

# Riassunto comandi latex

Matteo Mistri

5 aprile 2018

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Comandi</b>	<b>3</b>
2.1	Cartelle e file . . . . .	3
2.2	File . . . . .	3
2.3	Utility . . . . .	4
<b>3</b>	<b>Programmazione</b>	<b>5</b>
3.1	Comandi . . . . .	5
3.2	Costrutti base . . . . .	5

# 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: cat ls).

# Capitolo 2

## Comandi

### 2.1 Cartelle e file

`/` è la radice

`dev` è la cartella con i device

`ls -lph` mostra elenco file nella directory. Se aggiungo `| less` mi viene impaginato l'output

`cd` permette di muoversi nelle cartelle

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

`rmdir` elimina cartelle: `-f` forza il comando, `-r` è ricorsivo

### 2.2 File

`touch nomefile` crea un file

**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 prime righe del file

## 2.3 Utility

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

**gunzip** come sopra, ma decompime

**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 nomeinput 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

# 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

### 3.2 Costrutti base

---

```
for condizione; do  
    corpo  
    corpo  
    corpo  
done
```

```
if condizione; then
```

```
    corpo  
fi
```

---

Esempio di piccolo programma bash:

---

```
for d in *  
do  
test -d $d && echo "$d"  
done
```

---