

Lab. di Sistemi Operativi
- Esercitazione n° 1 -
a.a. 2019/2020

“Comandi Shell”

Sommario

④ Comandi shell:

④ ls, cd, pwd

④ echo

④ touch

④ cat

④ head

④ tail

④ cut

④ sort

Esercizio n° 0

- ④ 0a) Creare una cartella **EsercitazioneLSO-1** nella directory di lavoro
- ④ 0b) Creare un file testo chiamato **provaFile.txt** di almeno 5 righe utilizzando touch, echo, cat e ridirezione
- ④ 0c) Creare un file testo chiamato **provaFile2.txt** che ha il contenuto di **provaFile.txt** ripetuto 4 volte

Esercizio n° 0

- 0d) visualizzare il contenuto delle variabili di ambiente PATH, USER, SHELL, HOME
- 0e) creare un file testo chiamato **provaVar.txt** che contiene questi dati
- 0f) creare una variabile `useruser` in cui valore sia il nome dello USER ripetuto due volte

- head & tail -

head & tail

Comando/Sintassi	Descrizione
<code>head [-numero] file</code>	visualizza le prime 10 (o <code>-numero</code>) linee di un file
<code>tail [-numero] file</code>	visualizza le ultime 10 (o <code>-numero</code>) 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° 1

- ④ 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
```

- /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 -

cut

```
cut [options] [file...]
```

estrae delle colonne specifiche dalle linee di testo che compongono *file*.

Alcune opzioni:

- c *char_list*** definisce gli intervalli da estrarre espressi in caratteri.
- f *field_list*** definisce gli intervalli da estrarre espressi in campi. I campi sono distinti in base a un certo carattere usato come delimitatore. Quello predefinito è il carattere di tabulazione.
- d *delimiter*** definisce un delimitatore alternativo al carattere di tabulazione.

Esempi d'uso

Estrae la prima e la quinta colonna del file /etc/passwd

```
cut -d: -f1,5 /etc/passwd
```

Estrae i primi dieci caratteri da ogni riga del file /etc/passwd

```
cut -c1-10 /etc/passwd
```

Esercizio n° 2

- Ⓢ 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 -

Sort

```
sort [options] [file...]
```

permette di (ri)ordinare o fondere insieme il contenuto dei file passati come parametri, oppure di (ri)ordinare le linee passategli in input.

In assenza di opzioni che definiscano diversi criteri di ordinamento, quest'ultimo avviene in base al primo campo ed è alfabetico.

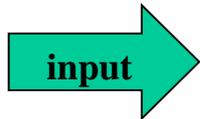
Alcune opzioni:

- f** ignora le differenze tra lettere minuscole e maiuscole
- n** considera numerica anzichè testuale la chiave di ordinamento
- r** ordina in senso decrescente anzichè crescente
- o *fileout*** invia l'output a fileout anzichè sull'output standard
- t *s*** usa *s* come separatore di campo
- k *s1,s2*** usa i campi da *s1* a *s2-1* come chiavi di ordinamento

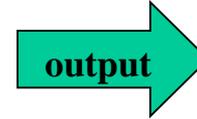
Esercizio n° 3

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

lab.txt
lab1.txt
lab2.txt
lso.txt
pluto.txt
prova.txt



Sequenza di
Comandi Unix



Risultato da ottenere

prova.txt

Soluzione

```
ls | sort -r | head -1
```

Esercizio n° 4

- Ⓢ 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° 4

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
```

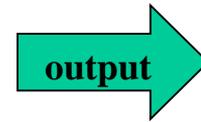
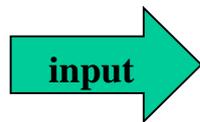
ordinamento inverso con scrittura del risultato nel file _risultato.txt invece che sullo standard output

Esercizio n° 5

- 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



bruno

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

Soluzione Esercizio n° 5

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**

aldo
bruno
paola
roberto
sandro
valeria



- comando sort: con chiavi di ordinamento -

- sort: chiavi di ordinamento -

- Ⓢ Tramite l'opzione -k è possibile definire una chiave di ordinamento su una porzione della linea contenuta nel file
- Ⓢ Specificando più opzioni -k è possibile definire più chiavi di ordinamento da usare in cascata.

Sintassi sort con opzione -k

```
sort -k campo_inizio [tipo] [,campo_fine [tipo]]
```

campo_inizio, campo_fine

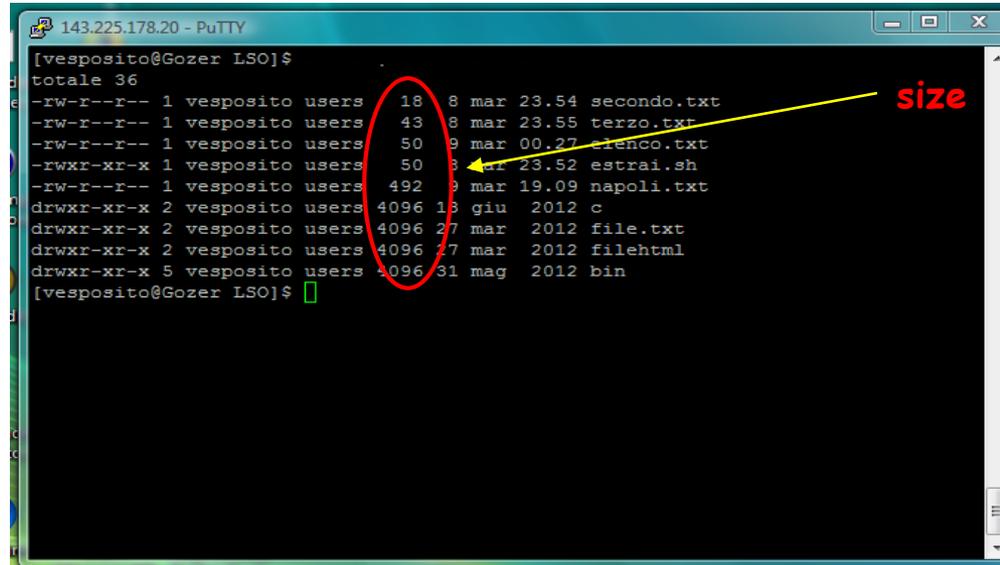
chiavi per restringere l'ordinamento su una porzione di linea

tipo

è il tipo di ordinamento applicato sui campi specificati

Esercizio n° 6

- Realizzare una combinazione di comandi Unix per visualizzare i file presenti nella directory di lavoro con un ordine di dimensione crescente.



```
143.225.178.20 - PuTTY
[vesposito@Gozer LSO]$
totale 36
-rw-r--r-- 1 vesposito users 18  8 mar 23.54 secondo.txt
-rw-r--r-- 1 vesposito users 43  8 mar 23.55 terzo.txt
-rw-r--r-- 1 vesposito users 50  9 mar 00.27 elenco.txt
-rwxr-xr-x 1 vesposito users 50  8 mar 23.52 estrai.sh
-rw-r--r-- 1 vesposito users 492  9 mar 19.09 napoli.txt
drwxr-xr-x 2 vesposito users 4096 13 giu  2012 c
drwxr-xr-x 2 vesposito users 4096 27 mar  2012 file.txt
drwxr-xr-x 2 vesposito users 4096 27 mar  2012 filehtml
drwxr-xr-x 5 vesposito users 4096 31 mag  2012 bin
[vesposito@Gozer LSO]$
```

Soluzione: size crescente

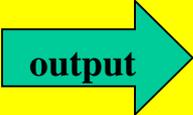
ls -l | sort -k5 -n

Esercizio n° 7

- Applicare al file `votoStudenti.txt` un ordinamento numerico con chiave:

Gianni	20	← Secondo campo numerico
Bruno	15	
Carlo	10	
Alice	30	

chiave di ordinamento: secondo campo del file da ordinare

`sort -k2 -n esempio.txt`  `output`

Carlo	10
Bruno	15
Gianni	20
Alice	30

Ordina numericamente in quanto il campo scelto come chiave di ordinamento è numerico.

Esercizio n° 8

- Realizzate una combinazione di comandi Unix per visualizzare gli ultimi n (numero scelto) utenti collegati.

Soluzione



- comandi: wc & mv -

wc (Word count)

```
wc [options] [file...]
```

fornisce il numero dei codici di interruzione di riga (in pratica il numero delle righe), delle parole o dei caratteri contenuti in *file*. Senza opzioni fornisce, nell'ordine suddetto, ciascuna delle precedenti informazioni.

Alcune opzioni:

- c emette solo il numero complessivo di caratteri di *file*.
- w emette solo il numero complessivo di parole in *file*.
- l emette solo il numero di righe in *file*.

Esempi di esecuzione

```
gio$ wc which_manpage
132          239          2083  which_manpage
gio$ wc -c which_manpage
2083  which_manpage
gio$
```

Esercizio n° 9

- Con opportuni comandi Unix modificare il file di testo **votoStudenti.txt** formato da m righe rimuovendo le prime n righe (con $m > n$)

Gianni 20

Carlo 10

Bruno 15

Alice 30

conta il numero di linee presenti nel file

Soluzione:

wc -l votoStudenti.txt (m righe è il valore restituito)

tail -k votoStudenti.txt > tmp (dove $k = m - n$)

mv tmp votoStudenti.txt (rinomina il file tmp nel file **votoStudenti.txt**)

- Fine Esercitazione -