

# Esempi shell

30 aprile 2015

# Esempio 1

- Creare uno script (file comandi) che abbia la sintassi:

**./ps\_monitor.sh [N]**

- Lo script:
  - in caso di assenza dell'argomento, deve mostrare i processi di tutti gli utenti (compresi quelli senza terminale di controllo) con anche le informazioni sul nome utente e ora di inizio;
  - Se viene passato come argomento un intero (N) deve mostrare i primi N processi

Suggerimento: usare il comando **ps**

NB: non tutte le righe prodotte in output da ps hanno contenuto informativo rilevante

# Soluzione

```
#!/bin/bash
if [ $# -eq 0 ] # se non ci sono parametri
then
    ps aux
else
    ps aux | head -n `expr $1 + 1`
#consideriamo che c'e' anche una riga di intest.
fi
```

# Esempio 2

- Creare uno script che abbia la sintassi

**./lines\_counter.sh <directory> [up | down]**

- Lo script deve elencare i file contenuti nella directory con relativo numero di linee, ordinati in senso crescente (up) o decrescente(down)

NOTA: controllare:

- Che il primo argomento sia effettivamente una directory
- Che il secondo argomento sia la stringa up o down

# Soluzione

```
if [ -d $1 ] #vero se $1 è una directory
then
    > tmp  #creo e inizializzo file temporaneo
    cd $1

    for i in *
    do
        echo `cat $i|wc -l` $i >> ../tmp
    done
```

```
if [ $2 = "up" ]
    then
        cat ../tmp | sort -n
    elif [ $2 = "down" ]
    then
        cat ../tmp | sort -nr #l'ordinamento è inverso
    else
        echo "ERROR: up or down"
        exit 2 #uscita anomala
    fi
    cd ..
    rm tmp #elimino file temporaneo
else
    echo "$1 should be an existent directory"
    exit 2 #uscita anomala
fi
```

# Esempio 3

- Creare uno script che abbia la sintassi

**./backup.sh <nomefile> <nomebackup>**

- Se il file è una directory, lo script deve:
  - creare una sottodirectory(rispetto a livello corrente) di nome:  
`<nomefile>_<nomebackup>`
  - copiare ricorsivamente in essa il contenuto della directory
- Se il file è un file normale, lo script deve crearne 5 copie di nome  
`<nomefile>*i<nomebackup> i=1..5`

# Soluzione

```
#!/bin/bash

#IPOTESI: considero solo file e direttori nel dir.corrente

if [ $# -ne 2 ]
then
    echo "USAGE: backup.sh <filename> <backupstring>"
    exit 1
fi

if [ -d $1 ]
#-restituisce 1 se il primo parametro e' una directory
then
    cp -R $1 "$1_\$2"
```

```
elif [ -f $1 ] #controlla che $1 sia un file normale
then
    for i in 1 2 3 4 5 #i cicla sugli el. della lista
    do
        cp $1 "$1*$i$2"
    #i doppi apici inibiscono l'espansione di * ma non di $
    done
else
    echo "$1 should be a valid directory or file"
fi
```