

# Lezione 4: Programmazione della Shell di Unix

Laboratorio di Elementi di Architettura e Sistemi Operativi

13 Marzo 2013

## **Avviso**

La lezione di martedì della prossima settimana (*19 Marzo*) non si terrà. La prossima lezione sarà *Mercoledì 20 Marzo*, nei soliti orari.

## Parte 1: Soluzione degli esercizi di ieri

### **Esercizio 7**

1. *fornire il numero di file (e directory) contenuti ricorsivamente nella propria home (si può utilizzare `ls -R?` ed il comando `find`?)*

L'idea di base per risolvere l'esercizio è quella di produrre una lista dei file e directory contenuti ricorsivamente nella propria home, contando le linee in output (supponendo che ogni linea corrisponda ad un file/directory).

```
$ find | wc -l
```

*Utilizzando `ls -R` non si ottiene il conteggio esatto!*

## Parte 2: Variabili e ambiente

### **Variabili**

- La shell permette di definire *variabili* per salvare dei valori
- Assegnazione: `variabile=valore`
- Per accedere al valore di una variabile, si usa l'operatore `$`
  - Esempio: se `x` vale 123, si può usarne il valore tramite `$x`
- Per visualizzare il valore di una variabile, si usa il comando `echo`
- Per acquisire un valore da standard input: `read variabile`
- *NOTA*: I valori delle variabili sono sempre STRINGHE
- L'output di un comando può essere assegnato ad una variabile con l'operatore `$( )`
- Per valutazioni aritmetiche si può usare l'operatore `$( ( ) )`, oppure il comando `let`

## Esempi

\$ x=0	Assegna il valore 0 alla variabile x
\$ echo \$x+1	Mostra la stringa 0+1
0+1	
\$ echo \$((x+1))	Mostra il risultato dell'operazione x+1
1	
\$ let "x+=1"	Incrementa di 1 il valore di x
\$ echo \$x	Stampa il valore di x
1	
\$ read x	Legge il valore di x dallo standard input
pippo	
\$ echo \$x	Stampa il valore di x sullo standard output
pippo	
\$ x=\$(ls)	Assegna l'output di ls come valore di x
\$ echo \$x	Stampa la lista dei file
[ bash cat chmod cp csh date ...	

## Le variabili d'ambiente

- Le variabili sono di norma locali alla shell
- Il comando `export` consente di passare i valori delle variabili ai processi creati dalla shell (in particolare alle sub-shell)
- L'*ambiente* della shell è una lista di coppie `nome=valore` trasmessa ad ogni processo creato
- Esempio: `export PS1='\h_mionome_\w>'`
- assegna un valore a una variabile di ambiente `export variabile[=valore]`
- stampa il valore di una o tutte le variabili d'ambiente `printenv [variabile]`
- stampa il valore di tutte le variabili d'ambiente `env`

## Variabili speciali

- Alcune variabili d'ambiente sono usate per memorizzare informazioni importanti:

PWD	percorso corrente
SHELL	nome della shell
PATH	percorsi dove cercare i comandi
PS1	prompt dei comandi
HOME	cartella home dell'utente
USER	nome dell'utente corrente
HOSTNAME	nome dell'host (computer a cui si è collegati)
HOSTTYPE	il tipo di architettura dell'host

## Parte 3: Script e programmazione della shell

## I file di comandi (script)

- È possibile memorizzare in un file una serie di comandi, eseguibili richiamando il file stesso
- Esecuzione:
  - tramite il comando `source script.sh argomenti`
    - \* i comandi vengono eseguiti all'interno della shell corrente
  - tramite il comando `bash script.sh argomenti`
  - eseguendo direttamente lo script: `./script.sh`
    - \* è necessario che il file abbia il *permesso di esecuzione*
  - Negli ultimi due casi viene lanciato un nuovo processo per il programma che deve interpretare lo script
- Per convenzione, la prima riga del file inizia con `#!/`, seguita dal nome dell'interprete entro cui eseguire i comandi (`#!/bin/bash`)

## Esempio

```
#!/bin/bash
date # restituisce la data
who # restituisce chi è connesso
```

Il carattere # indica che tutto quello che segue è un commento, e viene ignorato

## Variabili speciali

- La bash memorizza gli argomenti della linea di comando dentro una serie di variabili speciali:

**\$1 \$2 ... \$9**

- Altre variabili speciali:

`$$` PID del processo shell

`$0` Il nome dello script/processo corrente

`$#` il numero di argomenti

`$*` `$@` tutti gli argomenti

## Esempio

*numargomenti.sh*

```
#!/bin/bash
echo Questo script si chiama $0
echo "E' stato eseguito con" $# argomenti
echo Il primo argomento e' #1
echo "Il quarto argomento e' " #4
```

```
$ ./numargomenti.sh pippo pluto paperino
Questo script si chiama ./numargomenti.sh
E' stato eseguito con 3 argomenti
Il primo argomento e' pippo
Il quarto argomento e'
```

## Alcuni script speciali

- All'avvio, la shell cerca nella home dell'utente i seguenti *script di configurazione*, ed esegue il primo dei tre che trova:
  - `.bash_profile`
  - `.bash_login`
  - `.profile`
- Se la shell non è di tipo "login" (p.es, lanciata da un'altra shell) viene eseguito lo script `.bashrc`
- Se gli script non vengono trovati nella home, vengono usati gli script di sistema nella directory `/etc`
- È previsto anche un file `.bash_logout` che viene eseguito alla disconnessione