

# Corso di Sistemi Operativi e Reti

Prova scritta 1 SETTEMBRE 2020

## ISTRUZIONI PER CHI È IN PRESENZA:

1. **Rinomina** la cartella chiamata "Cognome-Nome-Matricola" che hai trovato sul Desktop e in cui hai trovato questa traccia, sostituendo "Cognome" "Nome" e "Matricola" con i tuoi dati personali e **lasciando i trattini**; se hai un doppio nome oppure un doppio cognome dovrai chiamare la cartella come in questo esempio:
  - a. DeLuca-MarcoGiovanni-199999
2. **Carica** tutto il materiale didattico che vorrai usare sul Desktop; puoi farlo solo nei primi 5 minuti della prova;
3. **Svolgi** il compito; lascia tutto il sorgente che hai prodotto nella cartella di cui al punto 1;
4. Quando hai finito lascia la postazione facendo logout,

**senza spegnere il PC.**

## ISTRUZIONI PER CHI SI TROVA ONLINE:

1. **Questo file contiene il testo che ti è stato dato ieri, incluso il codice;**
2. **Mantieni a tutto schermo** questo file per tutta la durata della prova; puoi scorrere liberamente tra le sue pagine, ma non puoi cambiare applicazione;
3. **Firma** preliminarmente il foglio che userai per la consegna con nome cognome e matricola;
4. **Svolgi** il compito; puoi usare solo carta, penna e il tuo cervello;
5. **Alla scadenza** termina *immediatamente* di scrivere, e attendi di essere chiamato, pena l'esclusione dalla prova;
6. **Quando è il tuo turno** mostra il foglio ben visibile in webcam, e poi metti una foto dello stesso foglio in una chat privata Microsoft Teams con il prof.

## ESERCIZIO 2, TURNO 1 - PERL

Si scriva uno script Perl dal nome `resources.pl` che sia in grado di fornire dettagli riguardanti l'utilizzo delle risorse (CPU o memoria) da parte di uno specifico utente.

In particolare, lo script riceve come argomenti da linea di comando esattamente 2 opzioni: la prima identifica il tipo di risorsa da andare a consultare (parametro `-c` per consultare la CPU, parametro `-m` per consultare la memoria) mentre la seconda identifica l'**utente** da filtrare.

Lo script dovrà essere eseguito, quindi, con la seguente sintassi:

```
./resources.pl [-c|-m] username
```

È possibile ottenere informazioni riguardanti l'utilizzo della memoria e del processore eseguendo, tramite opportuna funzione perl, il comando shell atto a visualizzare in tempo reale i processi attivi nel sistema seguito dai parametri `-n1 -b`. Un possibile output di questo comando è il seguente:

```
Tasks: 146 total,  1 running, 105 sleeping,  0 stopped,  0 zombie
%Cpu(s):  3,7 us,  1,7 sy,  0,0 ni, 93,0 id,  1,3 wa,  0,0 hi,  0,2 si,  0,0 st
KiB Mem : 5830408 total, 138856 free, 1162656 used, 4528896 buff/cache
KiB Swap: 2097148 total, 2081520 free,  15628 used. 4390164 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1041 frances+  20   0 860124 110028 16088 S  5.9  1.9 102:22.72 node
 1184 plex      35  15 817384  59372  6072 S  5.9  1.0 13:18.69 Plex Script Hos
11643 frances+  20   0  41780   3456  2948 R  5.9  0.1  0:00.01 top
   1 root      20   0 225672   6628  4336 S  0.0  0.1  2:00.78 systemd
   2 root      20   0     0     0     0 S  0.4  0.0  0:00.22 kthreadd
   3 root      20   0     0     0     0 S  1.0  0.0  0:01.22 apache2
```

Le colonne di interesse per l'esecuzione di questo script sono: `USER`, `%CPU` o `%MEM` (in base all'opzione inserita in input).

Ad ogni esecuzione, lo script deve stampare su un file dal nome `stat.log` un rapporto contenente la percentuale totale di `%CPU` (risp. `%MEM`) utilizzata dall'utente inserito come parametro dello script e, infine, il nome dell'utente che al momento dell'esecuzione dello script ha un utilizzo percentuale maggiore di `%CPU` (risp. `%MEM`) seguito dal valore totale calcolato.

**Esempio:**

Se l'output del comando fosse quello di cui sopra, e gli argomenti passati allo script fossero `-c root`, lo script creerebbe un file dal nome `stat.log` contenente le seguenti due righe:

```
utente root CPU: 1.4%  
Max uso CPU: frances+ 11.8%
```

**Importante:**

1. Tutti i match sulle stringhe devono essere effettuati tramite opportuna REGEXP case insensitive.
2. La stampa su file dovrà essere effettuata tramite funzione del linguaggio Perl.

## OUTPUT COMPLETO DEL COMANDO SHELL DA ESEGUIRE

Tasks: 146 total, 1 running, 105 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 3,7 us, 1,7 sy, 0,0 ni, 93,0 id, 1,3 wa, 0,0 hi, 0,2 si, 0,0 st  
KiB Mem : 5830408 total, 137388 free, 1162844 used, 4530176 buff/cache  
KiB Swap: 2097148 total, 2081520 free, 15628 used. 4389976 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
13292	frances+	20	0	41780	3528	3020	R	12,5	0,1	0:00.02	top
1	root	20	0	225672	6628	4336	S	0,0	0,1	2:00.93	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.22	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-kb
8	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
9	root	20	0	0	0	0	S	0,0	0,0	1:02.55	ksoftirqd/0
10	root	20	0	0	0	0	I	0,0	0,0	2:45.43	rcu_sched
11	root	rt	0	0	0	0	S	0,0	0,0	0:02.44	migration/0
12	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject/0
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
16	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
17	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_tasks_kthre
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kauditd
19	root	20	0	0	0	0	S	0,0	0,0	0:00.49	khungtaskd
20	root	20	0	0	0	0	S	0,0	0,0	0:00.00	oom_reaper
21	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	writeback
22	root	20	0	0	0	0	S	0,0	0,0	0:01.46	kcompactd0
23	root	25	5	0	0	0	S	0,0	0,0	0:00.00	ksmd
24	root	39	19	0	0	0	S	0,0	0,0	0:00.00	khugepaged
116	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kintegrityd
117	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kblockd

118	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	blkcg_punt_bio
119	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	tpm_dev_wq
120	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	ata_sff
121	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	md
122	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	edac-poller
123	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	devfreq_wq
124	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdogd
125	root	20	0	0	0	0	S	0,0	0,0	1:01.35	kswapd0