

PROCESSUS

SYSTEME MULTI-TACHES

Le système Unix (et donc Linux) a été conçu dès le début pour être un système multi-tâches, avec un système efficace de gestion des processus.

Un processus correspond à une tâche (une application, un service). Il répond à un certain nombre d'ordres (signaux) qui permettent de le piloter (arrêt, redémarrage, ...) et possède un environnement d'exécution, dans lesquels certaines valeurs sont définies.

Signaux

Linux définit un certain nombre de signaux auquel un processus doit répondre. Ces signaux peuvent être envoyés à un processus avec la commande « *kill* » :

- **SIGINT** (interruption) : interrompt le processus courant et le termine
- **SIGTERM** (termination) : interrompt le processus et le termine
- **SIGKILL** (kill) : interrompt le processus et le termine
- **SIGSTOP** (stop) : interrompt le processus et le met en pause
- **SIGCONT** (continue) : reprend le processus mis en pause

(il existe d'autres signaux, mais moins utilisés)

La différence entre *SIGTERM* et *SIGKILL* est que le processus peut ignorer le signal *SIGTERM*, ce qui arrive par exemple s'il est très occupé ou s'il est planté. Par contre, même planté, il ne peut pas ignorer le signal *SIGKILL*.

Un processus en pause peut être relancé, il peut aussi passer en tâche de fond ou au contraire au premier plan. Le processus au premier plan est celui qui écrit (des logs généralement) et lit les entrées de l'utilisateur (le cas échéant) dans le terminal qui a servi à le lancer.

Certains signaux peuvent aussi être directement envoyés au processus courant (celui au premier plan) depuis le shell, à l'aide de raccourcis claviers (CTRL désigne la touche « contrôle », à garder appuyée en même temps que l'autre lettre qui suit) :

- **CTRL-C** = **SIGINT**
- **CTRL-Z** = **SIGSTOP**

Dans le même registre, **CTRL-D** permet de se déconnecter d'un shell ou d'un terminal. Le raccourci n'envoie pas un signal mais le caractère **EOF** (**E**nd **O**f **F**ile) ce qui termine la connexion.

Environnement d'exécution

L'environnement d'exécution d'un processus contient un ensemble de valeurs regroupées sous le terme de variables d'environnement.

Une variable d'environnement fait correspondre un nom (le nom de la variable) à une valeur. Ce couple nom/valeur est accessible par le processus qui peut l'utiliser pour configurer son action. Il est ainsi possible, en modifiant les variables d'environnement, de modifier le comportement d'un programme sans avoir à lui passer tout le temps les mêmes paramètres au lancement.

PATH

PATH est une variable d'environnement très spécifique, qui est utilisée par le shell. Elle contient une liste de chemin séparée par des « : ».

Lorsque l'on entre une commande dans le shell, celui-ci vérifie chacun des chemins, dans l'ordre, pour trouver s'il existe un fichier exécutable avec ce nom dans le chemin en question. Dès qu'il en trouve un, il l'exécute. S'il n'en trouve aucun, la commande est marquée comme non trouvée. Attention, l'ordre est important : le premier fichier exécutable trouvé avec le bon nom est exécuté.

Fichiers de configuration

Le placement d'une variable d'environnement est locale au shell, et disparaît à la fermeture de ce dernier. Afin d'avoir certaines variables positionnées automatiquement dès le lancement d'un shell, ce dernier possède des fichiers de configurations qui sont lus au démarrage, notamment le fichier « .bashrc », à la racine de votre répertoire home.

shebang

Certaines commandes sont propres au shell et ne correspondent à aucun fichier exécutables (comme « cd »), d'autres sont des binaires, des exécutables directement écrits en langage machine. D'autres, enfin, sont simplement des scripts, des programmes au format texte qui nécessitent un interpréteur pour exécuter chacune des instructions (comme en python par exemple).

Pour exécuter ces scripts comme les autres commandes, sans avoir à lancer d'abord l'interpréteur, le système Unix a mis en place un mécanisme : le shebang

Si la première ligne d'un fichier texte commence par « #! », ce qui suit sur la même ligne est considéré comme étant un chemin vers l'interpréteur qui exécutera ce script.

Bien sûr, il faut quand même que le fichier soit exécutable.

COMMANDES

Voici une liste des commandes que vous devez connaître.

ps

Retourne la liste des processus en cours. L'option « *aux* » permet de voir tous les processus, y compris ceux qui n'appartiennent pas à l'utilisateur, l'option « *ww* » permet d'avoir les descriptions complètes (ATTENTION, pour une fois, les options de *ps* ne prennent pas de « - » devant : « *ps auxww* »). Permet notamment d'obtenir le pid d'un processus.

top

Retourne la liste des processus en cours, classé en temps réel suivant leur utilisation des ressources. Affiche aussi la charge de la machine. Comme pour « *less* », pour quitter, tapez la lettre « q ».

kill <signal> <pid>

Permet d'envoyer un signal au processus identifié par son pid. Par défaut, si aucun signal n'est donné, c'est un SIGTERM qui est envoyé. Le signal à envoyer peut être précisé soit sous la forme d'un tiret suivi d'un nombre (-9 pour SIGKILL), soit sous la forme du nom court du signal (KILL, TERM, STOP, CONT) précédé de « -s ». Pour les pid, « -1 » est un alias pour « tous les processus »... q

jobs -l

Liste les processus en tâche de fond ou suspendus.

fg

Fait passer en premier plan un processus en tâche de fond ou suspendu.

bg

Fait passer en tâche de fond un processus suspendu. Pour lancer directement un processus en tâche de fond, ajoutez le caractère « & » (séparé par un espace) à la fin de la ligne de lancement du processus.

env

Liste l'ensemble des variables d'environnement définies pour ce shell et ses processus.

export <var_name>=<var_value>

Crée ou modifie une variable d'environnement. Pour ajouter quelque chose à une variable d'environnement (par exemple un chemin à PATH), il faut référencer l'ancienne valeur de la variable à droite du signe égal avec « \$VAR » (par exemple PATH=\$PATH:/mon/nouveau/chemin).

which <command_name>

Permet d'afficher le chemin d'une commande. Retourne le premier répertoire de PATH dans lequel un fichier exécutable avec le nom de la commande a été trouvé.

source <fichier_config>

Permet de relire un fichier de configuration shell, par exemple .bashrc.