

Lab. di Sistemi Operativi

- Esercitazione n° 2-

- 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

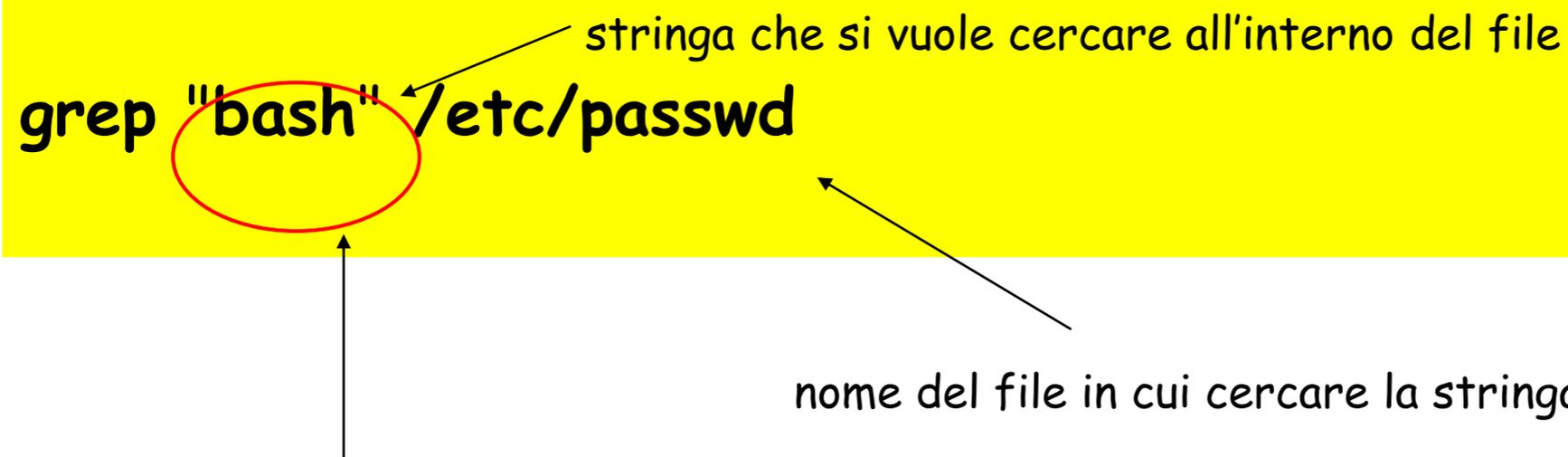
- ④ Scrivere un comando Unix che visualizza tutti gli utenti diversi contenuti nel file `/etc/passwd`, che usano bash come shell di default

Sintassi

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

Soluzione

`grep "bash" /etc/passwd`



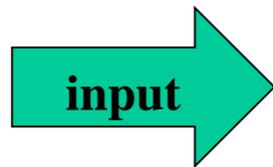
nome del file in cui cercare la stringa

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

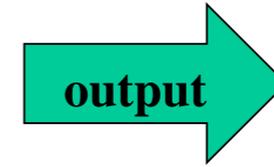
Esercizio n° 2

- 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

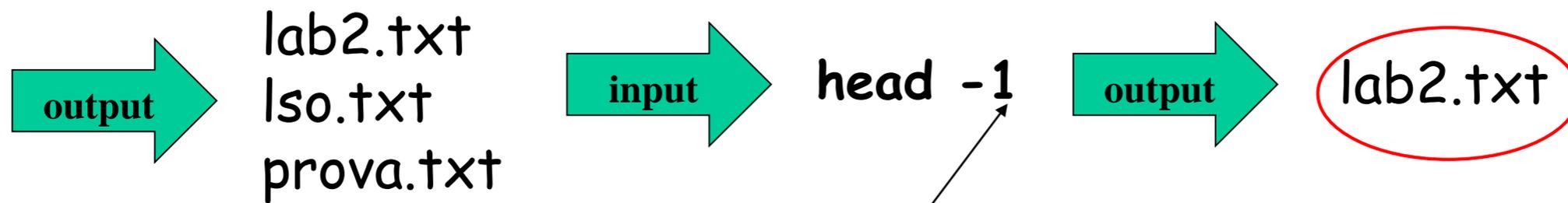
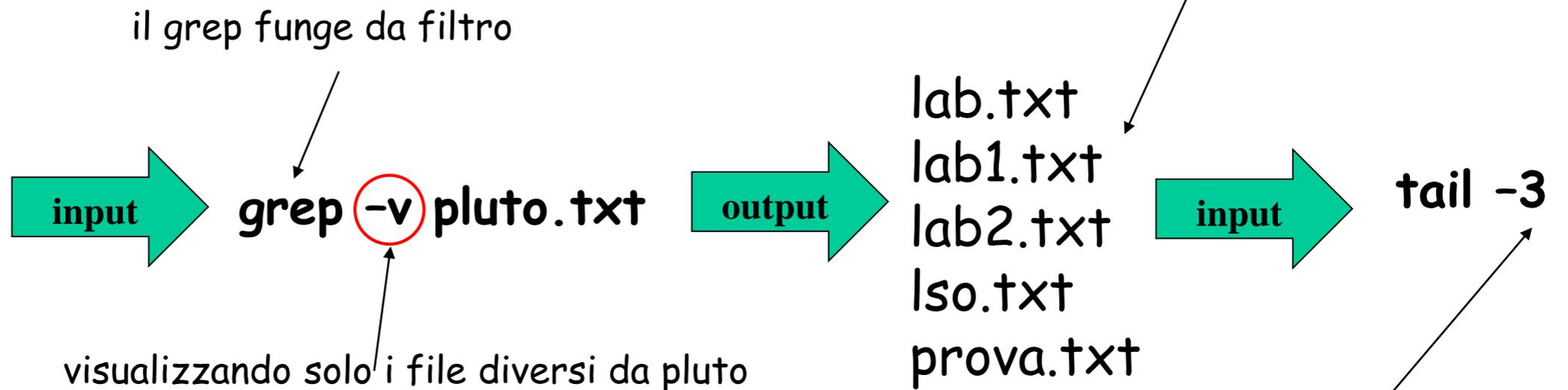


Soluzione Esercizio n° 2

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 -l | 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.*
```

Esercizio n° 5

- Ⓢ 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° 6

- Ⓒ 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
```

- Script shell -

Esercizio n° 7

- ④ Scrivere uno script shell (shell program) di nome **verificaFile** che verifica se tutti i file trovati con `ls` sono quelli trovati con `find`.

Nota: (Creare una sottodirectory **bin** all'interno della propria work directory in cui mettere gli script)

Esempio di lancio dello script:

```
$ ./verificaFile.sh
```

- Controllo di flusso: for -

- Esempi d'uso -

➤ Ciclo **for**

```
for var in lista  
do  
    comando/i che usano $var  
done
```

Esercizio n° 8

- ④ Scrivere uno script shell (shell program) di nome **cercaFileReg** che, nella directory corrente, (di lancio) crea un file di nome **fileReg** contenente l'elenco di tutti i file regolari.

Nota: (Creare una sottodirectory **bin** all'interno della propria work directory in cui mettere gli script)

Suggerimenti:

Usare `$(comando)` per assegnare il risultato del comando lista

Esempio di lancio dello script:

```
$ chmod +x cercaFileReg.sh (permessi per esecuzione)
```

```
$ ./cercaFileReg.sh
```

Soluzione Esercizio n° 8

Script Shell:

```
#!/bin/bash
```

La prima linea dello script deve iniziare con #!, che indica al kernel che lo script è direttamente eseguibile, poi nome dell'interprete dei comandi shell ([Bourne again shell](#)).

```
for file in $(ls)
```

fornisce il contenuto della nostra directory
variabile da testare

```
do
```

```
if [ -f $file ]
```

vero se il file esiste ed è un file regolare

```
then
```

```
    echo $file >> fileReg
```

```
fi
```

```
done
```

espressione
condizionale if

Per effettuare un ciclo tra una lista di valori di tipo stringa si può usare il comando for

Esercizio n° 9

- ④ Scrivere uno script shell di nome **copiaFileC** che, compia le seguenti operazioni:
 - ④ Cerca tutti i file con estensione **.c** nella home directory e in tutte le sottodirectory
 - ④ Crea una cartella **fileC** nella propria directory di lavoro e copia i file trovati in questa cartella
 - ④ Lo script dovrà anche produrre il numero di file trovati
- ④ Suggestimenti:
 - ④ Usare comando find
 - ④ Usare comando grep
 - ④ Usare \$(comando) per assegnare il risultato della ricerca

Soluzione Esercizio n° 9

```
#!/bin/bash ← interprete dei comandi shell
```

```
myhome=$HOME
```

```
echo "Numero file trovati $(find $myhome -name "*.c" |  
grep -c "\.c")"
```

```
cd ..
```

```
mkdir fileC
```

```
for file in $(find $home -name "*.c")
```

```
do
```

```
if [ -f $file ];
```

```
then
```

```
    cp $file fileC
```

```
fi
```

```
done
```

← vero se il file esiste ed è un file regolare

Esercizio n° 9'

- ④ Scrivere uno script shell di nome **copiaFile** che prende come argomento una stringa **<str>** e
 - ④ Cerca tutti i file con estensione **.<str>** nella home directory e in tutte le sottodirectory
 - ④ Crea una cartella **file<str>** nella propria directory di lavoro e copia i file trovati in questa cartella
 - ④ Lo script dovrà anche produrre il numero di file trovati
- ④ Suggestimenti:
 - ④ Usare comando find
 - ④ Usare comando grep
 - ④ Usare `$(comando)` per assegnare il risultato della ricerca
 - ④ `$./copiaFile.sh <str>`

Soluzione Esercizio n° 9'

```
#!/bin/bash ← interprete dei comandi shell
```

```
myhome=$HOME
```

```
echo "Numero file trovati $(find $myhome -name "*. $1" |  
grep -c "\. $1$")"
```

```
cd ..
```

```
mkdir file$1
```

```
for file in $(find $home -name "*. $1")
```

```
do
```

```
if [ -f $file ];
```

```
then
```

```
    cp $file file$1    vero se il file esiste ed è un file  
                       regolare
```

```
fi
```

```
done
```

Esercizio n° 9''

- ④ Scrivere uno script shell **cancellaFile** che prende una stringa **<str>** come argomento, e nella directory corrente, (di lancio) cancella tutti i file che terminano con **.<str>** (cancellare interattivamente con `rm -i`)

Soluzione

```
#!/bin/bash
for file in $(ls *.$1)
do
    if [ -f $file ];
    then
        rm -i $file
    fi
done
```

Esercizio n° 10

- ④ Scrivere uno script shell **cercaDirectory** che, nella directory corrente, (di lancio) crea un file di nome **fileDir** contenente l'elenco di tutte le directory il cui nome inizia per lettera maiuscola

Nota : nella directory di lancio, creare con il comando "**mkdir**" le seguenti directory :

mkdir Nome

mkdir Cognome

mkdir Nome della città di residenza

Soluzione Esercizio n° 10

```
#!/bin/bash      vero se il file esiste ed è una directory
for file in $(ls)
do
if [ -d $file ]
then
    a=$(echo $file | cut -c1)
    b=$(echo $a | grep "[A-Z]")
    if [ ! -z $b ]
    then
        echo $file >> fileDir
    fi
fi
done
```

*in questo modo si seleziona solamente il primo
caratteri di ogni linea. (la prima lettera del nome
della directory)*

Espressione regolare

*File contenente l'elenco delle directory il cui
nome inizia per lettera maiuscola*

- Controllo di flusso: while -

- Esempi d'uso -

- Ciclo **while** esegue la lista di comandi finchè la condizione è vera

```
while condition;  
do  
    comandi  
done
```

- Creazione file **touch**:
 - **touch** nomeFile

Esercizio n° 11 - while -

- Ⓢ Si realizzi uno script "**scriviNumeri.sh**" che scrive a video i numeri da 0 a N: **0,1,2,.....,N-1**
Il valore di N viene passato allo script da riga di comando.
- Ⓢ Esempio di lancio: \$./scriviNumeri.sh N

Soluzione

```
#!/bin/bash
COUNTER=0
while [ $COUNTER -lt $1 ];
do
    echo il valore di counter è
    $COUNTER
    COUNTER=$((COUNTER+1))
done
```

Esercizio n° 11 - while -

- Si realizzi uno script che chiameremo "**creaFiles.sh**" che genera n file vuoti denominati:

node1.html, node2.html, ...nodeN.html

nella directory di lancio. Il valore di N viene passato allo script da riga di comando.

- Esempio di lancio:
\$./creaFiles.sh N

Soluzione Esercizio n° 11 - while -

```
#!/bin/bash
```

```
if test $# -ne 1
```

```
then
```

```
    echo "Wrong number of parameters $#"
```

```
    echo "Usage: $0 param"
```

```
fi
```

```
i=0
```

Inizializzazione della variabile di ciclo

```
while [ $i -lt $1 ];
```

```
do
```

```
    i=$((i+1))
```

```
    touch node$i.html
```

Entra nel ciclo fintanto che la variabile i è minore di n (less than)

```
done
```

Creazione file

- Controllo di flusso: until -

- Esempi d'uso -

- Ciclo **until** esegue la lista di comandi finchè la condizione è falsa

```
until condition;  
do  
    comandi  
done
```

- Alcuni test relativi alle proprietà dei file :
 - **-e** file esiste
 - **-d** file directory
 - **-f** il file esiste ed è regolare

Esercizio n° 12 - until -

- ⓐ Si realizzi uno script che chiameremo "scriviNumeri.sh" che scrive a video i numeri da 20 a 10: **20,19,18,.....,10**
- ⓐ Esempio di lancio: `$./scriviNumeri.sh`

Soluzione

```
#!/bin/bash
COUNTER=20
until [ $COUNTER -lt 10 ];
do
    echo COUNTER: $COUNTER
    COUNTER=$((COUNTER-1))
done
```

Esercizio n° 13 - until -

- ④ Scrivere uno script shell **copiaFile.sh** che riceve da riga di comando due parametri il file da copiare e la directory di destinazione. Lo script inoltre dovrà effettuare il controllo sul numero di parametri passati, che il file da copiare esista e che la directory di copia sia una sottodirectory della work directory.
- ④ Esempio di lancio:
\$./copiaFile.sh <file> <dir>

Soluzione Esercizio n° 13 - until -

```
#!/bin/bash
```

```
if [ "$#" -ne 2 ]; then
```

```
echo "bash: $0: wrong number of arguments."; exit 1
```

```
fi
```

```
file=$1; destdir=$2
```

```
if ! [ -d "$destdir" ] || ! [ -f "$file" ];
```

```
then
```

```
    echo "bash: copiaFile: Usage copiaFile <file>  
        <dir>."; exit 1
```

Condizione comando cp

```
else
```

```
until cp $file $destdir; do
```

```
    echo "Attempt to copy failed. Waiting ...";
```

```
    sleep 5
```

ciclo until

```
done
```

```
fi
```

Se la copia fallisce attende 5 secondi
prima di riprovare

- Fine Esercitazione -