

## Soluzioni Esercitazione di Venerdì 23 Marzo 2018

- 1) (es26.sh) Creare uno script che accetti un solo argomento e che riporti sullo standard output se tale argomento rappresenta un nome assoluto, relativo, o relativo semplice. Verificarne il funzionamento per i tre casi.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat es26.sh
#!/bin/sh

if test $# -ne 1
then
    echo "usage: $0 arg"
    exit 1
fi

case $1 in
    /*) echo "Absolute filename" ;;
    /*/*) echo "Relative filename" ;;
    *) echo "Simple, relative filename" ;;
esac
#NOTA BENE: abbiamo controllato solo che tipo di nome sia e non che
sia il nome di un file!
exit 0
```

- 2) (es27.sh) Creare uno script che visualizzi tutti i file della directory corrente. Tutti i file con estensione .sh devono essere riportati su standard error invece che standard output. Invocare tale script in modo tale da verificare quali informazioni vengono riportate su standard output e quali su standard error.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat es27.sh
#!/bin/sh

for fname in *
do
    case $fname in
        *.sh) echo $fname 1>&2 ;;
        *) echo $fname ;;
    esac
done
#NOTA BENE: per evidenziare il fatto che il primo echo viene fatto
su standard error invocare lo script con la ridirezione o dello
standard output o dello standard error!
```

- 3) (es28.sh) Creare uno script che riporti su standard output tutti i file della directory corrente che contengono la stringa #!/bin/sh. Verificarne il funzionamento.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat es28.sh
#!/bin/sh

for fname in *
do
    grep '#!/bin/sh' $fname >/dev/null 2>&1
    if test $? -eq 0
    then
        echo "$fname"
    fi
done
```

done

- 4) (es29.sh) Creare uno script che controlli di essere invocato esattamente con 1 argomento e che tale argomento rappresenti un nome di directory esistente. Lo script deve scorrere l'intero contenuto della directory e riportare una F prima di ogni file regolare ed una D prima di ogni directory. Lo script deve anche riportare su standard output il numero totale di file e directory stampate. Verificarne il funzionamento.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat es29.sh
```

```
#!/bin/sh
```

```
case $# in
```

```
1) if test ! -d $1 -o ! -x $1  
    then
```

```
        echo $1 non directory esistente o non traversabile  
        exit 1
```

```
    fi;;
```

```
*) echo $# non corretto: ci vuole il nome di una directory  
   exit 2;;
```

```
esac
```

```
cd $1
```

```
F=0 #count for files
```

```
D=0 #count for directories
```

```
for fname in *
```

```
do
```

```
    if test -f $fname  
    then
```

```
        echo [F] $fname  
        F=`expr $F + 1`  
    fi
```

```
    if test -d $fname  
    then
```

```
        echo [D] $fname  
        D=`expr $D + 1`  
    fi
```

```
done
```

```
if test $F -eq 0 -a $D -eq 0
```

```
then
```

```
    echo "error: empty directory"  
    exit 3
```

```
fi
```

```
echo "Files =" $F
```

```
echo "Directories =" $D
```

```
exit 0
```

- 5) (for3.sh) Creare uno script che riporti su standard output tutti i parametri passati. Verificarne il funzionamento.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat for3.sh
```

```
#!/bin/sh
#file comandi con terzo esempio di for
echo si sta eseguendo per tutti i parametri: $*
for i
do
echo $i
done
```

- 6) (es25.sh) Creare uno script che riporti su standard output l'ultimo argomento della linea di comando e che riporti anche la lista degli argomenti escluso l'ultimo. Verificarne il funzionamento.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat es25.sh
#!/bin/sh
```

```
C=0      #count for arguments
LIST=    #list for the arguments apart of the last one
LAST=    #for the last argument
```

```
if test $# -eq 0
then
    echo NO ARGUMENTS
    exit 1
fi
```

```
for fname
do
    C=`expr $C + 1`
    if test $C -ne $#
    then
        LIST="$LIST $fname"
    else
        LAST=$fname
    fi
done
```

```
echo "LAST =" $LAST
echo "PREVIOUS ARGUMENTS =" $LIST
exit 0
```

- 7) (append.sh) Creare uno script che deve accettare uno o due argomenti, che devono essere nomi di file. Nel caso di un solo argomento, il contenuto dello standard input deve essere scritto in append sul file il cui nome è passato come unico parametro; nel caso di due argomenti, il contenuto del file il cui nome è passato come primo parametro deve essere scritto in append sul file il cui nome è passato come secondo parametro. Verificarne il funzionamento per i due casi.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat append.sh
#!/bin/sh
#file comandi append: invocazione append [dadove] sucosa
case $# in
1) cat >> $1;;
2) cat < $1 >> $2;;
*) echo "uso: append [dadove] sucosa";;
esac
```

- 8) (append1.sh) Stessa cosa dell'esercizio 7 ma con esattamente due parametri, introducendo anche i controlli su nomi relativi semplici, sulla esistenza e leggibilità del file il cui nome è passato come primo argomento, sulla esistenza e scrivibilità del file il cui nome è passato come secondo argomento, in caso di non esistenza del file il cui nome è passato come secondo argomento verificare che la directory corrente sia scrivibile. Verificarne il funzionamento nei vari casi.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat append1.sh
#!/bin/sh
#file comandi append: invocazione append [dovove] sucosa
#controllo sul numero di parametri
case $# in
2) echo Numero di par ok $#;; #caso giusto
*) echo uso: append dovove sucosa
    exit 1;;
esac
#controllo su nomi relativi
for i
do
    case $i in
    */*) echo $i non nome relativo semplice; exit 2 ;;
    *) echo OK nome $i ;;
    esac
done
#controllo su esistenza file $1 e leggibilita'
if test -f $1 -a -r $1
then
    #controllo esistenza file $2
    if test -f $2
    then
        #controllo se scrivibile
        if test -w $2
        then scrivibile=true
        else file $2 non scrivibile; exit 3
        fi
        else #il file non esiste e quindi dobbiamo verificare di
avere il diritto di scrittura sulla dir corrente settiamo scrivibile
a true
        if test -w .
        then scrivibile=true
        else echo directory corrente non scrivibile; exit 4
        fi
        fi
        if test $scrivibile = true
        then
            cat < $1 >> $2
        fi
    else
        echo $1 non esiste o non leggibile; exit 5
    fi
fi
```

- 9) (LeggiEMostra1.sh) Modificare lo script LeggiEMostra.sh in modo da poter funzionare per un numero di parametri maggiore o uguale a 1. Verificarne il funzionamento.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat LeggiEMostral.sh
```

```
#!/bin/sh
#file comandi leggi e mostra
#CONTROLO SU PRESENZA DI ALMENO UN PARAMETRO (uso di case)
case $# in
0) echo Usage is: $0 file1 [file2 file3 etc.]
    exit 1;;
*) ;;
esac

for i
do
    if test -f $i -a -r $i
    then
        echo -n "Vuoi visualizzare $i ? " > /dev/tty
        read risposta
        case $risposta in
        S* | s* | Y* | y*) ls -la $i; cat $i;;
        *) echo niente stampa di $i;;
        esac
    else echo file $i non presente o non leggibile
    fi
done
```

- 10) (TrovaFile.sh) Creare uno script che controlli di essere invocato esattamente con 1 argomento e che tale argomento rappresenti un nome relativo semplice. Lo script deve verificare se nella directory corrente esiste un file il cui nome è passato come argomento.

```
soELab@Lica02:~/file-comandi/terzaEsercitazione$ cat TrovaFile.sh
#!/bin/sh
#file comandi TrovaFile.sh
#ricerca in breadth-first
case $# in
1) echo OK un solo parametro $1 ;;
*) echo "Usage is: $0 file"
    exit 1;;
esac
#controllo che il primo parametro sia dato in forma relativa semplice
case $1 in
*/*) echo Errore: $1 non in forma relativa semplice
    exit 2;;
*) ;;
esac
if test -f $1
then
echo il file $1 si trova in `pwd`
else echo non esiste un file di nome $1 in `pwd`
fi
```