

TP4 : Gestion des processus Unix

Hossam Nazih

Filière : 3IIR

Exercice 1 :

1. Le cycle de vie d'un processus comporte plusieurs états :
 - NEW : Le processus est en cours de création
 - READY : Le processus attend d'être assigné à un processeur
 - RUNNING : Le processus est en cours d'exécution
 - WAITING : Le processus attend qu'un événement se produise
 - TERMINATED : Le processus a terminé son exécution
2. Oui, sous Linux, tout processus a un père (processus parent) à l'exception du processus init (PID 1) qui est le processus racine.
3. Le processus avec PID 1 est le processus "init" ou "systemd" sur les systèmes modernes. On peut le voir avec la commande `ps lax | grep "1 ?"`.
4. La commande `ps -f` affiche :
 - PID : Process ID (identifiant unique du processus)
 - PPID : Parent Process ID (identifiant du processus parent)
5. Différence entre `ps` et `top` :
 - `ps` : affiche un instantané des processus à un moment donné
 - `top` : affiche une vue dynamique et mise à jour en temps réel des processus
6. Différence entre `fg` et `bg` :
 - `fg` : met un processus en avant-plan (foreground)
 - `bg` : met un processus en arrière-plan (background)
7. `fork()` et `exec()` :
 - `fork()` : crée une copie du processus actuel (processus enfant)
 - `exec()` : remplace le processus actuel par un nouveau programme

Exercice 2 :

1. Après l'ouverture du terminal, la commande `ps -e` devrait afficher quelques processus de base :
 - Le shell (bash/zsh)
 - Le processus du terminal lui-même
2. Pour afficher le manuel de `ps` :

bash

Copy

```
man ps
```

Pour suspendre : Ctrl+Z

3. La commande `ps` devrait maintenant afficher :
 - Les processus précédents
 - Le processus `man` suspendu. Donc au moins 3 processus.
4. La commande `jobs` devrait afficher 1 job :
 - Le processus `man` suspendu [1]
5. La commande `jobs -l` affichera :
 - Le numéro du job [1]
 - Le PID du processus `man`
 - Son état (Stopped)
6. Pour arrêter le processus `man` :

bash

Copy

```
kill %1
```

ou Ctrl+C

7. La commande `jobs` devrait maintenant ne rien afficher car le processus est terminé.

Exercice 3 :

1. Non, quand `top` est en exécution en avant-plan, on ne peut pas exécuter d'autres commandes dans la même fenêtre.

2. `Ctrl+Z` suspend `top`. La commande `jobs` indiquera : `[1]+ Stopped top`
3. `fg` remet `top` en avant-plan. `Ctrl+Z` le suspend à nouveau.
4. `bg` fait exécuter `top` en arrière-plan. On peut maintenant exécuter d'autres commandes.
5. Avec un nouveau `top` suspendu, on aura maintenant deux processus `top`.
6. `jobs` montrera deux processus `top` : `[1]- Stopped top [2]+ Stopped top`
7. `kill -9 %1` tue le premier processus `top` en envoyant un signal `SIGKILL`.
8. `ps` affiche les PID de vos processus actuels.
9. `ps -f` ajoute des colonnes supplémentaires (UID, PPID, C, STIME, TTY, TIME, CMD).
10. `ps -af` affiche tous les processus du système avec les informations détaillées.
11. Tenter de tuer le processus d'un autre utilisateur avec `kill -9 pid` résultera en "Operation not permitted" sauf si on est root.
12. `top&` lance `top` en arrière-plan directement. Oui, on peut exécuter d'autres commandes.
13. `fg` ramène `top` en avant-plan.
14. `bg` remet `top` en arrière-plan.