

Processus

TD5 - séance 1

Objectifs

Un processus représente l'exécution d'un programme dans un système d'exploitation. Il peut être vu comme une structure de données regroupant les informations nécessaires à la gestion de l'exécution du processus (propriétaire, groupe, priorité, ressources utilisées...). Cette séance vous présente les principales commandes d'examen et de communication des processus en cours d'exécution. Nous étudierons aussi la notion d'exécution en *arrière-plan* et en *avant-plan*.

Collecte d'information

Manipulation 1

Essayez les commandes suivantes et commentez. Regardez la signification des différentes options en utilisant la commande man.

```
ps
ps -a
ps -u <nom_login>
ps x
```

Note : Vous verrez souvent la syntaxe suivante : *ps aux* (notez l'absence de - devant *aux*). Il s'agit de la survivance d'une ancienne syntaxe (dite BSD) utilisée par la commande *ps*. L'option *-u* en particulier a complètement changé de sens : elle permettait alors d'obtenir plus d'informations sur le processus.

Manipulation 2

Testez la commande ps aux et interprétez les champs supplémentaires.

Manipulation 3

Testez les options suivantes qui permettent d'obtenir plus d'informations sur les processus : -l. Interprétez les champs supplémentaires.

Manipulation 4

Testez la commande `top` pour obtenir des informations continuellement mises à jour.

La commande *kill*

La commande *kill* permet à l'utilisateur de communiquer avec le processus par l'envoi d'un *signal* au processus. Un signal peut être vu comme une sorte de message qui est reçu et traité par le processus. Il existe un traitement par défaut pour chaque signal. Certains peuvent être spécifiés par l'utilisateur.

Si aucun signal n'est fourni à la commande *kill*, c'est le signal SIGTERM (numéro 15) qui est envoyé par défaut afin d'interrompre un processus en cours d'exécution (d'où le nom de *kill*). Quelques signaux utilisés pour interrompre un processus :

SIGINT (2) Interruption à partir du clavier (équivalent à `^C`).

SIGQUIT (3) Quitter à partir du clavier (équivalent à `^D`).

SIGKILL (9) Tue un processus. Celui-ci ne peut ignorer ce signal. Il termine sans effectuer aucune tâche de nettoyage (fermeture de fichiers, effacement de fichiers temporaires...).

A n'utiliser que si aucune autre solution ne fonctionne.

SIGTERM (15) Terminaison propre d'un processus.

Manipulation 5

Essayez la commande suivante pour obtenir la liste des signaux pouvant être envoyés : `kill -l`. La commande `man 7 signal` permet d'obtenir une description des signaux et du comportement par défaut du processus à sa réception.

Exécution en avant-plan et en arrière-plan

Lorsque vous exécutez une commande à partir d'un interpréteur de commandes, ce dernier ne vous permet de lancer une nouvelle commande que lorsque la précédente est terminée. On appelle cela l'exécution en *avant-plan*. Si vous terminez la commande par le caractère `&`, l'interpréteur vous permet de commencer l'exécution d'une nouvelle commande alors que la précédente n'est pas terminée. On parle alors d'exécution en *arrière-plan*.

Manipulation 6

Exécutez et commentez les commandes suivantes :

```
xterm &  
ps  
jobs  
fg  
^C
```

Une commande peut-être stoppée (mais non interrompue) avec le raccourci clavier ^Z (équivalent à `kill -SIGSTOP PID`).

Manipulation 7

Exécutez et commentez les commandes suivantes :

```
xterm
^Z
bg
ps
kill <PID>
```

Manipulation 8

Exécutez et commentez les commandes suivantes :

```
xterm -name f1 &
xterm -name f2 &
jobs
fg <Numero associe a la fenetre f1 par jobs>
^C
```

1. Que se serait-il passé si aucun numéro n'avait été mentionné après la commande `fg` ?

Répertoire `/proc`

Pour chaque processus lancé, le système crée un répertoire dans `/proc` ayant pour nom le PID du processus. Ce répertoire contient divers fichiers stockant les informations utilisées par le système afin de gérer les processus.

Le système de fichier `/proc` est un pseudo-système de fichiers qui est utilisé comme interface avec les structures de données du noyau. Il est généralement monté sur `/proc`. La plupart des fichiers sont en lecture seule, mais quelques uns permettent la modification de variables du noyau.

Manipulation 9

Examinez à l'aide de la commande `cat`, le fichier `stat` d'un répertoire quelconque du répertoire `/proc`. En utilisant les différentes commandes vues dans les précédentes manipulations, retrouvez les informations suivantes :

- commande de lancement du processus,
- PID du processus,
- PPID du processus,
- statut du processus,