

TP - Gestion des processus et des ressources

M. Tellene

EXERCICE 1

1. Lancez les utilitaires `xclock` et `xcalc` depuis votre terminal (n'oubliez pas de faire suivre leurs noms par le caractère `&` pour ne pas perdre la main dans le terminal)
2. Sans fermer les programmes que vous venez de lancer, testez la commande `ps` depuis le terminal. Lancez un second terminal et recommencez. Décrivez les changements. Puis fermez le second terminal
3. L'option `-l` de `ps` permet d'afficher plus de détails sur les processus. Testez cette option et repérez les caractéristiques suivantes :
 - son numéro d'identification (PID ou process identifier), qui est un nombre entier positif
 - l'identifiant du processus qui lui a donné naissance, ou processus parent, appelé PPID (pour parent PID)
 - son propriétaire, en général (mais pas toujours) l'utilisateur qui l'a lancé
 - éventuellement le terminal dont il dépend (s'il existe)
 - son répertoire courant, sa priorité de travail, son temps d'exécution, etcDétaillez, en vous aidant, au besoin du manuel, la signification des colonnes `UID`, `PRI`, `TT`, `VSZ`, `TIME`
4. Comment peut-on afficher la liste de tous les processus du système ? Comment peut-on regrouper les processus par propriétaire ? Testez ces options

EXERCICE 2

La commande `kill` permet d'envoyer différents types de signaux à un processus dont on connaît l'identifiant (PID). Malgré son nom, elle ne sert pas seulement à « tuer » un processus. Il existe de nombreux signaux différents dont vous trouverez la signification dans le `man`. Les signaux les plus courants sont `SIGTERM` et `SIGKILL`, qui servent à terminer un processus

1. La liste des signaux que l'on peut envoyer aux processus s'obtient grâce à l'option `-l` de `kill`. Quels sont les numéros correspondant aux signaux `SIGTERM`, `SIGKILL`, `SIGSTOP` et `SIGCONT` ?

La syntaxe d'envoi d'un signal est : `kill -signal pid`

où `signal` est un numéro ou un nom de signal (le signal par défaut est `SIGTERM`)

2. Testez les signaux mentionnés ci-dessus sur un processus `xclock` préalablement lancé, et décrivez le résultat
3. Lancez un second terminal, puis testez les signaux `SIGTERM` et `SIGKILL` sur le processus de ce nouveau terminal. Que constatez-vous ? Qu'en déduisez-vous sur le fonctionnement des signaux `SIGTERM` et `SIGKILL` ?
4. Repérez parmi les processus actifs un processus dont vous n'êtes pas propriétaire, et tentez de le stopper à l'aide du signal `SIGSTOP`. Que se passe-t-il et qu'en déduisez-vous ?