

Lab. di Sistemi Operativi

Esercitazioni proposte per le lezioni del 25 e 26/03/10

Utilizzando la shell di Linux e disponendosi in gruppi di due persone per ogni PC del laboratorio.

1. Assegnare il nome assoluto della directory corrente alla variabile DIRCORR
2. Utilizzare il comando `expr` per sommare i numeri 10 e 12
3. Utilizzare il comando `expr` per moltiplicare fra loro i numeri 10 e 12
4. Creare una variabile d'ambiente di nome "numero" ed assegnarle il valore 10. Sommare 12 a numero con il comando `expr`
5. Assegnare alla variabile numero il risultato di `expr`. Visualizzare l'exit value di `expr`.
6. Creare uno script (`es41.sh`) che accetti un numero di argomenti fra 1 e 3 compresi e che emetta un messaggio di errore su `stdout` in caso contrario
7. Modificare lo script precedente in modo che il messaggio di errore venga emesso su `stderr`
8. Come verificare che il messaggio viene realmente ridiretto su `stderr`?
9. Modificare nuovamente lo script `es41.sh` in modo che l'exit value del processo sia 0 in caso di successo o 1 in caso di errore
10. Creare uno script (`es42.sh`) che invochi lo script `es41.sh` con gli stessi argomenti con cui viene eseguito e che riporti su `stdout` l'exit value di `es42.sh`
11. Modificare `es42.sh` in modo che `es41.sh` possa essere invocato come comando (senza `"/"`)
12. Si analizzi la differenza fra l'aggiunta di `pwd` e l'aggiunta di `.` nel `PATH`
13. Dopo l'esecuzione di `es42.sh`, la variabile `PATH` della shell viene modificata?
14. A cosa serve `export` prima di `PATH=`?
15. Creare uno script (`es43.sh`) che visualizzi il nome dei file nella directory corrente, supponendo di non poter utilizzare il comando `ls`
16. Cosa avviene se invochiamo `es43.sh` in una directory vuota?
17. Modificare `es43.sh` in modo che visualizzi un messaggio appropriato se lo script viene invocato in una directory vuota
18. Credete sia possibile creare un file di nome `*?`? Se si, come si comporterebbe `es43.sh` e come potremmo migliorarlo?
19. Modificare `es43.sh` in modo che stampi una F prima di ogni file regolare ed una D prima di ogni directory
20. Modificare `es43.sh` in modo da contare il numero di file regolari presenti, riportando questo valore nell'exit value
21. Riprendiamo `es42.sh` e modifichiamolo in modo che venga invocato `es43.sh` senza argomenti. Fatto questo, `es43.sh` diventerà un comando e potrà essere invocato qualunque sia la directory corrente del processo.
22. Modifichiamo `es43.sh` in modo che, per ogni directory trovata venga invocata una nuova istanza di `es43.sh` (processo figlio) in quella directory

23. Cosa avviene se invociamo `es43.sh` senza utilizzare `es42.sh`?
24. Se cambiamo nome al file `es43.sh`, lo stesso non funziona più: Modificarlo di conseguenza
25. Utilizziamo l'exit value dei processi figli in modo da contare tutti i file regolari presenti anche nelle sottodirectory
26. Progettare uno script (`es44.sh`) che visualizzi tutti i file della directory corrente ad eccezione di quelli che terminano con `.sh`
27. Modificare lo script precedente in modo che i file che terminano con `.sh` ma iniziano con `es4` vengano visualizzati su `stderr`
28. Modificare lo script precedente in modo che visualizzi su `stderr` solo i nomi di file che iniziano con `es4`, terminano con `.sh` e non contengono la stringa `#!/bin/bash`
29. Completiamo lo script in modo che la ricerca di `#!/bin/bash` venga fatta solo sulla prima linea e che il carattere cancellato `#` sia all'inizio della linea
30. Creare uno script (`es45.sh`) che riporti su `stdout` l'ultimo argomento della linea di comando
31. Creare uno script (`es46.sh`) che accetti un solo argomento e che verifichi se tale argomento rappresenta un nome assoluto
32. Modificare lo script `es46.sh` in modo che accetti un solo argomento e che verifichi se tale argomento rappresenta un nome relativo
33. Modificare lo script `es46.sh` in modo che accetti un solo argomento e che verifichi se tale argomento rappresenta un nome relativo semplice

Soluzione

Esercitazioni proposte per le lezioni del 25 e 26/03/10

Utilizzando la shell di Linux e disponendosi in gruppi di due persone per ogni PC del laboratorio.

1. Assegnare il nome assoluto della directory corrente alla variabile DIRCORR

Soluzione:

```
$ DIRCORR='pwd'  
oppure  
$ DIRCORR=$(pwd)
```

2. Utilizzare il comando expr per sommare i numeri 10 e 12

Soluzione:

```
$ expr 10 + 12  
22
```

3. Utilizzare il comando expr per moltiplicare fra loro i numeri 10 e 12

Soluzione:

```
$ expr 10 * 12  
expr: syntax error  
$ expr 10 '*' 12  
120
```

4. Creare una variabile d'ambiente di nome "numero" ed assegnarle il valore 10. Sommare 12 a numero con il comando expr

Soluzione:

```
$ numero=10  
$ expr $numero + 12  
22
```

5. Assegnare alla variabile numero il risultato di expr. Visualizzare l'exit value di expr.

Soluzione:

```
$ numero=$(expr $numero + 12)  
$ echo $?  
0  
$ echo $numero  
22  
# Attenzione: l'exit value del processo che ha eseguito  
# expr vale 0, nulla a che fare con il risultato della  
# espressione che vale 22 e viene riversato su stdout
```

6. Creare uno script (es41.sh) che accetti un numero di argomenti fra 1 e 3 compresi e che emetta un messaggio di errore su stdout in caso contrario

Soluzione:

```
#!/bin/bash  
if test $# -lt 1 -o $# -gt 3  
then  
    echo Errore nel numero di argomenti  
fi
```

7. Modificare lo script precedente in modo che il messaggio di errore venga emesso su stderr

Soluzione:

```
#!/bin/bash
if test $# -lt 1 -o $# -gt 3
then
    echo Errore nel numero di argomenti >&2
fi
```

8. Come verificare che il messaggio viene realmente ridiretto su stderr?

Soluzione:

Invocare lo script indirigendo prima stdout poi stderr su /dev/null

9. Modificare nuovamente lo script es41.sh in modo che l'exit value del processo sia 0 in caso di successo o 1 in caso di errore

Soluzione:

```
#!/bin/bash
if test $# -lt 1 -o $# -gt 3
then
    echo Errore nel numero di argomenti >&2
    exit 1
fi
exit 0
```

10. Creare uno script (es42.sh) che invochi lo script es41.sh con gli stessi argomenti con cui viene eseguito e che riporti su stdout l'exit value di es42.sh

Soluzione:

```
#!/bin/bash
./es41.sh $*
echo $?
```

11. Modificare es42.sh in modo che es41.sh possa essere invocato come comando (senza "./")

Soluzione:

```
#!/bin/bash
export PATH='pwd':$PATH
es41.sh $*
echo $?
```

12. Si analizzi la differenza fra l'aggiunta di pwd e l'aggiunta di . nel PATH

Soluzione:

pwd viene espanso nel nome assoluto della directory corrente, quindi anche se si modifica la directory corrente dopo l'impostazione di PATH, sarà possibile ancora invocare i comandi presenti. se avessimo invece inserito la directory . allora la modifica alla directory corrente renderebbe irraggiungibili i comandi, ma diventerebbero raggiungibili quelli nella nuova directory corrente

13. Dopo l'esecuzione di es42.sh, la variabile PATH della shell viene modificata?

Soluzione:

No. La variabile PATH che subisce la modifica e' solo quella della sottoshell.

14. A cosa serve export prima di PATH=?

Soluzione:

La variabile di ambiente PATH viene inserita nell'elenco delle variabili da passare ai processi figli. Normalmente PATH viene marcata in questo modo gia' dagli script di configurazione della shell (.profile e derivati) ma ... non sempre!

15. Creare uno script (es43.sh) che visualizzi il nome dei file nella directory corrente, supponendo di non poter utilizzare il comando ls

Soluzione:

```
#!/bin/bash
for filename in *
do
    echo $filename
done
```

16. Cosa avviene se invochiamo es43.sh in una directory vuota?

Soluzione:

Viene visualizzato il carattere *, quindi la shell lascia il wild card * inalterato se non puo' espanderlo

17. Modificare es43.sh in modo che visualizzi un messaggio appropriato se lo script viene invocato in una directory vuota

Soluzione:

```
#!/bin/bash
for filename in *
do
    # test e [ sono equivalenti al 100%
    # se usate [, ricordate di mettere
    # la ] di chiusura al termine
    if [ $filename = "*" ]
    then
        echo Directory vuota
        exit 1
    fi
    echo $filename
done
exit 0
```

18. Credete sia possibile creare un file di nome *? Se si, come si comporterebbe es43.sh e come potremmo migliorarlo?

Soluzione:

```
#!/bin/bash
# Facoltativo
for filename in *
do
    if [ $filename = "*" ]
    then
        if [ ! -e $filename ]
        then
            echo Directory vuota
            exit 1
        fi
    fi
    echo $filename
done
exit 0
```

19. Modificare es43.sh in modo che stampi una F prima di ogni file regolare ed una D prima di ogni directory

Soluzione:

```
#!/bin/bash
for filename in *
do
    if [ $filename = "*" ]
    then
        echo Directory vuota
        exit 1
    fi
    if [ -f $filename ]
    then
        echo F $filename
    fi
    if [ -d $filename ]
    then
        echo D $filename
    fi
done
exit 0
```

20. Modificare es43.sh in modo da contare il numero di file regolari presenti, riportando questo valore nell'exit value

Soluzione:

```
#!/bin/bash
conta=0
for filename in *
do
    if [ $filename = "*" ]
    then
        # Dir vuota, 0 file
        exit 0
    fi
    if [ -f $filename ]
    then
```

```

        echo F $filename
        conta='expr $conta + 1'
    fi
    if [ -d $filename ]
    then
        echo D $filename
    fi
done
exit $conta

```

21. Riprendiamo es42.sh e modifichiamolo in modo che venga invocato es43.sh senza argomenti. Fatto questo, es43.sh diventerà un comando e potrà essere invocato qualunque sia la directory corrente del processo.

Soluzione:

```

#!/bin/bash
export PATH='pwd':$PATH
es43 .sh
echo $?

```

22. Modifichiamo es43.sh in modo che, per ogni directory trovata venga invocata una nuova istanza di es43.sh (processo figlio) in quella directory

Soluzione:

```

#!/bin/bash
conta=0
for filename in *
do
    if [ $filename = "*" ]
    then
        # Dir vuota, 0 file
        exit 0
    fi
    if [ -f $filename ]
    then
        echo F $filename
        conta='expr $conta + 1'
    fi
    if [ -d $filename ]
    then
        echo D $filename
        cd $filename
        es43 .sh
        cd ..
    fi
done
exit $conta

```

23. Cosa avviene se invochiamo es43.sh senza utilizzare es42.sh?

Soluzione:

L'invocazione delle sottoshell non funziona perché es43.sh non è un comando.

24. Se cambiamo nome al file es43.sh, lo stesso non funziona più: Modificarlo di conseguenza

Soluzione:

```
#!/bin/bash
conta=0
for filename in *
do
    if [ $filename = "*" ]
    then
        # Dir vuota, 0 file
        exit 0
    fi
    if [ -f $filename ]
    then
        echo F $filename
        conta='expr $conta + 1'
    fi
    if [ -d $filename ]
    then
        echo D $filename
        cd $filename
        $0
        cd ..
    fi
done
exit $conta
```

25. Utilizziamo l'exit value dei processi figli in modo da contare tutti i file regolari presenti anche nelle sottodirectory

Soluzione:

```
#!/bin/bash
conta=0
for filename in *
do
    if [ $filename = "*" ]
    then
        # Dir vuota, 0 file
        exit 0
    fi
    if [ -f $filename ]
    then
        echo F $filename
        conta='expr $conta + 1'
    fi
    if [ -d $filename ]
    then
        echo D $filename
        cd $filename
        $0
        conta='expr $conta + $?'
        cd ..
    fi
done
```

```
exit $conta
```

26. Progettare uno script (es44.sh) che visualizzi tutti i file della directory corrente ad eccezione di quelli che terminano con .sh

Soluzione:

```
#!/bin/bash
for filename in *
do
    case $filename in
        *.sh) ;;
        *) echo $filename
    esac
done
```

27. Modificare lo script precedente in modo che i file che terminano con .sh ma iniziano con es4 vengano visualizzati su stderr

Soluzione:

```
#!/bin/bash
for filename in *
do
    case $filename in
        es*.sh) echo $filename >&2;;
        *.sh) ;;
        *) echo $filename
    esac
done
```

28. Modificare lo script precedente in modo che visualizzi su stderr solo i nomi di file che iniziano con es4, terminano con .sh e non contengono la stringa #!/bin/bash

Soluzione:

```
#!/bin/bash
for filename in *
do
    case $filename in
        es*.sh) cat $filename | grep '#!/bin/bash'
                # dalla man page di grep: torna 0 se almeno
                # una linea trovata, 1 se nessuna linea,
                # 2 in caso di errore
                if [ $? -eq 1 ]
                then
                    echo $filename >&2
                fi
                ;;
        *.sh) ;;
        *) echo $filename
    esac
done
```

29. Completiamo lo script in modo che la ricerca di #!/bin/bash venga fatta solo sulla prima linea e che il carattere cancellato # sia all'inizio della linea

Soluzione:

```
#!/bin/bash
for filename in *
do
    case $filename in
        es*.sh) cat $filename | head -n 1 | grep '^#!/bin/bash'
                if [ $? -eq 1 ]
                then
                    echo $filename >&2
                fi
                ;;
        *.sh) ;;
        *) echo $filename
    esac
done
```

30. Creare uno script (es45.sh) che riporti su stdout l'ultimo argomento della linea di comando

Soluzione:

```
#!/bin/bash
# Soluzione rapida:
eval echo `echo \${$#}`
# Soluzione semplice:
if [ $# -gt 0 ]
then
    while [ $# -gt 1 ]
    do
        shift
    done
    echo $1
else
    echo $0
fi
```

31. Creare uno script (es46.sh) che accetti un solo argomento e che verifichi se tale argomento rappresenta un nome assoluto

Soluzione:

```
#!/bin/bash
case $# in
    1)
        ;;
    *)
        echo Numero argomenti errato
        exit 1
        ;;
esac
case "$1" in
    /*)
        ;;
    *)
        echo Nome non assoluto
        exit 1
esac
```

```
        ;;
esac
```

32. Modificare lo script es46.sh in modo che accetti un solo argomento e che verifichi se tale argomento rappresenta un nome relativo

Soluzione:

```
#!/bin/bash
case $# in
    1)
        ;;
    *)
        echo Numero argomenti errato
        exit 1
        ;;
esac
case "$1" in
    [^/]*
        ;;
    *)
        echo Nome non relativo
        exit 1
        ;;
esac
```

33. Modificare lo script es46.sh in modo che accetti un solo argomento e che verifichi se tale argomento rappresenta un nome relativo semplice

Soluzione:

```
#!/bin/bash
case $# in
    1)
        ;;
    *)
        echo Numero argomenti errato
        exit 1
        ;;
esac
case "$1" in
    [^/]*/*
        echo Nome relativo ma non semplice
        exit 1
        ;;
    [^/]*
        ;;
    *)
        echo Nome non relativo
        exit 1
        ;;
esac
```