

Università di Ferrara
Laurea Triennale in Informatica
A.A. 2021-2022
Sistemi Operativi e Laboratorio

**Lab-02. Introduzione alla programmazione
in Shell di Bourne**

Prof. Carlo Giannelli

[http://www.unife.it/scienze/informatica/insegnamenti/
sistemi-operativi-laboratorio](http://www.unife.it/scienze/informatica/insegnamenti/sistemi-operativi-laboratorio)

<http://docente.unife.it/carlo.giannelli>

<https://ds.unife.it/people/carlo.giannelli/>

Lezione 2

- Lab01: esercizi su comandi shell e piping
- **Lab02: esercizi su file di comandi (script)**

Eseguire uno script bash

- La prima riga **DEVE** sempre essere **`#!/bin/sh`** (chiamato “hashbang” o “shebang”)
 - o in alternativa **`#!/usr/bin/env sh`**
- È possibile commentare il codice con il carattere **`'#'`**
 - in sede d'esame la seconda riga dovrà essere:
`# NOME COGNOME NUM_MATRICOLA`
- Esistono 2 modi per lanciare lo script da terminale
 1. passare il file con lo script al comando sh:
`sh myScript <parametri>`
 2. rendere eseguibile il file
`chmod +x myScript` o **`chmod 0755 myScript`**
e poi lanciare lo script
`./myScript <parametri>`

Debug script bash

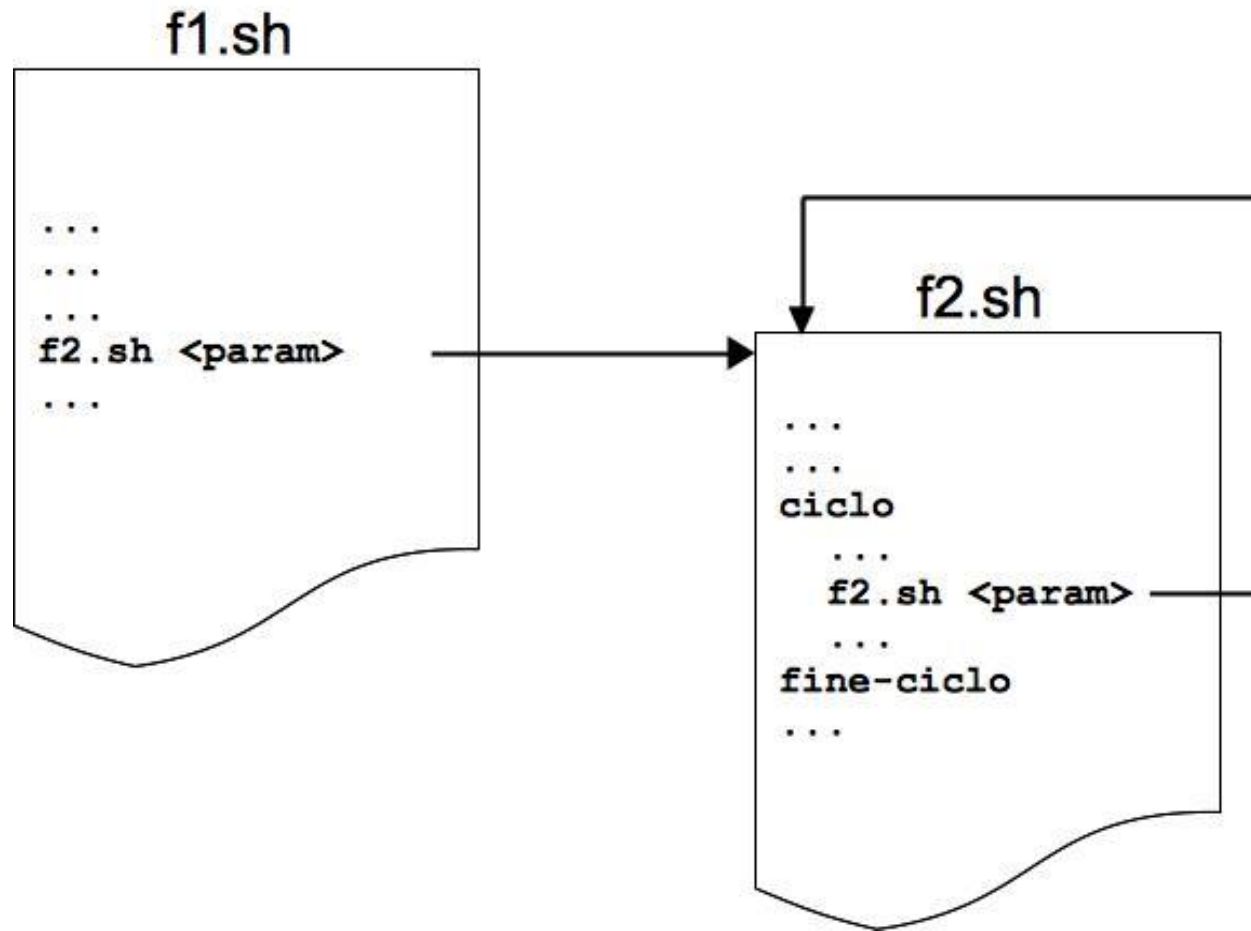
- Esistono 2 modi per fare il DEBUG: lo script da terminale
 - Passare il file con lo script al comando sh, opzione x:
`sh -x myScript <parametri>`
 - Inserendo i comandi `set [-x|+x]` nel file sorgente
`#!/usr/bin/env sh`
`...`
`set -x #Attivo tracing`
`<righe di codice da debuggare>`
`set +x #Disattivo tracing`

-x Print a trace of simple commands, for commands, case commands, select commands, and arithmetic for commands and their arguments or associated word lists after they are expanded and before they are executed.

Parametri

- Di solito, all'esame verrà richiesto di scrivere un file comandi che accetti dei parametri dall'utente. Tali parametri dovranno essere un certo numero (anche N generico) e verranno passati come argomenti quando lo script viene lanciato.
- Esempi:
 - nomi (assoluti o relativi) di directory
 - nomi (assoluti o relativi) di file
 - stringhe da cercare
 - ...
- È importante effettuare il controllo dei parametri all'inizio del file.

Script modulari 1/2



- Tipico caso d'esame:
 - *file comandi che chiama un altro file comandi*
 - *file comandi che chiama se stesso ricorsivamente*
- Perché questa struttura?

Script modulari 2/2

- Tipicamente i file `f1.sh` e `f2.sh` risiederanno in una sottodirectory all'interno della directory **\$HOME**
 - Esempio: `~/Desktop`
- Di solito, tale directory non sarà inclusa nel percorso di sistema **PATH**.
 - Se invece lo fosse, lo script potrebbe essere lanciato digitandone semplicemente il suo nome, senza spostarsi nella directory che lo contiene e senza farlo precedere da `./`
- Quindi il primo file dovrà essere invocato esplicitamente dall'utente:

```
cd script_dir  
./f1.sh
```
- **Ma come facciamo con il secondo file?**

Aggiornamento del path 1/2

- Prima della chiamata al secondo file, aggiungere alla variabile `$PATH` la directory in cui esso risiede:

```
# aggiunge directory script corrente al PATH
PATH=$PATH: `pwd`
# esporta il PATH
export PATH
```


Aggiornamento del path 2/2

- Soluzione alternativa: assumo di sapere che il file f2.sh risieda in una directory nota (`my_script_dir`):

```
# memorizza directory in una variabile
SCRIPT_DIR=/home/utente/my_script_dir
# aggiunge directory script corrente al PATH
PATH=$PATH:$SCRIPT_DIR
# esporta il PATH
export PATH
```

Esercizio 1

Si scriva il file comandi "verificafile" che chieda all'utente di **digitare una stringa a terminale** e verifichi se essa **corrisponda al nome di un file o di un direttorio** (specificando di quale dei due si tratti) nel direttorio locale.

Al termine dell'operazione, il file comandi deve mettersi in attesa che l'utente inserisca un'altra stringa e terminare nel caso in cui l'utente inserisca la stringa "fine".

Esempio:

```
$ sh verificafile.sh
Scrivi il nome del file:
aaaa
aaaa inesistente
Scrivi il nome del file:
dir1
dir1 è un direttorio
Scrivi il nome del file:
fine
```

Esercizio 1 - traccia

`verificafile.sh`

- Il file “verificafile.sh” non prevede parametri al lancio
- Una volta lanciato, lo script deve mettersi in attesa di un input: nome di un file/directory all'interno del direttorio corrente
- Controllare l'input fornito:
 - se corrisponde alla stringa “fine” terminare il programma
 - altrimenti verificare se corrisponde a un file o a una directory e stampare il risultato a video

Esercizio 2

Si scriva il file comandi "contaricorrenze" che conti (in un file il cui nome è passato come primo parametro) in quante righe ricorre (almeno un volta) ognuna delle parole passate come successivi parametri

Esempio:

```
$ sh contaricorrenze.sh processi sbin var
Apertura file processi...
Parametro: sbin - Ricorrenze: 26
Parametro: var - Ricorrenze: 4
```

Esercizio 2 - traccia

contaricorrenze.sh file stringa1 ...

- Il lancio del file “contaricorrenze.sh” richiede due o più parametri:
 - controllare il numero dei parametri passati
 - controllare se il primo parametro è un file
- Contare per ogni stringa passata il numero di righe nel file che contengono tale stringa

Esercizio 3

Si implementi un file comandi "trova" che effettui la ricerca di un file in modo **ricorsivo** a partire da una directory assoluta specificata dall'utente.

Esempio:

```
$ sh trova.sh nomedirectory nomefile
```

Esercizio 3 - traccia

trova.sh dir nomefile

- Controllo dei parametri:
 - controllare il numero dei parametri passati (numero e tipo)
 - dir deve essere un nome di directory assoluto
- Per ogni file all'interno della directory:
 - se è un file: verificare l'uguaglianza con il nome cercato e stampare a video il percorso assoluto del file
 - se è una directory: lanciare la ricerca all'interno della directory

Esercizio 4

Si scriva un file comandi in Shell di Bourne che abbia l'interfaccia:

esame estensione dir1 dir2

dove "estensione" è una qualsiasi stringa che inizia con un punto e "dir1"/"dir2" sono nomi assoluti di direttorio. Si svolgano gli opportuni controlli sui parametri di invocazione del file comandi.

Il file comandi deve cercare tutti i file con l'estensione passata come parametro in tutto il sottoalbero individuato da dir1 e li deve copiare in dir2 se e solo se la copia **NON** va a sovrascrivere file già esistenti con lo stesso nome in quel direttorio.

Alla fine dell'esecuzione si visualizzi il numero di file complessivamente copiati.

Esercizio 4 - traccia

esame.sh

- Controllo degli argomenti:
 - controllare i parametri passati (numero e tipo)
 - aggiunta della directory script corrente alla variabile \$PATH
- Chiamata a **esame_rec.sh <parametri>**

esame_rec.sh

- Per ogni file all'interno della directory con l'estensione scelta copiarlo nella directory di destinazione (se non esiste).
- Su ogni directory (su cui ho i diritti di esecuzione) presente nella directory corrente chiamare **esame_rec.sh <parametri>**