

# Esercizi di ricapitolazione su programmazione bash

Sistemi Operativi - A.A. 2023-2024

Corso di Laurea in Matematica

Annalisa Massini - Dipartimento di Informatica - Sapienza Università di Roma

# Esame 31 gennaio 2023 - Esercizio preliminare

Dalla directory home:

- Creare una directory che chiamerete **cognome-matricola** (usando il vostro cognome e la vostra matricola) e all'interno di essa:
  - Creare la directory esercizi e all'interno di essa creare i file **esercizio1**, **esercizio2** ed **esercizio3**
  - Creare la directory esame e all'interno di essa creare i file **file1**, **file2**
  - Nel **file1** scrivere il seguente testo:  
*Gentile Nome Cognome,*  
*con la presente la invitiamo a presentarsi il giorno gg mese aaaa presso il nostro ufficio.*  
*Cordiali saluti*
  - Nel **file2** scrivere una lista di almeno dieci ipotetiche partite della propria squadra del cuore, specificando in ogni riga la squadra avversaria, il giorno e il mese in cui è stata disputata la partita, seguendo il seguente schema:
    - Squadra avversaria spazio giorno (a numero) spazio mese (a lettere)

# Esame 23 febbraio 2023 - Esercizio preliminare

- Dalla **directory home** creare una directory che chiamerete **cognome-matricola** (usando il vostro cognome e la vostra matricola)
- Scrivere **uno script *bash*** denominato **crea-ambiente** che dovrà essere eseguito stando nella **directory cognome-matricola** ed esegue le seguenti azioni:
  - Crea la directory ***esercizi*** e all'interno di essa crea i file ***esercizio1***, ***esercizio2*** ed ***esercizio3***
  - Crea la directory ***esame*** e all'interno di essa crea i file ***file1***, ***file2***
  - Crea le directory ***risultati1***, ***risultati2***, e ***risultati3*** all'interno della directory ***esame***

- Scrivere **uno script *bash*** denominato **crea-ambiente** che dovrà essere eseguito stando nella **directory cognome-matricola** ed esegue le seguenti azioni:
  - Crea la directory ***esercizi*** e all'interno di essa crea i file ***esercizio1***, ***esercizio2*** ed ***esercizio3***
  - Crea la directory ***esame*** e all'interno di essa crea i file ***file1***, ***file2***
  - Crea le directory ***risultati1***, ***risultati2***, e ***risultati3*** all'interno della directory ***esame***

### Script versione lunga

```
mkdir esercizi
touch esercizi/esercizio1
touch esercizi/esercizio2
touch esercizi/esercizio3
mkdir esame
touch esame/file1
touch esame/file2
mkdir esame/risultati1
mkdir esame/risultati2
mkdir esame/risultati3
```

### Script versione compatta

```
mkdir esercizi
touch esercizi/{esercizio1,esercizio2,
    esercizio3}
mkdir -p esame/{risultati1,risultati2,
    risultati3}
touch esame/{file1,file2}
```

# Esame 31 gennaio 2023 - Esercizio 1

**Esercizio 1 (10 punti)** - Nel file ***esercizio1***, scrivere uno script *bash* che esegue le seguenti azioni (dovrà essere eseguito stando nella **directory *esercizi***):

- Crea la directory ***risultati1*** all'interno della directory ***esame***
- Cambia il nome di file ***file1*** presente nella directory ***esame*** in ***lettera.txt***
- Personalizza il contenuto del file ***lettera.txt*** sostituendo: la parola Nome con il proprio nome, la parola Cognome con il proprio cognome, la parola gg con il proprio giorno di nascita scritto a numero, la parola mese con il proprio mese di nascita scritto a parola, la parola aaaa con l'anno attuale scritto a numero, e manda il risultato in un nuovo file che si chiamerà ***lettera-cognome*** da posizionare nella directory ***risultati1***
- Trova le differenze tra il file ***lettera.txt*** presente nella directory ***esame*** e il file ***lettera-cognome*** presente nella directory ***risultati1*** e manda il risultato nel file ***differenze*** da posizionare nella directory ***risultati1***
- **N.B.** Eseguire lo script ***esercizio1*** stando nella directory ***esercizi***.

- Crea la directory **risultati1** all'interno della directory **esame**
- Cambia il nome di file **file1** presente nella directory **esame** in **lettera.txt**
- Personalizza il contenuto del file **lettera.txt** sostituendo:
  - la parola Nome con il proprio nome,
  - la parola Cognome con il proprio cognome,
  - la parola gg con il proprio giorno di nascita scritto a numero,
  - la parola mese con il proprio mese di nascita scritto a parola,
  - la parola aaaa con l'anno attuale scritto a numero, e
  - manda il risultato in un nuovo file che si chiamerà **lettera-cognome** da posizionare nella directory **risultati1**
- Trova le differenze tra il file **lettera.txt** presente nella directory **esame** e il file **lettera-cognome** presente nella directory **risultati1** e manda il risultato nel file **differenze** da posizionare nella directory **risultati1**

## Script

```
mkdir ../esame/risultati1

mv ../esame/file1 ../esame/lettera.txt

sed 's/Nome/Annalisa/;s/Cognome/
    Massini/;s/gg/01/;s/mese/gennaio/;
    s/aaaa/1900/;' ../esame/lettera.txt
>../esame/risultati1/letteracognome

diff ../esame/lettera.txt ../esame/
    risultati1/lettera-cognome >
    ../esame/risultati1/differenze
```

# Esame 31 gennaio 2023 - Esercizio 2

- Nel file ***esercizio2***, scrivere uno script *bash* che esegue le seguenti azioni (**dovrà essere eseguito stando nella directory *esercizi***):
- Crea la directory ***risultati2*** all'interno della directory ***esame***
- Copia ***file2*** presente nella directory ***esame*** in ***partite*** e lo mette nella directory ***risultati2***
- Usando il comando *awk* cercare le gare sostenute nei giorni pari presenti nel file ***partite*** e stampare prima la squadra avversaria e poi il mese per tutte le partite che soddisfano la condizione in un nuovo file che si chiamerà ***pari*** da collocare nella directory ***risultati2***
- Sempre usando il comando *awk*, cercare le gare sostenute nei giorni dispari presenti nel file ***partite*** e stampare prima mese e giorno e poi la squadra avversaria per tutte le partite che soddisfano la condizione in un nuovo file che si chiamerà ***dispari*** da collocare nella directory ***risultati2***
- **N.B.** Eseguire lo script ***esercizio2*** stando nella directory ***esercizi***.

- Crea la directory **risultati2** all'interno della directory **esame**
- Copia **file2** presente nella directory **esame** in **partite** e lo mette nelle directory **risultati2**
- Usando il comando awk cercare le gare sostenute nei giorni pari presenti nel file **partite** e stampare prima la squadra avversaria e poi il mese per tutte le partite che soddisfano la condizione in un nuovo file che si chiamerà **pari** da collocare nella directory **risultati2**
- Sempre usando il comando awk, cercare le gare sostenute nei giorni dispari presenti nel file **partite** e stampare prima mese e giorno e poi la squadra avversaria per tutte le partite che soddisfano la condizione in un nuovo file che si chiamerà **dispari** da collocare nella directory **risultati2**

## Script

```
mkdir ../esame/risultati2

cp ../esame/file2

    ../esame/risultati2/partite

awk '($2%2==0){print $1, $3}'

    ../esame/risultati2/partite >

    ../esame/risultati2/pari

awk '($2%2){print $3, $2, $1}'

    ../esame/risultati2/partite >

    ../esame/risultati2/dispari
```



# Esame 31 gennaio 2023 - Esercizio 3

**Esercizio 3 (8 punti)** - Nel file ***esercizio3***, scrivere uno script *bash* che esegue le seguenti azioni (dovrà essere eseguito stando nella directory ***esercizi***):

- Crea la directory ***risultati3*** all'interno della directory ***esame***
- Mette il risultato del comando `ls -l` eseguito ricorsivamente su tutte le directory a partire dalla directory **cognome-matricola** al file ***file-tutti*** da posizionare nella directory ***risultati3***
- Prende il file ***file-tutti*** come input per eseguire l'ordinamento delle parole in esso contenute e manda il risultato in un nuovo file che si chiamerà ***file-ordinati*** da posizionare nella directory ***risultati3***
- **N.B.** Eseguire lo script ***esercizio3*** stando nella directory ***esercizi***.

- Crea la directory ***risultati3*** all'interno della directory ***esame***
- Mette il risultato del comando `ls -l` eseguito ricorsivamente su tutte le directory a partire dalla directory **cognome-matricola** al file ***file-tutti*** da posizionare nella directory ***risultati3***
- Prende il file ***file-tutti*** come input per eseguire l'ordinamento delle parole in esso contenute e manda il risultato in un nuovo file che si chiamerà ***file-ordinati*** da posizionare nella directory ***risultati3***

## Script

```
mkdir ../esame/risultati3

ls -lR .. > ../esame/risultati3/file-tutti

sort ../esame/risultati3/file-tutti >

    ../esame/risultati3/file-ordinati
```

# Esercizio

- Creare una nuova directory chiamata `mydir` in cui mettete tre file con estensione `.txt` denominati `uno.txt`, `due.txt` e `tre.txt`, qualche riga di testo
- Scrivere uno script **estensioni** che:
  - controlla se nella directory ci sono file con estensione `.doc` e se non ci sono scrive su schermo : «Nessun file `.dat`», altrimenti per ogni file `.dat` crea un nuovo file con estensione `.doc` e ci appende il contenuto del file `.dat` scrivendo su schermo quale file si aggiunge
  - controlla se nella directory ci sono file con estensione `.txt` e se non ci sono scrive: «Nessun file `.txt`» altrimenti per ogni file `.txt` crea un nuovo file con estensione `.dat` e ci appende il contenuto del file `.txt` scrivendo su schermo quale file si aggiunge
- Eseguire più volte lo script **estensioni** e osservare come cambia la lista di file nella directory e i file in essa contenuti

|  |  |
|--|--|
| # questo script riconosce l'estensione dei file .dat e .txt, la cambia in doc e dat, appende il contenuto di uno all'altro |  |
| 1 listadat=`ls *.dat 2> testdat`   |  |
| 2 echo 'listadat:' \$listadat  |  |
| 3 if [[ -s testdat ]] ; then   |  |
| 4     echo "Nessun file dat "  |  |
| 5 else   |  |
| 6     for i in \$listadat ; do   |  |
| 7         echo \$i   |  |
| 8         j=`echo \$i   sed -e 's/dat\$/doc/'`   |  |
| 9         echo \$j   |  |
| 10        cat \$i >> \$j   |  |
| 11        echo "aggiunto \$i in \$j"   |  |
| 12    done   |  |
| 13 fi  |  |

|   |
|---|
| Alla riga 1 definiamo la variabile <b>listadat</b> che conterrà il risultato del comando <b>ls *.dat</b> , cioè l'elenco dei file, mettendo l'eventuale messaggio di errore (nel caso non ci siano file <b>.dat</b> ) nel file <b>testdat</b> |
| Alla riga 2 ci facciamo stampare il valore della variabile  |
| Alla riga 3 si controlla se il file <b>testdat</b> ha dimensione maggiore di 0 ( <b>-s</b> ), ovvero se non è vuoto, cioè se contiene una segnalazione di errore  |
| In caso positivo - riga 4 - si scrive a schermo il messaggio «Nessun file da dat»   |
| Se invece il file <b>testdat</b> è vuoto, si esegue un ciclo - righe da 6 a 12 - su tutti i file compresi nella variabile lista   |
| Alla riga 8 viene definita la variabile j contenente il nuovo nome da assegnare al file   |
| Alla riga 10 il file <b>.dat</b> viene appeso al file <b>.doc</b>   |
| Alla riga 11 si scrive quale file viene aggiunto e dove   |

|   |  |
|---|--|
| # questo script riconosce l'estensione dei file .dat e .txt, la<br>cambia in doc e dat, appende il contenuto di uno all'altro |  |
| 1 listadat=`ls *.dat 2> testdat`  |  |
| 2 echo 'listadat:' \$listadat   |  |
| 3 if [[ -s testdat ]] ; then  |  |
| 4     echo "Nessun file dat "   |  |
| 5 else  |  |
| 6     for i in \$listadat ; do  |  |
| 7         echo \$i  |  |
| 8         j=`echo \$i   sed -e 's/dat\$/doc/'`  |  |
| 9         echo \$j  |  |
| 10        cat \$i >> \$j  |  |
| 11        echo "aggiunto \$i in \$j"  |  |
| 12    done  |  |
| 13 fi   |  |

I comandi alle righe 7 e 9 visualizzano su schermo il nome degli elementi su cui agisce il ciclo **for** e il nuovo elemento j definito ad ogni iterazione alla riga 8. Sono comandi inseriti per seguire l'andamento dell'esecuzione e possono essere omessi

Nel comando alla riga 8 (in cui viene definita la variabile j contenente il nuovo nome da assegnare al file) si usano:

- l'operatore *backtick* che esegue un comando e ne restituisce l'output
- l'operatore pipe |, che concatena due comandi, passando l'output del primo comando in input al comando successivo
- il comando **echo** che invia i suoi argomenti su standard output
- il comando **sed** che viene utilizzato per manipolare la stringa contenente il nome del file

# questo script riconosce l'estensione dei file .dat e .txt, la cambia in doc e dat, appende il contenuto di uno all'altro

```
1 listadat=`ls *.dat 2> testdat`
2 echo 'listadat:' $listadat
3 if [[ -s testdat ]] ; then
4     echo "Nessun file dat "
5 else
6     for i in $listadat ; do
7         echo $i
8         j=`echo $i | sed -e 's/dat$/doc/'`
9         echo $j
10        cat $i >> $j
11        echo "aggiunto $i in $j"
12    done
13 fi
```

In particolare, per l'istruzione alla riga 8:

- utilizzando l'operatore backtick, la variabile `j` viene definita dall'output del comando

**echo \$i | sed -e 's/dat\$/doc/'**

- In questa istruzione, l'operatore pipe passa l'output del comando **echo \$i**, cioè il nome del file preso in esame, come input al comando **sed**
- **sed** applica sulla stringa che riceve in input il comando di editing **'s/dat\$/doc/'** specificato tramite l'opzione **-e**, cioè sostituisce la stringa **dat** che chiude il nome del file, con la stringa **doc**

# questo script riconosce l'estensione dei file .dat e .txt, la cambia in doc e dat, appende il contenuto di uno all'altro

```
1 listadat=`ls *.dat 2> testdat`
2 echo 'listadat:' $listadat
3 if [[ -s testdat ]] ; then
4     echo "Nessun file dat "
5 else
6     for i in $listadat ; do
7         echo $i
8         j=`echo $i | sed -e 's/dat$/doc/' `
9         echo $j
10        cat $i >> $j
11        echo "aggiunto $i in $j"
12    done
13 fi
```

Inoltre:

- N.B. Il carattere **\$** dopo la stringa dat (cioè **dat\$**) serve ad indicare che solo l'occorrenza della stringa **dat** che chiude il nome del file deve essere sostituita e non altre eventuali occorrenze comprese nel nome del file
- Quindi, se ad esempio il nome del file fosse **nuovi\_dati.dat**:
  - se non si usasse il **\$** (cioè **s/dat/doc/**) la sostituzione avrebbe come effetto che il file sarebbe rinominato **nuovi\_doci.dat** perché sarebbe sostituita la prima occorrenza di dat
  - se si usasse **s/dat/doc/g** la sostituzione avverrebbe su tutte le occorrenze di dat e avrebbe come effetto che il file sarebbe rinominato **nuovi\_doci.doc**

# questo script riconosce l'estensione dei file .dat e .txt, la cambia in doc e dat, appende il contenuto di uno all'altro

```
14 testtxt=errore_txt
15 listatxt=`ls *.txt 2> $testtxt`
16 echo 'listatxt' $listatxt
17 if [[ -s $testtxt ]] ; then
18     echo "Nessun file txt"
19 else
20     for i in $listatxt ; do
21         echo $i
22         j=`echo $i | sed -e 's/txt$/dat/'`
23         echo $j
24         cat $i >> $j
25         echo "aggiunto $i in $j"
26     done
27 fi
28 exit
```

- La seconda parte dello script agisce nello stesso modo sui file con estensione **.txt**
- In questo caso usiamo un altro modo per definire il file in cui mettere l'errore
- Alla riga 14, viene definita la variabile **testtxt** contenente il nome del file utilizzato per memorizzare eventuali segnalazioni di errore, cioè il file **errore\_txt**
- Se nella directory corrente non ci sono file **.txt**, la segnalazione viene indirizzata sul file **errore\_txt** usando la variabile **testtxt** preceduta dal **\$** per considerarne il valore e non semplicemente il nome

Come nella prima parte dello script, i comandi alle righe 21 e 23 visualizzano su schermo il nome degli elementi su cui agisce il ciclo **for** e il nuovo elemento **j** definito ad ogni iterazione alla riga 22. Possono essere omessi



# questo script riconosce l'estensione dei file .dat e .txt, la cambia in doc e dat, appende il contenuto di uno all'altro

```
1 listadat=`ls *.dat 2> testdat`
2 echo 'listadat:' $listadat
3 if [[ -s testdat ]] ; then
4     echo "Nessun file dat "
5 else
6     for i in $listadat ; do
7         echo $i
8         j=`echo $i | sed -e 's/dat$/doc/'`
9         echo $j
10        cat $i >> $j
11        echo "aggiunto $i in $j"
12    done
13 fi
```

```
14 testtxt=errore_txt
15 listatxt=`ls *.txt 2> $testtxt`
16 echo 'listatxt' $listatxt
17 if [[ -s $testtxt ]] ; then
18     echo "Nessun file txt"
19 else
20     for i in $listatxt ; do
21         echo $i
22         j=`echo $i | sed -e 's/txt$/dat/'`
23         echo $j
24         cat $i >> $j
25         echo "aggiunto $i in $j"
26     done
27 fi
28 exit
```