

Sistemi Operativi

Riassunto delle lezioni di Laboratorio A.A. 2018/2019

Matteo Franzil

7 marzo 2019

Indice

| | | |
|----------|---------------------|----------|
| 1 | Comandi Bash | 1 |
| 2 | Make | 2 |

1 Comandi Bash

La seguente è una lista, non esaustiva, dei comandi visti a lezione per l'uso della shell Bash.

- Comandi:
 - `free` → memoria libera
 - `pwd` → cartella corrente
 - `df` → partizioni
 - `dirname` → ottiene il nome della cartella di un dato file
- Piping:
 - `;` → esecuzione in sequenza
 - `&&`, `||` → esecuzione in sequenza con corto circuito
 - `|` → piping classico
 - `>` → redirect su file classico
 - `1>` → redirect di `stdout`
 - `2>` → redirect di `stderr`
- Scripting (intestazione):
 - `#!/bin/bash` → intestazione
 - `#Script "Hello World"` → intestazione nominativa
- Scripting:
 - `var=VALORE` → assegnazione (sono sempre stringhe)
 - `${var}` → stampa (con eventuale esecuzione)
 - `@` → equivalente ad `argv`
 - `#` → equivalente ad `argc`
 - `$1`, `$2`, `$3`, `$n` → i primi 9 parametri passati
 - `shift` → cestina il primo argomento nella lista
 - `" ... "` → crea un comando / stringa interpolando variabili
 - `$((...))` → contengono espressioni aritmetiche: se all'interno uso una `$var`, viene sostituita come fosse una macro, se uso il singolo contenuto di `var` allora viene inserito il valore come avesse le parentesi.

- `bc` → comando che supporta il piping in entrata, per eseguire operazioni in float
- `# Commenti` → commento classico di singola riga
- `$?` → valore di ritorno globale, usato dagli script (tipo `return 0` in C): ha significato booleano (`**0**` niente errori, arriva fino a `**256**`)

- Booleani:

- `test ...` → si aspetta un'espressione booleana, e internamente modifica il registro booleano visto prima:
- `-eq, -ne, -lt, -gt` → operandi booleani utilizzati
- `[...]` → sintassi di testing alternativa (gli spazi sono importanti!); attenzione che le parentesi quadrate sono considerate come ultimo comando eseguito
- `[[...]]` → raggruppamento di espressioni booleani per utilizzare operatori comuni (`>`, `<` ...)-
- `-f (file) -d (directory)` → verificano l'esistenza di un dato file/cartella.

- Cicli:

- `if [...]; then ... ; else ... ; fi` → costrutto if standard
- `case $var in; a|b) ... ;; c) ... ;; esac` → costrutto switch standard
- `for ((init ; case; step)); do; ...; done` → costrutto for standard
- `until [[...]] ; ...; done` → while negato
- `while [[...]] ; ...; done` → while standard

- Funzioni:

- `func() { ... }` → accessibile come fossero degli script (`func arg1 ... argn`)

- Varie:

- `$(...)` → sottoshell che esegue comandi in un processo separato
- `BASH_SOURCE[0]` → contiene il nome dello script in esecuzione
- `exit n` → uscita con codice d'errore
- `1> 2> ... n>` → redirectione dei diversi canali sui file, come visto prima (`1 = stdout`, `2 = stderr`); è possibile redirectione `stderr` su `stdout` e viceversa tramite il comando `2>&1`, oppure mettendo tutto su file: `1>output 2>&1`. L'ordine in cui vengono interpretati i redirect sono da destra a sinistra.

2 Make

Make è un tool utilizzato per automatizzare processi all'interno di sistemi Unix. Viene principalmente usato per automatizzare la compilazione dei file.

I Makefile sono composti da regole, composte da un **identificativo** (o nome) e da una **ricetta** (una serie di comandi indentati con una tabulazione `\t` rispetto al nome della ricetta):

regola:

```
echo "Ciao"
```

I file vengono poi eseguiti tramite il comando `make -f nome_file.makefile`. Notare come i comandi vengono anche stampati su `stdout` oltre a essere eseguiti. Questa funzionalità può essere sfruttata per stampare a video anche i procedimenti che vengono eseguiti dal file, ma può risultare fastidioso: si può quindi inserire una chiocciola `@` che impedisce la stampa del comando.

Il nome del file è opzionale e se non presente viene cercato all'interno della cartella corrente un nome corrispondente a `Makefile`.

In ogni ricetta, ogni riga è trattata singolarmente come un singolo processo. Bisogna quindi fare attenzione all'uso di `cd`.