

# Chap 16. Linux

Livre p 307 – Chap 24 Systèmes d'exploitation

## 1. Historique

### a. Systèmes d'exploitation

Vidéos d'introduction proposées par l'institut Mines-Télécom :

- <https://www.youtube.com/watch?v=4OhUDAtmAUo>
- <https://www.youtube.com/watch?v=bdSWj7Y50VY>

Le système d'exploitation est un programme qui sert d'intermédiaire entre les logiciels et la machine, il permet de gérer : la mémoire, les espaces de stockage, les périphériques d'entrée et de sortie, le réseau...

Quelques dates :

- 1960 – 1950 : On modifie directement la machine pour lui donner des instructions à effectuer.
- 1950 – 1960 : Les entrées/sorties s'effectuent à l'aide de cartes perforées, un programme interne alors est capable de déchiffrer ces cartes pour dire à l'ordinateur ce qu'il doit faire.
- 1960 – 1990 : Les premiers systèmes d'exploitation arrivent : Unix, MS-DOS, ...
- 1990 – ... : Une interface graphique vient améliorer l'utilisation des systèmes d'exploitation.

### b. Standards

Le système Unix a donné naissance à de nombreux dérivés : Linux, Android, mac-OS, i-OS, ...

Le système MS-DOS a donné naissance à Windows, les commandes ne sont pas les mêmes que Unix !

## 2. Shell

### a. Introduction

Aujourd'hui, tout le monde est habitué à interagir avec l'ordinateur, via une interface graphique, avec une souris ou directement avec le doigt !

Le Shell est une console qui permet d'écrire des lignes de commande au clavier. Moins facile à utiliser, il est pourtant plus puissant que l'interface graphique, mais moins intuitif...

Attention : La console est sensible à la casse, minuscules et majuscules sont des caractères différents !

### b. Structure des dossiers et fichiers

Tous les fichiers présents sur la machine sont rangés dans des dossiers sur un espace de stockage (disque dur, clef USB, ...). Tous les dossiers forment un unique arbre dont la racine est nommée "/".

- *pwd* : (Print Working Directory) Permet de savoir où on se trouve dans l'arborescence.  
Exemple : "/home/nicolas" signifie qu'on se trouve dans le dossier "nicolas", qui se trouve dans le dossier "home", qui est lui-même situé à la racine "/".
- *ls* : (LiSt) Permet de lister tous les dossiers et fichiers présent dans le dossier courant.  
Remarque : On peut y ajouter de nombreux arguments comme par exemple :
  - *ls -a* : Pour afficher aussi les fichiers cachés (dont le nom commence par un point)
  - *ls -l* : Pour afficher la liste détaillée avec la taille, les droits des utilisateurs, la date, ...
  - *ls -t* : Pour afficher la liste dans l'ordre chronologique. (*ls -alt* pour faire les trois !)
  - *ls nicolas* : On peut aussi lister le contenu d'un dossier.
- *cd* : (Change Directory) Permet de changer de dossier et se déplacer dans l'arborescence
  - *cd ..* : Pour remonter d'un niveau dans l'arbre.
  - *cd /* : Pour remonter directement à la racine de l'arbre.
  - *cd nicolas* : Pour aller dans le dossier "nicolas", à condition qu'il soit juste en dessous !

### c. Manipulation de dossiers ou fichiers

- *mkdir* : (MaKe DIRectory) Permet de créer un dossier.
- *cp* : (CoPy) Permet de copier un fichier (*cp fichier\_source fichier\_destination*)  
Attention : Pour copier un dossier, il faut utiliser l'argument *-r* (*cp -r nicolas vilboub*)
- *mv* : (MoVe) Permet de déplacer un fichier (*mv fichier\_source fichier\_destination*)  
Remarque : Cela permet aussi de renommer un fichier quand le dossier de destination est le même
- *rm* : (ReMove) Permet de supprimer un fichier  
Remarque : On peut aussi s'en servir pour supprimer un dossier non vide avec l'argument *-r*
- *rmdir* : (ReMove DIRectory) Permet de supprimer un dossier vide
- *cat* : (CA滕ate) Permet d'afficher le contenu d'un fichier au format ASCII  
Remarque : Si le fichier est très long on peut utiliser les commandes *head* ou *tail* pour n'afficher respectivement que le début ou que la fin du fichier...

Astuces : il n'y a pas vraiment de commande pour créer un fichier mais on peut utiliser :

- *touch* : Permet de modifier la date d'utilisation d'un fichier, mais en l'utilisant avec un nom de fichier n'existant pas, celui-ci est alors créé... vide ! (Exemple : *touch vide.txt*)
- *echo* : Permet d'afficher un texte dans la console, mais en redirigeant le texte vers un fichier n'existant pas, celui-ci est alors créé avec le texte dedans ! (Exemple : *echo salut > bonjour.txt*)

### d. Gestion des droits

Lorsqu'on affiche la liste détaillée des fichiers et dossiers, la première information visible sur la ligne est une suite de 10 caractères ressemblant par exemple à : "*-r w -r --r --*" ou "*d r w x r - x r - x*"

Le premier caractère est : "*d*" (Directory) pour un dossier, "*l*" (Link) pour un lien, ou "*-*" (autre chose).

Les 9 autres servent à gérer les droits, dans l'ordre suivant : pour l'utilisateur, le groupe, les autres.

On retrouve trois fois le triplet "*rw*" :

("-" pour une absence de droit)



- "*r*" : (Read) le droit de lecture
  - "*w*" : (Write) le droit d'écriture
  - "*x*" : (eXecute) le droit d'exécution
- *chmod* : Permet de modifier ces droits avec l'argument "*+w*" ou "*-w*" par exemple précédé de éventuellement de la cible "*u*" (User), "*g*" (Group), "*o*" (Other) ou même "*a*" (All) pour les trois.

### e. Caractères spéciaux et astuces

Lorsque l'on utilise une commande sur un dossier ou un fichier, on peut utiliser des caractères spéciaux :

- *\** : remplace entièrement n'importe quel texte (Attention : très dangereux lorsque l'on supprime)  
Exemple : *ls \*.txt* pour afficher la liste des fichiers texte.
- *?* : remplace n'importe quelle lettre.

Astuces : Dans la console, certaines touches peuvent vous simplifier la vie...

-  : (tabulation) Permet de compléter automatiquement la ligne avec le nom d'un fichier ou d'un dossier en cours d'écriture.
-  : (Flèche haut) Permet de recopier une ligne écrite précédemment en montant dans l'historique

## 3. M\$-Windows

### a. Invite de commande Windows

Windows possède lui aussi un mode console, on y accède avec l'application *cmd.exe*

Malheureusement, peu de commandes sont compatibles avec celles vues dans ce chapitre...

On peut donner ici quelques équivalences : *dir* ⇔ *ls*, *type* ⇔ *cat*, *del* ⇔ *rm*, *copy* ⇔ *cp*, ...

### b. Emulateurs en ligne ou à installer

Quand on travaille sous Windows, il faut soit installer un émulateur sur l'ordinateur, soit utiliser un

émulateur en ligne pour pouvoir tester les commandes du shell vues ici. (c'est ce que l'on fera dans le TP)

Si vous travaillez sur Mc-OS ou sous Linux, pas besoin d'émulateur... mais attention à ne pas faire n'importe quoi, vous risquez d'effacer des fichiers importants pour le système !!!