

## TD4: Les processus

Un processus est un programme ou un service chargé en mémoire vive, actif ou non, en premier plan ou en arrière plan.

Un processus possède un numéro unique que l'on appelle le PID (**P**rocess **I**dentifiant).

Au démarrage, le système charge le noyau Linux qui lance le premier processus **init** (PID n°1) qui lance lui-même d'autres processus.

Un processus descend d'un processus père identifié par le numéro PPID (**P**arent **P**ID) et appartient à un utilisateur. Il s'exécute avec les droits de l'utilisateur qui l'a lancé.

Les commandes *ps* et *ps tree* permettent d'obtenir des informations sur les processus en mémoire.

Aide sur l'utilisation de *ps*

- USER le propriétaire du processus.
- PID le numéro du processus
- PPID le numéro du père
- %CPU en % les ressources du microprocesseur utilisées par le processus.
- %MEM en % les ressources en mémoire vive utilisées par le processus.
- RSS mémoire utilisée en ko par le processus.
- TTY la console qui l'a lancé ou ? pour les services résidents(**daemons**)
- STAT Run Sleep Wait ...
- START l'heure à laquelle le processus a été lancé.

Utilisez *man ps* pour obtenir plus d'explications.

### Exercice 1:

Indiquez les informations données par la commande *ps* en fonction des options ci-dessous ?

*ps* :.....  
*ps -f* :.....  
*ps u* :.....  
*ps au* :.....  
*ps auf* :.....  
*ps x* :.....  
*ps tree* :.....

### Envoyer un signal à un processus

La commande « *kill* » permet d'envoyer un signal à un processus dont on connaît le PID

*kill PID* détruit le processus de numéro PID si c'est possible

*kill -9 PID* force la commande si la commande précédente n'aboutit pas

### Exercice 2 :

Connecter l'utilisateur INSSET dans la console tty1 (Alt+Ctrl+F1) et Luc dans la console tty2 (Alt+Ctrl+F2)

Luc lance la commande *yes* qui affiche y en permanence.

Quelle commande INSSET doit-il entrer pour afficher les processus de Luc ?

Réponse :.....

Quelle commande INSSET doit-il envoyer pour détruire le processus « yes » que Luc utilise ?

Réponse :.....

Vérifier.

Détruire le shell de Luc (le *bash* ou *sh de Luc*) puis basculer vers le terminal de Luc, Que s'est-il passé ?

Réponse :.....

### Exécution d'une commande en tâche de fond :

En faisant suivre une commande du symbole **&** le processus père reprend son exécution sans attendre la fin du processus fils qui continue alors en tâche de fond.

#### Exercice 3:

-Lancer la commande *yes>/dev/null* la sortie n'est plus affichée à l'écran, le programme s'exécute en premier plan, pour entrer une autre commande, il faut quitter avec **ctrl-c**.

-Lancer la commande *yes>/dev/null &* le programme s'exécute maintenant en arrière plan, vous pouvez entrer d'autres commandes sans arrêter « yes ». Lancer une deuxième fois *yes>/dev/null*.

-Vérifier avec *ps -l* que les deux programmes « yes » s'exécutent et noter le père de chacun des *yes*.

Réponse :.....

Détruire les deux processus « yes » et vérifier.

#### Exercice 4:

Créer un fichier script que l'on nommera *mon\_yes* pour lancer *yes>/dev/null*, il n'est pas utile ici d'utiliser *vi*, rediriger simplement la commande *echo* vers le fichier script :

*echo "yes>/dev/null" > mon\_yes*

- Positionner le droit en **exécution** pour le propriétaire du script *mon\_yes*
- Lancer *./mon\_yes &* en tâche de fond
- Examiner les processus avec *ps lf* pour établir la généalogie des processus
- Tuer le père de *yes*, *yes* survit-il ? Quel est son nouveau père?

#### Exercice 5: Arborescence des processus

- Afficher tous les processus avec *ps -auxf /less*
- *ps -aux | grep " R "* : pour afficher les lignes qui contiennent " R " (processus qui travaillent).
- A quoi sert la commande *top* ?

Réponses :.....

.....

.....

.....

L'administrateur peut utiliser la commande *kill* pour envoyer des signaux aux "démons" (relancer un serveur, l'arrêter...)

### **Exercice:**

- Entrer une commande pour afficher uniquement les processus dont les lignes contiennent le mot **getty**

Réponse :.....

-Essayer de supprimer un processus *getty* ? quelle est la particularité de ces processus ?

Réponse :.....

### **Les "démons" (deamons)**

Ce sont des processus qui ne sont pas exécutés dans une console (*ps* affiche alors un ? au lieu du nom de console *tt*).

Les "démons" exécutent des services du système (notamment des services réseaux) .

## **Exemple : Serveur ssh**

Installer `openssh-server` sur la machine serveur

```
sudo apt-get install openssh-server
```

Etat/Démarrer/ Arrêter le serveur :

```
service ssh status ou start ou stop
```

ou

```
/etc/init.d/ssh status ou start ou stop
```

Changer l'état du serveur avec la commande **service..**

Utiliser *ps* associé à *grep* pour afficher les caractéristiques du daemon *sshd*

Réponse :.....

Pour accéder à distance d'un ordinateur client :

Il faut avoir un login être un utilisateur déclaré de la machine distante et connaître son adresse IP

Pour connaître l'adresse IP en réseau local utiliser *ifconfig*

Adresse IP de votre machine (ou s'exécute le serveur ssh): IP = .....

D'une autre machine essayer de vous connecter à distance :

```
ssh login@adresse IP_serveur
```

répondre « yes » à la question Faire vérifier par l'enseignant lorsque ça fonctionne.

Fichier de configuration :

```
/etc/ssh/sshd_config
```