

Lezione 5

1. *Creare una sottodirectory **bin** all'interno della propria home directory in cui mettere gli script. Fare in modo che gli script contenuti in **bin** possano essere invocati da qualunque directory con il nome del file, senza dover specificare l'intero pathname.*

Digitare

```
mkdir ~/bin
```

```
export PATH=$PATH:~/bin.
```

2. *Qual è l'effetto della seguente sequenza di comandi? Perché?*

```
> cat >chdir
cd ..
Ctrl-d
> chmod 700 chdir
> chdir
> pwd
```

Contrariamente a quanto ci si potrebbe aspettare la directory corrente dopo l'esecuzione dello script non è cambiata. Infatti ogni script viene eseguito in una sottoshell della shell corrente e gli effetti dei comandi come `cd` hanno effetto soltanto nella sottoshell.

3. *Creare un alias permanente **lo** per il comando **exit**.*
Aggiungere al file `.bash_profile` nella propria home una linea contenente `alias lo=exit`.
4. *Progettare uno script che prende come parametro una stringa e un file di testo e controlla se la stringa compare nel file.*

```
grep "$1" "$2" 2>/dev/null | wc -l
```

Lo script precedente stampa sullo standard output il numero di linee in cui compaiono delle occorrenze della stringa fornita come primo argomento nel file fornito come secondo argomento. Quindi se il numero stampato è 0 significa che la stringa non occorre nel file.

5. *Il comando **read** assegna alla variabile speciale **REPLY** un testo acquisito da standard input. Qual è l'effetto dello script **words** contenente i seguenti comandi?*

```
echo -n 'Enter some text: '
read one two restoffline
echo 'The first word was: $one'
echo 'The second word was: $two'
echo 'The rest of the line was: $restoffline'
exit 0
```

La linea `echo -n 'Enter some text: '` provoca la stampa su standard output della stringa `Enter some text:` senza newline finale. Il comando `read one two restoffline` legge dallo standard input una linea di testo memorizzando la prima parola nella variabile `one`, la seconda parola nella

variabile `two` ed il resto della linea nella variabile `restofline`. Contrariamente a quanto si potrebbe pensare, i comandi nelle tre linee successive non stampano il valore di queste variabili assieme ai rispettivi messaggi, in quanto l'utilizzo degli apici disabilita il metacarattere `$` e, di conseguenza, viene stampato testualmente quanto contenuto tra gli apici stessi. Lo script termina con un exit status pari a 0, grazie al comando `exit 0`.

6. *Qual è l'effetto della seguente sequenza di comandi? Perché?*

```
> cat >data
echo -n the date today is:
date
Ctrl-d
> chmod 700 data
> data
```

Viene stampato il messaggio `the date today is:` seguito immediatamente dalla data ed ora corrente dato che viene specificata l'opzione `-n` del comando `echo` che sopprime la stampa del newline alla fine del messaggio da visualizzare.

7. *Scrivere uno script che estragga soltanto i commenti dal file con estensione `java` fornito come primo argomento, sostituendo `//` con la stringa `linea di commento del file` <nome del file>:. Inoltre i commenti estratti devono essere salvati nel file fornito come secondo argomento.*

```
sed "s?/?linea di commento del file $1:?w $2" -n $1
```