

# Filtraggio di stream di testo

- Contare quanti file esistono con una certa estensione, definita come la stringa posta dopo l'ultimo carattere "punto" presente nel nome del file, per tutte le estensioni trovate nei file presenti nel direttorio corrente e nei sottodirettori. Limitare l'output alle sole 5 estensioni più numerose.
- Comandi da utilizzare:
  - ls
  - rev
  - grep
  - sort
  - uniq
  - head

# Pattern per elaborazione testi

- Per trattare stream prodotti da un processo, lo stesso risultato si può ottenere in modi diversi
  - 1) Dati catturati da command substitution e assegnati a variabili
    - limiti?
  - 2) Dati inviati in input a un ciclo di lettura
    - cautele da osservare?
  - 3) Dati usati come elementi su cui iterare con **for**
    - limiti, cautele?

# Pattern per elaborazione testi / 1

```
#!/bin/bash
# Verificare se c'è un UID libero tra il più alto presente
# nel file /etc/passwd e quello immediatamente inferiore.

MAX=$(cut -f3 -d: /etc/passwd | sort -nr | head -1)
SECOND=$(cut -f3 -d: /etc/passwd | sort -nr | head -2 | tail -1)
DIFF=$(( $MAX - $SECOND ))

if test $DIFF -gt 1
then
    echo puoi aggiungere utenti tra $SECOND e $MAX
else
    echo non puoi aggiungere altri utenti
fi
```

# Pattern per elaborazione testi / 2

```
#!/bin/bash
# Verificare se c'è un UID libero tra il più alto presente
# nel file /etc/passwd e quello immediatamente inferiore.

HIGHEST=0
SECOND=0

cut -f3 -d: /etc/passwd | ( while read U; do
    if test $U -gt $HIGHEST ; then
        SECOND=$HIGHEST
        HIGHEST=$U
    elif test $U -gt $SECOND ; then
        SECOND=$U
    fi
done
echo $HIGHEST
echo $SECOND
)
```

# Pattern per elaborazione testi / 2

```
#!/bin/bash
# Verificare se c'è un UID libero tra il più alto presente
# nel file /etc/passwd e quello immediatamente inferiore.
# Stampare i nomi invece degli UID

cat /etc/passwd | ( while IFS=: read NAME X U RESTO ; do
    if test $U -gt $HIGHEST ; then
        SECOND=$HIGHEST
        SECONDNAME=$HIGHESTNAME
        HIGHEST=$U
        HIGHESTNAME=$NAME
    elif test $U -gt $SECOND ; then
        SECOND=$U
        SECONDNAME=$NAME
    fi
done
echo $HIGHESTNAME
echo $SECONDNAME
)
```

# Pattern per elaborazione testi / 2

```
#!/bin/bash
# Verificare se c'è un UID libero tra il più alto presente
# nel file /etc/passwd e quello immediatamente inferiore.
# Stampare i nomi invece degli UID

cat /etc/passwd | sort -t: -k3nr | head -2 | (
    IFS=: read HIGHESTNAME X HIGHESTID RESTO
    IFS=: read SECONDNAME X SECONDDID RESTO
    DIFF=$(( $HIGHESTID - $SECONDDID ))
    if test $DIFF -gt 1
    then
        echo puoi aggiungere utenti tra $SECONDDID e $HIGHESTID
    else
        echo non puoi aggiungere altri utenti
    fi
)
```

# Esercizio proposto (read, IFS)

Dato un file nel formato:

nomeutente1 indirizzo1

datanascita1 telefono1

nomeutente2 indirizzo2

datanascita2 telefono2

....

produrre in output

nomeutente1 telefono1

nomeutente2 telefono2

...

# Uso delle var. posizionali

- Predisporre uno script che stampi in modo riconoscibile
  - \$1, \$2
  - \$\*, @\$
  - "\$\*", "\$@"

e sperimentare cosa succede invocandolo con diversi parametri (che includano o no spazi)



# Pattern per elaborazione testi / 2

- Stampare gli username corrispondenti agli N più elevati userid presenti nel file /etc/passwd, essendo N un parametro passato sulla riga di comando.
  - awk

# Pattern per elaborazione testi / 3

- Modificare l'esercizio delle estensioni per contare quanti file esistono con una certa estensione, limitatamente all'elenco di estensioni passate come parametri sulla riga di comando.
- Mostrare tutti gli username, chiedendo all'utente conferma di un'ipotetica azione da svolgere per ciascuno.

# Interazione col terminale

- Mostrare il campo GECOS di tutti gli utenti , chiedendo all'utente conferma di un'ipotetica azione da svolgere per ciascuno.
  - limiti di **for** se ci sono spazi?
  - come leggo da tastiera se uso **while read**?
  - comando **tty**
  - file speciali **/dev/pts/\***