

Sistemi Operativi – a.a. 2016/2017

prova di laboratorio

– 28 luglio 2016 –

Creare un programma `cpu-monitor.c` in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

```
cpu-monitor [number of samples]
```

Il programma, attraverso l'utilizzo del file di testo speciale `/proc/stat` del file-system virtuale `/proc/`, raccoglie sample sull'utilizzo del comparto CPU del sistema su cui esso viene eseguito. Vengono raccolti un numero di sample pari a quelli eventualmente specificati sulla riga di comando (30 di default) per poi essere visualizzati a video tramite un semplice grafico ed in termini percentuali.

La prima riga del file di testo speciale `/proc/stat` ha il seguente formato:

```
cpu 10403824 223264 2676976 12200724 275004 0 21309 0 0 838
```

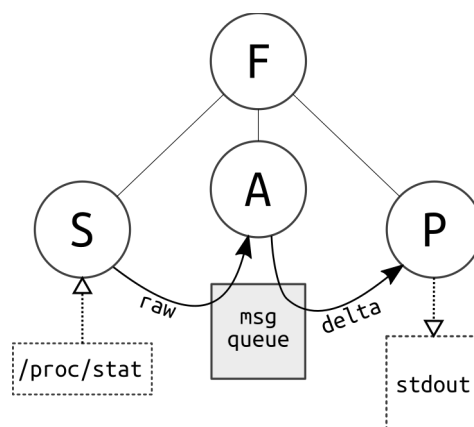
dove i numeri sono contatori cumulativi di unità di tempo (dette *jiffies*) spese dai vari core, rispettivamente, in:

- **user**: processi normali in modalità utente;
- **nice**: processi, sotto nice, in modalità utente;
- **system**: processi in modalità kernel;
- **idle**: momenti di inutilizzo;
- **iowait**: in attesa di completamento di I/O;
- **irq**: in gestione degli interrupt hardware;
- **softirq**: in gestione degli interrupt software.

Per semplicità, noi prenderemo in considerazione solo il tempo utilizzato in modalità utente (**user**), quello in modalità kernel (**system**) e quello di inattività (**idle**).

Il programma padre **F**, al suo avvio, creerà tre processi figli **Sampler**, **Analyzer** e **Plotter**. I figli comunicheranno tra di loro unicamente attraverso una coda di messaggi creata dal padre. Sono previsti due tipi di messaggi:

- **messaggi raw**: trasportano 3 interi che rappresentano i dati cumulativi (**user**, **system**, **idle**) presi direttamente dal file `/proc/stat`;
- **messaggi delta**: trasportano 2 numeri in virgola mobile che rappresentano, rispettivamente, le percentuali di utilizzo in modalità utente e in modalità kernel tra un sample ed il successivo.



I ruoli dei tre processi figli saranno i seguenti:

- il processo **Sampler**, leggendo dal file `/proc/stat`, campiona i sample interi (**user**, **system**, **idle**) richiesti e li invia con messaggi distinti di tipo **raw** al figlio **Analyzer**; ogni sample è preso a distanza di 1 secondo l'uno dall'altro;
- il processo **Analyzer** considera le differenze tra i vari sample grezzi ricevuti ed estrapola le percentuali di utilizzo del comparto CPU in modalità utente e modalità kernel nell'ultimo secondo; queste informazioni vengono man mano inviate al figlio **Plotter** attraverso messaggi di tipo **delta**;

- il processo **Plotter** visualizza sullo standard output i dati di tipo **delta** ricevuti (una riga per ogni coppia) con un semplice grafico su 60 colonne ed in termini percentuali (vedi esempio a seguire).

La fine delle sequenze di messaggi (EOF) dovrà essere opportunamente segnalata in modo che tutti i processi si arrestino spontaneamente alla fine dei lavori. Tutte le strutture persistenti di IPC dovranno essere correttamente rilasciate in uscita.

Note:

- nel campionare dal file speciale **/proc/stat** è necessario aprire/chiedere lo stesso per avere dati sempre aggiornati;
- notare che, se i sample campionati sono in numero **n**, i dati delta estrapolati saranno ovviamente **n-1**.

Tempo: 2 ore e 30 minuti

Un esempio di esecuzione potrebbe essere il seguente:

```

$ cpu-monitor 20

##### s: 7.65% u:31.66%
##### s: 5.45% u:21.56%
##### s: 5.91% u:76.86%
##### s: 4.50% u:95.50%
##### s: 2.01% u:97.99%
##### s: 4.31% u:95.69%
##### s: 7.32% u:46.21%
##### s: 4.30% u:44.81%
##### s: 4.57% u:95.43%
##### s: 5.79% u:73.80%
##### s: 4.07% u:20.10%
##### s: 6.91% u:23.94%
##### s: 4.63% u:23.14%
##### s: 5.51% u:23.88%
##### s:10.13% u:32.00%
##### s: 6.60% u:24.87%
##### s: 5.29% u:19.90%
##### s: 7.65% u:25.59%
##### s: 7.71% u:37.02%

```

Ricordarsi di inserire i propri dati (nome, cognome, matricola) nei commenti preliminari del codice sorgente.

Verrà valutata anche l'efficienza computazionale delle soluzioni algoritmiche utilizzate.

Per inviare il proprio elaborato sul server è necessario utilizzare il comando **exam-box-sync**. Verrà richiesta la password associata al proprio account e verrà data una conferma all'avvenuto caricamento. E' possibile, e fortemente consigliato, inviare il proprio elaborato più volte e periodicamente come copia di riserva (l'ambiente di lavoro degli esami risiede in memoria RAM e è pertanto di tipo non-persistente).