



UNIX

Cours 6 : Gestion des processus

Filipe Vasconcelos¹

¹ESME, Lille, filipe.vasconcelo@esme.fr

- 1 Afficher les processus**
- 2 Lancer une tâche**
- 3 Arrêter les processus**



AAV5

Analyser et explorer l'arborescence d'un système de fichiers d'un OS du type Unix à l'aide des commandes de base, administrer les droits d'accès des répertoires et fichiers de l'arborescence et gérer les processus.



① Afficher les processus

② Lancer une tâche

③ Arrêter les processus



Un processus correspond à l'exécution de n'importe quel programme, commande ou script. Si un processus 2 a été lancé par un processus 1, on l'appelle **processus fils**. Le processus 1 est appelé **processus parent**. Chaque processus est identifié par un nombre **PID** unique. Pour afficher tous les processus lancés par l'utilisateur courant on lance la commande ps.

```
ps
```

PID	TTY	TIME	CMD
10038	pts/0	00:00:01	bash
10364	pts/0	00:00:00	ps

Tous les processus (même ceux lancés par root)

```
ps -A
```

PID	TTY	TIME	CMD
1	?	00:00:31	init
10036	?	00:00:00	init
10037	?	00:00:02	init
10038	pts/0	00:00:02	bash
10666	pts/0	00:00:00	ps

Pour obtenir une arborescence des processus

```
bash      # pour lancer un processus bash
ps -elH
```

\$ ps -elH	F S	UID	PID	PPID	C PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4 S 0 1 0 0 80 0 - 380 - ? 00:00:31 init												
5 S 0 10036 1 0 80 0 - 293 - ? 00:00:00 init												
1 S 0 10037 10036 0 80 0 - 293 - ? 00:00:02 init												
4 S 1000 10038 10037 0 80 0 - 2730 de mai pts/0 00:00:04 bash												

① Afficher les processus

② Lancer une tâche

③ Arrêter les processus



Lancer une tâche

Pour lancer une tâche il suffit de lancer le script dans le terminal.

```
./simulation
```

```
iter : 0      value = 6438    Elapsed Time : 0      (s)
iter : 1      value = 7679    Elapsed Time : 3      (s)
iter : 2      value = 24387   Elapsed Time : 6      (s)
iter : 3      value = 25687   Elapsed Time : 7      (s)
iter : 4      value = 13268   Elapsed Time : 8      (s)
iter : 5      value = 20162   Elapsed Time : 10     (s)
```

Tâche au premier plan ou en arrière plan

Si on lance une tâche et que l'on souhaite garder la main sur le terminal, on dit que l'on envoie le **processus en arrière plan** ou **en tâche de fond**. Pour récupérer la main, on peut arrêter momentanément le processus (Ctrl+Z)

```
./simulation > log  
Ctrl+Z
```

```
^Z  
[1]+  Stopped                 ./simulation > log
```

Pour placer le processus en arrière plan, on utilise la commande **bg (background)**

```
bg 1
```

```
[1]+ ./simulation > log &
```

Placer en arrière plan directement

Pour placer un processus directement en arrière plan on ajoute & à la commande

```
./simulation > log &
```

```
[1] 12264
```

Si on lance plusieurs :

```
./simulation > log1 &
./simulation > log2 &
./simulation > log3 &
```

```
[1] 12725
[2] 12727
[3] 12730
```

Pour reprendre un processus au premier plan

Pour placer le processus en arrière plan, on utilise la commande fg (**foreground**)

```
fg 1
```

```
./simulation > log1
```

Commande nohup

nohup est une commande permettant de lancer un processus qui restera actif même après la déconnexion de l'utilisateur l'ayant initiée. Combiné à & qui permet le lancement en arrière-plan, nohup permet donc de créer des processus s'exécutant de manière transparente sans être dépendants de l'utilisateur. Utilisation simple

```
nohup <commande> &
```

Si l'on ne souhaite pas rediriger la sortie standard vers le fichier nohup.out.

```
nohup <commande> > /dev/null &
```



- ① Afficher les processus
- ② Lancer une tâche
- ③ Arrêter les processus



Commande kill

La commande kill envoie un signal d'arrêt à un processus. Il existe un grand nombre de signaux (liste des signaux kill -l).

```
kill -l
```

1) SIGHUP	2) SIGINT	3) SIGQUIT	...
9) SIGKILL	...	15) SIGTERM	...
19) SIGSTOP	...		



Commande top

La commande top est un outil avancé pour la gestion des processus. L'entête permet d'avoir une vue globale de l'état de santé de la machine physique ou virtuelle.

top

```
top - 13:16:38 up 1 day, 17:36, 0 users, load average: 0.08, 0.03, 0.01
Tasks: 11 total, 1 running, 10 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 6.9 sy, 0.0 ni, 93.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 6252.0 total, 5732.9 free, 98.6 used, 420.6 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 5932.4 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4097	filipe	20	0	8752	3728	3264	S	0.3	0.1	0:00.06	simulation
1	root	20	0	1060	576	468	S	0.0	0.0	0:26.82	init
487	root	20	0	924	108	20	S	0.0	0.0	0:15.40	init
2095	root	20	0	1252	436	20	S	0.0	0.0	0:10.16	init
3444	filipe	20	0	64196	24408	7288	S	0.0	0.4	0:00.96	mupdf-x11
3613	filipe	20	0	62620	23344	7672	S	0.0	0.4	0:04.40	mupdf-x11
4027	root	20	0	1060	244	20	S	0.0	0.0	0:00.01	init
4028	root	20	0	1060	244	20	S	0.0	0.0	0:00.12	init
4029	filipe	20	0	11144	6328	3492	S	0.0	0.1	0:00.59	bash

