Université Mohammed Premier Oujda
École Nationale des Sciences
Filières : GI / Niveau : GI4
Module : Administration des systèmes informatiques

Administration Des Systèmes Informatiques

Rapport Tp10: Gestion des processus sous un système Linux

Réalisé par :

Safae BOUNIETE

Année Universitaire : 2017/2018

Etape 1 : Test des commandes de Surveillance des ressources système

1. Se connecter en tant que «root» sur une console texte.
2. Comment peut-on obtenir à l'aide de ps des informations sur la priorité des processus en cours ? Ces informations sur la priorité peuvent être retrouvées aussi avec top mais en temps réel.

```
root@debian:/home/ensao# ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0  3043  3038  0  80   0 -  1562 wait   pts/0    00:00:00 su
4 S     0  3044  3043  0  80   0 -  1419 wait   pts/0    00:00:00 bash
0 R     0  3079  3044  0  80   0 -  1848 -      pts/0    00:00:00 ps
```

```
top - 10:39:39 up  1:14,  1 user,  load average: 0,20, 0,32, 0,36
Tasks: 151 total,   1 running, 150 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3,8 us, 20,0 sy,  0,0 ni, 74,8 id,  0,7 wa,  0,0 hi,  0,7 si,  0,0 st
KiB Mem :  1029060 total,   288248 free,   537016 used,   203796 buff/cache
KiB Swap:  1046524 total,   976928 free,    69596 used.   302928 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 2162 ensao     20   0  913964 151748  39420 S  8,5 14,7   5:51.31 gnome-shell
 2033 ensao     20   0  141384  51452  35092 S  7,5  5,0   4:09.89 Xorg
 2640 ensao     20   0 1171784 217132  82388 S  3,3 21,1   1:42.90 firefox-esr
 2129 ensao     20   0   18000   1076   1076 S  1,0  0,1   0:41.99 VBoxClient
 2682 ensao     20   0  603468 121420  56204 S  1,0 11,8   0:32.61 Web Content
 3032 ensao     20   0   91668  28528  23204 S  1,0  2,8   0:03.87 gnome-terminal-
  145 root      20   0       0      0      0 S  0,7  0,0   0:01.88 jbd2/sda1-8
 1439 avahi     20   0    6384   3016   2896 S  0,7  0,3   0:14.20 avahi-daemon
 3080 root      20   0    8108   3756   3228 R  0,7  0,4   0:00.09 top
    4 root      20   0       0      0      0 S  0,3  0,0   0:05.70 kworker/0:0
   11 root      rt   0       0      0      0 S  0,3  0,0   0:00.31 watchdog/0
 1950 root      20   0   28772   1900   1792 S  0,3  0,2   0:10.94 VBoxService
    1 root      20   0    3756   1492   1468 S  0,0  0,1   0:02.00 init
    2 root      20   0       0      0      0 S  0,0  0,0   0:00.01 kthreadd
    3 root      20   0       0      0      0 S  0,0  0,0   0:08.72 ksoftirqd/0
    5 root       0 -20       0      0      0 S  0,0  0,0   0:00.00 kworker/0:0H
    7 root      20   0       0      0      0 S  0,0  0,0   0:09.36 rcu_sched
    8 root      20   0       0      0      0 S  0,0  0,0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S  0,0  0,0   0:00.00 migration/0
   10 root       0 -20       0      0      0 S  0,0  0,0   0:00.00 lru-add-drain
   12 root      20   0       0      0      0 S  0,0  0,0   0:00.00 cpuhp/0
   13 root      20   0       0      0      0 S  0,0  0,0   0:00.00 kdevtmpfs
```

Avec ps, on obtient des informations statiques sur les processus, avec top on obtient des informations en temps réelle.

3. Lancer les commandes suivantes :
   - (a) CMD1 : nice sleep 240 &
   - (b) CMD2 : ps -l.

```
root@debian:/home/ensao# nice sleep 240 &
[1] 3081
root@debian:/home/ensao# ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0  3043  3038  0  80   0 -  1562 wait   pts/0    00:00:00 su
4 S     0  3044  3043  0  80   0 -  1419 wait   pts/0    00:00:00 bash
0 S     0  3081  3044  0  90  10 -   971 hrtime pts/0    00:00:00 sleep
0 R     0  3082  3044  0  80   0 -  1848 -      pts/0    00:00:00 ps
```

Le processus 3081 va être endormi pendant 240s en mode arrière-plan

4. Quelle est la valeur affichée dans la colonne NI ? Que-remarquez-vous ?
   La valeur NI=10 (la valeur de gentillesse, plus grand est le plus iportant)
5. Quelle est la valeur affichée dans la colonne PRI ? Que-remarquez-vous ?
   La Valeur PRI=90 (plus le nombre est grand, le processus est moins prioritaire)
6. Lancer les commandes suivantes :
   - (a) CMD3 : nice -n 19 sleep 240 &
   - (b) CMD4 : ps -l.

```
root@debian:/home/ensao# nice -n 19 sleep 240 &
[2] 3083
root@debian:/home/ensao# ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0  3043  3038  0  80   0 -  1562 wait   pts/0    00:00:00 su
4 S     0  3044  3043  0  80   0 -  1419 wait   pts/0    00:00:00 bash
0 S     0  3081  3044  0  90  10 -   971 hrtime pts/0    00:00:00 sleep
0 S     0  3083  3044  0  99  19 -   971 hrtime pts/0    00:00:00 sleep
0 R     0  3084  3044  2  80   0 -  1848 -      pts/0    00:00:00 ps
```

   Ici on va lancer un processus de valeur de gentillesse 19 qui va etre endormi pendant 240s en arrière-plan
7. Quelle est la valeur affichée dans la colonne NI ? Que-remarquez-vous ?
   La valeur NI=19
8. Quelle est la valeur affichée dans la colonne PRI ? Que-remarquez-vous ?
   La Valeur PRI=99
9. Lancer les commandes suivantes :
   - (a) CMD5 : nice -n -19 sleep 240 &
   - (b) CMD6 : ps -l.

```
root@debian:/home/ensao# nice -n -19 sleep 240 &
[3] 3085
root@debian:/home/ensao# ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0  3043  3038  0  80   0 -  1562 wait   pts/0    00:00:00 su
4 S     0  3044  3043  0  80   0 -  1419 wait   pts/0    00:00:00 bash
0 S     0  3081  3044  0  90  10 -   971 hrtime pts/0    00:00:00 sleep
0 S     0  3083  3044  0  99  19 -   971 hrtime pts/0    00:00:00 sleep
4 S     0  3085  3044  0  61 -19 -   971 hrtime pts/0    00:00:00 sleep
0 R     0  3086  3044  0  80   0 -  1848 -      pts/0    00:00:00 ps
```

10. Quelle est la valeur affichée dans la colonne NI ? Que-remarquez-vous ?
    La valeur NI=-19
11. Quelle est la valeur affichée dans la colonne PRI ? Que-remarquez-vous ?
    La Valeur PRI=61
12. La commande renice permet de changer la priorité d'un processus au cours de son exécution. Donner un exemple sur le modèle de la question précédente sur le processus de la commande CMD5, pour montrer l'utilisation de renice.

```
root@debian:/home/ensao# ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0  3043  3038  0  80   0 -  1562 wait   pts/0    00:00:00 su
4 S     0  3044  3043  0  80   0 -  1419 wait   pts/0    00:00:00 bash
0 S     0  3081  3044  0  90  10 -   971 hrtime pts/0    00:00:00 sleep
0 S     0  3083  3044  0  99  19 -   971 hrtime pts/0    00:00:00 sleep
4 S     0  3085  3044  0  61 -19 -   971 hrtime pts/0    00:00:00 sleep
0 R     0  3086  3044  0  80   0 -  1848 -      pts/0    00:00:00 ps
root@debian:/home/ensao# renice -n -17 sleep -p 3085
renice: valeur process ID erronée : sleep
3085 (process ID) priorité précédente -19, nouvelle priorité -17
[1]   Fini                    nice sleep 240
[2]-  Fini                    nice -n 19 sleep 240
root@debian:/home/ensao# ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0  3043  3038  0  80   0 -  1562 wait   pts/0    00:00:00 su
4 S     0  3044  3043  0  80   0 -  1529 wait   pts/0    00:00:00 bash
4 S     0  3085  3044  0  63 -17 -   971 hrtime pts/0    00:00:00 sleep
0 R     0  3092  3044  0  80   0 -  1848 -      pts/0    00:00:00 ps
```

Etape 2: Modes d'exécution des processus

1. Se connecter en tant que «root» sur une console texte.
2. Récupérer le programme «memoire.c» à partir le site.
3. Compiler le programme. (ne pas tenir compte des messages de warning s'il y en avait) «gcc memoire.c -o memoire.exe». L'exécutable généré s'appellera «memoire.exe».
4. Attribuer le droit d'éxecution pour le «memoire.exe». (Utilisation : la commande chmod).
5. Lancer le sous le nom «./memoire.exe» avec un paramètre entier inférieur à 10. Observez ce que fait le programme.

```
root@debian:/home/ensao# ./memoire.exe 1
Allocation de 1 Mo en memoire... OK
Initialisation des 1 Mo en memoire... OK
Vous pouvez faire Ctrl-Z pendant les 30 secondes a venir.
^Z
[1]+  Stoppé                  ./memoire.exe 1
```

On a alloué 1M de la mémoire.

6. Relancer le programme maintenant en l'interrompant avant sa terminaison par «Ctrl-C». Qu'observez-vous ?

```
root@debian:/home/ensao# ./memoire.exe 5
Allocation de 5 Mo en memoire... OK
Initialisation des 5 Mo en memoire... OK
Vous pouvez faire Ctrl-Z pendant les 30 secondes a venir.
^C
Reception d'une interruption Ctrl-C (2, 2). On s'arrete.
Au revoir
```

«Ctrl-C» : c'est pour tuer le programme avec le signal SIGINT, le programme se termine et affiche « Au revoir »

7. Relancer le programme maintenant en lui donnant 200 comme paramètre et interrompez-le avant sa terminaison par «Ctrl-Z». Qu'observez-vous ?

```
root@debian:/home/ensao# ./memoire.exe 200
Allocation de 200 Mo en memoire... OK
^Z
[5]+  Stoppé                 ./memoire.exe 200
```

«Ctrl-Z» : est utilisé pour suspendre un processus en lui envoyant le signal SIGSTOP

8. Refaites cette opération une, deux, trois, quatre, etc. fois de plus jusqu'à... ce que l'on ne puisse plus.

```
root@debian:/home/ensao# ./memoire.exe 200
Allocation de 200 Mo en memoire... OK
^Z
[20]+  Stoppé                 ./memoire.exe 200
root@debian:/home/ensao# ./memoire.exe 200
Allocation de 200 Mo en memoire... OK
^Z
[21]+  Stoppé                 ./memoire.exe 200
root@debian:/home/ensao# ./memoire.exe 200
Allocation de 200 Mo en memoire... OK
^Z
[22]+  Stoppé                 ./memoire.exe 200
root@debian:/home/ensao# ./memoire.exe 200
malloc() error
```

Maintenant la mémoire est saturée, on ne peut plus allouer de la mémoire

9. Pour se sortir de tous ses programmes qui ont saturé la machine, faites «jobs». Qu'observez-vous ?

```
Allocation de 200 Mo en memoire... root@debian:/home/ensao# jobs
[4]   Stoppé                 ./memoire.exe 5
[5]   Stoppé                 ./memoire.exe 200
[6]   Stoppé                 ./memoire.exe 200
[7]   Stoppé                 ./memoire.exe 200
[8]   Stoppé                 ./memoire.exe 200
[9]   Stoppé                 ./memoire.exe 200
[10]   Stoppé                 ./memoire.exe 200
[11]   Stoppé                 ./memoire.exe 200
[12]   Stoppé                 ./memoire.exe 200
[13]   Stoppé                 ./memoire.exe 200
[14]   Stoppé                 ./memoire.exe 200
[15]   Stoppé                 ./memoire.exe 200
[16]   Stoppé                 ./memoire.exe 200
[17]   Stoppé                 ./memoire.exe 200
[18]   Stoppé                 ./memoire.exe 200
[19]   Stoppé                 ./memoire.exe 200
[20]   Stoppé                 ./memoire.exe 200
[21]-  Stoppé                 ./memoire.exe 200
[22]+  Stoppé                 ./memoire.exe 200
```

10. Tuez tous les jobs qui sont suspendus. Comment procédez-vous ?

…

```
root@debian:/home/ensao# jobs
[20]   Complété               ./memoire.exe 200
[21]-  Stoppé                  ./memoire.exe 200
[22]+  Stoppé                  ./memoire.exe 200
root@debian:/home/ensao# kill %21

[21]-  Stoppé                  ./memoire.exe 200
root@debian:/home/ensao# kill %22
[21]-  Complété               ./memoire.exe 200

[22]+  Stoppé                  ./memoire.exe 200
root@debian:/home/ensao# jobs
[22]+  Complété               ./memoire.exe 200
root@debian:/home/ensao# jobs
```

11. Dupliquez le programme compilé précédemment en lui donnant un autre nom. Par exemple «memory.exe».

```
root@debian:/home/ensao# gcc memoire.c -o memory.exe
memoire.c: In function 'main':
memoire.c:27:18: warning: passing argument 2 of 'signal' from incompatible pointer t
ype [-Wincompatible-pointer-types]
   signal(SIGINT, interruption);
                  ^~~~~~~~~~~~
In file included from memoire.c:10:0:
/usr/include/signal.h:102:23: note: expected '__sighandler_t {aka void (*)(int)}' bu
t argument is of type 'void (*)(int,  int,  struct sigcontext *, char *)'
 extern __sighandler_t signal (int __sig, __sighandler_t __handler)
                       ^~~~~~
memoire.c:50:9: warning: implicit declaration of function 'sleep' [-Wimplicit-functi
on-declaration]
   (void)sleep(30);
         ^~~~~
```

12. Se connecter en tant que «root» sur deux consoles texte.
13. Dans la première fenêtre, lancez «memoire.exe 1» et suspendez-le par «Ctrl-Z».

```
root@debian:/home/ensao# ./memoire.exe 1
Allocation de 1 Mo en memoire... OK
Initialisation des 1 Mo en memoire... OK
Vous pouvez faire Ctrl-Z pendant les 30 secondes a venir.
^Z
[1]+  Stoppé                 ./memoire.exe 1
```

14. Dans la deuxième fenêtre, lancez «./memory.exe 1» et suspendez-le par «Ctrl-Z».

```
root@debian:/home/ensao# ./memory.exe 1
Allocation de 1 Mo en memoire... OK
Initialisation des 1 Mo en memoire... OK
Vous pouvez faire Ctrl-Z pendant les 30 secondes a venir.
^Z
[1]+  Stoppé                 ./memory.exe 1
```

15. Lancez encore un autre «./memory.exe 1» et suspendez-le aussi par «Ctrl-Z».

```
root@debian:/home/ensao# ./memory.exe 1
Allocation de 1 Mo en memoire... OK
Initialisation des 1 Mo en memoire... OK
Vous pouvez faire Ctrl-Z pendant les 30 secondes a venir.
^Z
[2]+  Stoppé                 ./memory.exe 1
```

16. Faites «jobs» dans chacune des deux consoles. Qu'observez-vous. Qu'en déduisez-vous sur ce que renvoi «jobs» ?

```
root@debian:/home/ensao# jobs
[1]-  Stoppé                 ./memoire.exe 1
[2]+  Stoppé                 ./memory.exe 1
```

```
root@debian:/home/ensao# jobs
[1]+  Stoppé                 ./memory.exe 1
```

La commande jobs permet d'afficher la liste des travaux pris en charge par l'interpréteur de commande sur lequel elle est lancée. Du coup on trouve que les processus lancer dans le même terminal.

17. Dans la console 1, donnez la commande pour tuer le job suspendu.

```
root@debian:/home/ensao# kill %1
[1]+  Complété                 ./memory.exe 1
root@debian:/home/ensao# jobs
```

18. Dans la console 2, donnez la commande pour tuer le job 1 suspendu. Donnez la commande pour remettre en premier plan, le job 2.

```
root@debian:/home/ensao# jobs
[1]-  Stoppé                   ./memoire.exe 1
[2]+  Stoppé                   ./memory.exe 1
root@debian:/home/ensao# kill %1
[1]-  Complété                 ./memoire.exe 1
root@debian:/home/ensao# kill %2
[2]+  Complété                 ./memory.exe 1
root@debian:/home/ensao# jobs
```

Étape 3: Exécution des processus en avant/arrière-plan

1. Se connecter en tant que «root» sur une console texte.
2. La commande «sleep» sert à attendre pendant un nombre de secondes spécifié. Par exemple, « sleep 5» attend 5 secondes. Cette commande va servir de base pour ces manipulations car c'est une commande qui permet de simuler l'exécution d'une longue tâche telle qu'une grosse compilation par exemple.
3. Lancez la commande «sleep 5». Que se passe-t-il ?

```
root@debian:/home/ensao# sleep 5
```

Le processus va être endormi pendant 5 seconde, on a pas la main pour exécuter des commandes pendant ces 5 secondes.

4. Lancez la commande «sleep 500» en arrière-plan.

```
root@debian:/home/ensao# sleep 5
root@debian:/home/ensao# sleep 500 &
[1] 3153
root@debian:/home/ensao# jobs
[1]+  En cours d'exécution   sleep 500 &
```

5. Vérifiez avec «jobs» que votre commande est toujours là.
6. Lancez la commande «sleep 5» en arrière-plan. Que se passe-t-il lorsqu'elle se termine ?

```
root@debian:/home/ensao# sleep 5 &
[2] 2677
```

7. Votre commande «sleep 500» est toujours active. Mettez-la en avant-plan. (Utilisation : la commande fg).

```
root@debian:/home/ensao# fg %1
sleep 500
```

8. Suspendez-la. Faites «jobs». Quel est son état ? Relancez-la en arrière-plan. (Utilisation : la commande bg).

```
root@debian:/home/ensao# sleep 500
^Z
[1]+  Stoppé                 sleep 500
root@debian:/home/ensao# jobs
[1]+  Stoppé                 sleep 500
root@debian:/home/ensao# bg %1
[1]+ sleep 500 &
root@debian:/home/ensao#
```

«sleep 500» est maintenant stoppé

9. Lancez une deuxième commande «sleep 100» en arrière-plan. Passez la première en avantplan. Suspendez-la. Suspendez la deuxième.

```
root@debian:/home/ensao# sleep 100 &
[2] 2714
root@debian:/home/ensao# fg %1
sleep 500
^Z
[1]+  Stoppé                 sleep 500
root@debian:/home/ensao# fg %2
sleep 100
^Z
[2]+  Stoppé                 sleep 100
```

10. Reprenez l'exécution de la première en avant-plan. Repassez la première en arrière-plan et reprenez la deuxième en arrière-plan. (Utilisation : les commandes bg et fg).

```
root@debian:/home/ensao# fg %1
sleep 500
^Z
[1]+  Stoppé                 sleep 500
root@debian:/home/ensao# bg %1
[1]+ sleep 500 &
root@debian:/home/ensao# bg %2
[2]+ sleep 100 &
```

11. Faites «ps» pour contrôler les processus actifs.

```
root@debian:/home/ensao# ps
  PID TTY          TIME CMD
 2708 pts/0    00:00:00 su
 2709 pts/0    00:00:00 bash
 2713 pts/0    00:00:00 sleep
 2716 pts/0    00:00:00 ps
[2]+  Fini                 sleep 100
```

13. Quelles sont les différences avec «jobs» ? Comment faire pour obtenir la liste de tous vos processus ?

On utilise la commande **Ps –aux.**