

- 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]]**

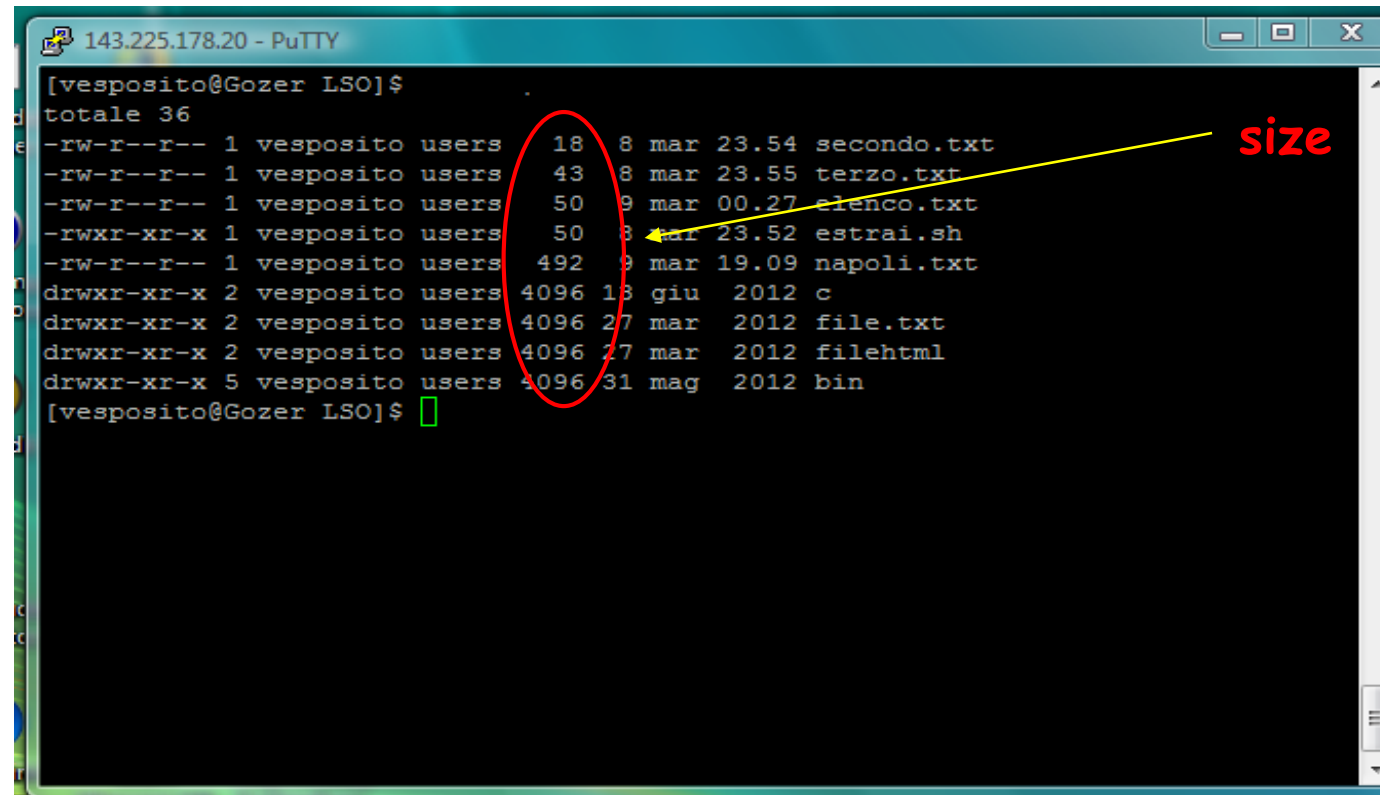
campo\_inizio, campo\_fine

chiavi per restringere l'ordinamento su una porzione di linea

tipo è il tipo di ordinamento applicato sui campi specificati

# Esercizio n° 1

- 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 18 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° 2

- @ 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° 3

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

## Soluzione

chiave di ordinamento in base al quinto campo

inverte il senso di ordinamento, producendo i risultati a partire dall'utente che si è collegato per ultimo

`who | sort -k 5 -r | head -3`

identifica gli utenti attualmente collegati al sistema

Visualizza gli ultimi tre utenti connessi

- comandi: `wc` & `mv` -

# Esercizio n° 1

- @ 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)

- comando di ricerca: `grep` -



# Ricordiamo che...


@ Le righe di `/etc/passwd` si presentano nella seguente forma:

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

➤ Esempio:

Indica la shell di default per quell' utente.

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



# Esercizio n° 1

- ④ Scrivere un comando Unix che visualizza tutti gli utenti diversi contenuti nel file `/etc/passwd`, che usano bash come shell di default

## Sintassi

`grep [opzioni] "stringa" nome_file ...`

## Soluzione

`grep bash$ /etc/passwd`



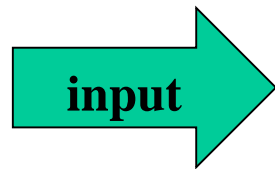
nome del file in cui cercare la stringa

visualizza tutti gli utenti in `/etc/passwd` che hanno come shell di default bash.

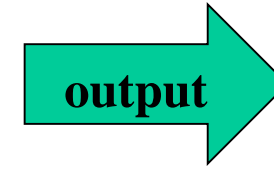
# Esercizio n° 2

- Realizzate una combinazione di comandi Unix per visualizzare il terzultimo file presente nel sistema diverso dal file di nome **pluto.txt**, secondo il normale ordinamento prodotto dal comando che permette di visualizzare tutti i file presenti nel sistema.

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



Sequenza di  
Comandi Unix



lab2.txt

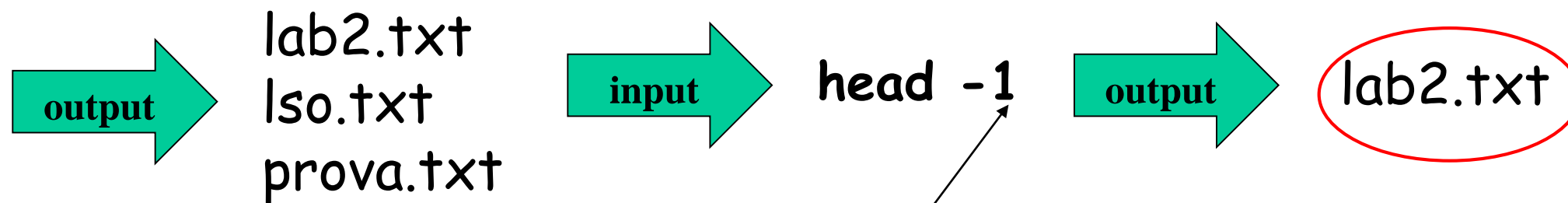
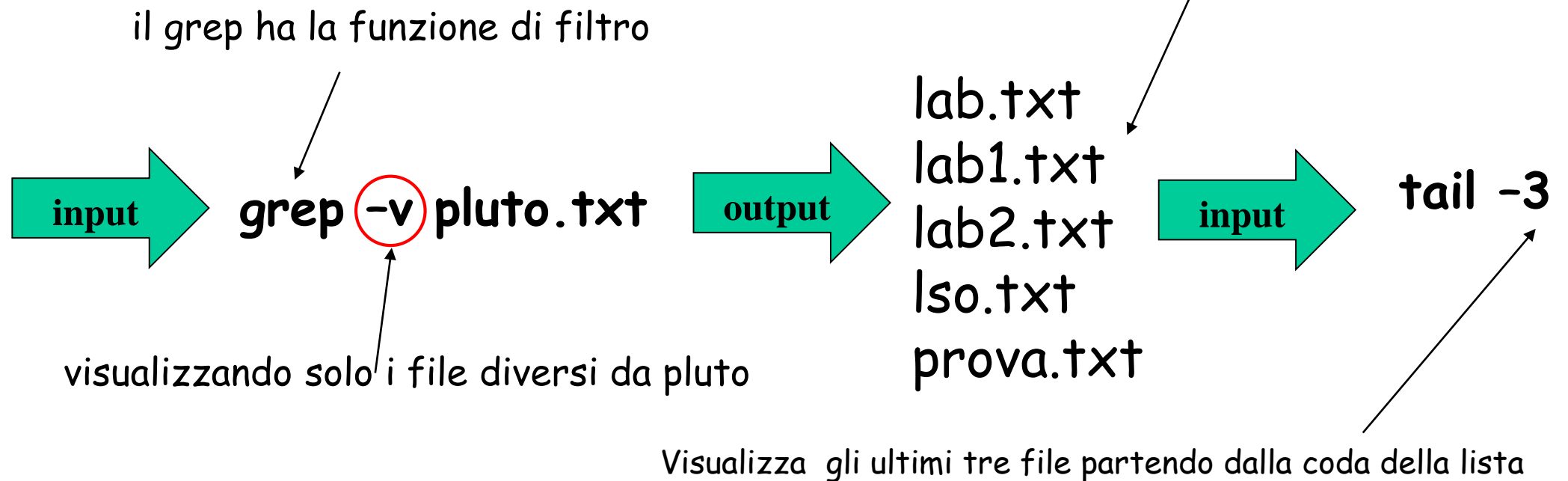
Risultato da ottenere

# Soluzione Esercizio n° 2

Primo comando: **ls -l** ottengo la lista dei file nella directory di lavoro

lista dei file ad esclusione di quello di nome pluto.txt ( filtrato dalla grep )

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



# Soluzione Esercizio n° 2

## Soluzione

Elimina dalla lista dei file passati in input il file di nome pluto.txt

```
ls | grep -v pluto.txt | tail -3 | head -1
```

mostra tutti i file

pipe

Visualizza le ultime tre righe

Visualizza la prima  
riga

# Esercizio n° 3

- Nella propria work directory e in tutte le sottodirectory elencare utilizzando un comando Unix tutti i file con estensione ".txt" (file di testo).

cerca nella directory corrente e nelle sottodirectory i file che terminano con .txt

## Soluzione con (ls + grep)

```
ls -R | grep "\.txt$"
```

cerca nella directory corrente che ha il nome speciale "punto"

## Soluzione con find

```
find . -name "*.txt"
```

al -name usando il carattere jolly "\*" si passa non il singolo nome del file ma tutti i file con estensione "txt"

elenca nel path (directory corrente) tutti i file con estensione .txt

# Esercizio n° 4

- Ⓢ Nella propria work directory creare un file out.txt che contiene tutti i nomi dei file regolari con permesso di eseguibilità per l'utente.

**Soluzione con (ls + grep)**

```
ls -l | grep ^-..x | tr -s ' ' | cut -d' ' -f10 > out.txt
```

# Esercizio n° 5

- Ⓢ Nella propria work directory creare un file out.txt che contiene tutti i nomi dei file regolari dell'utente attuale.

**Soluzione con (ls + grep)**

```
ls -l | cut -c15- | grep ^$USER | tr -s ' ' | cut -d' ' -f8 > out-  
txt
```



# Esercizio n° 6

- Ⓔ Dato il file out.txt creare il file result.txt che contiene tutte le righe di out.txt che finiscono con 'a' e che contengono una 'b'.

**Soluzione con (cat + grep)**

```
cat out.txt | grep .*b.*a$ > result.txt
```

# Esercizio n° 7

- ④ Dato un file test.txt, creare un file output.txt che contiene tutte le righe di test.txt escludendo tutte le righe di test.txt che contengono almeno un punto o una virgola.

**Soluzione con (cat + grep)**

```
cat test.txt | grep -v .*\. .* | grep -v .*,.* > output.txt
```