# Gestion des Processus sous Linux – Commande KILL

**I► Dans ce tutoriel, nous allons voir comment gérer et contrôler les processus du système.**
**Les informations des processus actifs sont enregistrés dans /proc, le pseudo FileSystem.**

Chaque processus possède un répertoire du PID du processus concerné. Toute une foule d'informations concernant ce même processus y est écrite.

Dans **/proc**, on va trouver également des fichiers d'information renseignant l'état du système, comme par exemple **meminfo**, que l'on peut interroger avec la commande **free -n**. **version** qui donne le nom de la distri, le kernel, que l'on peut interroger avec **uname -r ou -v**.

**ps aux** ou **ps ef** selon la norme listera tous les processus en cours. Il sera utile de piper cette commande avec un grep, ex : **ps aux | grep samba**
On peut personnaliser la commande ps comme suit : **ps -e -o%cpu,%mem,comm,time,pid,ppid**

**pstree -p** Montre l'affiliation des processus entre eux.

**top** ou **htop** permet de lister dynamiquement avec rafraîchissement les processus en cours . On peut passer toute sorte d'options à top comme **k**ill **r**enice **b**old **z(**couleur)

L'utilitaire graphique **qps** est assez intéressant pour contrôler et gérer les processus.

_____-

**Priorité des Processus :**

Les Processus dorment (**S**leep) , ils attendent que l'ordonnanceur du noyau autorise l'accès au processeur afin qu'ils puissent y exécuter (**R**unning) leurs commandes en manipulant les jeux d'instruction du processeur.

Certains processus sont plus prioritaire que d'autres et auront tendance à passer devant les autres dans la file d'attente (pile FIFO) de la RAM.
Certains processus peu gourmands dorment directement en SWAP sur le disque dur et quand le moment est venu pour eux d'aller éxécuter leurs commandes, ils réempruntent le chemin de la RAM et passent dans la file d'attente de la RAM.

Le **swapiness** par défaut du noyau est par défaut réglé à 40%, à partir de cette charge, certains processus peu gourmands iront dormir directement en SWAP.
Exemple : Une application lourde (graphique) charge à son initialisation un grand nombre de processus en RAM. Lorsque l'application fonctionne, ils ne sont plus forcément nécessaires. Le Swapiness fonctionne alors et envoie ces processus en SWAP.

Le **swapiness** en tant que directive du noyau peut être paramétré via /etc/sysctl.conf ou depuis : **/proc/sys/vm/swappiness.**
Autre exemple de tuning : **echo 3 > /proc/sys/vm/drop_caches** ? purge le cache mémoire, les dentries(relation entre fichier et inodes)

? Pour modifier la priorité d'un processus, on pourra utiliser les commandes **nice** ou **renice** (qui donnent une gentillesse de -20 à 19 )

---

La commande KILL permet d'envoyer des signaux aux processus

grâce à leur PID
KILLALL cible les processus par leurs noms.
KILL -l ? donne la liste de tous les signaux

KILL -15 PID = SIGTERM ? On demande au processus de se terminer proprement
KILL -9 PID = SIGKILL ? Le processus se termine ! impérativement !
KILL -1 PID = SIGHUP ? On demande au processus démon ou au service de relire sa configuration
KILL -19 PID = SIGSTOP ? On met en pause le processus, on le stoppe, il n'est pas terminé
KILL -18 PID = SIGCONT ? On redémarre le processus stoppé.

**? Autre manière de contrôler les processus :**

On lance un processus en tant que job :
firefox**& ?** Lance le procesus en tache de fond et donne le processus et le numéro de job (commande jobs pour lister tous les jobs en cours d'éxécution)
**fg** pour le ramener au premier plan ou fg N°deJob

on peut l aminpuler avec CTL+Z pour le stopper
**CTRL+Z** Stoppe le processus (équivalent du KILL -19)
**bg** pour le passer en tache de fond et le relancer (comme kill -18)

***CTRL+C*** est un signal d'interruption SIGINT qui termine l'éxécution de la commande en cours

Voici un exemple de manipumation des jobs :

Les commandes sont lancées en tant que job avec & (esperluette) et passent directement en bg(background). On peut les appeler en fg(ForeGround) et les manipuler avec CTRL+Z ou CTRL+C.

Un job passé en FG devra être stoppé avant de repasser en BG !!

*Michel BOCCIOLESI*