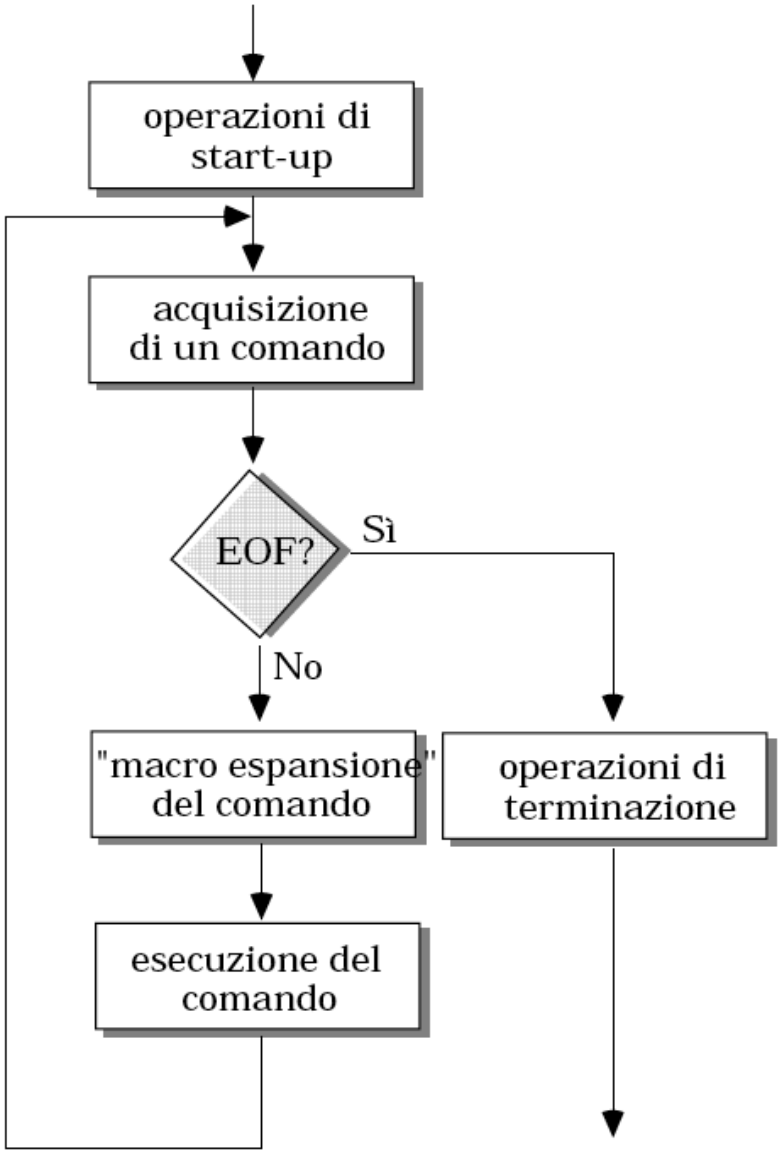
Lab. di Sistemi Operativi
- Esercitazione n° 1 -
a.a. 2015/2016

"Comandi Shell"

Ciclo Esecuzione Shell



Comandi

- Comandi shell:
  - mkdir
  - touch
  - echo
  - cp
  - echo
  - cat
  - cut
  - head
  - tail
  - sort

Esercizio n° 0

- 0a) Creare una cartella **EsercitazioneLSO-1** nella directory di lavoro

- 0b) Creare un file vuoto chiamato **provaFile.txt**
- 0c) Creare una copia del file chiamata **provaFile2.txt**
- 0d) Creare una cartella chiamata **esercizio-0**
- 0e) Spostare i due file nella cartella **esercizio-0**
- 0f) Visualizzare il contenuto della nuova cartella
- 0g) Cancellare i due file e la directory

- echo, variabili, ridirezione -

Comando echo

echo lista_argomenti

Mostra su stdout gli argomenti in ordine separati da spazi

Esempio:

echo a b c

Variabili di Shell

La shell gestisce variabili predefinite (di ambiente) che ne caratterizzano il comportamento:

SHELL	shell corrente
HOSTNAME	nome delle macchina
USER	nome dell’utente
PATH	percorsi di ricerca eseguibili
HOME	home directory

Variabili

Altre variabili:

- Scrittura/definizione: *<variabile>=<stringa>*
- Lettura: *$<variabile>* oppure *${<variabile>}*

Esempio:

```
> a=3
> echo $a
> 3

> echo $SHELL
> echo $HOME
> echo $USER
```

```
> b="a=$a"
> b='a=$a'
> a=${a}a
> a=${a}${a}
```

Command substitution

L’output del comando sostituito al comando:

$(command)

Esempio:

> list=$(ls)
> Echo $list

Canali di Comunicazione

I programmi dispongono di 3 canali di comunicazione:

- Standard input (codice 0), per input
- Standard output (1), per output
- Standard error (2), per errore

Standard input = tastiera
Standard output= schermo

Redirezione std I/O

La Shell può ridirezionare queste associazioni:

comando argomenti > **file**
comando argomenti >> **file**
comando argomenti < **file**
comando argomenti 2>

Dove:

> e >> per redirezione di stdout, < per redirezione stdin,
2> per redirezionare stderr

Esempi:

ls -a > listaFile.txt
echo aaa >> listaFile.txt

Concatenazione

Concatena i file e li mostra in stdout

cat [opzioni] [file]

Esempio:

cat listaFile.txt

cat listaFile.txt listaFile.txt

Esercizio n° 1

- 1a) Creare un file testo chiamato **provaFile.txt** di almeno 5 righe utilizzando touch, echo, cat e ridirezione

- 2b) Creare un file testo chiamato **provaFile2.txt** che ha il contenuto di **provaFile.txt** ripetuto 4 volte

Esercizio n° 1

- 0c) visualizzare il contenuto delle variabili di ambiente PATH, USER, SHELL, HOME
- 0d) creare un file testo chiamato **provaVar.txt** che contiene questi dati
- 0e) creare una variabile useruser in cui valore sia il nome dello USER ripetuto due volte
- Creare un file che si chiami come l'utente corrente
- Creare un file che si chiami come l'host corrente, e che contenga il nome dell'host corrente

Pipe (tubo)

```
comando1 | comando2
```

Pipeline di due o più comandi:
Lo standard output di com1 funge da input a com2...

- ***com1 [arg ..] | com2 [arg ..]..|..***

Esempi di comandi concatenabili: cat, sort, wc

> cat file | wc

- head & tail -

head & tail

Comando/Sintassi	Descrizione
head [-*numero*] file	visualizza le prime 10 (o -*numero*) linee di un file
tail [-*numero*] file	visualizza le ultime 10 (o -*numero*) linee di un file

Esempio d'uso head:

head -40 filename

oppure

head -n 40 filename

Esempio d'uso tail:

tail -30 filename

Esercizio n° 2

- Scrivere una combinazione di comandi Unix che consenta di visualizzare:
  1. la **terza e la quarta** riga del file **provaFile1.txt**
  2. le **penultime** 3 righe del file **provaFile1.txt**
  3. l' **n-esima** riga del file **provaFile1.txt**

Soluzione 1

```
head –4 provaFile1.txt | tail -2
```

Soluzione 2

```
tail –4 provaFile1.txt | head -3
```

Soluzione 3

```
head -n provaFile1.txt | tail  -1
```

Esercizio n° 2

- Definire una sequenza di comandi che assegna ad una variabile primofile il nome del primo file che compare nel listing di ls.
- Definire una sequenza di comandi che assegna ad una variabile ultimofile il nome dell'ultimo file che compare nel listing di ls.

Soluzione 4

```
primofile=$(ls -1 | head -1)
```

Soluzione 5

```
ultimofile=$(ls -1 | tail -1)
```

- /etc/passwd -

/etc/passwd

- Il file /etc/passwd è il database degli utenti su ogni sistema Unix.
- Ad ogni user è dedicata una riga che definisce quali sono i suoi principali attributi:

riga file passwd:

Username:Password:UserID:GroupID:Info:HomeDirectory:Shell

Esempio:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
[...]
diego:x:501:503:/home/diego:/bin/bash
```

/etc/passwd

- **Username:** Nome dell' user, la login con cui può accedere al sistema;
- **Password:** Campo riservato alla password dell'utente. Può essere scritta direttamente in forma criptata o esserci semplicemente una x (la password c'è ma è scritta altrove, di solito in /etc/shadow). Se c'è un * (asterisco) significa che l'utente o non ha una password o la password non è valida (in questo caso non gli è permesso di login);
- **UserID:** ID dell' user;
- **GroupID:** ID del gruppo di appartenenza;
- **Info:** Contiene informazioni sull'utente non necessarie al sistema (nome esteso, numero di telefono, mail ecc...);
- **HomeDirectory:** Indica la directory della home dell'utente;
- **Shell:** Indica la shell di default per quell'utente.

- comando cut -

Esercizio n° 3

- Con un opportuno comando Unix visualizzare:
  1. il contenuto del file **/etc/passwd**
  2. estrarre il primo campo dal file **/etc/passwd**

Soluzione 1

cat /etc/passwd

Soluzione 2

-f seguito dal numero del campo estrarre il campo indicato

cut -d: -f1 /etc/passwd

il separatore –d (delimiter) seguito dal simbolo del separatore nel file

- comando di ordinamento: sort -

Esercizio n° 4

- Realizzare una combinazione di comandi unix per visualizzare l'ultimo file in ordine alfabetico presente nella directory di lavoro

Soluzione

ls | sort –r | head -1

Esercizio n° 5

- Ordinare le righe del file di testo **votoStudenti.txt** (anche in senso inverso):

```
Gianni  20
Bruno   15
Carlo   10
Alice   30
```

Soluzione: ordinamento alfabetico default

```
sort votoStudenti.txt
```

```
Alice   30
Bruno   15
Carlo   10
Gianni  20
```

Soluzione Esercizio n° 5

Soluzione: ordinamento inverso

sort -r **votoStudenti.txt**

Gianni 20
Carlo 10
Bruno 15
Alice 30

inverte il senso di ordinamento

Soluzione: ordinamento inverso su file

sort -r **votoStudenti.txt** -o file_risultato.txt

ordinamento inverso con scrittura del risultato nel file _risultato.txt invece che sullo standard output

Esercizio n° 6

- Realizzare una combinazione di comandi unix che, dato un file di testo esistente **elenco.txt**, crea un nuovo file **nominativo.txt**, contenente la riga di **elenco.txt** che viene lessicograficamente per seconda.

Ad esempio se il contenuto di un file **elenco.txt** è il seguente:

valeria
aldo
roberta
bruno
sandro
paola

il comando deve creare il file **nominativo.txt** con il seguente contenuto:
bruno

Soluzione Esercizio n° 6

Soluzione:

```
sort elenco.txt | head -2 | tail -1 > nominativo.txt
```

ordina il contenuto del file **elenco** in ordine alfabetico

scrive il risultato sul file **nominativo**