

File system

Una volta collegati (login) alla shell, eseguiamo il comando per stampare la directory di lavoro ("print working directory"):

```
pwd
```

oppure

```
echo $PWD
```

Stampiamo anche la variabile HOME:

```
echo $HOME
```

Al momento, coincidono. Listiamo il contenuto della directory corrente:

```
ls
```

Per avere più dettagli, possiamo provare:

```
ls -lF
```

Creiamo una directory:

```
mkdir recas
```

quindi listiamo nuovamente, verificando di aver creato una directory:

```
ls -lF
```

Ora entriamo nella directory appena creata e listiamone il contenuto:

```
cd recas  
  
echo $PWD  
  
echo $HOME
```

Creiamo ora un file vuoto:

```
touch file1
```

```
ls -lF
```

Creiamo un file che contiene la stringa "ciao" e stampiamone il contenuto a video:

```
echo "ciao" > file2
```

```
cat file2
```

Analizziamo il contenuto del file e i suoi metadati:

```
hexdump file2
```

```
file file2
```

```
cat file2|xxd
```

Vogliamo ora creare uno script bash semplicissimo, ed eseguirlo:

```
date
```

```
echo "date" > esegui.sh
```

```
cat esegui.sh
```

```
./esegui.sh
```

Lo script non viene eseguito se non ha i permessi di esecuzione:

```
ls -lF esegui.sh
```

```
chmod u+x esegui.sh
```

```
ls -lF esegui.sh
```

```
./esegui.sh
```

Torniamo alla directory superiore:

```
cd ..
```

```
pwd
```

Listiamo ora la directory radice (root):

```
ls -lF /
```

Vogliamo cercare il comando date, che supponiamo in un file di nome "date" da qualche parte:

```
find /usr/ -name date
```

Vogliamo cercare tutti i file più grandi di 100M in /var:

```
find /var/  
  
find /var/ -size +100M  
  
find /var/ -size +100M 2> /dev/null  
  
ls -lF /var/lib/rpm/Packages  
  
ls -lFh /var/lib/rpm/Packages  
  
stat /var/lib/rpm/Packages
```

Vediamo le differenze tra atime, mtime e ctime di un file.

```
cd recas/  
  
touch file4  
  
stat file4  
  
cat file4  
  
stat file4  
  
echo "aaa" > file4  
  
stat file4  
  
chmod 700 file4  
  
stat file4  
  
cat file4  
  
stat file4  
  
history
```

Vogliamo cercare il comando "history", che supponiamo essere in un file di nome "history" da qualche parte (esercizio).

Creiamo un file che contiene l'history dei comandi usati finora:

```
history > h.txt
```

Ora vogliamo contare le linee del file h.txt:

```
cat h.txt | wc -l
```

Vogliamo tutte le righe che contengono la parola "find":

```
grep find h.txt
```

Vogliamo anche contarle (esercizio).

Vogliamo selezionare le righe che NON contengono la parola "find" e contarle (esercizio).

Vogliamo stampare l'inizio e la fine di un log di sistema (head e tail). Vogliamo "seguire" l'aggiornamento di un log di sistema (esercizio).

Esempi di uso di grep, awk e sed su /etc/passwd con anchors, pipes e OR:

```
echo "il mio nome e' _VAR1_" > test.txt  
  
echo "il mio cognome e' _VAR2_" >> test.txt  
  
cat test.txt  
  
cat test.txt | sed 's/_VAR1_/Vincenzo/g'  
  
cat test.txt | sed 's/_VAR1_/Vincenzo/g' | sed 's/_VAR2_/Spinoso/g'  
  
cat test.txt|awk '{print $1}'  
  
cat test.txt|awk '{print $2}'  
  
cat test.txt|awk '{print $3}'  
  
cat test.txt|awk '/cogn/{print $3}'  
  
cat /etc/passwd | egrep "marica|spinoso"  
  
cat /etc/passwd | awk -F ':' '{if(match($6,"var")){print $1 " "$6}}'
```

Esempio di uso di scp, copia da remoto a locale:

```
scp spinoso@resourcebroker.ba.infn.it:tsetup.1.4.3.tar.xz .
```

Decompressione di un archivio in formato XZ

```
tar xvJf tsetup.1.4.3.tar.xz
```

Compressione e rapporto di compressione:

```
tar cvzf recas.tar.gz recas/
```

```
du -hsc recas
```

```
du -hsc recas.tar.gz
```

Volumi e device, mount

```
mount -t <filesystem> /dev/<disk> <directory>
```

Esempio di uso di mount su un "loopback device"

```
wget https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-9.8.0-  
amd64-netinst.iso  
mount -o loop debian-9.8.0-amd64-netinst.iso recas  
sudo mount -o loop debian-9.8.0-amd64-netinst.iso recas  
mount -o loop debian-9.8.0-amd64-netinst.iso /root/blabla/  
mount  
cd /root/blabla/  
rm README.html
```