

13 - Raggruppamento di comandi e comandi utili

Racchiudendo tra parentesi tonde () una sequenza di comandi allora **verrà chiamata una subshell al fine di eseguire i comandi in esse specificati.**

Se inseriti tra parentesi tonde **i comandi al loro interno condividono GLI STESSI stdin, stdout e stderr, utilizzandoli, se necessario, uno dopo l'altro in modo consecutivo.** Il risultato di una sequenza di operazioni tra parentesi tonde è **il risultato dell'ultimo comando eseguito nella subshell**, ossia l'ultimo della sequenza.

```
ls; pwd; whoami > out.txt
# Output: visualizza i file della directory corrente, la mia directory
corrente e nel file out.txt trovo il mio username
(ls; pwd; whoami) > out.txt
# Output: non visualizzo niente, ma nel file out.txt troverò i file della
directory corrente, la mia directory corrente e il mio username.
```

bash ottimizza l'uso della concatenazione di comandi: se un solo comando è presente tra le parentesi, allora non viene creata una *subshell*.

Utilities di bash

Esiste una libreria di comandi bash che facilitano alcune operazioni, come per esempio la manipolazione di stringhe.

NOTA: Alcuni di questi comandi hanno un doppio comportamento: si può specificare un file a cui essi devono essere applicati, oppure accettano output da tastiera finché non viene inserito un carattere di terminazione. Questi comandi vengono specificati con 💡.

man

Il comando man , seguito da un altro comando, apre una specie di **manuale di istruzioni** per quello specifico comando; specificandone l'uso e i parametri accettabili. Esso è diviso in sezioni, sulla base di ciò che ogni comando è in grado di fare. Esiste anche il man di man .

head e tail 💡

I comandi head e tail mettono in output rispettivamente **la prima e l'ultima** parte di ciò che gli viene specificato. In particolare, nel caso dei file, vengono mostrate *di default le prime 10 righe* nel caso di head e **le ultime 10 righe** nel caso di tail . Il numero di righe da visualizzare è personalizzabile tramite parametri.

sed

Il comando `sed` è applicabile solo su file e rende possibile la **manipolazione di stringhe**. Ha numerosi utilizzi e funzioni:

- Può essere usato per rimuovere o aggiungere caratteri ad una riga
- Può essere usato per rimpiazzare certe occorrenze di una stringa con un'altra.
- Può essere usato per operare sul file per mezzo delle *regex*.
- Molto altro (`man sed`).

cut

Il comando `cut` viene utilizzato per considerare solo alcune porzioni di stringhe presenti all'interno di un file. È molto utile per alcuni formati di file che hanno più campi separati da delimitatori (quali virgolette o spazi), in quanto è possibile specificare il delimitatore tra campi e quale campo considerare una volta effettuata la separazione.

cat

Il comando `cat` visualizza sullo *standard output* il contenuto del file specificato - oppure se non specificato un file, inizierà a stampare ciò che inseriremo sullo *standard input*.

È il **comando di base** per leggere i contenuti di un file specifico.

grep

Il comando `grep` restituisce in output le righe di ciò che è passato in input **che rientrano nelle specifiche richieste dal comando stesso**. Con questo comando si è in grado di stampare le righe di un file che iniziano, finiscono, contengono (o meno) una certa serie di caratteri o di parole. Molto utile in concatenazione pipe con `cat`, cosicché una volta stampati i contenuti del file si considerino solo le righe che si preferiscono.

tee

Il comando `tee` esegue due operazioni contemporaneamente: **stampa sia in standard output che su un file specificato ciò che gli viene passato come input**.

È utile per memorizzarsi delle informazioni sulla base dell'output di altri file.

wc

Il comando `wc` conta i byte, le parole e le righe di ciò che gli viene inserito in input. È possibile specificare qualcosa da standard input, oppure specificare direttamente un file.

Particolarmente forte in combinazione con `grep` e `cat` per filtrare specifiche linee, oppure per sapere quanti dati sono da processare.