

IN010 - TME 2

Danaël Carbonneau

3 février 2023

1 Statistiques d'une commande shell

1.1 Commande `time sleep 5`

FIGURE 1 – `time sleep 5`

```
real    0m5,002s
user    0m0,002s
sys     0m0,000s
```

On constate que le temps réel est de 5,002s, mais que le temps compté en mode système est nul (selon les statistiques), et le temps en mode user de 0,002s.

1.2 `loopcpu.c`

Pour des raisons matérielles, on a fait dans `loopcpu.c` une boucle sur 5×10^8 et non 5×10^9

FIGURE 2 – `time ./loopcpu`

```
real    0m1,122s
user    0m1,119s
sys     0m0,000s
```

On constate que la commande `time` retient surtout le temps passé en mode user et note 0s en mode système. Cela peut s'expliquer par le fait que le programme ne fait aucun appel système et que le processus passe donc très rarement dans ce mode, le temps est donc peu capturé par la commande `time`.

1.3 `loopsys.c`

On constate qu'en faisant beaucoup d'appels système, `time` nous permet désormais de mesurer un temps système utilisé de manière significative.

FIGURE 3 – `time ./loopsys`

```
real    0m5,706s
user    0m2,039s
sys      0m3,665s
```

FIGURE 4 – `./mytimes "sleep 5" "sleep 10"`

```
Le temps mis par l'exécution de la commande sleep 3 est de 3.00 secondes
Le temps mis par l'exécution de la commande sleep 5 est de 5.00 secondes
```

2 Calcul du temps en utilisant `gettimeofday`

3 Calcul de statistiques

FIGURE 5 – `./mytimes "sleep 5" ./loopcpu ./loopsys`

```
Statistiques de sleep 5 :
Temps total : 5.000000 s
Temps utilisateur : 0.000000 s
Temps systeme : 0.000000 s
Temps utilisateur fils : 0.000000 s
Temps systeme fils : 0.000000 s

Statistiques de ./loopcpu :
Temps total : 1.090000 s
Temps utilisateur : 0.000000 s
Temps systeme : 0.000000 s
Temps utilisateur fils : 1.090000 s
Temps systeme fils : 0.000000 s

Statistiques de ./loopsys :
Temps total : 5.660000 s
Temps utilisateur : 0.000000 s
Temps systeme : 0.000000 s
Temps utilisateur fils : 3.180000 s
Temps systeme fils : 3.560000 s
```

4 Changement de priorité

4.1 Commande ps -l

La priorité affichée est de 80.

FIGURE 6 – ps -l

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	85631	82052	0	80	0	-	3474	do_wai	pts/1	00:00:00	bash
4	R	1000	88774	85631	0	80	0	-	3514	-	pts/1	00:00:00	ps

4.2 Commande nice -19 ps -l

FIGURE 7 – nice -19 ps -l

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	85631	82052	0	80	0	-	3474	do_wai	pts/1	00:00:00	bash
4	R	1000	88782	85631	0	99	19	-	3514	-	pts/1	00:00:00	ps

Elle a désormais une priorité de 99

4.3 Priorités et lancement parallèle

La machine utilisée possède 4 cœurs, on lancera donc 5 processus en parallèle de loopcpu.

FIGURE 8 – `./mytimes ./loopcpu & ./mytimes ./loopcpu & ./mytimes ./loopcpu & ./mytimes ./loopcpu & ./mytimes "nice -19 ./loopcpu -1"`

```
[1] 89918
[2] 89919
[3] 89920
[4] 89921

Statistiques de ./loopcpu :
Temps total : 1.130000 s
Temps utilisateur : 0.000000 s
Temps systeme : 0.000000 s
Temps utilisateur fils : 1.110000 s
Temps systeme fils : 0.000000 s

Statistiques de ./loopcpu :
Temps total : 1.130000 s
Temps utilisateur : 0.000000 s
Temps systeme : 0.000000 s
Temps utilisateur fils : 1.110000 s
Temps systeme fils : 0.000000 s

Statistiques de ./loopcpu :
Temps total : 1.140000 s
Temps utilisateur : 0.000000 s
Temps systeme : 0.000000 s
Temps utilisateur fils : 1.110000 s
Temps systeme fils : 0.000000 s

Statistiques de ./loopcpu :
Temps total : 1.140000 s
Temps utilisateur : 0.000000 s
Temps systeme : 0.000000 s
Temps utilisateur fils : 1.110000 s
Temps systeme fils : 0.000000 s

Statistiques de nice -19 ./loopcpu -1 :
Temps total : 2.180000 s
Temps utilisateur : 0.000000 s
Temps systeme : 0.000000 s
Temps utilisateur fils : 1.080000 s
Temps systeme fils : 0.000000 s
[1] Fini ./mytimes ./loopcpu
[2] Fini ./mytimes ./loopcpu
[3]- Fini ./mytimes ./loopcpu
[4]+ Fini ./mytimes ./loopcpu
```

Avec 5 exécutions en parallèle, on voit bien que c'est le processus dont on a baissé la priorité qui a le plus grand temps total passé en machine : il a attendu qu'un des cœurs se libère.