



---

# ADMINISTRATION LINUX

---

Support et mise à disposition d'outil informatique



22 AOUT 2023

[NOM DE LA SOCIETE]

[Adresse de la société]

## I. Origine

GNU / Linux a deux origines : le courant philosophique et le courant technique.

### a. Le courant philosophique

Richard STALLMAN, MIT, 1980.

Free Software Foundation (FSF) et GNU Public Licence (GPL)

### b. Le courant technique

Linus TORVALDS, Finlandais, 1991

Noyau Minix (mini unix), ouvert à tous

GNU est un projet d'UNIX libre. Objectif : rassembler et créer des logiciels libres pour les systèmes UNIX et assurer leur compatibilité.

La GPL protège et définit les droits et obligations de l'auteur d'un programme : mettre le code source à disposition de la communauté. Un tiers peut librement utiliser et modifier le code source mais ne peut pas se l'approprier ; il doit rediffuser dans les mêmes conditions. Ici le mot « libre » ne signifie pas forcément « gratuit » mais plutôt « non aliénable ».

### c. Composition de Linux

**Le noyau (kernel)** : c'est le programme qui assure la gestion de la mémoire, le partage du processeur entre les différentes tâches à exécuter et la gestion des entrées/sorties de bas niveau. Lancé au démarrage et jusqu'à l'extinction. Programme relativement petit chargé en mémoire centrale. Accès illimité au matériel : mode superviseur (administrateur).

Tous les programmes fonctionnent en mode utilisateur.

**L'éditeur de commande (shell)** : il sert d'interface entre le noyau et l'utilisateur. IL permet d'exécuter des commandes et de lancer des programmes. Il permet de réaliser des mini-programmes appelés « scripts ». Il existe plusieurs shell : **bash**, tcsh, ksh... Il fonctionne en mode texte. Il affiche une invite en début de ligne (le prompt) par exemple \$ ou # pour indiquer à l'utilisateur qu'il attend une commande

**Un éditeur de texte** : essentiel pour éditer et modifier des fichiers. Le plus simple VI, nano.

**Des langages de programmation** : C, C++, Pascal, Fortran, Java, Perl, Python, PHP.

**Des programmes (logiciels)**

**Des distributions** : au départ, il était difficile d'installer un linux : télécharger les sources, installer, compiler. On a inventé des distributions toutes prêtes.

Deux branches principales :

- Debian : Ubuntu, Edubuntu....
- Redhat : CentOS, Fedora, Mageia

## II. Procédure de connexion

### a. La connexion

Identifiant (login) + mot de passe (password)

La casse doit être respecté : toto différent de Toto

Si l'identification est valide, le prompt va s'afficher.

Répertoire personnel par défaut (« /home/{user} »)

Pour se déconnecter : exit

Un utilisateur peut avoir plusieurs connexions simultanées

### b. Les consoles

Linux offre par défaut 6 pseudo-terminaux appelés « consoles ». Pour passer de l'une à l'autre on utilise la combinaison de touches ALT + Fn avec n = 1 à 6. Le numéro de la console est affiché sur l'invite.

En mode graphique on utilise CTRL + ALT + Fn.

Les commandes **whoami** et **who** affichent les informations de connexion.

Deux types d'utilisateurs : superutilisateur « root » et les autres.

### c. Simuler un autre utilisateur

Objectifs : pour tester les procédures d'un autre utilisateur et pour devenir admin le temps d'une série de commandes.

Commande : **su login** (**su** équivalent à **su root**) / **exit** pour revenir

Commande **pwd** pour afficher le répertoire courant.

## III. Système de fichiers

### A) Introduction

Le filesystem permet d'accéder à toutes les informations du système. **Sous Linux, tout est fichier !**

Contrairement à MS-DOS on n'utilise pas de lettre de lecteur. Sous Linux, il y a seulement une seule grande arborescence : la racine « / »

#### a. Les répertoires

/ : racine

/bin : répertoire des principales commandes

/dev : répertoires des périphériques

/dev/tty : terminaux

/dev/lp : imprimantes

/dev/cdrom : lecteur DVD

/etc : fichiers de configuration

/home : dossiers utilisateurs

/usr : utilitaires

/usr/bin

/usr/man : manuels

/root : dossier personnel de l'administrateur

/tmp : fichiers temporaires

/var : données serveurs

#### b. Les fichiers normaux

Textes, images....

#### c. Les fichiers spéciaux

Pour communiquer avec les périphériques.

## B) Les permissions

Chaque fichier appartient à un propriétaire et un groupe avec des permissions sur 12 bits. Les permissions sont représentées par un triplet r, w, x (read, write, execute). Trois triplets existent : un pour le propriétaire, un pour le groupe et un pour les autres.

### a. Les effets de r,w,x

Pour les fichiers classiques : lecture, écriture (création, modification), exécution.

Pour les dossiers : voir le contenu (entrer dedans), écrire dedans, permet de le traverser et de percevoir les sous-dossiers.

Exemples :

Texte.doc: `rw_rw_r__`

Le propriétaire a le droit de lecture et écriture, le groupe aussi, mais les autres n'ont que le droit de lecture.

DossierPerso : `drwxr_x__` (d : directory)

Ce dossier a tous les droits pour le propriétaire, uniquement les droits de lecture et de traversée pour le groupe, et aucun droit pour les autres.

Autre notation :

R	W	X	R	W	X	R	W	X
4	2	1	4	2	1	4	2	1

`rwrxw_r__` : 764

`rw_r_____` : 640

### b. Les commandes pour les répertoires et les fichiers

Dossier :

`ls` : lister un dossier

`ls -a` : avec les fichiers cachés (commencent par un . )

`ls -l` : affichage avec les détails

`ls -la` : avec les deux précédents

`cd` : changer de dossier

`cd /home/thib` équivalent à `cd`

`cd /home/thib/perso` équivalent à `~/perso`

`pwd` : afficher le dossier courant

`mkdir` : créer un dossier

`rmdir` : supprimer un dossier

Fichiers :

`Cp` : copie

`Mv` : déplacer

`Rm` : supprimer

`Cat` : afficher

`Less` : afficher page par page

`Chmod` : pour changer les droits

`Chown` : changer le propriétaire

`Chgrp` : changer le groupe

`Find` : recherche de fichier

**Grep** : recherche de chaînes  
**Head/tail** : affiche le début/fin du fichier  
**Ln** : créer un lien symbolique  
**Ln -s** /var/www/html site (crée un raccourci site vers /var/www/html)  
**Sort** : trie les lignes d'un fichier  
**Wc** : compte le nombre de caractères  
Commandes d'archivage :  
Tar

#### IV. L'organisation d'un disque

##### a. Les partitions

Maximum 4 partitions primaires (ou principales)  
Une partition primaire peut être étendue  
Maximum 3 partitions logiques sur une partition étendue  
Sous Linux, à chaque partition correspond une entrée périphérique dans /dev :  
/dev/sda1  
/dev/sda2  
  
/dev/sdb1  
/dev/sdb2

##### b. Les types de systèmes de fichiers

Ext ou XFS, ReiserFS....  
Swap : pagination de la mémoire  
Pour utiliser une partition il faut la monter dans l'arborescence  
Par exemple : /dev/sda1 montée à la racine / ou /home ou /var...  
Commande mount/umount

#### V. l'éditeur VI

C'est l'éditeur commun de base de tous les systèmes Unix-Linux. Quand plus rien ne fonctionne, il reste vi !  
l'interface utilisateur se fait uniquement par le clavier.

Lancement : vi

Mode insertion : i

Mode commande : es

Commandes :

- : q pour quitter
- : w pour enregistrer
- : wq pour enregistrer et quitter
- : q ! quitter sans enregistrer

Difficile à prendre en main au début, mais très puissant. Préférez vim et nano.

## VI. les commandes de bases

### 1) généralités

- syntaxe : chemin/lin/commande -option arguments
- Si le chemin est dans le PATH, seul le nom de commande suffit. Écho \$path
- Si c'est une commande dans le répertoire current : ./commande
- aide : commande -h
- Page de manuel : man commande

### 2) manipulation de fichiers

- Touch pour créer un fichier vide
- Find pour trouver un nom de fichiers /chemin de départ pour trouver un fichier
- Locate pour trouver un fichier dans une base de données
- Updatedb pour mettre à jour la base.

### 3) les filtres et tubes

- Le filtre >> transmet la sortie cat fichier >> /dev/tty2
- Le tube | signale un filtre cat fichier|more
- Le filtre more permet la pagination.
- Le filtre grep pour chercher une chaîne.
  - Cat fichier|grep 'toto'|more
- Le pipeline lpr pour imprimer
  - Cat fichier|lpr

## VII. le processus

Certains processus sont utilisés en mode noyau (administrateur), mais la plupart sont exécutés en mode utilisateur et lui appartiennent. Un processus est créé lorsqu'on lance une commande. Il bloque la console jusqu'à sa fin lorsqu'il est en avant plan. Il libère la console s'il est lancé en arrière-plan.

>Firefox avant plan, console bloquée jusqu'à fermeture de Firefox

>Firefox & arrière-plan

Le processus 0, lancée au démarrage, est le père de tous les processus ; il appartient à root.

Les daemons sont des processus lancés en arrière-plan.

- Top liste les processus en cours.
- Kill n tue le processus n
- Ps liste les processus à un instant.
  - Ps -aux

## VIII. le shell

Élément très important : 99% de l'initialisation du système par des commandes. Savoir le maîtriser.

### 1) les variables

- Set pour visualiser toutes les variables du système
- Nom\_variable = valeur pour définir la variable
- Echo \$nom\_variable pour afficher le contenu de la variable
- Unset nom\_variable pour supprimer la variable

La variable PATH détermine le chemin de recherche des commandes.

Exemple : grep PATH = /etc/\*

La variable PS1 contient le prompt du système. Par défaut, les variables sont locales (accessible dans le script en cours). Il faut les exporter pour qu'elles deviennent des variables publiques.

Export nom\_variable

## 2) Écriture d'un script

Il faut créer un fichier texte et le rendre exécutable.

-soit on lance le script avec l'interpréteur :

```
Bash nom_du_script.sh
```

-soit on inscrit l'interpréteur sur la première ligne du script qu'on a rendu exécutable :

```
# ! /bin/bash : script.sh
```

```
Chmod u+x script.sh
```

```
ls -l
```

```
script.sh -rw_r_____4ko
```

```
chmod 751 script.sh
```

## IX. les paquets

Configure les sources de la distribution. Mises à jour, installation, désinstallation.

Quelle branche : -Redhat : [paquets.rpm](#) gestionnaire rpm ou yum

-Debian : [paquets.deb](#) gestionnaire apt

Exemple : `apt install libreoffice`