
TP administration de Linux

Administration d'un système Linux à travers un shell

Olivier LASNE

2020-11-26

Administration de Linux

Introduction

Le but de ce TP est de vous mettre à l'aise avec l'administration d'un système Linux à travers la ligne de commande. Nous y verrons à la fois des commandes d'administrations, des commandes puissantes.

Linux, et les fichiers

Sous Linux, il est coutume de dire de "Tout est fichier". À peu près tout est représenté sous la forme d'un fichier.

Les disques durs de l'ordinateur sont représentés par les fichiers `/dev/sda`, `/dev/sdb`, etc. Les processus de l'ordinateurs sont représentés dans `/proc/`. Les paramètres du noyau sont représentés par `/sys/kernel/`.

Un exemple de cela, est que l'on peut ouvrir une connexion TCP en ouvrant un fichier. Par exemple `/dev/tcp/10.0.0.1/8080` pour la machine `10.0.0.1` sur le port `8080`.

Nous ne rentrerons pas ici dans les détails, retenir juste que l'on peut configurer et accéder à presque tout à travers des fichiers.

Richesse de cette culture, Linux va utiliser des fichiers pour représenter la plupart des paramètres de configuration.

Lister les processus

On utilise la commande **ps** pour lister les processus. On peut lister l'ensemble des processus avec **ps aux**.

```
1 $ sudo ps aux
2 USER          PID %CPU %MEM    STAT   START    TIME COMMAND
3 root             1  0.1  0.1    Ss     13:47    0:06 /sbin/init splash
4 root             2  0.0  0.0    S       13:47    0:00 [kthreadd]
5 root             3  0.0  0.0    I<      13:47    0:00 [rcu_gp]
6 root             4  0.0  0.0    I<      13:47    0:00 [rcu_par_gp]
7 root             6  0.0  0.0    I<      13:47    0:00 [kworker/0:0H-kblockd]
8 root             8  0.0  0.0    I<      13:47    0:00 [mm_percpu_wq]
9 root             9  0.0  0.0    S       13:47    0:00 [ksoftirqd/0]
10 ...
```

Le **PID** (Process ID) est l'identifiant unique de chaque **processus**.

Exercice :

1. Lancer `cmatrix` dans un terminal.
2. Ouvrir un second terminal, et utiliser `ps aux | grep cmatrix` pour obtenir le **PID** de **cmatrix**.

Tuer un processus

On peut tuer un processus avec la commande **kill** et son **PID**. Par exemple, pour un processus dont le PID serait 2843. On peut utiliser la commande suivante :

```
1 $ kill 2843
```

`kill` sans option demande au processus de s'arrêter "proprement". Si jamais le processus ne répond pas on peut utiliser `kill -9` pour forcer l'arrêt du processus.

```
1 $ kill -9 2843
```

Exercice :

1. Lancer `cmatrix` dans un terminal.
2. Ouvrir un second terminal, trouver le **PID** de `cmatrix` puis tuer le avec **kill**.

Htop

On peut voir une **liste dynamique** des processus avec la commande **top**.

Pour voir gérer l'ensemble des processus, un programme très pratique est **htop**. Il est nécessaire de l'installer avec `sudo apt install htop`.

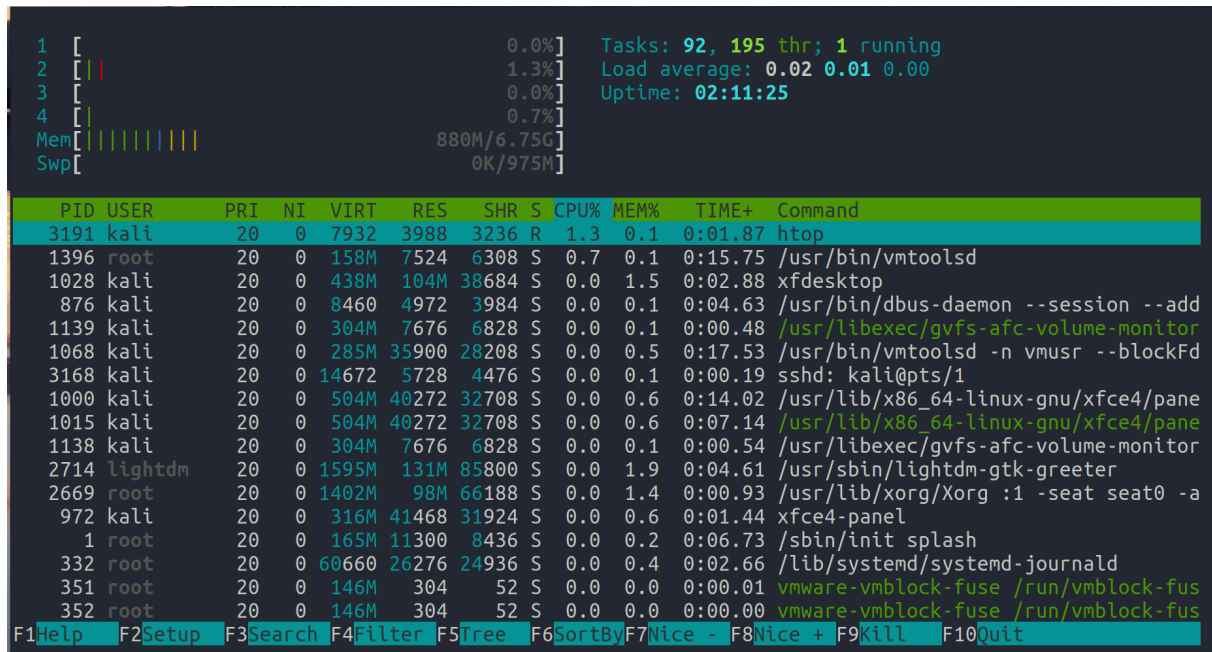


FIG. 1: Et c'est classe

On peut utiliser **F5** pour afficher les processus sous forme d'arbre.

Exercice :

1. Lancer **Firefox**
2. Utiliser **Htop** pour voir les processus fils (**F5**), observer l'arborescence
3. Utiliser **F4** pour filter firefox
4. Supprimer le filtre, et regarder l'aide avec la touche **h**.
5. Lister les processus par CPU (**P**), mémoire (**M**)
6. Tuer **firefox** avec **F9**

Autre commandes

killall : Tuer tous les processus portant un nom

```
1 $ killall firefox
```

pgrep : Trouver un PID à partir du nom d'un processus. `pgrep zsh`

Les utilisateurs et les groupes

Nous ne verrons pas ici comment ajouter et supprimer des utilisateurs. Je vous invite à faire ce tutoriel en savoir plus.

<https://openclassrooms.com/fr/courses/43538-reprenez-le-contrôle-a-l'aide-de-linux/39044-les-utilisateurs-et-les-droits>

Sous Linux, les utilisateurs font partis de groupes.