

TP8 : Gestion des Processus

- 1) Lancez la commande `top` et observez les processus qui tournent. `top` met à jour cette liste périodiquement. Tapez `c` pour voir plus de détails sur les processus qui tournent. Quittez avec `q`. À l'aide de `man`, expliquez les colonnes suivantes de la commande `top` : **PID, USER, PR, RES, S, %CPU, %MEM, TIME+, COMMAND**.
- 2) Quels sont les différents états (*states*) que peut prendre un processus, selon la commande `top` ?
- 3) Sur votre terminal, lancez les commandes `ps aux` et `ls /proc`. Voyez-vous une correspondance entre les numéros affichés ?
- 4) Trouvez le PID de votre shell `bash` en lançant `ps`.
- 5) En utilisant `man ps`, trouvez la commande qui permet:
 - a) d'afficher les processus qui tournent pour vous seulement (pour votre utilisateur)
 - b) d'afficher tous les processus en arbre, pour voir les relations de parenté entre processus
 - c) de trouver le nom du processus père de votre shell `bash`.
- 6) *Faites ce numéro sur votre terminal seulement.* Créez-vous un fichier de script `bash` intitulé **`forever.sh`** qui contient le code suivant :

```
#!/bin/bash

while [ 1 ]; do
    for ((x=1; x <= 100000; x++))
    do
        echo 'hello' > /dev/null
    done
    echo "bonjour, il est "`date`";
done
```

- a) Donnez-lui les droits d'exécution pour vous (`chmod`) et lancez-le. Est-ce que le script va s'arrêter un jour ?
- b) Ouvrez une autre console pendant que le script s'exécute et trouvez le PID de `forever.sh` avec la commande `pgrep`.
- c) Expliquez son état avec la commande `ps -o stat,comm -p PID`.

d) Revenez à la console d'où vous avez lancé **forever.sh** et faites **CTRL-Z**. Il est maintenant interrompu par le shell. Quel est son état ?

e) Revenez à la console d'où vous avez lancé **forever.sh** et faites **bg**.

Le processus est maintenant en « background » et tourne, vous laissant utiliser la console (même si c'est un peu difficile, puisque le programme affiche pendant que vous tapez...).

Lancez **fg** pour le mettre au premier plan, et tuez le processus avec **CTRL-C**.

f) Relancez **forever.sh** avec, à la fin de la commande, l'esperluette (**&**). Ceci lance la commande en arrière plan et vous redonne le contrôle de la console.

g) Lancez deux autres **forever.sh** avec une esperluette, vous avez trois processus qui tournent concurremment. Faites **jobs** pour vous en assurer.

h) Trouvez le PID de vos trois **forever.sh** et faites, sur l'autre console, la commande **kill** suivie du **pid** de chacun des processus. À quoi sert la commande **kill** ? Pourquoi **killall** aurait-il pu accélérer votre travail ?

7) Lancez les commandes suivantes :

a) Dans une première console, lancez la commande **cat**. Dans la deuxième console, à l'aide de **top**, trouvez son état, puis faites un **kill** pour mettre fin à **cat**.

b) Expliquez l'état que vous avez observé pour **cat**.

8) Repérer un processus qui vous appartient de votre choix et trouvez-en le **PID**. Allez ensuite dans le répertoire **/proc** pour trouver les informations suivantes :

a) Les fichiers que le processus a ouverts

b) Le nom complet de l'exécutable auquel correspond ce processus.

9) Programmez le lancement du script **forever.sh** à 10h30 du Lundi à Vendredi, avec **crontab**. (Attention à faire une redirection de la sortie standard du script dans un fichier: **~/FileForever**).

10) Programmez une autre fois le lancement du script **forever.sh** en tâche de fond avec la commande **at** à 10:40.

11) Programmez un **signal pour tuer** tout les processus **forever.sh** avec **crontab** à 10h50 du Lundi à Vendredi.

☛ Utilisez la commande: **\$ killall [process_name]**

12) Programmez l'enregistrement des informations: **date et utilisateur connecté** à 08h30 avec la commande **at**.

13) Donnez "Toujours" comme **alias** pour le lancement du script **forever.sh**.