Nom et Prenom : CHAKIR Manale
Filiere : GI4
A.U : 2018/2019

Rapport Tp10: Gestion des processus sous un système Linux

Enoncé 1 : Modes d'exécution des processus :

Un processus est soit :

- **prêt** (suspendu par le système d'exploitation)
- **élu** (en exécution)
- **bloqué** (en attente d'un événement quelconque pour poursuivre)
- **zombie** (Terminé on exécution)

Tout d'abord on va compiler le programme 'mémoire.c' en utilisant la commande gcc mémoire.c –o mémoire.exe . Ensuite, on va executer ce programme en lui donnant 20 Mo en Ram , puis en interrome-le en cliquant **Ctrl+Z**.

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

➔ Le processus est maintenant en « background », et son execution a été arrété ;

➔ Pour tuer (arreter) le programme avant sa terminaison, en clique **Ctrl+C ;**

➔ **Alors la différence entre Ctrl+Z et Ctrl+C est : la première sert à mettre le programme en background en arrêtant son exécution, mais la deuxième sert a tuer le programme et arrêter son exécution même s'il n'est pas encore terminer .**

➔ **Pour visualiser tous les processus en utilise la commande jobs ;**

```
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
```

➔ Pour Tuer un job suspendu, on utilise la commande **kill** suivi par **%** et l'identifiant du job :

```
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
```

Enoncé 2 : Exécution des processus en avant/arrière-plan :

1- 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.

- Si on lance la commande ' sleep 5' , Le processus va être endormi pendant 5 seconde, on a pas la main pour exécuter des commandes pendant ces 5 secondes.
- Pour lancer la commande «sleep 500» en arrière-plan, en utilise '&' :

```
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 &
```

- Par la commande **jobs**, on verifie que la commande sleep 500 est en cours d'execution ; mais en arriere-plan .

```
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
```

- Pour rendre un processus qui est en arriere-plan tourne en avant-plan, en utilise la commande **fg** suivi par **%** suivit par le PID trouvé à l'aide de la commande **jobs ;**
- Pour rendre un processus tourne en arriere-plan, en utilise la commande **bg** suivi par **%** suivit par le PID trouvé à l'aide de la commande **jobs ;**
- Pour lancer un processus en arriere-plan (càd : le processus va s'executer , et rendre la
- et rendre la main pour taper d'autre commandes) en utilise **&**

```
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 &
```

Enoncé 3: Utilisation de la mémoire :

free est une commande permettant d'afficher des informations de disponibilité sur la mémoire vive du système.

Elle offre plusieurs options telles que :

- -b,-k,-m,-g donne la mémoire en KB, MB, ou GB
- -l montre des statistiques détailles sur les hauts et les bas de la mémoire
- -o utilise l'ancien format (sans la ligne -/+ cache/tampon)
- -t affiche la RAM totale + swap
- -s actualise à toutes les [delay] secondes
- -c actualise [count] fois
- -V affiche l'information sur la version et ferme

```
root@it-connect-serv01 ~ $ free
             total       used       free     shared    buffers     cached
Mem:        502008     495520       6488          0      63804     176952
-/+ buffers/cache:     254764     247244
Swap:       477180        456     476724
root@it-connect-serv01 ~ $
```