

Esercizi Comandi Unix e Script Shell

Esercizio 1:

Realizzate una combinazione di comandi Unix per visualizzare il terzultimo processo presente nel sistema diverso da *pluto*, secondo il normale ordinamento prodotto dal comando che permette di visualizzare tutti i processi presenti nel sistema. (Assumete per semplicità che non esistano processi diversi da *pluto* il cui nome contenga come sottostringa *pluto*).

Ad esempio, se l'elenco dei processi presenti nel sistema è il seguente:

```
3958 pts/28 S 0:00 netscape
6854 pts/29 S 0:00 -csh
6873 pts/29 S 0:00 pluto
6878 pts/29 S 0:01 netscape
245 pts/30 S 0:00 -csh
17181 pts/31 S 0:00 pluto
3963 pts/28 S 0:01 pine
23878 pts/32 S 0:00 pluto
```

il comando richiesto deve restituire la quintultima riga, (cioè la terzultima escludendo quelle associate al processo *pluto*) ossia:

```
6878 pts/29 S 0:01 netscape
```

Esercizio 2:

Realizzare una combinazione di comandi unix che, dato un file di testo esistente *elenco*, crea un nuovo file *nominativo*, contenente la riga di *elenco* che viene lessicograficamente per seconda.

Ad esempio se il contenuto di un file *elenco* è il seguente:

```
valeria
aldo
roberta
bruno
sandro
paola
```

il comando deve creare il file *nominativo* con il seguente contenuto:

```
bruno
```

Esercizio 3:

Scrivere uno script shell di nome ***estrai*** che ha tre argomenti di input. I primi due sono file che esistono. Il terzo argomento è il nome di un file che deve essere creato e deve essere formato dalle prime 3 righe del primo file e dalle ultime 2 righe del secondo file. Non devono essere usati file temporanei.

Esercizio 4:

Scrivere uno script shell di nome **controlla** che ha un argomento di input *nomefile*. Lo script deve verificare ogni 10 secondi se il file *nomefile* esiste nella directory corrente, e notifica con un messaggio quando il file viene creato.

Esercizio 5:

Realizzate una combinazione di comandi Unix per visualizzare gli ultimi *n* utenti collegati.

Esercizio 6:

Scrivere uno script shell di nome **sposta** che ha due argomenti di input. Lo script *sposta* (su richiesta) tutti i file specificati come primo parametro nella directory specificata come secondo.

Esempio:

```
$sposta dir file1 file2 file3 .....
```

Soluzioni

Soluzione Esercizio 1:

```
ps agx | grep -v pluto | tail -3 | head -1
```

Soluzione Esercizio 2:

```
sort elenco | head -2 | tail -1 > nominativo
```

Soluzione Esercizio 3:

```
head -3 $1 > $3
```

```
tail -2 $2 >> $3
```

Soluzione Esercizio 4:

```
#!/bin/bash
```

```
until [ -f $1 ]; do sleep 10; done
```

```
echo "Il file $1 esiste"
```

Soluzione Esercizio 5:

```
who | sort -k 5 -r | head -3
```

Soluzione Esercizio 6:

```
#!/bin/bash
```

```
case $# in
```

```
0|1) echo Usage: sposta dir file1 file2 ... fileN ; exit 1;;
```

```
esac
```

```
if test -d $1
```

```
then
```

```
for i in $* # per tutti gli argomenti
```

```
do
```

```
    if test $i != $1
```

```
    then
```

```
        echo $i in $1 (y/n)?
```

```
        read risposta
```

```
        if test $risposta = "y"
```

```
        then
```

```
            if mv $i $1
```

```
            then echo si $1/$i
```

```
            else echo Error for $1
```

```
            fi
```

```
        fi
```

```
    fi
```

```
done
```

```
fi
```