

## Esercizio 1

Si scriva uno script perl dal nome "sharedata" utilizzato per la condivisione e l'estrazione di porzioni di un file data.txt tra più utenti. Il comando deve rispettare la seguente sintassi:

sharedata OPTION

Il comando accetta diversi valori per OPTION; in particolare, sharedata invocato con l'opzione -i legge da **standard input** una serie di comandi (uno per riga) nel formato:

MITT:[DEST1,DEST2]:NOMEFILE:DIM

La descrizione dei diversi elementi che compongono il comando è la seguente:

- MITT: rappresenta il nome del mittente ;
- [DEST1,DEST2,...]: rappresenta una lista di nomi utente destinatari del pacchetto di dati (tra parentesi quadre separati da virgola);
- NOMEFILE: è il nome del file in cui vengono salvati i dati condivisi;
- DIM: è il numero di byte da condividere.

Ogni elemento è separato dal successivo da : (due punti).

Il comando sharedata lanciato con l'opzione -i esegue le seguenti azioni:

- per ogni DEST:
  - crea nella directory corrente una directory di nome DEST se questa non esiste;
  - all'interno della directory DEST crea un file di nome NOMEFILE se questo non esiste;
  - all'interno del file NOMEFILE **accoda** i primi DIM byte del file data.txt .

### Esempio1:

supponiamo di scrivere da linea di comando

sharedata -i

Il comando si mette in attesa dell'input dell'utente; da tastiera digitiamo i seguenti comandi:

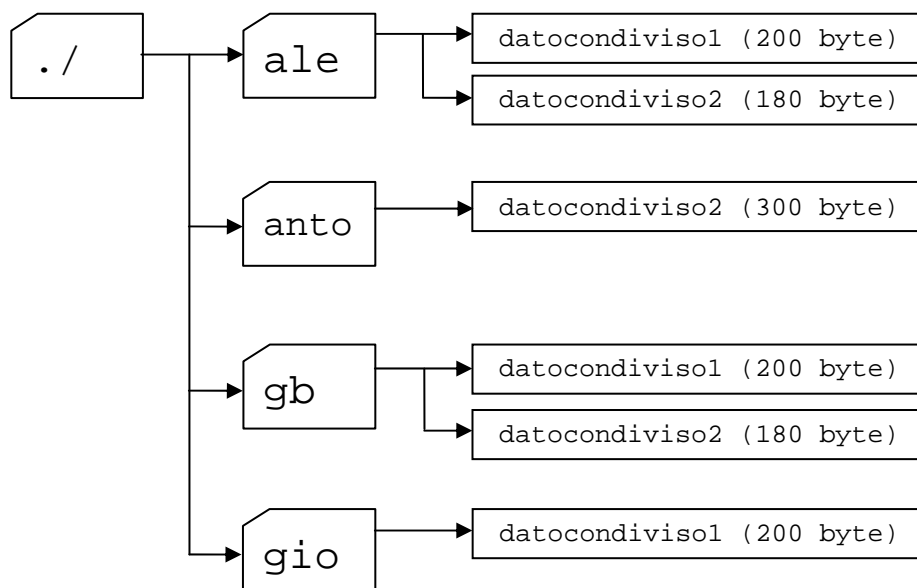
```
cla:[gb,ale,gio]:datocondiviso1:200  
vik:[gb,ale,anto]:datocondiviso2:180  
gio:[anto]:datocondiviso2:120  
(Ctrl-d)
```

Nell'ipotesi in cui la directory corrente è vuota, il primo comando creerà le directory *gb*, *ale* e *gio* all'interno delle quali creerà un file, *datocondiviso1*, contenente i primi 200 byte del file *data.txt*.

Il secondo comando creerà la directory *anto* (le directory *gb* e *ale* sono già esistenti) e creerà all'interno delle directory *gb*, *ale* e *anto* il file *datocondiviso2* contenente i primi 180 byte del file *data.txt*.

Il terzo comando accoderà nel file *datocondiviso2* presente nella directory *anto* i primi 120 byte del file *data.txt*; Ctrl-d indica la fine dell'input.

La situazione della directory dopo l'esecuzione del comando *sharedata* sarà la seguente:



## Esercizio 2

Si consideri il comando *sharedata* definito nell'esercizio precedente, e si assuma di avere nella cartella una struttura di directory e file ottenuta tramite l'invocazione di uno o più comandi *sharedata* (lanciato con l'opzione `-i`).

Si implementino in perl le OPTION `-s` e `-f` con il seguente significato:

- `sharedata -s` stampa su standard output, per ogni nome utente, la dimensione complessiva in byte dei file ricevuti attraverso il comando `sharedata`.
- `sharedata -f` legge da **standard input** il nome di un file e stampa su standard output l'elenco degli utenti che hanno ricevuto quel file con la dimensione in byte del file stesso.

### Esempio2:

supponiamo di avere la struttura di directory e file vista nell'Esempio1 il comando

```
sharedata -s
```

stamperà il seguente output:

user		bytes
----	+	-----
ale		380
anto		300
gb		380
gio		200

mentre il comando

```
sharedata -f
```

si metterà in attesa dell'input dell'utente; da tastiera digitiamo:

```
datocondiviso2
```

il comando stamperà:

```
datocondiviso2: owners (bytes)
+++++
ale (180)
anto (300)
gb (180)
```

## NOTE

- 1) L'opzione `-c` del comando `head` consente di leggere i primi  $n$  byte di un file:

Per esempio lanciando il comando:

```
head -c 180 /etc/fstab
```

su standard output verranno stampati i primi 180 byte del file `/etc/fstab`.

- 2) Utilizzare il file `data.txt` e `sharing.txt` forniti con il file contenente la traccia d'esame per testare lo script `sharedata`.