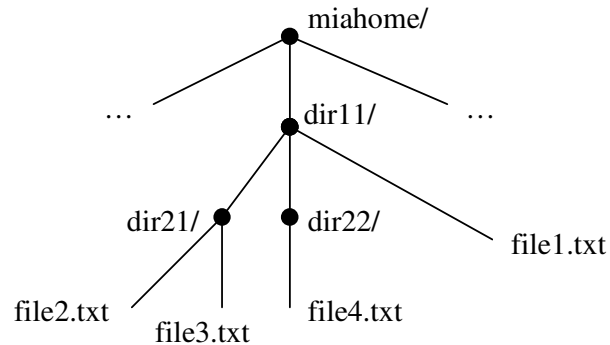


Laboratorio di Metodi Informazionali
Esercitazioni – soluzione di esercizi selezionati

Esercizi relativi a Introduzione a Linux e File System

1. Partendo dalla propria home (“miahome” nel disegno), attraverso opportuni comandi di shell, si creino le cartelle e i file necessari a riprodurre la situazione raffigurata di seguito:



Soluzione:

```
mkdir dir11
cd dir11
mkdir dir21
mkdir dir22
> file1.txt
cd dir21
>file2.txt
>file3.txt
cd ..
cd dir22
>file4.txt
cd ~
```

2. Si impostino come segue i permessi delle cartelle e dei file: si assegnino per la cartella dir11/ ed a tutto il suo contenuto diritti di lettura, scrittura ed esecuzione al proprietario, diritti di sola lettura agli utenti del gruppo e nessun diritto agli altri utenti. A questo punto, solo per il file file3.txt, si impostino invece solo diritti di lettura per il proprietario, nessun diritto per il gruppo e nessun diritto per gli altri utenti.

Soluzione:

```
chmod 740 dir11
cd dir11
chmod 740 *
cd dir21
chmod 740 *
cd ..
cd dir22
chmod 740 file4.txt
cd ..
cd dir21
chmod 400 file3.txt
```

3. Partendo dalla cartella dir11/, attraverso opportuni comandi di shell si sposti il file file4.txt nella directory dir11/ rinominandolo come file5.txt, si sposti la directory dir21/ con il suo contenuto nella directory dir22/ ed infine si cancelli la directory dir11/ con tutto il suo contenuto. Modificare i permessi di lettura/scrittura quando necessario.

Soluzione:

```
cd ~
cd dir11
mv ./dir22/file4.txt ./file5.txt
mv ./dir21 ./dir22
cd ..
chmod u+w ./dir11/dir22/dir21/file3.txt
rm -r dir11
```

4. Scrivere i comandi grep per cercare nel file "parole.txt" le parole con le seguenti caratteristiche:

- Parole che iniziano per "em" e finiscono con la lettera "n".
- Parole che iniziano per "em" o per "en" e finiscono con la lettera "n" o con la sequenza di lettere "ns".
- Parole che contengono al loro interno la sequenza di lettere "ron".
- Parole che contengono al loro interno la lettera "t" oppure la sequenza di lettere "tt".
- Parole che contengono la sequenza "En" seguita dalla lettera "c" o dalla lettera "v".

Soluzione:

```
a) grep "\<[eE][mM][a-zA-Z]*[nN]\>" parole.txt
b) grep "\< [eE] [(mM|nN)] [a-zA-Z]*[nN] [sS]*\>" parole.txt
c) grep "[^ ]ron[^ ]" parole.txt
d) grep "[^ ]t\{1,2\}[^ ]" parole.txt
e) grep "[eE][nN] [(vV|cC)]" parole.txt
```

Esercizi relativi a programmazione e script di shell

5. Creare uno script col nome di fattoriale per il calcolo del fattoriale di un numero passato come primo argomento. Si ricorda che il fattoriale di un numero naturale n, indicato con n!, è pari a $n! = 1*2* \dots *(n - 1)*n$. Nel caso particolare in cui n sia zero si ha $0! = 1$.

Soluzione:

```
echo Fattoriale
if [ $1 -eq 0 ]
then
    echo "$1! = 1"
else
    j=1
    i=0
    while [ $i -lt $1 ]
    do
        i=$(expr $i + 1)
        j=`expr $j \* $i`
        #echo $i
    done
    echo "$1! = $j"
fi
```

6. Allo script precedente si aggiungano dei controlli iniziali in modo di valutare che il parametro passato sia un numero e che sia maggiore o uguale a zero. Nel caso in cui il parametro non sia un numero o sia negativo, restituire un opportuno messaggio a video.

Soluzione:

```
echo "Fattoriale con check argomenti"
if [ $1 -ge 0 ]
then
    if [ $1 -eq 0 ]
    then
        echo "$1! = 1"
    else
        j=1
        i=0
        while [ $i -lt $1 ]
        do
            i=$(expr $i + 1)
            j=`expr $j \* $i`
            #echo $i
        done
        echo "$1! = $j"
    fi
else
    if [ $1 -lt 0 ]
    then
        echo "L'input deve essere un numero maggiore di zero"
    else
        echo "L'input deve essere un numero"
    fi
fi
```

7. Creare lo script **infofile** che restituisce per ognuna delle estensioni .txt, .wav, .tif, e .png, il numero di file con tale estensione, e il numero di cartelle trovate nella cartella da cui viene lanciato. Quindi, per esempio, se ci sono 10 file .txt, 2 file .wav e 3 cartelle, l'output sarà:

```
$ ./infofile
Nella cartella ... ci sono:
10 file .txt
2 file .wav
0 file .tif
0 file .png
3 cartelle
```

Soluzione:

```
for j in txt wav tif png
do
    n=`ls -l | grep -c "\.$j$"`
    echo "ci sono $n file di tipo $j"
done

n=`ls -l | grep -c "^d"`
echo "e ci sono $n sottocartelle"
```

8. Scrivere la procedura `rifai.sh` che esegue un comando dato come argomento ogni 5 secondi. Lanciare la procedura in *background*. Suggerimento: usare il comando `sleep`.

Soluzione:

```
echo $@
while [ 1 ]
do
    $@ &
    sleep 5
done
```

9. Scrivere una procedura che, utilizzando il `for`, effettui la copia dei file presenti nella cartella corrente nella cartella `./riserva`, escludendo, se sono presenti, i file `prova1` e `prova`

Soluzione:

```
rm -r riserva
mkdir riserva
x=`ls`
for i in $x
do
    if [ $i != prova -a $i != prova1 ]
    then
        cp $i ./riserva
    fi
done
```

10. Scrivere la procedura `biggerthan.sh` che esamina i file della cartella corrente e stampa il nome di quelli la cui dimensione in Kb è superiore ad un valore assegnato come argomento. Suggerimento: il comando `stat` ritorna informazioni su un file.

Soluzione:

```
x=`ls`

for i in $x
do
    if [ `stat -c %s $i` -gt $1 ]
    then
        echo $i
    fi
done
```

11. [variante di un es. precedente] Creare lo script `infofile` che restituisce **per ogni estensione di file trovata**, il numero di file con tale estensione, e il numero di cartelle trovate nella cartella da cui viene lanciato. Quindi, per esempio, se ci sono 10 file `.txt`, 2 file `.wav` e 3 cartelle, l'output sarà:

```
$/infofile
Nella cartella ... ci sono:
10 file .txt
2 file .wav
3 cartelle
```

Suggerimento: può essere utile usare i comandi `cut` e/o `uniq`

Soluzione:

```
list=`ls | grep "\.*$" |cut -d. -f2 |sort -f |uniq`

poi ciclo sulla lista
```

12. Si scriva uno script di shell il quale contenga al suo interno l'informazione sulla seguente lista ordinata di valori: 1,2,3,5,8,13,21. Acquisito un numero naturale intero positivo n dallo standard input, lo script visualizza sullo standard output la posizione nella lista ordinata del numero minore o uguale a n .
Per esempio, se $n = 6$ allora si avrà in uscita il valore 4 (suggerimento: inserire la lista come argomento di un ciclo FOR).

Soluzione:

```
listaord="1 2 3 4 8 13 21"
pos=0
for i in $listaord
do
    if [ $i -le $1 ]
    then
        pos=`expr $pos + 1`
    fi
done
echo "la posizione e\' ` $pos"
```

13. Si prepari uno script di shell il quale, acquisito un numero naturale intero positivo n dallo standard input, visualizza sullo standard output la somma della serie aritmetica limitata a n .
Per esempio, se $n = 6$ allora si avrà in uscita il valore $1 + 2 + 3 + 4 + 5 + 6 = 21$.

Soluzione:

```
i=1
somma=0
while [ $i -le $1 ]
do
    somma=`expr $somma + $i`
    i=`expr $i + 1`
done
echo "la somma della serie aritmetica limitata a $1 e\' ` $somma"
```

14. Si prepari uno script di shell il quale, acquisiti tre numeri naturali interi positivi n_1 ; n_2 ; n_3 dallo standard input, visualizza sullo standard output il minimo tra essi.
Per esempio, se $n_1 = 4$; $n_2 = 7$; $n_3 = 2$ allora si avrà in uscita il valore $\min(n_1; n_2; n_3) = \min(4; 7; 2) = 2$.

Soluzione:

```
min=$1
if [ $2 -le $min ]
then
    min=$2
fi
if [ $3 -le $min ]
then
    min=$3
fi
echo "il minimo tra $1, $2 e $3 e\' ` $min"
```