

Un nouveau Monde :

Le mouvement du Libre.



Copyright (c) 2006 Patrice ANDREANI.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover

Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

SOMMAIRE.

Introduction

Mon histoire

- *formation, découverte de l'informatique* 6
 - *chômage, le temps libre* 7
 - *la réussite, mon nouveau système* 8
 - *système opérationnel* 9
 - *découverte d'un nouveau monde* 10

L'histoire du libre

- *histoire* 12
- *les grandes dates* 14

Le mouvement du libre

- *l'économie du libre* 16
 - *la philosophie* 17
 - *les licences libres* 18

Annexes

- *licence de ce document* 20
- *l'installation du système* 27
 - *les bases techniques* 28
- *les alternatives logiciels* 58
 - *glossaire* 59
- *bibliographie et liens utiles* 62

Introduction.

J'ai découvert GNU/Linux et le libre peu de temps après m'être mis à l'informatique, il y a quelques années déjà.

Très vite, j'ai été passionné par GNU/Linux, j'ai passé des jours et des jours à me former seul, pour comprendre un peu ce système.

Puis, après la technique, je suis passé à la théorie qu'il y avait derrière. Et j'ai, de nouveau, passé des jours entiers à surfer pour lire des textes sur le mouvement du libre. Je me suis rapidement senti très attiré par lui. J'ai appris à me servir de GNU/Linux, pour être un utilisateur autonome, sachant faire tout ce dont j'avais besoin de faire. J'ai aussi appris la philosophie qui motivait ce mouvement. Et j'ai adhéré complètement. Le libre partage des connaissances, dans le but de l'amélioration de toute la communauté, quoi de plus beau ?

Maintenant, je parle de GNU/Linux autour de moi, sur mon site. J'aide mes proches (ou d'autres) quand ils ont un souci informatique, et j'en profite souvent pour remplacer quelques-uns de leurs logiciels propriétaires par du libre (avec leur consentement bien sûr), j'ai fait une petite documentation sur GNU/Linux, qui est à disposition sur mon site, pour aider les débutants. Je partage mes connaissances.

Et me voilà en train d'écrire ce petit livre, sorte de mini encyclopédie sur le mouvement du libre, pour aider les débutants, et faire connaître un peu mieux GNU/Linux et le libre.

J'ai réellement envie d'aider les autres dans leur apprentissage de l'informatique, en promouvant le libre. Développer le libre pour lutter contre les grands monopoles, et pour améliorer la société.

Voilà le pourquoi de ce livre.

Alors, vous aussi, installez GNU/Linux, rejoignez la communauté du logiciel libre, et, ensemble participons à plus de liberté et d'égalité sur cette planète.

PREMIERE PARTIE :

MON

HISTOIRE

Formation, découverte de l'informatique.

Tout a commencé il y a quelques années. Après avoir obtenu un bac économique, plus par dépit que par conviction (mon niveau en physique et en math ne me permettant pas de passer en scientifique), j'ai arrêté les études. J'ai commencé à travailler, en intérim, faisant tout ce que l'on me proposait.

En fait, je cherchais ma voie. Je n'avais aucune idée de ce qui pourrait vraiment m'intéresser.

Puis, après quelques années passées comme cela, alternant petits boulots et chômage, j'ai décidé de faire une formation (rémunérée), comme l'ANPE me le proposait. Pourquoi, à cette époque, avoir choisi l'informatique (technicien de maintenance), je ne me rappelle pas...

Mon père avait un ordinateur, mais, à part quelques jeux, je n'y touchais pas. Bon, le choix est fait, la formation acceptée, et me voilà donc parti pour une formation d'un an dans un domaine que je ne connais absolument pas, et qui, à dire vrai, ne m'intéresse pas plus que cela.

Est-ce le destin, la providence, le hasard ? Cette formation a changé ma vie. Elle m'a permis de trouver ma voie, d'une part, et m'a fait apercevoir un autre monde.

Dans mon groupe de formation, il y avait de tout : homme, femme, jeune, plus anciens, des nuls (comme moi) et des bons. Autant le formateur et les cours ne m'ont pas appris grand chose, autant le groupe m'a beaucoup appris. Nous travaillions en commun, nous entraïdant mutuellement, partageant nos idées et notre savoir (si maigre fût-il pour certains d'entre nous). J'ai très vite progressé, je me sentais de plus en plus à l'aise avec un PC, commençant à sentir un changement en moi : l'informatique, c'est ma voie. Mais le plus gros bouleversement ne vint qu'après quelques mois de formation. Je m'étais lié d'amitié avec un homme de mon âge, plutôt timide, et qui était d'un meilleur niveau que moi. Un jour, il arriva avec quelques cd gravés, me disant qu'il allait essayer d'installer Mandrake. J'étais encore débutant, et, à part windows, je ne connaissais rien. Il m'a donc expliqué que c'était un système d'exploitation, totalement gratuit, et dont chacun pouvait obtenir, et modifier, le code source. Ce n'est pas allé plus loin, j'ai repris mon boulot, lui le sien, et quelques jours se sont passés. Puis il est revenu me voir, tout content, me montrer son nouveau système, fraîchement installé. Cela ressemblait à du windows, mais la ressemblance, je l'ai vite vue, n'était que visuelle. Il m'a un peu expliqué comment marchait ce système, GNU/Linux donc. Mais il parlait chinois : Shell, Scripts, Kernel, Démons, Root, Init...

J'étais intrigué, intéressé par ce système d'exploitation disponible gratuitement, avec les sources. La formation a suivi son cours (sans un mot sur GNU/Linux, pas encore assez répandu à cette époque), et, de temps en temps, j'allais prendre des

nouvelles du pingouin (la mascotte de ce système, en fait un manchot). Mais la formation me prenait beaucoup de temps, et je n'ai pas pu, malgré un intérêt croissant, approfondir ce sujet.

Puis la formation s'est terminée, nous avons eu nos certificats de technicien informatique, et chacun a repris sa vie.

Pour moi, une longue période de chômage commençait...

Chômage, le temps libre.

Une fois la formation finie, je me suis retrouvé au chômage. J'avais beau répondre à des annonces, faire des candidatures spontanées, rien...

Et j'ai donc eu, par la force des choses, beaucoup de temps libre. Je continuais à penser à GNU/Linux, et, un jour, j'ai lancé mon premier téléchargement d'une distribution. Je ne me rappelle plus laquelle, car, depuis, il y en a eu beaucoup. A peine fini le téléchargement, je grave les CD, puis redémarre le PC pour tenter ma première installation. Totalement à l'aveuglette, je n'y connaissais rien du tout, c'était pour moi l'aventure ! Et mon PC parlait chinois, je ne comprenais pas ce qu'il me disait, les questions qu'il me posait, et je répondais au hasard.

L'installation se finit, il me dit qu'il redémarre, et là, surprise : une invite de commande ! Normalement, j'aurais du avoir une belle interface graphique, et pas ça. Que faire ? Eh bien, puisque celle-là ne marche pas, je vais en essayer une autre. Je ne vais pas garder cela puisque je ne sais absolument pas quoi en faire comme ça.

Et donc, c'est reparti pour un tour, téléchargement, gravure, et nouvelle installation. Et même résultat : ligne de commandes. Cette série a duré longtemps, j'ai essayé beaucoup de distributions, et à chaque fois, pas d'interface graphique. Mais, au moins, je commençais à mieux comprendre le processus d'installation de GNU/Linux. J'ai passé des heures, des journées entières à faire des installation, changer de distribution, en réessayer une autre... Mon matériel ne devait sûrement pas être compatible avec GNU/Linux, mon écran ou peut-être ma carte graphique. J'ai envoyé un mail au constructeur de mon écran, qui m'a dit que si je voulais un driver pour GNU/Linux, je n'avais qu'à l'écrire moi ! Moi qui étais à peine capable de faire une installation... Même si je commençais à mieux me débrouiller. Je passais désormais toutes mes journées dessus (je n'avais que ça à faire), parfois sans même prendre le temps de manger. C'était devenu comme une obsession : installer GNU/Linux (ma quête du Graal en quelque sorte). Je ne savais trop pourquoi je faisais ça, je ne connaissais pas ce système, je ne pouvais donc pas m'en servir, mais il fallait que je le fasse. Cela a duré longtemps comme ça, j'étais seul face au PC. Ce n'était pas très gratifiant, mais j'ai pas mal appris pendant cette période, notamment, qu'il y avait pas mal de documentations sur internet, et des forums. Je me suis inscrits sur plusieurs, et j'ai téléchargé et lu beaucoup de documents. J'avais toujours des réponses rapides sur les forums, il y avait toujours quelqu'un prêt à m'aider.

Puis, un jour, après avoir acheté un magazine avec la Red Hat 7.3 en CD, j'ai fait l'installation, et, enfin, après le redémarrage, j'avais une interface graphique ! J'ai enfin un pingouin dans la machine !

La réussite : mon nouveau système.

Ca y est ! Après quelques mois de tentatives infructueuses, un système GNU/Linux est enfin installé sur mon PC et il marche. Je vais enfin pouvoir commencer à voir à quoi cela ressemble. J'ai donc, petit à petit, découvert ce système. J'ai regardé les programmes installés (il y a vraiment tout, c'est impressionnant), regardé la configuration, bref, j'ai appris. J'ai lu des tas de docs, passé des journées entières sur le net pour chercher des informations sur tel et tel sujet, commandes, programmes. J'étais, malheureusement, toujours au chômage, j'avais donc le temps.

Je me suis formé, à mon rythme, et seul, à GNU/Linux. Et, au fur et à mesure, j'ai aussi pris des notes sur mes recherches.

J'étais aux anges, mon système fonctionnait, et je commençais, un minimum, à savoir m'en servir. J'avais toujours windows sur une autre partition, mais, petit à petit, je m'en servais de moins en moins, trouvant presque à chaque fois une solution sous GNU/Linux. Et, logiquement, ma Red Hat a fini par devenir mon système principal (même si j'étais encore loin d'en maîtriser toutes les possibilités). J'arrivais, à force de recherches, de bidouilles et de tentatives, à faire de plus en plus de choses avec, et je continuais à prendre des notes sur ce que j'apprenais. J'étais, je l'avoue, très fier de moi et de mon passage réussi à GNU/Linux, tout cela seul.

J'ai commencé à prendre de l'assurance face au système, à tenter des modifications de plus en plus délicates, j'essayais de plus en plus de choses, pour voir « comment ça marche ». J'étais très curieux, et je voulais comprendre comment tout cela fonctionnait. Et, à force de tests, j'ai planté mon pingouin ! A force de bidouilles et modifications, le système n'a pas apprécié, et s'est complètement bloqué. Donc, un beau matin : « kernel panic ». Ce message m'a fait sourire tout d'abord, puis, ensuite, j'ai ri jaune... plus moyen de faire quoi que ce soit, et je n'étais pas assez calé pour essayer de le dépanner moi-même évidemment. J'ai donc décidé de formater, et de tester une autre distribution, déjà installée dans une plus ancienne version : une Mandrake 9.1.

C'est parti pour le téléchargement, puis la gravure, l'installation, et redémarrage. Interface graphique ! J'étais, de nouveau, tout content, avec mon nouveau pingouin. Reparti pour un tour, j'ai commencé à arpenter le système, un peu différent de celui d'avant. J'ai appris de nouvelles choses, de nouvelles commandes, et ce GNU/Linux ne se gérait pas comme l'autre, il y avait de nouveaux logiciels pour l'administration.

Je le maîtrisais de plus en plus, windows ne tournait qu'en de très rares occasions maintenant, et j'en étais ravi. Je contribuais, à mon tout petit niveau, à briser son monopole, et je participais au développement de l'utilisation de GNU/Linux.

Système opérationnel.

A part quelques programmes spécifiques, je peux tout faire sous GNU/Linux, avec des logiciels libres. C'est la révolution. Sur le net, j'ai pu me former grâce à toute la documentation, les forums, les sites spécialisés. J'ai appris à maîtriser, à peu près, ce système. J'ai lu énormément de tutoriaux pour approfondir mes connaissances : comment fonctionne le réseau, le kernel et ses modules, le système de fichiers. Je suis, enfin, un utilisateur « opérationnel » du système GNU/Linux.

Finalement, un jour, j'ai fini par trouver du boulot, mais, n'ayant strictement rien à voir avec mes compétences. J'ai accepté, faute de mieux, et puis, ce n'était qu'un remplacement, ça ne durerait sûrement pas bien longtemps. Et j'ai donc eu beaucoup moins de temps libre... mais toujours la même passion pour le pingouin. Je continue à travailler dessus (configuration de Samba, recompilation du noyau...) et à surfer pour me former. C'est une lutte de longue haleine de se former seul, mais la communauté aidant, les choses se passent bien, et l'on apprend assez vite (grâce aux forums et documents, malheureusement trop souvent en anglais).

Et puis, après avoir appris beaucoup de choses sur le côté technique de GNU/Linux et des logiciels libres, j'ai commencé à regarder ce qu'il y avait derrière tout cela, la philosophie du mouvement, ses origines, son fonctionnement.

Découverte d'un nouveau monde.

GNU/Linux est donc mon système principal depuis un moment déjà. Je ne retourne sous windows qu'en de rares occasions, pour des tâches spécifiques. Mais, qu'y a-t-il derrière ce GNU ? Qu'est-ce que cette GPL que je vois un peu partout ? Quelle est la philosophie qui anime ce mouvement des logiciels libres ?

J'ai enfin décidé de me renseigner sur tout cela, et je me suis mis à chercher sur le net des sites traitant du sujet. Et j'ai été agréablement surpris par mes découvertes : le GNU, GNU/Linux, les logiciels libres sont développés et maintenus le plus souvent par des bénévoles ! Des gens qui font cela sans chercher à être rémunérés, parfois sur leur temps libre, pour leur plaisir. Ils passent des heures, des journées à travailler sur des programmes qui, une fois finis, sont offerts gracieusement à la communauté. Peut-on faire plus beau, plus généreux ? Autour de chaque logiciel libre, il y a une communauté, plus ou moins grande suivant la taille du programme, souvent éparpillée sur toute la planète, qui travaille sur un même projet (grâce à internet), en collaboration totale. Une fois fini, chacun peut l'utiliser à sa convenance, et même le modifier puisqu'il est diffusé avec le code source. Donc, le mouvement des logiciels libres est maintenu par toute une communauté, éparpillée sur tous les continents, de développeurs, traducteurs, bêta-testeurs, mainteneurs (sans oublier les utilisateurs), reliés entre eux par le réseau des réseaux. Un des principes forts de ce mouvement, c'est le partage des connaissances pour le bien de toute la communauté. Toute cette communauté se soutient, partage son savoir et le diffuse à quiconque voudrait la rejoindre.

C'est vraiment un autre monde, tout le monde travaille ensemble, partageant ses propres connaissances avec tous, ne travaillant pas pour l'argent mais pour l'amélioration de la communauté toute entière. Tout le monde peut participer, il faut des traducteurs pour la documentation, des testeurs pour trouver les problèmes, des utilisateurs pour essayer d'améliorer le programme, et, bien évidemment, des développeurs de tous langages.

Du fait de la gratuité (ou du prix modique) de tous ces logiciels et de GNU/Linux, cela permet à tous d'accéder à l'outil informatique, il n'y a pas à se ruiner pour payer des licences exorbitantes pour pouvoir utiliser les logiciels windows. Et on lutte comme cela contre la « fracture numérique » entre les classes aisées et les classes plus populaires, et aussi entre les pays développés et ceux en développement. On lutte aussi contre le monopole de windows. C'est vraiment un autre monde, plein d'espoir, pas seulement pour une petite communauté, mais pour le monde entier, face à nos sociétés capitalistes qui ne voient que le profit financier.

Ce mouvement des logiciels libres n'est pas une utopie à construire, c'est un mouvement qui existe bel et bien, depuis des années déjà, et qui continue à se

développer encore et encore ! Les logiciels libres prennent de plus en plus d'ampleur, de nouveaux contributeurs se joignent aux projets existants, ou lancent leurs propres projets. En France et en Europe, certaines administrations passent au libre (avec notamment OpenOffice, une suite bureautique, et FireFox, un navigateur internet), et en Asie, ils ont développé leur propre distribution Linux pour leurs pays.

Il faut poursuivre ce mouvement, l'encourager, contribuer, le diffuser !

DEUXIEME PARTIE :

L'HISTOIRE

DU

LIBRE

Histoire.

C'est en septembre 1983 que Richard M. Stallman (RMS), alors chercheur au MIT (Massachusetts Institute of Technology, grand centre de recherche américain, dans des domaines très divers), lance le projet GNU (Gnu's Not Unix, acronyme récursif) : un système d'exploitation libre basé sur Unix. Il fait cela en réaction à la mentalité d'alors qui change dans le milieu informatique, et qui consiste maintenant à garder son code source secret et faire payer des licences pour ses logiciels.

Au début, ce n'était pas du tout le cas, et les sources des logiciels étaient disponibles gratuitement. RMS s'insurge contre cela, et quitte le MIT.

Il souhaite développer un système d'exploitation basé sur Unix, mais totalement libre (il est l'auteur du fameux éditeur Emacs, du compilateur GCC...). Après avoir publié le Manifeste GNU en 1985, il crée la F.S.F. (Free Software Foundation), pour s'assurer la logistique nécessaire, et avoir un cadre juridique légal pour promouvoir l'éthique, la politique et la philosophie du projet GNU.

Ce qu'il souhaite développer grâce à ce projet est la liberté :

- d'utiliser le programme quel qu'en soit l'usage.
- d'étudier le fonctionnement du programme, et de l'adapter à ses propres besoins.
- de le redistribuer (aider son prochain).
- d'améliorer le programme et de publier ses améliorations, pour que tout le monde puisse en profiter.

Le point fondamental à tout ceci est l'accès au code source.

En 1990 est lancé le projet Hurd, un microkernel Mach qui devra devenir le noyau du système GNU. Mais il n'avance pas assez vite, et le projet s'enlise.

Le 5 octobre 1991, Linus Torvalds, étudiant finlandais de 21 ans, annonce sur le forum usenet la mise à disposition des sources d'un système d'exploitation, tout petit pour le moment. Il l'a créé seul, au départ pour ses propres besoins. Il s'est basé sur Minix, un petit clone libre d'Unix. Il va finalement l'appeler Linux.

Et le noyau Linux s'est naturellement associé au projet GNU, car le noyau officiel n'était pas encore prêt.

Le projet GNU a défini une licence pour garantir toutes ces libertés, la GPL (General Public Licence, dont la première version date de 1989), qui est très utilisée dans le monde du Libre. Il existe un grand nombre de licences dérivées de celle-ci, mais c'est encore aujourd'hui la GPL qui est la plus utilisée.

Il existe aussi la licence BSD, compatible GNU mais tirée du projet BSD (un autre Unix libre), et la FDL (Free Documentation Licence) qui vient du projet GNU, et qui s'applique à la documentation.

Le mot Linux pour désigner une distribution est un abus de langage, car Linux ne désigne en fait que le noyau, le reste étant en grande partie tiré du projet GNU, c'est pourquoi on parle de GNU/Linux. Les composants du système d'exploitation sont :

- le noyau, le coeur du système.
- les composants GNU : le Shell, le compilateur, l'éditeur Emacs...
- l'environnement graphique.
- les logiciels serveurs.

Le noyau Linux devient libre en 1992.

GNU/Linux est donc basé sur Unix, qui lui, date des années 70, créé dans les laboratoires de AT&T. La première version d'Unix date de 1969, il était alors écrit en assembleur (langage machine). En 1973, il est réécrit en C, puis distribué, avec ses sources, dans les universités.

De nombreuses versions d'Unix vont alors voir le jour en quelques années, qui donneront plusieurs branches : la famille des BSD, les GNU/Linux.

GNU/Linux ressemble beaucoup à Unix (beaucoup de leurs commandes sont équivalentes), il en a les avantages : fiabilité, stabilité, performances.

GNU/Linux, est un système d'exploitation Multi-Utilisateurs, chacun ayant ses propres fichiers, applications et environnement de travail, inaccessibles aux autres.

Il est aussi multi-tâches, il gère très bien le réseau (très utilisé en tant que serveur Internet par exemple), et il est plus sécurisé que la plupart des autres systèmes.

La première version officielle de GNU/Linux date de 1992.

GNU/Linux s'étant beaucoup développé, il intéresse aussi les entreprises, et il existe un nombre impressionnant de logiciels libres pour lui.

Lorsque l'on installe un système GNU/Linux, une distribution, on a directement accès à une kyrielle de logiciels qui comblent tous les besoins. Il existe aujourd'hui des centaines de distributions différentes, certaines spécialisée réseaux, serveur, firewall, ou facile à utilisée, stable. De plus c'est un système multi plateforme, qui

tourne sur PC, PowerPc, Mac, mobile...

En 1996 est organisé un concours pour réaliser un logo pour GNU/Linux, et c'est Larry EWING qui gagne en faisant donc Tux, le pingouin (qui est un manchot), avec The Gimp, le meilleur logiciel de retouche d'images sous GPL.

Les grandes dates.

1983 : En septembre, annonce initiale de création du projet GNU par Richard M. STALLMAN (RMS).

1985 : Septembre, publication par RMS du Manifeste GNU.
Octobre, fondation de la Free Software Foundation (FSF).

1989 : La FSF publie la licence GNU GPL version 1.

1990 : Création du projet HURD.

1991 : 5 octobre, post sur le forum usenet de Minix du fameux message de Linus TORVALDS annonçant les débuts de GNU/Linux.
Publication de la GNU GPL version 2.

1994 : Création des distributions Suse et Red Hat.

1996 : Concours pour le logo du noyau Linux, Tux, un manchot, créé par Larry EWING avec The Gimp, un très bon logiciel libre de retouche d'images.
Octobre, création de KDE, une interface graphique pour Linux.

1997 : Création de Gnome, autre principale interface graphique de GNU/Linux, en concurrence de KDE qui n'est pas totalement libre.
Eric S. RAYMOND publie son fameux essai « La cathédrale et le bazar », où il analyse le modèle de développement du libre, comparé au modèle propriétaire.

1998 : Naissance du projet Mozilla, suite à la libération des sources du navigateur Netscape.
Création de la distribution française Mandrake, par Gaël DUVAL.

2000 : SUN libère les sources de Star Office 5.2, sa suite bureautique, qui va devenir Open Office.

TROISIEME PARTIE :

LE

MOUVEMENT

DU

LIBRE

Fonctionnement du libre.

Le monde des logiciels libres fonctionne évidemment différemment de celui des logiciels propriétaires, notamment du fait de la gratuité (souvent mais pas toujours) du logiciel.

Un projet libre est, le plus souvent, décentralisé :

Un développeur commence un projet, seul, puis le publie sur internet. Si d'autres développeurs s'y intéressent, ils rejoignent le projet, qu'ils se trouvent dans la même ville ou sur un autre continent. Une communauté peut ainsi se former, si le projet marche bien.

Ce mode décentralisé est rendu possible par la forte utilisation d'internet, dont les standards (tels HTTP, SMTP, TCP/IP sont issus du libre ou du domaine public). Les contributeurs peuvent travailler ensemble sur un même projet sans jamais se rencontrer, en habitant sur des continents différents. Sur de tels projets, il n'y a pas de chef ni de véritable hiérarchie, et chacun peut participer selon ses compétences.

Les Sociétés de Services en Logiciels Libres (S.S.L.L.) ne vendent pas de logiciels, mais des services autour d'un logiciel ou d'une technologie, comme des formations, de la maintenance, des développements spécifiques, donc, un savoir-faire, un support.

Voici quelques exemples de modèles fondés sur le libre :

- une société peut faire du développement spécifique sur un programme,
- créer un club d'utilisateurs payant,
- trouver des sponsors auprès de grandes entreprises.

Le mouvement du libre a un objectif principal : la qualité. C'est très important, surtout parce qu'il ne fait pas de publicité pour ces produits (pas de budget marketing).

Les qualités principales des logiciels libres sont :

- la réactivité de la communauté (pour trouver, corriger les bugs par exemple),
- la fiabilité,
- l'ouverture du code,
- la sécurité des logiciels créés (puisque l'on « voit » ce qu'il y a dedans),
- la qualité (pour que les développeurs, contributeurs puissent être reconnus pour leur travail par leur pairs),
- la pérennité, du fait de l'utilisation de standards reconnus et ouverts.

Les logiciels libres sont bénéfiques pour tous :

- ils offrent aux pays en voie de développement un moyen de rattraper leur retard technologique, et de réduire la fracture numérique,
- en europe, ils permettent de lutter contre l'hégémonie de l'économie américaine,
- dans les administrations, pour réduire les dépenses et être plus indépendant,
- le libre partage des connaissances va de pair avec l'éducation, le monde de l'enseignement et des scientifiques.

La philosophie.

La philosophie qui soutend le mouvement des logiciels libres est, je trouve, vraiment très bien : le libre partage des connaissances pour l'amélioration globale de la société.

Il existe deux principaux courants :

L'Open Source Initiative (O.S.I.), qui a mis en place 10 critères pour qualifier un logiciel de libre. Elle est plutôt tournée vers le monde de l'entreprise et de la politique.

La Free Software Foundation (F.S.F.), qui n'a que 4 critères, est plus tournée vers le grand public, la recherche, le mouvement social ;

Ces quatre critères si importants, les voici :

- pouvoir utiliser un logiciel pour n'importe quel usage.
- pouvoir l'étudier et le modifier.
- pouvoir en faire des copies.
- pouvoir l'améliorer et diffuser ses améliorations.

Le point principal de tout ceci est l'accès au code source, la recette du logiciel.

Lorsque l'on achète un logiciel propriétaire, notre seul pouvoir est de l'utiliser.

Malgré des différences de point de vue entre la F.S.F. et l'O.S.I., elles ont le même but : promouvoir et développer l'utilisation des logiciels libres.

Les gens qui travaillent sur ces projets le font souvent sur leur temps libre, sans être payés pour cela, il y a une éthique particulière à ce milieu :

- ils travaillent surtout par passion.
- la reconnaissance par leurs pairs est plus importante qu'une forte rémunération.
- ils travaillent pour la communauté, et pas pour les bénéfices d'une société privée.

Il y a une grande coopération entre tous les acteurs du libre, coopération nécessaire à la pérennité du mouvement. Et il faut être curieux, avoir envie d'apprendre, et, surtout, vouloir partager son savoir avec les autres.

Dans ce milieu, les rôles peuvent changer, et un utilisateur peut facilement devenir contributeur, tout le monde peut participer (pour la traduction, les tests, les documentations, et le développement).

Ce monde n'existerait pas sans l'aide de la communauté, et si l'on utilise des logiciels

libres, il faut soutenir ce mouvement, en adhérant à une association, en faisant un don, en achetant des distributions, en aidant des débutants. Les utilisateurs sont très importants ici.

L'utilisation des logiciels libres permet une réelle égalité et une liberté pour tous pour l'accès à l'outil informatique grâce, notamment à leur faible coût (qui dit libre ne dit pas forcément gratuit).

Ce mouvement évolue et conquiert de nouveaux domaines, tels l'art, la culture (Wikipedia), et la communauté s'agrandit, pour le bien de tous.

Les licences libres.

Licence libre : c'est une licence s'appliquant à une oeuvre (livre, programme...) et donnant à son utilisateur certains droits quant à l'utilisation, la modification, la réutilisation de ce document.

Copyleft : ou « gauche d'auteur », permet qu'une oeuvre (programme, livre) publiée sous licence libre, reste libre même si elle est modifiée.

Les termes du copyleft sont contenus dans la GNU (General Public Licence).

La première, et la plus utilisée encore aujourd'hui, des licences libres est la GNU GPL. Elle a été écrite par RMS et Eben Moglen pour garantir les droits des logiciels libres du projet GNU. Cette licence en est à la version 3 (la version 2 date de 1991). Sa plus importante caractéristique est le copyleft. Elle a été écrite à l'origine (en 1989) pour créer une licence libre utilisable par tous, afin que les développeurs puissent facilement partager leurs codes et donc s'entraider et coopérer.

Aujourd'hui, il existe une multitude de licences libres, sortant même du cadre de l'informatique :

- la LGPL : Lesser Public General Licence, plus permissive que la GPL,
- la licence CeCILL : CEA CNRS INRIA Logiciel Libre : la licence libre Française,
- la licence BSD,
- Apache Software Licence,
- MIT,
- Mozilla Public Licence,

Licences autres que pour l'informatique :

- la GFDL : Gnu Free Documentation Licence, licence pour la documentation, non compatible avec la GPL,
- la LAL : Licence Art Libre,
- la CC : Creative Commons, pour les textes, documents.

On le voit, le mouvement du libre touche de plus en plus de domaines, et le droit s'adapte à ces évolutions.

ANNEXES

- licence de ce document
- l'installation du système
- les bases techniques
- les alternatives logiciels
- glossaire

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant

Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and

you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the

- terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
 - H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements",

"Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

L'installation du système.

Tout d'abord, il faut savoir si GNU/Linux sera le seul système sur le disque dur, ou s'il cohabitera avec windows.

Dans ce cas, il vaut mieux faire une défragmentation sur windows, et faire une sauvegarde complète de ses données.

Allez sur le site www.tldp.org, pour vérifier la compatibilité de son matériel et pour prévenir les mauvaises surprises.

Ensuite, vérifier dans le bios que le système peut booter du CD-ROM.

Pour le partitionnement, il faut prévoir environ 4 Gigas pour un système avec X (graphique), plus 256 à 512 Méga pour la swap (le double de votre mémoire vive).

Il est recommandé de faire plusieurs partitions :

Une partition racine (/), pour le système,

Une partition home pour les utilisateurs (possibilité de formater le système sans perdre ses données),

Et on peut en faire d'autres (/usr, /tmp).

Ensuite, il faut choisir ses paquetages, et il y a du choix!

Eviter les paquetages dont vous n'aurez pas de réelle utilité, moins il y en a, mieux c'est pour la sécurité.

Puis, l'emplacement du programme de boot :

S'il n'y a que le pingouin, c'est simple, sur le MBR,

S'il y a windows, à vous de voir, sur le MBR, tout le démarrage sera géré par Linux, ou sur la partition racine de GNU/Linux.

Préférez installer l'amorçage sur le MBR, au cas où, vous pourrez toujours l'enlever (sous Dos, fdisk/mbr).

Une fois l'installation finie, faire une disquette de boot, ça peut toujours servir.

Les bases techniques.

- quelques détails sur le système
 - introduction au Shell
 - le système de fichier
 - les Runlevels
 - les droits des utilisateurs
 - le réseau
 - introduction à samba
 - installation de logiciels
- informations sur le système
 - recompilation du noyau
 - sécuriser le système
 - trucs et astuces
 - les bases de vi
 - les bases d'Emacs
 - les bases de GCC
- rappel des principales commandes

Quelques détails sur le système.

Les partitions pour pouvoir installer le système sont ext3 ou ReiserFs (ce sont les systèmes de fichiers) et une swap (partition d'échange, d'environ 2 fois la taille de la mémoire vive).

Avec ces systèmes de fichiers, il n'y a pas besoin de faire de défragmentation de disque, comme avec d'autres systèmes.

Contrairement à d'autres systèmes d'exploitation, sur celui-ci, tout est fichier; que ce soit les périphériques, disques durs ou lecteurs amovibles, ils sont tous considérés comme des dossiers.

Une distribution de GNU/Linux est fournie avec les sources du Kernel.

Cela permet de compiler son noyau, pour l'adapter à ses propres besoins, pour y ajouter des fonctionnalités ou en retirer certaines qui prennent de la place inutilement. On peut aussi installer un nouveau noyau, pour upgrader son système.

Il existe un grand nombre d'environnements de bureau, mais, les deux principaux sont KDE et Gnome.

Toutes les commandes dont il est question dans ce document sont à effectuer dans une console, souvent avec le compte Root (*su*, puis mot de passe super utilisateur).

Pour avoir l'aide de ces commandes, taper : *man commande*, ou aussi *info commande*. Ces commandes fonctionnent sur une distribution basée sur Mandriva. Certaines commandes varient suivant la distribution utilisée.

Avant toute installation, faire des sauvegardes de ses données.

Ensuite, il faut créer plusieurs partitions :

Une swap (partition d'échange), de 512 Mo maximum, deux fois la Ram.

Une partition racine (*/*), qui contient le système.

Une partition */home*, pour les répertoires des utilisateurs.

On peut en faire d'autres, évidemment.

Il n'y a plus qu'à booter sur le CD ou DVD pour lancer l'installation, si le bios est configuré comme il faut.

Introduction au shell.

Le Shell est un interpréteur de commandes, c'est une interface entre le système d'exploitation et l'utilisateur. Il reçoit les commandes que l'on saisi et les exécute, si on a les droits nécessaires.

Car, lorsque l'on ouvre une session sur le système (qu'on se logue), avec son nom d'utilisateur (login) et son mot de passe, le Shell, avant de se lancer, vérifie si le compte est valide, nos droits et nos permissions sur les fichiers, puis charge notre environnement personnel (chaque processus a lui aussi ses propres droits et permissions).

Certaines variables de cet environnement peuvent être modifiées, adaptées à nos besoins. Il faut taper *stty -a* (qui signifie set teletype, c'est-à-dire configurer le télécriteur, puisque le Shell date des débuts d'Unix et que la plupart des postes étaient alors des terminaux) pour visualiser et modifier ces variables. Sinon, elles sont aussi accessibles par le fichier */etc/profile*.

Il existe aujourd'hui de nombreux Shell, le plus ancien (à l'origine de GNU/Linux) est le Shell sh (Bourne Shell) créé dans les laboratoires de AT&T.

Celui fourni en standard sur les systèmes est le Shell Bash (Bourne Again Shell), mais il en existe d'autres : Zsh, Korn Shell, le Shell C... Ils ont chacun leurs spécificités.

Pour avoir de l'aide, il y a les pages de man, ainsi que la commande *help*.

Pour connaître votre Shell, tapez :

```
echo $SHELL
```

Ce qui commence par \$ indique qu'il s'agit d'une variable, et echo demande de l'afficher.

Une commande Shell se présente ainsi :

```
commande -option(s) paramètres (optionnel)
```

Brève introduction sur la programmation shell.

Le Shell est aussi un véritable langage de programmation, très utile pour automatiser des tâches répétitives ou fastidieuses. Il est très utile pour l'administration système car il permet de faire beaucoup de choses.

On peut se servir d'un simple éditeur de texte pour faire des scripts Shell, tel Kwrite.

Tous les scripts Shell commencent par la ligne `#!/bin/bash`, cela indique que l'on utilise le Shell Bash, ils acceptent des paramètres que l'on peut récupérer dans le script. L'extension du fichier créé sera `sh`, et il faut le rendre exécutable (`chmod a+x prog`) pour pouvoir le lancer simplement avec `./prog`.

Les commentaires se notent `#`, on déclare une variable comme cela : `var=valeur`, et on l'affiche en faisant : `echo $var`.

Il a plusieurs méta caractères, comme par exemple `[!0-9]` qui désigne tous caractères n'étant pas un chiffre.

`cmd1 && cmd2` permet d'exécuter `cmd2` uniquement si `cmd1` c'est terminée correctement, inversement avec `||`.

Les pipes `|` permettent de rediriger une sortie standard vers l'entrée d'une autre commande, c'est très pratique.

Il y a évidemment les `if`, `while`, `case`, `until`...

Voilà, ce n'est qu'un bref aperçu des grandes possibilités qu'offre la programmation shell.

Voici quelques commandes utiles pour les scripts Shell :

Redirections :

- `>` : redirection de la sortie standard ;
- `2>` : redirige seulement les messages d'erreur ;
- `2>&1` : permet de rassembler la sortie standard et la sortie d'erreurs ;
- `<` : redirection en entrée.

Jokers avancés :

- `[a]` : signifie égal à 'a' ;
- `[!a]` : signifie différent de 'a' ;
- `[abc]` : signifie l'un des caractères a, b ou c ;
- `[a-l]` : signifie tous les caractères compris entre a et l.

Les différentes expressions régulières sont :

- `^` : début de ligne ;
- `.` : un caractère quelconque ;

- \$: fin de ligne ;
- x* : zéro ou plus d'occurrences du caractère x ;
- x+ : une ou plus occurrences du caractère x ;
- x? : une occurrence unique du caractère x ;
- [...] : plage de caractères permis ;
- [^...] : plage de caractères interdits ;
- \ {n\} : pour définir le nombre de répétition n du caractère placé devant.

La commande *grep* permet de rechercher une chaîne de caractères dans un fichier.

- v affiche les lignes ne contenant pas la chaîne ;
- c compte le nombre de lignes contenant la chaîne ;
- n chaque ligne contenant la chaîne est numérotée ;
- x ligne correspondant exactement à la chaîne ;
- l affiche le nom des fichiers qui contiennent la chaîne.

Recherche de fichier :

find <répertoire> <critères>

- name : recherche sur le nom du fichier ;
- perm : recherche sur les droits d'accès du fichier ;
- links : recherche sur le nombre de liens du fichier ;
- user : recherche sur le propriétaire du fichier ;
- group : recherche sur le groupe auquel appartient le fichier ;
- type : recherche sur le type (d = répertoire, c = caractère, f = fichier normal) ;
- size : recherche sur la taille du fichier en nombre de blocs (1 bloc=512octets).

Trier un fichier avec sort :

- b : ignore les espaces et les tabulations en début de champ ;
- d : tri sur les caractères alphanumériques (caractères, chiffres et espace) uniquement ;
- r : inverse l'ordre de tri ;
- f : pas de différence entre minuscule et majuscule ;
- tx : Le caractère x est considéré comme séparateur de champ ;
- u : supprime les lignes doublons ;
- n : trie sur des chiffres.

Compter les lignes d'un fichier : *wc*

La commande *wc* permet de compter le nombre de lignes d'un fichier, mais aussi le nombre de mots ou de caractères.

wc -l mon-fichier

Pour avoir le nombre de mots l'option est `-w`, `-c` compte le nombre de caractères. Sans options, elle donne le nombre de lignes, de caractères et de mots.

Edition de champ d'un fichier : *cut*

La commande `cut` permet d'extraire certains champs d'un fichier. Les options sont les suivantes :

- `c` : extrait suivant le nombre de caractères ;
 - `f` : extrait suivant le nombre de champs ;
 - `dx` : Le caractère `x` est le séparateur de champ ;
- `cut -c-10 fichier` : extrait les 10 premiers caractères de chaque ligne ;
`cut -d : -f6 /etc/passwd` : affichera le 6ème champ du fichier `/etc/passwd`, dont le séparateur de champs est le caractère double point.

cat : permet d'afficher un fichier à l'écran, sur la console ;

`cat -n` : affiche les numéros de ligne.

head : affiche, par défaut, les 10 premières lignes d'un fichier.

tail : affiche les 10 dernières lignes d'un fichier.

more : affiche un fichier à l'écran, page par page.

La commande `cp` sert à copier des fichiers :

- `i` : permet de demander confirmation avant écrasement ;
- `u` : ne copie pas si le fichier de destination est plus récent.

`mv` déplace ou renomme des fichiers :

- `f` : efface la cible ;
- `i` : demande confirmation pour écrasement.

Le système de fichiers.

Sous GNU/Linux, tous les fichiers, périphériques, disques durs sont considérés comme des dossiers

L'arborescence.

<i>/bin</i>	Fichiers exécutables des commandes systèmes principales ;
<i>/boot</i>	Noyau et fichiers de démarrage ;
<i>/dev</i>	Fichiers spéciaux, points d'entrées des périphériques ;
<i>/etc</i>	Fichiers de configuration du système ainsi que les principaux scripts de paramétrages ;
<i>/home</i>	Contient les répertoires des utilisateurs, ce dossier peut être une partition à part ;
<i>/lib</i>	Bibliothèques système ;
<i>/mnt</i>	Racine des points de montage (pour lecteur de Cd-rom, disquette, partage Windows) ;
<i>/opt</i>	Installation de packages supplémentaires ;
<i>/root</i>	Répertoire personnel du super utilisateur ;
<i>/sbin</i>	Binaires systèmes pour l'administration ;
<i>/tmp</i>	Fichiers temporaires ;
<i>/usr</i>	Programme accessibles à tous les utilisateurs, c'est une sous-arborescence ;
<i>/var</i>	Données variables ;
<i>/proc</i>	«image» du système.

Même si l'arborescence se présente toujours comme cela, les dossiers (*/tmp*, */home*, */usr*) peuvent être situés sur d'autres partitions.

Nommage des partitions et des périphériques.

Les disques durs sont nommés *hda* pour le premier, *hdb* et ainsi de suite. Les différentes partitions sont ensuite numérotées : *hdb2* serait donc la deuxième partition du deuxième disque dur.

Les lecteurs de disquettes sont nommés *fd0*, *fd1*... (dans le répertoire */dev/...*).

Tous les périphériques (cd-rom, disquettes, partitions...) sont dans le répertoire */dev*, et ils sont en général montés dans le répertoire */mnt* (*/mnt/cdrom*, */mnt/floppy*,

/mnt/windows).

Montage des partitions et des périphériques.

Pour pouvoir travailler sur un cd-rom, une disquette, un dossier partagé, il faut tout d'abord le monter, pour qu'il soit accessible.

Pour cela, on utilise la commande `mount`. Elle a besoin de plusieurs arguments, dont le périphérique à monter, le point de montage (l'endroit où vous souhaitez monter le périphérique), le système de fichier (ext3, vfat, ntfs, smb...), et certaines options.

Inversement, pour démonter une partition, c'est la commande `umount` qu'il faut utiliser.

Par exemple, pour monter un lecteur de cd-rom, il faut taper :

```
mount /mnt/cdrom
```

Pour le démonter :

```
umount /mnt/cdrom
```

Et pour l'éjecter, il faut faire :

```
eject /mnt/cdrom
```

Pour une partition :

```
mount /dev/hda1 /mnt/windows
```

Toutes les partitions et périphériques que l'on peut monter sont configurables grâce au fichier `/etc/fstab`. Il est préférable d'utiliser ce fichier plutôt que de monter à chaque fois telle ou telle partition.

Voici un exemple de fichier `/etc/fstab`, qui monte un lecteur de CD-ROM, un graveur de DVD, un lecteur de disquette, une partition windows, et les partitions GNU/Linux :


```
/dev/hda3 / ext3 defaults 1 1
none /dev/pts devpts mode=0620 0 0
/dev/scd0 /mnt/cdrom auto user,icharset=iso885915,codepage=850,noauto,ro,exec
0 0
/dev/scd1 /mnt/cdrom2 auto
user,icharset=iso885915,codepage=850,noauto,ro,exec 0 0
/dev/fd0 /mnt/floppy auto user,icharset=iso885915,sync,codepage=850,noauto,exec
0 0
/dev/hda1 /mnt/windows ntfs icharset=iso8859-15,ro 0 0
none /proc proc defaults 0 0
/dev/hda2 swap swap defaults 0 0
/dev/scsi/host0/bus0/target1/lun0/cd /mnt/cdrom2 auto ro,noauto,user,exec 0 0
/dev/scsi/host0/bus0/target0/lun0/cd /mnt/cdrom auto ro,noauto,user,exec 0 0
```

Les runlevels.

Sous GNU/Linux, le système peut démarrer sous différents Runlevels (niveau d'exécution) :

- 0** Arrêt ;
- 1** Mode mono-utilisateur (maintenance système, Failsafe) ;
- 2** Mode Multi-Utilisateurs, sans réseau ;
- 3** Mode Multi-Utilisateurs, réseau, sans environnement graphique, donc en console ;
- 4** Non utilisé ;
- 5** Mode Multi-Utilisateurs, réseau et environnement graphique ;
- 6** Redémarrage.

grep :initdefault /etc/inittab

Revoie le Runlevel du système.

Les services sont lancés par des scripts dans /etc/init.d. Les fichiers stockés dans /etc/rc.d, avec des numéros correspondants aux Runlevels (rc1.d à rc6.d), servent aussi à lancer les services, suivant leur Runlevel.

Chaque service démarre à son propre Runlevel, que l'on peut modifier.

Sous une distribution basée sur Mandriva :

/etc/init.d/service start/stop/restart/status

Permet d'agir sur le service voulu.

chkconfig -list

Liste tous les services pour connaître leur niveau d'exécution.

chkconfig -level 0123456 nom_service off / on

Active ou désactive le service voulu.

chkconfig -add service

Ajoute un service au démarrage.

Sous une distribution basée sur Debian :

update-rc.d -f service remove

Pour supprimer un service (-f force).

update-rc.d service defaults

Pour ajouter un service dans les Runlevels par défaut.

Liste non exhaustive des services sous GNU/Linux.

anacron : planifier des tâches ;
apmd: gestion d'énergie pour les ordinateurs portables ;
cupsys: services d'impression ;
devpts: montage des systèmes de fichiers ;
exim: service mail automatique ;
gated: lancement de protocoles de routage ;
gdm: lancement du gestionnaire de session de Gnome ;
hotplug: détection de périphériques branchés à chaud ;
ifupdown : configuration des interfaces réseau ;
kdm: lancement du gestionnaire de session X de KDE ;
keytable: gestion de fichiers pour le clavier ;
makedev: création de fichiers spéciaux pour les périphériques ;
modutils: chargement des modules indiqués dans /etc/modules ;
named: serveur DNS. ;
network: initialisation du réseau ;
klogd: enregistrement des événements systèmes ;
inetd: lancement d'un ensemble de services réseaux ;
gpm: support de la souris en mode texte ;
xntpd: Network time protocol ;
portmap: pour les services rpc, comme NIS ou NFS ;
rstatd , rusersd, rwhod, rwalld: à éviter, fournissent trop d'informations aux utilisateurs distants ;
bootparamd: boot par réseau ;
squid: serveur proxy ;
ssh: accès à une machine à travers une communication chiffrée ;
sysklogd: traçage du système Linux et du noyau ;
dhcpcd: serveur dhcp ;
atd: pour le service at, similaire à cron ;
pcmcia: gestion des cartes pcmcia ;
snmpd: daemon SNMP;
routed: RIP;
squid: serveur proxy ;
sudo: gestion des droits ;
syslog: enregistrement des événements systèmes ;
lpd : services d'impression ;
nfs: serveur NFS ;
amd: daemon AutoMount, pour systèmes de fichiers distants ;
sendmail: pour envoyer/recevoir des mails ;
httpd: serveur web Apache ;

xfst: xfont server ;

crond: planifier des tâches ;

rawdevices: partitions spécifique pour SGBD ;

random: améliore la génération aléatoire de nombres ;

nfslock: pour serveur NFS.

Les droits des utilisateurs.

Les utilisateurs sous GNU/Linux.

L'utilisateur Root (super utilisateur) a tous les droits sur le système, et peut faire tout ce qu'il veut. Il faut donc utiliser ce compte avec parcimonie.

Il faut créer des utilisateurs, qui auront, eux, des droits plus réduits et seront donc moins «dangereux» pour le système.

Chaque utilisateur aura ensuite ses propres fichiers dans son propre répertoire (sans avoir accès aux répertoires des autres utilisateurs), et ne pourra lancer que certains programmes, dépendant du groupe auquel il appartient ou des droits que le Root (l'administrateur du système) lui aura accordés.

Quand un utilisateur se log, le script `/etc/profile` s'exécute, puis recherche le fichier `.bash_profile` ou `.profile`, où l'on stocke les variables d'environnement de cet utilisateur (par exemple, les alias).

La commande `sudo` permet à un utilisateur, autorisé par root, d'obtenir temporairement (par tranche de 5 min) les droits de root. Il faut pour cela configurer le fichier `/etc/sudoers` pour y ajouter les personnes autorisées.

Pour créer un utilisateur, il faut se servir de la commande :

adduser utilisateur

Et ensuite créer son mot de passe (ne jamais laisser un compte sans mot de passe!) :

passwd utilisateur

On peut ensuite regrouper des utilisateurs dans différents groupes, qui auront chacun des droits différents.

Modification des droits sur les fichiers.

Il existe 5 types de fichiers sous GNU/Linux :

Les fichiers ordinaires représentés par un -

Les fichiers spéciaux en mode bloc b

Les fichiers spéciaux en mode caractères c
Les répertoires d
Les liens symboliques l

Pour pouvoir agir sur un fichier, il faut pour cela avoir les droits nécessaires, ou modifier ses droits pour un fichier précis.

Ceci se fait grâce à la commande :

chmod paramètres fichier

Les paramètres peuvent être écrits de manière symbolique (en lettres) ou octal (avec des chiffres).

Les droits d'un fichier se présentent comme ceci :

roman.txt - rwx -x -x

Comme vous le voyez, le premier symbole correspond au type du fichier, ici un fichier standard, et ensuite, les droits de ce fichier, découpé en 3 parties, elles-mêmes découpées en 3 accès différents :

Les 3 premières lettres correspondent aux droits du propriétaire, ensuite le groupe, et les 3 derniers, tous les autres utilisateurs.

Le r signifie des droits de lecture (read), le w donne les droits d'écriture (write), et le x donne les droits d'exécution.

Les paramètres de chmod sont :

A qui donner les droits :

a : tous les utilisateurs ;
g : le groupe du propriétaire ;
o : tous les utilisateurs, sauf ceux du groupe ;
u : uniquement à l'utilisateur.

Les opérateurs pour ajouter ou enlever des permissions :

+: ajoute les droits ;
- : soustrait des droits.

Les droits :

x : pour exécuter ;

r : pour lire ;

w : pour écrire.

Le réseau.

Les cartes réseaux sont nommées eth0, eth1... et Io pour la boucle locale (127.0.0.1).

Les fichiers principaux de configuration du réseau (que l'on peut modifier avec un simple éditeur de texte) sont :

Le fichier /etc/hosts :

Pour affecter l'adresse IP et le nom d'hôte de la machine.

Le fichier /etc/networks :

Celui-ci permet de gérer les sous réseaux, en général, il n'a pas besoin d'être modifié.

La commande *hostname* permet d'afficher votre nom d'hôte.

Quelques commandes utiles pour gérer le réseau :

ifconfig : Cette commande permet de connaître la configuration de sa carte réseau et de la modifier si besoin est ;

Ifconfig IP interface : pour affecter IP à interface.

Par exemple :

ifconfig 192.168.0.1 eth0 affecte l'adresse 192.168.0.1 à la carte réseau eth0.

/sbin/route : Pour afficher ou modifier la table de routage du noyau. On lui indique par exemple la passerelle pour se connecter à internet.

netstat : Affiche la liste des ports en écoute, ainsi que les protocoles et connexions actives.

Introduction à Samba.

Le serveur Samba, qui date de 1987, est un serveur de fichiers, très souvent fourni en standard sur les diverses distributions GNU/Linux.

Il utilise le protocole SMB (Server Message Bloc). C'est un programme qui fonctionne en mode client/serveur, et qui permet de partager des dossiers, répertoires, imprimantes entre des machines en réseau, qu'elles tournent sous GNU/Linux ou windows.

La partie serveur est composée de deux démons : smb, le coeur du serveur, et nmbd qui fournit les services.

Le client est smbclient.

Le fichier de configuration se trouve généralement dans /etc/samba et s'appelle smb.conf.

Il est composé de plusieurs sections, pour la configuration du réseau, de la sécurité, des utilisateurs, des différents partages. Voici un petit aperçu de ce fichier :

```
#=====GlobalSettings=====
```

```
[global]
```

```
# 1. Server Naming Options:
```

```
# workgroup = NT-Domain-Name or Workgroup-Name
```

```
workgroup = linux
```

```
# netbios name is the name you will see in "Network Neighbourhood",
```

```
# but defaults to your hostname
```

```
netbios name = Tux
```

```
# server string is the equivalent of the NT Description field
```

```
server string = Samba Server
```

```
:[public]
```

```
; path = /usr/somewhere/else/public
```

```
; public = yes
```

```
; only guest = yes
```

```
; writable = yes
```

```
; printable = no
```

```
:[fredsprn]
```

```
; comment = Fred's Printer
```

```
; valid users = fred
```

ANDRE Ani

Un nouveau Monde

```
; path = /homes/fred  
; printer = freds_printer  
; public = no  
; writable = no  
; printable = yes
```

Pour lancer/arrêter/voir l'état de samba :

```
/etc/init.d/smb start/stop/status
```

Pour vérifier que le fichier de configuration est correct, faites :

```
testparm /etc/smb.conf hostname
```

Pour créer les utilisateurs samba :

```
smbpasswd -a utilisateur
```

ensuite, il demande le mot de passe.

Pour voir les dossiers partagés par samba :

```
smbclient -L hostname
```

Accéder à un dossier en partage sous windows :

Créer un dossier, dans /mnt par exemple, puis, dans une console :

```
smbmount //host_distant/fichier_à_récupérer /mnt/dossier_dans_mnt
```

Installation de logiciels.

Avec une installation standard de GNU/Linux, de nombreux logiciels sont fournis, et pour tous les besoins, que ce soit la bureautique (OpenOffice), le multimédia (Xmms), Internet (FireFox), des éditeurs de texte, des outils de programmation et bien d'autres encore.

Les programmes sous GNU/Linux se présentent soit en fichier compressé (archive .tar, .gz,...), soit en .rpm (redhat package manager).

Tout d'abord, les rpm :

C'est un format standard sous GNU/Linux. Il suffit tout d'abord de télécharger le programme voulu (un package), puis ensuite de l'installer grâce aux commandes suivantes, que l'on saisit dans une console, après être allé dans le répertoire où est le package :

rpm -ivh programme.rpm
Pour installer un package.

rpm -Uvh programme.rpm
Pour mettre à jour.

rpm -e programme
Pour supprimer un package.

rpm -qa
Lister les packages installés.

Puis le système Debian :

apt-get install package
dpkg -i package
Pour installer un package.

apt-get remove package
Pour supprimer un package.

apt-get update permet de mettre à jour la liste des packages.

apt-get -u upgrade permet de mettre les packages à jour.

Et les archives :

Souvent au format .tar.gz ou .tar.bz2.

Comme les rpm, il faut se placer dans le répertoire où est l'archive, et taper dans une console :

```
tar xvf archive.tar
```

Décompresse une archive .tar.

```
gunzip archive.gz
```

Décompresse une archive .gz.

```
bzip2 -d archive.bz2
```

Décompresse une archive .bz2.

```
tar zxvf archive.tar.gz
```

Décompresse une archive .tar.gz.

```
tar jxvf archive.tar.bz2
```

Décompresse une archive .tar.bz2.

Une fois l'archive décompressée, pour installer le programme, il faut taper les commandes suivantes, dans son répertoire (mais lisez les fichiers readme ou install pour de plus amples informations sur la façon d'installer ou de configurer tels ou tels logiciels) :

```
./configure
```

```
make
```

```
make install
```

Informations sur le système.

Voici quelques commandes pour se renseigner sur son système :

arch

Permet de connaître l'architecture de son système (i386, i686, ppc...).

uname -r

Permet de connaître la version du noyau.

top

Permet de savoir quels sont les processus en cours, la mémoire, l'usage du processeur.

ps -aux

Permet de voir tous les processus qui tournent, ainsi que leur PID (Process ID, numéro d'identification d'un processus actif, le premier lancé sous GNU/Linux est le processus init, le «père» de tous les autres processus), leur utilisateur, la commande qui les lance, leur usage mémoire...

df -h

Donne l'utilisation de la mémoire (libre, occupée) sur les partitions montées.

env | more

Affiche les variables d'environnement.

who

Permet de savoir qui est connecté sur le système.

last

Affiche les dernières personnes connectées.

man programme ou commande

Permet d'afficher les pages de man (format standard sous GNU/Linux pour les pages de manuel) d'une commande ou d'un programme.

lsdev

Affiche le matériel installé.

more /proc/cpuinfo

Donne des informations sur le processeur.

scanpci

Pour voir les cartes PCI du système.

/sbin/fdisk -l

Affiche les partitions.

Recompiler le noyau.

La recompilation du noyau peut se faire lorsque l'on désire installer une version du noyau plus récente, ou lorsque l'on souhaite modifier des modules, ajouter de nouvelles fonctionnalités, ou enlever des pilotes non utilisés.

Pour télécharger un nouveau noyau (au format .tar par exemple), le site incontournable est www.kernel.org.

Décompressez l'archive sous /usr/src, ce qui crée un répertoire linux.

Allez ensuite dans ce répertoire (/usr/src/linux), puis utilisez une des commandes suivantes : *make config* (outil en mode texte), *make menuconfig* (mode texte mais avec des menus), ou encore *make xconfig* (menus en mode graphique).

Il y a beaucoup de menus, avec chacun plusieurs sous-menus, par conséquent, je ne les énumérerai pas ici.

Une fois votre configuration finie, cliquez sur *save and exit*.

Tapez *ls -al* depuis ce même répertoire. Vous pouvez éditer le fichier Makefile de ce répertoire, et modifier la ligne EXTRAVERSION=2.4.21-nouveau_kernel pour donner un nom compréhensible à votre noyau.

Ensuite, taper *make dep clean, make bzImage* modules. La deuxième commande compile le noyau, et, normalement, crée un fichier nommé bzImage dans /usr/src/linux/arch/i686/boot.

Tapez ensuite, depuis /usr/src/linux :

```
cp .arch/i386/boot/bzImage /boot/vmlinuz-2.4.21-nouveau_kernel
```

```
Puis : cp System.map /boot/System.map-2.4.21-nouveau_kernel
```

Ensuite, on installe les modules avec *make modules_install*.

Toujours depuis /usr/src/linux, faites :

```
mkinitrd /boot/initrd-2.4.21-nouveau_kernel 2.4.21-nouveau_kernel
```

Et là, il n'y a plus qu'à indiquer à Lilo ou Grub qu'il y a un nouveau noyau que l'on souhaiterait pouvoir lancer. Il faut donc configurer l'un des deux pour qu'il puisse lancer ce nouveau noyau (n'effacez pas l'entrée de l'ancien noyau, au cas où le nouveau ne démarrerait pas, mais dupliquez-la pour la modifier). Pour Lilo, tapez ensuite : */sbin/lilo* pour vérifier que la nouvelle entrée est ajoutée correctement.

Il n'y a plus qu'à redémarrer pour vérifier que ce noyau fonctionne.

Sécuriser le système.

Voici quelques conseils pour sécuriser un minimum son poste sous GNU/Linux.

Ce n'est qu'un début car évidemment, les problèmes de sécurité sont complexes et il y aurait beaucoup à faire.

> Lors de l'installation, il faut penser aux programmes que l'on utilisera, et n'installer que ceux nécessaires. Moins il y a de programmes installés, moins il y a de risques.

> Après l'installation, vérifier ce qui a été installé (car certains programmes s'installent par défaut). Il faut aller regarder dans les scripts de `/etc/xinetd.d` pour voir quelles sont les applications démarrées suivant votre Runlevel, et désactiver celles présentant des risques, ou inutiles. Il suffit d'éditer le script du service et de modifier `disable`.

> Pour activer ou désactiver un service au démarrage, faire :

```
/sbin/chkconfig -level 0123456 service on/off
```

Il vaut mieux désactiver les services tels que telnet, rwho, finger, httpd, sendmail, netfs, rusersd...

> Pour voir la liste de tous les services et leurs statuts, faire :

```
/sbin/chkconfig --list
```

> Il faut aussi vérifier les fichiers `/etc/passwd` et `/etc/group` pour vérifier qu'il n'y ai pas de compte sans mot de passe.

Il existe un programme (Shadow), qui permet de stocker les mots de passe dans un autre répertoire, et surtout, de les crypter pour qu'ils ne soient accessibles que par le Root. Il est donc conseillé d'installer Shadow, qui ensuite, se lance comme ceci :

```
/usr/sbin/pwconv
```

Pour vérifier que les mots de passe Shadow sont activés, éditez le fichier `/etc/passwd`. S'il y a un `x` après le nom du compte, c'est que Shadow est activé. S'il y a une `*`, c'est que le compte est désactivé.

> Très important aussi, maintenir son système à jour, en téléchargeant régulièrement sur le site de l'éditeur les patchs et correctifs (pour le système, mais aussi pour les programmes installés).

> Evidemment, installer (et bien configurer) un firewall, souvent basé sur

Iptable, tel Shorewall. Et surveiller les ports ouverts, pour vérifier qu'il n'y ai rien d'anormal, avec un scanner tel que Nmap.

> Faire des contrôles de Rootkit réguliers avec chkrootkit. Il permet de vérifier la présence de Rootkits, des programmes malveillants utilisés par les crackers pour corrompre le système.

> Eviter d'utiliser (et d'installer) les outils en «r» : rlogin, ruser, rsh... car leur authentification est assez faible.

> Utiliser des mots de passe de 6 ou 8 caractères, en utilisant des minuscules, majuscules, symboles et chiffres. Ne pas prendre de mots du dictionnaire.

> Utiliser un programme de vérification d'intégrité du système, tel Tripwire, pour vérifier qu'il n'y ait pas eu de modification du système par un intrus.

- > Penser à contrôler régulièrement les fichiers de log (généralement dans /var/log) pour être au courant de tout ce qui se passe sur son système.
- > On peut aussi installer un IDS (détection d'intrusion système) afin de pouvoir contrecarrer les tentatives d'intrusion.
- > Eviter de se servir du compte root, et utiliser son compte utilisateur le plus possible.
- > Faire des sauvegardes régulières de ses données, sur CD ou DVD.
- > Et enfin, il faut se tenir au courant des évolutions matérielles et logicielles, des nouvelles failles et virus et des moyens de s'en protéger.

Gérer son poste GNU/Linux.

Les tâches d'administration d'un poste font partie des tâches de sécurisation.

Il faut bien gérer les utilisateurs, leurs différents groupes, les droits que chacun a sur les programmes, et aussi, les former sur les problèmes de sécurité.

Penser à réaliser régulièrement des sauvegardes de ses documents, des fichiers de configuration...

Il existe deux outils principaux, souvent fournis en standard, pour gérer et configurer son poste :

Webmin : c'est une interface web qui permet de pratiquement tout configurer ;
LinuxConf : outil pratique pour configurer bon nombre de choses sur un poste.

Trucs et astuces.

Voici divers petits trucs à connaître :

> Pour faire un raccourci sur le bureau pour un programme ou une commande (comme par exemple /sbin/reboot) :

Clic droit sur le bureau,

«créer un nouveau»,

«lien vers une application»,

puis remplissez les cases utiles (chemin de l'exécutable, droits pour l'exécution de l'application).

> Pour avoir le lecteur CD-ROM, ou une disquette, une partition en partage sur le bureau :

Clic droit sur le bureau, puis :

«créer un nouveau»,

«CD-ROM» ou «disquette» ou «disque dur».

> Pour modifier le PATH quand on installe java, il faut ajouter ces lignes au fichier /etc/bash.bashrc (ce qui modifie pour tout le système) :

```
export JAVA_HOME="/usr/local/j2re1.4.2_10"
```

```
export PATH="${JAVA_HOME}/bin:${PATH}"
```

Et pour avoir le plugin sous FireFox, allez dans son répertoire plugin et faites un lien avec votre jdk :

```
ln -s /usr/java/jdk1.5.0_03/jre/plugin/i386/ns7/libjavaplugin_oji.so
```

> Le pavé numérique ne s'allume pas toujours par défaut au démarrage, alors pour y remédier, il faut tout d'abord vérifier si vous avez le package numlockx. Si oui, faite «numlockx on» dans une console, sinon, il vous faudra le télécharger puis l'installer. Ensuite, pour qu'il se lance au démarrage de la session, ajouter ces lignes au fichier /etc/X11/gdm/Init/Default :

```
if [ -x /usr/X11R6/bin/numlockx ]; then
/usr/X11R6/bin/numlockx on
fi
```

> Commandes pour gérer les services :

ksysv ou *chkconfig*.

> Pour démarrer en mode mono-utilisateur, utile pour faire du dépannage :

linux single au moment du boot.

> Pour rechercher un fichier, utiliser la commande «find» :

```
find / -name fichier_à_chercher -print
```

> Pour agir sur un service :

```
/etc/rc.d/init.d/service start/stop/restart
```

> Créer un alias pour une commande, si vous l'utilisez souvent, où qu'elle est longue à taper et que l'on a du mal à s'en rappeler :

Par exemple, pour monter le CD-ROM, la commande est :

```
mount /mnt/cdrom
```

On peut remplacer ceci simplement par `cdrom` ou autre chose, en ajoutant dans le fichier `/home/bashrc` (mais cela dépend de votre Shell) :

```
alias cdrom="mount /mnt/cdrom"
```

> Pour rediriger le résultat d'une commande ailleurs que sur la sortie standard (stdout, l'écran, stdin est l'entrée standard, le clavier, et stderr la sortie d'erreur), il suffit d'ajouter > et un nom de fichier par exemple :

```
ps aux > test.txt
```

> Si vous avez besoin d'aspirer un site, le votre par exemple pour le sauvegarder, utilisez :

```
wget -m adresse_du_site
```

> Pour manipuler les modules :

Tout d'abord, les modules, ce sont, en général, des drivers qui sont extérieurs au noyau (pour gagner en place et en «modularité»). Ils peuvent se charger en mémoire soit manuellement, soit automatiquement. Ils se situent souvent dans `/lib/modules`.

`lsmod` : affiche les modules chargés à cet instant ;

`insmod module` : charge le module ;

`rmmod module` : décharge le module ;

`modprob module` : charge le module, ainsi que les modules dont il dépend (les dépendances).

> Pour tuer un processus récalcitrant, il faut utiliser la commande `kill`, et, parfois, `kill -9`, ce signal ne pouvant être intercepté.

> Pour configurer le serveur X, on peut se servir de `xf86cfg` ou `xf86config`. Le fichier de configuration s'appelle `XF86Config-4` ou `xorg.conf` et les deux se situent dans le répertoire `/etc/X11`.

> Pour créer une disquette de démarrage avec Lilo, tapez :
/sbin/mkbootdisk numéro noyau
numéro noyau est récupéré grâce à la commande *uname -r*.

> Pour mettre à jour les bibliothèques dynamiques, après des mises à jour par exemple, faites */sbin/ldconfig*. Cette commande lit le fichier */etc/ld.so.conf*, où sont indiqués les chemins des diverses bibliothèques utilisées. Il y a aussi la commande *bash*.

> Pour passer en mode console, en cas de problème, taper ctrl + alt + f1 (ou f2, f3...).

Pour revenir en mode graphique, il suffit de faire ctrl + alt + f7.

Configuration de Lilo.

Lilo (Linux Loader), est un programme permettant de démarrer GNU/Linux, il en existe d'autres, notamment Grub (Grand Unified Bootloader) qui est aussi très utilisé (et plus puissant).

Il peut se configurer grâce au fichier /etc/lilo.conf, dont voici un exemple :

```
boot=/dev/hda
map=/boot/map
vga=normal
default="TuX"
keytable=/boot/fr-latin1.klt
prompt
nowarn
timeout=150
message=/boot/message
menu-scheme=wb:bw:wb:bw
image=/boot/vmlinuz-2.4.22mdk
    label="TuX"
    root=/dev/hda2
    initrd=/boot/initrd-2.4.22mdk.img
    append="quiet devfs=mount hdc=ide-scsi hdd=ide-scsi acpi=off"
    vga=788
    read-only
other=/dev/hda1
    label="win"
    table=/dev/hda
```

Ce fichier montre qu'il n'y a qu'un disque dur (hda), le système lancé par défaut est Tux, et il y a aussi un système windows sur la première partition du disque (hda1).

Le système est en français (keytable), et il attend 15 secondes (timeout), puis, si aucune touche n'est pressée, il lance GNU/Linux.

Append permet de passer des paramètres au noyau lors du démarrage, ici, par exemple, il faut émuler des graveurs SCSI.

Les bases de l'éditeur Vi.

Vi est un éditeur datant d'Unix, et il est encore présent sur toutes les plateformes GNU/Linux. Il est donc nécessaire d'en connaître les bases, car parfois, il est le seul éditeur disponible.

C'est un éditeur plein écran, il n'y a donc pas d'interface graphique, ni même de menu. Mais il est interactif, car il communique avec l'utilisateur, grâce à la ligne d'état, qui affiche des messages.

Vi possède deux modes différents :

> Le premier, celui qui est actif par défaut à son lancement, est le mode commande. Toutes frappes de l'utilisateur sont interprétées comme des commandes.

> Le mode insertion, lui, interprète chaque frappe comme du texte à ajouter dans le fichier ouvert.

Pour passer du mode commande au mode insertion, il faut taper *a* (pour ajout) ou *i* (pour insertion). Inversement, pour quitter le mode insertion, taper *esc*.

Vi peut être configuré pour s'adapter à l'utilisateur, ceci se fait par la commande *set*.

Lorsque l'on démarre Vi, en tapant simplement *vi* dans une console, ou *vi fichier*, les lignes vides apparaissent avec un *~* (tilde) en début.

Maintenant, voici quelques commandes de base :

Pour se déplacer dans un fichier, outre les flèches, on peut utiliser *h*, *j*, *k*, *l* ;

Pour faire avancer l'écran, *ctrl B* ;

:help : permet d'obtenir de l'aide ;

A : ajoute du texte en fin de ligne ;

R : remplace du texte à partir du curseur ;

R : remplace un caractère ;

O : ouvre une ligne sous la ligne courante pour ajout de texte ;

U : permet d'annuler une action ;

Maj h : va à la première ligne ;

Maj l : va à la dernière ligne ;

/mot : cherche mot dans le fichier ;

:n fichier : ouvre fichier ;

:q! : quitte sans sauvegarder ;

:w : enregistre ;

:wq : enregistre et quitte ;

i : insert du texte avant le curseur ;
o : insert une ligne après la ligne courante ;
x : supprime le caractère courant ;
:set number : affiche les numéros de ligne.

Les bases d'Emacs.

Emacs est un éditeur de texte extrêmement puissant. Il a été écrit par R.M.S (Richard Stallman, fondateur de la Free Software Foundation) entre 1984 et 1985, lorsqu'il travaillait encore au M.I.T. Le langage utilisé est principalement le Elisp (Emacs Lisp, spécialement conçu pour pouvoir implémenter facilement Emacs, et donc lui ajouter des fonctionnalités), ainsi que le C.

Pour lancer Emacs, tapez simplement son nom dans une console. Celui-ci est le plus répandu des éditeurs de texte sous GNU/Linux. Pour se servir d'Emacs, on se sert de combinaisons de touches, par exemple, Ctrl-a, pour aller en début de ligne.

Il existe différents modes pour travailler, des modes majeurs, et dans ces modes, d'autres modes encore, dits mineurs. Par défaut, Emacs essaye d'ouvrir un fichier dans le bon mode majeur : un fichier .c sera ouvert dans le mode de programmation C.

Il y a un mode pour les mails, un pour telnet, pour java, lisp évidemment, et bien d'autres encore. Il y a même un mode Shell.

Dans chacun de ces modes majeurs, on peut ajouter des modes mineurs : le saut de ligne automatique, la gestion des versions, inclure des abréviations...

On peut paramétrer les combinaisons de touches comme ceci :

(define-key keymap «séquence de touches» 'nom de commande)

(global-set-key «séquence de touches» 'nom de commande)

(local-set-key «séquence de touches» 'nom de commande)

Et pour modifier le contenu d'une variable, on utilise la commande : *setq nom_de_variable nouvelle_valeur*

Pour ouvrir un fichier avec Emacs, tapez *emacs fichier*.

Voici maintenant quelques commandes de bases (le C signifie Ctrl, et le M Alt) :

C-p	Monter d'une ligne ;
C-n	Descendre d'une ligne ;
C-a	Début de ligne ;
C-e	Fin de ligne ;
C-v	Descendre d'une page ;
M-v	Monter d'une page ;
C-x C-s	Sauvegarde le Buffer courant sur le disque ;
C-x u	Annule la commande précédente ;
C-c C-f	Ouvre un fichier ;
C-x C-c	Quitte Emacs ;
C-d M-x	Efface le caractère sur lequel est le curseur ;

M-d M-x Efface le mot à partir du curseur ;
M-backspace M-x backward Efface le mot précédent ;
C-k M-x Efface la ligne à partir du curseur.

Il existe beaucoup d'autres commandes, car on peut vraiment tout faire avec Emacs, c'est vraiment un environnement de travail complet.

Les bases de GCC.

GCC est le compilateur standard du projet GNU, il a été écrit en grande partie par R.M.S. C'est un logiciel libre, publié sous licence GNU GPL, et fourni sur quasiment tous les systèmes d'exploitation libre (dont GNU/Linux évidemment).

GCC signifiait, au départ, GNU C Compiler, mais au fur et à mesure de son évolution, il compile de plus en plus de langages (C++, objective C, fortran, ada, java), donc maintenant cet acronyme veut dire GNU Compiler Collection.

Il s'utilise simplement en ligne de commandes, en tapant :

```
gcc prog.c
```

Faite *gcc -help* pour voir l'aide.

Pour avoir tous les messages d'erreur d'un programme que vous compilez, faites :

```
gcc -Wall prog.c prog
```

Pour nommer le fichier de sortie, qui par défaut s'appelle a.out, utilisez l'option *-o*

Pour pouvoir utiliser GDB, le débogueur standard du projet GNU, écrit par R.M.S. en 1988, publié lui aussi en GPL, il faut utiliser l'option *-g*

On compile un programme en C :

```
gcc -Wall -g -o prog.c prog
```

Une erreur s'affiche lors de cette compilation, un fichier Core est donc créé, dans le répertoire courant, qui contient tous les paramètres du programme lors de son plantage.

C'est grâce à ce fichier que GDB va pouvoir nous indiquer ce qui a provoqué cette erreur.

On tape alors :

```
gdb core prog
```

GDB se lance, on tape la commande *where*, pour qu'il nous indique où se situe l'erreur, puis, une fois cela fait, on tape *quit*, pour revenir au Shell.

Rappel des principales commandes.

cd.. : remonte d'un niveau.

ls -l : liste les attributs de fichiers.

ls -a : liste aussi les fichiers cachés.

ls -t : liste par date.

cp origine destination : permet de copier un fichier.

cp -i : demande pour éviter l'écrasement de fichiers.

cp -r : copie récursivement le répertoire et ses sous-répertoires.

rm -d : supprime un répertoire, vide ou non.

rm -r : supprime un répertoire et ses sous-répertoires.

mkdir : crée un répertoire.

mv : déplace un fichier.

cat : affiche un fichier.

cat -n : permet de numéroter les lignes.

file : permet de connaître le type de fichier.

at options time : permet de lancer une commande à un moment précis.

at -c cmd : affiche cmd.

at -d cmd : efface cmd.

crontab : exécution d'une commande à intervalle régulier.

fdformat : formate une disquette.

free : affiche la mémoire libre.

mkbootdisk : crée une disquette de démarrage.

mount : pour monter un système de fichier (CD-ROM, partition...).

netstat : informations sur la configuration du réseau.

ps : affiche la liste des processus.

tar : archivage de fichiers.

tar -z : décompacte une archive.

tar -t : affiche les fichiers de l'archive.

tar -c : crée une archive.

uname : informations système.

uname -a : affiche toutes les informations possibles.

uname -i : platform.

uname -n : hostname.

uname -p : processeur.

uname -r : release du Kernel.

who : affiche les utilisateurs courants.

dd options : copie suivant certaines options un fichier sur la sortie spécifiée ou la sortie standard.

dd conv=flags : permet de convertir la sortie.

dd if=file : lit depuis l'entrée spécifiée.

su options users args : permet de lancer un Shell pour un autre utilisateur, ou le Root par défaut.

sudo options cmd : permet de lancer une commande, si l'on est autorisé, avec les droits du Root.

sudo -b : commande en arrière-plan.

sudo -k : enlève la permission.

route options cmd : affiche la table de routage du noyau.

route add adress : ajoute une route.

route del adress : enlève une route.

top : affiche sur la console les processus actif en temps réel.

passwd user : change le mot de passe de user.

chkconfig : pour gérer les services et le niveau de démarrage.

chmod : modifier les droits des fichiers.

hostname : affiche le nom d'hôte, ou le définit.

lpr : inscrit un fichier dans la file d'attente d'impression.

lpc : contrôle les tâches d'impression.

Les alternatives logicielles.

Voici les principaux logiciels sous GNU/Linux, mais la liste est loin d'être exhaustive.

<i>Catégories</i>	<i>Libre</i>
Suite bureautique	OpenOffice
Gravure	K3B
Mail / Nnewsgroups	ThunderBird
Navigateur	FireFox
Messagerie instantanée	Amsn, Gaim, WengoPhone
Lecteur audio	Xmms
Lecteur vidéo	VLC
Extraction audio	Grip, Audacity
Edition vidéo	KDEnLive
Firewall	FireStarter
HTML	Nvu, Writer
Graphisme	The Gimp
3D	Blender
Développement	Eclipse, GCC, Emacs
Base de données	MySQL, Base
Visualiseur d'image	gThumb
PDF	pdftk, OpenOffice
Compression	Gzip
Traitement de texte	Writer, Emacs
Partitionnement	QTParted, GParted
Serveur	Apache, Samba

Glossaire.

Bit :

C'est l'unité de base en informatique, qui possède deux états : 0 ou 1.

Binaire :

Programme directement exécutable par un ordinateur.

Bug ou Bogue :

Erreur de programmation.

Compiler :

C'est le fait de transformer un code source en binaire, directement exécutable et compréhensible par l'ordinateur. Ceci se fait grâce à un compilateur.

Console :

C'est un Shell, une fenêtre avec un prompt, dans laquelle vous tapez les commandes. Le Shell (il en existe plusieurs : bash, tcsh, ksh, ash..., le plus utilisé étant le Shell Bash) est le programme qui se charge d'exécuter vos commandes. Sous GNU/Linux, on parle aussi de console virtuelle, au nombre de 12.

Démons :

Ce sont des programmes chargés au démarrage.

Hardware :

Désigne tout le matériel.

Hoax :

C'est un canular, par exemple un faux virus.

Logiciel libre :

Un logiciel libre est distribué avec son code source, que l'on peut étudier, améliorer, modifier ou redistribuer selon les termes de la licence Gnu Public Licence (GPL), ou une licence de type BSD. C'est le principe de développement de GNU/Linux et d'une grande majorité des applications livrées avec celui-ci.

Lien symbolique :

C'est un concept d'Unix. Cela permet à un fichier d'apparaître à plusieurs endroits dans l'arborescence, sans qu'il ait été copié.

Mémoire vive :

C'est la RAM (Random Access Memory), les barrettes que l'on branche sur la carte mère.

Noyau ou kernel :

C'est le programme de base d'un système d'exploitation (et donc de GNU/Linux), qui gère l'ordinateur, les périphériques, le réseau... A la base, Linux n'est que le nom du noyau, les autres programmes venant du projet GNU (GNU is Not Unix). Mais tout le monde parle de Linux, alors que l'on devrait dire GNU/Linux.

Octet :

Groupe de 8 bits.

Partition :

Lorsqu'on doit utiliser un disque dur, il faut d'abord le structurer en une ou plusieurs sous-parties, afin d'y installer un système d'exploitation. On crée donc des partitions (logiques, primaires).

Point de montage :

C'est le répertoire particulier (cela vient d'Unix) auquel est attaché un système de fichier, qu'il se trouve sur une disquette, un cd-rom ou un disque dur.

PID :

C'est le Process ID, le numéro d'identification alloué par le noyau à chaque processus (puisque GNU/Linux est un système multi-tâches), et qui lui permet de le suivre et de communiquer avec lui. Chaque PID est unique.

Pixel :

Point élémentaire d'une image.

Processus :

Chaque programme exécuté par le processeur. Il existe trois types principaux de processus : les processus interactifs, les traitements par lots, et les démons. Chaque processus possède son propre identifiant (le P.I.D).

Reboot :

Redémarrer l'ordinateur.

Reset :

Redémarrage « sauvage » du PC.

Runlevels :

Ce sont les différents niveaux d'exécution des programmes au démarrage.

Sauvegarder :

Fait d'enregistrer ses données sur une mémoire de masse (CD-ROM, DVD, lecteur ZIP...).

Script :

Un script est un programme écrit dans un langage interprété. Contrairement aux programmes écrits en langage compilé, les scripts sont traduits en langage machine au fur et à mesure de leur exécution.

Software :

Désigne les logiciels.

Spam :

E-mails non sollicités.

Swap :

Partition d'échange, c'est la mémoire virtuelle de GNU/Linux, qu'il utilise lorsque la mémoire vive est pleine.

Système d'exploitation, Operating System (OS) :

C'est l'interface entre le matériel et les logiciels.

Système de fichier :

C'est la manière dont sont structurées les données sur une partition, un cd-rom, une disquette. Pour GNU/Linux, on utilise généralement le type ext3 et reiserfs (qui sont «journalisées», c'est à dire qu'elle gardent sur le disque un «journal» des opérations à effectuer, pour, ensuite, les effectuer, ce qui rend ce système plus fiable.

Copyleft :

Gauche d'auteur, si un logiciel est dérivé d'un logiciel libre, il doit forcément rester libre lui aussi.

Bibliographie et liens utiles.

Documentation et news sur GNU/Linux et l'open source :

fr.tldp.org

www.linux-center.org (il y a aussi des programmes).

slashdot.org (en anglais, news sur l'informatique, la programmation et le libre).

linuxfr.org (très bon site, plein d'infos)

www.lea-linux.org (très bon site pour les Linuxiens débutants, il y a tout : progs, tutoriaux, astuces...).

Philosophie GNU/Linux et Open Source :

www.gnu.org (site incontournable sur le mouvement GNU).

www.france.fsfeurope.org (site incontournable, la Free Software Foundation).

www.stallman.org (site incontournable, le fondateur du mouvement libre, du projet GNU : Richard Stallman (R.M.S.)).

www.aful.org

www.gnu.org/licenses/gpl.html (la licence GPL).

www.gnu.org/software/hurd/hurd.html (site du noyau officiel du projet GNU, le Hurd).

www.funoc.be/etic/doss003/art001.html

<http://www.april.org/>

www.framasoft.net/article4127.html

www.aldil.org/projets/fiches_libres/

www.livretdulibre.org/

www.aful.org/presentations/libre.html

www.logiciellibre.net/

Les licences libres :

www.gnu.org/philosophy/license-list.fr.html

fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU

www.aful.org/presentations/licences/index.html

fr.wikipedia.org/wiki/Licence_libre

Distributions et programmes :

www.linuxiso.org (les distributions GNU/Linux)
www.framasoft.net (logiciels libres Win... et GNU/Linux)
www.kde.org
www.gnome.org
www.mandrakelinux.com
www.redhat.com
freshmeat.net (répertoires de projets libres)
sourceforge.net (idem)
us1.samba.org/samba/index.html (site officiel de Samba)
www.linux.org/apps/ (plein de progs, site en anglais)
www.vim.org/ (page officielle de l'éditeur Vi)
gcc.gnu.org/ (page officielle du compilateur GCC)
www.gnu.org/software/bash/bash.html (site du shell Bash)
www.acm.uiuc.edu/workshops/linux_install/lilo.html (site de lilo)
www.gnu.org/software/emacs/emacs.html (site d'emacs)

Les documents importants :

www.linux-france.org/article/these/manifesto/
Le manifeste de linux
www.freescape.eu.org/eclat/3partie/Raymond2/raymond2txt.html
Comment devenir un hacker
www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar_monoblock.html
La cathédrale et le bazar
www.linux-france.org/article/these/magic-cauldron/magic-cauldron-fr_monoblock.html
Le chaudron magique
www.linux-france.org/article/these/noosphere/homesteading-fr_monoblock.html
A la conquête de la noosphère

Site web de l'auteur : <https://andre-ani.fr>
Contact auteur : andreani.patrice@net-c.fr