

Découverte de Linux

Quelques notions à connaître pour utiliser un système Linux

I) Généralité

GNU/Linux, abusivement appelé Linux est un **système d'exploitation** construit autour du noyau Linux.

De ce système d'exploitation, découle un ensemble de **distributions** qui permettent d'avoir un environnement d'utilisation clef en main. Dans notre cas, on utilise **ArchLinux**, mais il beaucoup d'autre (Debian, Ubuntu, Mint, Fedora, etc.). Chacun a des composants qui lui sont propre et certains proposent même des optimisations du noyau.

Les distributions proposent (entre autre) des **environnements de bureau**. Sur les postes, on utilise **XFCE**, mais encore une fois, il existe beaucoup d'alternatives (Mate, Gnome, Unity, etc.). Un environnement n'est pas lié à une distribution (on peut très bien avoir Gnome sur ArchLinux).

Linux est **multi-utilisateurs** (plusieurs personnes peuvent en même temps travailler sur le même ordinateur), **multitâches** (plusieurs applications peuvent être lancées en même temps sans qu'aucune n'affecte les autres) et **multiprocesseurs**.

II) Organisation

L'intégralité du système de fichier Linux part d'un seul point : **la racine** ou « / »

Lorsque l'on doit spécifier où se trouve un fichier dans l'arborescence, il y a deux manières de faire :

- **l'adresse absolue** : on part de la racine et on remonte jusqu'au fichier ex. : /usr/bin
- **l'adresse relative** : à partir de l'endroit où on se trouve (et dans ce cas, on ne met pas / au début de l'adresse).

III) Le shell

A) Qu'est ce que c'est ? Comment l'utiliser ?

Tout système d'exploitation utilise une interface avec ses utilisateurs, sous forme graphique ou sous forme de ligne de commandes. On parle alors de **shell** ou d'**interpréteur de commandes**.

Pour l'ouvrir, il suffit de chercher l'application **Terminal** dans le menu d'application. Plus rapide encore, depuis le bureau, le raccourcis **Ctrl+Alt+T** ouvre directement le terminal.

Un autre raccourcis peut être utile (uniquement sur XFCE, à tester sur d'autre environnement): **Alt+F3**. Cela ouvre le menu application au centre de l'écran, mais avec une barre de recherche où il suffit de taper les première lettre du nom de l'application pour la trouver.

B) A quoi ça sert ?

Le but du terminal est d'interagir avec le système d'exploitation. Il permet tout ce que fait déjà l'interface graphique (et plus encore) mais de manière bien plus rapide. L'inconvénient est que l'utilisation n'est pas très intuitive et nécessite un peu d'apprentissage.

Pour utiliser le Shell, on doit taper des **lignes de commandes** qui correspondent à des fonctions en programmation. Mais l'analogie avec la programmation ne s'arrête pas là car il existe un vrai langage pour automatiser des fonctions systèmes. Sur ArchLinux, ce langage est **bash** (Bourne Again Shell).

C) Quelques commandes de bases

i) Changer de répertoire

Par défaut, lorsque vous ouvrez un terminal, vous vous trouvez dans le répertoire Home de l'utilisateur, c'est à dire **/home/nomUtilisateur**

Pour changer de répertoire, il faut utiliser la commande `cd` (change directory)

<code>cd /</code>	Permet d'accéder à la racine
<code>cd ~</code> ou <code>cd</code>	Accéder directement au répertoire de l'utilisateur.
<code>cd /var/www</code>	Permet d'accéder au répertoire /var/www
<code>cd ..</code>	Permet d'accéder au répertoire parent
<code>cd -</code>	Permet de revenir au répertoire précédent.

« *Mais c'est pas très pratique de taper en toutes lettres une adresse absolue !* »

Il existe un raccourcis en Shell pour compléter automatiquement les mots que vous avez commencer. En utilisant la touche **Tabulation**, le shell complète automatiquement la suite de la commande pour peu que vous ayez taper quelques lettres. N'hésitez pas à user et abuser de cette touche, elle délivre toute la puissance de la ligne de commande et vous évitera de faire des fautes de frappes !

ii) Consulter le contenu d'un répertoire

Changer de répertoire, c'est bien beau, mais voir ce qu'il y a dedans, c'est mieux !

Pour cela, on utilise la commande `ls`

<code>ls -l</code>	Afficher les informations de manière détaillée.
<code>ls -a</code>	Afficher les fichiers cachés.
<code>ls -h</code>	Afficher la taille des fichiers de façon lisible.
<code>ls -r</code>	Tri inversé.
<code>ls -t</code>	Trier les fichiers par date du plus récent au plus ancien.
<code>ls -S</code>	Trier par taille décroissante.
<code>ls -la</code>	Afficher tous les fichiers y compris les fichiers cachés.
<code>ls -lhS</code>	Afficher les informations des fichiers, avec des tailles lisibles le tout ordonné du plus grand au plus petit

Il existe beaucoup d'autre options pour cette commande. Vous pouvez consulter les possibilités à l'aide de la commande **man ls** (la commande `man` marche pour l'ensemble des commandes Linux)

Une autre commande permet de visualiser le contenu de répertoire : **tree**. Par contre, cette commande affiche le contenu d'un répertoire mais également toute l'arborescence qui se trouve en dessous !

iii) Gérer les répertoires

Pour créer des répertoires, on utilise la commande `mkdir`

<code>mkdir -v</code>	Retourner des informations lors de la création d'un répertoire.
<code>mkdir -p</code>	Cette option permet de créer une arborescence complète.
<code>mkdir foo</code>	Créer le répertoire foo.
<code>mkdir -v foo /tmp/bar</code>	Créer les répertoires foo et /tmp/bar.
<code>mkdir -p foo/bar/baz</code>	Créer l'arborescence foo/bar/baz.

L'option `-v` (pour verbose) est commune a de nombreuses commandes et permet d'avoir des détails d'exécution de la commande.

Pour supprimer des répertoires il existe deux commandes.

Une pour les répertoires vides : **rmdir**

Une autre pour tous les autres cas: **rm**

<code>rm *.txt</code>	Supprimer tous les fichiers ayant pour extension txt.
<code>rm foo.txt bar.txt</code>	Supprimer les fichiers foo.txt et bar.txt.
<code>rm -rf baz/</code>	Supprimer le répertoire baz et tout son contenu.

ATTENTION ! La suppression en ligne de commande est définitive ! Il n'existe pas de corbeille.

iv) Lancer des exécutable

Lorsque l'on utilise la commande **ls,-l**, on obtient des détails sur les fichiers dans le répertoire.

Au début de chaque ligne, on trouve une série de symbole comme ceci :

```
-rw-rw-r--
```

Cette série peut être découpée en 4 groupes.

Le premier caractère correspond au type d'élément. - correspond à un fichier, d correspond à un dossier. Il existe d'autre type, mais ce n'est pas utile pour le moment.

Le second groupe correspond au 3 lettres qui suivent et indique les droit d'accès de l'utilisateur courant : r indique que l'utilisateur a le droit de lecture, w le droit d'écriture et x le droit d'exécution. Si 3 tirets apparaissent à la place des 3 lettres, cela veut dire que l'utilisateur n'a aucun droit.

Les deux groupes suivant font également 3 lettres et indique également des droit, mais pour des groupes d'utilisateurs défini (premier groupe) et pour n'importe qui (second groupe)

Remarque! Comme on peut le voir, n'importe quel fichier peut donc être exécutable pour peut que l'on ait les droits. Il n'existe pas d'extension lié à ça comme sur Windows avec le .exe.

Avant de pouvoir lancer un programme en ligne de commande, il faut d'abord savoir si notre programme est accessible depuis là où on se trouve.

Linux (et Windows aussi d'ailleurs) propose une variable d'environnement listant l'intégralité des programmes accessible depuis n'importe où dans le système (les commandes elles même sont présente dans cette variable). Il s'agit de la variable **PATH**. Pour faire apparaître le contenu de cette variable, il suffit de taper **echo \$PATH**. Le \$ permet de signaler au terminal que l'on accède à une variable.

Pour lancer un programme non présent dans la variable PATH, (exemple : votre petit programme C fraîchement compilé), il suffit de vous placer dans le répertoire ou se trouve votre fichier et le lancer via la commande suivante : **./nomDuFichier**

Il faut cependant que le fichier soit exécutable. Si ce n'est pas le cas, il suffit de taper la commande suivante : **chmod u+x ./nomDuFichier**. Pour vérifier que tout s'est bien passé, il suffit de faire **ls -l** et vérifier que le x soit présent dans les autorisations de l'utilisateur.