The Bash Language

Esecuzione di uno script

- Diretta:
  - ./scriptname args
    - Il file scriptname deve essere e includere “#!/bin/bash” come prima riga
- Indiretta
  - source ./scriptname args
    - È la shell corrente a eseguire lo script

Assegnazione di variabili

nome_var=valore

Comando echo (per stampa su stdout)

echo [OPZIONI] [STRINGA]

- Opzioni:
  - -n: per non andare a capo.
  - -e: per interpretare i caratteri di escape.

Comando read (per la lettura da stdin)

read
  - con una o più variabili passate come argomento
  - uso della variabile $REPLY

Quoting

- Single quoting ''
  - le variabili non vengono espanse
- Double quoting “”
  - le variabili vengono espanse
- Es:
  - a=pippo
  - echo "$a pippo "'$a'" pluto"
  - pippo pippo $a pluto
  - echo $a pippo '$a' pluto
  - pippo pippo $a pluto

Utilizzo delle parentesi { } per delimitare il nome di una variabile

- Es.
  - nome=Gian
  - echo ${nome}marco
  - Gianmarco

Cattura del stdout di un comando

- $(<comando>)

Comando exit

- exit [numero]
  - termina l'esecuzione di un processo restituendo un valore al processo chiamante

- Es:
  - exit 0
  - restituisce un valore vero

Esecuzione di calcoli aritmetici

- un metodo a scelta dello studente
- Es.
  - let s=$n1+$n2
  - Assegna alla variabile $s la somma di $n1 e di $n2

Variabili di shell speciali

- $0, $1, $2, ...
  - passaggio di parametri sulla riga di comando
- $*
  - lista completa dei parametri, escluso il nome dello script
- $#
  - Numero di parametri
- $$
  - PID del processo
- $?
  - valore di ritorno dell'ultimo processo eseguito

Costrutto if-then-else (e elsif)

```
if condizione ; then
  statements
elif condizione
then
  statements
else
  statements
fi
```

Costrutto while (compresa ridirezione di stdin e stdout)

```
while condizione
do
  statements
done << $fileIn >> $fileOut
```

Formati richiesti nella condizione dei costrutti if e while

Solo le condizioni espresse tra simboli [...] sono richieste (non sono invece richieste le condizioni basate sulla parola chiave test)

- Confronti numerici:
  - -eq uguaglianza (==)
  - -ne non uguaglianza (!=)
  - -gt maggiore (>)
  - -ge maggiore o uguale (>=)
  - -lt minore (<)
  - -le minore o uguale (>)

- Confronti tra stringhe:

    - = uguaglianza
    - != non uguaglianza

- Condizioni su file:
    - -d <arg> vero se <arg> è un direttorio
    - -f <arg> vero se <arg> è un file
    - -r <arg> vero se <arg> ha il permesso di lettura
    - -w <arg> vero se <arg> ha il permesso di scrittura
    - -x <arg> vero se <arg> ha il permesso di esecuzione

- Operatori logici utilizzabili all'interno di una condizione:
    - ! not
    - -a and
    - -o or

- Operatori logici utilizzabili in un elenco di condizioni:
    - && and
    - || or

Costrutto for

```
for var in [ list ]
do
  statements
done
```

Istruzioni

- break
- continue

Vettori

```
# Dichiarazioni
array[3]="valore"
array=( 4 8 7 )
array=( [0]=4 [1]=8 [2]=7 [5]=10 )
# Accesso
echo ${array[1]}  # Accesso all'elemento 1 dell'array (valore 8)
echo ${array[*]}  # Stampa tutti gli elementi dell'array
echo ${#array[*]} # Numero di elementi contenuti nell'array
```