

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

Il comando `bash top -n 1 -b | tail -n +7` mostra alcune informazioni riguardanti l'utilizzo della CPU e della memoria da parte di utenti e processi.

Esempio 1

Un possibile output del comando `top -n 1 -b | tail -n +7` è:

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
2226 neutron 20 0 277852 124484 11932 S 6.2 0.1 6:30.42 neutron-linuxbr
2573 nova 20 0 298392 136396 7400 S 6.2 0.1 10:43.46 nova-conductor
2611 nova 20 0 445904 191720 13152 S 6.2 0.1 1:09.51 nova-api
21269 pacenza 20 0 41932 3920 3144 R 6.2 0.0 0:00.02 top
1 root 20 0 37976 6140 4092 S 0.0 0.0 0:11.74 systemd
... ..
```

In questo esercizio bisogna realizzare uno script Perl, che verrà salvato nel file `top_stat.pl`. Lo script accetta un parametro opzionale su linea di comando che specifica il tipo di verifica che si vuole fare. L'argomento in input può essere `-c` (statistica sull'utilizzo della CPU in percentuale) o `-m` (statistica sull'utilizzo della memoria in percentuale).

Se non dovesse essere inserito alcun parametro, lo script, di default, restituirà informazioni riguardanti l'utilizzo del processore.

Ad ogni esecuzione, lo script dovrà mostrare su Standard Output, in ordine **decrescente**, l'utilizzo totale del processore o della memoria in percentuale, raggruppato per ogni singolo utente.

Bisogna visualizzare il tutto seguendo il formato specificato nella sezione **Formato Output – Standard Output**.

Successivamente sarà necessario salvare l'output su un **file** di log che chiameremo `top_stat.log`.

Bisogna visualizzare il tutto seguendo il formato specificato nella sezione **Formato Output – Log File**.

BONUS (max 1pt su parte Perl): A parità di utilizzo delle risorse richieste, si richiede di ordinare in primo luogo per valore decrescente di utilizzo delle risorse, ed in secondo luogo anche per ordine lessicografico degli utenti.

Esempio Output - Standard Output

pacenza@pcino:~\$ perl script.pl -c

```
USER --- %CPU
root --- 23.6%
pacenza --- 11.8%
clamav --- 0%
cvsd --- 0%
```

pacenza@pcino:~\$ perl script.pl -m

```
USER --- %MEM
root --- 4.5%
pacenza --- 0.8%
clamav --- 0%
cvsd --- 0%
```

N.B.: Le stampe sono ordinate prima per valore decrescente di utilizzo delle risorse e a parità di uso delle risorse in ordine lessicografico degli utenti (si notino gli utenti `clamav` e `cvsd` che usano entrambi lo 0% delle risorse, di conseguenza sono ordinati lessicograficamente).

È **obbligatorio** ordinare in ordine **decrescente** l'uso dei processi, mentre è facoltativo e rientra nel **bonus** ordinare anche **lessicograficamente**.

Esempio Output - Log File

```
USER --- %CPU
root --- 23.6%
pacenza --- 11.8%
clamav --- 0%
cvsd --- 0%
#####
USER --- %MEM
root --- 4.5%
pacenza --- 0.8%
clamav --- 0%
cvsd --- 0%
#####
```

Formato Output - Standard Output (DA RISPETTARE RIGOROSAMENTE)

Per ogni utente u mostrato nell'output del comando `top` (si veda colonna `USER`), sia s la quantità totale di risorsa consumata da u .

Stampare sulla prima riga:

```
USER - - - %CPU
```

Oppure

```
USER - - - %MEM
```

Successivamente stampare ogni utente su una riga con la quantità di risorsa utilizzata separata da 3 trattini - - -

```
 $u_1$  - - -  $s_1\%$ 
```

```
 $u_2$  - - -  $s_2\%$ 
```

```
 $u_3$  - - -  $s_3\%$ 
```

```
...      ...
```

Formato Output - Log File (DA RISPETTARE RIGOROSAMENTE)

Si deve stampare su file, in modalità `append`, l'output di ogni esecuzione dello script.

Ogni nuova stampa deve essere separata dalla precedente da 20 #

```
USER - - - %MEM
```

```
 $u_1$  - - -  $s_1\%$ 
```

```
 $u_2$  - - -  $s_2\%$ 
```

```
 $u_3$  - - -  $s_3\%$ 
```

```
...      ...
```

```
#####
```

```
USER - - - %CPU
```

```
 $u_1$  - - -  $s_1\%$ 
```

```
 $u_2$  - - -  $s_2\%$ 
```

```
 $u_3$  - - -  $s_3\%$ 
```

```
...      ...
```

```
#####
```

```
USER - - - %CPU
```

```
 $u_1$  - - -  $s_1\%$ 
```

```
 $u_2$  - - -  $s_2\%$ 
```

```
 $u_3$  - - -  $s_3\%$ 
```

```
...      ...
```

```
#####
```