

# **Comprendre les concepts sous-jacents à un système de type Unix tel que Linux**

—+—

## **Une comparaison avec l'existant sous Windows.**

# 1. Un système d'exploitation

Afin de comprendre l'informatique, il faut savoir ce qu'est un système d'exploitation, ce qu'il contient et à quoi il sert.

Un système d'exploitation est composé de plusieurs éléments : un *noyau*, des commandes ou applications, des fichiers. Le noyau est l'élément central d'un système d'exploitation. Il fait la liaison entre le matériel et les applications en permettant aux applications d'accéder au matériel (disques, imprimantes, écran, clavier/souris, réseau, etc.) et supporte de nombreux protocoles de communication ; il gère aussi les interactions entre les applications (communications, ordonnancement des tâches) mais aussi les droits des utilisateurs en fonction de leur niveau d'autorisation.

## 1.1. Liberté

Il n'est pas possible d'aborder Linux sans parler de Liberté. Linux et la plupart des applications qui composent un système d'exploitation Linux sont des logiciels dits *logiciels libres*. C'est-à-dire que les sources des applications (et de Linux donc) sont librement disponibles, modifiables et (ré)utilisables. Une définition du *logiciel libre* est basée sur quatre libertés :

- Liberté 1 : La liberté d'exécuter le programme – pour tous les usages ;
- Liberté 2 : La liberté d'étudier le fonctionnement du programme – ce qui suppose l'accès au code source ;
- Liberté 3 : La liberté de redistribuer des copies – ce qui comprend la liberté de vendre des copies ;
- Liberté 4 : La liberté d'améliorer le programme et de publier ses améliorations – ce qui suppose, là encore, l'accès au code source.

## 1.2. Mode de développement

Le mode de développement de Linux et des logiciels libres est particulier. Les systèmes comme Windows sont développés par des équipes de développeurs/programmeurs travaillant chez Microsoft. En ce qui concerne les logiciels libres, le développement est fait par des équipes diverses et variées, éparpillées de par le monde et dont les membres sont eux aussi éparpillés un peu partout.

Si au début le développement de Linux et des logiciels libres était principalement le fait de bénévoles, depuis quelques temps de grandes entreprises (comme IBM, Intel, Oracle et bien d'autres) ont pris à leur charge des équipes de développement afin d'implémenter les fonctionnalités qui leur sont nécessaires. Les développeurs bénévoles restent cependant encore prépondérants.

### **1.3. Présentation physique**

Un système d'exploitation Windows ou MacOS est vendu sur CD ou DVD. Il n'y a qu'un seul Windows et qu'un seul MacOS chacun d'eux ayant plusieurs versions en fonction des utilisations cibles (personnelle, entreprise, serveur).

En ce qui concerne Linux c'est totalement différent. Linux n'est qu'une partie du système d'exploitation : le noyau. Cependant on nomme par extension « Linux » tout système d'exploitation basé sur le noyau Linux (ou presque, eg : Android). On obtient Linux sous forme de CD ou DVD avec le reste des éléments du système d'exploitation. Ces CD et DVD sont créés, assemblés par des équipes de développeurs pour former ce qu'on appelle une *distribution*. De par la nature libre de Linux et des applications qui composent le système d'exploitation, il existe des dizaines, des centaines de distributions Linux. Une poignée d'entre elles sont des distributions grand public et généralistes.

## **2. Le paradigme de base**

L'informatique est la science de l'information. Ce que nous gérons sont des informations, sous forme de fichiers. Sous Windows, les fichiers sont limités au stockage des données et des programmes. Sous Linux, cette vision est étendue, généralisée. Le paradigme de base sous Linux est que tout est fichier (ou presque). Nous reviendrons souvent sur cette notion dans la suite de ce document car c'est un concept de base des systèmes Unix.

## **3. Les disques et les systèmes de fichiers**

### **3.1. Les systèmes de fichiers**

Un système de fichier est un type d'organisation des fichiers sur le disque. Il existe de nombreux systèmes de fichiers. Sous Windows, il existe les variantes FAT (FAT16, FAT32 ou VFAT), NTFS, ExFAT et bientôt WinFS. Sous Linux il en existe bien plus dont certains ont une utilisation spécifique. Si Linux peut lire les systèmes de fichiers Windows, l'inverse n'est pas vrai dans les installations standards.

Les fichiers sont donc organisés dans des systèmes de fichiers, sous forme d'arborescence en répertoires (aussi appelés dossiers dans les interfaces graphiques). Pour accéder à un fichier sous Windows, on part d'un « disque » ayant une lettre (par exemple C:\) et on descend les répertoires jusqu'au fichier. On peut voir, lorsqu'on affiche la barre d'adresse dans l'explorateur de fichiers, quel est le chemin parcouru pour accéder au fichier et que le séparateur entre les noms de répertoires est « \ ».

Sous Linux, cette notion de lettres associées à des disques n'existe pas. Tout part de la racine (l'équivalent de C:) nommée « / ». Celle-ci contient un certain nombre de répertoires et leur organisation est codifiée (comme sous Windows les répertoires « C:\Program files » et « C:\WINDOWS »). C'est à partir de là que l'on accède à tous les répertoires et fichiers du système.

Un système de fichier, sous tous les systèmes d'exploitation, est créé en *formatant* une partition.

### **3.2. Les disques**

Les disques sous Windows sont nommés par des lettres (C:, D:...) ces disques/lettres représentent des *partitions* d'un disque (une partition pouvant occuper un disque entier). Sous Linux les disques ne sont pas représentés de la même façon.

Comme indiqué plus haut, sous Linux tout est fichier. Un disque est donc accessible par un fichier. Un fichier spécial toutefois. Ce fichier est relié directement au disque par le noyau. Lire le contenu (ou y écrire) permet de lire le contenu brut du disque correspondant.

Les fichiers spéciaux se retrouvent, dans l'arborescence des fichiers, dans un répertoire particulier : `/dev` (pour *device* en anglais, ie « périphérique »). Par exemple seuls quatre disques durs (ou lecteur/graveur de CD/DVD) de type IDE peuvent être branchés sur un PC. Le bus IDE étant divisé en 2, il y a 2 disques durs par nappe IDE, un maître et un esclave. Sur le premier bus IDE, il y aura donc *hda* et *hdb* (hard drive a et hard drive b). De même, sur le second bus IDE, il y aura *hdc* et *hdd*. Tous quatre sont référencés dans le système de fichier par `/dev/hda`, `/dev/hdb`, `/dev/hdc` et enfin `/dev/hdd`.

Les disques SATA, SCSI et USB sont, quant à eux, tous référencés par `/dev/sda`, `/dev/sdb` et ainsi de suite...

### **3.3. Les partitions**

Une partition est un découpage d'un disque en plusieurs entités. Sous Windows ce sont les partitions qui sont nommées par les lettres habituelles et auxquelles nous accédons pour y lire dossiers et fichiers.

Sous Linux les partitions sont nommées différemment. En reprenant ce qui a été dit ci-dessus, un disque est représenté par un fichier spécial. De même, une partition est aussi représentée par un fichier spécial, se conformant au paradigme que tout est fichier. Le nom de ce fichier reprend le nom du disque sur laquelle est présente la partition en y adjoignant un numéro : le numéro d'ordre de la partition (la première partition aura le numéro 1, la seconde le numéro 2 et ainsi de suite). Par exemple, la première partition du disque *hda* (disque maître du premier bus IDE) sera nommée `/dev/hda1`. La sixième partition du disque SATA 2 sera nommée `/dev/sdb6`.

Jusque là nous n'avons pas parlé des fichiers sous Linux. C'est que tout simplement on n'accède pas aux fichiers sous Linux directement en accédant aux disques et aux partitions présent dans `/dev` tel que nous l'avons vu. Alors que sous Windows ces partitions sont accessibles dans un explorateur de fichiers via des lettres, sous Linux il n'en est pas de

même.

### **3.4. Relation entre disques et systèmes de fichiers**

Pour être plus précis, sous Windows, ce n'est pas directement aux partitions que l'on accède grâce aux lettres des disques. En réalité c'est au système de fichiers présent sur la partition que l'on accède.

C'est aussi ce que nous faisons sous Linux. Cependant cela est fait de manière différente. Sous Linux, cette notion de lettre de disque n'existe pas. Sous Linux il y a une partition principale dite « partition racine » car elle contient le « système de fichier racine » ou, en anglais *root-filesystem*. Ce système de fichier racine est nommé ainsi car c'est lui qui contient la base du système d'exploitation qu'est Linux et principalement les fichiers servant à contrôler et gérer le démarrage.

Les systèmes de fichiers des autres partitions sont ensuite « greffés » sur ce système de partition racine dans des répertoires choisis par l'administrateur système (l'utilisateur nommé *root* – et qui n'a rien à voir avec le *root filesystem* – aussi surnommé *superuser* ou *super-utilisateur* car il a tous les droits sur le système – cf chapitre 4). Cette opération de greffe s'appelle « monter » une partition. On monte une partition dans un répertoire appelé point de montage. La commande effectuant le montage d'une partition s'appelle logiquement `mount`. Elle permet aussi d'afficher la liste des systèmes de fichiers montés. Son pendant, `umount`, fait l'opération inverse : elle démonte une partition montée.

Dans un système pour le grand public, peu de partitions sont créées lors de l'installation de Linux. En général il y en a deux, voire trois. Les deux principales sont d'une part la partition racine. Comme son nom l'indique elle est à la racine de l'arborescence des fichiers. On la nomme « / » et c'est à partir de là que sont référencés tous les répertoires et fichiers du système (comme `/dev/hda1`, vu précédemment).

La seconde est une partition particulière. C'est une partition de mémoire virtuelle ou partition d'échange (*swap* en anglais). Elle joue le rôle de la mémoire virtuelle de Windows mais sur une partition dédiée afin d'en accélérer l'utilisation.

La troisième partition généralement utilisée dans les systèmes dits grand public est la partition utilisateur. C'est une partition où sont stockés les données des comptes utilisateurs. Elle est montée dans le répertoire `/home` et chaque utilisateur y a son répertoire personnel.

Un système professionnel compte en général bien plus de partitions (et donc de points de montage associés). Chaque répertoire est virtuellement le point de montage d'une partition même si ce n'est pas le cas la plupart du temps.

Les partitions contenant d'autres systèmes d'exploitation ainsi que les médias amovibles sont montés dans des répertoires dédiés. Nous y reviendrons plus tard.

L'arborescence des répertoires dans un système Linux est donc particulière. La partition racine est à la base de tout et les autres partitions y sont greffées, montées, de telle sorte

qu'accéder à un fichier d'une partition se fait en allant simplement dans le répertoire où cette partition est montée.

### **3.5. Organisation d'un système de fichier Linux**

Le système de fichiers Linux est hiérarchisé et organisé d'une façon particulière qui est commune aux systèmes Unix. Cette organisation est codifiée depuis les débuts des systèmes Unix. On trouve à la racine plusieurs répertoires dont voici une brève description des plus importants à connaître, dans l'ordre alphabétique :

`/bin`

Ce répertoire contient les programmes et commandes de base du système pour un utilisateur.

`/boot`

Ce répertoire contient les éléments nécessaires au démarrage de l'ordinateur (noyau et menu de démarrage pour le chargeur de démarrage).

`/dev`

Ce répertoire contient les fichiers spéciaux d'accès au matériel.

`/etc`

Ce répertoire contient la configuration du système ainsi que la liste des opérations à effectuer au démarrage et des services à lancer.

`/home`

Ce répertoire contient les répertoires personnels des utilisateurs.

`/lib`

Ce répertoire contient les bibliothèques partagées des applications principales.

`/media`

Ce répertoire contient les points de montage des partitions supplémentaires de l'ordinateur et des médias amovibles.

`/proc`

Ce répertoire est le point de montage d'un système de fichiers virtuel permettant d'accéder à certaines informations rapportées par le noyau. Étant un système de fichiers virtuel, il n'occupe aucune place sur le disque.

`/root`

Ce répertoire est le répertoire personnel du super-utilisateur.

`/sbin`

Ce répertoire contient des applications et commandes indispensables au démarrage.

`/sys`

Ce répertoire est le point de montage d'un système de fichiers virtuel permettant

d'accéder à la liste des périphériques reconnus par le noyau.

`/tmp`

Ce répertoire contient des fichiers et répertoires temporaires, il est accessible à tous les utilisateurs.

`/usr`

Ce répertoire contient les principales applications du système et leurs configurations, images, informations...

`/var`

Ce répertoire contient des informations sur les services en cours de fonctionnement, sur les logiciels installés, les journaux système etc.

Ces répertoires (tel `/usr`) contiennent une sous-arborescence de répertoires tout aussi hiérarchisée et codifiée sur laquelle nous ne nous étendrons pas.

## 4. Les utilisateurs

Linux est un système multi-utilisateurs depuis le début (en tant que système de type Unix), contrairement à Windows où c'est, comme pour le support réseau, une pièce rapportée. C'est un système multi-utilisateurs dans le sens où chaque utilisateur a son répertoire personnel et des droits sur l'accès aux fichiers et applications. C'est aussi un système multi-utilisateurs dans le sens où tous les utilisateurs peuvent utiliser la machine **en même temps** (notamment par des accès réseau).

Il y a un utilisateur particulier dans le système que nous avons évoqué précédemment : *root*. Cet utilisateur est le super-utilisateur, l'administrateur du système et, contrairement à l'administrateur système sous Windows qui n'est qu'un utilisateur comme les autres avec des droits particuliers, celui-ci a tous les droits... Y compris celui de détruire le système.

### 4.1. L'identifiant utilisateur

Un utilisateur est identifié auprès du système par son numéro d'utilisateur (*UID* – ou « User IDentifier » en anglais). Cet identifiant est unique sur le système. Il est associé avec un nom d'utilisateur, un mot, nom ou pseudo en minuscules et sans accent. Un mot de passe pour accéder à un compte utilisateur est indispensable (mais non obligatoire, son absence n'est toutefois pas recommandée pour des raisons de sécurité notamment lorsqu'il s'agit de l'utilisateur ayant des droits d'administration du système).

### 4.2. Les groupes

Sur le système les utilisateurs sont regroupés en groupes. Un utilisateur peut faire partie d'un certain nombre de groupes lui donnant des droits particuliers sur certaines parties du

système.

### 4.3. L'accès aux fichiers

Afin de pouvoir accéder à un fichier, un répertoire ou une application, l'utilisateur doit avoir les droits nécessaires à l'opération qu'il souhaite effectuer. Chaque fichier et répertoire se voit donc attribuer des droits envers les utilisateurs voulant utiliser le fichier ou le répertoire. Un fichier appartient à un et un seul utilisateur et fait partie d'un et un seul groupe.

Les droits sur les fichiers sont répartis en trois catégories : les droits pour le propriétaire du fichier, les droits pour les membres du groupe auquel appartient le fichier et enfin les droits pour tous les autres utilisateurs. Il est à noter que l'utilisateur *root* n'est soumis à aucune contrainte concernant les droits rendant l'administration plus aisée s'agissant de l'accès aux données, mais aussi plus risquée (en particulier lors de l'effacement).

Ces droits sont de mêmes types pour les trois catégories citées ci-dessus :

- le droit de lecture : l'utilisateur a le droit de lire le contenu du fichier ou la liste des fichiers du répertoire
- le droit d'écriture : l'utilisateur a le droit d'écrire des données dans le fichier ou de l'effacer, de créer ou effacer des fichiers dans le répertoire
- le droit d'exécution : l'utilisateur a le droit d'exécuter le fichier (si c'est une application ou un *script*), de changer le répertoire courant pour le répertoire concerné (ou de le traverser).

Lorsqu'on liste les fichiers d'un répertoire, ces droits sont notés `rwXrwxrwx`, avec un tiret à l'endroit où le droit n'est pas accordé. Le `r` est là pour signifier le droit en lecture (*read* en anglais), le `w` pour le droit en écriture (*write* en anglais) et le `x` pour le droit en exécution (*execute* en anglais). Le premier groupe `rwX` est associé aux droits pour l'utilisateur auquel appartient le fichier/répertoire, le second groupe est pour les membres du groupe auquel appartient le fichier/répertoire et enfin le troisième groupe ne s'applique qu'aux autres utilisateurs du système qui ne sont ni le propriétaire du fichier ni un membre du groupe du fichier.

Il existe d'autres types de droits plus complexes dont nous ne discuterons pas ici : leur description est disponible dans la page de manuel de la commande `chmod` (cf plus bas).

## 5. La ligne de commande

La ligne de commande est l'interface préférée des opérations de maintenance d'un système de type Unix. Il est en effet plus simple de lister, sur quelques lignes, les commandes à saisir plutôt que de montrer toutes les copies d'écran permettant d'accéder au résultat final. De plus cela permet d'automatiser nombre d'opérations longues et complexes en particulier sur les serveurs.

De plus ces commandes, souvent répétitives, peuvent être alignées dans un fichier exécutable appelé *script shell* (le *shell* étant le nom générique désignant l'interpréteur de commandes). Ceci a pour effet de limiter les risques d'erreur et d'éviter de fastidieuses saisies de lignes de commandes.

## 5.1. Les commandes de base

Les principales commandes à connaître de la ligne de commandes sont liées à l'accès aux fichiers.

Commande	Signification
<code>ls</code>	Liste les fichiers et sous-répertoires du répertoire courant
<code>cd <i>répertoire</i></code>	Change le répertoire courant pour celui spécifié
<code>mkdir <i>répertoire</i></code>	Crée un nouveau répertoire
<code>cp <i>fichier(s) dest</i></code>	Copie un ou plusieurs fichiers (ou répertoires) vers une destination
<code>mv <i>fichiers(s) dest</i></code>	Déplace/renomme un fichiers et répertoires vers un répertoire destination
<code>less <i>fichier</i></code>	Affiche le contenu d'un fichier
<code>chown <i>uid fichier</i></code>	Change le propriétaire du fichier / répertoire
<code>chgrp <i>groupe fichier</i></code>	Change le groupe du fichier / répertoire
<code>chmod <i>mode fichier</i></code>	Change les droits d'accès au fichier / répertoire
<code>pwd</code>	Affiche le répertoire courant depuis la racine
<code>find <i>options</i></code>	Recherche des fichiers suivant les options spécifiées
<code>rm <i>fichier</i></code>	Efface un fichier
<code>rmdir <i>répertoire</i></code>	Efface un répertoire vide
<code>man <i>commande</i></code>	Affiche la page de manuel de la commande

Les commandes (et les fichiers) sous Unix sont sensibles à la casse (majuscules / minuscules). Il faut donc veiller à respecter la façon dont elles sont écrites. Toutefois, la grande majorité des commandes sont entièrement en minuscules.

Ceci n'est qu'un exemple de quelques commandes utilisables. Il en existe de nombreuses autres, stockées dans les répertoires `/bin` et `/usr/bin`. Il existe cependant une commande à part, *indispensable* à tous : la commande `man`. En lui passant en paramètre le nom d'une autre commande, elle permet d'afficher le manuel d'utilisation de la commande spécifiée.

Toutes les commandes spécifiées dans la table ci-dessus ont une page de manuel qu'il est indispensable de consulter. En cas de présence d'une commande inconnue il est *toujours* recommandé de consulter sa page de manuel avant toute utilisation. Dans le cas où une commande n'aurait pas de page de manuel il y a une chance qu'elle soit une commande interne de l'interpréteur de commandes, auquel cas la commande `help commande`

permet d'avoir une aide sur celle-là.

## 5.2. Utilisation plus avancée

Le *shell* a aussi d'autres fonctionnalités : il permet de rediriger la sortie de commandes vers l'entrée d'autres commandes, de programmer des opérations, de lancer des applications en arrière plan et de les remettre au premier plan, etc.

La maîtrise de la ligne de commande n'est pas le but de ce document, cependant essayer de comprendre un système Unix sans comprendre la ligne de commande n'est pas chose aisée, d'autant que beaucoup de tâche d'administration se font encore en ligne de commande pour des raisons de sécurité, de concision et de rapidité. De plus toutes les commandes du *shell* n'ont pas d'équivalent graphique.

L'utilisation avancée de la ligne de commande nécessite temps et apprentissage pour avoir l'expérience et le recul nécessaire à la maîtrise de cet outil. Toutefois il y a quelques notions indispensables afin de pouvoir suivre et comprendre ce qui se passe lorsqu'on lit les forums sur Internet.

Les applications en ligne de commande peuvent communiquer entre elles par le biais de leurs entrées et sorties. Typiquement une application prend ses entrées via le clavier de l'utilisateur et affiche le résultat de son opération sur l'écran. On appelle le clavier « l'entrée standard » et l'écran « la sortie standard ». Plus exactement l'entrée standard d'une application est, par défaut, reliée au clavier, et sa sortie standard est reliée à l'écran.

Avec les systèmes Unix on peut brancher la sortie d'une commande sur l'entrée d'une seconde afin que celle-ci traite les résultat de la première. Voici un petit bréviaire des opérations possibles :

- Redirection en sortie  
au lieu d'utiliser l'écran comme sortie standard, on utilise un fichier :  
`$ commande > fichier`
- Redirection en entrée  
au lieu d'utiliser le clavier comme entrée standard, on utilise un fichier :  
`$ commande < fichier`
- Redirection de la sortie sur une entrée  
on branche la sortie d'une commande sur l'entrée d'une seconde :  
`$ commande1 | commande2`

Dans ces exemples le caractère \$ indique l'invite (*prompt* en anglais) de la ligne de commande. Bien sûr il est possible d'enchaîner autant de commande de cette façon qu'on le veut. Il y a beaucoup de commande sous Linux qui permettent de manipuler des chaînes de caractères telles que celles renvoyées par la plupart des commandes.

Le séparateur entre les éléments d'une commande est l'espace et le séparateur entre plusieurs commandes distinctes est le point-virgule. En ce qui concerne l'arborescence, le caractère / sépare les noms des répertoires à l'instar de \ sous DOS/Windows.

## 6. L'interface graphique

L'interface graphique sous Linux est quelque chose d'assez particulier à comprendre. Elle est composée de plusieurs couches à tel point que certains la comparent à un mille-feuille.

Pour commencer il y a le matériel (la carte vidéo et l'écran) géré par un pilote de périphérique (*driver* en anglais) inclus ou non dans le noyau. Puis il y a le système graphique X11 qui gère l'accès au matériel (en passant par le driver) et permet d'y afficher des lignes, des points etc. Il supporte aussi un protocole réseau lui permettant d'exporter l'affichage sur une autre machine de façon totalement transparente pour les applications.

Au-dessus il y a le gestionnaire de fenêtres. Ce programme particulier permet de dessiner des bordures autour des fenêtres, de les déplacer, redimensionner, de les fermer... Il est souvent associé de nos jours à un gestionnaire de bureaux permettant d'afficher des chose sur le fond de l'écran (icônes, applications spéciales, etc). Les gestionnaires de fenêtres et de bureau permettent depuis très longtemps de pouvoir gérer plusieurs bureaux virtuels offrant ainsi un moyen de classer les applications ouvertes dans plusieurs bureaux. Il existe de nombreux gestionnaires de fichiers et de bureau pour les systèmes Unix, chacun offrant à l'utilisateur des fonctionnalités particulières, des façon de se configurer différentes, des « look & feel » différents permettant à l'utilisateur de faire ressembler son système à de vieux systèmes Unix, des ordinateurs Apple, NeXT, Windows, ou d'avoir d'autres interfaces innovantes et originales. Les principaux utilisés sont *KDE*, *Gnome*, *XFCE*, *WindowMaker* et *Enlightenment*, mais il en existe de nombreux autres.

Enfin on trouve les applications. Celles-ci utilisent un *toolkit* graphique pour afficher les éléments qui les composent (l'ensemble des composants graphiques que sont les menus, les boutons, les zones de saisie, etc). Il existe plusieurs *toolkits* graphiques. Les deux principaux utilisés sont *GTK+* et *QT*. Les applications sont souvent conçues pour un environnement de bureau spécifique (principalement Gnome ou KDE) et utilisent des fonctionnalités particulières à cet environnement de bureau. C'est pourquoi il est recommandé d'installer des applications qui ont été conçues pour l'environnement de bureau qu'on utilise. Il est toutefois, bien entendu, possible d'installer n'importe quelle application avec pour seul risque la possibilité de non cohérence de l'apparence et l'oubli de quelques traductions.

## 7. L'installation de nouveaux logiciels

Sous Windows, l'installation d'applications passe par deux principales sources : les CD d'installation ou la recherche d'application sur Internet. De plus, les mises-à-jour des applications sont laissées à la libre volonté des vendeurs tant par la fréquence que par les moyens logiciels et structurels (les serveurs). Il y a bien une section du « panneau de configuration » intitulée « installer/désinstaller des logiciels » mais ça ne marche pas bien et personne ne l'utilise vraiment.

Sous Linux ça marche et il faut l'utiliser : l'installation et la mise-à-jour de logiciels est centralisée. Certes les sources d'applications sont toujours les CD ou Internet. Cependant pour installer ces applications il faut passer par une application particulière qu'on nomme *gestionnaire de paquets*.

En effet les logiciels à installer se présentent (comme sous les autres systèmes d'exploitation) sous forme d'une archive à installer.

Sous Windows ces archives sont auto-extractibles et installables, laissant la possibilité à des gens mal-intentionnés de générer de fausses archives d'installation pour un logiciel, installant à la place un virus, un cheval de Troie, un logiciel espion ou tout autre *malware* (publicité, barre d'outils, etc).

Sous Linux un seul utilisateur a le droit d'installer des logiciels : *root*. Toutefois, dans les distributions récentes, le système permet de déléguer des droits *root* à certains utilisateurs afin qu'ils puissent effectuer des tâches d'administration sans avoir à devenir root et risquer de faire des erreurs potentiellement dangereuses pour le système.

Les logiciels viennent donc sous forme d'archives nommées *paquets*. Toutes les applications sont empaquetées par les développeurs de la distribution, rassemblés au même endroit sur des serveurs Internet et installables avec un outil unique (spécifique à la distribution) mais pouvant avoir plusieurs interfaces : graphique, ligne de commande...

Le *gestionnaire de paquets* permet de :

- chercher des paquets à installer
- installer des paquets
- désinstaller des paquets
- effectuer des mises-à-jour (automatiques ou non) de logiciels
- effectuer une mise-à-jour vers une nouvelle version du système d'exploitation

Les informations concernant les paquets logiciels sont récupérées initialement sur le média d'installation mais sont ensuite récupérées périodiquement sur Internet. L'installation d'applications par ce moyen assure de la compatibilité de l'application avec le reste du système. **L'utilisation des logiciels de gestion de paquets permet de tenir un système à jour vis-à-vis des failles de sécurité.**

Il reste cependant possible d'installer des applications récupérées sur Internet. Tout d'abord sous forme binaire (application/archive) directement installable sur le système – et cela peut être le cas pour certaines applications propriétaires grand public (mais ayant presque toujours un équivalent libre sous Linux).

Ensuite il y a les sources. En effet, une application, avant d'être utilisable est composée de code source créé par des programmeurs et utilisé pour générer l'application utilisable. Il est toujours possible de récupérer les sources d'une application libre et de les compiler afin de générer l'application exécutable. Cependant cette opération nécessite un certain nombre de connaissances en informatique.

Les paquets fournissent donc plus que de simples applications : certains contiennent de la documentation, de la « localisation », des fichiers de développement, des bibliothèques...

## 8. Pour aller plus loin

Dans ce chapitre nous allons décrire plus en détail certains points d'un système d'exploitation de type Unix.

### 8.1. Le démarrage de l'ordinateur

L'étude de la façon dont démarre votre système vous permettra de mieux appréhender certains points lorsque vous serez confrontés à des problèmes de maintenance.

Le démarrage de l'ordinateur commence par le chargement d'un programme particulier appelé *bootloader* (« chargeur de démarrage » en français). Actuellement le plus utilisé est *grub*. Celui-ci charge ensuite le noyau Linux avec certains paramètres comme, par exemple, le nom de la partition racine, puis lui passe la main.

Linux démarre et commence à configurer le matériel. Lorsqu'il a fini sa tâche, il charge le programme de démarrage et de gestion des services : `/sbin/init`. Le programme `init` charge alors sa configuration résidant dans `/etc` et démarre tous les services et programmes configurés pour être lancés au démarrage.

C'est là qu'est lancée l'invite d'identification (*login* en anglais) en mode texte et en mode graphique.

### 8.2. Les fichiers devices

Les fichiers *device* se trouvent dans `/dev`. Ces fichiers sont principalement de deux types : les fichiers de type caractère et les fichiers de type bloc. Les premiers sont généralement des fichiers d'accès à des éléments périphériques communicants caractère par caractère (un clavier, une souris). Les seconds sont des fichiers d'accès à des périphériques utilisant des blocs (disques, partitions).

Ces fichiers sont créés au fur et à mesure de la détection du périphérique correspondant par le système. Ils sont un moyen pour une application de communiquer avec le matériel auquel ils correspondent.

Le paradigme selon quoi tout est fichier ne s'applique cependant pas à deux catégories d'éléments sous Linux : les périphériques réseau et la mémoire partagée. Nous n'allons parler que des périphériques réseau ici. Il est possible de les lister avec la commande `ifconfig`. Les périphériques réseau sans fil *wifi* ont des fonctionnalités supplémentaires listées avec la commande `iwconfig`.

### 8.3. Les fichiers de `/proc` et `/sys`

Les fichiers présents dans les répertoires `/proc` et `/sys` ne sont pas réels. Ils ne sont pas réels dans le sens où ce sont des fichiers créés en mémoire par le noyau et n'existent pas sur le disque et ne prennent, par conséquent, pas de place réelle. En particulier le fichier `/proc/kcore` qui représente la mémoire physique (la RAM) de l'ordinateur.

Ces fichiers servent à récupérer des informations sur l'état du système, des processus, des périphériques reconnus par le noyau, etc.

#### **8.4. Les médias amovibles**

Nous l'avons vu précédemment, les disques et partitions contenant un système de fichiers ont besoin d'être montés dans l'arborescence du système de fichiers principal de Linux. Il en est de même pour les CD, DVD, clés USB et autres médias amovibles.

Il y a plusieurs années, ce montage devait être fait manuellement. De nos jours, l'insertion d'un média amovible est automatiquement détectée et son système de fichiers est monté dans `/media/<nom-du-périphérique>`.

Une icône représentant le média est alors affichée sur le bureau pour les gestionnaires de bureau les plus modernes. Il faudra toutefois penser à « démonter » ou « éjecter » le périphérique avant de le retirer.

Cette détection automatique ne se limite d'ailleurs pas aux médias amovibles mais fonctionne aussi pour les imprimantes USB par exemple.

#### **8.5. La documentation des applications**

Les applications installées sous Linux ont souvent de la documentation associée en plus de la page de manuel. On trouve cette documentation dans le répertoire `/usr/share/doc/<nom-de-l'application>`.

#### **8.6. L'interpréteur de commandes**

Comme dit précédemment la maîtrise de la ligne de commande est vivement recommandé afin de pouvoir comprendre le fonctionnement d'un système Unix. Sous Linux, comme sous tous les systèmes Unix, il existe un certain nombre d'interpréteurs de commande que l'on peut installer et utiliser à sa préférence. Habituellement l'interpréteur de commande par défaut est *bash*.

Afin de connaître et de maîtriser son fonctionnement, la lecture de sa page de manuel est fortement recommandée même si elle peut paraître rébarbative au premier abord.

## 8.7. D'autres types de fichiers

Il existe d'autres types de fichiers sous Linux que les fichiers spéciaux et ceux dits « normaux ». Tout d'abord il y a les liens symboliques et les liens durs qui sont un peu le pendant unixien des « raccourcis » sous Windows. Un lien symbolique est un raccourci vers un fichier ou un répertoire ailleurs dans le système (mais prenant *beaucoup moins* de place que son équivalent Windows).

Ensuite il y a les *FIFO* (First In – First Out, soit « premier entré – premier sorti » en français) ou, plus précisément, les tubes nommés : l'équivalent du « | » entre commandes vu plus haut, mais sous forme de fichier.

Enfin il y a les *sockets*. Ces fichiers un peu bizarres sont des moyens de communications, à la façon des communications réseau, mais passant par des fichiers.

## 8.8. Les logiciels propriétaires

Sous Linux la plupart des logiciels sont libres. Il existe tout de même des logiciels propriétaires, c'est-à-dire des logiciels dont le vendeur ne souscrit pas aux règles de liberté énoncées en début de ce document. De tels logiciels sont fournis avec simplement le programme exécutable et non les sources (et les droits qu'apporte le logiciel libre avec). Il n'est pas recommandé d'installer ce type de logiciel mais cela peut parfois s'avérer nécessaire.

Notamment lorsqu'on se trouve en présence de matériel dont les spécifications techniques n'ont pas été fournies par le constructeur et que celui-ci ne fournit que des pilotes de périphérique propriétaires. C'est le cas en ce qui concerne les cartes vidéo Nvidia ainsi que quelques fabricants de cartes wifi.

De même certains types de logiciels, des *codecs* (pour « codeurs-décodeurs ») ne sont pas libres (en particulier pas libre de droits). Ils permettent de décoder des fichiers afin de pouvoir les lire. C'est principalement le cas des fichiers audio et vidéos. Ces codecs n'étant pas libres, ils ne sont donc pas installés par défaut dans un système Linux. Il revient à l'utilisateur de les installer manuellement mais des outils logiciels facilitent leur installation en détectant les tentatives d'accès à des fichiers de ce type et en trouvant le codec adéquat à installer.

## 9. Quelques références

1. Liberté  
<http://www.gnu.org/philosophy/philosophy.fr.html>
2. Mode de développement  
<http://www.linux-france.org/article/these/cathedrale-bazar/>
3. Initiation à Unix, commandes  
<ftp://ftp.linux-france.org/pub/article/debutant/ini-unix.pdf>

4. Environnement graphique Gnome  
<http://www.gnomefr.org/>
5. Environnement graphique KDE  
<http://fr.kde.org/>
6. Fiches pour débutants (pas forcément à jour)  
<http://lea-linux.org/cached/index/Accueil.html>
7. Le site web de la distribution utilisée.

Merci à tous ceux qui m'ont aidé à rédiger et corriger ce document.

Dernière mise-à-jour le 18 juillet 2014 à 9h50.