

Riassunto comandi latex

Matteo Mistri

27 aprile 2018

Indice

1	Introduzione	2
2	Comandi	3
2.1	Cartelle e file	3
2.2	File	4
2.3	Permessi	4
2.4	Utility	5
3	Programmazione	6
3.1	Comandi	6

Capitolo 1

Introduzione

Windows → tutto è un oggetto(c++)

Linux → tutto è un file

Perfino i processi sono descritti da file nelle cartelle denominate con il PID.

La filosofia è avere molti programmi piccoli e ottimizzati che fanno poche cose.

Quello che in windows è un servizio, in Linux è un deamon.

Il simbolo | permette di passare un output come input di un comando successivo.

I nomi delle variabili non dovrebbero contenere spazi, se voglio usare spazi devo mettere il nome tra " "

man seguito dal comando mostra la help del comando (Es: `man ls`). Per cercare all'interno del manale uso `/paroladacercare` e premo invio. Mi muovo nelle pagine con barraspaziatrice, esco con `q`

Posso usare `ctrl+r` e scrivere l'inizio di un vecchio comando per recuperarlo

Capitolo 2

Comandi

2.1 Cartelle e file

`/` è la radice

`dev` è la cartella con i device

`ls` mostra elenco file nella directory. Varie opzioni, `lh` aggiunge dettagli, `a` per i file nascosti

`pwd` mostra il percorso assoluto alla cartella

`cd` permette di muoversi nelle cartelle

`mkdir` crea cartelle: l'opzione `-p` crea anche directory più esterne se fornisco una path

`rm` elimina un file/cartella(se forzato e ricorsivo): `-f` forza il comando, `-r` è ricorsivo

`mv` rinomina il file o lo sposta. Si chiama `mv` vecchionome nuovonome.
Esempio(`mv` prova ../../) sposta indietro di due cartelle

2.2 File

touch nomefile crea un file

vi è un editor di testo. I per iniziare a scrivere, scrivo, esc per uscire dalla scrittura, x per cancellare. Per salvare, entro nella linea di comando con :, w per salvare, q per uscire.

nano editor di testo più avanzato. Comandi con ctrl + lettera consigliata. Salva all'uscita

cat mostra il contenuto del file

less seguito dal nome del file da aprire, apre il file nella riga di comando. Col / faccio partire una ricerca

test verifica che il file esista e abbia certe proprietà che specifico nelle opzioni. Restituisce 1 se è corretto, 0 se no. *test -f nomefile* verifica che esista il file, -s permette di specificare una dimensione minima, -r permette di indicare se è leggibile

head restituisce le prime righe del file

tail restituisce le ultime righe del file

2.3 Permessi

Con il comando `ls -l` vedo i permessi del file o cartella. I due nomi a fianco alle lettere indicano in ordine il possessore del file, il secondo il gruppo a cui appartiene il file. I gruppi di lettere sono 3(user, gruppo, other). Le lettere rappresentano read, write e execute(attraversamento se è una cartella).

Lo user è il padrone del file, il gruppo è un insieme di utenti, other sono utenti fuori dal gruppo. I gruppi di 3 lettere sono modificabili come numeri binari. r è 4, w è 2, x è 1. Usando `chmod valorepermessiutentevalorepermessigruppovalorepermessiiother file`, modifico questi permessi. Esempio: `chmod 744 prova.txt`

Per cambiare il proprietario di file/cartella uso `chown nomeutente nomefile`

2.4 Utility

gzip comprime il file dato in input. Di default elimina il file originale, se voglio tenerlo devo usare -k

gunzip come sopra, ma decomprime

grep cerca una stringa in un file, restituisce le righe del file che la contengono

time prima di un comando restituisce il tempo di esecuzione

| **sort** ordina l'output di un altro comando

wc nomefile conta il numero di righe, parole e caratteri di un file

tar è un programma che crea archivi compressi. Si invoca con *tar cvzf nomeinoutput cartelladacomprimere*. Al posto di z posso usare j o J per aumentare il grado di compressione(aumenta anche il tempo richiesto per l'elaborazione)

which restituisce la path completa di un programma

pwd stampa la working directory

htop è il monitor di sistema

Capitolo 3

Programmazione

3.1 Comandi

!! esegue l'ultimo comando all'interno di un altro comando. Ecco un esempio:

```
sudo man !!
```

Posso salvare l'output di un comando su file con *comando > nomefile*
Se non c'è il file viene creato. Se uso » non sovrascrivo ma concateno il
l'output con il contenuto del file.

Ctrl + C ferma il programma in esecuzione

Ctrl + Z mette in pausa il processo in esecuzione

fg riprendere il programma messo in pausa