

COURS - INTRODUCTION A LINUX

Systèmes d'exploitation

Le programme qui tourne en permanence sur l'ordinateur, c'est le système d'exploitation (Windows 8, OsX, Linux, etc...). Ce programme est chargé au démarrage de l'ordinateur, parfois il plante : vous avez vu des écrans en ligne de commande en cas d'erreur majeure sur l'*os* (opérating system). Les rôles principaux du système d'exploitation sont les suivants :

- Fournir une «interface» entre l'ordinateur et l'utilisateur pour permettre à ce dernier de donner des ordres à la machine (par exemple : lire ou écrire des informations dans la mémoire, lancer une impression...) ou pour lui signaler les erreurs d'exécution ; cette interface prend soit la forme d'un langage de commande (comme «MS-DOS», Shell) soit la forme d'objets graphiques à manipuler (fenêtres, menus...) ;
- Gérer les «ressources» de l'ordinateur, à savoir ses mémoires, son microprocesseur et ses périphériques : les systèmes d'exploitation actuels, en effet, sont «multitâches» ; cela signifie qu'ils permettent à plusieurs programmes de s'exécuter en même temps, et se chargent de répartir l'occupation des ressources utilisées par chacun d'eux (par exemple si deux programmes P1 et P2 sont lancés en même temps, le système d'exploitation permettra à un petit bout de P1 de s'exécuter, puis laissera la place à un petit bout de P2, puis de nouveau à un petit bout de P1, etc., de sorte que l'utilisateur aura l'impression que P1 et P2 sont exécutés en parallèle, alors que le processeur est toujours unique et séquentiel) ;
- Être indépendant du matériel, masquer les particularités de la machine en substituant aux ressources physiques des abstractions (par exemple, la notion de fichier, sur laquelle nous reviendrons, est une notion abstraite, indépendante de la nature du support sur lequel les données de ce fichier sont réellement stockées) ;
- Contrôler les usagers en leur donnant des droits différents selon leur statut (associés par exemple à différents mots de passe).

Linux est un système d'exploitation open-source et libre, qui gagne du terrain grâce à sa grande légèreté, sa puissance, et son côté open-source qui minimise les bugs. Si vous ne disposez pas d'un ordinateur fonctionnant sous Linux, Windows 10 permet de travailler avec les instructions Linux, grâce au sous-système WSL, de manière très simple.

Les instructions d'installation sont données à l'adresse <https://docs.microsoft.com/fr-fr/windows/wsl/install-win10> . Installer Ubuntu 16.04 (ou un autre éventuellement, cela n'a pas d'importance). Lancer Ubuntu chez soi et se créer un compte avec un mot de passe simple à retenir.

On peut aussi faire un dual boot, c'est-à-dire que l'ordinateur peut démarrer sous plusieurs systèmes au choix, ou encore démarrer sur une clé usb externe contenant une version de Linux.

On exécute les instructions dans un *shell*, qui correspond à l'invite de commande de Windows. Maîtriser ces instructions demande un temps réellement conséquent, de l'ordre d'une année de travail à temps plein sous shell. Par contre, une fois les commandes maîtrisées, le gain de temps est important.

1. Memento Linux (source Gilles Roussel Polytechnique)

a. *Navigation dans l'arborescence des fichiers*

Les fichiers sont organisés de façon arborescente à partir de la racine /.

Un fichier est référencé par un chemin (*path*) dans cette arborescence. Exemple :

```
/tmp_mnt/usr/fmandon/cours_NSI/architecture.pdf.
```

Les répertoires (*directory*) sont des fichiers spéciaux qui peuvent contenir d'autres fichiers.

Chaque processus (en particulier le *shell*) à son répertoire de travail (*working directory*).

Il existe deux façon de référencer un fichier :

- La référence absolue qui commence par un /.
- La référence relative au répertoire de travail qui ne commence pas par /.

Il existe des références particulières :

- .. qui référence le répertoire au-dessus dans l'arborescence.

- `.` qui référence le répertoire courant.
- `~` qui référence le répertoire *home*. Répertoire de travail juste après s'être logué. `~` est une référence absolue.

Le disque Windows `C:\` se trouve à `/mnt/c/` dans l'environnement Bash. Par exemple, le dossier Documents est à l'emplacement : `/mnt/c/Users/fmandon/Documents/`

b. Commandes de base

<code>sudo</code>	<i>super utilisator do</i>	Permet d'effectuer des commandes non autorisées. Le mot de passe est demandé. S'écrit avant une autre commande, quand une erreur du type « permission denied » est soulevée. Très fréquemment utilisée avec le système WSL sous Windows 10
<code>pwd</code>	<i>print working directory</i>	Affiche le répertoire de travail
<code>cd arg/</code>	<i>change directory</i>	Change de répertoire de travail ; <i>arg/</i> est ce nouveau répertoire. <code>cd</code> sans argument positionne dans le répertoire <i>home</i> .
<code>ls arg</code>	<i>list</i>	Liste le contenu <i>arg</i> si celui-ci est un répertoire. Sans argument, <code>ls</code> liste le contenu du répertoire de travail.
<code>ll arg</code> ou <code>ls -l arg</code>	<i>long list</i>	Affiche des informations <u>détaillées</u> sur le fichier <i>arg</i>
<code>ls -a arg</code>	<i>list all</i>	Affiche tous les fichiers de <i>arg</i> , même ceux commençant par <code>..</code> , si celui-ci est un répertoire. Les options de <code>ls</code> peuvent être combinées : <code>ls -al</code> .
<code>chmod autorisations arg</code>	<i>change mode</i>	Change les autorisations (droits) du fichier/dossier, cf. ci-dessous.
<code>mkdir arg</code>	<i>make directory</i>	Crée le répertoire <i>arg</i> .
<code>rmdir arg</code>	<i>remove directory</i>	Supprime le répertoire s'il est vide.
<code>rm arg</code>	<i>remove</i>	Supprime le fichier <i>arg</i> .
<code>rm -r arg</code>	<i>remove recursively</i>	Supprime <i>arg</i> et, éventuellement, tous les fichiers qu'il contient.
<code>mv arg1 arg2</code>	<i>move</i>	Renomme <i>arg1</i> en <i>arg2</i> (si <i>arg2</i> n'est pas un nom de répertoire), ou déplace <i>arg1</i> dans le répertoire <i>arg2</i> (si <i>arg2</i> est un nom de répertoire).
<code>touch arg</code>	<i>touch</i>	Crée un fichier vide de nom <i>arg</i> , s'il n'existe pas, sinon met à jour, avec la date courante, sa date de dernière modification s'il existe déjà.
<code>ps</code>	<i>processus</i>	Liste les processus (programmes) en cours d'exécution
<code>kill arg</code>	<i>kill</i>	Arrête le processus numéro <i>arg</i>

Voir aussi `locate`, `find`, `cp`, `more`

c. Liste détaillée et permissions

Les autorisations sur un fichier, ou plus généralement sur une ressource informatique, permettent de contrôler l'accès à cette ressource. Sous Linux, on parle de *permissions*. Un autre terme équivalent est *droits*. Pour connaître les droits (entre autres) on utilise l'option `-l` de `ls`.

La commande `ls -l` retourne des informations sur les fichiers de la forme :

```
-rw-r--r--  1 fmandon  users      4104 Sep 24 23:07 texte.txt
drwxr-xr-x  3 fmandon  users      1024 Feb 20 2019 tmp
```

Le premier champ se divise en quatre parties :

- Le premier caractère de la ligne indique le type du fichier : - fichier normal, d répertoire (il existe d'autres types) ;
- Les trois caractères suivant déterminent les droits du propriétaire du fichier, ici `fmandon` ;
- Les trois d'après les droits du groupe du propriétaire du fichier, ici `users` ;
- Les derniers les droits des autres utilisateurs.
- *Les valeurs des trois caractères de droit sont :*
 - `r` si le fichier (resp. le répertoire) est accessible en lecture (resp. peut être listé) ;
 - `w` si le fichier (resp. le répertoire) est accessible en écriture (resp. est ouvert pour la création de fichiers) ;
 - `x` si le fichier (resp. le répertoire) est exécutable (resp. peut être traversé).
- Les champs qui suivent sont :
 - Le nombre de références au fichier.
 - Le nom du propriétaire du fichier.
 - Le nom du groupe du propriétaire du fichier.
 - La taille du fichier en octets.
 - La date de dernière mise à jour.
 - La référence du fichier (son nom)

d. *Changer les autorisations (droits, permissions) d'un fichier*

Une des formes pour les options de `chmod` est *dest op autorisations*. Les valeurs possibles de *dest* sont :

- `a` change les droits pour tout le monde ;
- `u` change les droits pour le propriétaire du fichier ;
- `g` change les droits pour le groupe du propriétaire du fichier ;
- `o` change les droits pour les autres.

Les valeurs possibles de *op* sont :

- `+` ajoute les *autorisations* pour *dest* ;
- `-` retire les droits *autorisations* pour *dest*.

Les valeurs principales de *droit* sont :

- `r` les droits en lecture ;
- `w` les droits en écriture ;
- `x` les droits en exécution.

Exemples :

- `chmod a+rx` donne à tout le monde les droits en lecture et en exécution.
- `chmod ug+w` donne à l'utilisateur le droit en écriture.
- `chmod go-wx` enlève aux autres (pas à l'utilisateur ni au groupe) les droits en écriture et en exécution.

e. *Caractères spéciaux*

Il existe des caractères spéciaux qui sont interprétés par le *shell*.

`*` remplace 0 ou n caractères quelconques. Exemple : `ls a*` liste tous les fichiers dont le nom commence par `a`.

`?` remplace 0 ou 1 caractère quelconque. Exemple : `ls *.py?` liste tous les fichiers dont le nom a pour extension `.py` ou `.py` suivi de n'importe quel caractère.

`[liste]` remplace 1 caractère de *liste*. Exemple : `ls *.[ch]` liste tous les fichiers dont le nom a pour extension `.c` ou `.h`.

Pour déspecialiser ces caractères il faut les mettre entre *apostrophes*. Exemple :

`rm '?'` efface le fichier de nom `?`, et non pas tous les fichiers dont le nom fait un caractère.

f. *Exécuter une commande*

Pour exécuter une commande comme `ifconfig` il suffit de taper son nom dans une fenêtre *shell*.

Par défaut le *shell* reste bloqué tant que la commande n'est pas terminée.

Pour arrêter la commande taper simultanément les touches `Control` et `C` (`Ctrl-C`).

Pour exécuter la commande en tâche de fond (ce qui permet d'utiliser le *shell*) il existe deux possibilités : soit taper le nom de la commande suivi de `&`, soit lancer la commande comme précédemment, interrompre la commande en tapant `Ctrl-Z` dans le *shell* et la faire passer en tâche de fond en tapant `bg` (*background*).

g. *Editer un fichier*

Pour visualiser un fichier les commandes de base sont `cat` et `more`. La commande `more` est beaucoup plus conviviale... Taper `h` pour avoir la liste des commandes disponibles.

Pour éditer un fichier il faut utiliser un programme comme `emacs` ou `vi`. `emacs` est *a priori* plus convivial. On utilise `vim` (*vi improved*) sous WSL avec Ubuntu dans le TD.

h. *Manuel*

Le manuel est accessible avec la commande `man`.

Exemple :

```
man tcsh
man man
```

Pour rechercher une commande dont on ne connaît pas le nom, `man -k` recherche un mot dans le descriptif de toutes les commandes.

Exemple :

```
man -k shell
man -k game
```

2. *Autres commandes*

Il y a beaucoup d'autres commandes importantes.

Deux exemples : nous en verrons éventuellement quelques-unes lors du cours sur les réseaux. Si l'on dispose d'une interface graphique, `xkill` permet de terminer un processus en cliquant dans sa fenêtre.

Une liste partielle peut être trouvée ici et dans les pages qui y sont liées :

https://doc.ubuntu-fr.org/commande_shell

Pour ceux qui sont très avancés, quelques exemples de commandes très puissantes ici :

<https://www.commandlinefu.com/commands/browse>


3. *Compléments sur WSL. Livrés sans garantie, non testés.*

Avoir du son sur WSL : <https://trzeci.eu/configure-graphic-and-sound-on-wsl/>.

Tester alors `apt-get install bb` puis `bb`. Le professeur n'est pas responsable d'une éventuelle crise d'épilepsie.

Avoir une interface graphique sous WSL :

- <https://korben.info/linux-wsl-gui-interface-graphique-windows-10.html> (la méthode la plus simple a priori)
- <https://doc.ubuntu-fr.org/wsl>
- <https://www.laurentbloch.net/MySpip3/Installer-un-environnement-graphique-X-Linux-et-de-programmation-sous-Windows-10-Windows-Services>

La méthode d'Antoine pour l'interface graphique (méfiez-vous c'est un ):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt install wget
```

wget <https://kali.sh/xfce> (attention : changer l'adresse, interface graphique pour Kali et non pour Ubuntu. Trouver xfce pour Ubuntu).

```
chmod u+x xfce.sh
./xfce.sh
sudo etc/init.d/xrdp start
```

Puis se connecter sous windows avec un bureau à distance, IP 127.0.0.1 :3390