Introduction aux systèmes d'exploitation

D.Touazi Faycal

Maitre de conférences

Université M'hamed Bougara – Boumerdes

f.touazi@univ-boumerdes.dz

Les processus

f.touazi@univ-boumerdes.dz

C'est quoi un processus?

Un **processus** est **un programme** en cours d'exécution

Exemples:

- un terminal lancé est un processus
- une commande exécuté est un processus
- un répertoire ouvert est un processus
- un fichier ouvert est un processus
- navigateur ouvert Firefox ou chrome est un processus
- etc

Comment peut-on identifier un processus?

Chaque processus est identifié par :

- une identité PID (Process ID)
- l'identité de son parent PPID (Parent Process ID)
- l'utilisateur
- groupe
- le répertoire courant
- priorité
- etc

Comment peut-on avoir des informations sur les processus?

Il y a plusieurs commandes pour d'avoir des informations sur les processus:

- la commande **ps**
- la commande **pstree**
- la commande **top**
- la commande **htop**
- la commande **pgrep**
- **●●● etc**

La commande ps (process status) :

- **$ ps**

Permet d'affiche **les processus en cours lancés par l'utilisateur** et depuis **la console actuelle**

Exemple :

```
user1@PC:~$  ps
   PID  TTY          TIME  CMD
  5493  pts/2    00:00:00 bash
  5501  pts/2    00:00:00 ps
```

- **PID** Process ID, numéro du processus
- **TTY** nom du terminal depuis lequel le processus a été lancé
- **TIME** durée de traitement du processus
- **CMD** Commande exécutée

La commande ps (process status) :

- **$ ps -f**

Le paramètre -f pour avoir **plus d’informations**

Exemple :

```
user1@PC:~$ ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
user1       5492  5491  0 11:29 pts/2    00:00:00 bash
user1       5493  5492  0 11:40 pts/2    00:00:00 ps -f
```

- **UID** User ID, nom de l'utilisateur
- **PPID** Parent Process ID, numéro du processus père
- **C** facteur de priorité, plus la valeur est grande plus la priorité est élevée
- **STIME** heure de lancement du processus

La commande ps (process status) :

$ ps -ef

Le paramètre -e donne des informations sur **tous les processus en cours**

Exemple :

```
user1@PC:~$  ps -ef | head -n        6
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 10:20 ?        00:00:06 /sbin/init splash
root           2       0  0 10:20 ?        00:00:00 [kthreadd]
root           4       2  0 10:20 ?        00:00:00 [kworker/0:0H]
root           6       2  0 10:20 ?        00:00:00 [mm_percpu_wq]
root           7       2  0 10:20 ?        00:00:00 [ksoftirqd/0]
```

Quelques options intéressants :

- **-u** permet de préciser une liste d'un ou plusieurs utilisateurs séparés par une virgule
- **-g** pour préciser les groupes
- **-t** pour préciser les terminaux
- **-p** pour préciser les PID
- **-l** propose plus d'informations techniques

Exemple (-u):

```
user1@PC:~$  ps  -u   user1
   PID  TTY          TIME CMD
  8938  pts/0    00:00:00   bash
  9140  pts/0    00:00:00   vi
  9145  pts/0    00:00:00   nano
  9819  pts/0    00:00:00   lynx
  9822  pts/0    00:00:00   ps
```

Exemple (-l):

```
user1@PC:~$    ps   -l
F  S     UID      PID          C  PRI   NI  ADDR  SZ          TTY          TIME  CMD
   PPID                                                 WCHAN
4  S    1001    9885     9884  0    80    0  -     7601  wait        pts/0    00:00:00  bash
0  T    1001    9899     9885  0    80    0  -     7745  signal      pts/0    00:00:00  vi
0  T    1001    9902     9885  0    80    0  -     5469  signal      pts/0    00:00:00  nano
0  T    1001    9903     9885  0    80    0  -    15253  signal      pts/0    00:00:00  lynx
0  R    1001    9910     9885  0    80    0  -     9027  -           pts/0    00:00:00  ps
```

PRI Priorité du processus

La commande pstree :

- **$ pstree**

La commande **pstree** donne une bonne illustration de **la hiérarchie** des processus

Exemple :

```
user1@PC:~$ pstree | head -n 10
systemd-+-ModemManager---2*[{ModemManager}]
        |-NetworkManager-+-dhclient
        |                |-dnsmasq
        |                `-2*[{NetworkManager}]
        |-accounts-daemon---2*[{accounts-daemon}]
        |-acpid
        |-2*[agetty]
        |-at-spi-bus-laun-+-dbus-daemon
        |                 `-3*[{at-spi-bus-laun}]
        |-at-spi2-registr---2*[{at-spi2-registr}]
```

La commande pstree :

- **$ pstree -p**

Le paramètre **-p** permet pour afficher les PID

Exemple :

```
user1@PC:~$ pstree -p | head -n 8
systemd(1)-+-ModemManager(809)-+-{ModemManager}(850)
           |                   `-{ModemManager}(853)
           |-NetworkManager(812)-+-dhclient(7814)
           |                     |-dnsmasq(2566)
           |                     |-{NetworkManager}(864)
           |                     `-{NetworkManager}(866)
           |-accounts-daemon(750)-+-{accounts-daemon}(782)
           |                      `-{accounts-daemon}(836)
```

La commande pstree :

- **$ pstree -u**

Le paramètre **-u** permet pour afficher les utilisateurs

Exemple :

```
user1@PC:~$ pstree -u | head -10
systemd-+-ModemManager---2*[{ModemManager}]
        |-NetworkManager-+-dhclient
        |                |-dnsmasq(nobody)
        |                `-2*[{NetworkManager}]
        |-accounts-daemon---2*[{accounts-daemon}]
        |-acpid
        |-2*[agetty]
        |-at-spi-bus-laun(user1)-+-dbus-daemon
        |                        `-3*[{at-spi-bus-laun}]
        |-at-spi2-registr(user1)---2*[{at-spi2-registr}]
```

La commande top :

- **$ top**

permet d'afficher **les informations** des processus en **temps réel**. **Le rafraîchissement** des informations se fait à chaque **03 seconds**

Exemple :

```
top - 17:23:09 up 2 days,  2:11,  1 user,  load average: 0,10, 0,17, 0,22
Tâches: 1468 total,   2 en cours, 1407 en veille,   0 arrêté,   1 zombie
%Cpu(s):  3,8 ut,  1,2 sy,  0,0 ni, 95,0 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem : 43196568 total, 26142844 libr, 14617496 util,  2436228 tamp/cache
KiB Éch: 65535996 total, 65535996 libr,        0 util. 27794572 dispo Mem
```

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
3492	1cpig3b+	20	0	3399508	358556	89620	S	5,7	0,8	34:07.85	cinnamon
610	root	20	0	107776	13840	9616	S	4,4	0,0	132:59.49	plymouthd
9631	**guerrout**	**20**	**0**	**44704**	**5264**	**3252**	**R**	**4,4**	**0,0**	**0:00.51**	**top**
5355	1cpig3b7	20	0	3404632	400920	89712	S	1,3	0,9	46:14.62	cinnamon
6395	**1cpig3b4**	**20**	**0**	**3365964**	**435636**	**91156**	**R**	**1,3**	**1,0**	**56:36.17**	**cinnamon**
10149	1cpig1b2	20	0	3345332	384096	90408	S	1,3	0,9	48:42.86	cinnamon
10644	1cpig3b+	20	0	3282168	356252	90236	S	1,3	0,8	52:30.53	cinnamon
11205	1cpig3b+	20	0	3267288	368580	91040	S	1,3	0,9	45:35.40	cinnamon
19750	1cpig1b7	20	0	3339216	412792	90248	S	1,3	1,0	28:28.14	cinnamon
22901	1cpig1b3	20	0	3401196	374224	89764	S	1,3	0,9	32:54.96	cinnamon
1769	1cpig3b2	20	0	3397292	358072	89656	S	1,0	0,8	47:30.99	cinnamon
2303	sehad	20	0	2754684	228076	90344	S	1,0	0,5	38:19.15	cinnamon
2913	1cpig3b6	20	0	3371916	430000	91060	S	1,0	1,0	39:56.10	cinnamon
3215	1cpig3b+	20	0	960084	63152	35436	S	1,0	0,1	4:07.54	Xorg
3683	1cpig3b+	20	0	988640	60828	38756	S	1,0	0,1	2:56.72	cinnamon-s+
8369	guerrout	20	0	2819852	220612	89836	S	1,0	0,5	41:57.88	cinnamon
9657	1cpig3b1	20	0	3351252	398540	91928	S	1,0	0,9	45:32.29	cinnamon

La commande htop :

- **$ htop**

C’est une amélioration de **top,**

Exemple :

```
1  [||||||||                 23.6%]   5  [||||||                 16.6%]
2  [|||                       5.0%]   6  [|||                    8.8%]
3  [|||                       7.5%]   7  [|||                    6.0%]
4  [||||                     11.1%]   8  [|||||                 12.9%]
Mem[|||||||||||||||      14.2G/41.2G]   Tasks: 1347, 3735 thr; 1 running
Swp[                        0K/62.5G]   Load average: 0.28 0.25 0.22
                                        Uptime: 2 days, 02:20:07

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
    1 root       20   0  221M 10120  6644 S  0.0  0.0  1:31.72 /sbin/init splash
26206 1cpig1b11  20   0 1266M 51848 34248 S  0.0  0.1  0:04.08 ├─ evince /home/1
26245 1cpig1b11  20   0 1266M 51848 34248 S  0.0  0.1  0:00.17 │  ├─ evince /hom
26225 1cpig1b11  20   0 1266M 51848 34248 S  0.0  0.1  0:00.00 │  ├─ evince /hom
26209 1cpig1b11  20   0 1266M 51848 34248 S  0.0  0.1  0:00.00 │  ├─ evince /hom
26208 1cpig1b11  20   0 1266M 51848 34248 S  0.0  0.1  0:00.00 │  └─ evince /hom
25974 1cpig1b11  20   0  493M 12584 10852 S  0.0  0.0  0:00.02 ├─ /usr/lib/x86_6
25985 1cpig1b11  20   0  493M 12584 10852 S  0.0  0.0  0:00.00 │  ├─ /usr/lib/x8
25982 1cpig1b11  20   0  493M 12584 10852 S  0.0  0.0  0:00.00 │  └─ /usr/lib/x8
25886 1cpig1b11  20   0 1019M  7664  6284 S  0.0  0.0  0:03.79 ├─ /usr/bin/pulse
26028 1cpig1b11  20   0 1019M  7664  6284 S  0.0  0.0  0:03.65 │  └─ /usr/bin/pu
25863 1cpig1b11  20   0  278M  7820  6832 S  0.0  0.0  0:00.05 ├─ /usr/bin/gnome
25872 1cpig1b11  20   0  278M  7820  6832 S  0.0  0.0  0:00.00 │  ├─ /usr/bin/gn
F1Help  F2Setup F3SearchF4FilterF5SortedF6CollapF7Nice -F8Nice +F9Kill  F10Quit
```

Comment peut-on arrêt un processus?

La commande kill: envoyer un signal à un processus

- **$ kill -l**

-l : permet d'afficher la liste des signaux

- **$ kill -Num_signal PID [PID2...]**

-9 : Signal ne pouvant être ignoré, force le processus à finir **brutalement**

-15 : Signal envoyé par défaut par la commande kill, demande au processus de se terminer **normalement**

Exemple : (-9)

User1 @PC:~$ kill -9 7847

Comment peut-on arrêter ou détacher un processus lancé en terminal?

- La combinaison de touches **Ctrl + C** permet arrêter un processus lancé en terminal
- Il y a deux manières de détacher un processus lancé en terminal

-La combinaison de touches **Ctrl + Z** permet de détacher un processus lancé en terminal ; puis on doit exécuter la commande **bg** pour qu'il continue à s'exécuter

-Ou on doit rajouter **ET Commercial « & »** à la fin de la commande

Comment modifier la priorité d'un processus?

- Tous les processus ont une priorité initiale identique au lancement égale à **0**
- La valeur de priorité doit être comprise entre **-20 et 20**
- Plus la valeur est élevée et plus le traitement est ralenti
- **seul** l'utilisateur **root** peut **diminuer** la valeur de priorité
- **les autres utilisateurs** peuvent juste **augmenter** la valeur de priorité

La commande **nice** permet de **modifier** la priorité d'un processus **au démarrage du processus**

- **$ nice [-valeur] commande [arguments]**

Attention : Une valeur **positive** causera une baisse de priorité seul

La commande **renice** comme **nice** mais elle permet de **modifier** la priorité d'un processus **après démarrage du processus**

- **$ renice [-n prio] [-p PID] [-g GID] [-u UID]**

-n : pour préciser la nouvelle valeur de priorité

-p : pour changer la priorité d'un processus en utilisant le PID

-u : pour changer la priorité de tous les processus d'un utilisateur

-g : pour changer la priorité de tous les processus d'un groupe d'utilisateur