

## ESERCIZIO 2 (Linguaggi di scripting. Punti 0-10)

Il comando `bash cat /proc/meminfo` visualizza alcune informazioni riguardo la memoria

### **Esempio 1**

Un possibile output del comando `cat /proc/meminfo` è:

```
MemTotal:      8167848 kB
MemFree:       1409696 kB
Buffers:       961452 kB
Cached:       2347236 kB
...
```

In questo esercizio bisogna realizzare uno script Perl, che verrà salvato nel file `memory_script.pl`. Lo script accetta un parametro opzionale su linea di comando (che chiameremo `TIME`), che rappresenta un tempo espresso in secondi, ed esegue il comando **C**, dove **C** = “`cat /proc/meminfo`”, una volta ogni `TIME` secondi. Se il parametro `TIME` non è specificato, bisogna assumere un valore di default di **2** secondi.

Lo script dovrà girare finché non verrà interrotto con `CTRL-C`.

Ad ogni esecuzione del comando **C**, lo script deve visualizzare su Standard Output la differenza tra memoria libera (`MemFree`) registrata a questa esecuzione rispetto a quella dell'esecuzione precedente. Bisogna visualizzare il tutto seguendo il formato specificato nella sezione **Formato Output - Standard Output**.

Si assuma che inizialmente la memoria libera sia **0**.

Inoltre, ogni **5** volte che viene eseguito **C** (per esempio ogni 10 secondi (**5\*2**) se non c'è nessun parametro specificato), lo script dovrà stampare, in modalità *append*, su un file chiamato `mem_free.log`, un report di tutte le differenze trovate. Tale report deve seguire il formato specificato nella sezione **Formato Output - Log File**.

### **Formato Output - Standard Output (DA RISPETTARE RIGOROSAMENTE)**

Sia `D` la differenza numerica tra il valore di `MemFree` restituito da **C** durante l'esecuzione corrente e quella precedente. Se `D` è negativo bisogna stampare il carattere '<'; se invece `D` è positivo bisogna stampare il carattere '>'; se `D` è nullo bisogna stampare il carattere '='. Bisogna poi stampare uno spazio vuoto seguito dal valore di `D` con l'unità di misura (come riportato nell'output del comando **C**), seguito da un carattere di nuova linea '\n'.

### **Formato Output - Log File (DA RISPETTARE RIGOROSAMENTE)**

Bisogna raggruppare gli eventi accaduti nelle ultime 5 esecuzioni di **C** per “tipo”. I due tipi possibili sono gli “incrementi” e i “decrementi”.

Si devono prima visualizzare i dati relativi agli incrementi, stampando il carattere 'i', seguito da un carattere di nuova linea '\n', seguito dalla lista di tutti gli eventi di incremento (separati dal carattere ','). Sul rigo successivo bisogna stampare il carattere 'd', seguito da un carattere di nuova linea '\n', seguita dalla lista di tutti i decrementi (separati dal carattere ','). Infine bisogna stampare sul rigo successivo 10 volte il carattere '-', seguito da un carattere di nuova linea '\n'.

**Nota** in Perl è possibile sospendere l'esecuzione per X secondi usando la primitiva 'sleep X'.

**Nota** Si può assumere che l'unità di misura sia sempre la stessa

## Esempio Output - Standard Output

```
> 13384 kB
> 9844 kB
= 0 kB
< -2292 kB
< -372 kB
```

## Esempio Output - Log File

```
i
13384,9844
d
-2292,-372
-----
```