



# Utilisation du système d'exploitation Linux et réseau

Master Pro 1ere année Modélisation Statistique, Module POO

Richard FERRERE  
Laboratoire de Mathématiques CNRS  
Université de Franche-Comté

*octobre 2015*

# Table des matières

<b>I</b>	<b>INTRODUCTION</b>	<b>3</b>
<b>1</b>	<b>Des précisions et quelques définitions</b>	<b>4</b>
1.1	Déroulement, objectifs . . . . .	4
1.2	Qu'est-ce qu'un ordinateur ? . . . . .	5
1.3	Qu'est-ce qu'un système d'exploitation ? . . . . .	5
1.4	Qu'est-ce qu'un réseau informatique ? qu'est-ce qu'un protocole ? . . . . .	6
1.5	Qu'est-ce que la virtualisation ? . . . . .	7
1.6	Qu'est-ce qu'une distribution GNU/Linux ? . . . . .	7
<b>2</b>	<b>Un peu d'histoire...</b>	<b>9</b>
2.1	UNIX et son histoire . . . . .	9
2.2	D'où vient LINUX et sa mascotte ? . . . . .	10
<b>3</b>	<b>Une liste non exhaustive de raisons pour installer un système Linux...</b>	<b>11</b>
<b>II</b>	<b>PRISE EN MAIN D'UNE MACHINE LINUX, en plus de Windows</b>	<b>12</b>
<b>4</b>	<b>Quelques notions clés !</b>	<b>13</b>
4.1	le BIOS (Basic Input/Output System)- de quoi s'occupe le BIOS?...et l'UEFI . . . . .	13
4.1.1	BIOS en version classique . . . . .	13
4.1.2	Version alternative du Bios, l'interface logicielle UEFI . . . . .	13
4.2	Disque dur, partition et montage . . . . .	14
4.3	En quoi consiste le MBR ou le 'Master Boot Record' ou la zone d'amorce ? . . . . .	15
4.4	En quoi consiste le chargeur d'amorçage ou bootloader?... . . . . .	16
4.5	En quoi consiste le noyau linux et son lancement?... . . . . .	16
<b>5</b>	<b>Le système d'exploitation Linux Ubuntu</b>	<b>19</b>
5.1	Classification de Ubuntu, d'où vient cette distribution ? . . . . .	19
5.2	Son installation, un jeu d'enfant ! . . . . .	20
<b>6</b>	<b>Découverte de l'environnement graphique GNOME</b>	<b>22</b>
6.1	Présentation du serveur X et et du gestionnaire de fenêtres et de bureau . . . . .	22
6.2	Fonctionnalités des bureaux graphiques Gnome : bureau à adopter ? . . . . .	23
6.3	Logithèque riche par de nombres applications et logiciels intégrés à GNOME . . . . .	23
6.3.1	Bureautique et multimédia . . . . .	23
6.3.2	Graphisme . . . . .	24
6.3.3	Internet . . . . .	24
<b>7</b>	<b>Gestion des paquets : mises à jours et installations de programmes</b>	<b>25</b>
7.1	Gestion des paquets et des mises à jours . . . . .	25
7.2	Installation de programmes hors paquets . . . . .	26
7.3	Est-il possible de jouer avec sa distribution Linux ? . . . . .	27

<b>III EXPLORATION DU SYSTEME LINUX</b>	<b>28</b>
<b>8 Connexion et session</b>	<b>29</b>
8.1 Type de connexion : session et fin de session de travail	29
8.1.1 Gestion des utilisateurs	29
8.1.2 Gestion des mots de passe	30
8.2 le mode console : le shell bash	30
8.3 la ligne de commande et le prompt	32
8.4 Type et format de commande	32
8.4.1 Majuscules/Minuscules	33
8.4.2 Caractères spéciaux	33
8.4.3 Répertoire d'accueil et variable 'HOME'	33
8.4.4 Redirections et tubes	33
8.4.5 Commandes de déplacement	34
8.4.6 Expressions régulières	34
8.5 Démons et processus	35
<b>9 Organisation du système de fichiers</b>	<b>36</b>
9.1 Arborescence du système de fichiers	36
9.2 Chemin d'accès relatif et absolu	37
9.3 Organisation des répertoires et fichiers	37
9.4 Gestion des protections sur les fichiers	38
9.5 Création et destruction de répertoires et de fichiers	39
9.6 Création et destruction de liens	39
9.7 Gestion d'autres périphériques	40
9.8 Documentation en ligne...	40
<b>10 Outils de base et de programmation</b>	<b>41</b>
10.1 Outils de base	41
10.1.1 Archivage et sauvegarde	41
10.1.2 Découpage de fichiers	42
10.1.3 Comparaison et analyse de fichiers	42
10.1.4 Mécanisme de tube et de pipe	42
10.1.5 Mécanisme de recherche et de tri	42
10.2 Editeurs de texte et outils de programmation	43
10.2.1 Des éditeurs simples comme Kate, Kwrite et Gedit	43
10.2.2 Quelques éditeurs évolués comme Nano et Vi	43
10.2.3 Un IDE comme Kdevelop	45
<b>11 Communication réseau</b>	<b>46</b>
11.1 Principe général du fonctionnement réseau	46
11.1.1 Ethernet et couches réseau	46
11.1.2 TCP/IP	48
11.1.3 Connectivité, interfaces et routage	49
11.2 Principe de connexion, de communication réseau et utilitaires	50
11.2.1 connexion sécurisée	50
11.2.2 transfert sécurisé ou non de fichiers	51
11.2.3 connaître son environnement réseau	51
<b>12 Quelques références bibliographiques et internet...</b>	<b>52</b>

Première partie

**INTRODUCTION**

# Chapitre 1

## Des précisions et quelques définitions

### 1.1 Déroulement, objectifs

#### Déroulement du parcours :

- 7 séances partagées entre cours et TD/TP
- Beaucoup d'exemples d'illustration en séances de TP
- Des outils fondamentaux pour la suite de travaux pratiques, le Master 2, la recherche et dans le monde professionnel.
- Contrôle continu sur ordinateur avec une série de questions. Un examen final avec la partie de M. LANGROGNET.
- Note finale avec le reste du module POO. La partie utilisation du système linux et réseau compte pour environ 35%).

#### Décomposition des 7 séances :

<b>semaine</b>	<b>date</b>	<b>horaire</b>	<b>avancement</b>
<b>41</b>	<i>mar</i> 06/10	9h30 – 12h30	<i>cours TD</i>
<b>41</b>	<i>ven</i> 09/10	13h30 – 16h30	<i>cours TD</i>
<b>43</b>	<i>lun</i> 19/10	13h30 – 16h30	<i>cours TD</i>
<b>45</b>	<i>mer</i> 04/11	13h30 – 16h30	<i>cours TD</i>
<b>46</b>	<i>lun</i> 09/11	13h30 – 16h30	<i>cours TD</i>
<b>47</b>	<i>mar</i> 17/11	13h30 – 16h30	<i>cours TD</i>
<b>48</b>	<i>mar</i> 24/11	13h30 – 16h30	<i>cours TD</i>

#### Objectifs multiples et fondamentaux :

- Comprendre ce qu'est un ordinateur et son système d'exploitation linux et interface graphique ; et il s'agit d'avoir une vue d'ensemble des différents composants.
- Apprendre à utiliser l'environnement de travail dans un shell de commandes ; apprendre à utiliser des éditeurs comme les programmes 'gedit', 'nano' et éventuellement 'vi'.
- Acquérir les principales commandes de linux afin de pouvoir réaliser des tâches courantes d'administration et de gestion.
- Comment sont stockées les données sur le disque dur et comment les manipuler ?
- Comment installer un programme et lancer un programme sous linux ?
- Connaître et gérer son système de fichiers.
- Découvrir la programmation de scripts shell avec des instructions.
- Découvrir des utilitaires tels que : sed,awk, grep, expressions régulières.
- Découvrir le fonctionnement réseau d'une machine linux, la gestion et la surveillance de votre système.
- Des outils fondamentaux pour travailler en efficacité.

## 1.2 Qu'est-ce qu'un ordinateur ?

*Un ordinateur est une machine dotée d'une unité de traitement lui permettant d'exécuter des programmes enregistrés. C'est un ensemble de circuits électroniques permettant de manipuler des données sous forme binaire, ou bits. Cette machine permet de traiter automatiquement les données, ou informations, selon des séquences d'instructions prédéfinies appelées aussi programmes.*

[lien internet sur ordinateur](http://fr.wikipedia.org/wiki/Ordinateur) <http://fr.wikipedia.org/wiki/Ordinateur>

*Un ordinateur est une machine capable de résoudre des problèmes en appliquant des instructions préalablement définies. La suite des instructions décrivant la façon dont l'ordinateur doit effectuer un certain travail est appelé programme. . Extrait du livre de Andrew TANENBAUM "Architecture de l'ordinateur".*

A chaque ordinateur est souvent associé une notion d'architecture processeur qui n'est plus de type Mac ou PC, mais plutôt du type 32 ou 64 bits ; un programme comme un système d'exploitation (ou encore son code source) est compilé pour ladite architecture de processeur ; le code source est donc transformé en une succession d'instructions machines (codes binaires) qui sont exécutées par le processeur type par le biais d'un compilateur (gcc par exemple pour le langage C).

Le système d'exploitation va être destiné à une architecture 'x86' (32bits) ou 'x64' (64 bits) et plus précisément, la largeur de bus qui est sur 64 bits au lieu de 32 devrait permettre une transmission, sur la carte mère, de deux fois plus d'informations entre les différents composants.

[lien internet sur l'architecture des ordinateurs](http://www-int.impmc.upmc.fr/impmc/Enseignement/ye/informatique/systemes/chap1/111.html) <http://www-int.impmc.upmc.fr/impmc/Enseignement/ye/informatique/systemes/chap1/111.html>

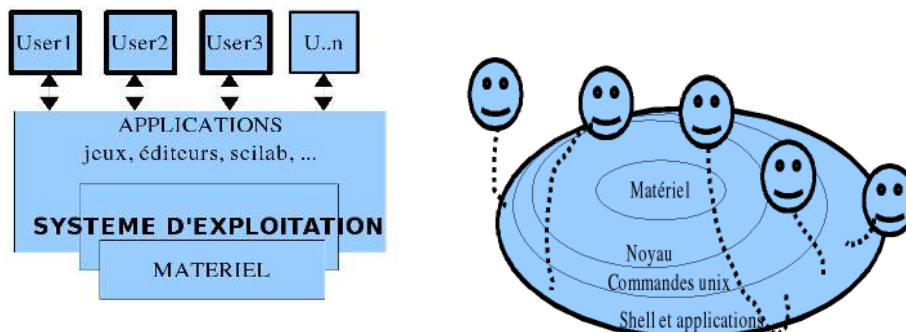
## 1.3 Qu'est-ce qu'un système d'exploitation ?

*Un système d'exploitation (SE, en anglais Operating System ou OS) est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications informatiques de l'utilisateur (traitement de texte, jeux vidéo, etc. . . ). Il fournit aux programmes d'applications des points d'entrée génériques pour les périphériques.*

[lien internet sur Système d'exploitation](http://fr.wikipedia.org/wiki/Système_d'exploitation) [http://fr.wikipedia.org/wiki/Système d'exploitation](http://fr.wikipedia.org/wiki/Système_d'exploitation)

Un système d'xploitation est composé de deux parties : le noyau, un programme de base qui pilote le matériel et qui sera chargé en mémoire dès le démarrage de l'ordinateur ; et un ensemble de programmes qui communiquent avec le noyau et permettant de réaliser des actions précises (déplacer un fichier, supprimer un fichier, créer un répertoire...). Le noyau est un peu comme les fondations d'une maison ; il doit être solide et stable car le bon fonctionnement des applications (ou programmes) en dépend directement.

Les couches fonctionnelles et l'architecture du système linux peuvent se schématiser :



## 1.4 Qu'est-ce qu'un réseau informatique ? qu'est-ce qu'un protocole ?

Un réseau informatique est un ensemble d'équipements reliés entre eux pour échanger des informations. Par analogie avec un filet (un réseau est un « petit rets », c'est-à-dire un petit filet), on appelle nœud (node) l'extrémité d'une connexion, qui peut être une intersection de plusieurs connexions (un ordinateur, un routeur, un concentrateur, un commutateur).

[lien internet sur réseau informatique](http://fr.wikipedia.org/wiki/Réseau_informatique) [http://fr.wikipedia.org/wiki/Réseau\\_informatique](http://fr.wikipedia.org/wiki/Réseau_informatique)

L'informatique était autrefois centralisée avec de grosses machines qui travaillaient pour plusieurs utilisateurs. Et les utilisateurs n'avaient qu'à leur disposition que de simples "terminaux bêtes". Ces grosses machines étaient reliées entre elles par un des premiers réseaux en France, appelé ' RENATER ', lequel réseau reliant les universités et les centres de recherche entre eux.

[lien internet sur RENATER](http://www.renater.fr) <http://www.renater.fr>

Puis arrive une ère de l'ordinateur personnel ' Personnel Computer ou PC ', de leurs systèmes d'exploitation orientés réseau et surtout de l'explosion de l'Internet ; les ordinateurs sont des entités qui communiquent entre eux et avec le reste du monde ; il est maintenant extrêmement facile de communiquer des informations à son entourage, de changer de poste de travail sans pour autant devoir transporter ses fichiers.

Cette communication réseau existe également grâce au succès de la technologie ' ETHERNET ' et où tous les ordinateurs sont égaux vis-à-vis du réseau.

Cela n'est possible que s'il existe le même protocole de communication. Un protocole est un mode opératoire qui doit être commun à tous les éléments qui désirent communiquer entre eux ; et donc il n'y a pas de communication possible sans avoir recours à un protocole.

## 1.5 Qu'est-ce que la virtualisation ?

Il ne faut pas confondre avec l'émulation qui par le fait d'utiliser un logiciel imite la façon de réagir d'un autre ordinateur, ce qui limite sérieusement les performances et n'exploite pas directement les ressources de l'ordinateur hôte. Donc, cela exige peu de puissance processeur et mémoire !

*En informatique, on tente de définir la virtualisation comme un ensemble de techniques matérielles et/ou logicielles qui permettent de faire fonctionner sur une seule machine plusieurs systèmes d'exploitation et/ou plusieurs applications, séparément les uns des autres, comme s'ils fonctionnaient sur des machines physiques distinctes. Les outils de virtualisation servent à faire fonctionner ce qu'on appelle communément des serveurs privés virtuels (Virtual Private Servers ou VPS) ou encore environnements virtuels (Virtual Environments ou VE).*

[lien internet sur Virtualisation](http://fr.wikipedia.org/wiki/Virtualisation) <http://fr.wikipedia.org/wiki/Virtualisation>

Habituellement c'est la méthode du 'dual-boot' qui est utilisée pour faire cohabiter un système Linux avec un autre système (tel que Windows). Depuis quelques années, c'est la virtualisation qui permet de faire fonctionner simultanément deux systèmes d'exploitation quelqu'ils soient, un système 'invité' au-dessus d'un système dit 'hôte'. Plusieurs logiciels de virtualisation existent sur le marché : des libres comme QEMU, certaines versions VMWare, VirtualPC, Hyper-V de Microsoft et Virtualbox de Sun. C'est ce dernier logiciel 'Virtualbox' [lien internet sur Virtualbox](#) qui a été choisi et fera croire au système d'exploitation invité Linux (sur l'ordinateur virtuel) qu'il se trouve sur un ordinateur physique, en l'occurrence Windows. Pourquoi VirtualBox ? parce qu'il est multi plates-formes, libre et gratuit ; mais il propose un grand choix de systèmes d'exploitation à virtualiser, et il supporte une taille de mémoire vidéo pouvant aller jusqu'à 32 Mo. Etant donné que deux systèmes d'exploitation vont fonctionner en même temps, la virtualisation reste très gourmande en ressources mémoire et processeur. La virtualisation se fait à partir d'un logiciel qui va créer un ordinateur (virtuel) utilisant un ordinateur matériel relativement récent (à partir de 2003). Mais le système d'exploitation de l'ordinateur virtuel est installé "normalement". On peut, avec ce principe installer Windows dans Linux, ou Linux dans Windows. C'est donc une technologie qui permet d'exécuter un système d'exploitation à partir d'un autre même si deux systèmes apparaissent côte à côte : le système "virtualisé" est démarré comme une application sur le système réel, comme n'importe quelle autre application. Pour virtualiser, il est possible d'utiliser des programmes comme 'Virtualbox' ou 'KVM', en sachant que ce dernier est intégré aux distributions linux et basé sur une version modifiée de 'Qemu'.

## 1.6 Qu'est-ce qu'une distribution GNU/Linux ?

Une distribution est un ensemble cohérent de logiciels comprenant : un noyau, des outils système liés au noyau et tous les autres logiciels indispensables pour le fonctionnement de l'ordinateur. Elle se présente sous la forme d'un support d'installation sur CD-Rom ou DVD qui comprend tout une batterie de logiciels et outils ; souvent la notion de distribution est associée à celle de noyau linux ou de noyau... ; c'est donc un concept ou une philosophie qui peut s'accompagner d'une histoire.

Le site internet de 'distrowatch.com' donne les distributions de logiciels souvent libres. Une distribution a un rythme d'évolution qui peut montrer soit une certaine stabilité, soit une très de très grandes innovations ; et se pose donc la question du choix d'une distribution ! il en existe plus de 200 distributions différentes ; le choix dans ce cours porte sur une distribution K(ubuntu) qui est une distribution généraliste car elle est basée interface graphique KDE qui permet de tout faire par sa simplicité.



Linux est souvent associé aux logiciels libres ou sous souvent sous licence GNU GPL ; qui dit «libre» dit liberté : d'utiliser le programme, de l'adapter, de l'améliorer et de le redistribuer. Le noyau linux est un système d'exploitation (avec Linus Torvalds en 1991) sous licence GPL et on parle souvent de GNU/Linux. Après d'autres logiciels se sont greffés au noyau Linux pour devenir des projets libres afin de fournir tout à chacun de quoi satisfaire au mieux la plupart des besoins. GNU désigne le premier projet de développement d'un système d'exploitation libre et porté par Richard STALLMAN en 1983 ; son objectif était de créer à nouveau un système complet tout comme UNIX, mais libre par le biais d'une fondation FSF (Fondation pour le logiciel libre). GNU est donc un acronyme qui signifie 'Gnu is Not Unix'. Ce travail va dans le sens d'une protection des productions libres sous la forme d'une licence GPL (General Public Licence), et donc autorise l'utilisateur à copier et à distribuer à volonté tout en protégeant tous les travaux.

Linux désigne seulement au sens strict le noyau ou un bout de code... ; mais c'est sous la notion de GNU/Linux qu'on peut trouver un Linux et donc un noyau, des outils et bien d'autres morceaux de logiciels.

[lien internet sur la notion de Distribution GNU/Linux](http://fr.wikipedia.org/wiki/Distribution_Linux) [http://fr.wikipedia.org/wiki/Distribution Linux](http://fr.wikipedia.org/wiki/Distribution_Linux)

[lien internet sur la découverte de quelques distribution GNU/Linux](http://www.linuxfacile.net/decouvrir/presentation-des-principales-distributions.html)  
<http://www.linuxfacile.net/decouvrir/presentation-des-principales-distributions.html>

[lien internet sur les comparatifs et actualités des distribution GNU/Linux](http://distrowatch.com/) <http://distrowatch.com/>

Le concept de logiciel libre est venu avec le projet GNU Linux qui a pour principe un total libre accès aux codes sources et une liberté d'utiliser, de modifier et d'améliorer, et de redistribuer des copies. Et les progrès en matière de développement et de technologies ne peuvent qu'être améliorées de par la participation et le suivi d'un plus grand nombre de personnes, appelé souvent la communauté. Il faut garder à l'esprit que tout logiciel libre sous licence GPL (General Public Licence) n'est pas forcément 'gratuit' au sens du terme, mais plutôt apporte une certaine liberté dans un cadre de protection. Et la notion 'd'OpenSource, s'appliquant aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, est plus là maintenant pour garantir une certaine forme de qualité par rapport au code source. Linux devient un système 'UNIX' et donc un logiciel libre sous licence GPL avec codes ouverts ou libres, mais comment ?

# Chapitre 2

## Un peu d'histoire...

### 2.1 UNIX et son histoire

C'est en 1969 que Kenneth THOMPSON, employé chez Bell, développe un nouveau prototype de système à temps partagé ; son nom est Unics et sera Unix.

Entre 1970 et 1975, Unix devient le système portable et officiel pour les institutions et les universités ;

En 1980, des chercheurs de l'université de Berkeley développent leur propre UNIX (BSD) ;

En 1983, la société ATT tente une commercialisation d'un UNIX système V et de concurrencer l'UNIX BSD ;

En 1984, Richard Stallman lance le projet GNU/Linux qui vise à créer son système 'Unix' complètement libre ;

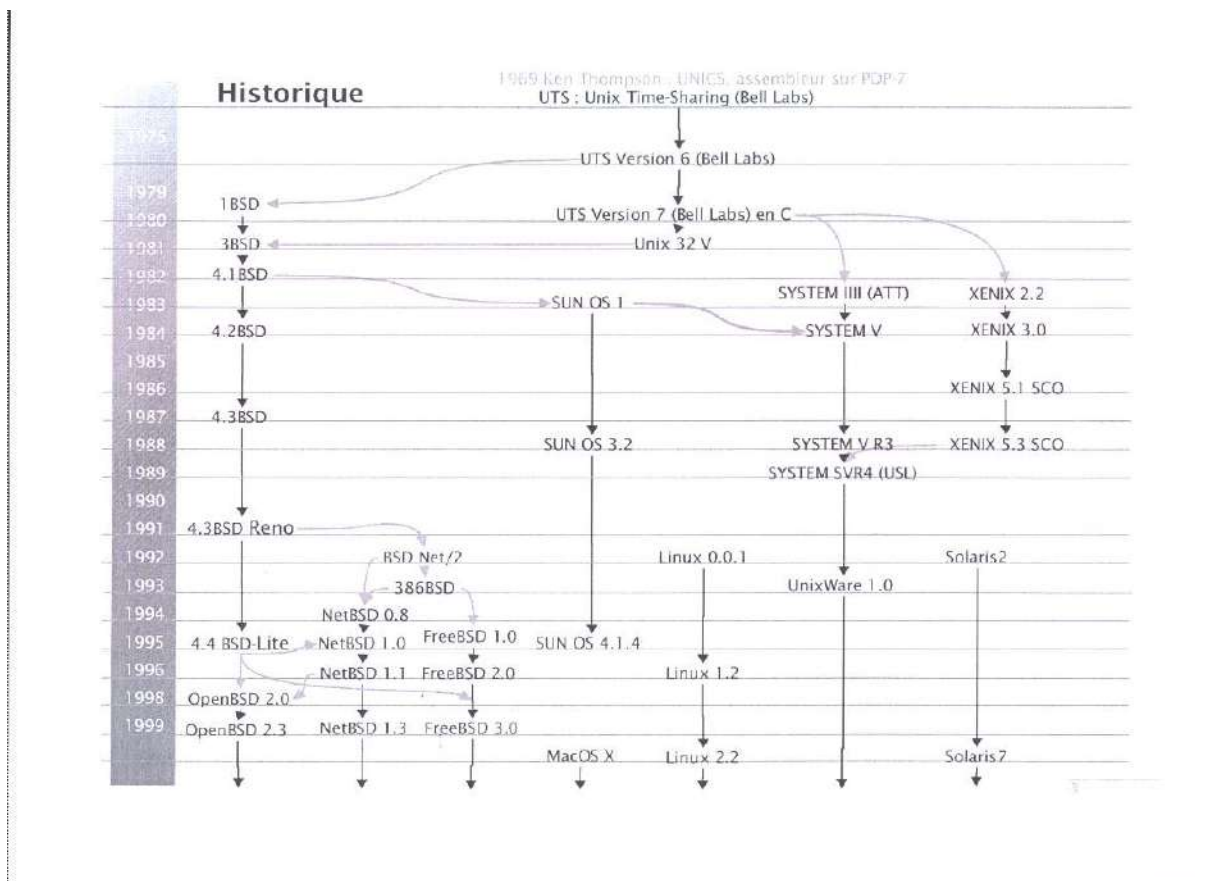
En 1987, la société ATT finalise sa version de Unix...et donc des scissions se créent pour arriver au choix d'une licence BSD !

En 1988, c'est le début d'une normalisation avec l'organisation 'IEEE' et la norme 'POSIX' pour les développements d'applications autour d'un système UNIX.

En 1991, un étudiant finlandais, Linus TORVALDS, crée un noyau UNIX qui a été ajouté aux travaux du projet GNU de Stallman, a donc donné naissance à GNU/Linux.

En 2004, un sud-africain, Mark SHUTTLEWORTH, qui est un développeur de la distribution Linux DEBIAN, lance son projet UBUNTU ; l'objectif étant de rendre populaire GNU/Linux en fournissant une interface facile à utiliser.

Un schéma historique avec les ramifications... de Jean-Luc Charles de Lamefip Bordeaux - Administration Unix' et... [lien sur l'histoire des UNIX](http://www.unix.org/) [http ://www.unix.org/](http://www.unix.org/)



Un site internet sur l'histoire des ordinateurs.

[lien sur l'histoire des ordinateurs](http://www.computerhistory.org/) [http ://www.computerhistory.org/](http://www.computerhistory.org/)...un site sur l'histoire du système d'exploitation Unix.


[lien sur Unix](http://fr.wikipedia.org/wiki/UNIX) [http ://fr.wikipedia.org/wiki/UNIX](http://fr.wikipedia.org/wiki/UNIX)

un historique plus complet à l'adresse suivante :

[lien sur l'histoire de Unix](http://www.levenez.com/unix/) [http ://www.levenez.com/unix/](http://www.levenez.com/unix/)

## 2.2 D'où vient LINUX et sa mascotte ?

L'histoire du système Linux ou GNU/Linux remonte alors bien longtemps. Il a impliqué de nombreux acteurs et le projet GNU est mature capable de rivaliser avec les systèmes UNIX. Mais c'est en 1991 qu'un étudiant finlandais, Linus TORVALDS, diffuse le message comme quoi il conçoit un système d'exploitation gratuit 'c'est juste un hobby'. Tout est parti de là car plusieurs volontaires se sont intéressés et ont fait remonté à Linus Torvalds des correctifs. Et c'est en 1992 que naît le noyau Linux couplé au système GNU donnant le premier système d'exploitation libre.

Linux a une mascotte (manchot pygmée) depuis 1996, date à laquelle de nombreux développeurs se sont accaparés le noyau linux et ont décidé que leur projet ait un logo. C'est Linus Torvald, l'initiateur du projet, qui décida que cela devait prendre l'allure chétive d'un manchot pygmée face au gros projet Unix. Et c'est Larry Ewing qui donna l'image originale du mythique 'Tux' pour quelque chose comme 'Torvalds UniX...'.  


[lien sur la mascotte Linux](http://fr.wikipedia.org/wiki/Tux) [http ://fr.wikipedia.org/wiki/Tux](http://fr.wikipedia.org/wiki/Tux)

## Chapitre 3

# Une liste non exhaustive de raisons pour installer un système Linux...

1. Installer et mettre à jour légalement et sans avoir à payer, installer la même copie de mon OS (Ubuntu) sur plusieurs machines sans me soucier des restrictions de licences ou de clés d'activation ; Distribuer des copies de mon système d'exploitation et des programmes qui tournent dessus sans violer aucune loi ;
2. Fonctionner sans utiliser d'anti-virus, de protection anti-adware ou spyware, ne pas avoir à redémarrer ma machine pendant des mois tout en recevant toujours les derniers correctifs de sécurité ;
3. Installer et exécuter facilement différentes interfaces graphiques si je n'aime pas la configuration par défaut ;
4. Installer une centaine de programmes par une simple commande ; les applications qui sont très modulaires et basées sur des bibliothèques communes en font un système extrêmement léger ; il existe des mini-distributions (Small Linux, Deli Linux,...) linux capables de s'installer sur des machines vieillissantes ;
5. Fonctionner sans avoir à 'défragmenter' mon disque dur, car Linux écrit là où il y a de la place et fait une gestion optimisée de l'espace ;
6. Avoir le contrôle total du matériel installé sur ma machine et savoir qu'il n'y a pas de porte dérobée dans mes logiciels ;
7. Pouvoir faire une énorme erreur qui nécessite la réinstallation complète de mon système et être capable de le faire en moins d'une heure, parce que j'ai mis toutes mes données sur une partition séparée du système d'exploitation et des programmes ;
8. Pouvoir démarrer mon système avec de "SUPERS" effets, aussi sympa que ceux de Vista, sur une machine qui a 3 ans... en moins de 40 secondes, temps d'identification compris (nom d'utilisateur + mot de passe),
9. Être capable de configurer tout ce que je veux, légalement, y compris mes programmes fétiches. Je peux même contacter les développeurs du logiciel concerné pour leur poser des questions, leur donner des idées et être impliqué dans la construction ou le développement de la version en cours si j'en ai envie ;
10. Pouvoir utiliser le même matériel pendant plus de 5 ans avant qu'il n'ait réellement besoin d'être remplacé... J'ai toujours du matériel qui a presque 10 ans, qui tourne sous Linux et qui est toujours utile ;
11. Pouvoir surfer sur internet pendant que l'OS s'installe!!!
12. Utiliser à peu près n'importe quel matériel en sachant que le pilote est déjà présent dans le système d'exploitation... éliminant ainsi la nécessité de rechercher le site du fabricant pour trouver ce pilote ;
13. Obtenir le code source de pratiquement n'importe quoi, y compris celle du noyau du système d'exploitation ou celle de la plupart de mes applications.
14. Disposer réellement d'une plus grande autonomie avec sa batterie d'ordinateur portable...
15. Lancer un plus grand nombre d'applications sans pour autant monopoliser toute la mémoire disponible ; une gestion de la mémoire virtuelle optimale qui évite un trop grand nombre de 'swaps' ;
16. Compiler soi-même son noyau et/ou ses applications afin de tirer le meilleur parti de son matériel ;

Deuxième partie

**PRISE EN MAIN D'UNE MACHINE  
LINUX, en plus de Windows**

# Chapitre 4

## Quelques notions clés !

Avant de pouvoir lancer le noyau linux et l'ensemble de ses services, il est nécessaire d'aborder quelques notions clés qui permettront de comprendre comment installer et faire démarrer le système d'exploitation Linux.

### 4.1 le BIOS (Basic Input/Output System)- de quoi s'occupe le BIOS ?...et l'UEFI

#### 4.1.1 BIOS en version classique

C'est un programme situé en mémoire morte au niveau de la carte mère qui s'exécute dès que l'ordinateur est mis sous tension avec une séquence POST 'Power On Self Test'. Il initialise le matériel présent sur l'ordinateur en effectuant des vérifications sur celui-ci :

1. contrôle le bus système,
2. vérifie tous les connecteurs d'extension,
3. vérifie la mémoire de la carte graphique et les signaux commandant l'affichage,
4. interroge le BIOS de la carte vidéo et ajoute son code de reconnaissance et c'est à ce moment-là que l'affichage apparaît à l'écran,
5. teste la RAM en tentant une écriture sur chaque zone mémoire et tente de lire ensuite pour les comparer à ce qu'il a écrit,
6. vérifie si le clavier et la souris sont bien connectés,
7. envoie des signaux à tous les périphériques de stockage (lecteur disquette, cd, et DVD, USB ...) pour définir quels sont les différents lecteurs en sachant que tous les résultats sont comparés sur le CMOS, ce qui permet au BIOS de savoir si la configuration matérielle a changé depuis le dernier démarrage ou pas,
8. amorce en mémoire le secteur d'amorce principal du disque dur, appelé MBR.

Par cet utilitaire, il est possible de configurer différents composants matériels et l'ordre de démarrage des périphériques par exemple...est donc accessible très souvent au démarrage de l'ordinateur (en appuyant sur une touche du clavier comme la touche SUPPRESSION ou F2, F10...). Le BIOS va surtout vérifier que la présence du premier secteur du disque est 'bootable' et que cela puisse faire démarrer le système d'exploitation.

#### 4.1.2 Version alternative du Bios, l'interface logicielle UEFI

Une interface logicielle située entre le micrologiciel et le système d'exploitation Windows, apparue sur les ordinateurs depuis 2010 avec de nouvelles fonctionnalités : adressage de la mémoire, l'amorçage par le réseau, la cryptographie des disques durs et leur exploitation avec de nouvelles tables de partitions labellisées GPT pour les gros volumes. C'est le boot EFI, ou plutôt la norme UEFI (Unified Extensible Firmware Interface) adoptée en 2007 par plusieurs grands constructeurs (HP, ASUS, APPLE, INTEL). Il représente un logiciel intermédiaire entre le micrologiciel (firmware) et le système d'exploitation (OS) l'ordinateur.

- > Un "boot manager" UEFI charge l'OS loader et les drivers indispensables au démarrage.
- > Les formats de tables de partition sont de type MBR, mais également GPT (GUID Partition Table).
- > La norme n'est plus basée sur un secteur de boot,
- > Le GPT peut permettre 128 partitions principales sur un support disque de capacité de grande taille (zettaoctet, milliard de téraoctets).
- > Le GPT peut permettre de démarrer sur des disques de plus de 2 Téraoctets (To).
- > Un boot sécurisé car cela utilise un mécanisme de vérification par signatures numériques -> attention aux installations d'autres systèmes d'exploitation !

ordinateur avec Bios UEFI <http://doc.ubuntu-fr.org/uefi>

## 4.2 Disque dur, partition et montage

Un disque dur (ou HDD, Hard Disk Drive) est un empilement de plateaux métalliques et rigides en rotation sur un axe. C'est une mémoire de stockage de grande capacité qui peut être lue ou écrite. Les couches des plateaux sont recouvertes d'une couche magnétique sur laquelle les données sont stockées. Des têtes de lecture/écriture écrivent les données en binaire (0 et 1) selon un flux électrique ; ces données sont organisées en cercles concentriques appelés « pistes » ; les pistes sont séparées en quartiers que l'on appelle secteurs, contenant les données (au minimum 512 octets par secteur).

Les dernières technologies de disque dur sont dites 'SDD' (Solid State Disk) basées sur de la mémoire flash à la place des plateaux car encore plus fiables.

Une partition est une partie du disque dur sur laquelle va s'installer un système d'exploitation ou un système de fichiers. Un disque dur peut contenir une ou plusieurs partitions qui peuvent être primaires (ou principales), logiques et/ou étendues. Plus précisément, le disque dur peut contenir jusqu'à quatre partitions principales ou trois partitions principales et une partition étendue. Le fait de partitionner le disque dur consiste à aménager des zones distinctes et continues de cylindres. L'intérêt d'avoir plusieurs partitions est triple : il s'agit de minimiser la fragmentation des données, de mieux protéger les données et d'avoir plusieurs systèmes d'exploitation. Les informations sur les partitions sont conservées dans une zone appelée table des partitions ; on peut y trouver les adresses de début des partitions et leur taille.

Tous les systèmes UNIX/Linux ont besoin au minimum de deux partitions pour fonctionner : la partition principale représentée par le slash ( / ) qui contient la racine du système et la partition d'échange dédiée dit '/swap' ; cette dernière permet au noyau linux de libérer de la place en y stockant des informations dessus.

Tous les systèmes Unix possèdent un point de montage, 'un répertoire' par lequel il est possible d'accéder à des données se trouvant sur une partition du disque dur ou alors sur un périphérique comme le lecteur de DVD ou la clé USB. L'accès au système de fichiers n'est possible que s'il est monté et alors les fichiers et répertoires font partie intégrante de ce répertoire monté. Sous Linux, tout est fichier ou encore pour lire ou écrire des données, il faut procéder au montage (sans oublier de démonter après!). Les distributions Linux ont comme point de montage par défaut le répertoire '/mnt' ou '/media' ; par exemple, pour une distribution Ubuntu c'est '/media/cdrom'. Tous les points de montage (partitions et périphériques) sont consignés dans un fichier important sous '/etc/fstab' ; il contient toutes les partitions du disque, incluant le système de fichier, qui vont être montées automatiquement au démarrage du système ; celui-ci permet également de conserver toutes les options sur les montages des partitions et des périphériques ; chaque ligne de ce fichier représente une partition ou un périphérique à monter automatiquement et comportant un numéro d'identification unique (UUID : Universal Unique Identifier).

Exemple du contenu de /etc/fstab :

```
/ was on /dev/sda2 during installation
```

```

UUID=024225e4-39ac-4c57-8934-d71c5b0fa6a0 / ext4 relatime,errors=remount-ro 0 1
/boot was on /dev/sda1 during installation
UUID=aa8d19ba-73b7-469c-9cb2-b14858d5e9e2 /boot ext4 relatime 0 2
/home was on /dev/sda5 during installation
UUID=3518001b-40ef-4043-96bb-ff16bf3374e2 /home ext4 relatime 0 2
swap was on /dev/sda3 during installation
UUID=3f737b90-caed-4d32-8c88-cff97d19f115 none swap sw 0 0
/dev/scd0 /media/cdrom0 udf,iso9660 user,noauto,exec,utf8 0 0

```

Sous Windows, cette notion n'existe pas de cette façon car toute partition est considérée comme un élément complètement indépendant avec sa propre arborescence de fichiers et représenté avec des lettres C :, D :,....

### 4.3 En quoi consiste le MBR ou le 'Master Boot Record' ou la zone d'amorce ?

Mais la table de partition principale est stockée dans le premier secteur du disque ou premier secteur de démarrage. Il contient donc le programme de démarrage ou d'amorçage 'bootloader' en déterminant à partir de quelle partition le système d'exploitation doit démarrer.

1. Le MBR permet de trouver (ou d'identifier) la partition active du disque,
2. Le MBR va charger le secteur de boot correspondant et transférer ensuite l'exécution à ce dernier.

[lien sur le MBR](http://fr.wikipedia.org/wiki/Master_boot_record) [http://fr.wikipedia.org/wiki/Master boot record](http://fr.wikipedia.org/wiki/Master_boot_record)

Elle contient :

- 440 octets de code de démarrage...
- 4 octets contenant les information du volume...
- 2 octets non utilisés! ...
- 64 octets contenant les tables de partition...et
- 2 octets qui sert à vérifier l'exactitude de la MBR

**Rappel** : la quantité d'information s'exprime dans une unité qui est le 'bit' ou information binaire (0 ou 1). Mais le plus souvent, l'unité utilisée c'est l'octet qui est composé de 8 bits et permettant de coder 256 caractères différents, soit  $2^8$ .

-  $10^3$  : 1 kilo octet ou 1000 octets, ce qui correspond environ à une page d'information contenue dans un livre;

-  $10^6$  : 1 Méga octet ou 1000 kilooctets, ce qui correspond environ à un livre de 1000 pages;

-  $10^9$  : 1 Gigaoctet ou 1000 mégaoctets, ce qui correspond environ à une bibliothèque de 1000 livres ou volumes;

-  $10^{12}$  : 1 Téraoctet ou 1000 gigaoctets, ce qui correspond environ à une collection de millions de livres;

-  $10^{15}$  : 1 Pétaoctet ou 1000 téraoctets ,ce qui correspond environ à de grands ensembles (cultures humaines,...).

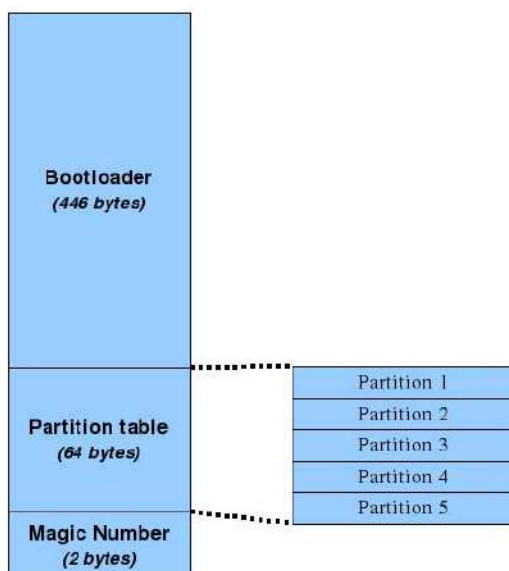
Plus précisément, c'est un bloc de 512 octets (ou bytes) localisé physiquement sur le premier secteur du disque (au secteur 1 du cylindre 0 et tête 0 (ou secteur 0 en logique); un système Linux peut s'installer



également sur une partition primaire logique. La MBR contient donc la routine d'amorçage ou le 'bootloader' présente sur la partition active et le nombre magique sur les 2 derniers octets (0xAA55) sert à vérifier si la MBR est correcte.

La MBR peut être éditée et même sauvegardée dans un fichier. Si votre disque dur est représenté comme périphérique 'sda', pour éditer la MBR, la commande 'sudo dd if=/dev/sda count=1 | hexdump -C'; pour sauvegarder la MBR sur une clé USB, la commande 'sudo dd if=/dev/sda of=/media/sdb1/Mambr446.img bs=446 count=1'.

Est-ce que votre disque dur est tatoué? ce qui est contraire au code de la consommation! la commande 'sudo dd if=/dev/sda count=63 | hexdump -C'. Le tatouage d'un ordinateur est un dispositif technique imposé dans des ordinateurs de certaines marques (DELL, MEDION...) et développé en accord entre ces constructeurs et Microsoft; il interdit à la version de Microsoft Windows® livrée avec votre ordinateur d'être installée sur un autre ordinateur.! Souvent cela se fait via le disque dur mais cela peut se trouver également au niveau du BIOS ou d'autres composants de l'ordinateur.



#### 4.4 En quoi consiste le chargeur d'amorçage ou bootloader ?...

Le chargeur d'amorçage est un gestionnaire de démarrage pour les systèmes d'exploitation (Windows, Linux, Mac...). C'est un programme de bas niveau chargé de démarrer le noyau Linux juste après la phase du BIOS; pour cela, il doit être capable de retrouver sur le disque le noyau Linux à démarrer. Pour Windows, ce programme est appelé NTLDR, mais après installation de GNU/Linux, c'est le programme GRUB (GRand Unified Boot loader) qui a la main. Ce petit bout de logiciel qui se trouve sur la MBR est représenté par une interface graphique au démarrage de l'ordinateur permettant de choisir un système d'exploitation.

#### 4.5 En quoi consiste le noyau linux et son lancement ?...

Dans le cas de Linux, la présence à la racine d'un lien de 'vmlinuz' vers /boot/vmlinuz-2.6.32...ou -3.2.0 (où 2.x.x, 3.x.x sont les versions du noyau), image compressée du noyau dont le premier rôle, après création des tables et structures nécessaires à son fonctionnement et montage de la partition contenant la racine du

système de fichier va lancer le programme `'/sbin/init'`, père de tous les processus ou le premier processus système qui lancera tous les autres processus.

Pour cela, une image d'un système minimal compressé est initialisé au démarrage du système et chargé en mémoire; c'est le système `'initrd.img'` (INITial RamDisk) qui contient des modules du noyau nécessaires au boot comme les modules de système de fichiers (`ext3`, `etc`) et de matériel (contrôleur ide, sata). A ce stade, le noyau est chargé en mémoire et exécuté. Après quelques opérations d'initialisation sur le matériel (pilotes) et lui-même, le noyau monte la partition correspondant à la racine du système de fichiers.

Puis il exécute un programme appelé `'init'` (ou `/sbin/init`) qui est chargé de démarrer complètement le système avec toutes les tâches se trouvant dans le répertoire `'/etc/rcX.d'`; le `'X'` correspond au niveau d'exécution de chaque tâche système (niveau 2 ou 5). Chacune de ces tâches est `'un script'` qui se charge d'effectuer une opération spécifique, comme par exemple de monter toutes les partitions et tous les périphériques déclarés dans le fichier `'/etc/fstab'`, ou alors de démarrer des services comme l'interface graphique (KDE ou GNOME).

A l'origine, le programme `'init'` lit le fichier `'/etc/inittab'` afin de connaître les services ou processus à lancer lors de chaque démarrage de l'ordinateur et donc du système d'exploitation; c'est ainsi que ce fichier `'inittab'` comporte les entrées nécessaires et qui vont faire appel aux différents scripts. Le fichier `'/etc/inittab'` définit le niveau de fonctionnement (runlevel) par défaut 2, 3 ou 5 en général. Le niveau 0 déclenche l'arrêt de la machine, 6 son redémarrage, 1 le passage en mode mono-utilisateur appelé `'single user'` pour des tâches de maintenance (réparation du système) et S ou s qui sont en fait le même niveau et ne sont utilisés que dans la phase d'initialisation au démarrage. Les niveaux 2 à 5 sont des niveaux multi-utilisateurs qui en fonction des distributions permettent d'activer ou non un certain nombre de fonctionnalités. Par exemple, la distribution GNU/Linux Ubuntu utilisent le niveau 3 qui active les fonctionnalités réseau (serveurs) et le niveau 5 active le serveur graphique (Xwindows).

Le format du fichier `'/etc/inittab'` est, id runlevels action commande

id = identificateur

runlevels = tous les niveaux pour lesquels l'action spécifiée doit être exécutée

action = les actions sont...

initdefault= défini le niveau de fonctionnement par défaut, aucune commande n'est lancée

sysinit = commande lancée au moment du démarrage du système

wait = la commande est lancée (souvent un script) et `init` attend qu'elle se termine

respawn = la commande est relancée quand elle se termine (exemple la connexion depuis une terminal)

processus = la commande qui sera exécutée

Actuellement, les distributions Linux ont tendance à utiliser d'autres schémas plus optimaux de démarrage de leur système. C'est ainsi que la distribution UBUNTU utilise plutôt un script de type `'Upstart'` qui est un `'init'` sophistiqué; les services sont démarrés non plus de façon séquentielle et d'après une liste de scripts, mais plutôt en fonction de la réception d'évènements.

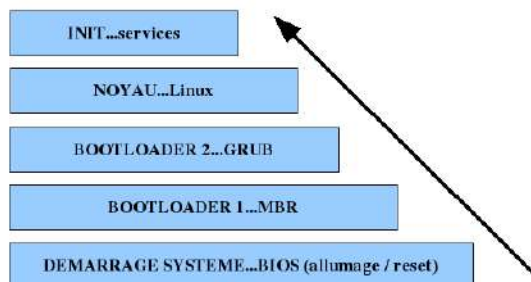
L'outil `'upstart'` gère des scripts de niveau de démarrage et de façon plus optimale car la méthode de démarrage devient plus dynamique basée sur des tâches et des évènements; c'est un ensemble de fichiers de configuration situés dans `/etc/init`; et donc l'évènement `'startup'` est généré par le processus `init`, qui entraîne d'autres évènements. le niveau par défaut de démarrage pour Ubuntu est le niveau 2 (dans le fichier `'/etc/event.d/rc-default'`) et la commande `'initctl list'` montre la liste des tâches en fonctionnement ou en attente; souvent il existe des alias des commandes `'initctl restart, initctl reload, initctl status...'`. Au final, le programme `'upstart'` (à la différence de `'inittab'`) agit sur une tâche bien précise dans un niveau d'exécution sans pour autant changer tout le niveau! En conséquence, le temps de démarrage est amélioré et les services non utilisés ne sont pas démarrés.

[lien sur Upstart](http://upstart.ubuntu.com/) <http://upstart.ubuntu.com/>

Le processus 'init' et ses scripts shell de service, inclus dans tous les systèmes Unix et Linux dans l'arborescence au niveau '/etc/init.d/scriptXXX' permettent des actions d'arrêt et de démarrage avec les paramètres 'stop'/'start'; et leur lancement se fait par '/etc/rcX.d/'. Si 'upstart' reste l'alternative à ce processus 'init', une nouvelle architecture de processus existe de plus en plus au sein des distributions Linux (Fedora, Mandriva, OpenSuse, RedHat et bientôt Debian), c'est 'systemd'. Cette nouvelle technologie va gérer et contrôler les processus (PID) pères et fils et les logs en jouant le rôle de coordonnateur et d'arbitre afin d'améliorer les performances de démarrage des systèmes Linux. Les services deviennent dépendants et inter-dépendants car 'systemd' sera capable par anticipation de lancer en parallèle ces services et donc des différents processus. Et un socket Unix sera créé pour chacun des services à démarrer.

Systemd se caractérise par des objets de type unités comportant un état (active, inactive, failed) prédéfinis et équivalents aux valeurs des champs des lignes du fichier '/etc/inittab'. La commande centrale pour gérer toutes les actions est 'systemctl' en remplacement des commandes 'chkconfig', 'service', ...

Un schéma simplifié du processus de démarrage du système Linux :



## Chapitre 5

# Le système d'exploitation Linux Ubuntu

### 5.1 Classification de Ubuntu, d'où vient cette distribution ?

Ubuntu est un ancien mot africain qui signifie « Humanité » et qu'on prononce 'Ou-boun-tou', est un système d'exploitation libre fondé sur Debian. Ubuntu signifie également « Je suis ce que je suis grâce à ce que nous sommes tous ». C'est un état d'esprit qui fait que c'est une distribution très populaire, assez conviviale et la plus suivie par une grande communauté ; Ubuntu découlant du projet (2004) de vulgarisation du projet GNU/Linux ; elle fonctionne avec une interface de type 'Gnome' ou 'KDE' ; K(Ubuntu), c'est sur une distribution Ubuntu basée sur un environnement ou interface KDE avec un certain nombre de dépôts supplémentaires à installer en plus de la distribution classique.

Par ailleurs, le logo de Ubuntu représente trois personnes se tiennent la main...



Mark Shuttleworth (homme d'affaires sud-africain) a créé la société 'Canonical LTD.' et par la même occasion la première version de Ubuntu 4.10.

Le principe de sortie de Ubuntu :

Ubuntu sort une nouvelle version à peu près tous les 6 mois car elle est basée sur l'environnement GNOME (qui sort lui également une version tous les 6 mois). Et entretemps, une version appelée L.T.S. (Long Time Support) sort tous les 2 ans dont les mises à jour de sécurité sont possibles et assurées pendant 5 ans pour les versions avec environnement de bureau. En général, deux versions de Ubuntu apparaissent à peu près chaque année en avril et en octobre sous un nom de code d'un animal accompagné d'un qualificatif commençant par la même lettre ! (HARDY HERON ou JAUNTY JACKALOPE), d'où les numéros de versions correspondants :

4.10 = version de l'année 2004.mois d'octobre et nom de code 'WARTY WARTHOG' (pour phacochère verruqueux)...

5.10 = version de l'année 2005.mois d'octobre et nom de code 'BREEZY BADGER' (pour blaireau jovial)

6.06 LTS = version de l'année 2006.mois de juin et nom de code 'DAPPER DRAKE' (pour canard pimpant)...

7.10 = version de l'année 2007.mois d'octobre et nom de code 'GUSTY GIBBON' (pour gibbon courageux)

8.04 LTS = version de l'année 2008.mois d'avril et nom de code 'HARDY HERON' (pour héron robuste)

8.10 = version de l'année 2008.mois d'octobre et nom de code 'INTREPID IBEX' (pour bouquetin intrépide)

9.04 = version de l'année 2009.mois d'avril et nom de code 'JAUNTY JACKALOPE' (pour jackalope enjoué)

9.10 = version de l'année 2009.mois d'octobre sous le nom de code 'KARMIC KOALA' (pour koala karmic)

10.04 LTS = version de l'année 2010.mois d'avril sous le nom de code 'LUCID LYNX' (pour lynx lucide)

10.10 = version de l'année 2010.mois d'octobre sous le nom de code 'MAVERICK MEERKAT' (pour maverick meerkat)

11.04 = version de l'année 2011.mois d'avril sous le nom de code 'NATTY NARWHAL' (pour narval chic)

11.10 = version de l'année 2011.mois d'octobre (le 13 octobre 2011) sous le nom de code 'ONEIRIC OCELOT' (pour ocelot onirique)

12.04 LTS = version LTS de l'année 2012.mois d'avril sous le nom de code 'PRECISE PANGOLIN' (pour pangolin précis) et soutenue et maintenue 5 années (au lieu de 3) jusqu'en avril 2017

12.10 = version de l'année 2012.mois d'octobre (le 18 octobre 2012) sous le nom de code 'QUANTAL QUETZAL' (pour quetzal quantique)

13.04 = version de l'année 2013.mois d'avril (le 25 avril 2013) sous le nom de code 'RARING RINGTAIL' (pour bassaris impatient)

13.10 = version de l'année 2013.mois d'octobre (le 17 octobre 2013) sous le nom de code 'SAUCY SALAMANDER' (pour salamandre impertinente)

14.04 LTS = **de l'année 2014.mois d'avril sous le nom de code 'TRUSTY TAHR'** (bélier confiant, ou rappelle le tahr de l'himalaya cousin du bouc et du chamois)

14.10 = **de l'année 2014.mois d'octobre sous le nom de 'UTOPIIC UNICORN'** (pour licorne utopique)

15.04 = **de l'année 2015.mois d'avril sous le nom de 'VIVID VERVET'**(le Singe Vervet Vif)

15.10 = **de l'année 2015.mois d'octobre sous le nom de 'WILY WEREWOLF'** (Le Loup-Garou Rusé)

[lien vers les versions de Ubuntu](http://fr.wikipedia.org/wiki/Liste_des_versions_d'Ubuntu) [http://fr.wikipedia.org/wiki/Liste\\_des\\_versions\\_d'Ubuntu](http://fr.wikipedia.org/wiki/Liste_des_versions_d'Ubuntu)

## 5.2 Son installation, un jeu d'enfant !

Les distributions de Ubuntu 14.04 LTS 'TRUSTY TAHR' et la 15.04 'VIVID VERVET' peuvent être installées selon trois méthodes :

La première méthode consiste à installer Ubuntu sur une partie physique du disque dur appelée partition, mais cela comporte des risques : perte de données suite à une erreur d'installation en dualboot et réinstallation du système Windows en plus !

La deuxième méthode consiste à installer Ubuntu à partir du système Windows déjà installé ; pour cela, il faut utiliser un programme d'installation spécialement conçu pour Windows et téléchargeable à l'adresse suivante : [windows-installer](#) . Cette méthode n'est pas optimale et comporte des défauts : composants matériels pas toujours reconnus et nécessite beaucoup de mémoire vive.

La troisième méthode consiste également à installer Ubuntu à partir du système Windows, mais en utilisant le programme VIRTUALBOX qui permet de créer une machine virtuelle Ubuntu à part entière ; cette méthode a l'avantage de garder le système windows intact en simulant un ordinateur sous Ubuntu. Cette méthode sera installée à l'occasion d'un TP à partir d'une image '.iso' téléchargée sur internet via le programme de virtualisation (Virtualbox déjà installé) ; son installation ne change en rien du support DVD ou CD-Rom.

Ces deux dernières versions (16eme et 17eme) de Ubuntu ont un processus d'installation graphique complètement repensé et facilité afin de permettre son installation en quelques étapes.

[lien sur l'installation de ubuntu](http://doc.ubuntu-fr.org/installation) <http://doc.ubuntu-fr.org/installation>

[lien documentation Ubuntu...si vous êtes perdu](http://doc.ubuntu-fr.org/) <http://doc.ubuntu-fr.org/>

[cohabitation avec windows7 ou 8](http://doc.ubuntu-fr.org/cohabitation_ubuntu_windowssi_windows7_ou_8_est_installe_sur_1_ordinateur) [http://doc.ubuntu-fr.org/cohabitation\\_ubuntu\\_windowssi\\_windows7\\_ou\\_8\\_est\\_installe\\_sur\\_1\\_ordinateur](http://doc.ubuntu-fr.org/cohabitation_ubuntu_windowssi_windows7_ou_8_est_installe_sur_1_ordinateur)

Ces nouvelles versions de Linux Ubuntu apportent de plus en plus :

- un démarrage toujours plus rapide (en moins de 20 secondes...);
- une meilleure compatibilité des matériels;
- une véritable identité visuelle avec des nouveaux thèmes graphiques apportant légèreté et transparence;
- un service 'Ubuntu One' avec des outils intégrés pour le stockage des données et la musique en ligne;
- une interface 'UNITY' tentant de remplacer l'interface GNOME!;
- un noyau Linux 3.XX [lien sur le noyau linux](#) et des pilotes de carte graphique et d'autres périphériques mieux gérés;
- un système de fichiers amélioré en EXT4 mais également du BTRFS...;
- mais il est toujours possible d'avoir les deux environnements graphiques toujours présents : GNOME classic, KDE.

En plus des deux variantes officielles (Ubuntu basée sur Gnome et Kubuntu basé sur KDE) soutenues par Canonical et la communauté Ubuntu (Ubuntu en mode Desktop et en mode Server), il est possible de trouver d'autres variantes reconnues de Ubuntu : Edubuntu, Xubuntu, Lubuntu... Ces variantes sont destinées à un usage particulier (domestique, adaptée à un serveur, pour le multimédia...) ou uniquement pour inclure une sélection différente de logiciels et d'environnement graphique par défaut (Xubuntu ou Lubuntu).

## Chapitre 6

# Découverte de l'environnement graphique GNOME

Une fois l'ordinateur démarré, le service GDM affiche le login graphique. Et alors après la saisie de login et de mot de passe, s'affiche un vaste environnement de bureau avec la barre UNITY.

### 6.1 Présentation du serveur X et du gestionnaire de fenêtres et de bureau

GNOME (GNU Network Object Model Environment) est un gestionnaire de bureau. KDE est également un gestionnaire de bureau dont le projet lancé en 1996 par l'allemand Matthias ETRICH ne cesse d'évoluer ; sorti en première version en 1998, il est basé sur la librairie graphique 'QT' et est passé sous licence GPL en 2000. GNOME quant à lui, est basé sur une librairie 'GTK'. Le projet GNOME Shell, centrée autour de la notion d'activités et d'environnement de bureau par défaut de Ubuntu, n'est plus l'unique interface. L'interface UNITY est venue prendre le relais sauf sur la version de Ubuntu 10.04.

Ces services de gestionnaire graphique constituent un ensemble cohérent de couches et donc une application aux multiples facettes basée sur le serveur X ou 'X.org'. Il est couplé également à des gestionnaires de fenêtres (appelé 'window manager'...) en proposant également des thèmes de bureau et des menus de lancement d'applications.

La gestion graphique sous linux est basée sur plusieurs couches : la couche système qui fournissant les outils de base, le gestionnaire d'affichage X Window ou Display Manager (GDM ou KDM) se chargeant de la connexion, le gestionnaire de fenêtres permettant d'avoir plusieurs fenêtres, et l'environnement de bureau intégré proprement contenant utilitaires et applications.

Il y a des programmes conçus spécifiquement pour KDE, mais cela n'empêche pas dans la pratique de lancer indifféremment une application écrit pour KDE (QT) ou GNOME (GTK+). C'est la bibliothèque de 'widgets' qui fera la différence car sous KDE, l'ensemble des éléments comme les boutons radio, cases à cocher, onglets...seront propres à un environnement de développement (QT ou GTK). Mais au sein d'un même bureau GNOME, il sera tout à fait possible d'utiliser une application pour KDE, et donc d'utiliser une bibliothèque de widgets, et inversement ; et cela revient à charger en mémoire les deux types de bibliothèques !

## 6.2 Fonctionnalités des bureaux graphiques Gnome : bureau à adopter ?

GNOME offre un environnement de bureau homogène, riche, stable, sobre et autant que configurable que KDE. Un ensemble d'outils et de fonctionnalités s'intègrent toujours mieux au bureau par harmonisation du 'look' des différentes applications.

### Configuration et gestion du système

Gnome est configurable par le menu 'Système/Preferences...' qui permet de choisir directement ce que l'on souhaite modifier et présente un point d'entrée sur la configuration du système permettant d'accéder très rapidement aux paramètres importants du système Linux ; alors que l'interface KDE est accessible par le menu 'K/Favoris/Configuration du système', puis deux onglets 'Général' et 'Avancé'.

### interface encore plus simple et rapide avec une barre de lanceurs Unity Launcher

L'interface enrichie avec la barre Unity permet un accès encore plus rapide à un ensemble de raccourcis vers les applications installées et les applications ouvertes. Des indicateurs (application indicators) affichent l'état de certains des périphériques et des logiciels courants. La recherche des documents est facilitée par la barre de recherche du Dash.

## 6.3 Logithèque riche par de nombres applications et logiciels intégrés à GNOME

Les applications ou programmes restent accessibles et sont rangés par thèmes via les menus 'Système' et 'Applications' quand c'est Gnome Classic. Sinon, c'est la barre UNITY (à gauche de l'écran) qui devient le point d'entrée avec le tableau de bord (ou le Dash) et le lanceur d'applications (ou launcher). De plus, Ubuntu fait peau neuve avec la nouvelle méthode 'HUD' (Head-Up Display ou affichage tête haute) de parcours ou de recherche de logiciels...(au lieu de se faire directement à partir des menus). La logithèque Ubuntu (ou Ubuntu Software Center) est maintenant un catalogue de plusieurs milliers d'applications accessible en un simple clic depuis l'installation d'Ubuntu.

### 6.3.1 Bureautique et multimédia

#### Produits bureautique de type office

Libreoffice (en version 3.5.2) est le logiciel de suite bureautique leader permettant de faire du traitement de texte, du tableur, de la présentation, du dessin, de la base de données et du dessin vectoriel ; cela est possible par ses composants API ouverts, un format de données XML et un format 'opendocument' ouvert.

#### Gravure avec K3b et Brasero

K3b et Brasero permettent de graver des CDs et des DVDs autant dans les environnements GNOME que KDE.

#### Audio et Vidéo

Sous KDE c'est Amarok qui est le lecteur audio libre disposant de beaucoup d'utilitaires car il interagit avec des logiciels multiples et externes (K3b, des Web services...) et avec des matériels comme l'iPod. Sous GNOME c'est Banshee qui est à l'honneur en tant que lecteur audio ; Totem est le lecteur vidéo bien pratique. Pour faire du montage vidéo c'est Pitivi qu'il faut utiliser. Rhythmbox (en version 2.96) devance le logiciel Banshee et permet la lecture audio libre tout en mettant de l'organisation dans les fichiers musicaux ainsi que l'écoute de webradios grâce à la plateforme de développement GStreamer sous GNOME. Totem (en version 3.01) permet également de lire les vidéos.



### 6.3.2 Graphisme

#### Okular, Evince

Okular et Evince sont des visionneuses de documents sous KDE et GNOME supportant un grand nombre de formats de fichiers comme : pdf, ps, eps, dvi, jpj, png, bmp, ...

#### Gwenview, Shotwell

Gwenview est la visionneuse d'images très simple d'utilisation mais aux capacités très étendues par l'ajout de plugins. Shotwell (en version 0.12) est également un gestionnaire de photos très simple d'utilisation.

#### Gimp

Gimp permet de manipuler et de traiter des images bitmap ; il est une alternative libre au logiciel Adobe Photoshop.

### 6.3.3 Internet

#### Mozilla Firefox et messagerie

Mozilla Firefox et Thunderbird (en version 11) permettent de naviguer sur internet et de lire ses e-mails.

#### messagerie instantanée et réseaux sociaux

La messagerie instantanée se fait avec MSN, JABBER et EMPATHY ; l'accès aux réseaux sociaux (Twitter, Facebook et Identi.ca) est facilité. Kopete est quant à lui le logiciel de messagerie instantané (sous KDE) reposant sur une nouvelle bibliothèque et supportant une grande quantité de protocoles différents dont jabber, msn, yahoo,...

#### Ekiga

C'est le logiciel par excellence de vidéoconférence et de téléphonie, et ce quelque soit l'environnement de bureau (KDE ou GNOME) ; il permet également d'appeler et de recevoir des appels de téléphones portables ou fixes ; il possède aussi une messagerie instantanée.

Et bien d'autres logiciels à découvrir...

## Chapitre 7

# Gestion des paquets : mises à jours et installations de programmes

Il s'agit de voir comment est-ce possible d'installer des programmes et de les mettre à jour, d'installer de nouveaux logiciels sous Linux hors paquets. Par ailleurs, est-il possible de jouer à des jeux sous Linux ?

### 7.1 Gestion des paquets et des mises à jours

En fonction de la distribution Linux, en l'occurrence dans ce cours, dans la distribution Kubuntu, les paquets s'installent et se mettent à jour selon un format de fichier '.DEB' (format utilisé par DEBIAN) ; ces paquets sont en cache dans la distribution ou alors peuvent se télécharger sur internet. Souvent il existe un gestionnaire de paquets graphique selon l'environnement de bureau dans lequel on se trouve. Sous Kubuntu, c'est le programme 'KpackageKit' (ex Adept Manager sous KDE 4.6) qui fait le travail presque tout seul ; il est accessible par le menu K/Applications/Système/Gestion de logiciels. Sous Ubuntu et dans l'environnement GNOME c'est le programme 'Synaptic'. Et dans tous les cas et quelque soit l'outil de l'environnement, c'est le puissant programme de DEBIAN (APT Advanced Package Tool) qui est à la base car cet ensemble d'outils permet de gérer une base de données de tous les paquets disponibles pour la distribution en question. Alors, soit il faut utiliser l'interface graphique, ou alors soit des commandes en ligne de commande (avec `apt-get install NomPaquet`, `apt-get update`, `apt-get remove NomPaquet`...). Une autre commande qui gère de façon optimale les dépendances : `aptitude install NomPaquet`, `aptitude remove NomPaquet`, `aptitude update`, `aptitude upgrade`...).

Via l'interface graphique, il est possible de consulter les paquets installés sur la distribution Linux, d'effectuer des recherches par mot-clé, d'installer d'autres paquets et de voir les dépendances de tel ou tel paquet. Alternativement à la ligne de commande (dans un shell), c'est surtout l'outil graphique 'Synaptic' (par le menu K/Applications/Système/Gestion de logiciels ou en tapant la commande 'synaptic') qui est souvent utilisé pour installer et désinstaller des paquets ou logiciels.

Par ailleurs, il existe un autre outil ou commande que 'apt-get', c'est 'aptitude' qui permet de mieux gérer les dépendances entre les logiciels à plusieurs niveaux : une recherche plus facile des paquets obsolètes, la consignation des modifications dans un fichier de 'log' et une recherche plus avancée incluant les versions. C'est alors qu'il existe une série de commandes : pour installer un paquet (`aptitude install NomPaquet`), désinstaller un paquet (`aptitude remove NomPaquet`), chercher un paquet (`aptitude search NomPaquet`), supprimer un paquet et ses fichiers de configuration (`aptitude purge NomPaquet`), ... Pour cela, ces commandes pourront s'exécuter dans une interface en mode texte : une première de menus par laquelle on accède avec la souris ou encore avec les touches CTRL+T ou F10 ; et en dessous on peut accéder à une arborescence des paquets classés par état, par catégorie et par dépôt.

Derrière toutes ces interfaces et ces commandes, la distribution ubuntu possède des sites internet HTTP et FTP (mirroirs) sur lesquels se trouvent tous les dépôts de base, de mises à jour et de sécurité; les liens internet sont répertoriés dans le fichier `/etc/apt/sources.list`. Les branches des dépôts des programmes Ubuntu sont réparties en quatre sections : `main` et `restricted` qui sont les deux sections maintenues par les développeurs de Ubuntu; puis, `universe` et `multiverse` qui sont les deux sections des dépôts officiels. Il est possible de rajouter, dans le `'sources.list'`, des dépôts partenaires pour le multimédia (comme par exemple `'Medibuntu'` pour Ubuntu ou `'debian-multimedia'` pour Debian).

Dans ce cas, il faut relancer la synchronisation avec le contenu des dépôts par la commande : `'sudo aptitude update'`. Pour rechercher des informations sur un paquet ou rechercher un paquet dans les dépôts, c'est par la commande : `'sudo aptitude search [nomdupaquet]'`; cela renvoie l'ensemble des paquets comportant ce nom de paquet dans leur description; et le premier caractère de chaque ligne désigne l'état courant ou actuel du paquet, à savoir :

- la lettre `'p'` signifie que le paquet n'est pas présent sur le système d'exploitation Ubuntu;
- la lettre `'i'` signifie que le paquet est installé; ...

## 7.2 Installation de programmes hors paquets

En revanche, quand un logiciel ne se trouve pas en paquet (`.deb`), alors il reste la solution de télécharger sur un internet un fichier au format `'XXXX.deb'` et à l'installer par la commande `'dpkg -i XXXX.deb'`; ou alors, de télécharger une archive compressée en `'XXXX.tar.gz'` ou en `'XXXX.tar.bz2'`. Pour cela, il faut lancer son navigateur internet ou utiliser la commande `'wget'` avec le lien de l'adresse de téléchargement; puis après, il faut décompresser l'archive dans un répertoire qui va contenir soit un script exécutable de suite, soit les sources du programmes qu'il faudra compiler.

Si c'est ce dernier cas qui se présente, il faudra dans un premier temps :

- 1 - lire le fichier ou les fichiers `'README'` et `'INSTALL'` afin d'obtenir des informations sur l'installation du programme;
- 2 - rendre exécutable le programme et pour cela, il faut d'abord le configurer (par la commande `configure` ou `'./configure'`) afin de créer le fichier `'Makefile'` qui contient toutes les instructions nécessaires à la compilation du code source;
- 3 - exécuter la commande `'make'` permet de lancer la compilation et enfin Il faut passer sur les droits `'root'` avec plus de privilèges (et donc de changer d'utilisateur) pour installer les programmes hors de son répertoire personnel;
- 4 - puis enfin, exécuter la commande `'make install'` pour copier les fichiers binaires dans le répertoire d'installation.

Il arrive quelques fois que lors de la configuration, des erreurs surviennent du fait de l'absence d'outils de compilation (comme `gcc`, `g++`). Mais l'installation de paquets téléchargés et précompilés évitent ces mésaventures. Quelquefois, l'installation de programmes trop récents et le risque que peut engendrer une compilation à outrance peuvent ouvrir une autre possibilité qui est celle de changer de version de distribution ou même de distribution!

## 7.3 Est-il possible de jouer avec sa distribution Linux ?

Il est tout à fait possible de jouer sous Linux car il existe une multitudes de programmes de jeux libres intégrés à sa distribution Linux ou bien alors sur des portails internet. Et surtout, de plus en plus les éditeurs de logiciels prévoient les mêmes jeux qui existent pour le monde Windows (Doom, Quake...).

Dans la réalité, soit on garde fondamentalement un dual-boot avec Windows et Linux, soit il est possible d'utiliser un émulateur (comme Wine) sous Linux afin de lancer des jeux Windows, ou alors on utilise une solution de virtualisation (comme Virtualbox) en installant un Windows sous sa distribution Linux ; cette dernière solution ne garantit pas du tout le fonctionnement normal des jeux sollicitant beaucoup l'interface graphique par exemple.

Si l'on s'en tient à sa propre distribution Linux Kubuntu, il est possible de trouver un certain nombre de jeux sous KDE : Killbots (poursuite par des robots), Kapman, Ksirk (conquête du monde), Kmines (démineur), KBattleship (bataille navale), Bomber (lâcher de bombes en avion), une série de jeux de carte, une série de jeux de logique...et Pinball (un flipper en D).

A côté de cela, il est possible de trouver des projets comme 'Wormux' (un Worms-like avec des bataillons de vers) ou le jeu Teeworlds ; il y a également des jeux de guerre comme 'Spring'. [Lien internet vers Spring](#) ; des jeux de course de voiture comme 'SuperTuxKart', 'VDrift', 'Tux Racer', 'Trigger', 'TORCS'... Un portail internet 'Jeuvinux' permet de se tenir au courant de l'activité ludique sous Linux. [Lien internet vers Jeuvinux](#)  
<http://www.jeuxlinux.fr/>

Troisième partie

# EXPLORATION DU SYSTEME LINUX

# Chapitre 8

## Connexion et session

Chaque utilisateur du système se voit attribué un environnement de travail qui se compose d'un login, d'un mot de passe, d'un répertoire personnel d'accueil, d'un identifiant de groupe et d'un shell. Toutes ces informations sont consignées dans le fichier `/etc/passwd` accessible uniquement par l'administrateur.

### 8.1 Type de connexion : session et fin de session de travail

Une fois l'installation de Kubuntu terminée, l'ordinateur redémarre et il sera possible d'ouvrir une session de travail soit graphique (via un écran de connexion KDE ou GNOME), soit en mode terminal et texte. Une session consiste à saisir un nom de connexion (ou identifiant ou username) et un mot de passe (ou password) afin d'accéder à un compte utilisateur sur le système d'exploitation Kubuntu. Dans le cas de Kubuntu c'est KDM 'KDE Display Manager' (et GDM pour GNOME), le gestionnaire d'affichage pour le système de fenêtre X qui permettra de saisir l'identifiant et mot de passe, et également de choisir des options de démarrage. Une fois ces actions faites, une session s'ouvre jusqu'au moment où l'action 'déconnexion' (ou éteindre) soit faite.

Une session de travail graphique repose sur un très vieux projet graphique appelé 'X Window' ou 'Xorg' ou 'X11'(système de fenêtrage) ; ce dernier installe un terminal X suite au démarrage du serveur X (équivalent à la commande `'startx'`). Et c'est sur ce socle que reposent le gestionnaire de fenêtres et tout l'environnement de bureau (KDE ou GNOME ou d'autres). Concrètement, ce serveur Xorg se configure avec plusieurs sections dans un fichier `/etc/X11/xorg.conf` ; le bon affichage de l'interface graphique en dépend directement. Par ailleurs, sous Ubuntu, il existe un gestionnaire de fenêtres appelé 'Fluxbox' qui est particulièrement léger et peu exigeant en ressources.

[lien pour télécharger le manuel de Linux Ubuntu en version 13.10](https://ubuntu-manual.org/) `https://ubuntu-manual.org/`

#### 8.1.1 Gestion des utilisateurs

Kubuntu est multi-utilisateurs car chaque utilisateur (ou futur utilisateur) dispose de son propre compte, des droits attachés et de ses propres dossiers courants (Documents, Bureau, Images...) et quelques fichiers cachés `'.XXXXX'`. Dès sa connexion l'utilisateur se trouve systématiquement dans son répertoire d'accueil `'/home/nomutilisateur'`. La distribution Ubuntu est à peu près la seule distribution GNU/Linux, et de façon générale, à ne pas comporter de compte du super-administrateur 'root' ; c'est lors de l'installation de la distribution Kubuntu qu'un premier compte utilisateur sera créé et il sera assimilé à un compte dit administrateur ; pour cela, lors d'une action critique sur le système le mot de passe sera demandé de nouveau via une fenêtre ou boîte de dialogue.

### 8.1.2 Gestion des mots de passe

Les droits d'administration se gèrent sous Kubuntu par un mécanisme de 'sudo' (superuser do) ou kdesu (gksudo sous GNOME) et se situent à un niveau au-dessous des droits du super-utilisateur (compte de pleins pouvoirs sur le système). Donc, l'utilisateur par ce mécanisme de 'sudo' peut effectuer certaines tâches d'administration en lançant la commande 'sudo NomCommande'. Par ailleurs, des applications graphiques peuvent exiger de passer dans ce mode en demandant de nouveau le mot de passe. C'est alors que le mot de passe du compte 'root' est bloqué et par conséquent ne permet d'utiliser la commande 'su -'. Le 'sudo' s'utilise dans un shell de commandes alors que 'kdesu' lors du lancement des applications graphiques.

## 8.2 le mode console : le shell bash

Le shell Bash est un programme ou une interface pour communiquer avec le système Linux Ubuntu. il est lié à chaque utilisateur du système et comporte un jeu de caractères spéciaux, un ensemble de commandes internes et des fichiers d'initialisation et de configuration de l'environnement de travail. Plus concrètement, le shell est un interpréteur de commandes accessible via un émulateur de terminal (xterm ou Gnome terminal) qui permet de naviguer dans le système de fichiers, de créer ou de modifier des fichiers et/ou des répertoires, d'exécuter des commandes et/ou des fichiers de scripts.

Historiquement, il existe plusieurs sortes de shell : le plus courant et celui par défaut est le bash ('/bin/bash'), le Korn shell ('ksh'), le 'tchsh' ou le C-shell 'csh' . Depuis 1977, bash ou le shell Bourne-Again est devenu le shell par défaut sur de nombreuses plate-formes GNU/Linux ou Mac OS X et donc l'outil puissant et ultime quand il n'y a plus d'interface graphique KDE ou GNOME accessible.

Au moment de la connexion de l'utilisateur au système, le shell Bash exécute le fichier '/etc/profile', puis le fichier './profile' (ou le fichier './bash\_profile' si './profile' n'existe pas!), et après cela active le fichier './bashrc' qui comporte des options de personnalisation du shell propre à l'utilisateur. Le fichier '/etc/skel' est lu également car il peut contenir des modèles de fichier de démarrage qui sont recopiés automatiquement dans le répertoire personnel de l'utilisateur au moment de la création du compte utilisateur.

Suite à l'ouverture d'une session, le système Linux place l'utilisateur dans son répertoire personnel d'accueil mais lance également un programme ou une interface shell permettant d'interagir avec le noyau linux. L'utilisateur est donc connecté au système et retrouve avec un environnement de connexion composé d'un terminal et d'un shell. Et cet environnement possède plusieurs variables sous la forme 'VARIABLE=valeur' et souvent avec des valeurs prédéfinies; les commandes 'env' ou 'printenv' permettent de voir toutes ces variables et on peut trouver :

la variable 'HOME' = /home/nomlogin qui est le répertoire principal ou d'accueil où commence la session

la variable 'LOGNAME' = login ou valeur du nom de session

la variable 'PATH' = la liste des répertoires dans lesquels le shell recherche des commandes

la variable 'PWD' = l'emplacement ou la position dans le système de fichiers

la variable 'SHELL' = l'emplacement du programme shell utilisé mais qui peut être modifié dans le fichier './profile'.

Après le processus de démarrage et de connexion, voici un peu plus dans le détail et dans l'ordre des explications sur les fichiers de démarrage :

/etc/profile : sert à définir des variables d'environnement; il va exécuter tous les fichiers de script avec l'extension '.sh' qui se situent dans le répertoire '/etc/profile.d/'; le fichier '/etc/bash.bashrc' sera également exécuté.

./bash\_profile : contient les variables d'environnement et la configuration personnelles; '/etc/bashrc' contient quant à lui les variables d'environnement système.

`/.bashrc` : contient les alias et les fonctions personnelles de l'utilisateur et sera lu ou chargé à chaque ouverture d'un terminal. Si des modifications sont apportées aux variables d'environnement et pour qu'elles soient prises en compte, il faut pater la commande `source ~/.bashrc` ou fermer et ré-ouvrir le terminal. Les modifications possibles de variables sont les suivantes :

- l'invite de commandes ou le prompt, attend des commandes et par défaut l'utilisateur se trouve dans le répertoire personnel courant noté `' '` (ou HOME) ; pour cela, il faut modifier la variable `'PS1'`, mais pour connaître sa valeur, il faut saisir la commande `'echo signedollarPS1'` ;
- la variable `'PATH'` qui contient les répertoires ou les chemins lors du lancement des commandes ;
- il est possible de configurer des alias ou des commandes raccourcies lorsque des commandes sont utilisées fréquemment ; exemple : `alias PS='ps aux | grep'` ;
- la variable `'HOME'` est le chemin absolu vers le répertoire personnel de l'utilisateur ;
- la variable `'TERM'` est le type de terminal utilisé.

`/.bash_logout` : sert à la fermeture du shell et donc de la déconnexion de l'utilisateur.

Par ailleurs, la complétion automatique des commandes au sein du shell est très pratique : moins de frappe et d'erreurs de saisie au clavier ! ; pour cela, il faut appuyer sur la touche [TAB] (ou tabulation) afin que le shell complète automatiquement le début de la saisie et en affichant la liste des possibilités (en faisant deux tabulations) ; c'est le fichier `"/etc/bas_completion"` qui comporte les règles d'autocomplétion.

L'historique des commandes est quant à lui stocké dans le fichier `"/.bash_history"` du répertoire personnel de l'utilisateur ; la commande `'history'` permet de lister toutes les commandes saisies associées à un numéro ; pour rappeler la commande précédente il faut appuyer sur la touche flèche vers le haut (ou !!) ; pour rappeler rapidement une commande il faut saisir `'!n'` (où n est le numéro de la commande) ; et ce qui peut être très pratique c'est d'avoir la date et l'heure devant chaque historique de commande passé ; pour cela, il faut saisir la variable `HISTTIMEFORMAT='%D %H :%M '` et la commande `'export HISTTIMEFORMAT'`.

L'environnement du terminal est un pilote de périphérique nécessaire pour la saisie et l'interprétation des commandes ; le terminal étant défini comme un pilote de périphérique et comme tout est fichier sous Linux, alors vous pouvez visualiser les paramètres de votre terminal avec la commande `'stty -a'`.

Pour communiquer avec le shell via un émulateur de terminal, il y a plusieurs possibilités et donc de lancer soit :

- le programme `'Xterm'` qui est le plus ancien ;
- GNOME Terminal qui se trouve dans le menu Applications/Accessoires/Terminal quand on est sous Gnome Classic avec la possibilité d'utiliser des profils et des touches de raccourci clavier ;
- le programme `'Konsole'` qui est l'émulateur de terminal intégré à l'interface de bureau KDE ;
- le programme `'LXTerminal'` est quant à lui intégré à l'environnement de bureau LXDE ;
- le programme `'ROXTerm'`, similaire à Gnome Terminal, est très rapide au démarrage et très configurable.

Le shell reste le moyen d'interpréter des commandes en ligne de commandes et donc un programme servant d'interface entre l'homme et l'ordinateur. Pour cela, il est courant d'ouvrir une interface de base appelée console ou shell.

Il existe quelques raccourcis clavier pour naviguer dans le shell Bash : [CTRL]+[P] pour rappeler la commande précédente [CTRL]+[N] pour rappeler la commande suivante [CTRL]+[A] pour aller au début de la ligne [CTRL]+[E] pour aller en fin de ligne [CTRL]+[D] pour supprimer un caractère [CTRL]+[K] pour supprimer tout ce qui est à droite du curseur [CTRL]+[U] pour supprimer tout ce qui est à gauche du curseur



## 8.3 la ligne de commande et le prompt

La ligne de commande est donc une l'interface utilisateur (ou un langage) avec une syntaxe particulière et un prompt nécessitant un certain apprentissage. Cette interface est personnalisable, tout comme l'environnement KDE, afin de la rendre proche des besoins d'utilisation. L'interface est également automatisable car reposant sur un langage qui permet de manipuler à outrance le système et de réaliser des programmes dits scripts shell. Sa légèreté fait qu'elle ne nécessite aucune ressource mémoire ou processeur, contrairement à l'environnement KDE ou GNOME.

Cette interface de base est accessible par un émulateur de terminal qui est un programme spécifique 'getty'; c'est en fait un écran/clavier virtuel avec 6 entrées dans une partie de l'arborescence '/dev/tty1 à tty6'; ces terminaux sont accessibles par la séquence des touches CTRL + ALT + F1 à F6. L'accès dans l'environnement KDE ou GNOME se fait par ailleurs par application graphique émulant un terminal; pour cela, soit il s'agit de lancer une fenêtre XTERM, ou alors 'Gnome-terminal' pour GNOME ou encore 'Konsole' par le menu K/Applications/Système/Konsole'. (programme yakuake)

Par exemple, si l'utilisateur 'formation' (ou étudiant) a ouvert une session de travail sur l'ordinateur 320B-03, alors le prompt sera :

```
formation@320B-03 :signedollar, etudiant@320B-03 :signedollar (et le shell attend donc une commande...)
```

## 8.4 Type et format de commande

Une commande est un programme qui sera appelé à être exécuter via un interpréteur de commandes qui est le shell. Celui-ci va réagir en fonction des commandes qui seront saisies et ce sont les possibilités du système d'exploitation qui vont permettre d'exécuter telle ou telle commande.

Pour cela, il existe plusieurs types de commandes : celles internes au shell (fonctions intégrées au noyau Linux) qui sont exécutées directement, comme par exemple les commandes 'ls -l' ou 'cd'; puis celles qui sont des programmes exécutables (ou binaires) faisant partie de l'arborescence du système et enfin des alias qui sont des commandes de type pointeur ou raccourci. Ainsi les commandes 'alias=...' ou 'ln -s chemin nonraccourci' définissent des raccourcis sur des fichiers ou des commandes.

Dans le cas, où la commande XX n'est pas trouvée ni par le noyau et ni au sein de l'arborescence du système, alors un message apparaît comme suit :

```
bash :XX :command not found
```

Sous Linux il existe deux mécanismes particulièrement intéressants pour l'utilisateur : la complétion et l'historique des commandes.

La plupart des commandes Unix/Linux disposent de nombreuses options qu'il est possible de juxtaposer derrière la commande elle-même ; il existe un certain nombre de commandes Shell qui peuvent être très utiles ; leur syntaxe est :

nomcommande [-options] [-options] [argument 1] [argument 2] ... (en sachant que ce qui est entre crochets est facultatif et tout est séparé par un blanc ou un espace).

[documentation sur les commandes système linux](http://doc.ubuntu-fr.org/commande_shell) [http ://doc.ubuntu-fr.org/commande\\_shell](http://doc.ubuntu-fr.org/commande_shell)

### 8.4.1 Majuscules/Minuscules

Un message d'erreur peut provenir du fait que la commande respecte une certaine casse avec des majuscules et des minuscules. Le système Linux (comme tout Unix) attend une certaine exactitude car il respecte la casse.

### 8.4.2 Caractères spéciaux

La distribution Linux est facile à utiliser de par la gestion des majuscules accentuées et caractères spéciaux ; il est nullement besoin d'utiliser la combinaison de touches ALT+XXX, mais plutôt touche 'CAPS-LOCK' ou 'ALTGR' ou 'SHIFT' ; par exemple : pour avoir le caractère '«' il faut utiliser la combinaison 'ALTGR+w'. Une commande peut être saisie dans le shell avec une sélection spécifique et donc, avec des caractères spéciaux ; comme par exemple : la commande 'ls -l Doc\*' affiche toute la liste des fichiers et répertoires commençant par 'Doc' ; ou encore la commande 'ls /var/log/\*.log' qui affiche tous les fichiers '.log' du répertoire. Ainsi le caractère étoile '\*' remplace une suite de caractère quelconque, le caractère point d'interrogation '?' remplace le caractère en question.

Etant donné que l'ordinateur ne comprend le texte en lui-même, alors chaque caractère est représenté par un nombre. Traditionnellement, chaque jeu de nombres est utilisé pour représenter un alphabet et des caractères appelé un système de codage ou d'encodage des caractères. Le jeu de caractères le plus commun, ou du moins le plus universellement accepté, est l'ASCII (American Standard Code for Information Interchange). Et afin d'éviter que chaque pays standardise son propre codage (comme en europe occidentale avec la norme ISO-8850-15 ou Latin9, ISO-8859-1 ou Latin1), en 1991 une table Unicode (UTF) mondiale fut normalisée pour être ensuite transformée dans la version normalisée UTF-8. Et c'est bien ce dernier jeu de caractères qui est supporté par tous les navigateurs internet la plupart des systèmes d'exploitation Linux (et Windows) y compris les programmes de bases de données.

### 8.4.3 Répertoire d'accueil et variable 'HOME'

Le répertoire courant de l'utilisateur est '/home/nomutilisateur' ; c'est à partir de cet endroit qu'il a les pleins pouvoirs et qu'il peut organiser ses documents, mais s'il se déplace dans l'arborescence des répertoires, il peut revenir très rapidement dans son 'home' par la simple commande 'cd ~' (cd espace tilde). Lors de son déplacement, un simple point '.' indique le répertoire lui-même et un double point '..' le répertoire parent ou situé au-dessus.

Par ailleurs, deux fichiers importants sont présents dans le 'home' de l'utilisateur : '.bashrc' est le script d'exécution des commandes avant la session et le '.profile' qui est le script de positionnement de l'environnement de l'utilisateur ; et c'est ce dernier qui lance le fichier '.bashrc'. Ces fichiers sont modifiables afin d'aménager la session avec les commandes dans le shell. Ainsi, la commande 'echo "dollarUSER"' renvoie le nom d'utilisateur ; la commande 'echo "dollarHOME"' renvoie le répertoire d'accueil de l'utilisateur.

### 8.4.4 Redirections et tubes

Lors d'une session de travail, l'utilisateur effectue un ensemble d'actions (exécution de commandes dans le shell, impression de documents...) qui sont des flux ou des redirections. C'est ainsi que cela commence par une entrée qui est le clavier et une sortie (résultat attendu) vers l'écran ou une imprimante.

Il est possible de rencontrer sur un système d'exploitation plusieurs types de redirections : des fichiers qui enregistrent des messages d'erreurs, un périphérique comme une imprimante et des tubes qui redirigent la sortie des commandes vers d'autres commandes. C'est alors qu'on utilise le caractère '>' pour rediriger le résultat d'une commande vers un fichier, le caractère '>>' pour rediriger ou ajouter le contenu à la suite d'un fichier existant ; per exemple : la commande 'ls /home/Documents > mesrepertoires.txt' redirige la liste des répertoires se trouvant sous '/home/Documents'.

Le tube qui est représenté par un tiret vertical '|' relie une sortie de commande vers l'entrée d'une autre commande ; par exemple, la commande 'lspci | grep Intel' affiche le détail d'un périphérique Intel s'est trouvé.

### 8.4.5 Commandes de déplacement

Pour se déplacer dans l'arborescence du système, plusieurs commandes sont utilisées :

La commande 'pwd' (print working directory) afin de savoir où est-ce que l'utilisateur se situe ? ;

La commande 'cd' (change directory) pour se déplacer dans un dossier de façon absolue ou relative ;

La commande 'ls' (list segments) pour lister le contenu d'un dossier (fichiers et sous-dossiers).

### 8.4.6 Expressions régulières

Ce sont des motifs constitués d'un seul ou d'un groupe de caractères permettant de désigner tout un ensemble de possibilités de complétion. Elles sont donc utilisées par de nombreuses commandes Linux car elles facilitent souvent les opérations sur les fichiers...

Il y a deux sortes d'expressions régulières : les basiques et les étendues. Cependant, il y a des caractères communs aux deux sortes :

- le caractère ^ qui représente le début de la ligne ;
- le caractère \$ qui représente la fin de la ligne ;
- le caractère \* qui représente 0,1 à n fois le caractère qui précède ;
- le caractère () qui représente un groupement d'expressions ;
- le caractère . (point) qui représente un seul et unique caractère quelconque ;
- le caractère [liste\_de\_caractères] qui représente l'un des caractères compris dans les crochets ;
- le caractère [^ liste\_de\_caractères] qui représente un caractère différent de ceux qui sont compris dans les crochets.

Exemple d'utilisation :

- pour rechercher toutes lignes commençant par la lettre R, on aura la commande linux suivante : grep -E '^ R' Monfichier ;

- pour rechercher les lignes comportant un terme exact, on aura la commande : grep -E 'expression régulière' Monfichier ;

- pour renvoyer toutes les lignes qui contiennent un caractère alphabétique compris entre a et z ou entre A et Z, on aura la commande : grep -E [a-zA-Z] ~.bashrc ;

On utilise soit la commande 'grep' ou 'grep -E' ou 'egrep' ; 'grep' signifie 'Global Regular Expression Print' et utilise les expressions régulières pour effectuer des recherches et permet de : rechercher une chaîne de caractères dans des fichiers, de filtrer.. .

## 8.5 Démons et processus

Le système Linux démarre et charge le noyau et le programme 'init'. Ce programme est le père de tous les autres programmes et donc des processus lancés. La plupart du temps, le système Linux alloue le processeur à un programme pendant un quantum de temps très court, puis le fait passer à un autre et ... ; le fait que le délai de temps alloué à chaque programme soit très court alors donne l'impression que tout fonctionne en même temps ! L'état des processus endormis est préservé pendant que le processus actif s'exécute.

Un processus est donc l'exécution d'un programme à un moment donné ou son état en cours d'exécution ; il est représenté par un identifiant appelé PID (Process IDentifier). Un processus ou un programme qui dispose de sa propre zone mémoire et ses propres variables est dit lourd.

Par exemple, le programme OpenOffice lance un processus qui lui est propre ; mais, il qu'un programme, processus père, lance d'autres processus appelé fils, c'est le cas de 'init'.

Pour cela, si l'utilisateur lance la commande 'pstree', il peut voir les processus sous forme d'arborescence et donc voir leurs inter-dépendances.

La commande 'top' permet d'afficher des informations en continu et en temps réel sur l'activité du système ; elle permet surtout de suivre les ressources que les processus utilisent (quantité de RAM, pourcentage de CPU, la durée de ce processus depuis son démarrage).

La commande 'ps' permet de connaître tous les processus actifs du système à un moment donné (ou à un instant t) et de procéder à une certaine surveillance ; la commande 'ps aux|grep office' (utilisé avec un tube) permet de rechercher le processus associé au programme 'Openoffice writer'. Cette surveillance peut se faire aussi bien au sein d'un shell de commande qu'avec le moniteur graphique.

La commande 'ps -aux' (avec les 3 options 'a', 'u' et 'x') permet d'afficher (sur 11 colonnes) les processus de tous les utilisateurs, leurs noms, l'heure de lancement des programmes et processus et également les processus en arrière-plan.

Par exemple, la commande 'ps -A u' ramène comme informations : PID, ressources CPU et mémoire et surtout le propriétaire.

```
USER PID CPU MEM VSZ RSS TTY STAT START TIME COMMAND
rferrere 10222 0.0 0.0 5984 3168 pts/3 Ss+ 16 :47 0 :00 /bin/bash
rferrere 11408 0.2 1.9 192280 65792 ? Sl 17 :06 0 :07 /usr/lib/openoffice/program/soffice.bin -writer -
splash-pipe=6
rferrere 14248 0.0 0.0 2768 1036 pts/2 R+ 17 :58 0 :00 ps -A u
```

Par ailleurs, l'utilisateur peut agir sur ces processus : changer leur priorité, mais la plupart du temps les arrêter ou encore les tuer ! Pour cela, la commande 'kill -9 numeroprocesusPID' permet de tuer définitivement avec le signal 9.

Le terme démon (ou deamon Disk And Execution MONitor) quant à lui, désigne un processus lancé en arrière-plan du système. Après le chargement du noyau linux, les modules sont chargés également dans le noyau et des services sont démarrés (par 'initd') en tant que démons. Les démons exécutent des tâches déterminées d'avance de façon régulière ou alors à la suite d'actions ou d'évènements ; par exemple, un serveur web est un service en arrière-plan qui attend les requêtes des clients ; lors qu'une tâche d'impression est soumise alors le démon 'lpd' lance un processus fils ; par ailleurs, ce sont des démons qui gèrent les services réseaux TCP/IP sur lequel reposent un ensemble de services réseau (NFS, SAMBA,...) ; certains démons ont comme nom 'lpd', 'crond', 'httpd'...et d'autres se nomment ssh, samba... ; c'est sous '/etc' que se trouvent les divers démons du système Linux.

## Chapitre 9

# Organisation du système de fichiers

Les systèmes UNIX ont un système de fichiers basé sur une architecture standard de type FHS (Filesystem Hierarchy Standard) qui permet d'organiser les fichiers dans des répertoires et de façon hiérarchique. Les premières versions de Linux ont eu comme système de fichiers 'Minix' qui a été créé à l'origine en 1987 par le professeur Andrew TANENBAUM. L'installation de tout système GNU/Linux consiste à créer des partitions et un système de fichiers inscrit sur ces partitions ; et à ce moment-là il est possible de choisir entre plusieurs types de systèmes de fichiers afin qu'ils soient les plus stables et les plus performants possibles. En 1992, la communauté Linux développe au sein du noyau linux une interface VFS (Virtual File System) capable de s'adapter à n'importe quel système de fichiers. Puis sont arrivés les systèmes de fichiers EXT (2, 3 et 4) développés à l'origine par le français Rémy CARD. D'autres systèmes de fichiers journalisés apparaissent par la suite et encore actuellement : XFS et JFS, BTRFS (prononcé Butter FS et basé sur la technologie copy-on-write sous le sigle anglais 'COW') qui est le dernier en date et développé par la société ORACLE. Le principe sous Linux et Unix, est que 'tout est fichier' à savoir que tout sera considéré comme un fichier (une imprimante rattachée à l'ordinateur sera représentée par un fichier).

Ainsi, le disque dur sera représenté par un fichier spécial, 'hda ou sda' ; le lecteur de DVD sera 'hdc ou sdc' ; la carte son sera 'dsp'.... ; tous ces périphériques se situent sous '/dev'. Les périphériques IDE et SATA se nomment /dev/sda1 à sdaN, la disquette se nomme /dev/fd0, les périphériques USB se nomment /dev/sdb ou /dev/sdc.

### 9.1 Arborescence du système de fichiers

Le sommet de l'arborescence est le répertoire racine '/'. Voici une description succincte des différents répertoires de l'arborescence :

- / = la racine, elle contient les répertoires principaux comme /bin, /dev, /tmp, /home, /etc, /var...
- /bin = contient des fichiers exécutables (ou binaires) essentiels au système, employés par tous les utilisateurs (par exemple, les commandes ls, cd, rm, cp, chmod, ...).
- /boot = les fichiers nécessaires au démarrage du système.
- /dev = les fichiers spéciaux créés au moment de l'installation du système et qui permettent de communiquer avec les périphériques (disques, adaptateur réseau, cartes son etc...).
- /etc = les fichiers de configuration du système (Editing Text Config) et les principaux scripts de paramétrage ;
  - /etc/rcN.d ( 0 < N < 6, N=S) = les scripts de démarrage du système.
  - /etc/X11 = les scripts de configuration du serveur X.
  - /etc/apt = les mises à jour des paquets de logiciels de la distribution Debian et Ubuntu.
  - /etc/init.d = les script de contrôle des serveurs.
  - /etc/event.d/rcN (0 < N < 6, N=S) = les fichiers d'évènements comme l'arrêt et le redémarrage du système.

/etc/cron.d (et /cron.daily, /cron.hourly, /cron.weekly, /cron.monthly) = la description des tâches périodiques à effectuer.

/etc/skel = les fichiers copiés dans le répertoire personnel d'un nouvel utilisateur.

/home = les répertoires personnels des utilisateurs incluant leurs propres fichiers de configuration.

/lib = les bibliothèques partagées utilisées par les programmes binaires du système (modules du noyau, /lib/module/) et les logiciels.

/mnt et /media = la racine des points de montage des périphériques. (mnt pour le montage de partitions système et media pour les périphériques USB).

/opt = lieu d'installation d'applications supplémentaires (java ..).

/root = le répertoire personnel du super-utilisateur root pour lequel tous les utilisateurs standards n'ont pas les permissions d'entrer.

/sbin = les fichiers exécutables pour l'administration du système.

/tmp = le stockage des fichiers temporaires.

/usr = Unix System Resources pour les programmes ressources accessibles à tout utilisateur ; les fichiers exécutables de toutes les applications sont sous '/usr/bin' et les bibliothèques associées sous '/usr/lib'.

/var = les données variables liées à la machine (fichiers d'impression, traces de connexions http, smb .. dans /var/log).

/proc = le répertoire qui représente le point de montage du pseudo système de fichiers du noyau ; ce dernier contient des fichiers permettant d'accéder de façon dynamique aux informations sur le matériel, la configuration du noyau et sur les processus en cours d'exécution et contient donc une image du système avec son état et ses processus en cours..

## 9.2 Chemin d'accès relatif et absolu

Sous Linux il est nécessaire d'indiquer le chemin à effectuer à travers l'arborescence pour atteindre un fichier ou un répertoire ; pour cela, il est courant d'utiliser le caractère slash '/' celui qui relie chaque répertoire d'un chemin. C'est alors qu'on utilise deux types de chemins : absolu et relatif ; le chemin absolu part de la racine de l'arborescence pour arriver de façon absolue au fichier en indiquant le chemin complet, comme par exemple, le chemin '/etc/network/interfaces' pour arriver au fichier 'interfaces'.

Le chemin relatif, quant à lui, dépend directement du répertoire courant où l'utilisateur se trouve et donc au moment où il fait référence au fichier ; par exemple, si l'utilisateur se trouve sous le répertoire '/Bureau' et qu'il souhaite faire référence au fichier 'image.png' qui se trouve dans son répertoire '/Documents/images/', alors le chemin est '../Documents/images/image.png' ; les deux points permettent de remonter d'un cran dans l'arborescence. Par ailleurs, il est courant d'assimiler le caractère tilde '~' au chemin absolu '/home/nomutilisateur/', quand l'utilisateur veut revenir très rapidement dans 'home', ce sera par la commande 'cd ~'.

Et pour savoir à tout moment où l'utilisateur se trouve, il faut utiliser la commande 'pwd' qui renvoie tout simplement le chemin absolu de l'endroit où il se trouve et lui laissant prendre la bonne décision pour la suite.

## 9.3 Organisation des répertoires et fichiers

L'organisation des fichiers et des répertoires s'articule autour d'un système de gestion de fichiers qui gère le stockage des informations. Cela passe d'une part, par un niveau logique qui consiste à gérer les requêtes de lecture/écriture et d'autre part, par un niveau physique via des adresses absolues des blocs sur le support physique.

Les fichiers et répertoires peuvent être perçus comme des objets qui subissent des actions ou des flux d'entrées et de sorties. Sous le système Linux, il est possible de rencontrer plusieurs types de ces objets :

- les fichiers simples ou ordinaires sont des fichiers comme les fichiers texte (.txt, .doc, .odt...), les sons (.wav, .mp3, .ogg), mais aussi les programmes ou les exécutables ; c'est-à-dire tous ceux qui sont manipulés et structurés par les utilisateurs.

- les fichiers standards associés à un descripteur comme respectivement 0, 2 et 1 pour l'entrée (saisie clavier), l'affichage de messages d'erreur et la sortie.

- les répertoires ou dossiers sont des catalogues ou dossiers contenant des fichiers et une description attenante sur la taille, le nom, le type, l'identification du propriétaire... ; le caractère 'd' indique le type de fichier et donc que c'est un répertoire (pour directory) ; et donc, une hiérarchie se crée avec le répertoire racine (/), le répertoire courant représenté par le point (.) et le répertoire parent représenté par les point et point (..).

- les fichiers spéciaux car ils représentent quelque chose comme par exemple, le lecteur CD et/ou graveur CD/DVD ; en mode bloc (b) c'est une bande ou un disque et en mode caractère (c) c'est tous les périphériques de façon générale.

- les liens symboliques sont des raccourcis ou des pointeurs vers des fichiers ordinaires ou des répertoires ; ils sont symbolisés par la lettre 's' devant les droits.

Donc, au sein de l'arborescence linux il est courant de rencontrer plusieurs types de fichiers : le tirt 'l' est un fichier standard, le 'd' indique un répertoire, le 'l' un lien symbolique, le 's' un socket souvent réseau, le 'b' un périphérique de type bloc et le 'c' pour un périphérique de type caractère.

## 9.4 Gestion des protections sur les fichiers

Comme le système Linux est multi-utilisateurs, il est nécessaire de prévoir un système de protection et donc de permissions contrôlant les opérations de chaque utilisateur ; d'autant plus que comme sous Linux (ou Unix) tout est fichier, il faut pouvoir contrôler chaque périphérique qui est un fichier ou un répertoire. Des droits d'accès déterminent si on a les droits de lecture, d'écriture et d'exécution sur tels fichiers et répertoires pour tel utilisateur. Et pour cela, chaque fichier ou répertoire dispose de permissions spécifiques pour trois types d'utilisateurs :

- le propriétaire du fichier ou du répertoire (représenté par 'u' comme user)

- le groupe propriétaire (représenté par 'g' comme group) qui sont les utilisateurs membres du groupe propriétaire du fichier

- et les autres (représenté par 'o' comme other) ou le reste des utilisateurs.

Et sur ces fichiers et répertoires viennent s'appliquer trois types de droits de sécurité (ou d'autorisation) qui peuvent se combiner :

- la lecture (représenté par 'r' pour 'read' ou le chiffre 4 en valeur octale) ;

- l'écriture ou la modification (représenté par 'w' pour 'write' ou le chiffre 2 en valeur octale) ;

- l'exécution (représenté par 'x' pour execute ou le chiffre 1 en valeur octale).



Par exemple, la commande `'ls -l /Documents'` donne :

```
drwxr-xr-x 15 rferrere rferrere 4096 2009-09-21 20 :37 Documents
```

Le répertoire `'Documents'` ayant les droits positionnés à `755` signifie que le propriétaire a tous les droits de lecture, écriture et d'exécution sur le répertoire et les fichiers inclus dedans, mais que le groupe et les autres n'ont que le droit de lecture/exécution et pas le droit d'écriture. Alors, par exemple pour ajouter au groupe et aux autres le droit d'écriture, cela peut se faire par la commande `'chmod 777 /Documents'` ou alors `'chmod go+w /Documents'`.

C'est ainsi que l'on utilise le signe moins `'-'` pour enlever un droit, le signe plus `'+'` pour ajouter un droit et le signe égal `'='` pour assigner un droit.

## 9.5 Création et destruction de répertoires et de fichiers

Un fichier sur le système Linux représente à la fois une source de données à lire ou une destination pour ces données à écrire. Un fichier possède un descripteur appelé noeud d'index ou `'inode'` et le système de fichiers tient à jour la table d'inodes.

La création de fichiers se fait soit en utilisant un éditeur (comme `'Vi'`, `'Kate...'`) et en donnant au fichier, soit en utilisant la commande `'touch nomdudossier'`; cette dernière s'utilise pour créer un fichier vide ou changer la date de dernière modification, ce qui est souvent utile pour l'utilitaire `'make'`. La destruction de fichiers se fait en utilisant la commande `'rm nomdudossier'` et la commande `'rm -r'` effectue la suppression de façon récursive.

**ATTENTION à la commande `'rm -rf /*'` qui est souvent destructrice!!! et EFFACE TOUT le disque dur...**

La création de répertoires se fait par la commande `'mkdir nomdurepertoire'` en utilisant soit des chemins relatifs ou des chemins absolus. Et la destruction de répertoires se fait également par la commande `'rm -r nomdurepertoire'`; pour supprimer un répertoire vide il faut utiliser la commande `'rmdir nomdurepertoire'`.

Avant toute destruction de fichiers et de répertoires, il est possible d'éditer les contenus des fichiers et de lister le contenu des répertoires et sous-répertoires; Pour éditer un fichier, il est possible d'utiliser les commandes `'cat nomfichier'`, `'more nomfichier'` ou `'less nomfichier'`. Pour lister le contenu d'un répertoire, la commande `'ls'` ou `'ls -l'`.

## 9.6 Création et destruction de liens

Les liens sont souvent des chemins d'accès (absolus ou relatifs) vers des fichiers ou répertoires, encore appelés pointeurs (ou liens matériels) vers des noeuds d'index. La commande `'ls -i nomfichier'` ou `'ls -i nomlien'` permet de constater que l'on obtient 2 numéros d'inodes différents pour un lien symbolique alors que ce sont les mêmes pour un lien ordinaire.

Pour créer un lien ordinaire, c'est la commande `'ln fichier1 fichier2'`; pour créer un lien symbolique, c'est la commande `'ln -s fichier1 nomlien'`; ce dernier pointe vers un autre fichier et cela est visible par la commande `'ls -l'` qui ramène un fichier logique (et non physique) :

```
lrwxrwxrwx 1 rferrere rferrere 5 2009-10-24 15 :21 fichiers -> fich1
```

La destruction de lien passe également par la commande `'rm nomlien'` ou `'rm nomfichier'`. Et si le fichier source est détruit, alors dans le cas du lien ordinaire il reste toujours une copie du fichier source.



## 9.7 Gestion d'autres périphériques

La gestion des périphériques sous Linux se fait en général par l'arborescence de fichiers et pour cela, l'accès et l'utilisation de systèmes extérieurs (lecteurs de disquettes et CDROM, disque ou clé USB...) se fait par un mécanisme de montage.

Et donc, cela se concrétise par un fichier de périphérique situé souvent dans `'/dev'` (devices) et un noeud d'insertion dans l'arborescence Linux appelé souvent point de montage.

Par ailleurs, il est possible de fixer le montage à chaque démarrage de l'ordinateur et cela se fait par le fichier `'/etc/fstab'` qui est vue comme une table de montage.; contrairement au fichier `'/etc/mstab'` qui est un état dynamique des montages des périphériques.

## 9.8 Documentation en ligne...

Sous Linux, il n'est pas possible de se retrouver perdu en raison d'une documentation bien fournie et large...en interne sur son installation de Ubuntu et sur internet.

C'est ainsi qu'en tapant la commande `'man nomcommande'`, ou la commande `'xman'` (avec interface graphique) qu'il sera possible d'accéder au manuel de la commande ou alors la commande `'nomcommande -help'` donne toutes les options possibles. La commande `'info nom-de-la-commande'` permet également d'obtenir la documentation sur la commande et sur son système.

le manuel des commandes linux <http://doc.ubuntu-fr.org/man>

Sous l'environnement KDE, il existe le centre d'aide de KDE par le menu `'K/AIDE'`.

Sur internet, il existe une multitude de liens d'aide en ligne ou de documents en tout genre à lire ou à télécharger; et la possibilité de s'inscrire à des listes de diffusion et de discuter sur des forums...

Et avec l'outil `'Doxygen'` vous pouvez générer votre propre documentation qui extrait l'information à partir du code source directement. Ce système de documentation pour différents langages (C, C++, Python...) génère la documentation venant directement du code à la condition d'avoir posé des `'tags'` ou des commentaires.

# Chapitre 10

## Outils de base et de programmation

L'environnement de bureau KDE se positionne directement face aux environnements de développement les plus populaires comme la technologie ActiveX de Microsoft.

La technologie 'KParts' permet aux développeurs de créer rapidement des logiciels en faisant usage des dernières technologies. Ainsi, Un module KPart qui est un composant graphique peut exécuter une tâche spécifique ; par exemple, le navigateur Konqueror fait plus d'une fois aux modules Kparts : afficher un fichier .odt, une page HTML, un fichier PDF, une image et faire du copier/coller entre plusieurs fenêtres....

### 10.1 Outils de base

#### 10.1.1 Archivage et sauvegarde

Des outils d'archivage et de compression sont indispensables sur un système Linux si on veut tout simplement envoyer par mail ou par le réseau, un ou plusieurs fichier(s) ; mais cela sert également à faire des sauvegardes. Pour cela, il faut utiliser la commande 'tar' pour assembler plusieurs fichiers en un seul fichier appelé archive. Puis après, il faut utiliser des outils de compression comme 'gzip' ou 'bzip2'.

Par exemple :

```
tar -cvf MesFichiersProjet.tar /Documents/projet/ (pour assembler tous les fichiers du répertoire projet en un seul fichier 'MesFichiersProjet.tar')
```

```
tar -xvf MesFichiersProjet.tar (pour extraire les fichiers de l'archive)
```

Après cela, il faut compresser avec l'outil le plus courant 'gzip' ; par exemple :

```
gzip MesFichiersProjet.tar (compresser le fichier tar afin d'avoir un fichier sous la forme 'MesFichiers-Projet.tar.gz')
```

```
tar -zcvf MesFichiersProjet.tar /Documents/projet/ (pour archiver et compresser en même temps)
```

Par ailleurs, il est possible de compresser également avec 'bzip2' ('bunzip2' pour décompresser) , en mieux et plus lent ; le fichier compressé est de la forme 'MesFichiersProjet.tar.bz2'. Et pour décompresser des fichiers de type '.zip' ou '.rar', il faudra utiliser les programmes 'unzip' et 'unrar'.

La commande 'dump' s'utilise surtout pour les sauvegardes incrémentielles en prenant en compte le système de fichiers et les inodes ; et pour restaurer on utilise la commande 'restore'.

### 10.1.2 Découpage de fichiers

Avant le découpage d'un fichiers, il peut être utile d'afficher le début ou la fin d'un fichier ; pour cela, on utilise les commandes 'head' et 'tail' ; la commande 'head' (tête) récupère simplement les premières lignes d'un fichier ; la commande 'tail' récupère les dernières lignes d'un fichier ; et souvent ces commandes s'utilisent avec le paramètre '-n' suivi d'un chiffre pour indiquer le nombre exact de lignes à récupérer.

Il est possible de découper un fichier avec la commande 'cut' et de récupérer une partie des chaînes de caractères ; par exemple :

la commande 'cut -c 4 notesM1.csv' récupère les chaînes de la première colonne du 1er au 4eme caractère...

Par ailleurs, la commande 'grep' est essentielle car en plus d'être couramment utilisée, son rôle est de rechercher un mot dans un fichier et d'afficher les lignes dans lesquelles ce mot a été trouvé. La commande 'grep' s'utilise de façon simple ou avec des expressions régulières, ce qui peut devenir un outil puissant !

### 10.1.3 Comparaison et analyse de fichiers

La comparaison de fichiers se fait par les commandes 'diff' et 'cmp' ; la commande 'diff' compare deux fichiers cibles ligne par ligne séquentiellement et affiche la différence. La commande 'cmp' quant à elle, indique si 2 fichiers sont identiques et pour cela ne génère aucune sortie, et dans le cas contraire cela indique la différence avec le nombre de caractères et la ligne. par exemple, si on a 2 fichiers, contenu1 et contenu2 :

la commande 'diff contenu1 contenu2' ou la commande 'diff contenu1 contenu2 | more'

la commande 'cmp contenu1 contenu2'

L'analyse de fichiers se fait par la commande 'wc' (word count) qui à priori, pour un fichier donné, compte les mots mais compte également le nombre de lignes et le nombre de caractères. Par exemple, sur le fichier de notes 'notesM1.csv' cela donne 7 lignes :

```
wc -l notesM1.csv
```

### 10.1.4 Mécanisme de tube et de pipe

Sur le système Linux, un tube (ou le pipe) enchaîne deux commandes, le résultat de la première devenant la donnée de la seconde. Il s'agit de connecter la sortie d'une commande à l'entrée d'une autre commande et donc tout ce qui sort de la première commande est envoyé directement à la deuxième commande. Cela se concrétise par le caractère au clavier obtenu par la combinaison de touches : ALT GR + 6 ; exemple d'utilisation : si on veut récupérer les noms du fichier 'notesM1' et les trier par ordre alphabétique...

```
cut -d , -f 1 notesM1.csv | sort
```

### 10.1.5 Mécanisme de recherche et de tri

Deux outils existent pour rechercher un fichier sur le disque dur et à travers l'arborescence. La recherche rapide d'un fichier peut se faire par la commande 'locate nomfichier' (localiser un fichier) ; c'est une lecture dans une base de données qui contient la liste des fichiers et leur position. La recherche par la commande 'find' est plus approfondie et efficace ; cette commande effectue des recherches récursives de fichiers relatifs à des critères tels que la taille, date ...et s'avère être plus efficace car scrute tout le disque dur.

Exemple d'utilisation : rechercher 'où' et 'quoi'...

```
find /home/Documents/ -name "fichier1"
```

Et le tri dans un fichier peut se faire avec la commande 'sort' ; le tri porte par défaut sur un seul champ, à savoir le premier caractère de la ligne.

## 10.2 Editeurs de texte et outils de programmation

Afin de pouvoir programmer et de compiler normalement, il y a un ensemble d'outils qu'il est nécessaire d'avoir avec sa distribution Linux : il y a donc des programmes (et paquets) indispensables à installer comme `make`, `cmake`, `gcc`, `g++`, `autoconf`, `build-essential`. Par ailleurs, il faut installer quelques éditeurs dont la liste n'est pas exhaustive : des éditeurs texte (`nano`, `vi`) et des éditeurs graphiques (`kate`, `gedit`, `gvim` et `Kdevelop`).

### 10.2.1 Des éditeurs simples comme Kate, Kwrite et Gedit

#### L'éditeur Kate...et Gedit

Kate est l'éditeur de texte avancé et orienté développement ; il est complètement intégré à l'environnement KDE car il est possible de le lancer depuis un widget plasma ou avec le programme 'KRunner'. Il intègre la coloration syntaxique, l'émulateur de terminal 'Konsole' et de nombreux plugins. Il est accessible depuis l'interface par le menu 'K/Applications/Utilitaires/Kate'.

En revanche, 'Gedit' est l'éditeur de texte officiel de l'environnement graphique GNOME ; il propose également une interface simple et facile d'utilisation. C'est l'éditeur de texte de la distribution Ubuntu installé par défaut avec toutes ses fonctionnalités : l'édition simple et l'enregistrement d'un fichier texte, support intégral de l'encodage de caractères Unicode, vérification orthographique, indentation automatique... ; et il est accessible par le menu 'K/Applications/Utilitaires/Text Editor'.

#### L'éditeur Kwrite

L'éditeur 'Kwrite' est l'alternative à 'Kate' mais en version allégée ; il est également adapté pour la programmation en C/C++ grâce à la coloration syntaxique et la mise en page automatique. Il est accessible depuis l'interface par le menu 'K/Applications/Utilitaires/Kwrite'.

### 10.2.2 Quelques éditeurs évolués comme Nano et Vi

#### L'éditeur Nano

Si l'éditeur de texte s'appelle 'Nano', c'est parce qu'il est tout petit. Il s'agit d'un programme très simple comparé à 'Vi' et 'Emacs' qui possèdent beaucoup de fonctions. D'ailleurs, en sciences, le terme "nano" représente une toute petite unité. Le nom complet de nano est "GNU nano". Il s'agit d'un éditeur qui s'inspire de "pico", un éditeur de texte plus ancien qui se voulait lui aussi très simple d'utilisation.

Pour cela, l'ensemble de ses commandes commence par la touche CTRL (qui est l'accent circonflexe!). Le site officiel de Nano est :

[lien sur le site de Nano](http://www.nano-editor.org/) `http://www.nano-editor.org/`

La commande 'nano' dans une console (avec éventuellement le nom du fichier à ouvrir et 3 paramètres -m, -A et -i) lance l'éditeur texte. En bas de la fenêtre écran, vous pouvez voir un espace d'aide ou un aide-mémoire pour rappeler à tout moment les commandes principales qu'il est possible de lancer. Et en voici les principaux raccourcis :

Ctrl + G : afficher l'aide

Ctrl + K : couper la ligne de texte (et la mettre dans le presse-papier)

Ctrl + U : coller la ligne de texte que vous venez de couper

Ctrl + C : afficher à quel endroit du fichier votre curseur est positionné (numéro de ligne...)

Ctrl + W : rechercher dans le fichier ; écrire le mot à rechercher et taper sur la touche 'ENTREE' ; appuyer sur 'Ctrl + C' pour sortir de la recherche

Ctrl + O : enregistrer le fichier (écrire)

Ctrl + X : quitter nano

Par ailleurs, Nano a un fichier de configuration qui permet d'indiquer toutes vos préférences; celui-ci s'appelle ".nanorc"; mais si ce fichier n'existe pas, alors il suffit juste de le créer par la commande 'nono .nanorc'; dans ce fichier, il faut écrire une commande par ligne et chaque commande commence par un 'set' (pour activer) ou un 'unset' (pour désactiver) suivi de l'option voulue. Par exemple : 'set mouse' pour la prise en charge de la souris à chaque fois que 'nano' démarre.

## L'éditeur Vi

Cet éditeur a deux modes à son utilisation :

- le mode commande qui consiste à saisir une commande, les caractères tapés sont interprétés comme des commandes d'édition. vi démarre dans ce mode, il faut donc lui indiquer (commande 'i') que l'on veut insérer du texte;

- et le mode saisie qui est la saisie d'un texte dans le buffer, les caractères sont insérés dans le texte édité; pour quitter ce mode il faut appuyer sur la touche "ESC" (ou "Echap" sur certains claviers).

Pour lancer Vi depuis le shell, la commande 'vi [nomdufichier]'; et au démarrage, il est directement en mode commande et pour savoir dans quel mode l'on se trouve, il faut appuyer deux fois sur la touche ECHAP pour être sûr d'être en mode commande. En voici, quelques commandes essentielles :

Les mouvements du curseur (en mode commande seulement) :

'flèches' : Déplace curseur (pas toujours bien configuré).

'ESPACE' : Avance à droite.

'h' : Recule à gauche.

'CTRL-n' : Descend d'une ligne.

'CTRL-p' : Monte d'une ligne.

'CTRL-b' : Monte d'une page.

'CTRL-f' : Descend d'une page.

'nG' : Va à la ligne n (n est un nombre).

Les commandes passant en mode insertion sont :

'i' : Avant le curseur.

'I' : Au début de la ligne.

'A' : A la fin de la ligne.

D'autres commandes :

'r' : Remplace le caractère sous le curseur.

'x' : Supprime un caractère.

'ddollar' : Efface jusqu'à la fin de la ligne.

'dd' : Efface la ligne courante.

'/chaîne' : Cherche la prochaine occurrence de la chaîne.

'?chaîne' : Cherche la précédente occurrence de la chaîne.

Quitter, sauvegarder : (terminer la commande par la touche 'Entrée')

' :w' : Ecrit le fichier.

' :x' ou ' :wq!' : Ecrit le fichier puis quitte vi.

' :q!' : Quitte vi sans sauvegarder les changements.

' !' : commande Exécute commande shell sans quitter l'éditeur.

Vi est l'éditeur de texte en mode écran qu'il faut absolument connaître car il reste accessible quand l'environnement graphique ou la souris ne fonctionnent plus!

l'éditeur vi amélioré <http://doc.ubuntu-fr.org/vim>

### 10.2.3 Un IDE comme Kdevelop

Kdevelop est devenu un très bon environnement de développement sous Linux et ce également pour les plate-formes Windows et Mac OSX. C'est un logiciel complet de programmation avec de nombreux outils de conception d'interface et de génération automatique de code. C'est plus précisément un environnement de développement intégré en français (ou IDE en anglais) pour KDE permettant le développement de projets en divers langages de programmation tels que C, C++, Java, Pascal, PHP, Perl, Python, Ruby, Ada... Il a été développé avec le toolkit graphique Qt, conçu par la société Trolltech.

Il comporte des outils tels que : la complétion de code, browsers de classes ou de fichiers du projet, un debugger, une liaison à un SVG (Système de Gestion de Version) et également à un CMS (Configuration Management System).

Il permet la création de la documentation Doxygen à tout moment du développement d'un projet via le programme 'KDevelop Assistant' (visualiseur de documentation) et constitue donc une aide à la création d'une documentation ou encore une aide au développement d'interfaces utilisateur.

Il intègre les outils de debug comme 'GDB' et 'VALGRIND' qui permettent de faire un 'débugage' mémoire des programmes et donc de traquer les fuites de mémoire et d'autres problèmes de mémoire dans programme en C++.

Le logiciel 'Kate' est l'éditeur par défaut qui peut être changé.

Pour lancer le logiciel Kdevelop, il faut cliquer sur 'K/Applications/Développement/KDevelop4.

Par ailleurs, il est possible d'installer les paquets 'qt4-dev-tools' et 'qt4-designer' afin de concevoir et implémenter des interfaces utilisateur.

[lien sur le site de Kdevelop](http://www.kdevelop.org/) <http://www.kdevelop.org/>

[lien sur le site Wiki de Doxygen](http://fr.wikipedia.org/wiki/Doxygen) <http://fr.wikipedia.org/wiki/Doxygen>

[lien sur le site de Doxygen](http://www.stack.nl/~dimitri/doxygen/) <http://www.stack.nl/~dimitri/doxygen/>

[lien sur le site de Valgrind](http://valgrind.org/) <http://valgrind.org/>

# Chapitre 11

## Communication réseau

Le système d'exploitation linux intègre tout ce qu'il faut pour communiquer avec les autres ordinateurs et serveurs d'un établissement ou de ceux du monde entier. C'est dans les années 1970 qu'un premier réseau à commutation de paquets est né celui de APRANET (Advanced Research Project Agency Network). Puis en 1974 est arrivé le protocole TCP (Transmission Control Protocol) qui rend standard le mode de communication, en 1978 TCP est fragmenté en TCP/IP ( avec Internet Protocol) et dans les années 1980 est arrivé INTERNET et toutes les évolutions et technologies qui vont suivre... Ce n'est qu'après une succession d'évolutions et de collaborations que nos systèmes d'exploitation respectent un mode de fonctionnement. [lien internet sur la pile des protocoles internet TCP/IP](http://fr.wikipedia.org/wiki/Suite_des_protocoles_Internet) [http ://fr.wikipedia.org/wiki/Suite\\_des\\_protocoles\\_Internet](http://fr.wikipedia.org/wiki/Suite_des_protocoles_Internet)

### 11.1 Principe général du fonctionnement réseau

Un ensemble d'équipements (commutateurs et routeurs) reliés entre eux forment un réseau informatique avec des noeuds aux extrémités des connexions.

#### 11.1.1 Ethernet et couches réseau

Un réseau informatique repose sur une infrastructure ou des supports (câbles cuivre, fibre optique) et des équipements physiques reliés entre eux selon une topologie réseau et des protocoles émanant du standard de transmission de données ou une norme internationale qu'est ETHERNET (IEE 802.3) ; c'est également un protocole de réseau à commutation de paquets utilisant une technologie d'écoute et de détection de paquets pour une optimisation de la bande passante ; cela peut être représenté par une simple discussion entre deux personnes ! Ainsi des trames aux formats et valeurs différents traversent le réseau physique local sur lequel sont connectés des ordinateurs. les réseaux locaux Ethernet les plus courants sont appelés : 10BASE-T (fils de cuivre en paires torsadées à 10 Mégabits/seconde), 100BASE-T (fils de cuivre en paires torsadées à 100 Mégabits/seconde), 1000BASE-T (fils de cuivre en paires torsadées à 1 Gigabits/seconde) et 10GBASE-T (fils de cuivre en paires torsadées à 10 Gigabits/seconde).

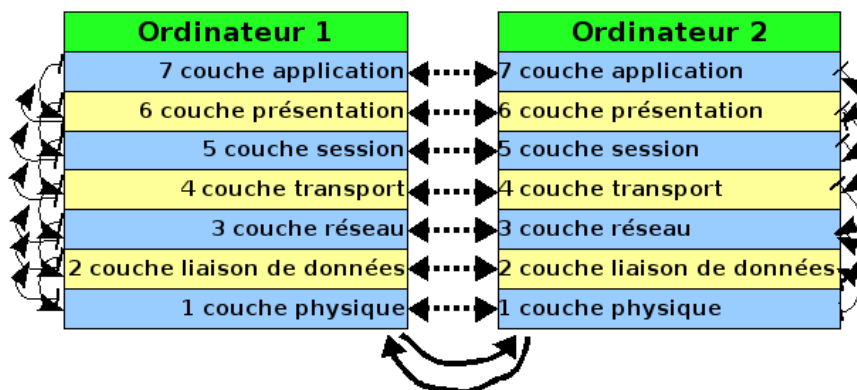
Par ailleurs, le fondement d'un bon réseau est que les systèmes d'exploitation Linux des ordinateurs soient capables de gérer la transmission de données et de fournir aux applications des interfaces standards pour leur permettre d'exploiter les ressources du réseau et ce, même si les ordinateurs et systèmes d'exploitation sont très différents.

C'est le rôle des 'R.F.C.' (Request For Comment), [exemple de RFC sur la couche de communication réseau](#), et des normes que de définir des critères et des procédures ; d'où la norme ou le modèle théorique O.S.I.(Open System Interconnection) établi par l'I.S.O. (International Standard Organisation) qui décrit comment un système d'exploitation réseau doit être construit.

Pour cela, il décrit l'architecture en 7 couches logicielles présentant chacune des interfaces standard pour communiquer entre elles :

- chaque couche est conçue de manière à communiquer avec son homologue, comme si une liaison virtuelle était établie directement entre elles ;

- et chaque couche fournit des services clairement définis à la couche immédiatement supérieure, en s'appuyant sur ceux, plus rudimentaires, de la couche inférieure, lorsque celle-ci existe.



1 - La couche physique = tuyau réseau : fournit des outils de transmission de bits à la couche supérieure, qui les utilisera sans se préoccuper de la nature du média utilisé ; et c'est à ce niveau qu'intervient l'adresse physique de la carte réseau MAC (Medium Access Control).

2 - La couche liaison = contrôle de la transmission des données : fournit des outils de transmission de paquets de bits (trames) à la couche supérieure. Les transmissions sont "garanties" par des mécanismes de contrôle de validité.

3 - La couche réseau = routage des données entre les réseaux : fournit des outils de transmission de paquets de bits (trames) à la couche supérieure ; les transmissions sont routées et la congestion est contrôlée avec correction d'erreurs s'il y a lieu !

4 - La couche transport = superviseur pour optimisation : se doit de fournir à la couche supérieure des outils de transport de données efficaces et fiables.

5 - La couche session = session à 1 ou plusieurs connexions avec synchronisation : fournit donc à la couche supérieure des outils plus souples que ceux de la couche transport pour la communication d'informations, en introduisant la notion de session.

6 - La couche présentation = codage, compression et cryptage des données : c'est le "fourre tout" de la conversion entre représentation interne et externe des données pour assurer la transparence dans le format.

7 - La couche application = services : propose des services comme (FTP, HTTP...) directement utilisable par les applications..



### 11.1.2 TCP/IP

Le modèle O.S.I. est tellement théorique qu'il va à l'encontre de l'efficacité. Il est donc souvent simplifié et cela consiste à regrouper les fonctions de plusieurs couches O.S.I. en une seule.

Par exemple le modèle D.O.D. est utilisé dans le protocole ' TCP/IP ' ("Transfer Control Protocol/Internet Protocol" ou protocole de contrôle de transfert/protocole internet. . ). et le modèle TCP/IP simplifié basé sur 4 couches a été produit de la manière suivante :

Modèle D.O.D.	Modèle O.S.I.
4 couche application	7 couche application
	6 couche présentation
	5 couche session
3 hôte à hôte	4 couche transport
2 couche internet	3 couche réseau
1 couche accès réseau	2 couche liaison de données
	1 couche physique

Le protocole IP est majeur dans la pile TCP/IP car c'est le protocole réseau de niveau 3 (dans le modèle OSI). C'est un système particulier d'adressage car toute adresse IP est constituée de quatre octets lesquels déterminent une adresse de réseau unique ; une adresse en IPv4 (version 4) permet d'identifier toute machine connectée sur un réseau informatique et comporte deux parties dont une partie pour identifier le réseau (net id) et une autre partie pour identifier la machine sur ce réseau (host id). cette adresse représente pour une partie un réseau ou un sous-réseau en particulier, et pour une autre partie un noeud du réseau en question.

De façon plus concrète, une adresse IP comporte une série de 4 octets ordonnés (séparés par des points) qui sont des entiers non signés et ayant comme valeur 0 à 255 (en sachant que 255 est réservé pour une fonction précise). Si l'on a par exemple, 195.57.89.123 (de classe C) et 172.16.1.22 (de classe B), alors l'adresse de classe C comporte 3 premiers octets comme représentant le réseau et le dernier octet comme l'identifiant de l'ordinateur sur ce réseau ; tandis que pour l'adresse de classe B les 2 premiers octets représentent le réseau et le 3eme un sous-réseau et également le 4eme comme l'identifiant de l'ordinateur sur ce sous-réseau.

Puis, c'est le masque du sous-réseau qui permet de savoir exactement la partie de l'adresse qui correspond au réseau et la partie correspondant à l'ordinateur sur ce réseau ; pour cela, cette adresse a la même longueur que celle d'une adresse IP.

Et enfin, c'est une adresse de passerelle (correspondant au réseau) qu'il faudra indiquer afin de permettre le routage des paquets IP et la communication entre les réseaux.

Pour cela, l'organisme comme l'IANA est responsable de la coordination mondiale de l'Internet Protocol systèmes d'adressage ; il est chargé de distribuer des adresses IP. Puis, l'organisme InterNIC, géré par l'ICANN, fournit les informations publiques relatives aux services de domaine et d'enregistrement de nom. De plus, une liste de RFC (Request For Comments ou demande de commentaires) permet de fournir des informations techniques pour la mise en place des réseaux IP.

lien internet sur l'IANA <http://www.iana.org:80/numbers/>

Historiquement le réseau Internet était défini selon des classes d'adresses :

- Adresses de la classe A : le premier octet a une valeur comprise entre 1 et 126 (soit un bit de poids fort égal à 0) et désigne le numéro de réseau et les 3 autres correspondent à l'adresse de l'hôte. Soit 126 réseaux possibles de classe A et 16 millions d'ordinateurs ; utilisé pour les grands réseaux...

- Adresses de la classe B : le premier octet a une valeur comprise entre 128 et 191 (soit 2 bits de poids fort égaux à 10) et les 2 premiers octets désignent le numéro de réseau et les 2 autres correspondent à l'adresse de l'hôte. Soit 16384 réseaux possibles de classe B et 65534 ordinateurs ; utilisé des réseaux entre 256 et 65 536 noeuds...

- Adresses de la classe C : le premier octet a une valeur comprise entre 192 et 223 (soit 3 bits de poids fort égaux à 110) et les 3 premiers octets désignent le numéro de réseau et le dernier correspond à l'adresse de l'hôte. Soit 2 millions de réseaux possibles de classe C et 254 machines ;

L'adressage internet en version 6 (IPv6) a le même rôle que la version 4 (IPv4), mais en revanche, comporte une série de 16 octets ordonnés afin de répondre au développement très rapide de l'internet du nombre de machines connectées dessus ; ce n'est plus une représentation décimale (avec IPv4), mais plutôt sous la forme hexadécimale et où 8 groupes de 16 bits sont séparés par deux-points ; par exemple :

Une adresse en IPv6 comme 'fedc :6482 :cafe :ba05 :a200 :e8ff :fe65 :df9a' signifie que la machine 'a200 :e8ff :fe65 :df9a' est sur le réseau 'fedc :6482 :cafe :ba05'.

### 11.1.3 Connectivité, interfaces et routage

Sur un réseau local chaque ordinateur (ou équipement réseau) possède une interface (carte physique Ethernet avec une adresse MAC sur 6 octets représentée sous forme hexadécimale et les doubles point) qui elle-même comporte une adresse IP fixée par l'administrateur réseau local. Cette adresse IP est soit saisie manuellement ou soit délivrée par un service d'attribution automatique des adresses (D.H.C.P.). Celle-ci doit être unique sur le réseau avertisseur d'un conflit d'adresse et de collisions. Concrètement, chaque ordinateur (ou serveur) est relié par un câble ou cordon (RJ45) à la prise réseau d'un côté et sur un port d'un commutateur de l'autre.

Tout ordinateur étant relié à internet (et donc au réseau) devient donc identifiable par une adresse IP unique (par exemple :197.57.92.121). Mais à chaque adresse IP, il est souvent associé un nom d'hôte ou de machine (hostname) ; la commande 'hostname' donne le nom de l'ordinateur associé à l'adresse IP. Au départ, c'est le fichier local '/etc/hosts' qui permet s'associer un nom d'ordinateur à une adresse IP, mais cela a été canonisé par les services DNS des ordinateurs connectés à l'internet. Il s'agit d'associations entre les IP et les noms d'ordinateur qui sont faites sur des serveurs DNS (ou DNS, système de noms de domaine). Ce qui signifie que chaque fournisseur d'accès internet met en place des serveurs DNS qui fournissent la liste des correspondances des adresses IP/noms d'ordinateurs. La commande 'host NomMachine' ou 'host adresseIP' donne respectivement l'adresse IP associée à l'ordinateur et le nom de l'ordinateur associé à l'adresse IP. Derrière chaque nom de domaine, la règle est de trouver des informations précises (contact, nom de la personne...) sur le fournisseur d'accès internet.

Par exemple :

la commande 'whois NomDNS' donne toutes les informations sur le DNS.

la commande 'host h17-math' donne le résultat 'h17-math.mathst.univ-fcomte.fr has address 172.16.2.127'

la commande 'host 172.16.2.127' donne le résultat '127.2.16.172.in-addr.arpa domain name pointer h17-math.mathst.univ-fcomte.fr.'

La commande `'ifconfig'` liste toutes les interfaces réseau (wifi, filiaire,...) de l'ordinateur et permet de les gérer ces interfaces via le fichier `'/etc/network/interfaces'`. La commande `'iwconfig'` est l'équivalent de `'ifconfig'` mais pour les interfaces réseaux sans fil (souvent `'wlan0'`); ESSI est l'identifiant ou le nom du réseau wifi; la commande `'iwconfig wlan0 essid NomSSIDdelaBorne'` permet d'associer rapidement le nom du réseau de la borne wifi à l'interface `wlan0`. La plupart du temps, la commande `'ifconfig'` porte sur `'eth0'` qui correspond à la connexion par câble réseau (le fameux câble RJ45). Cette commande permet donc de faire des réglages réseau, comme `'ifconfig eth0 down'` et `'ifconfig eth0 up'`.

La commande `'ping'` permet d'envoyer une requête primaire de surveillance sur le réseau vers un autre ordinateur afin de voir si cet ordinateur répond et de voir le temps que met le paquet de données envoyé à revenir. Si toutefois la commande `'ping'` n'arrive pas à résoudre le nom de l'ordinateur en adresse IP, alors peut venir du DNS et il faut déjà vérifier les paramètres du fichiers `'/etc/resolv.conf'`; ou sinon, la commande `'nslookup NomSiteInternet'` afin de voir si le serveur DNS local peut résoudre les noms!

Par exemple : la commande `'ping www.google.com'`...

```
PING www.l.google.com (74.125.77.147) 56(84) bytes of data.  
64 bytes from ew-in-f147.1e100.net (74.125.77.147) : icmp seq=1 ttl=46 time=32.5 ms  
64 bytes from ew-in-f147.1e100.net (74.125.77.147) : icmp seq=2 ttl=46 time=32.6 ms  
64 bytes from ew-in-f147.1e100.net (74.125.77.147) : icmp seq=3 ttl=46 time=31.7 ms  
64 bytes from ew-in-f147.1e100.net (74.125.77.147) : icmp seq=4 ttl=46 time=32.4 ms
```

Comme tout ordinateur comporte tous les éléments nécessaires au fonctionnement réseau, il pourra communiquer avec tous les autres même s'il n'appartient pas au même réseau. Pour cela, c'est le mécanisme de routage (via des routeurs) qui permettra à tout ordinateur de communiquer et de transférer des informations à d'autres ordinateurs et ce entre réseaux différents. Et c'est ainsi qu'une communication s'établira entre les noeuds des différents réseaux de façon directe ou indirecte et ce au niveau de la couche 3 du modèle O.S.I.

Ainsi et de façon très résumée, les tables de routage se mettent à jour au fur et à mesure par des mécanismes de diffusion entre les noeuds des différents réseaux. Et des mécanismes d'association et de résolution d'adresses et différents protocoles permettent cette communication toujours plus rapide et avec des chemins multiples entre les ordinateurs de l'internet.

## 11.2 Principe de connexion, de communication réseau et utilitaires

Tout ordinateur allumé sous Linux (ou tout autre système d'exploitation) possède tout ce qu'il faut pour communiquer avec les autres ordinateurs du même réseau ou avec ceux de l'internet. Un ordinateur est capable de se connecter également sur un autre ordinateur.

### 11.2.1 connexion sécurisée

Un ordinateur qui veut communiquer avec un autre doit le faire de façon sécurisée afin que les données qui transitent soient protégées; pour cela, il faut utiliser le protocole SSH (Secure SHell) avec un peu de cryptographie! Cela est très utile si on veut gérer à distance une machine qui se situe très loin (un serveur ou un ordinateur allumé en permanence) ou transférer des pages Web ou des fichiers sur une autre machine.

L'ordinateur qui se connecte au serveur est un client et utilise donc un programme client de cryptage (comme SSH) avec des clés (privé et publique) pour crypter et decrypter. Le protocole SSH permet un échange sécurisé et crypté, mais il existe un autre moyen encore plus sécurisé pour communiquer entre ordinateurs, c'est le tunnel VPN (Virtual Private Network) avec l'utilisation de certificats.

De façon plus concrète, il faut utiliser la commande 'ssh NomMachine' ou 'ssh AdresseIPMachine' pour accéder à l'ordinateur distant ; ou encore il est possible d'utiliser un programme de transfert de fichiers (de type FTP) basé sur ce protocole sécurisé.

### 11.2.2 transfert sécurisé ou non de fichiers

Le transfert de fichiers consiste à se connecter à distance pour lire, télécharger ou copier des fichiers.

La commande 'wget lienURL' permet de télécharger un fichier depuis le mode terminal en indiquant l'adresse d'un site internet ou d'un site FTP. la commande 'wget -c lienURL' permet de reprendre un téléchargement arrêté...! La commande 'scp NomFichier testcopie@poste-04 :/home/testcopie/images/' permet de copier sur l'ordinateur 'poste-04' le fichier 'NomFichier'. Cette commande 'scp' se base sur le principe de fonctionnement de SSH.

Le protocole FTP (File Transfer Protocol) qui permet d'échanger des fichiers sur le réseau est toujours utilisé mais non sécurisé ; et il est souvent utilisé en mode 'anonyme' ou bien alors en mode 'authentifié' mais non sécurisé au niveau de l'échange. Pour sécuriser l'échange, on a un programme 'sftp' (Secure File Transfer Protocol) qui se base sur SSH pour sécuriser la connexion.

Un autre programme 'rsync' qui est une sorte de 'scp intelligent', car il compare et analyse les différences entre deux répertoires (ou dossiers) puis copie uniquement les changements ; d'où le nom souvent de sauvegarde incrémentielle pour obtenir une sauvegarde entre deux répertoires sur le même ordinateur, ou bien entre deux répertoires sur des ordinateurs différents.

### 11.2.3 connaître son environnement réseau

Chaque ordinateur sous Linux est forcément dans un contexte ou un environnement réseau : il peut être utile de connaître la configuration de ses interfaces réseaux (filaire, wifi) pour en modifier leur configuration, et pour cela il faut connaître quelques commandes de base. Il peut être intéressant par exemple de savoir la liste des ports ouverts, l'adresse IP et l'adresse MAC (adresse physique) de sa carte réseau...

La commande 'netstat' (net comme network et stat comme statistics) permet de faire de surveillance réseau TCP/IP ; elle permet d'afficher la table de routage, l'état des connexions actives et des statistiques sur les interfaces réseau présentes sur l'ordinateur. Cette commande s'utilise avec quelques options intéressantes comme : 'netstat -uta' (donne la liste de toutes connexions ouvertes), 'netstat -i' (donne les statistiques d'utilisation de l'interface), 'netstat -r' (donne la table de routage du noyau linux). Cette commande permet de lister l'ensemble des "sockets" utilisés, ou des processus de communication dans les modes TCP et UDP. La commande 'netstat -s' permet d'obtenir des statistiques résumées par UDP, TCP...

La commande 'traceroute' permet de localiser un peu plus un éventuel goulet d'étranglement réseau jusqu'à une destination ; cet utilitaire (non installé par défaut) envoie sur le réseau des paquets de données en montrant leur emplacement au travers de l'internet via des sauts de routeurs (ou de noeuds) ; et chaque ligne affichée indique un saut de routeur par où le paquet de données est passé ! Par exemple : 'traceroute www.google.com'.... traceroute to www.google.com (74.125.77.106), 30 hops max, 60 byte packets

```
1 172.16.0.1 (172.16.0.1) 0.313 ms 0.293 ms 0.324 ms
2 gw-maths.univ-fcomte.fr (194.57.84.126) 0.842 ms 0.944 ms 0.980 ms
3 172.20.240.234 (172.20.240.234) 0.661 ms 0.668 ms 0.694 ms
4 7206-fw.univ-fcomte.fr (194.57.89.126) 1.159 ms 1.256 ms 1.255 ms
5 3750-gw-ufc.sequanet.fr (194.57.79.194) 2.493 ms 2.494 ms 3.077 ms
```

.....

Très souvent les grands réseaux possèdent des tables de routages dynamiques sur leurs noeuds ; en revanche, sur l'ordinateur local la commande 'route' permet de définir des routes statiques. La commande 'route -n' donne la table de routage avec les adresses IP (au lieu des noms d'ordinateurs) et chaque ordinateur connecté au réseau dispose donc d'une route par défaut pour pouvoir communiquer avec le reste du monde. Et il est donc toujours possible d'ajouter des routes pour d'autres destinations...

## Chapitre 12

# Quelques références bibliographiques et internet...

Mathieu Nebra, Reprenez le contrôle à l'aide de LINUX, Le livre du Zéro, septembre 2010 (ISBN 978-2-9535278-2-7)

Andrew Tanenbaum, Systèmes d'exploitation, Pearson Education, septembre 2008 (ISBN 978-2-7440-7299-4)

Andrew Tanenbaum, David Wetherall, Réseaux, 5eme édition chez Pearson Education, 2011 (ISBN-13 978-2-7440-7521-6)

Michael Kofler, Linux Installation, configuration et applications 8e édition 2008, Pearson, collection Le Campus, octobre 2008 (ISBN 978-2-7440-2286-9)

Jean-François Bouchaudy, Linux Administration, Tome 1, Les bases de l'administration système, Eyrolles, 2007, (ISBN 978-2-212-12037-0)

Gilles Chamillard, UBUNTU Administration d'un système Linux, Eni, collection Ressources Informatiques, janvier 2009 (ISBN 978-2-7460-4684-9)

Yann Le Thieis et G. Chamillard, Debian GNU/Linux Administration du système, Eni, collection Ressources Informatiques, Nov. 2007 (ISBN 978-2-7460-3991-9)

Guy Pujolle, Les réseaux, Eyrolles 2011 (ISBN13 : 978-2-212-12878-9)

Douglas E. Comer, TCP/IP - Architecture, protocoles et applications, 5eme Edition Pearson, janvier 2009 (ISBN13 : 978-2-7440-7380-9)

Raphael Hertzog, Debian GNU/Linux, Editeur Eyrolles, collection Cahiers de l'Admin, avril 2005 2eme édition (ISBN 978-2-212-11639-7)

Revue Linux Pratique Essentiel, Numéro 8, Juin-Juillet 2009

<http://www.laissus.fr/cours/cours.html> <http://www.laissus.fr/cours/cours.html>

[lien internet sur le fonctionnement TCP/IP](http://fr.openclassrooms.com/informatique/cours/apprenez-le-fonctionnement-des-reseaux-tcp-ip) <http://fr.openclassrooms.com/informatique/cours/apprenez-le-fonctionnement-des-reseaux-tcp-ip>

[lien internet sur les commandes fondamentales](http://wiki.linux-france.org/wiki/Les_commandes_fondamentales) [http://wiki.linux-france.org/wiki/Les\\_commandes\\_fondamentales](http://wiki.linux-france.org/wiki/Les_commandes_fondamentales)

[lien sur la documentation officielle du noyau linux](https://www.kernel.org/doc/) <https://www.kernel.org/doc/>

[lien sur un complément d'installation de Ubuntu](http://doc.ubuntu-fr.org/tutoriel/completer_installation_ubuntu) [http://doc.ubuntu-fr.org/tutoriel/completer\\_installation\\_ubuntu](http://doc.ubuntu-fr.org/tutoriel/completer_installation_ubuntu)

[lien internet sur l'histoire de l'informatique](http://histoire.info.online.fr/) <http://histoire.info.online.fr/>

[lien internet cours informatique architecture des ordinateurs de Jean-Michel Richer](http://www.info.univ-angers.fr/~richer/ensl3i_crs2.php) [http://www.info.univ-angers.fr/~richer/ensl3i\\_crs2.php](http://www.info.univ-angers.fr/~richer/ensl3i_crs2.php)

[lien internet sur le cours systèmes d'exploitation des ordinateurs de Yves Epelboin](http://www-int.impmc.upmc.fr/impmc/Enseignement/ye/informatique/systemes/general/index.html) <http://www-int.impmc.upmc.fr/impmc/Enseignement/ye/informatique/systemes/general/index.html>

[lien internet sur Init](http://fr.wikipedia.org/wiki/Init) <http://fr.wikipedia.org/wiki/Init>

[lien internet sur Systemd](http://fr.wikipedia.org/wiki/Systemd) <http://fr.wikipedia.org/wiki/Systemd>

[lien internet sur l'introduction aux systèmes d'exploitation](http://www.commentcamarche.net/systemes/sysintro.php3) <http://www.commentcamarche.net/systemes/sysintro.php3>

[lien internet sur l'adresse IP](http://www.commentcamarche.net/contents/internet/ip.php3) <http://www.commentcamarche.net/contents/internet/ip.php3>

[lien vers GRUB](http://coursinfoev.ovh.org/dokuwiki/doku.php?id=grub) <http://coursinfoev.ovh.org/dokuwiki/doku.php?id=grub>

[lien internet sur cours d'introduction à TCP/IP](http://www.laissus.fr/cours/cours.html) <http://www.laissus.fr/cours/cours.html>

[lien sur le cours réseaux et télécoms de G. Pujolle](http://www.editions-eyrolles.com/Chapitres/9782212124156/Cours-13_Pujolle.pdf) [http://www.editions-eyrolles.com/Chapitres/9782212124156/Cours-13\\_Pujolle.pdf](http://www.editions-eyrolles.com/Chapitres/9782212124156/Cours-13_Pujolle.pdf)

[lien internet sur les distributions live CD](http://www.livecdlist.com/) <http://www.livecdlist.com/>

[lien internet sur la recherche de distribution linux par catégorie](http://distrowatch.com/search.php) <http://distrowatch.com/search.php>

[lien internet sur GNU](http://www.gnu.org/gnu/gnu-history.fr.html) <http://www.gnu.org/gnu/gnu-history.fr.html>

[lien internet sur Doxygen](http://www.stack.nl/~dimitri/doxygen/) <http://www.stack.nl/~dimitri/doxygen/>

[lien internet sur les formats de sortie de doxygen](http://www.stack.nl/~dimitri/doxygen/output.html) <http://www.stack.nl/~dimitri/doxygen/output.html>