

# Introduction au Système d'Exploitation Linux

## Plan:

- 1) Chapitre I : Introduction et Historique
- 2) Chapitre II : Le système Linux
- 3) Chapitre III : Prise en main du Système Linux

Réalisé par : BOUDAA Boudjema  
2009/2010

## Chapitre 1

### Introduction & Historique

#### 1. Système d'exploitation (OS)

##### 1.1 Définition:

Le système d'exploitation est le logiciel le plus important d'un ordinateur. C'est un ensemble de programmes qui permettent l'utilisation de l'ordinateur et la gestion de ses ressources (processeurs, mémoires, disques, périphériques, communication inter-processus et inter-machines, ...etc.).

##### 1.2 Exemple de système d'exploitation :

Windows, Linux, Unix, MS-Dos, MacOs, etc.

#### 2. UNIX

##### 2.1 Historique

- Unix est né aux laboratoires Bell en 1969, Développé par **Ken Thompson et Dennis Ritchie** (le premier à avoir développé le langage C).
- En 1973, Unix a pu être réécrit, presque entièrement, en C ( pour faciliter la Portabilité)
- Code source vendu à un prix bas aux sociétés.
- Plusieurs sociétés ( IBM , Sun ,...) se sont intéressées au système et elles ont repris son développement pour avoir leur propre version ( Solaris:Sun , AIX:IBM, HP-UX: HP, FreeBSD:Université de Berkely ,...)

##### 2.2 Caractéristiques d'Unix :

- Basé sur le principe tout est fichier.
- multi-tâches en temps partagé
- multi-utilisateurs
- Utilisation du **Shell** comme interpréteur de commandes
- La configuration du système est stockée sous forme de texte
- Disponibilité sur un large gamme d'architecture matérielle (du PC jusqu'au Super calculateur massivement parallèle)
- C'est le système le plus utilisé (Dans les universités, les centres de recherches, les serveurs d'Internet, ...)
- Unix produit commercial (système payant)

**Remarque :** L'objectif des chercheurs dans le temps était de rendre UNIX accessible sans frais.

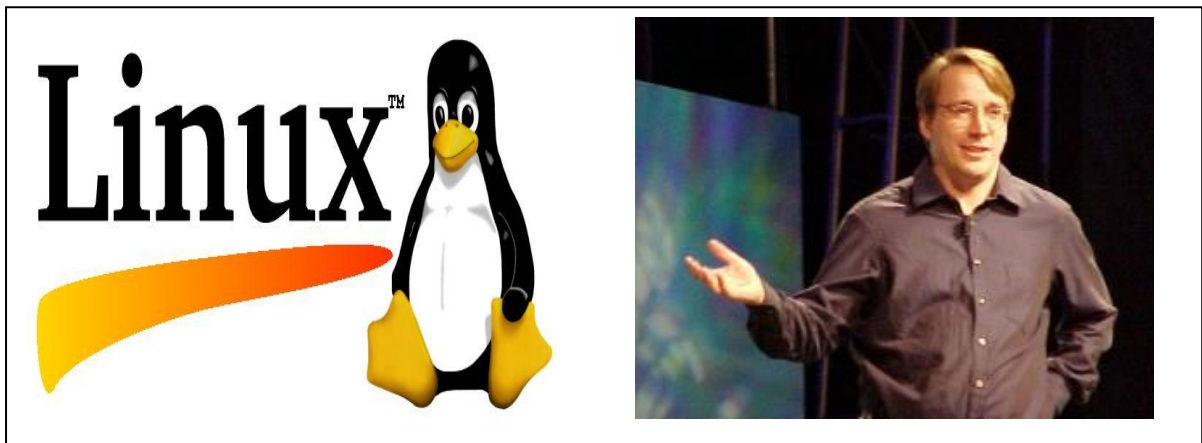
### 3. **LINUX** : Linux est une version librement diffusable de l'Unix.

#### 3.1 **Historique**

- En 1991, En Finlande un autre noyau était disponible : c'est « Linux » par **Linus Torvalds** (jeune étudiant à l'université d'Helsinki, né le 28/12/1969)
- Des milliers de personnes participent à son développement
- Code source est disponible sur Internet (logiciel libre et gratuit)

**Remarque** : Le nom LINUX peut signifier, au choix :

- LINUs uniX
- Linux Is Not UniX



#### 3.2 **Logiciel libre**

Logiciel libre = Logiciel Open Source

Un logiciel libre est un logiciel dont la licence prévoit:

- La disponibilité des codes sources.
- La possibilité de modifier, améliorer, adapter le logiciel sous réserve que ces modifications soient rendues accessibles à toute personne intéressée.
- La possibilité de copier et de diffuser le logiciel sous réserve que les termes de la licence ne soient pas modifiés.

#### **Exemples de logiciels libres**

Linux, OpenOffice , Apache, Sendmail, MySQL, PostgreSQL, gcc, PHP, ...

#### 3.3 **Distributions de Linux**

Le noyau (ou kernel) développé par Linus Torvalds est l'élément essentiel de toutes les **distributions Linux** existantes.

Chaque distribution essaye d'offrir de la valeur ajoutée sous la forme des outils d'**installation** et d'**administration**.

**Une distribution Linux** = noyau + outils d'installation + outils d'administration  
+ un ensemble de logiciels d'application

### **3.4 Exemples des distributions**

- Red Hat Linux (USA [www.redhat.com](http://www.redhat.com)) [05 CD]
- Mandrake Linux (France [www.linux-mandrake.com](http://www.linux-mandrake.com))
- SuSE Linux (Allemagne [www.suse.com](http://www.suse.com))
- Debian (Internet [www.debian.org](http://www.debian.org)) [23 CD d'installation]
- Fedora:([www.fedora.org](http://www.fedora.org)) [ version libre de Red Hat ]
- Ubuntu([www.ubuntu.com](http://www.ubuntu.com)) [version de debian]
- KNOPPIX, slackware, Mandriva...

### **3.5 Domaines d'utilisation de Linux ?**

- Station de travail : Multimédia et bureautique (openoffice, koffice,...)
- Réseaux et Internet : serveur Web (Apache), messagerie (sendmail), Explorateur (FireFox de Mozilla)....
- Développement : C/C++, Delphi, Java, PHP,...
- SGBD ( Oracle, Informix, MySQL, PostgreSQL,...)
- Recherche scientifique
- ...

## Chapitre 2

### Le Système Linux

#### 1. Accès à la Machine sous Linux

Le système linux étant un système multi-utilisateur, pour y accéder on doit entrer un nom et mot de passe utilisateur.

##### a) **Système Multi-Utilisateur :**

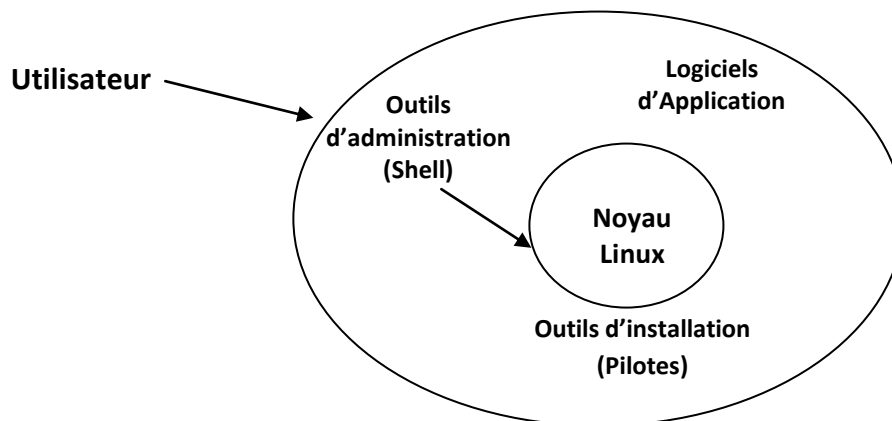
- Utilisateur simple : droits restreints et chaque utilisateur possède un répertoire de travail.
- Super Utilisateur ou Administrateur (root) : tous les droits.

##### b) **Organisation d'utilisateurs en groupes:** chaque utilisateur doit appartenir à un groupe

##### c) **Connexion au système :** login/passaword requis et la connexion se fait de deux mode différents :

- Mode texte ou console (exécution des commandes Shell)
- Mode graphique (à l'aide du serveur de graphisme X-Window)

#### 2. Position du noyau Linux dans une distribution



- Noyau Linux : ensemble de programmes pour gérer la machine et ses ressources
- Shell : Interpréteur de commandes
- Pilotes: Drivers des périphériques (disques durs, carte graphique, carte réseau, imprimantes,...)
- Logiciels d'applications : différents logiciels
  - X-Window : pour fournir un environnement de graphisme
  - KDE et Gnome : pour gérer l'interface graphique (fenêtres,...)
  - ...

#### 3. Le Shell :

C'est l'interface entre les utilisateurs et le système. Le Shell est un interpréteur qui exécute les commandes une à une après traduction de l'instruction (écrit en langage évolué) en langage interne (langage machine).

### 3.1 Versions du Shell : sh, ksh, bash, csh, ...

## 4. La distribution Red Hat:

- a) **Connexion en mode graphique** : voir interface graphique
- b) **Connexion en mode texte** : Si les login/password sont valides alors un message de la forme suivante sera affiché:

```
[user@machine ~]$
```

- user: le nom de l'utilisateur connecté
- machine : le nom de la machine
- ~ : caractère spécial désigne le répertoire de travail de l'utilisateur connecté
- Le signe \$ : indique que vous êtes en un simple utilisateur
- Le signe # : indique que vous êtes en mode super-user.

## 5. L'invité de commandes (utilisation du Shell)

### 5.1 Syntaxe d'une commande :

**Commande [option] [paramètres]**

**Option** : non nécessaires pour exécuter la commande

**Paramètres** : arguments nécessaires pour exécuter la commande

- Les trois champs sont séparés par des espaces
- Plusieurs commandes sur la même ligne : séparer par ';' ;

### 5.2 Quelques commandes :

- **date** : Afficher la date et l'heure
- **cal** : afficher un calendrier
- **uname** : afficher le nom et les caractéristiques du système.
- **passwd** : modifier son mot de passe
- **man** nom-commande :
  - manuel pour afficher une page d'aide (forme de commande, option,...)
  - exemple : **man man** : plus d'information sur l'utilisation du man

**Remarque** : **commande --help** ou **commande -h** : une option commune à la majorité des commandes pour afficher l'aide (exp : ls --help, man -h)

- **adduser** user1 : ajoute l'utilisateur user1
- **passwd** user1 : pour donner ou changer le mot de passe user1
- **addgroup** grp1 : ajoute le groupe grp1
- **adduser** user1 grp1 : ajoute l'utilisateur user1 au groupe grp1
- **groupadd** grp2 : crée un groupe grp2
- **chgrp** grp2 user1 : le groupe de user1 est maintenant grp2
- **userdel** : pour supprimer un utilisateur ;
- **groupdel** : pour supprimer un utilisateur ;
- **adduser --ingroup** grp1 user1 : crée un utilisateur user1 dont le groupe principal est grp1  
(consulter les fichiers : **/etc/passwd, etc/group, etc/shadow**)

- **su user1** : bascule vers l'utilisateur user1
- **su -** : bascule vers le super-utilisateur (administrateur)
- **ifconfig eth0 192.168.10.10** : configure la carte réseau ethernet eth0 et lui affecte l'adresse indiquée.
- **mount -t vfat /dev/hdc1 /mnt/hd1** : montage de la partition 1 du disk 3 sur le répertoire /mnt/hd1, si le point de mount est déclaré dans /etc/fstab, l'utilisateur peut le monter, s'il lui est permis. Autrement il faut être (**umount** pour démonter)

**Exemple** : pour lire un CDROM

- Consulter le fichier fstab (par **cat /etc/fstab**) pour savoir si le point de montage (dans notre cas, c'est /dev/hdc)
- Créer un répertoire : **mkdir tt** (tt répertoire pour contenir le point de montage)
- Monter par : **mount /dev/hdc tt**
- Accéder au répertoire tt pour utiliser votre CDROM

**Exercice** : comment monter un flash disk ?

**N.B** : il est déconseillé de se logger en superutilisateur (root). Si à un moment donné nous avons besoin de lancer une commande d'administrateur, il suffit d'utiliser « su → » (switch user), qui permet de passer momentanément en root.

## Kernel

Le "**kernel**" ou noyau (en Français) c'est ce que l'on désigne par "**Linux**".

Linux c'est le noyau, c'est à dire la base du système qui a vu le jour en 1991 grâce à un ingénieur étudiant, **Linus Torvalds**.

Linus B. Torvalds, pour étudier la programmation du processeur Intel **80386** avec un PC et un système **Minix** (clone d'UNIX pas cher), a entrepris de réécrire une partie du code du noyau Minix qui est devenu le **noyau Linux**.

Si le terme **Linuxien** est très souvent employé pour désigner tout ce qui se rapporte à Linux, vous rencontrerez aussi le terme **Linusien** en référence à Linus Torvalds .

## GNU

**Gnu is Not Unix**, projet fondé en 1984, par **Richard Stallman** (RMS), pour restituer aux utilisateurs la liberté de copier, modifier et redistribuer un logiciel . Un logiciel GNU, que l'on nomme souvent application sous Linux, se voit très souvent ajouter un **g** devant son nom

La Debian par exemple reste du domaine de l'expert plus que de celui du débutant, mais cette distribution est 100% GNU et devrait être appelée Debian GNU Linux (voir plus bas).

## GPL

**General Public License**, c'est la licence qui régit le principe fondamental de l'esprit **GNU Linux**.

Les utilisateurs des applications (logiciels) placés sous licence GPL peuvent dupliquer et redistribuer ces applications, mais aussi modifier le code source qui est fourni gratuitement avec l'application (sous forme d'archives).

## Packages

### Tarball, RPM ou DEB :

Sous Linux, une application est distribuée sous forme d'archive qui contient l'application, ses bibliothèques, ses utilitaires et fichiers de configuration :

Tous ces fichiers ont été compressés pour former une archive tarball qui peut être installée sur n'importe quelle distribution.

Ce format est repérable par son nom, le nom de l'application suivi de sa version et de l'extension **tar.gz** ou **tgz** .

Vous trouverez également des archives avec l'extension **bz2**, **tar** ou **z** .

Cette archive peut contenir soit les sources pour être compilées sur votre machine et/ou modifiées, soit les binaires (exécutables) qui ne nécessitent pas de compilation .

Pour faciliter l'installation ces archives ont été adaptées à l'arborescence de la distribution et sont présentées sous forme de **packages** (certains disent paquets) et ont l'extension **rpm** pour les distributions à base de RedHat et pour SuSE ou l'extension **deb** pour la distribution Debian (qui peut également installer des RPMs).

### Montage

Pour accéder à vos données, il faut que la partition ou le périphérique comme le CDRom soit déclaré et affecté à un point de montage en fonction de son type de système de fichiers (FS) ?

Cela veut dire que, le système de fichiers "**ext2fs**" de la partition "**hda2**" contenant "**Linux**" sera monté à la racine "/",

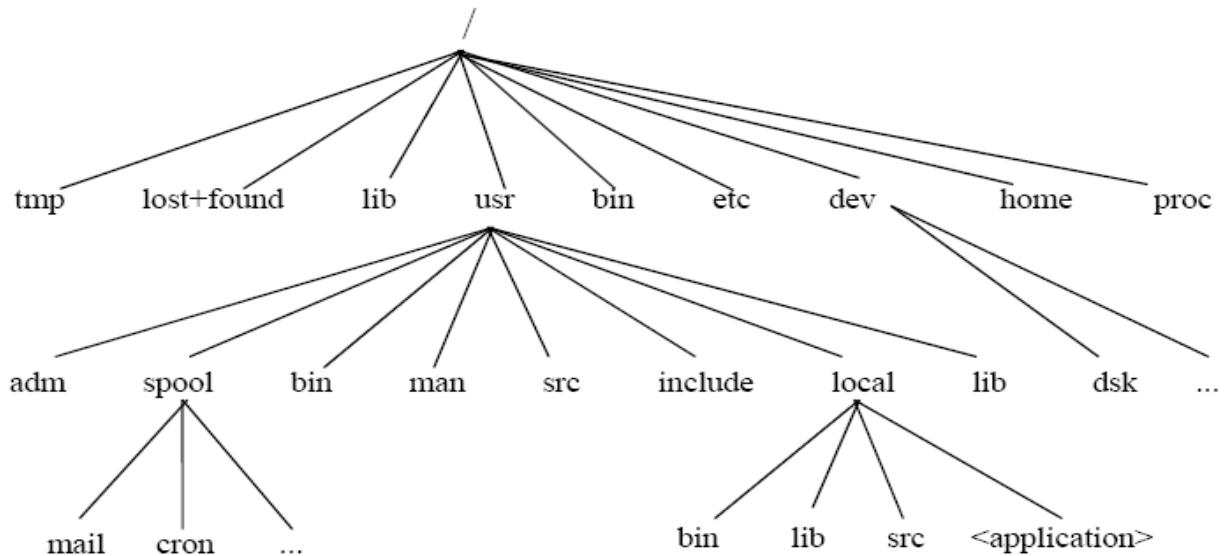
le système de fichiers "**iso9660**" du "**CDRom**" sera monté en **"/mnt/cdrom"**.



## Chapitre 3

### Prise en main du système Linux

#### 1. Arborescence du système de fichiers :



- Structure hiérarchique des fichiers et répertoires
- La racine est dénoté par `^`
- Les chemins sont séparés par `^`
- Les noms des objets sont des séquences de maximum 255 caractères sensibles à la casse.
- Il est préférable de ne pas utiliser les caractères : ?, \*, &. `', ", <, >
- Il convient de se limiter à : - A à Z , a à z, 0 à 9, le caractère de soulignement ` \_ `, le tiret ` - ` et le point ` . `
- Le point ( . ) comme premier caractère d'un nom signifie un fichier caché.
- Éviter des noms contenant des caractères accentués ou des blancs.
- Le point ( . ) ne sépare pas forcément le nom de son extension  
(Exemples : rapport.ps.gz, postgresql-2.3.6-src.tar.Z, README)

#### 1.1 Les principaux répertoires :

- /boot : contient les fichiers nécessaires au démarrage du système.
- /bin : contient les commandes de base (binaire)
- /sbin : contient les commandes du super utilisateur (administrateur)
- /usr/bin, /usr/sbin, ... : contient des fichiers binaires
- /etc : contient les fichiers de configuration du système et des services
- /home : contient les répertoires personnels des utilisateurs simples
- /root : contient le répertoire personnel de l'administrateur.
- /usr : contient le reste des programmes du système et les logiciels.
- /dev : contient les noms périphériques.

- /mnt/ , /media/ : contient les points de montage d'autres systèmes de fichiers.
- /proc : est un répertoire virtuel, n'a pas d'existence sur disque, contient les informations sur les processus. Faire par exemple `cat /proc/meminfo`, pour des informations sur la mémoire.

## 2. Quelques commandes de base :

### La commande `man`

**syntaxe** : `man [options] nom_de_commande`

**description** : Elle permet d'afficher un manuel en ligne sur la commande spécifiée. Cet affichage s'effectue en mode console. La commande `q` permet de quitter le manuel.

**options** : `-h` : affiche l'aide concernant la commande **man** et donc permet notamment de savoir que `q` permet de quitter.

`-w` or `--path` : affiche le lieu où est stocké le fichier contenant le manuel de la commande.

### La commande `ls`

**syntaxe** : `ls [options] [chemin_d'un_repertoire]`

**description** : Affiche l'ensemble des fichiers passés en argument puis la liste des fichiers contenus dans les répertoires passés en argument. Par défaut, elle affiche le répertoire courant ".".

**options** : `-l` : permet d'obtenir des informations détaillées sur chaque fichier listé (date, taille, droits, etc.).

`-a` : permet d'afficher tous les fichiers contenus dans un répertoire, y compris les fichiers commençant par un point (les fichiers cachés).

`-R` : permet d'afficher récursivement le contenu des sous-répertoires.

`--color` , `--color=yes`: permet d'afficher les fichiers en couleur selon leur type.

`--color=no` : annule l'affichage en couleur des fichiers.

**exemples** : `ls .` : lister le répertoire courant

`ls ..` : lister le répertoire parent

`ls -lrt /etc` : Donne la liste des fichiers et répertoires de /etc avec les détails (option `-l`)  
trie par temps (`-t`) en ordre inverse (`-r`)

`ls -l > f1.txt`: le symbole « > » redirige la sortie de « ls » vers le fichier `f1.txt` avec écrasement. Si on utilise « >> » il y aura ajout à la fin (append).

### La commande `cd`

**syntaxe** : `cd [chemin]`

**description** : Elle permet de se déplacer dans l'arborescence du système de fichier.

Le chemin peut-être absolu ou relatif.

**exemples** : `cd ..` : permet de remonter d'un cran dans l'arborescence.

`cd` ou `cd ~` : permet de se placer directement à la racine de votre répertoire de travail.

`cd -` : pour retourner au répertoire précédent

`cd /usr/local` : déplacement selon un chemin absolu.

`cd bin` : déplacement selon un chemin relatif.

**remarques** : - chemin absolu : chemin d'accès complet à partir la racine (/)

Par exemple : `/usr/local/seminaires/apprentissage/text1`

- chemin relatif : spécifier le chemin d'accès relativement au répertoire courant

Par exemple : `seminaires/apprentissage/text1` représente le nom relatif de `text1`  
par rapport au répertoire courant `/usr/local`

**La commande mkdir**

**syntaxe** : mkdir [options] repertoire

**description** : Elle permet de créer un nouveau répertoire.

**options** : -p, --parents : s'assure que chaque répertoire spécifié existe et crée les répertoires parents manquants.

**exemples** : mkdir -p Informatique/MuPAD : permet de créer le répertoire Informatique et le sous répertoire MuPAD.

**La commande pwd**

**syntaxe** : pwd

**description** : Elle permet de connaître le répertoire courant.

**La commande rmdir**

**syntaxe** : rmdir [options] répertoires

**description** : Elle permet de supprimer un ou plusieurs répertoires vides.

**options** : -p, --parents : efface les répertoires parents s'ils deviennent vides.

**La commande rm**

**syntaxe** : rm [options] nom

**description** : Elle efface chaque fichier spécifié et par défaut n'efface pas les répertoires.

**options** : -f, --force : efface les fichiers en ignorant ceux qui n'existent pas et en ne demande de confirmation à l'utilisateur.

-i, --interactive : demande à l'utilisateur de confirmer chaque suppression.

-r, -R, --recursive : supprime récursivement les contenus des répertoires et le répertoire lui-même.

**La commande cp**

**syntaxe** : cp [options] fichier1 fichier2

cp [options] fichier repertoire

**description** : Elle sert à copier des fichiers et éventuellement des répertoires depuis un endroit précis vers une destination précise ou un répertoire.

**options** : -i : interroge l'utilisateur avant d'écraser les fichiers existants.

-R : copie récursivement les répertoires et gère correctement les fichiers spéciaux.

-f : force l'effacement des fichiers cibles existants.

-p : conserve le propriétaire, le groupe, les permissions d'accès, et les horodatages du fichier original.

**La commande mv**

**syntaxe** : mv [options] source destination

mv [options] source repertoire

**description** : Elle sert à déplacer ou renommer les fichiers. Si le dernier argument est un nom de répertoire alors tous les fichiers sources seront déplacés, en conservant leur nom, vers ce répertoire sinon il déplacera le premier pour remplacer le second.

**options** : -i : interroge l'utilisateur avant d'écraser les fichiers existants.

-f, --force : écrase les fichiers de destination existants sans demander de confirmation à l'utilisateur.

-u, --update : ne pas déplacer un fichier régulier qui écraserait un fichier destination existant ayant une date de modification plus récente.

**La commande touch**

**syntaxe :** touch nom\_fichier

**description :** Elle permet de créer un fichier.

**exemple :** touch linux.txt

```
ls
linux.txt
```

**La commande file**

**syntaxe :** file nom\_fichier

**description :** Elle permet de connaître le type d'un fichier. L'extension des fichiers n'est pas toujours utilisée.

**La commande cat**

**syntaxe :** cat fichier1 fichier2...

**description :** défiler à l'écran le contenu d'un ou plusieurs fichiers.

**remarques :** - <ctrl-c> : Mettre fin au défilement

- <ctrl s> : Interrompre le défilement avec possibilité de reprise

- <ctrl q> : Reprendre le défilement

**exemple :** cat /etc/passwd | grep -v bash: Le symbole « | », pipe permet de récupérer la sortie d'une commande pour l'envoyer en entrée dans la suivante. Ici le contenu du fichier passwd est récupéré puis filtré avec **grep** pour garder les lignes ne contenant pas le mot « bash ».

cat /etc/profile > ~/confenv : pour copier le contenu du fichier profile dans le fichier confenv de son répertoire HOME.

Ls \*.txt | xargs grep -l ce\_mot : rechercher les fichiers ayant l'extension txt qui contiennent le texte ce\_mot

cat \*.txt | grep ce\_mot : rechercher les lignes contenant ce\_mot dans tous les fichiers ayant l'extension txt.

**La commande more**

**syntaxe :** more fichier

**description :** défiler le contenu une page d'écran à la fois.

**remarques :** - taper la barre d'espace : Défiler la page suivante

- taper <entree> : Défiler une ligne supplémentaire la touche

- taper la lettre q : Quitter la commande **more**

**Les commandes less, head et tail**

**Less :** similaire à la commande more.

**head :** afficher les dix premières lignes d'un fichier

**tail :** les dix dernières.

**exemple :** tail -15 file.txt: Donne les 15 (les 10, si rien n'est indiqué) dernières lignes du file.txt, la commande **head** est pareille pour le début du fichier.

**Les commandes nl et wc**

**nl :** permet la numérotation des ligne d'un fichier donné

**wc :** permet le calcul du nombre de ligne, de mot et de caractère contenu dans un fichier.

**exemples:** - nl /etc/passwd

- wc /etc/mail/sendmail.mc

**Commandes pour les process (processus : programme en exécution)**

**ps aux** : liste tous les process qui tournent sur la machine.  
**kill -9 2345**: tue le process dont le numéro est 2345, (s'il vous appartient).  
**top** : donne l'utilisation des ressources mémoire et CPU, en réel.  
**free** : affiche les informations sur la mémoire

**Commandes pour éditer les fichiers**

**nano** nom\_fichier  
**vi** nom\_fichier  
**emacs** nom\_fichier ( voir aussi **gedit** )

**Commandes pour modifier les droits d'accès**

**chmod** : change les droits des répertoires et fichiers (chmod xxx)  
**chown** : change le propriétaire et le groupe  
**exemples : chmod o+w lmd**: permet de changer les permissions, ici par exemple on ajoute « w », permission d'écriture aux autres (other), g-x : enlève la permission d'exécution au groupe. ou bien avec des valeurs par exemple  
**chmod 756 lmd veut dire rwxr-xrw-** pour lmd.  
**chown lmd :mongroup f1.txt**: change le propriétaire et le groupe du fichier f1.txt deviennent lmd et mongroup.

**Autres commandes utiles**

**echo "message"**: renvoie l'écho message sur écran.  
**echo \$HOME**: donne le contenu de la variable d'environnement HOME (attention au \$, qui doit précéder toute variable shell).  
**echo \$PATH** : indique les chemins déclarés pour accéder aux bibliothèques et binaires.  
**locate**: localise un fichier dans la base de données des fichiers  
**find -name nom\_fich** : recherche le fichier nom\_fich  
**gzip fich.txt**: compresse le fichier fich.txt et sera fich.txt.gz. Pour l'inverse, utiliser **gunzip**.  
**tar -zvf dir.tgz ladir**: tarez et compresse les fichiers de la directory « ladir » sous le nom dir.tgz, avec x : pour extraire et t pour lister ou bien tester  
**su**: switch user, on peut aussi utiliser « sudo ».  
**df**: disk free, donne la taille des partitions montées, option -h : human readable (lecture appropriée)  
**du /home**: disk used, donne la utilisée par le répertoire /home. On peut aussi utiliser l'option -h  
**tty**: donne le terminal en cours d'utilisation. Le premier est /dev/pts/0, ensuite /dev/pts/1, ...etc , on peut faire sortir le résultat d'une commande dans un autre terminal. Par exemple si on est dans terminal 1, alors ls > /dev/pts/2 sort le résultat dans le deuxième terminal (console).  
**who** : indique qui est connecté et sur quelle console (tty)  
**tree -d -L 2 / > tree.txt**: permet de sortir l'arborescence du système et de stocker le résultat dans le fichier tree.txt  
**diff f1 f2**: donne la différence entre les fichiers f1 et f2. diff3 : donne la différence entre 3 fichiers.  
**Le fichier caché « .bash\_history »**: Contient l'historique des commandes exécutées, périodiquement mis à jour. les éléments les plus anciens sont supprimés. La même information peut être obtenue avec la commande **history**.  
**ps2pdf**: Convertit le fichier du format ps en format pdf. **pdf2ps** fait l'inverse.  
**convert f1.gif f2.jpg** : convertit le fichier image de format gif en image jpeg. Les extensions déterminent le type de conversion à faire. Cet outil fait partie du logiciel imagemagik.

### 3. Les liens :

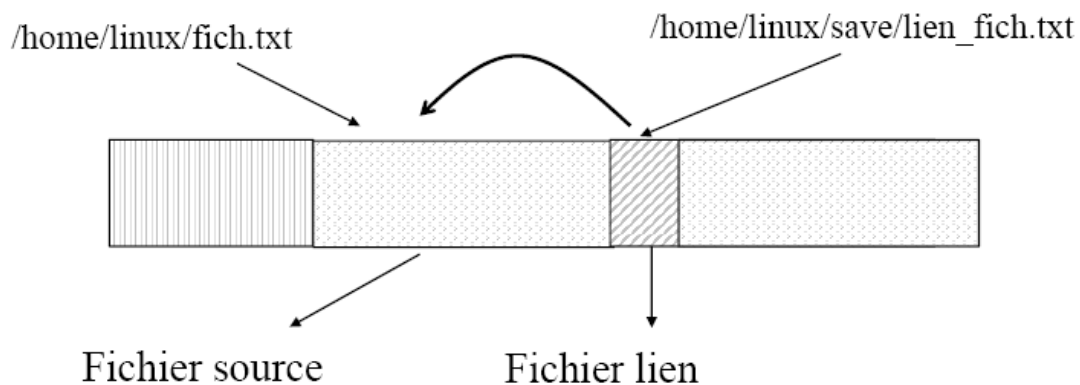
Les liens sont utiles pour faire apparaître un même fichier dans plusieurs endroits, même avec des noms différents.

**3.1 Les types des liens :** il existe deux types de liens, à savoir :

- a) Lien symbolique
- b) Lien physique

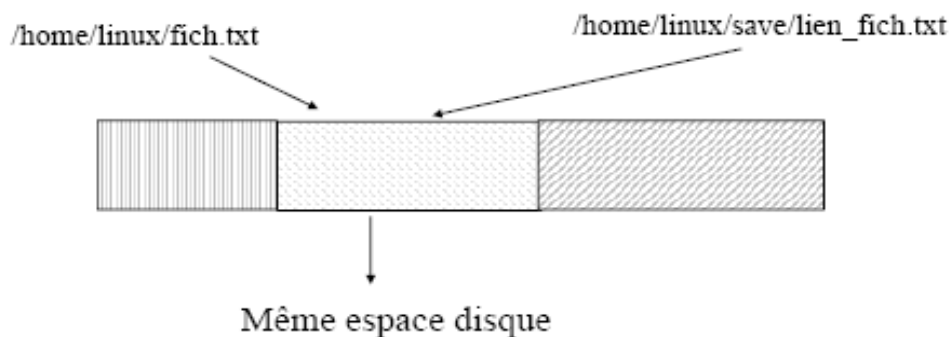
#### 3.1.1 Les liens symboliques :

- Fait référence à un fichier dans un répertoire.
- Si suppression du fichier source alors le lien sera considéré comme "cassé".
- Utile dans le cas des fichiers binaires (commandes)



#### 3.1.2 Les liens physiques :

- Associe deux ou plusieurs fichiers à un même espace disque
- Les deux fichiers restent indépendants.
- Le fichier sera supprimé seulement si tous ces liens sont supprimés
- Utile dans le cas des fichiers de données



**3.2 Comment créer un lien :** à l'aide de la commande « ln »

#### La commande ln

**Description :** c'est la commande de création des liens

**Syntaxe :** ln [option] source lien

- Cas des liens physiques : ln linux.txt linux1.txt /\*linux1.txt est un lien physique\*/
  - Cas des liens symboliques : ln -s linux.txt linux1.txt /\*linux1.txt est un lien symbolique\*/
- ln -s /home/linux/linux.txt rep/linux1.txt

#### 4. Questions

- (1) Donner le nombre de répertoires ou de fichiers, ou liens de la directory /usr/lib, lister les fichiers.
- (2) Reprendre la question précédente et donner la taille totale des fichiers.
- (3) Dans quelle partition vous êtes, combien y en a-t-il dans votre système, quelle est la taille de votre installation ?
- (4) Quelle est la taille de la mémoire étendue utilisée de votre machine.
- (5) Quelle est votre chemin (PATH), compléter le avec le répertoire courant, faire l'export vers les autres terminaux?, quel est votre répertoire courant ? Rendre votre répertoire accessible uniquement à votre groupe, pas de lecture, ni d'écriture.
- (6) Prendre votre nouveau PATH, donner un message à chaque connexion (nouveau shell)

#### Références:

- Cours de Mme Yahiaoui ([www.esi.dz](http://www.esi.dz))
- Administration et réseau sous Linux de Mr. A.Belahcene et S.Khelifati