

# Code snippets

Pietro Gambazza

15 giugno 2024

## 1 Unix

### 1.1 Miscellanea

**uptime** dice da quanto tempo il sistema e' in funzione

**who** lista utenti collegati alla macchina

**chmod +xrw**

**chmod ugo-xrw** u-tente g-roup o-thers x-ecute r-read w-rite (tutto attaccato!)

**mv \*.??? /dir** sposta tutti i file che finiscono con un punto e tre caratteri

**ls cxx, h** mostra tutti i file che finiscono per cxx o h

**echo a..z** stampa tutte le lettere tra la a e la z

**echo 2..32** stampa tutti i numeri fra 2 e 32

**echo a..c1..3** stampa tutte le combinazioni

**cat file1 file2 ...** concatena i file e li stampa a terminale

**head** stampa prime 10 righe

**tail -2** stampa ultime 2 righe

**>** reindirizza output in un file

**»** reindirizza e sovrascrive

**env** stampa tutte le variabili definite nella shell corrente

| reindirizza l'output di un comando nell'input di un altro

**sort -nr -k 2 file.txt** riordina -n numerico -r reversed -k 2 seguendo la seconda colonna

**find . -name "file"** cerca . a partire dalla directory corrente -name per nome nelle sottodirectory

**locate "file"** cerca i file in un database aggiornato giornalmente

**ps -ef** stampa tutti i processi

**top**

**kill PID**

**touch file.txt** aggiorna la data di un file - se non esiste crea un file vuoto

**diff file1 file2** trova le differenze, se sono ASCII le stampa, se binari dice solo se diversi

**sleep n** freeza per n second

**time** restituisce tempo di esecuzione di un comando

**file** ti dice il tipo di file e altre info

**which** cerca un comando nella lista PATH e stampa l'indirizzo

**gzip/gunzip**

**gzip -v "file"** -v dice di quanto è stato compresso

**tar czf "out" "in"** c-reate z-ip f-nome fileout

**tar tzf** lista contenuto archivio

**tar xzf** espande l'archivio

**hexdump -C file** -C compara esadecimale e ASCII

**hexdump -C file | head -2** ottengo header del file (utile per capire il tipo)

**df -hT** mostra file system (cerca /dev e ext)

**mount/umount** per s-montare una partizione

**cat /proc/cpuinfo** info CPU

**cat /proc/meminfo** info RAM (prima riga)

**free -giga** RAM

**ls -l /proc/PID** mostra tutte le info di un programma in esecuzione con quel PID

**bc** calcolatrice

**tr ":" ", " file** sostituisce tutti i : con ,

**tr -d ":" file** rimuove i :

**uniq** rimuove linee doppie (di solito prima sort poi uniq)

**paste file1 file2** affianca due file riga per riga

**seq 1 4 21** genera sequenza fra 1 e 21 con passo 4

**basename -s .txt file.txt** rimuove suffisso -s da un file

**watch -n 5** mostra l'output di uno stesso comando ogni n secondi

**pdffont** mostra font di un pdf

**pdfimages** estrae pagine di un pdf

**pdfunite** unisce più pdf

**pdfseparate** separa pdf

**pdftjam "file" -o "out" -nup 2x1 -landscape** unisce due pagine in una

## 1.2 grep, awk, sed

**grep filter file** filtra solo le linee che contengono filter

**grep -iv filter file** -i case insensitive -v invert (non contengono filter)

**grep -n filter file** - n prende numero di linea

**column -t file** allinea output in colonne

**sed -e 's/testo/altrotesto/g'** -e esegui s sostituisci g per ogni occorrenza

**sed -n -e 7,42p file** stampa solo le righe tra 7 e 42

**sed -n -e '/inizio/,/fine/p' file** stampa solo le righe tra le stringhe "inizio" e "fine"

**sed -e /windows/d file** rimuovi tutte le righe contenenti la parola "windows"

**sed -e 's/^/ /' file** aggiungi spazi all'inizio di ogni linea (simbolo ^)

**awk '{print \$2, \$5}' file.txt** " delimitano inizio-fine script delimitano blocco istruzioni \$num seleziona colonna

**awk '!/word/ {print \$2, \$5, \$6/\$7}' file.txt** solo linee che non contengono "word" / divisione fra colonne

**awk '\$4 > 42 && \$2 < 7' file** stampa solo le righe con il valore della colonna 4 maggiore di 42 e il valore della colonna 2 minore di 72

**awk '/inizio/,/fine/' file** stampa solo le righe comprese tra la parola inizio e quella fine

### 1.3 Shell script

**A="assegno stringa"** assegnazione variabile, NO SPAZI

**'comando' \$(comando)** command substitution

**Esempio Backup tesi :**

```
FILE=tesi.'date +%y%m%d.%H%M' '.tgz
DIR=tesi
echo "ciao_ciccio_faccio_il_backup"
tar czf $FILE $DIR
ls -l $FILE
echo "fatto_capo"
```

**for f in \*.data; do mv \$f \$f.old; done**

**for i in {0..7}; do touch \$i.txt; done** genera la sequenza

**for i in `seq 0 7`; do touch \$i.txt; done** genera la sequenza

**while true; do date; sleep 3; done**

**if then else :** gli spazi servono!

```

if [ $X -gt 42 ]           # Se la variabile X e' maggiore di 42
then
                                # Fai una determinata operazione
elif [ $X -eq 42 ] # Se invece la variabile X e' uguale a 42
then
                                # Fai un'altra operazione
else
                                # Se nessuna delle precedenti
                                # Fai un altro tipo di operazione
fi
                                # Chiusura dell'if

```

**-gt, -eq, -ne, -lt** greater, equal, notequal, less

**Command line parameters :**

```

echo "Numero_di_parametri:_$#"
echo "Nome_dello_script:_$_0"
echo "Primo_parametro:_$_1"
echo "Tutti_i_parametri:_$_@"
for p in $_@; do echo "Par_=$_p"; done

```

## 1.4 Network

**ifconfig** visualizza IP, MAC wlp2s0(wifi) enp0s31f6(ethernet) lo(loopback)

**ip addr** come ifconfig

**curl ifconfig.me** stampa indirizzo IP WAN

**22** porta SSH

**80** porta HTTP

**25** porta email

**ping info.cern.ch** verifica se nodo attivo o meno (tempo di volo)

**tracpath www.infn.it** ricostruisce tragitto pacchetti

**telnet info.cern.ch 80** per connettersi a mano alle porte

**nmap -A -T4 192.168.1.1** scan delle porte aperte in un nodo

**nmap 192.168.1.\*** scan di tutta la rete (controlla chi è connesso)

**lsof** mostra tutti file aperti

**Script esplora rete locale :**

```
#!/bin/bash

BASE=192.135.12.          # Rete da esplorare
for i in 'seq 1 254'      # loop per i che va da 1 a 254
do
    IND=$BASE$i           # indirizzo, es: 192.168.1.7
    ping -c 1 -W 1 $IND > /dev/null # esegue ping
                                # buttando via l'output
    if [ $? -eq 0 ]        # se ping ritorna OK (0)
        then echo "$IND_usato" # scrive: 192.168.1.x usato
        else echo "$IND"    # oppure scrive: 192.168.1.x
    fi
done
```