ↵
- Per passare al prossimo risultato: n
- Per passare al risultato precedente: SHIFT+n
- Per uscire da man: q

Esempio:

man man

Esplorare il File System

In ogni momento la shell è posizionata in una cartella

- `pwd` ne stampa il nome
- `ls` elenca i file al suo interno
- `ls -hal` mostra più informazioni
- `cd [nomecartella]` permette di passare ad un'altra

Ricordate che:

- `.` è la cartella corrente
- `..` è la cartella un livello sopra
- `~` è la nostra home

Filesystem Hierarchy Standard

È uno **standard** per l'organizzazione del file system.

Le distribuzioni più comuni si ispirano ad esso:

- /bin file **binari** per tutti gli utenti
- /boot file per il **bootloader**
- /dev **periferiche**
- /etc file di **configurazione** del sistema
- /opt pacchetti **software statici**
- /proc informazioni del **kernel** (fs virtuale)

File e Cartelle

- `touch NOMEFILE` crea un file vuoto se non esiste
 - Se esiste ne aggiorna i metadati sull'ultimo accesso
- `mkdir NOMECARTELLA` crea una cartella vuota
 - `-p` crea tutte le cartelle del path fornito

Ricordate che il vostro filesystem è **case sensitive**:

Quindi `POuL` \neq `poul` \neq `POUL`.

Io ti ho creato, io ti distruggo

- `rm NOMEFILE` cancella un file
- `rmdir CARTELLA` cancella una cartella **vuota**
- `rm -r` cancella una cartella **piena**

Questi comandi eliminano i file **per sempre**, per non rischiare possiamo usare:

- `trash` rimuove un file in maniera non permanente
 - Distribuito nel pacchetto `trash-cli`
 - `trash-restore` per riprendere i file eliminati

Gestire i file

- `cp` `SORGENTE` `DESTINAZIONE` copia un file
 - `-r` per copiare una directory
- `mv` `SORGENTE` `DESTINAZIONE` sposta un file
 - Se la destinazione è nella stessa cartella rinomina i file
- `ln -s` `SORGENTE` `PERCORSO` crea un collegamento simbolico al file `SORGENTE` in `PERCORSO`

Permessi

- L'utente amministratore è chiamato **root**
- In Linux ad ogni **shell è associato un utente**
- Gli utenti possono appartenere ad **uno o più gruppi**
- Ogni file **appartiene ad un utente** ed un gruppo
- Chi possiede un file può **cambiarne i permessi**
- I permessi regolano chi può leggere **r**, scrivere **w**, e eseguire **x** ogni file

drwxrwxrwx!

Per ogni file, i permessi per l'utente proprietario, il gruppo proprietario e gli altri sono espressi come:

- rwxrwxrwx
- Tre cifre in notazione ottale

I permessi si modificano con:

```
chmod ([augo][+ -=][rwx])+
```

Per cambiare il proprietario di un file:

```
chown UTENTE[:GRUPPO]
```

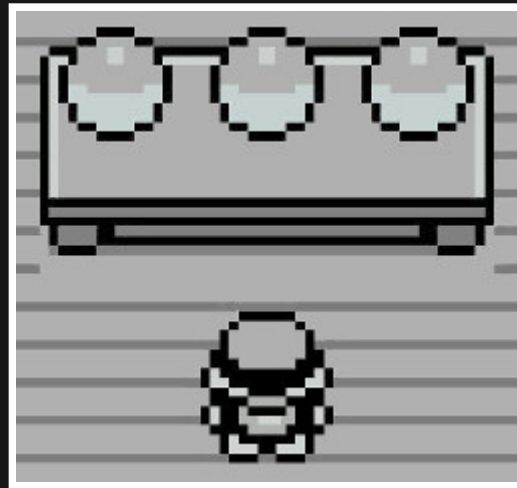
sudo

Permette di eseguire comandi come **root**.

- Solitamente è meglio **usare un terminale non-root**
- `sudo` comando esegue il comando **come se fossimo root**
- Il **file di configurazione** di sudo è `/etc/sudoers`
- Per modificarlo usare **sempre** il comando `visudo`

Editor di Testo

Strumento fondamentale del sysadmin. È importante sceglierne uno e **amarlo**.



nano è semplice da usare ma limitato.

vi e **emacs** sono complessi e potenti.

lo ho scelto vi

Perché lo trovate in ogni unix-like,
infatti è richiesto dalla **Single UNIX Specification**.

vi basics

Useremo `vim`, una versione migliorata di `vi`.

`vi` ha diverse modalità di funzionamento, le due fondamentali sono:

- `INSERT` per inserire il testo
- `NORMAL` per manipolare il testo

Premendo `i`, `o` oppure `a` si passa da `NORMAL` a `INSERT`.

Premendo `ESC` si passa da qualunque modalità a `NORMAL`.

Modalità NORMAL

- *h, j, k, l* sono le vostre **frecce**
- */* **ricerca** nel documento, come man
- *u* **annulla** l'ultima operazione
- *.* **ripete** l'ultima operazione
- *:* inizia un **comando**
- *:wq* per **salvare ed uscire** dal documento

Pro tip: tornare sempre a NORMAL dopo aver inserito del testo

Plugins

Il vero potere di vim è la sua estensibilità.

Un **Plugin Manager** è un buon punto di partenza.

Io consiglio **vim-plug** disponibile su **GitHub**.

Da provare:

- vim-airline
- auto-pairs
- gitgutter
- flake8
- table-mode

Bash++

- `cat` concatena e stampa file su schermo
- `less` visualizza un file in una finestra scorrevole
- `echo` stampa il valore di un'espressione
- `find` cerca un file nel filesystem

Standard streams

`stdin`, `stdout` e `stderr` sono le strutture base per usare bene le shell unix.

- `< FILE` connette `stdin` di un processo ad un file
- `> FILE` connette `stdout` di un processo ad un file
- `2> FILE` connette `stderr` di un processo ad un file
- `&>` connette `stdout` e `stderr` ad un file
- Questi comandi di default **sovrascrivono** il file
- `>>` se usato al posto di `>` per **aggiungere in coda**
- `|` connette `stdout` di un processo a `stdin` di un altro

Quasi come il C

- `comando1 && comando2` esegue `comando2` se e solo se **il primo ha successo**
- `comando1 || comando2` esegue `comando2` se e solo se **il primo fallisce**
- `comando1; comando2` li esegue **entrambi**

Gestire i processi

Dopo aver eseguito un comando:

- CTRL+Z lo sospende
- bg riprende l'esecuzione in background
- fg riprende l'esecuzione in foreground
- comando & avvia un processo in background

Filtri

Un filtro è un programma che:

- Legge dati in ingresso sullo `stdin`
- Trasforma questi dati
- Produce dell'output su `stdout`

I filtri Unix sono **blocchi funzionali semplici** che possiamo combinare per realizzare **logiche complesse**

cut

Estrae colonne da ogni riga di un file secondo un delimitatore

- - d specifica il delimitatore (default Tab)
- - f specifica quale colonna estrarre (one-based)

sort

Ordina le righe di un file

- - k specifica quali colonne del file usare come chiave
- - t specifica il delimitatore tra le colonne (default “ “)

uniq

Stampa le righe uniche di un file già ordinato

- -c conta le **occorrenze**
- -d mostra solo i **duplicati**
- -u mostra solo i **non duplicati**

WC

Conta righe, parole e caratteri

- -l mostra solo il numero di **righe**
- -w mostra solo il numero di **parole**
- -c mostra solo il numero di **caratteri**

tee

Come un tubo a T, scrive il suo `stdin` sullo `stdout` e su uno o più file

Utile per mostrare un comando a schermo e fornirlo come input ad un altro comando

head/tail

Mostrano le prime/ultime 10 righe di un file:

- `-nX` mostra le prime/ultime X righe
- `tail -f` permette di *tenere d'occhio* un file a cui vengono aggiunte righe in coda (ad esempio un log)

Gestione del Software

- In linux **non serve un browser** per scaricare il software
- Ogni distribuzione mantiene delle **repository**
- Le repository possono essere **binarie** o a **sorgenti**
- Tramite i **package manager** possiamo scaricare e aggiornare i software

Azione	Ubuntu/Debian	Fedora	ArchLinux
Aggiorno i repo	apt update	dnf makecache	pacman -Syy
Scarico un pacchetto	apt install NOME	dnf install NOME	pacman -S NOME
Rimuovo un pacchetto	apt remove NOME	dnf remobe NOME	pacman -R NOME
Aggiorno i pacchetti	apt upgrade	dnf upgrade	pacman -Syu

Domande?

Grazie per l'attenzione!



<https://www.poul.org>