Survol de GNU/Linux

Christophe Benz
cbenz@tuxfamily.org
http://cbenz.tuxfamily.org

16 février 2003

Note : Ce document en constante amélioration est destiné aussi bien à un débutant en informatique qu'à quelqu'un qui s'y connaît déjà. Aucune notion ne doit être obscure, après avoir lu une section (souvent un point obscur s'éclaircit en lisant la suite). Si quelque chose est incompréhensible, il faut le noter dans la marge. Les fautes d'orthographe et de grammaire sont également à éradiquer. Souvent je plaisante ouvertement de façon à rendre la lecture agréable, ou pour faire de l'ironie. Si une plaisanterie est incomprise, il faut le signaler, elle perd toute sa valeur. Barrer les paraphrases ainsi que les « longueurs », comme on dit pour le cinéma.

Ce document est mis en ligne sur Internet et disponible gratuitement à tout internaute. C'est pour cela que ce document doit être irréprochable.

Corriger également toutes les grossièretés : parfois je me suis emporté ou bien j'ai oublié que je serai relu ! Je veux également que tout reproche à Microsoft soit justifié, sinon je perds ma crédibilité. Noter tous les passages qui semblent trop durs envers Microsoft, dans la mesure où ils relèvent plus de l'insulte que de la critique.

Si certaines personnes ne veulent pas être citées ou veulent corriger la façon dont je parle d'elles, qu'elles me le fassent savoir.

Si tu envoies des corrections, c'est beaucoup mieux si tu donnes le numéro de la section au lieu du numéro de page. Les numéros des pages change à chaque nouvelle version, et pas les numéros de section.

Merci pour le temps consacré à la relecture, j'attends impatiemment les commentaires.

Table des matières

Chapitre 1

Introduction

Sommaire

Salut ! Enfin prêt ? Ça fait une éternité que je t'attends ! On va se tutoyer, maintenant qu'on a fait connaissance. Dépêche-toi, on embarque dans le monde fantastique dont tout le monde parle : GNU/Linux ! Ce livre est, depuis presque un an, en constante évolution. J'y note tout ce que je découvre, aussi bien au niveau technique qu'au niveau philosophique. Il ne doit pas être lu comme un manuel standard, c'est plutôt un carnet de bord que j'ai complété au fil du temps. Il n'a pas pour but d'être exhaustif. Ce livre s'adresse à ceux qui n'osent pas franchir le pas, et qui restent sous Windows tout en lorgnant timidement du côté de Linux. Il s'adresse aussi à des gens qui ne savent pas du tout ce qu'est Linux : je pense que c'est une bonne idée de leur faire lire ce livre.

Les experts ou les connaisseurs vont me reprocher le manque d'exactitude, c'est vrai que je simplifie beaucoup de choses (notamment au niveau de l'histoire de l'informatique). La réalité est plus complexe, je schématise pour que les débutants ne soient pas ennuyés et passent rapidement sur les détails. S'ils veulent plus renseignements, libre à eux d'aller se documenter sur Internet : je n'ai pas l'intention de reproduire les sites déjà existants.

J'ai choisi la méthode de la provocation. C'est un choix délibéré. Mais j'ai choisi cette méthode plutôt qu'une simple présentation des fonctionnalités de GNU/Linux pour plusieurs raisons :

- Il existe déjà une flopée de manuels « neutres », qui parlent de Linux. Ils sont bien faits, je ne veux pas les copier.
- Beaucoup de personnes utilisent Windows tout en s'en plaignant quotidiennement. On voit même des sites anti-Microsoft fonctionner grâce à des logiciels Microsoft ! Je pense, selon mon expérience, que ces personnes-là n'attendent qu'une seule chose : une bonne douche froide. Elles ont besoin de passer du stade de la critique niaise de Microsoft au stade « Je cherche une solution efficace ».
- Si j'y allais avec des pincettes, je n'aurais pas d'effet. Le lecteur se dirait : « Tiens, c'est intéressant ». Il poserait le livre sur son étagère puis continuerait comme avant. C'est évidemment son droit, d'ailleurs je pense que pas mal de personnes le feront quand même. Seulement, avec cette méthode de provocation[1], j'espère que les lecteurs feront leurs propres recherches, qu'ils s'intéresseront au sujet. Je ne leur donne pas un manuel du genre « Linux en trois heures sans difficulté ». Non, je leur met l'eau à la bouche, et à la fin je leur lâche la main pour les laisser aller s'inscrire dans des LUGs (clubs d'utilisateurs), chercher une distribution, se réunir avec des amis pour l'installer...
- Quant au titre, il est certes racoleur, mais c'est fait exprès. Il traduit

[1] Rassure-toi, ce n'est pas toi que je vais critiquer, mais les pratiques de Microsoft !

l'euphorie que j'ai ressentie pendant l'année lors de laquelle j'ai découvert Linux.

Nous allons commencer de façon très douce afin que tout le monde puisse suivre. Ensuite on s'attaquera à des notions de plus en plus complexes pour le débutant, mais toujours très palpitantes.

1.1 Mais c'est quoi, Linux ?

Eh oui, tu as le droit d'ignorer ce qu'est Linux. Si je ne te donnais pas la réponse à cette question dès le début, ce serait si dur de comprendre la suite ! Linux est un système d'exploitation pour ordinateur (son nom exact est GNU/Linux, on détaillera par la suite).

Système d'exploitation : c'est tout simple, dans le principe. Si tu as un ordinateur, il faut bien que tu lui confies des travaux, donc que tu lui donnes des ordres. Actuellement on envoie le plus souvent ces ordres depuis la souris ou le clavier. Mais on ne les envoie certainement pas directement au microprocesseur, qui est le cerveau de l'ordinateur ! Il est bien trop bête, il ne sait faire que des calculs hyper-simples. Non, on les envoie à un « machin truc » qui est logé dans le disque dur, qu'on appelle le *noyau* d'un système d'exploitation, et son boulot est de capter tes ordres et de faire bosser l'ordinateur en donnant le boulot au microprocesseur, le traitement des sons à la carte son, les graphismes à la carte graphique... chacun son boulot. Le chef d'orchestre est le *noyau*. Pourquoi « système d'exploitation » ? Parce qu'il exploite l'ordinateur, de façon à en tirer le maximum des performances. Sans système d'exploitation, l'ordinateur ne sait (presque) rien faire. Tu veux des exemples de systèmes d'exploitation : Windows, MacOS, BeOS, FreeBSD, OpenBSD,..., et GNU/Linux ! GNU/Linux permet de tirer le meilleur parti de ton ordinateur, et c'est ce qu'on va voir tout au long de ce guide. Nous n'en dirons pas plus sur les systèmes d'exploitation, car le sujet est incroyablement vaste.

Ce qui est très particulier et unique avec GNU/Linux, c'est la philosophie qui l'accompagne. Si tu es pressé d'en prendre connaissance, tu peux lire le chapitre 3. Cette philosophie est formidable, et si tu as entendu que Linux était gratuit, sache que c'est *bien plus que ça* : il est libre ! Il est bien gratuit, certes, mais il y a tellement plus d'avantages à l'utiliser que cette économie d'argent ! Utiliser GNU/Linux, c'est voir l'informatique d'une autre façon. Mais chaque chose en son temps.

1.2 Meilleur système, mais pour qui ?

Cette question est vite apparue : aucun système ne peut être meilleur qu'un autre. C'est vrai. Il existe plusieurs types d'utilisateurs des ordinateurs. Je partage la poire en deux :

- Pas mal d'entreprises ont vite compris qu'elles avaient tout intérêt à utiliser GNU/Linux pour sa stabilité, ses performances... ce n'est pas l'objet de ce livre. Beaucoup de professionnels s'en servent pour mettre en place des serveurs Internet, pour piloter les navires de la marine américaine, pour piloter les robots qui vont dans l'espace... Je ne m'adresse pas à eux, car mon discours ne les concerne pas. Ils ont choisi Linux la plupart du temps pour ses performances, pas pour sa liberté.
- Les particuliers : c'est à eux que je m'adresse. GNU/Linux fait peur aux gens, qui imaginent que ce système est réservé aux experts. Ce livre va leur enlever cette appréhension, mais c'est à eux de faire le choix final. Au moins, ceux qui resteront sous Windows l'auront choisi, et n'auront plus aucune raison de se plaindre. S'ils ne se plaignent pas, tant mieux ! Certaines personnes utilisent Windows et en sont très contentes. Mon but est de leur présenter GNU/Linux, car moi aussi j'étais très bien sous Windows il y a deux ans. Ces personnes gagneront toujours à lire ce livre, elles découvriront un nouvel univers. Il est vrai que les débutants en informatique, ceux qui se servent très peu de leur PC, n'éprouvent pas le besoin de changer. Je me pose toujours cette question : « Les débutants ont fait un effort pour se mettre à Windows. Pourquoi pas un petit effort supplémentaire pour Linux ? ».

  GNU/Linux n'est pas encore adapté à tout le monde. Par exemple, les amateurs de jeux vidéo dernier cri trouveront ridicules les jeux sous Linux. Personnellement, je trouve que les jeux disponibles depuis quelques mois sont tout à fait palpitants. Même si l'aspect graphique des jeux n'est pas hyper-évolué, le plaisir est au rendez-vous, comme avec les vieux jeux d'arcade. Les utilisateurs de logiciels de musique ne trouveront pas d'équivalent à Cubase ou Cool edit pro. Cependant, des logiciels comme Brahms ou Audacity sont en train d'être achevés.

  Ceux à qui Linux ne convient pas retourneront sous Windows ! Mais en termes de liberté, GNU/Linux est le meilleur.

  Les utilisateurs de Macintosh sont situés entre les utilisateurs de Windows et ceux de Linux. Ils ont un système très stable, comme Linux (d'ailleurs MacOSX est basé sur un noyau BSD), mais propriétaire (pas libre).

1.3 Historique de GNU/Linux

Je sais, c'est pénible pour certains, mais il faut bien connaître un peu d'histoire ! J'ai bien, dit : « un peu ». On pourrait détailler bien plus l'évolution des systèmes Unix. Seulement je ne pense pas qu'un débutant soit très intéressé par les détails. C'est pourquoi je ne ferai que survoler l'histoire de GNU/Linux.

La plupart des utilisateurs de Linux utilisent des logiciels du projet GNU sans même savoir ce que c'est. De même, les débutants sous GNU/Linux passent à côté de ce genre de considérations et pensent tout de suite aux partitions du disque dur ou à la façon de trouver des logiciels pour Linux. Bref, c'est *indispensable* de parler du logiciel libre !

Au début il y avait le néant. Puis les animaux, les Hommes et les civilisations. Ensuite le moyen-âge et la révolution industrielle. Puis, dans les années 1970 Unix est apparu. Unix est un logiciel qui servait à exploiter les ordinateurs de l'époque, à les programmer, et les faire travailler. C'était très puissant mais cher aussi. Les ordinateurs eux aussi coûtaient cher, très cher. Ce n'étaient pas des PC (Personnal Computer : ordinateur personnel).

Ensuite il y a eu dans les années 1980 les premiers ordinateurs *personnels*. Chacun pouvait avoir son PC. Avant de lancer les PC, IBM (à l'époque c'était le seul constructeur) a voulu les équiper d'un système d'exploitation. Mais lequel choisir ? Quel dilemme ! Des dizaines de jeunes gens se sont mis à travailler dur pour créer ce fameux système d'exploitation. C'est là que notre très cher Billou entre en scène : quelque temps après, IBM a choisi le système DOS de Bill Gates. DOS pour Disk Operating System, et MS-DOS pour Microsoft DOS, car il existait plusieurs DOS. La légende veut que Bill Gates ait créé DOS mais c'est faux. Il l'a acheté à d'autres programmeurs et l'a simplement étiqueté « Microsoft ». Pour moi, DOS est une simple invite de commandes, dépourvue de tout ce qui fait l'avantage d'Unix : la facilité d'utilisation. DOS est vraiment limité et l'utilisateur de base perd énormément de temps à taper les commandes au clavier.

En parallèle, Richard Stallman fonde en 1984 la Free Software Foundation (FSF) : la fondation pour le logiciel libre (à ne surtout pas traduire par gratuit !). Richard Stallman est l'un des rares informaticiens sur cette planète qui n'a qu'un seul souhait : produire un système d'exploitation de qualité, vraiment complet, extrêmement fiable et sécurisé, et qui par dessus tout serait libre. Tout le monde pourrait y participer et les codes source seraient obligatoirement livrés au public. Le code source d'un logiciel est comme une recette de cuisine : comment modifier un gâteau si on ne connaît pas la recette ? L'avantage, à la longue, est qu'on pourrait utiliser ce système sur n'importe quel ordinateur, que ce soit un PC, un Macintosh, un Atari, une station Sun, etc...

La fondation qu'il a créée a pour but de rassembler tous les programmeurs de la planète qui souhaitent participer au projet. Ce système tant espéré porte le nom de GNU. Cela signifie Gnu's Not Unix : c'est un acronyme gigogne (GNU est présent dans sa propre définition !). Unix est commercial, propriétaire (se dit d'un logiciel dont on ne dispose pas des codes source) alors que GNU n'a rien de tout cela. GNU est certes un système qui rapporte de l'argent à des personnes, mais ce n'est pas le but premier du projet. Pour donner l'exemple, Richard Stallman a programmé Emacs vers 1985 (un des meilleurs logiciels de toute l'histoire de l'informatique) et GCC (Gnu Compiler Collection, une collection de compilateurs libres, pouvant compiler du C, du C++, du Fortran, de l'Ada, du Java...).

Un grand nombre de programmeurs a suivi et très vite des logiciels libres sont apparus aux quatre coins de la planète. Richard Stallman est donc louable pour ses intentions, il a compris une grande chose : rien ne vaut la collaboration. Voici certaines de ses convictions :

> La mission de la FSF est de convaincre les utilisateurs de ne plus tolérer de logiciels propriétaires. Tout ce que je dis, c'est : vous avez une copie d'un programme, vous en faites ce que vous voulez, et c'est tout.
>
> Si vous développez un programme mais avez peur qu'on vous emprunte des idées, alors, ne faites rien !
>
> Trop de gens ne recherchent pas la liberté car ils ont été éduqués à supposer que la liberté est impossible dans le domaine des logiciels. Il faut leur apprendre, ou qu'ils comprennent d'eux-mêmes, que les entreprises leur font croire que le logiciel propriétaire est quelque chose de naturel. Pourtant, c'est faux.

Le but du projet GNU est désintéressé : c'est avant tout de créer un système d'exploitation de qualité exceptionnelle. Et comme par hasard, tous les logiciels que le projet a fourni sont d'excellente facture. Bien sûr, ceux qui le veulent bien peuvent envoyer des dons à la Free Software Foundation pour les aider à avancer.

Bon, revenons à nos moutons. En 1991, Linus Torvalds, un étudiant en informatique finlandais, trouve MS-DOS très mauvais et décide de créer un clone d'Unix mais libre : il respecte les règles du projet GNU en donnant ses codes source sur Internet afin que tout le monde puisse en profiter et surtout y participer. Très vite, son noyau (c'est la partie essentielle d'un système d'exploitation), baptisé Linux, est fin prêt. Linux est la contraction de Linus et de Unix. La légende veut[2] que le noyau ait été créé de toutes pièces mais c'est faux : Linus a acheté à son professeur d'informatique le système d'exploitation baptisé Minix. Ce professeur avait programmé une mini version d'Unix pour PC, de façon à faire des démonstrations à ses étudiants. Linus s'en inspire pour continuer le noyau Linux.

Avant, on utilisait les programmes GNU sous Unix quand on avait la chance, sinon sous DOS. Avec Linux, les utilisateurs des PC peuvent enfin avoir un clone d'Unix à la maison, ce qui fait qu'ils peuvent tirer un bien meilleur rendement de leur ordinateur personnel, et gratuitement en plus ! Remarque que j'ai dit « en plus » : on voit bien que la gratuité n'est pas l'argument principal. Si jamais GNU/Linux était de mauvaise qualité tout en étant gratuit, je ne m'en servirais pas ! On peut avoir un système à base de logiciels libres sans disposer du noyau Linux de même qu'on peut utiliser des applications commerciales qui ne font pas partie GNU sous Linux. Ce n'est pas parce qu'on est sous Linux qu'on utilise forcément du logiciel libre.

C'est pour cela qu'on ne doit pas dire « Linux » mais « GNU/Linux ». Pour simplifier, on ne dit que Linux, ça va plus vite. Mais il ne faut pas oublier que Linux n'est que le cœur du système. Le noyau « officiel » du projet GNU n'est pas Linux, mais Le Hurd. Ce noyau est encore en développement. Pour l'instant, les distributions Linux ont choisi Linux comme noyau, mais rien ne t'empêche d'essayer Le Hurd (il faut tout de même avoir des connaissances très solides, j'aimerais bien apprendre à le faire !). En fait le but du projet GNU est de fournir un système d'exploitation *plus* les logiciels qui vont avec, mais pour l'instant, comme Le Hurd n'est pas prêt, ils n'en sont qu'à l'étape des logiciels qui vont avec. Mais Le Hurd avance…

Mais sous Linux il existe des tas de projets libres ! GNU est le plus important, il accueille même des sous-projets comme Gnome (un bureau graphique pour Linux). Il existe aussi KDE, un autre environnement graphique, qui lui-même ne comporte pas moins d'un millier de logiciels gravitant autour (des navigateurs Internet, lecteurs d'e-mails, jeux, applications en tous genres, outils d'administration, suite bureautique KOffice,…). Pour Gnome c'est pareil d'ailleurs. Pourtant, KDE ne fait pas partie de GNU ! Voir la section 4.3 sur les environnements graphiques pour plus de détails.

Rappelons au passage qu'un système d'exploitation doit être fourni avec des logiciels sinon les utilisateurs ne peuvent rien faire. Le système d'exploitation lui-même doit être léger, mais les utilisateurs apprécient d'avoir des logiciels fonctionnels fournis avec. Les gens qui achètent Windows ne peuvent rien faire de concret avec leur PC, à moins de jouer au démineur, au solitaire ou de taper leurs textes avec Wordpad ! Ils sont encore obligés de payer

[2]Décidément, les légendes…

les logiciels. Finalement on se rend compte que les logiciels applicatifs reviennent plus cher que le matériel qui lui est à la pointe du progrès. C'est un comble pour un système d'exploitation. Imagine le gars qui vient d'acheter son nouvel ordinateur. On lui dit : « Soit tu utilises WordPad, soit tu payes Office très cher, soit je te donne les CD-Roms d'office que j'ai gravés (et tu ne payes rien) ». Ce gars, il va dire « Wordpad ne me suffit pas, j'veux pas payer Office, donc... je fais des copies illégales. » Mais il a un autre choix : les logiciels libres (par exemple utilise StarOffice). Les distributions Linux, elles, fournissent plusieurs centaines de logiciels libres.

Voilà, en deux mots, la petite histoire. Il y a peut-être des erreurs, si c'est le cas, merci de me corriger. Bien sûr, j'ai fait exprès de simplifier les choses.

Tu me pardonneras si j'écris Linux à la place de GNU/Linux.

1.4 Attention aux licenses

GNU/Linux n'est pas le seul Unix-like[3]. En effet, il existe tout un tas d'autres systèmes tels que OpenBSD, FreeBSD... qui sont tout aussi performants, parfois plus que le noyau Linux dans certains domaines. Mais ce qui fait que Linux est spécial, c'est qu'il est sous license GPL. La série des BSD est sous license... BSD.

La principale différence entre les license GPL et BSD est que

la license GPL spécifie que tout logiciel utilisant un logiciel GPL doit obligatoirement être lui aussi GPL.
Par exemple, si tu veux écrire un nouveau logiciel qui a besoin d'une portion de code existant déjà en GPL (par exemple un logiciel qui utilise MySQL[4] comme gestionnaire de bases de données), ton nouveau logiciel *devra* être sous license GPL. Cela garantit que ceux qui ont écrit MySQL n'auront pas travaillé pour rien ! Ils auront contribué (sans le faire exprès) à la création d'un *autre* logiciel libre.
Certaines personnes morales comme Microsoft voient cette clause comme un « cancer » qui les empêche de faire ce qu'ils veulent[5], mais d'autres personnes comme moi pensent que c'est le seul moyen pour que tout le monde puisse utiliser des logiciels comme on respire de l'air, et éviter de tomber dans le piège de la license BSD.

la license BSD n'est pas à mon humble avis la license idéale pour les logiciels libres car elle permet justement d'utiliser un logiciel sous BSD

[3] système ressemblant à Unix...

[4] MySQL est un gestionnaire de bases de données libre sous license GPL. Il est utilisé par énormément de personnes et de services sur Internet.

[5] les pauvres

> dans un logiciel propriétaire. Quel dégoût ! Microsoft utilise du code BSD dans Windows, Apple aussi dans MacOS... c'est bête car si on écrit un logiciel libre c'est qu'on défend des valeurs élémentaires de partage et si on laisse des gens qui ont des intérêts à l'opposé de cette philosophie s'emparer de notre travail, autant collaborer directement avec eux en écrivant un logiciel propriétaire dès le début !

C'est pour ça, à cause de la license BSD, que Apple ou Microsoft utisent du code libre dans leurs logiciels propriétaires.

1.5 Pourquoi ce guide ?

GNU/Linux est le reflet d'une façon de penser l'informatique : le logiciel libre. Les logiciels libres sont partout, sans que tu ne t'en rendes compte. Eh oui, tu n'es pas obligé d'utiliser GNU/Linux pour utiliser des logiciels libres. Si tu es déjà allé sur Internet, tu as forcément utilisé du libre, puisque le protocole TCP/IP qui permet aux ordinateurs de communiquer *est* libre. La majorité des sites web sont hébergés sur des ordinateurs fonctionnant grâce au serveur http (ou serveur web) « Apache », dont le boulot est de t'envoyer le site demandé. Eh bien Apache, c'est un logiciel libre. Bien sûr il existe le serveur Microsoft, mais il n'est pas fiable. La preuve : Hotmail n'a pas toujours appartenu à Microsoft. Avant de lui appartenir, Hotmail fonctionnait sur des serveurs à base de logiciels libres. À l'époque où Microsoft l'a racheté, ils ont décidé de passer à leur serveur. Il y a eu énormément de problèmes, et ils ont décidé de reprendre les logiciels libres !

Les plus grands logiciels libres sont assez souvent disponibles pour Windows : tu peux faire de la retouche d'image avec « The Gimp », naviguer avec Mozilla (version libre de Netscape) sur Internet, tout cela sous Windows. Mais sous Linux *tout* est libre. De plus malheureusement tous les logiciels libres n'existent encore pas sous Windows, et ceux dont c'est le cas ne sont pas toujours bien « portés », donc ceux qui veulent vraiment s'en servir, soit les compilent pour Windows (les développeurs n'ont pas toujours envie de le faire) soit passent à Linux.

Linux commence à se faire connaître du grand public depuis trois ans environ, mais depuis un ou deux ans les choses s'accélèrent. Le débutant croit souvent, à tort, que GNU/Linux est nouveau (Linux a 10 ans et GNU a plus de 20 ans !). Il croit donc que GNU/Linux n'est pas mature, qu'il n'est pas efficace... J'ai dressé la liste des idées reçues au chapitre 2.

Certaines personnes qui utilisent Linux ressentent au bout d'un moment l'envie de faire une démonstration à des amis, collègues, etc... Si je te mets devant un ordinateur qui marche sous Linux avec un *super* environnement graphique, tu vas tout de suite dire « wouah ! C'est extraordinaire Linux, je le veux ». Si je te montre cette fois-ci la console (le fameux mode texte), tu diras « pouah, c'est vilainie messire ! Je ne veux point de ce système rustre ». Et tu ne penseras même pas (si tu n'es pas informé) que sous Linux on peut choisir et avoir tout à la fois, du coup ton choix s'arrêtera là. Dommage.

C'est ce problème-là (évidemment caricaturé) que rencontrent les gens qui veulent faire une rapide démonstration de Linux. Ce n'est pas si simple que ça d'expliquer Linux à quelqu'un qui n'y connaît encore rien. Évidemment, tu vas te dire : « encore un guide de plus ! », il y en a tellement ! Seulement ce que j'essaie de faire dans ce guide, c'est *présenter* en priorité Linux aux débutants, et pas forcément apprendre à s'en servir (on le verra vers la fin du document). Justement, tu trouveras juste après ceci une grande partie de texte qui explique comment on *pense* quand on utilise Linux, alors que d'autres guides te plongent d'entrée au cœur du système. Ici on va simplement « survoler » tous les aspects de Linux sans entrer trop profond. D'autres guides plus spécifiques sont cités à la section 6.2.

Les magazines « Linux » parlent souvent de Linux comme simple alternative, et ne parlent finalement même pas de GNU. Elles font même tout un tas de pubs pour les logiciels commerciaux sous Linux. Les seuls magazines qui respectent cette éthique sont cités à la section 6.5. Bien sûr que certains logiciels commerciaux sont les meilleurs actuellement dans leur domaine, et on les utilise volontiers. Seulement ils sont en train de se « libérer ». Par exemple, Netscape avance en parallèle avec Mozilla, navigateur Internet libre. De même, Star Office, une excellente suite bureautique commerciale mais gratuite, libère son code source et s'appelle Open Office, une version parallèle. En fait ces deux versions ne sont pas exactement les mêmes : la version commerciale garde une partie de son code source fermé, comme les correcteurs d'orthographe... mais Open Office les ré-implémentera sûrement à la longue. Utiliser des logiciels commerciaux sous Linux permet toujours de profiter d'un système de base solide et libre. Bien sûr il existe aussi des suites bureautique libres comme KOffice, faisant partie intégrante de KDE. Voir la section 4.3 pour plus de renseignements sur les environnements graphiques.

1.6 Bienvenue dans le monde de Linux

Étant donné que je suis un vrai débutant qui est en train de progresser petit à petit, je vais mieux pouvoir te dire ce qui paraît le plus curieux quand on arrive sous Linux, comme ça tu seras dès le début libéré d'un tas de questions (voir le chapitre 2). Désolé pour les personnes du sexe féminin mais je n'ai pas envie, en tant que grand paresseux, de mettre des « e » entre parenthèses à la suite de chaque adjectif. Vous m'excuserez, c'est une simple question de facilité d'écriture, et je suis très content que les femmes comme les hommes utilisent les logiciels libres.

Afin d'amener le débutant dans le monde merveilleux de Linux tout en douceur (j'ai découvert moi-même Linux il y a 4 mois[6] seulement, pour dire si c'est simple de s'y mettre), nous aborderons les diverses caractéristiques de Linux, nous l'observerons sous différents angles, le décortiquerons dans ses entrailles (pas trop profond, le pauvre !) mais tout en restant *accessible* et *compréhensible*. Rien n'est plus pénible que de trouver des documentations soi disant « pour débutants » mais avec beaucoup trop de mots incompréhensibles. Donc, si tu ne comprends pas les mots que je vais employer, c'est que je suis allé trop vite. Envoie-moi un e-mail : je répondrai volontiers.

Nous aborderons donc :
- un élagage des *idées reçues* les plus stupides
- la *philosophie* de Linux
- introduction à l'*utilisation* de Linux
- les *commandes fondamentales* de Linux
- comment trouver de la *documentation* (très abondante d'ailleurs) sur Internet ou dans ton disque dur. Je donnerai une foule de très bonnes adresses à aller visiter sur Internet
- quels sont les logiciels *qu'il faut absolument connaître* pour se sentir bien entouré, et je te donnerai un inventaire complet de tous les logiciels qui m'ont permis de réaliser ce document (tous sous Linux)
- un tableau donnant *l'équivalent* des logiciels Windows pour Linux
- un guide pour réaliser les travaux les plus courants (imprimer, visualiser certains fichiers. . .)
- un rapide descriptif d'une *installation typique* de GNU/Linux

Mais toujours de façon facile et progressive. En fait je vais imaginer que je ne connais rien en informatique et que je dois me l'expliquer tout seul (c'est très difficile pour moi de jouer le jeu). Tu trouveras de nombreux schémas pour mieux comprendre. Je me trouve en face d'une autre difficulté : est-ce que tu connais l'informatique en général ? le monde de Windows ? (attention, remarque que je fais des efforts pour ne pas balancer un tas de « $*# ?@ ! » ; mais ça viendra dès que j'aurai fait la liste des inconvénients de Windows). Ou alors tu entres dans l'informatique tout court ? (c'est plus rare, les tout débutants sont en face de leur ordinateur tout neuf avec Windows pré-installé : comment pourraient-ils connaître Linux d'eux-mêmes, les pauvres ?). Eh oui, selon ton niveau actuel, c'est pas pareil, donc je mettrai les informations destinées aux débutants les plus *purs* sous la forme suivante :

---

[6] hum hum. . . depuis que ce document existe, ça fait un peu plus !

notion difficile explication qui la rend facile ;)

Pour moi, un débutant en informatique est pur comme la blanche neige, c'est-à-dire qu'il n'est pas souillé, infecté de notions souvent fausses, en tout cas inutiles, apportées par Windows. Quand je dis notions, il ne s'agit pas seulement des notions d'informatique, mais aussi des notions morales (questions de piratage). Par exemple on baratine le débutant en lui faisant croire que le disque dur d'un ordinateur s'encrasse inévitablement et qu'il faut le défragmenter s'il veut que l'ordinateur « aille plus vite ». Ce qu'on ne dit pas, c'est que Windows ne sait pas gérer les disques durs correctement : son système de fichiers fat32 est une réelle catastrophe pour ton disque dur ! C'est lui qui fragmente ton disque, et c'est toi qui perds du temps pour le défragmenter. En fait, c'est simple à comprendre : si les données sont éparpillées, pour les atteindre, la tête de lecture du disque dur doit se déplacer pour les atteindre, d'où un ralentissement notoire. Par contre si les données sont condensées effectivement ça va plus vite. Les systèmes Linux utilisent des systèmes de fichiers qui ne se fragmentent tout simplement pas. Même dans les années 70, Unix utilisait un bien meilleur système de fichiers que Windows aujourd'hui. Évidemment, les systèmes de fichiers d'aujourd'hui sous Linux ont dépassé ceux des vieux Unix.

Je vais essayer d'être le plus court possible, mais il faut aussi que tu fasses un effort de lecture. En rédigeant ce document, j'ai été confronté à l'éternel problème : par quoi commencer, que dire et à quel moment, etc... C'est pourquoi je te demanderai de lire le document jusqu'au bout avant d'abandonner. Merci de t'intéresser à Linux, tu ne le regretteras vraiment pas ! Note aux amoureux de Windows : lisez d'abord la section 11.2, il y a un message qui vous est spécialement destiné.

En fait tout au long de ce document je vais essayer de te prouver à quel point la politique menée par Microsoft est dangereuse pour tous, et aussi pourquoi tu *mérites* d'utiliser les logiciels libres. Ce n'est pas une contrainte, mais un droit à la qualité et au respect.

Chapitre 2

Élagage des idées reçues les plus stupides

Sommaire

2.1 Linux, c'est pour les...

Beaucoup de gens donnent leur avis sur Linux alors qu'ils ne l'ont *même pas* essayé ou regardé fonctionner. Ils s'arrêtent sur leur jugement superficiel. Toutes ces personnes parlent sans connaître le sujet, et du coup elles disent n'importe quoi. J'ai dressé la liste des phrases les plus courantes que j'ai entendues, et je les ai commentées, en montrant du doigt à chaque fois l'énorme erreur que font ces gens.

Je recommande donc à ces personnes de dire plutôt « Je ne connais pas Linux et je n'ai pas envie pour le moment de m'y intéresser ». Ce serait plus sage que d'accabler Linux de reproches vraiment injustifiés. Je recommande aussi à ces personnes de lire ce document en entier pour se débarrasser de leur ignorance et ainsi *profiter de Linux.*

Voici donc ces phrases-types :

- « Linux, c'est pour les techniciens, les experts » : c'était vrai il y a trois ans (et encore...). Aujourd'hui Linux est accessible à tout le monde grâce aux distributions qui ont travaillé sur l'auto-détection du matériel, la facilitation de l'installation, etc... GNU/Linux est aujourd'hui facile à installer, et de plus en plus de logiciels grand public apparaissent : tu n'as qu'à aller sur le site des logiciels pour KDE (on verra ce que c'est) : `http://apps.kde.com`.
- « Linux n'est utilisé que par des groupuscules » : je réponds que Linux n'est plus seulement utilisé par des particuliers accrocs au bidouillage, mais que des entreprises (là ça devient sérieux!) l'ont adopté depuis deux ans déjà. Ne parlons pas des administrateurs système (les personnes qui s'occupent de gros réseaux d'ordinateurs) qui n'utilisent que des systèmes Unix (dont Linux fait évidemment partie puisqu'il en est un clone libre). Utiliser Linux ne te mettra pas à part, d'ailleurs dans beaucoup de villes et même dans le monde entier il existe des LUGs (Linux Users Groups). Tu trouveras la liste des LUGs dans le chapitre 6. Dans les LUGs, on organise des réunions afin de mieux connaître Linux. De plus, le monde de Linux est ouvert à tous, libre, non commercial... Voir le chapitre 3.
- « Linux est un phénomène de mode engendré par la presse » : alors là, c'est la réaction typique des gens qui n'ont jamais entendu parler de Linux et qui voient depuis quelque temps fleurir des magazines Linux chez les marchands de journaux. Linux existe, dans sa première version stable, depuis 1991 environ (date de sortie du noyau version 1). Son développement a commencé un an auparavant. A l'époque on utilisait Windows 3.1 : c'était il y a très longtemps à l'échelle informatique. Si tu voyais comment il a évolué depuis par rapport à Windows, mais

ça c'est une autre affaire. La presse parle de Linux parce que Linux prend de l'ampleur *et non pas l'inverse*. D'ailleurs il *faut* dénoncer les magazines pro-Windows qui balancent tout un tas de mensonges sur Linux.

- « Ceux qui utilisent Linux le font par esprit de contradiction » : bien sûr qu'on utilise Linux pour ne pas faire comme tout le monde, *puisque* la plupart des gens utilisent un système d'exploitation qui n'est pas fiable ! Par contre si on utilise Linux, c'est parce qu'on a remarqué de façon explicite que Linux était bien mieux, qu'il *dépassait incroyablement* Windows (on va le voir tout au long de ce document), et non pas par pur esprit de contradiction ! Si Linux était nul, ses utilisateurs ne l'utiliserait pas, et alors ils utiliseraient Windows comme tout le monde évidemment. Faire comme tout le monde, c'est un comportement stupide, si on le fait sans *réfléchir*. C'est comme les phénomènes de mode éphémère.
- « Linux n'est pas à la hauteur des bons vieux Unix » : c'est faux, et c'est même le contraire. Unix bénéficie auprès de certaines personnes d'une réputation extraordinaire. Attention, je n'ai pas dit qu'Unix était mauvais en soi ! C'est un très bon système d'exploitation digne de ce nom, mais le problème est qu'il est propriétaire. Tu vas comprendre tout au long de ce document les avantages *concrets* à utiliser du logiciel libre, et tu te rendras compte qu'utiliser un Unix propriétaire est certes mieux que Windows mais au niveau de l'évolutivité c'est la même chose : on est prisonnier de l'éditeur et des programmeurs du logiciel. Linux a fait ses preuves depuis ces 10 dernières années : il a calculé les effets spéciaux du film « Titanic », il sert au tri postal aux États-Unis, sert à piloter des navires de la marine, etc... En fait on a droit à tous les avantages d'Unix en administration système *plus la liberté* !

  Administration système : c'est une discipline qui consiste à gérer un réseau de plusieurs ordinateurs, la moyenne étant peut-être de 50 à 100 postes. Il faut savoir manipuler le système avec aisance pour que les utilisateurs ne rencontrent jamais de problèmes. Le métier s'appelle : « administrateur système ». Les administrateurs système le savent : rien ne vaut les systèmes du type Unix.

  À vrai dire, ceux qui croient qu'Unix est toujours meilleur que Linux s'appuient sur un argument qui leur paraît inébranlable : Linux est un clone d'Unix et tout clone n'arrive jamais à la hauteur de son original. Eh bien qu'ils essayent Linux, ils verront bien ! Grâce à la liberté, le clone a dépassé l'original.
- « Moi j'utilise Windows et j'en suis très content. Il plante mais je fais

avec » : ces gens-là sont ceux qui mentent le moins, mais sont quelque peu hypocrites envers eux-mêmes. Qui ne s'est jamais plaint de Windows ? Si jamais c'est le cas, que Windows te plaît et que tu ne l'as jamais critiqué, je l'admets, tu as raison de penser que tu n'as pas *besoin* de te mettre à utiliser Linux. Cependant ce serait dommage de ne pas essayer Linux même si Windows te paraît très bien. À toi de voir mais je ne peux pas te forcer : j'ai été dans ton cas pendant *très* longtemps. Ce n'est pas une question de fonctionnalités, mais plutôt une ambiance de travail différente lorsqu'on est sous un système libre[1].
Non, ces gens-là devraient plutôt dire : « j'ai un ordinateur qui est on ne sait combien de fois plus puissant que ceux qui ont envoyé des hommes dans l'espace, et le mien avec windows n'est pas capable de rester un jour entier sans redémarrer ou planter ! ».
De plus ces gens-là ne se rendent pas compte à quel point ils sont méprisés par l'industrie du logiciel. Même si GNU/Linux leur paraît trop technique ou inabordable, ils y gagneront à prendre connaissance du logiciel libre.

2.2 Inutile d'abandonner windows pour installer Linux

Évidemment qu'on peut garder windows ! Bien que quelques mois après avoir installé GNU/Linux, cela te démangera (moi j'ai encore windows, mais je ne l'ai plus utilisé depuis des mois !). Cependant, pas d'affolement, pour le moment je te conseille vivement de *garder* windows. Parce que sinon tu vas t'affoler et tu vas mettre un bordel pas possible dans ton ordinateur et après tu diras que tout est de la faute de Linux et que Linux « c'est de la merde » et ainsi de suite. Attention, je répète : tu peux garder windows après avoir installé Linux, ce n'est pas pour ça que tu seras rejeté par la communauté du logiciel libre !

[1]Certains me prendront pour un politicien, mais le fait que « ça marche » n'est pas le seul critère dans le choix d'un système d'exploitation. D'ailleurs, qui a dit que Windows marche (bien) ?

2.3 Installer Linux ne relève plus de l'impossible

C'est vrai qu'il y a quelques années (jusqu'en 1998 je pense), c'était encore difficile (mais pas impossible !) d'installer Linux pour un débutant. Il fallait *tout* faire à la main (repartitionner le disque dur par exemple). Bien sûr, cela existe encore pour certaines versions de Linux (on dit distributions), mais l'avantage aujourd'hui pour un débutant, ou même pour quelqu'un qui ne veut pas se casser la tête, est qu'il existe des distributions qui simplifient l'installation de façon considérable.

distribution : tout système Linux est constitué de divers éléments (noyau ou kernel en anglais, commandes système, logiciels, etc, on le verra dans le chapitre 3). Une distribution est un ensemble plus ou moins exhaustif de ces éléments, le tout diffusé sur des CD-roms (ou sur Internet). Il existe un grand nombre de distributions. Les plus connues sont Mandrake, RedHat, Suse, Debian, Slackware, Caldera, Corel Linux, Turbo Linux... Certaines sont distribuées gratuitement (ou librement devrais-je dire, on verra la différence entre les termes gratuit et libre dans le chapitre 3) dans des revues, on dit qu'elles sont sous licence GPL, et d'autres sont vendues dans des boîtes en magasin, mais toutes sont de très bonne qualité : surtout ne pas penser que gratuit signifie médiocre et que payant signifie « de bonne qualité ». Les versions boîte sont plus complètes, c'est vrai. Par exemple la distribution Mandrake GPL est constituée de 3 CD et la version boîte contient 7 CD plus toute la documentation papier, qu'on peut par ailleurs trouver sur Internet au format PostScript, PDF ou encore HTML. En fait, s'il existe plusieurs distributions, c'est parce que chacune est adaptée à une utilisation particulière (serveur, grand public, programmation, bureautique...). Si j'ai choisi Mandrake, c'est parce qu'elle couvre tous les domaines de façon uniforme. De plus, on peut toujours rajouter des programmes provenant d'une autre distribution : *les différentes distributions sont toutes compatibles au niveau des logiciels* car elles utilisent à la base les mêmes logiciels, et c'est simplement l'enrobage qui change. Personnellement, je n'ai jamais eu besoin de chercher des programmes dans d'autres distributions car la mienne est très complète. Si jamais un programme venait à manquer, je préfère aller le chercher sur Internet puis l'installer (ou le compiler) plutôt que de le prendre dans une autre distribution.

En fait, le travail d'une distribution est de récolter les meilleurs logiciels (pas toujours libres malheureusement) et de les mettre à la disposition

> du public, le tout enrobé dans une procédure d'installation. En plus, les distributions doivent tester et re-tester, afin de s'assurer que tous les composants logiciels fonctionnent bien ensemble. Avant que les distributions existent, on allait sur Internet et on téléchargeait le noyau Linux puis les logiciels GNU les uns après les autres, puis diverses choses. On compilait tout soi-même et on avait un système d'exploitation *sur mesure*, ce qui est très excitant pour l'informaticien avancé. Tout le monde ne peut pas se le permettre seul, c'est pour cela que les distributions Linux existent. Un système bâti sur mesure est appelé « Linux From Scratch » c'est-à-dire : Linux à partir de rien du tout.

Certains puristes pensent que Linux s'éloigne de sa route en facilitant ainsi l'installation et l'utilisation, au détriment de la maîtrise et de la connaissance exacte de son système d'exploitation : quelqu'un qui s'oblige à installer Linux de façon archaïque, c'est-à-dire en faisant tout à la main, connaîtra exactement son système, saura ce qu'il aura installé en détail, et surtout n'aura rien d'inutile, donc son système d'exploitation sera idéal pour lui. C'est pourquoi on dit qu'il y a « une version différente de Linux pour chaque utilisateur de Linux ».

Donc, si tu débutes, je te conseille d'utiliser la distribution « Linux Mandrake », la version actuelle est la version 8.2. Peut-être que le numéro de version de Mandrake aura évolué entre l'écriture de ce livre et le moment où tu le lis. C'est celle qui est, à mon goût, la plus facile à installer, elle est globalement francisée (MandrakeSoft est une entreprise française qui travaille dans le logiciel libre). Vers la fin de ce document, je décrirai comment installer une distribution. Bien sûr, tu peux opter pour RedHat ou Slackware, c'est ton choix et nous le respectons. Slackware est la distribution la plus proche des Unix d'antan, c'est-à-dire sans fioritures... Ceux qui ont connu Unix l'aiment bien.

2.4 Quelle est la difficulté majeure quand on passe à Linux ?

Linux est un système d'exploitation très simple, contrairement à ce qu'on entend souvent. Cela est dû au fait que GNU/Linux est conçu de façon robuste et précise dès le départ. Bien sûr, il y a plusieurs niveaux d'utilisation, de d'utilisateur simple à l'expert, mais un débutant peut comprendre tout et très vite, *parce que* tout est logiquement agencé. Pas de contradictions, rien n'est caché ni secret. Tout coule de source. J'ai remarqué au contraire qu'un débutant en informatique est assez frustré s'il débute sous windows car il a l'impression qu'on lui cache tout. Sous Linux, comme le système est construit de la tête aux pieds de façon logique, le débutant n'aura pas de mal à comprendre son fonctionnement. Mais la plus grande difficulté n'est pas là : elle réside dans la philosophie, dans la façon de penser l'informatique.

La difficulté majeure est de changer d'état d'esprit. Oui tu as bien lu ! La difficulté majeure n'est pas vraiment concrète mais c'est le changement de mentalité. Nous allons le voir par la suite, dans le chapitre 3. Ensuite quand tu auras enfin compris ce que j'aurai expliqué, tu verras que tout découle de cette philosophie et il ne te restera plus qu'à apprendre le reste, sans difficulté puisque tout est à portée de main.

La plupart des débutants se *ruent* sur leurs CD d'installation et installent Linux comme ils installeraient un vulgaire logiciel trouvé dans une revue ! Ensuite ils posent des questions provenant tout droit de la mentalité de windows : « où est-ce qu'on télécharge des programmes », « quel logiciel utilise-t-on pour repartitionner le disque dur »... Ces questions n'auraient pas été posées si ils avaient pris conscience *pour de bon* de la mentalité qu'on a sous GNU/Linux. Sinon ils se seraient dit d'eux-mêmes : « nul besoin de télécharger les programmes, la distribution me les fournit » et pour la deuxième question « la procédure d'installation de ma distribution prévoit toutes les étapes nécessaires ». Bien sûr, si on veut trouver un logiciel qui n'est pas fourni par la distribution, on peut aller voir sur Internet. Je donne la liste des meilleures adresses à la section 6.2.5.

Tu sais bien qu'il est très difficile de changer d'état d'esprit d'un seul coup. On a peur de se rendre compte qu'on s'est trompé si longtemps, et du coup on préfère garder ses habitudes. Il faut savoir admettre ses erreurs. Si ce petit survol de Linux ne t'a pas convaincu, c'est sûrement que tu n'es pas prêt à franchir le pas, et ce n'est pas en te forçant que ça va arranger les choses. Je vais plutôt essayer de te faire saliver pour que tu veuilles, par toi même, installer Linux. J'ai quelque peu incité un copain à installer Linux sans qu'il le veuille trop et, comme il n'en ressentait pas assez le besoin, il en a conclu que Linux n'était pas si important que ça. Si je lui avais mis l'eau à la bouche, si je lui avais parlé de la philosophie des logiciels libres, il aurait lui-même demandé à l'installer et tout se serait bien passé. En fait il était déçu parce que je lui ai quelque peu forcé la main. Ce n'est pas trop de ma faute puisque moi aussi je débutais, et à l'époque je n'aurais pas été capable d'écrire un dixième de ce document. C'est pour cela que je te recommande de lire quand même jusqu'au bout, pour que tu puisses juger en toute connaissance de cause et éviter de passer à côté de choses qu'il serait dommage de ne pas connaître.

Tu pourras te contenter des environnements de bureau (pour ne citer qu'eux : KDE, Gnome...), interfaces homme-machine passant par des fenêtres et des icônes, interfaces qui sont d'ailleurs bien mieux que windows, mais à la longue tu préfèreras la console, le joli mode texte comme disent les débutants. Ce n'est pas un *mode texte* puisque on peut y accéder en même temps que le *mode graphique*. C'est les deux à la fois, on bascule entre les deux en appuyant sur Ctrl+Alt+F1 (ou F2 etc...) et F7 est normalement le mode graphique. On peut aussi ouvrir une console dans une fenêtre graphique. La console est *puissante* car on peut *tout faire* en un tour de main (en tapant quelques commandes) au lieu de se plier aux interfaces graphiques qui nous permettent de faire *uniquement ce qui est prévu* alors qu'avec la console on peut faire de ces combinaisons !!! Tu peux ouvrir une console depuis un environnement graphique en cliquant sur « Terminal » ou « Konsole » selon que tu sois sous Gnome ou KDE.

2.5 Un débutant a besoin de se sentir assisté, et Linux paraît difficile

Que Linux soit difficile, c'est une idée fausse. Tu as déjà vu KDE 2, ou Gnome 1.4 ? Mais c'est même bien mieux que windows, du point de vue de l'aisance lors de l'utilisation ! Et puis sommes-nous vraiment bien assisté sous windows ? Pas si sûr ! Le fichier d'aide accessible dans « Démarrer - Aide » est si pauvre que je n'y ai jamais rien appris. Par contre l'aide de Linux est très intéressante. On trouve sur son disque dur des centaines de pages de manuel, si bien triées qu'on accède à l'information recherchée en une seconde.

De plus, il existe une multitude de forums, de documents sur Internet,... consacrés à GNU/Linux et il suffit que tu poses une question dans un forum pour voir la réponse arriver quelques jours après ! Les utilisateurs de Linux sont très solidaires entre eux. Va voir la section 6.2 pour plus de renseignements.

Le débutant se sentira donc très aidé, et l'idée que Linux est difficile est complètement décalée de la réalité.

En fait, Linux est organisé différemment de windows, mais dans le bon sens du terme ! GNU/Linux a bien été conçu dès le départ, sur le modèle des systèmes Unix très organisés. Le problème est qu'un utilisateur de windows est « moulé » par Microsoft, qui lui transmet des idées fausses le plus souvent pour vanter les mérites de ses produits. Comme disent les informaticiens : « Il suffit que Bill Gates décide qu'un défaut est une fonctionnalité pour qu'il le devienne ». Ai-je besoin de rappeler le cas du logiciel defrag (on verra plus tard) ? Et les anti-virus ? Et les trous de sécurité d'Outlook Express ? Et des numéros de carte...

Donc celui qui passera de windows à Linux sera « sali » de mauvaises connaissances. Le plus difficile sera pour lui de les oublier pour ré-apprendre les notions simples, logiques... qui sont à la base de Linux. Sous Linux, rien n'est incompréhensible. Je me permets de citer une phrase assez drôle qu'a prononcée un ami de l'association AXUL (Association aiXoise des Utilisateurs de Logiciels libres) : « Sous windows il y a des problèmes à subir, par contre sous Linux il y a peut-être plus de problèmes pour un débutant mais il sont à corriger ! ».

C'est donc cela le plus difficile, aussi faut-il être bien entouré pour résoudre tous ses problèmes, au lieu de se morfondre seul pendant des semaines. Attention à cette solitude car elle pourrait bien déboucher sur la déception.

2.6 Quelle distribution dois-je choisir ?

Cette question, je me la suis posée dès le début ! Que choisir entre RedHat, Debian, Slackware, Mandrake, SuSe, Caldera, Corel,... ? Personne n'osera te répondre de façon précise. C'est bien ce qui m'a énervé au début, mais aujourd'hui je comprends tout à fait cette « timidité » : aucune distribution est meilleure que les autres. Simplement elles sont orientées vers des tâches ou ciblées vers des utilisateurs bien précis. Par exemple, Mandrake a choisi de faciliter la vie des débutants... ben tiens, justement, tu es débutant ! Pour ma part c'est la distribution que je te conseille. Elle est devenue si simple à utiliser que même les professionnels s'en servent.

Par contre, SuSe est plus robuste, elle a moins de problèmes de configuration... ce qui ne veut pas dire que Mandrake en a tant que ça !

RedHat est, elle, très « business ». Ce n'est pas ce que recherche un débutant : elle inclut en priorité des logiciels de serveur Internet... ce qui ne veut pas dire que Mandrake n'en inclut pas ! Mandrake fournit *tout* ce qu'il faut.

Debian est par contre à fond dans la philosophie GNU : elle n'inclut que des logiciels libres alors que certaines distributions donnent des logiciels commerciaux (comme Mandrake !). Cependant tu es prévenu lors de l'installation de la distribution que certains logiciels ne seront pas libres (situés sur le 3$^{\text{ème}}$ CD-Rom), et tu as le choix de ne pas les installer.

Et taratatam : il existe une distribution hors du commun. J'ai nommé « DemoLinux », celle qui prend le moins de place sur le disque dur. En effet, elle ne s'installe pas, mais s'utilise *entièrement* depuis un CD-Rom que l'on insère au démarrage de l'ordinateur. Après avoir répondu à quelques questions concernant le réseau (ne rien mettre si tu n'en as pas), tu te retrouves en face d'une distribution « clone » de Debian, avec l'écran d'accueil graphique.

Ainsi on peut essayer un Linux totalement fonctionnel, avec tous les outils nécessaires[2].

2.7 Tout le monde peut introduire des virus, alors ?

Beaucoup de gens, notamment les pseudo admistrateurs de réseaux mal informés, pensent que le fait que Linux soit Open Source (on a droit aux codes source, voir la section 3.1) entraîne que les programmeurs peuvent mettre exprès des virus notamment dans le noyau, camouflés dans des *super* fonctionnalités ! Ce n'est pas le cas pour plusieurs raisons :

- déjà ce serait stupide car déranger les personnes qui travaillent en leur envoyant des virus ne fait pas partie de la philosophie des utilisateurs de Linux (mais cet argument est si loin de la véritable nature humaine...)
- le noyau est vérifié méticuleusement à chaque version, et tous les ajouts au noyau sont vérifiés par plusieurs « kernel hackers » et rien n'y est ajouté à la légère.
- les programmeurs envoient leurs modifications sous la forme de *patchs*, qui sont appliqués au noyau afin d'en changer une *petite* partie. L'équipe qui applique ces patchs (dont le responsable n'est autre que Linus Torvalds) les vérifie soigneusement et saurait tout de suite si ce patch est vérolé, et ce de façon très rapide puisque les programmeurs qui font ces patchs n'envoient qu'une minuscule portion de code et pas *tout* le noyau modifié. C'est donc bien plus facile à vérifier.
- si le virus était quand même intégré au noyau pour cause d'inattention, ce qui n'est jamais arrivé, alors dès qu'on s'en rendrait compte, le lendemain on trouverait le correctif ! C'est ça le logiciel libre. C'est arrivé qu'il y ait des trous de sécurité (pas des virus !) dans les versions anciennes du noyau, et 3 heures (seulement) après on trouvait le correctif. La question est alors : « Pourquoi ceux qui découvrent les bugs ne gardent-ils pas le secret et ne font-ils pas des misères au monde entier ? » Eh bien c'est très simple : Linux est à la base programmé par des experts qui veulent obtenir un noyau parfait. Trouver les failles dans leur propre logiciel fait partie de leur boulot, espérons qu'ils ne sont pas trop narcissiques. Ensuite ces gens-là sont ceux qui ont pour conviction de faire partager la connaissance à tous. Pourquoi garderaient-ils un tel secret ? L'histoire l'a *toujours* démontré : la liberté finit par l'emporter sur les complots. Nous sommes dans un complot : Microsoft s'approprie

[2] commandes shell, compilateurs, environnements graphiques...

la connaissance alors que GNU/Linux la diffuse au maximum. Bouh, c'est la guerre, tout le monde tremble de peur[3]. Alors en tant qu'utilisateur, que choisis-tu ?

Sous windows, au cas où tu ne le saurais encore pas, des failles de sécurité présentes dans « Outlook Express » ou encore dans « office 2000 » permettent à des petits malins de lancer des scripts (petits programmes) qui peuvent faire le pire (comme effacer tes données), ou alors des macros pour la suite bureautique ! Depuis que « office » est sur Macintosh, Microsoft a enfin démocratisé une plate-forme de développement de virus multi-systèmes d'exploitation. Le pied !

Ou alors, récemment, Microsoft vient d'admettre que la version XP du Windows Media Player envoyait par Internet la liste des CD audio lus ainsi que des DVD.

Le logiciel libre règle tout ça car ceux qui programment, ce sont ceux qui utilisent les logiciels, et vice versa. Et ces gens n'ont pas envie de subir de telles choses. Voir la fin de la section 3.1 pour plus de détails.

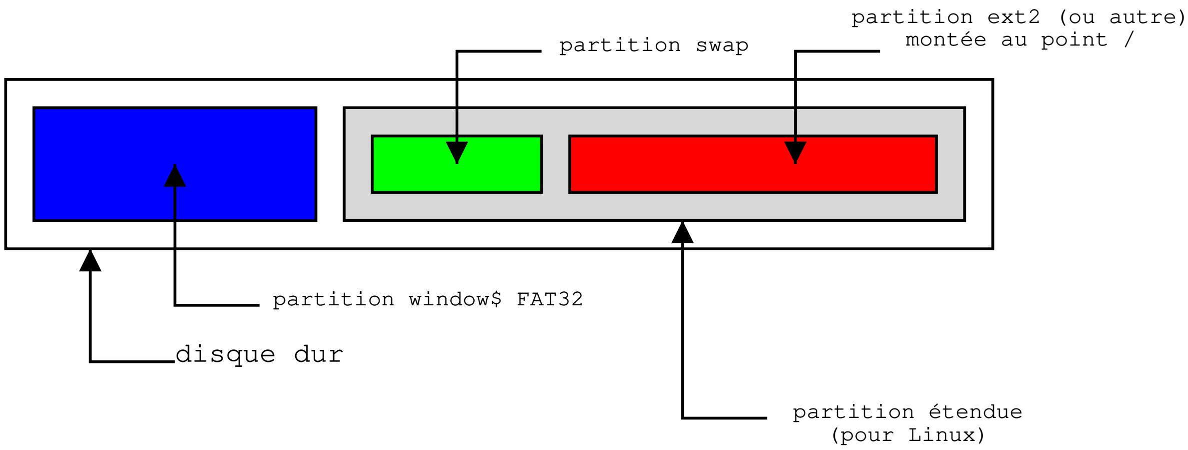
2.8 Re-partitionner le disque dur, c'est enfantin !

Eh oui, c'est enfantin ! Tout du moins si l'on connaît les bases incontournables. Évidemment, si tu te lances dans le re-partitionnement de ton disque dur à l'aveuglette, comme un fou, et en plus avec les mauvais logiciels, c'est foutu ! Je suppose que pour le moment tu n'as que windows d'installé sur ton disque dur. windows occupe une partition du disque dur de type fat32.

partition : imagine que ton disque dur est une bande rectangulaire, comme sur la figure 2.1. Le disque dur peut être partagé en plusieurs parties, afin d'installer plusieurs systèmes d'exploitation ou encore afin de séparer physiquement certains répertoires des autres. Par exemple, windows se loge dans une partition de type fat32, et Linux dans une partition ext2, ext3, ReiserFS..., mais si tu installes Mandrake, je te conseille d'utiliser une partition ReiserFS (voir le chapitre *9*). En effet, ce système de fichiers est disponible d'un simple clic lors de l'installation, et il est bien plus robuste et performant que ext2. Par exemple, il résiste bien mieux aux coupures brutales de courant. Le disque dur peut contenir au maximum 4 partitions, appelées « primaires ». Comme la plupart du temps on veut en utiliser plus, on a créé des partitions dites « étendues » qui peuvent, elles, accueillir d'autres partitions dites « logiques ». Tout de suite un exemple : regarde la figure 2.1 La par-

[3] Je plaisante au fait.

FIG. 2.1 – Représentation du disque dur



tition windows fat32 (en bleu) est une partition primaire. La partition grise nommée « partition étendue » est dite étendue parce qu'elle en accueille plusieurs autres. A l'intérieur de la partition étendue, on a les deux partitions utiles à Linux (on peut en vouloir plus de deux) : la partition swap (partition d'échange des données) et la partition du système de fichiers, partition sur laquelle seront stockés programmes et données personnelles. Linux occupe *tout le temps* toute la mémoire RAM afin d'optimiser les performances. C'est vrai, la RAM est faite pour être utilisée et pas pour rester libre. Si Linux a besoin d'encore plus de RAM alors il va utiliser la partition swap (comme un écrivain utilise du papier de brouillon) pour travailler, mais ce sera un peu plus lent que de la vraie RAM. windows utilise, lui, un fichier d'échange (swap en anglais) dans la racine de C :, fichier qui fait près de 300 Mo chez moi (j'ai choisi cette valeur). Sous Linux je me suis contenté d'une partition swap de 128 Mo. Comme par hasard, Windows rame avec 300 Mo et GNU/Linux ne rame pas avec seulement 128 Mo. Je ne sais pas si le fait que l'ordinateur rame vient du swap, mais Linux est plus économe... tout en étant plus performant.

Donc en gros tu dois réduire la partition windows (qui au début occupe 100% du disque) jusqu'à laisser assez de place pour Linux. Cette étape se fait lors de l'installation de Linux (je décrirai plus en détail l'opération dans le chapitre 9). Mais attention ! L'erreur à ne surtout pas faire est d'oublier de défragmenter la partition windows. En effet, comme windows n'est pas parfait (eh oui encore...) il se fragmente. C'est-à-dire : les données occupent *tout* le disque dur au lieu de se tasser au début. Donc, si tu réduis la partition windows sans avoir défragmenté le disque dur, tu perdras toutes les données qui seront remplacées par Linux. Solution : DÉFRAGMENTER WINDOWS AVANT DE COMMENCER À INSTALLER LINUX. Je sais, cette opération est longue, elle prend jusqu'à 10 heures (et encore ça dépend de la taille du disque), mais elle est obligatoire, sinon on perd ses données.

Pour repartitionner ton disque dur, il *suffit* de suivre les instructions qui te sont données lors de l'installation de Linux. Ne te lance pas bêtement (comme je l'ai fait la première fois...) dans l'opération avec un utilitaire windows ! Presque tous les programmes de partitionnement du disque sous windows sont incomplets (il en existe bien sûr de très bien faits) mais de toute façon, sous Linux, ils sont tous bons. Alors fais confiance à Linux pour repartitionner ton disque dur. Nous verrons cette étape primordiale, lors de l'installation type de Mandrake, chapitre 9.

Une petite précision tout de même pour les utilisateurs de partitions NTFS[4] : sachez que réduire une partition NTFS n'est pas encore très stable sous Linux, et qu'il vaut mieux passer par un outil windows. En fait, les spécifications du NTFS sont tenues secrètes par Microsoft et personne ne peut écrire de programme qui fonctionne bien à cause de ça.

2.9 Choisir entre windows et Linux à l'allumage

Très simple ! Pendant l'installation de Linux, tu seras amené à une étape qui consiste à installer un utilitaire appelé « chargeur de démarrage » (ou « boot loader » en anglais). C'est un petit programme qui va te laisser le choix, sous forme d'un menu, entre les différents systèmes d'exploitation (windows ou Linux dans ce cas). Les deux chargeurs de démarrage les plus connus sont « Lilo » (LInux LOader) et « Grub ». En gros, tu allumes ton ordinateur. Juste après, le BIOS (puce implantée sur la carte mère) fait plein de trucs (test de la mémoire, pendant lequel tu vois défiler un compteur, test du matériel, etc...). Ensuite le BIOS lit le premier secteur du disque dur, appelé « MBR » (Master Boot Record). Si tu regardes le dessin précédant, le MBR se trouverait tout à gauche du rectangle extérieur, même pas dans le rectangle bleu (le MBR n'a rien à voir avec la partition windows). C'est dans le MBR qu'est stocké le chargeur de démarrage. Je te conseille Lilo, surtout avec la Mandrake, car il te fournit un menu sublime, avec une belle image de

[4] utilisée par windows XP, windows NT...

bienvenue. Donc, plus de souci, au démarrage de l'ordinateur, pour le choix du système d'exploitation : si tu choisis windows dans Lilo, windows va démarrer, alors que si tu choisis Linux... Linux va démarrer ! Bien sûr, il est probable que d'autres options figurent, à part windows et Linux, mais c'est en cas de panne, pour réparer Linux, ou pour charger une disquette de démarrage, rien de bien important.

2.10 Et mes données ?

Pas de problèmes ! Linux sait très bien accéder à la partition de windows ! J'écoute mes musiques au format MP3[5], regarde mes images depuis Linux sur ma partition windows car, en tant que grand paresseux, je n'ai pas eu envie de sauvegarder mes données puis de les copier sur la partition Linux. Par défaut, la plupart du temps, tu peux accéder à ta partition windows en tapant « `cd /mnt/windows` » ou « `cd /mnt/dos` » depuis une console. Bien sûr, c'est dans le chapitre 5 que nous verrons tout ça plus en détail.

Encore un coup de pied dans Microsoft : je constate que mon disque dur va plus lentement et fait beaucoup de bruit lorsque je vais justement chercher mes fichiers sur la partition windows. La fat32 se fragmente vraiment trop.

2.11 Mais je refuse de taper les commandes !

Alors là, c'est un point important. Beaucoup de gens sont persuadés qu'utiliser Linux revient à retourner dix ans en arrière et utiliser Dos. Non, non et non ! Ce n'est pas parce que les plus expérimentés aiment manier l'ordinateur en tapant des commandes dans une fenêtre que tout le monde doit le faire. Avant d'en dire plus, je t'envoie sur la page `http://www.kde.org` et clique sur « screenshots » pour voir les captures d'écran. KDE est un environnement graphique pour GNU/Linux, et tu vas voir comme c'est convivial !

Oui, utiliser des interfaces graphiques est très possible, et quasiment tous les domaines d'application sont couverts. L'avantage d'utiliser les commandes est de tirer un bien meilleur parti du système, mais il ne faut pas te sentir obligé ! Le jour où tu te sentiras capable, tu t'y mettras (je pense que tout le monde s'y met un jour ou l'autre).

J'ai consacré la section 4.3 à ces environnements graphiques.

[5]plutôt du Ogg Vorbis, format audio libre, contrairement au MP3.

Chapitre 3

La philosophie de GNU/Linux

Sommaire

Et quelle philosophie ! GNU/Linux, c'est le monde sain et pur du logiciel libre, dans lequel l'argent et le commerce ne salissent pas l'informatique. C'est vrai : ne t'es-tu jamais demandé pourquoi tu devais payer un logiciel ? Ou plutôt, si je reformule ma question, pourquoi un programmeur professionnel devrait-il être payé à chaque fois que quelqu'un achète son programme *pour un seul et unique travail, effectué une seule fois* ! Je vais te démontrer, si tu n'en es pas encore convaincu, que l'avenir de l'informatique se tourne vers le logiciel libre. Nous allons voir quelle est la philosophie de Linux parce que c'est une étape qui me semble importante lorsqu'on débute sous Linux. Ce serait *vraiment* dommage qu'un débutant démarre sous Linux sans connaître le magma qui s'anime par dessous ! Ce serait d'autant plus dommage qu'il ignore ce qu'est le logiciel libre, qu'il pense aux stations service quand on lui parle de GPL, ou encore qu'il croie, comme le laissent entendre trop de revues, que Linux est un *simple* concurrent de windows. Linux est bien plus que ça... mais je vais trop vite ! Commençons par le début.

3.1 Plus qu'un ensemble de logiciels...

Si tu as lu la définition d'une distribution Linux, tu as déjà une petite idée. Linux, c'est vrai, peut être considéré comme un ensemble de logiciels fournis avec une version (la dernière le plus souvent) du noyau Linux. On verra ce qu'est le noyau plus tard. Mais Linux, c'est bien plus que ça. C'est toute une façon de penser, une vision (nouvelle) de l'informatique. C'est assez simple, mais pour mieux comprendre, je pense qu'il faut d'abord distinguer les différentes licences « protégeant » les logiciels. Il existe :

- **les logiciels commerciaux** : ils sont distribués par des commerces. par exemple : Microsoft windows, Adobe Photoshop, 3D studio max, steinberg Cubase, etc., bref, tout ce qui va avec windows la plupart du temps.
- **les sharewares** (ou partagiciel, pour faire plaisir aux anglophobes) : en général ce sont des particuliers doués en programmation ou bien un petit groupe de programmeurs qui conçoivent les sharewares. Ce sont des petits programmes qui sont proposés en téléchargement, distribués dans les magazines, que tu peux essayer pendant un certain temps, et le jour où tu payes l'auteur, il te vire la limite temporelle et/ou la limite en fonctionnalités. En fait, l'auteur n'a travaillé qu'une fois, mais trouve normal d'être payé des dizaines, centaines ou millions de fois. Exemples de sharewares : winzip, winrar, ftp expert, web expert...
- **les freewares** (ou graticiels, encore un mot vraiment stupide !) : ils sont gratuits, complets, et leur auteur ne demande pas d'argent (une

carte postale !). Cependant, ce ne sont pas des logiciels libres *parce que les codes source des freewares ne sont pas accessibles*. J'insiste car de nombreux magazines font souvent cette erreur et les gens sont bernés.

code source d'un logiciel : un logiciel peut exister dans plusieurs états. Il est soit compilé, soit non compilé, et dans ce cas on a le code source. En fait, un programmeur est un humain, et un humain fonctionne grâce à un langage évolué (comme le français, l'anglais...). Le langage machine est vraiment complexe à déchiffrer pour un humain, alors on a inventé des langages évolués, comme le C, le C++, le langage Ada, le Pascal, le Fortran, (me permets-tu de citer le Basic ?). Ainsi, nos amis les programmeurs tapent des instructions destinées à l'ordinateur, dans un langage qui ressemble à de l'anglais de cuisine. Ce fichier texte qui contient ce langage est appelé le code source du programme. Ensuite, on compile le code source, grâce à un autre programme qui va transformer le code source compréhensible par l'humain en langage machine compréhensible par l'ordinateur. Et alors là, plus question d'essayer d'y comprendre quelque chose. C'est une suite de caractères bizarres. Pour modifier le logiciel, on reprend notre code source, on le modifie et on le re-compile. Pour donner une image, le code source est à un logiciel ce qu'une recette de cuisine est à un plat bien préparé.

– **les LOGICIELS LIBRES :** comme les freewares, ils sont gratuits, et complets. Certaines personnes ou entreprises commercialisent des logiciels à base de logiciels libres, mais personne n'est obligé d'acheter ce produit payant. En plus, *on a accès à leur Code Source*, donc on peut les modifier, ce qui est une grande joie. On peut trouver le code source dans la distribution Linux, ou bien en tous cas sur Internet. On dit que les logiciels libres sont « Open Source » (source ouverte, plutôt dans le sens de accessible). Même si tu te dis : « moi je ne sais pas programmer donc je m'en fous d'avoir le code source », alors je te réponds « mais sache que si le code source est accessible, des gens qui eux savent bien programmer feront évoluer le logiciel pour qu'il corresponde à leurs envies et cela à la vitesse de la lumière, car imagine le nombre de bons programmeurs amateurs (ou même professionnels) sur terre qui programment pour leur plaisir ! Tu profiteras toi aussi de l'Open Source ».Ne plus confondre libre et gratuit : un logiciel libre est gratuit mais un logiciel gratuit n'est pas forcément libre. Attention : ne pas penser non plus que logiciel libre signifie logiciel de mauvaise qualité ! Au contraire, nous allons le démontrer. Les logiciels libres sont légalement régis par une licence : la GPL (General Public license) qui écrit noir sur blanc les caractéristiques d'un logiciel libre. Et cela ne s'arrête pas à l'open source : quand tu utilises un logiciel libre GPL (la majorité des logiciels qui fonctionnent sous GNU/Linux) tu peux en faire *ce que tu veux*, alors que certains logiciels, même open source, ne te le permettent pas.

Tu vois donc que le logiciel libre, contrairement aux freewares, sharewares et logiciels commerciaux, évolue et s'améliore très vite ! Excuse moi, mais as-tu vu d'énormes différences entre winzip 6 (version d'il y a 3 ans) et winzip 8 (version actuelle) ? En tout cas, par rapport aux logiciels libres, la différence est de la taille d'un grain de poussière. Le plus scandaleux est que, par derrière, la société qui édite winzip nous fait croire que son logiciel est le meilleur (le format zip est vraiment nul : essaie le bz2 le format en vogue sous Linux) et qu'il apporte tout un flot d'améliorations. Tout ça pour l'argent encore une fois[1]. Si tu veux des explications sur la compression sous GNU/Linux, va voir la section 8.2 ou encore la section 5.1.3.

Autrement dit, il faut que tu prennes conscience qu'il y a deux types d'entreprises qui produisent des logiciels commerciaux : celles qui te prennent pour un porte-monnaie sur pattes et celles qui fournissent un travail à la hauteur du prix. Par exemple, Adobe est à mes yeux une entreprise exemplaire, car :

- elle a participé à l'élaboration de standards de grande qualité, comme le PostScript, le PDF
- elle documente abondamment ces derniers pour que tout le monde puisse s'en servir
- elle fournit des logiciels qui sont tous de très bonne qualité
- lorsqu'elle sort une nouvelle version d'un logiciel, cela vaut le coup d'effectuer la mise à jour car les nouvelles fonctionnalités sont au rendez-vous

Par contre, certaines entreprises (dont je tairai pour une fois le nom) abusent financièrement de leurs clients, c'est-à-dire :

- elles mettent sur le marché des produits non terminés
- elles en profitent pour vendre une nouvelle version qui corrige les imperfections de la version précédente. Bien sûr la nouvelle version (qui devrait être gratuite, puisque c'est finalement de la maintenance) coûte aussi cher que la version précédente.
- de plus, la nouvelle version contient aussi des imperfections, qui seront prétexte à vendre encore une version ultérieure... on ne s'arrête plus.
- l'entreprise en question fait en sorte que si le client n'achète pas la nouvelle version, il n'est pas en mesure de collaborer avec ceux qui l'ont

[1] L'argent pourrit les gens, j'en ai le sentiment... l'argent appelle l'argent, le bénef moi j'l'encaisse, l'laisse rien dans la caisse.

achetée. C'est pourquoi tout le monde est obligé de suivre le mouvement de foule. Cette entreprise ne définit aucun standard (contrairement à Adobe par exemple), seulement des normes qui changent sans arrêt.
– de plus elle ferme complètement ces normes, et il faut payer pour avoir accès au décodeur qui les déchiffrera.

On voit bien que toutes les entreprises ne sont pas mauvaises. Par contre...

Un ami m'a dit : « si aucune entreprise ne soutient un logiciel, il ne peut pas se développer ». Je lui ai ri au nez, car l'entreprise est en quelque sorte un frein : si une entreprise emploie 10 programmeurs, alors seuls ces 10 programmeurs pourront s'occuper du projet. Alors que si *tous les programmeurs du monde entier* peuvent s'y mettre, c'est pas mieux peut-être ? De plus, les logiciel libres étant indépendants des entreprises, si on les abandonne, d'autres les reprennent, tout cela étant prévu par la loi et par la philosophie Linux. D'ailleurs il n'est pas rare qu'un auteur abandonne un freeware ou un shareware et que personne ne le reprenne, faute de code source. J'ai moi-même commencé au début de l'année à programmer un logiciel d'apprentissage du solfège en visual basic (honte à moi, je sais, pardonnez-moi, j'étais un pauvre gars perdu et trompé par la publicité). A cause de mes études, je n'ai plus eu le temps de le finir. Comme je ne voulais pas non plus donner mon code source (il faut dire que dans cet état d'esprit renfermé, on se sent vulnérable en donnant son code source), personne n'a pu finir. Aujourd'hui, c'est une épave : quelques exercices fonctionnent bien, mais les autres sont à peine ébauchés. Lorsque j'ai vu le logiciel Solfège (`http://www.solfege.org`) pour Linux, j'ai sauté de joie ! C'est exactement ce logiciel que je voulais programmer. Alors je l'ai téléchargé et ça m'a coupé l'envie de continuer le mien. J'espère que tu comprends l'extraordinaire sentiment que j'ai ressenti en voyant avec quel enthousiasme les gens participaient au projet, comme ça, pour avoir *le logiciel dont ils rêvent*, tout simplement. En plus, si tu regardes la liste des changements, tu peux voir qu'en 2 mois le logiciel a évolué en doublant ses exercices, ce que j'aurais fait en 6 mois ou plus !

> Le monde du logiciel libre est un champ de fleurs agréables à regarder, qui viennent d'éclore il y a un mois et qui n'ont pas fini de se développer, mêlées à des graines qui vont bientôt sortir du sol, avec une force immense, si grande qu'elles peuvent casser une autoroute si jamais elles décident de pousser dessus. Le monde du logiciel commercial dirigé par Microsoft est un champ comportant une dizaine de très belles fleurs, toutes les autres étant déjà en train de se faner. Les jeunes fleurs ne grandissent jamais jusqu'au bout : soit le jardinier en a marre de les arroser et les abandonne, soit les grosses fleurs les dévorent, empêchant ainsi toute nouvelle

race de se développer.

Grâce à l'état d'esprit du logiciel libre, on obtient des logiciels proches des utilisateurs. Mais pourquoi donc ? Parce que ce sont les utilisateurs qui font les logiciels ! Tout le monde peut participer à un projet, ne serait-ce que pour donner son avis ou ses désirs. Sous windows, ce sont des entreprises qui programment, en répondant à un cahier des charges, celui-ci établi non pas par les futurs utilisateurs, mais par... un bureau d'études, qui va regarder plusieurs critères (par exemple combien les gens sont prêts à payer pour un tel logiciel) mais pas assez les attentes des utilisateurs. Ce système est un véritable paquebot très lent à manœuvrer car il faut sans cesse investir de l'argent. C'est pour cela que l'on voit plusieurs versions quasiment identiques d'un logiciel commercial avant de voir arriver les innovations majeures tant attendues. Les éditeurs profitent de la crédulité des clients pour leur extorquer autant d'argent que possible, et au dernier moment ils sortent le produit rêvé. Bilan : le client a payé deux ou trois fois le même produit ! Sous Linux, les programmeurs, ce sont les utilisateurs, et vice versa. En adoptant Linux, tu es sûr d'avoir des logiciels fonctionnels, qui rendent *vraiment* service.

Tu vois bien maintenant, si tu m'as compris, que Linux représente toute une façon de penser, et cette façon de penser n'existe que dans le monde du logiciel libre.

3.2 Pourquoi window$ mérite un dollar à la place du « s » ?

Le monde de windows, c'est-à-dire Microsoft ainsi que tous les autres éditeurs de logiciels aspirés dans le sillon de Microsoft, est une pompe à fric. En effet, les éditeurs de logiciels travaillent pour... gagner de l'argent, et non pas forcément pour satisfaire leurs utilisateurs. Je m'explique :

Pourquoi les hotlines (numéro de téléphone que l'on appelle pour se faire dépanner) ? Parce qu'elles rapportent. Sous Linux, quand il y a un bug, j'envoie un gentil e-mail à son auteur et il me répond gentiment. Coût de la manœuvre ? 1 € de connexion et encore j'en ai profité pour regarder les news sur `http://www.linuxfr.org`. C'est bien mieux que « oui bonjour monsieur, que puis-je faire pour vous » ou encore que « oui votre demande a bien été enregistrée veuillez patienter », et pour finir avec un « hmm... je crois bien que la meilleure solution serait de formater votre disque dur puis de réinstaller windows. Autrement il y aura toujours des problèmes ». Ces pauvres hommes-micro-casques souffrent dans leurs box, comment peuvent-ils être efficaces ?

A croire que certains utilisateurs de windows sont contradictoires : ils se plaignent de problèmes impossibles à résoudre, que la hotline ne répond pas, qu'elle est occupée... qu'ils passent à Linux ! C'est tellement plus libre...

Évidemment, sous windows les programmeurs donnent quelquefois leur e-mail, mais sous Linux, c'est *systématique*.

Donc pour répondre à la question (c'est moi qui l'ai posée !), je dirai, au cas où tu ne l'aies pas encore compris, que le fait de mettre un dollar à la place des « s » dans les noms de logiciels ou de marques me permet de bien montrer, du premier coup d'œil, combien ils sont friands d'argent. Sous Linux, les programmeurs programment « non pas parce qu'ils sont alléchés par l'appât du gain, mais juste parce qu'ils en éprouvent le besoin ».

Une autre question me vient à l'esprit : « pourquoi quand on baigne dans le monde windows on n'entend pas parler des logiciels libres ? » C'est dommage, car n'importe qui, s'il possède un compilateur, peut compiler n'importe quel logiciel libre sous windows, puisqu'on dispose des sources[2] ! Les magazines pro-windows pourraient distribuer des logiciels libres, mais non, on n'entend parler que de sharewares, freewares, logiciels commerciaux, de piratage... Certains en distribuent mais oublient toujours d'expliquer ce qu'est un logiciel libre. Si on n'entend pas parler du logiciel libre à grande échelle, c'est bien la preuve que ce monde est dominé par des bourriques de commerciaux, qui privilégient le logiciel commercial pour satisfaire leurs intérêts et bâillonnent le logiciel libre car il ne leur rapporte rien. Ils privilégient le commerce au détriment de l'informatique, c'est vraiment déplorable.

Bill Gates exprime souvent ses idées au sujet du futur de l'informatique, notamment dans les écrans publicitaires. Il pense que l'informatique doit être à la portée de tous, qu'elle doit venir nous faciliter la vie jusque dans notre poche. Tout va bien, la vie est belle. Le problème est que Microsoft est en train de s'acaparer les protocoles de communication qui gèrent Internet, protocoles qui ont toujours été libres et documentés. De plus Microsoft est en train d'inventer de nouveaux « standards » totalement obscurs, et ce n'est dans l'intérêt de personne, sauf lui ! Les gens ne se rendent pas compte qu'en ayant une confiance aveugle en Microsoft, ils se font du mal. En effet, lorsque Microsoft aura acquis tous les protocoles et qu'il sera le seul à pouvoir les décoder, il sera le maître du monde des télécommunications. Il pourra tout contrôler, la vie privée de n'importe qui... tu comprends mieux maintenant ? Microsoft a donc logiquement qualifié GNU/Linux d'un « cancer qu'il faut éradiquer au plus vite » car évidemment GNU/Linux nuit à l'hégémonie de Microsoft ! C'est presque comique... Bien sûr tout cela n'existe pas encore,

[2]Certains logiciels nécessitent d'être retravaillés pour passer d'un système à l'autre, mais lorsqu'on a les sources, rien de plus simple.

mais pense aux ActiveX, ASP, par rapport aux plugins Netscape et au PHP qui eux sont libres et documentés. En achetant Microsoft, on se rend complice de tout ce qui pourrait arriver, et en plus ce n'est pas dans notre intérêt. Il faut vraiment y penser, même si on se sent quelque peu obligé d'utiliser des produits Microsoft. D'ailleurs ce sentiment d'obligation fait partie de la stratégie de Microsoft : as-tu réussi à acheter un PC sans payer windows ? Non. Le vendeur perdrait de l'argent.

As tu envie de rester dans un monde pareil ? ? ?[3]

3.3 À propos du piratage...

Le piratage, c'est mal. Vraiment. Mais pas pour les raisons que l'on te donne habituellement. Je ne vois pas le piratage du même œil que tout le monde, et j'ai trouvé d'autres raisons de ne pas pirater. Les médias, comme par exemple les journaux télévisés ou la radio, nous inondent d'histoires de piratage, et de ce fait la majorité de la population a une image déformée du piratage. Je me suis fixé comme objectif dans cette section d'éclaircir ce point.

Tout d'abord, voici les raisons *officielles* de ne pas pirater :

- c'est mal, car on vole le travail des programmeurs. Je réponds que le programmeur n'a pas à être payé plusieurs fois pour un seul et même travail.
- on risque d'avoir des amendes et d'aller en prison. Cette répression est bien belle, mais je vais te montrer que la solution n'est pas de passer entre les mailles laissées grandes ouvertes du système judiciaire.
- il faut dire que le fameux BSA (Buisness Software Alliance) est une organisation montée par les géants du logiciel afin de se protéger contre le manque à gagner. Cette organisation diffuse parfois des spots publicitaires incitant à « légaliser » ses logiciels en les achetant plutôt que de garder les éventuelles versions copiées illégalement que l'on serait en train d'utiliser. Mais la BSA n'est rien d'autre qu'une auto-défense pour Microsoft, qui en est le principal dirigeant.

Voici maintenant MES raisons de ne pas pirater :

- À quoi bon pirater des logiciels pourris par l'argent ?
- À quoi bon pirater des logiciels qui plantent sans arrêt ?

[3]Il faut lire le livre « Le hold-up planétaire, la face cachée de Microsoft » le Roberto DiCosmo, dans lequel il parle du piège que Microsoft a tendu au monde entier. Un très bon livre présenté sous forme d'interview : c'est Dominique Nora qui pose les questions. En plus il n'est vraiment pas cher !

Ne me dis pas que windows ne plante jamais, sinon je te dirai d'utiliser trois logiciels à la fois et tu verras ! Pendant la rédaction de ce document, j'ai trois ou quatre logiciels qui marchent en permanence : j'imprime de la doc, je compile un programme et je tape au clavier ce texte. Depuis ce matin le système n'a pas bronché et réagit toujours au quart de tour. Mon ordinateur ? Un Pentium 200 MMX avec 64 Mo de RAM. Eh oui, c'est tout. Et je connais un ami, Samuel, qui a un Pentium 75 avec 48 Mo de RAM et il fait pas mal de choses aussi.
Si tu m'assures que j'exagère de vouloir utiliser plusieurs programmes à la fois sans que ça plante, je te rétorque que ton ordinateur est incroyablement plus puissant que celui qui a envoyé des Hommes sur la Lune, et qu'il n'est même pas capable de faire fonctionner une suite bureautique, un logiciel graphique et naviguer sur Internet. De qui se moque Microsoft ? Encore une fois, pourquoi t'acharnerais-tu à le défendre ? Tu te sens à l'abri en faisant confiance à une soi-disant entreprise « reine » de l'informatique ? Allons... un peu de perspicacité. Malheureusement, trop de gens défendent Microsoft, on ne sait pourquoi.

planter : ça ne t'es jamais arrivé de travailler, de surfer sur Internet ou je ne sais quoi, et que d'un coup tu voies un écran bleu signé windows ? Ça t'est obligatoirement arrivé, à moins que ton ordinateur ait trois jours et que tu n'aies rien installé comme programmes (à ce moment là à quoi te sert ton ordinateur ? !). Eh bien voilà, ça veut dire que windows a planté, et tu peux pester sur windows car sache que sous Linux c'est vraiment très rare.

– À quoi bon pirater des logiciels qui sont développés par des gens qui ne pensent pas à nous (ou plutôt quand ils pensent à nous c'est à notre argent), voire qui nous prennent pour des imbéciles ! ? Preuve à l'appui, nous allons prendre le cas de... Microsoft pour changer. Je peux te dire qu'ils excellent dans un domaine très pointu, domaine qui nécessite une énorme dose de mesquinerie : prendre ses propres clients pour des abrutis finis. Tu connais sans doute aMath, un petit logiciel d'extension de word, qui est en fait une super barre d'outils faite pour faciliter les saisies d'équation (quand on ignore que LaTeX existe... voir la section 7.1) car il faut dire que l'« equation writer » de Microsoft est quelque peu lourdingue.
Sur sa page web, Marc Guillemot, qui développe cette barre d'outils depuis plus de quatre ans, a demandé aux utilisateurs de aMath d'envoyer un email (dont il fournit le modèle) à notre bon vieux Microsoft, email qui demande que l'equation writer de word soit amélioré. En 1997, Microsoft a reçu 41 messages de profs de maths ou de physique

et a odieusement répondu, d'une part, que word était utilisé par un trop large public pour que ce soit la peine d'approfondir ce module de saisie d'équations, mais le plus horrible est que le directeur de la programmation (ou autre, je n'ai pas su traduire sa fonction en français) a expressément demandé que tout envoi d'emails de ce type cesse car ils ne changeraient rien. Le côté poilant de la chose est que Microsoft a envoyé 2 ou 3 emails avant de comprendre l'objet de la demande, qui était très bien expliqué !
Est-ce là une manière de remercier tous ces utilisateurs de word ? Le côté attristant est que ce cher Marc Guillemot s'accroche à adapter son programme à chaque version de word, et ce depuis 4 ans ! Est-ce que ce genre de réponse carrément incorrecte a déjà existé dans le logiciel libre ? Jamais ! Je peux même témoigner du contraire : j'utilise le logiciel TiLP sous Linux, qui permet de transférer des programmes du PC vers ma calculatrice Texas Instruments, et vice versa. J'ai envoyé une dizaine d'emails au programmeur Romain Liévin et il m'a toujours répondu très rapidement (dans les deux jours maximum) et donné des réponses précises, et oh... surprise ! La version suivante du logiciel corrige un bug que je lui avais signalé (loin de moi l'idée de me vanter).
Si tu veux voir par toi-même les emails échangés entre M. Guillemot et Microsoft, va voir à l'adresse `http://www.amath.net` et clique en bas à gauche de l'écran pour accéder à la bonne rubrique.
- À quoi bon pirater des logiciels qui n'évoluent pas ?
  Franchement, quelle est la *vraie* différence entre office 97 et office 2000 au niveau des fonctionnalités ? Aucune ! quelques détails changent comme les menus semi-ouvrables, ou bien un bouton par document dans la barre des tâches, mais pas dans le fond.
- À quoi bon pirater des logiciels qu'on t'oblige à mettre à jour sinon t'es largué ?
  Si tu n'as pas word 2000 alors tu es hors jeu. D'ailleurs Microsoft fait exprès d'interdire la compatibilité descendante dans ses logiciels, c'est-à-dire que word 2000 peut lire les fichiers de word 97 mais pas l'inverse. Bien sûr, on peut exporter au format word 97 depuis word 2000 mais on perd toujours des informations. Pourquoi ne pas avoir prévu un seul format qui fait tout ? Il n'y a aucune raison de changer de format de fichier quand il n'y a pas d'améliorations. Sous Linux, c'est très très très rare. Je prévois l'avenir : office xp va faire pareil, sauf si Microsoft a enfin compris. Imagine une entreprise qui fonctionne encore sous Office 97 ! Un jour ou l'autre elle va devoir passer à Office XP, mais surtout elle va devoir passer à la caisse. Désolé messieurs dames les commerçants, vous êtes le dernier maillon (faible) d'un système abominable. Faibles

car vous ne pouvez plus faire autrement que de vendre des PC avec windows, sous peine d'être ruinés par Microsoft.
– À quoi bon répéter encore une fois « à quoi bon » ?

Ce serait, au contraire, montrer aux marques qu'on aime leurs logiciels commerciaux que de les pirater. Mieux vaut utiliser des logiciels libres. Le piratage est pour eux une fierté : « ha ha ha!!! notre logiciel est si bien qu'on va jusqu'à le pirater ! », et le pirate est en quelque sorte à leur service pour diffuser leur(s) programme(s). Si les jeunes utilisent des logiciels commerciaux piratés, quand ils seront adultes, ils les achèteront pour travailler avec, car professionnellement on ne peut plus se permettre de pirater, les contrôles sont trop fréquents. Par exemple, un gamin passionné d'infographie 3D va se procurer « 3D studio max » sans l'acheter, qui est normalement vendu à plus de 450 €, et apprendre à s'en servir. Plus tard, pour réaliser des films 3D professionnellement, il va continuer à l'utiliser, et, afin de rester dans la légalité, le fera acheter par son entreprise ! Et hop, on passe à la caisse ou bien on va en cabane. C'est pas plus compliqué. Mais cette personne ne pensera pas à l'alternative qu'offre le logiciel libre.

On assiste à une certaine *incitation* au piratage. Les revues d'informatique windowsiennes s'en donnent à cœur joie de distribuer des démos de ces logiciels. Les éditeurs de logiciels savent très bien que pirater leurs démos est très facile. Donc on peut dire qu'ils laissent faire. Preuve à l'appui : 3D studio max, encore lui, en est aujourd'hui à la version 4. Jamais on n'a trouvé de démo de la version 3.1 (la version précédente). On remarquera l'attrait qu'exerce le changement de numérotation sur le consommateur, malgré que les nouveautés entre les deux versions ne soient pas extraordinaires. Sous Linux, les numéros de version des logiciels sont significatifs : quand on passe de la version 1.xx à la version 2, c'est qu'il y a du *réel* changement. Dès que la version 4 est sortie, on a pu trouver des démos dans les magazines, démos qu'il était très facile de transformer en version complète, pour peu que l'on fouille un peu sur Internet...

De toute façon, je me dis que même si les démos sont facilement craquables (= transformables en versions complètes), il vaut mieux utiliser du logiciel libre ! Par exemple, Blender et Moonlight Atelier sont très bien pour la 3D sous Linux. Mince, je devais attendre le tableau comparatif page 125 ! Tant pis.

Officiellement, on montre à la télé que les pirates sont arrêtés tels des assassins, sans pour le moins expliquer le fond du débat (ce que je suis en train de faire), mais par derrière les pirates sont pour ces entreprises de bons futurs clients sérieux. Paradoxalement, le terme « pirate » a été inventé par les éditeurs de logiciels, pour accentuer le côté sanguinaire chez celui qui copie les logiciels commerciaux. Si tu « pirates » des logiciels commerciaux, c'est que tu as compris que ce n'était pas la peine de payer. Tu méprises les industriels du logiciel car tu penses qu'ils abusent de ton argent. Si on te propose un système d'exploitation dont les créateurs se réjouissent si tu le copies, tu dois donc sauter dessus et l'utiliser. Sinon c'est que tu fonctionnes paradoxalement : tu restes avec tes « ennemis ». Ne te mets pas à critiquer GNU/Linux car ce serait pire : tu insulterais tes propres amis !

Ce qui est bien sous Linux, c'est qu'on ne peut plus pirater les logiciels puisqu'on a le droit de les copier et de les diffuser. Ils sont faits pour ça ! GNU/Linux est l'antidote contre le piratage, ou contre les risque d'amendes. Pensez-y, messieurs les patrons d'entreprises, directeurs d'établissements ou autres : Linux vous fera économiser des sommes non négligeables ! Motivez-vous : de plus en plus d'entreprises, d'administrateurs système,... utilisent Linux ! Car utiliser du Microsoft vous fourre, comme vous l'avez constaté, dans un guêpier infernal.

3.4 Pour l'éducation

Dans l'Éducation Nationale c'est honteux, oui m'dame, hon-teux. Les professeurs qui nous enseignent l'informatique sont censés nous apprendre l'informatique et c'est tout. Ils ne méritent pas du tout d'être obligés de débourser de l'argent pour cela, et n'ont pas plus le devoir de faire grandir Microsoft en lui livrant toute une fournée d'élèves-moutons bien éduqués à l'avoine Microsoft. Aux États-Unix c'est pire : Microsoft distribue gratuitement des copies de ses logiciels aux élèves à la sortie des écoles pour qu'ils restent dociles. Je ne sais pas ce que tu en penses, mais ces méthodes me font froid dans le dos. Que le logiciel libre se diffuse ! Félicitons l'académie de Grenoble qui va dans ce sens. Au collège et au lycée, on ne doit pas faire de bourrage de crâne en faveur de Microsoft.

Je vais citer plusieurs logiciels scolaires (physique, maths, chimie, mécanique, électronique, etc...), à la section 7.2.

Le même ami que j'ai cité tout à l'heure m'a dit : « à l'école ils ont le devoir de nous apprendre à se servir des logiciels que l'on trouvera sur le marché du travail ». Je lui ai encore rit au nez[4].

Je me permets de céder la parole à un membre de la mailing list à laquelle j'adhère (AXUL, le LUG d'Aix en Provence), qui explique cela très bien et en deux paragraphes seulement :

> [...] Sinon je suis on ne peut plus d'accord avec toi sur le fait que windows n'a rien à faire dans l'éducation. Et d'ailleurs même si

[4] le pauvre garçon !

> Microsoft donnait gratuitement ses logiciels je pense qu'il n'aurait toujours rien à faire dans l'Éducation. En effet je ne pense pas que le but de l'école soit de promouvoir Microsoft ni de fabriquer des consommateurs. Quand j'étais à l'école, les profs nous formaient à décrypter la télévision, à former notre esprit critique sur les journaux télévisés et sur les publicités. Ils essayaient de nous alerter sur les dangers des médias.
>
> L'informatique est un média qui est en train de passer sous le contrôle d'une seule entreprise (il y a déjà un certain nombre de sites internet qui ne sont plus accessibles que par « internet explorer », et ça risque d'empirer). L'informatique nécessite elle aussi un décryptage, et l'école devrait être un lieu privilégié pour en faire l'apprentissage.[5]

Je suis obligé de dire que bien des cours, tous niveaux confondus, du collège à l'université, inculquent aux élèves l'informatique façon Microsoft. Par exemple je me souviens que des anciens collégiens que je connais apprenaient il y a quelques années l'informatique de la façon suivante : il fallait savoir faire certaines choses, et le professeur signait et validait la compétence. Le problème, c'est que les compétences à acquérir étaient ciblées Microsoft. Bien sûr, les noms des logiciels n'étaient pas explicitement écrits, mais on comprenait bien :

- être capable de mettre en gras, en italique du texte
- savoir créer des icônes sur le bureau
- savoir utiliser le menu des programmes de la barre des tâches
- ...

Les professeurs jouent le rôle de démonstrateurs pour les produits Microsoft. Ils n'ont pas l'occasion d'aborder les différentes visions de l'informatique, d'ailleurs eux-mêmes ne connaissent pas le logiciel libre.

Un autre cas semblable : j'ai un ami qui est à l'université et il suit des études de biologie. Il a choisi une option informatique. On lui apprend à se servir uniquement de windows, de word, excel et on lui inculque la loi du logiciel façon, Microsoft. Si bien que la plupart de ces jeunes ne savent même pas qu'on peut penser l'informatique autrement : ils ne connaissent pas le logiciel libre. Ce n'est pas de leur faute puisqu'on leur inculque systématiquement la vision commerciale de l'informatique.

Passons à la suite : il existe des *normes* de formats de fichiers. Par

[5] Il faut absolument lire l'article de Roberto DiCosmo intitulé « Piège dans le cyberespace », disponible à l'adresse : `http://www.dmi.ens.fr/~dicosmo`. Cet article remarquable dévoile la face cachée de micro$oft et propose des solutions pour remédier à son monopole sur l'information.

exemple, un dessin technique a l'extension « *.dxf » et tous les logiciels de dessin techniques dignes de ce nom, commerciaux ou libres, savent utiliser cette norme. Donc, dans une entreprise, rien n'empêche un ingénieur en génie mécanique (par exemple !) de travailler sur un logiciel libre puis de sauvegarder son dessin dans un format standard. Ses collaborateurs pourront ensuite le ré-ouvrir avec le logiciel de leur choix ! Comme quoi dans le monde du travail on n'a pas d'obligation (sauf si aucun logiciel équivalent à celui de l'entreprise n'existe sous Linux, mais ce sont des cas très rares). Essaie QCad pour t'en apercevoir.

Autre point non négligeable : le monde du travail dont mon cher ami me parlait *évolue* ! Dans les entreprises, les patrons optent de plus en plus pour Linux, vraiment ! Linux est en passe de s'imposer sur le marché du travail, les PME ont bien compris ses avantages. Donc le fait d'apprendre à se servir de Linux à l'école serait *une très bonne idée*, qui contribuerait à une meilleure préparation au *futur* monde du travail, qui sera sans doute plus *libre* qu'aujourd'hui.

De plus, si les élèves apprennent Linux, ils n'auront *aucun mal* à utiliser windows car windows est très archaïque par rapport aux interfaces utilisateur de Linux (KDE ou Gnome). Alors que si tu apprends windows tu devras « compléter » tes connaissances notamment au niveau des commandes Linux (ou Unix, ce sont les mêmes). Donc, même si windows ne sera pas tout à fait mort dans dix ans, les élèves d'aujourd'hui sauront très bien l'utiliser en ayant suivi une formation Linux à l'école.

Il existe des tas de logiciels libres en maths, électronique, mécanique, physique, chimie, etc... que l'on peut trouver sur Internet.

Conclusion : il *faut* enseigner Linux à l'école, au collège, au Lycée, et encore plus tard ! Il faut souligner que dans les IUT et les universités Linux est bien plus enseigné que dans le secondaire... et ça changera, coûte que coûte !!!

Chapitre 4

Comment est fait Linux

Sommaire

Ça y est ! On rentre un peu plus dans le vif du sujet, un peu comme un baigneur qui entre dans l'eau froide et qui a l'eau au niveau des cuisses. Fini la philosophie et les idées formidables (j'espère que windows te dégoûte un peu maintenant... un tout petit peu, pour me faire plaisir !), on passe à la pratique. On va maintenant étudier comment marche Linux, son architecture notamment.

4.1 Un système de marbre

Linux est un système multi-avantages !

- **multi-tâche :** le système sait exécuter plusieurs programmes à la fois. En fait, un programme se bloque quand il est *mal programmé*. C'est donc la faute de son auteur, qui n'a pas encore « éliminé tous les bugs ». Normalement un programme qui plante s'arrête sans gêner les autres programmes et encore moins le système ! Linux gère très bien les plantages et si un programme plante le système entier reste de marbre. Sous windows, c'est le contraire (tiens, c'est bizarre !) : un programme qui plante fait planter en cascade tout le reste. On a droit à l'écran bleu, au CTRL+ALT+SUPPR puis, oh ! désespoir, au reset. Avec Windows 2000 ou XP, la gestion des processus en exécution est améliorée, mais bon, outre le fait qu'ils sont très chers, je les ai vus souvent planter à cause de ce genre d'erreurs.

bug : erreur de programmation qui fait planter (se bloquer) le programme. Il ne réagit plus, et devient inutilisable. Dans les années 50, les ordinateurs étaient dans de grandes salles et les cartes électroniques reliées par des fils. Les bugs (cafards, en français) rongeaient les fils et faisaient faire des erreurs. En français on est censé dire « bogue » mais je trouve que ce mot ressemble à un cheveu tombé sur la soupe. On a toujours dit « bug » depuis 20 ans, et maintenant que l'informatique se démocratise, qu'on en parle à la télévision, on devrait dire « bogue » !

- **multi-utilisateurs :** plusieurs utilisateurs peuvent utiliser le système, et chacun peut faire *ce qu'il veut sans risque de tout casser* ! As-tu déjà pensé aux dégâts que peut faire un enfant de 10 ans sur un ordinateur tournant sous windows ? Il pourrait tout effacer ! Rien que ça. Combien de parents ont peur de laisser leurs enfants toucher à leur ordinateur, de peur qu'ils ne détruisent par mégarde leur travaux ? Ces parents s'arment souvent de logiciels de soi-disant « protection » de l'ordinateur, et ne pensent même pas que ces logiciels sont des pansements autour de fuites d'eau ! Sous Linux, tout cela est impossible, car

le système est protégé à la base (voir la suite). De plus, si plusieurs utilisateurs du même ordinateur ont des goûts différents, ils pourront régler leur environnement de travail à leur goût, et chaque utilisateur aura ainsi l'impression d'être unique et isolé des autres. Cela est dû au fait que les fichiers de configuration de tous les programmes sont stockés non pas dans les répertoires système mais dans le répertoire personnel de chacun, sous forme de fichiers cachés. Ce qui fait que chacun peut aisément configurer les logiciels communs à sa façon. ? Microsoft a maintenant permis ce genre de choses avec Windows XP (alors qu'avant il fallait acheter NT pour y avoir droit). Je rappelle quand même que GNU/Linux permet le multi-utilisateur depuis le début, pas besoin d'attendre l'année 2001.
J'apporte un petite précision : il y a autant d'utilisateurs qu'on veut pour un ordinateur et *un super-utilisateur* qui lui peut *tout faire*. Ce dernier ne doit se connecter au système que pour effectuer des opérations de maintenance, et *jamais* pour travailler normalement. Sinon ce serait renier tout le système des permissions et des interdictions qui existe sous Linux. C'est bien ça qui se passe sous windows : chaque utilisateur a le droit de tout faire et même le pire. Et après ils osent dire que windows est le système d'exploitation idéal pour la famille ! Je suis sans cesse demandé chez les gens que je connais car ils ont des problèmes avec leur windows : l'enfant a effacé tel ou tel fichier, leur connexion Internet a fichu le camp...
Je m'adresse plus particulièrement aux parents : vous avez peut-être l'obligation de travailler avec les logiciels bureautique de Microsoft. Bien que Linux dispose de StarOffice (gratuit), qui lit et écrit des documents Microsoft office (300 € à 800 €) si nécessaire, certains hésitent encore. Mais vos enfants n'ont pas cette obligation. Réfléchissez : ce serait une très bonne idée que de leur montrer comment se servir des logiciels libres. Je connais des gens qui le font et leurs enfants s'en sortent très bien.

– **multi-plateformes :** Linux peut fonctionner sur toutes sortes de machines (on dit aussi architectures) différentes. Par exemple, tu peux l'utiliser sur un Macintosh, une grosse station de travail à processeur Alpha ou Sparc, et bien sûr les PC. En fait, ce qui compte, c'est le microprocesseur utilisé par la machine. Les PC peuvent accueillir plusieurs microprocesseurs : les Intel 8086, 286, 386, 486 (plus trop utilisés), puis les Pentium (ou 586), les Pentium 2 (ou 686), les Celeron, les Pentium 3, les Pentium 4 les nouveaux Itanium (64 bits !), mais aussi des AMD ou encore des Cyrix. Les Macintosh étaient basés sur des Motorola 68k et aujourd'hui sur des PowerPC, des G3 ou encore des G4. Il suffit d'obte-

nir une version de Linux compilée pour le processeur que l'on possède, ou la compiler soi-même, mais là il faut s'y connaître un peu plus. Mandrake 8.1 est disponible par défaut pour les 586, donc les Pentium, ce qui veut dire qu'il faut avoir au minimum un Pentium pour s'en servir. Cette version fonctionne aussi avec des AMD, des Pentium 2, 3 ou 4, car tous ceux-là sont plus récents que le Pentium et ont été conçus pour être compatibles avec lui. Si tu n'en as pas (par exemple si tu as un 486) il faudra dénicher des CD de Mandrake compilés pour un 486. Au passage, on trouve une version de Mandrake pour les tous nouveaux processeurs Intel Itanium, fonctionnant avec 64 bits. Linux fonctionnera donc en 64 bits et non en 32 bits si tu as un Itanium avec une version de Linux compilée pour Itanium. Tout ça, encore une fois, pour accroître les performances.
Lorsqu'on lit sur les publicités que windows XP a été optimisé pour les Pentium ou autre, c'est simplement qu'il a été recompilé avec un compilateur dédié à ce microprocesseur. C'est tout. Linux peut donc être « optimisé » pour n'importe quel processeur, même les plus petits, c'est pour cela qu'on le retrouve dans certains appareils comme des montres très sophistiquées ou bientôt des téléphones, réfrigérateurs reliés à Internet (ah le progrès !)... sans avoir à payer du tout.

- **sécurisé :** reprenons l'exemple de l'enfant de 10 ans : s'il veut effacer tout le système, *rien* ne l'en empêchera, sous windows (9.x), alors que sous Linux, il ne pourra pas, tout simplement : permission non accordée. Et là, pas question de penser aux cracks, car dès qu'une faille de sécurité est découverte, dans les trois *heures !* elle est « bouchée » et un correctif (patch) est diffusé sur Internet. Bien sûr, certains vont me dire : « et windows 2000, tu l'oublies ? ». Mais non pas un brin, j'allais en parler : windows 2000 est, lui, super protégé (tu auras compris mon ironie). Certains trous de sécurité existent, sous windows NT, depuis *trois ans.* et perdurent encore avec windows 2000 ! Quelle hypocrisie de la part de Microsoft ! A croire qu'ils le font exprès pour vendre des anti-virus. D'ailleurs, j'allais oublier, les anti-virus ne servent donc à rien sous Linux, car si on s'identifie en tant qu'utilisateur, même le virus, quel qu'il soit, ne *pourra pas* agir. Même si on en inventait un qui y arrivait, le lendemain on trouverait l'antidote (libre, lui au moins) sur Internet. Pourquoi mettre plein de pansements autour d'un tuyau mou au lieu de prendre d'entrée un bon tuyau en acier ?
  Attention : si tu exécutes un simple programme qui ordonne d'effacer tous les fichiers de ton répertoire personnel, alors Linux ne te l'interdira pas ! Sauf si tu as retiré les permissions d'écriture à ton répertoire personnel. Mais ce n'est pas la faute de Linux. Si tu tapes la commande :

```
[christophe /home/christophe]$ rm -rf *
```

alors adios muchachos ! Tu ne reverras plus tes fichiers car la commande « rm -rf » ne gère pas de corbeille. KDE, lui, possède une corbeille. Mais on peut très bien faire en sorte que lorsqu'on tape « rm » en fait les fichiers soient déplacés dans un répertoire corbeille. C'est ça la puissance des scripts.
Mais pour revenir aux virus, tout repose sur les permissions système. Et là impossible de les contourner car c'est le noyau qui les gère, et le noyau est sans défaut, ce que j'ai dit plus haut. Encore une fois, si on découvrait un défaut il serait corrigé le jour même comme ce fut le cas il y a quelques années.

- **dissocié :** Linux est un système d'exploitation qui est basé sur plusieurs « composants », chacun faisant *son boulot et rien d'autre*. Nous verrons la liste hiérarchisée (simplifiée évidemment) de ces composants dans la section 4.2. La dissociation apporte toute une foule d'avantages. D'abord, le plaisir de *comprendre* et d'être au courant de ce qui se passe dans ton ordinateur. N'as-tu jamais l'impression que windows se rebelle, ou marche contre toi ? N'as-tu jamais dit à ton ordinateur : « mais qu'est-ce tu fais ? » ou bien « je t'ai jamais demandé de faire ça ! ». Eh bien sous Linux ça n'existe pas, l'ordinateur fait exactement ce que tu veux *parce que tu sais comment Linux fonctionne*. Enfin, si tu ne le sais pas, tu peux *toujours* trouver le moyen de le savoir ! Ensuite, la dissociation des composants permet une meilleure *adaptation* de Linux à ton propre ordinateur. Tu n'es enfin plus obligé d'utiliser le même windows que tout le monde (ne joue pas au débutant en me parlant des thèmes de bureau ! je parle ici du cœur du système). Comme ça, si les performances de ton matériel sont basses, tu choisiras des composants qui n'en demandent pas beaucoup (pour les environnements de bureau il y a IceWM ou Xfce... qui sont très légers), et si tu as « the best of the computers of the moment », alors tu pourras tout utiliser (encore pour les environnements de bureau : KDE, Gnome... sont les plus gourmands en ressources système). Enfin, pour réparer une panne (très rare, ça ne m'est jamais arrivé !), tu sais d'où ça vient et tu t'attaques à un tel composant et pas à un autre, comme ça tu répares en trois minutes. Sous windows, pour trouver la cause de la panne, c'est toujours presque impossible. Combien de fois une hotline t'a conseillé de réinstaller windows ? *Ça me met hors de moi* ! Pourquoi, comme le disait Descartes, reconstruire ailleurs toute une ville sous prétexte que les rues abîmées empêchent les diligences de passer alors qu'il suffit de remettre des pavés sur ses routes ?

Windows *n'est pas* un système dissocié, ce qui fait que l'utilisateur (et encore plus celui qui veut réparer) a l'impression de se trouver en face d'un sac de nœuds. Plutôt gênant pour réparer. Sous windows, c'est dur à réparer et ça tombe sans arrêt en panne, tandis que sous Linux, c'est facile de réparer, mais ça tombe jamais en panne. Drôle de paradoxe. Quelle tranquillité ! Tu ne peux pas comprendre le gain de temps obtenu, car sous Linux, une fois que tout est installé, ça peut marcher pendant des années entières sans problème ! Tu le sais sans doute, des serveurs internet utilisant Linux fonctionnent depuis cinq ans ou plus de façon intensive *sans avoir planté ni redémarré une seule fois* !

– **stable :** On entend souvent dire que Linux ne plante (presque) jamais. Rassure-toi, je ne vais pas dire le contraire. Cependant, je voudrais préciser un petit truc. Tu as maintenant compris que Linux, ça ne veut rien dire de spécial, et qu'il faut préciser de quoi on parle. Là, je parle du noyau de Linux quand je dis que Linux ne plante pas. En fait, il faut dire : « le noyau de Linux ne plante presque pas ». Lis la section 4.2 pour comprendre ce que c'est que le noyau (et tout ce qui va avec). Linux a désormais prouvé sa stabilité. Je rappelle qu'un plantage classique sous windows génère un affreux écran bleu annonçant que « le système est peut-être occupé », que « le système est devenu instable », qu'« une erreur système est survenue inopinément » sans mentionner les bombes menaçantes qui apparaissent sur l'écran dans d'autres systèmes (elles affolent l'utilisateur qui ne sait plus quoi faire. Pire : il se sent fautif). J'adore me moquer de ces messages, comme dans le sketch de Anne Roumanov, parce qu'ils ne veulent *rien* dire ! À ce propos, il existe sous Linux un écran de veille qui reproduit les plantages typiques des divers systèmes d'exploitation : « écran bleu de la mort », ou « blue screen of death (bsod) ». C'est très utile pour faire peur à quelqu'un ! C'est bien beau tout ça, mais que dois-je faire pour réparer l'erreur qui survient ? Sous Linux, les messages d'erreur sont très explicites, donc on les résout très vite ! Du coup, on n'est pas obligé de faire reset. De toute façon, les erreurs ne sont jamais graves, ce sont plutôt des messages d'explication. Personnellement, je n'ai planté qu'une seule fois depuis deux ans. Et il faut dire que le noyau a planté au moment d'*éteindre* l'ordinateur ! Si ça te paraît impossible à envisager, que cela relève du miracle, c'est que ta vision des choses est « polluée » par windows. Essaie Linux et tu seras, comme moi il y a deux ans, illuminé. Bien sûr sous Linux il se peut qu'un programme plante, dans le cas où il serait mal programmé. Mais là où Linux se démarque de windows, c'est que *le programme responsable du plantage ne fait pas planter les autres programmes.* En fait si windows plante c'est parce qu'il gère mal la mémoire (RAM). Quand un programme est exécuté, il utilise de la mémoire. Quand un autre programme est utilisé, il utilise une autre quantité de mémoire. Et ainsi de suite... (je rappelle que le fait d'exécuter plusieurs programmes à la fois s'appelle le « multi-tâche »). Si jamais un des programmes en cours plante à cause d'un bug de programmation, sous Linux, la plage de mémoire utilisée va être déclarée libre et le programme sera « tué ». Tout rentre dans l'ordre et pas besoin de faire reset. Sous windows, non seulement la plage mémoire occupée par le programme qui plante n'est pas libérée (donc de la mémoire est occupée pour rien, les performances générales du système baissent !) mais en plus à cause des imperfections flagrantes de windows, le programme qui plante fait planter à son tour windows qui lui fait planter les autres programmes... un comble pour un système d'exploitation ! On a même besoin d'utilitaires comme Memcheck qui défragmentent la RAM ! Bien sûr, sous les version NT, XP de Windows, la gestion de la mémoire s'est améliorée. Sous Linux, la gestion de la mémoire RAM est « hermétique ».

Voilà pour cette section, on a bien vu *comment* Linux était plus fort. Note qu'il existe d'autres Unix-like (systèmes ressemblant à Unix) libres, comme FreeBSD, qui eux aussi disposent de performances très élevées. Cependant ces systèmes ne sont presque employée à l'heure actuelle que par les professionnels, et GNU/Linux dispose d'une quantité d'utilisateurs tous niveaux de compétences confondus. Cette communauté est plus réduite sous les BSD.

4.2 Architecture de Linux

Quand j'ai dit que Linux était simple, c'était vrai bien sûr, et je ne t'ai pas fait une fausse joie ! C'est ici, en expliquant comment fonctionne Linux, que nous allons voir *pourquoi* Linux est si simple.

- Linux fonctionne autour de son *noyau*. Le noyau permet de communiquer entre les programmes et le matériel, afin de faire travailler ton ordinateur. Tout le reste ne fait que s'adresser au noyau, et le noyau gère tout de façon quasi-parfaite. C'est pour cela qu'on dit que Linux, ou plutôt le noyau Linux, est stable.
- le *shell* (le plus courant est Bash) est un programme très pratique qui permet de saisir les commandes dans la console. Il lit les touches du clavier sur lesquelles tu appuies, et inscrit les lettres à l'écran, puis exécute les commandes... On peut lier plusieurs commandes entre elles, rediriger leur résultat vers un fichier texte au lieu de l'afficher à l'écran, etc... On verra tout ça dans le chapitre 5. Il existe un nombre incroyable de

shells, tous ayant leurs propres caractéristiques :
 - Bash est le shell développé par le GNU, ce qui fait que c'est le shell par défaut. Ce n'est pas pour ça que c'est le meilleur.
 - Zsh : la Rolls Royce des shells. Tout peut être programmé : les raccourcis clavier, les auto-complétions...
 - Ash : il est très petit (65 Ko), ce qui fait qu'il peut être aisément utilisé pour faire une distribution tenant sur une seule disquette.
 - Tcsh : son langage de script ressemble énormément au langage C.
- quand un programme est exécuté, le noyau le gère comme un *processus*. Mais qu'est-ce que c'est ? Sous les systèmes Unix, il y a le noyau et les processus. Rien d'autre. Sous GNU/Linux, le noyau, à la sortie de Lilo, donne naissance au premier processus : « init ». Ce processus lance à son tour les scripts de démarrage (voir la section 4.4.1). Le démarrage se poursuit, puis on arrive au shell. Contrairement au DOS, par exemple, où l'invite de commandes est intégrée au système, le shell sous Linux est un processus comme les autres, qui lance d'autres processus lorsque tu tapes les commandes. Et le processus fils correspondant à la commande exécutée meurt dès que la commande a terminé son travail. Un processus occupe une certaine part de la mémoire RAM, une certaine part des ressources processeur (CPU), etc... Il peut être arrêté, tué sauvagement (si le programme ne répond plus) et dans tous les cas le noyau gère les processus de façon stable. Ce qui fait que si un programme ne répond plus, on affiche la liste des processus, on tue le processus qui correspond au programme planté, le système « vide » la mémoire occupée par le programme planté afin que les autres programmes en bénéficient, et on repart de plus belle !
- pour afficher des graphismes (fenêtres par exemple), on passe par plusieurs composants. C'est là que tu dois te rappeler la notion de dissociation, vue à la page 51. On utilise le *serveur graphique* très connu : XFree86. Il a *pour seul but* de charger les pilotes de *ta* carte graphique et les fichiers de configuration, afin d'afficher *quelque chose* (attention ce n'est justement pas lui qui va afficher les fenêtres !). C'est lui aussi qui gère la souris et le clavier (et d'autres périphériques comme par exemple une tablette graphique).
- le *gestionnaire de connexion* est un petit programme graphique, qui s'appuie sur le serveur graphique, qui te demande de t'identifier et de donner ton mot de passe au démarrage de l'ordinateur.
- ensuite après le login (identification) le *gestionnaire de fenêtres* va s'appuyer aussi sur le serveur graphique pour afficher... des fenêtres ! Les gestionnaires de fenêtres sont très nombreux : on a Enlightenment, Sawfish, IceWM, Xfce, etc...

- le gestionnaire de fenêtres s'appuie sur le serveur graphique : il a pour but de gérer la taille des fenêtres qui s'affichent, mais aussi leur barre de titre (pour les déplacer par exemple), les boutons de la barre de titre (fermer, agrandir la fenêtre...), il doit permettre de les redimensionner... c'est tout ! L'intérieur de la fenêtre, c'est-à-dire l'interface graphique du logiciel, ne l'intéresse pas.
- les *environnements de bureau* s'appuient sur un gestionnaire de fenêtres (Gnome peut utiliser Sawfish, Enlightenment ou autre pour afficher les fenêtres). Ils intègrent en plus un *panel* qui est l'équivalent mais en mieux de la barre des tâches de windows. Tout ceci est très modulaire. Ce n'est donc pas Gnome qui va afficher les fenêtres, mais le gestionnaire de fenêtres que tu lui auras dit de prendre.
- les *gestionnaires de fichiers* gèrent les fichiers. (Non, pas possible...) Eh bien oui il faut le préciser ! Gérer les fenêtres n'a rien a voir avec gérer les fichiers de façon graphique (icônes représentant fichiers et dossiers). Il existe plusieurs gestionnaires de fichiers (windows ne propose *que* son explorateur) : Konqueror, Krusader, Nautilus, GMC, XWC, ROX, XFM, etc... et en console le très célèbre Midnight Commander

Sous windows, tout est mélangé pour le plus gros sac de nœuds possible.

Pourquoi faut-il la dissociation ? Parce que comme ça on a le choix d'adapter la configuration aux performances. On n'est pas obligé d'accepter ce que l'on ne veut pas, on choisit *tout* à la carte.

4.3 Les environnements graphiques

Nous allons nous intéresser aux environnements graphiques (ou environnements de bureau). Il existe plusieurs gestionnaires de fenêtres sous GNU/Linux, mais deux projets ont dépassé ce stade et sont devenus des environnements de bureau. Il s'agit de KDE et de Gnome. Nous allons étudier ces deux projets incontournables puis nous parlerons des « simples » gestionnaires de fenêtres.

4.3.1 KDE

KDE signifie K Desktop Environnement (Environnement de bureau K). Pourquoi K ? Parce que c'est la lettre qui vient avant le L de Linux. C'est pas plus compliqué que ça.

KDE est donc un projet lancé vers 1995, qui a pour but de donner un environnement complet à tout utilisateur de GNU/Linux. KDE s'est inspiré de CDE, un environnement graphique pour les systèmes Solaris. Voilà où KDE en est aujourd'hui :

- un gestionnaire de fenêtres : K Window Manager
- un tableau de bord (barre en bas de l'écran qui permet de placer des icônes, des menus, des tiroirs, des applets (petits programmes qui s'y intègrent) très facilement configurable
  - par exemple comme applet on a un moniteur de surveillance des ressources système, un dictionnaire, une pipette pour nous donner les composantes rouge-vert-bleu d'une couleur, une horloge, des résultats sportifs (connexion à Internet nécessaire), un sélecteur de caractères spéciaux...
- tous les logiciels nécessaires à l'utilisation de l'ordinateur :
  - applications Internet
    - KMail : lire, écrire et gérer ses e-mails
    - KMago : gérer des téléchargements
    - KSirc : pour chatter sur l'IRC
    - Kaim : version KDE du logiciel d'AOL pour chatter
    - KNode : consultation des newsgroups
  - applications d'administration/configuration
  - applications multimédia (lecteurs musique/vidéo...)
  - utilitaires (gravure des CD-Roms,...)
  - jeux
  - applications éducatives (mathématiques, géométrie, vocabulaire de langues étrangères, apprendre à taper au clavier avec les 10 doigts...)
  - suite bureautique (KOffice)
    - Traitement de texte : KWord (mi-chemin entre Word et Publisher pour ceux qui connaissent Microsoft Office !)
    - Processeur de texte : KLyX (comme LyX, logiciel sur lequel je travaille, décrit à la section 7.1)
    - Tableur : KSpread (en anglais tableur se dit Spreadsheet)
    - Logiciel de graphiques : KChart (représente les données issues du tableur sous forme de camemberts, histogrammes...)
    - Éditeur de formules mathématiques : KFormula
    - Logiciel de présentations : KPresenter
    - Logiciel de création de diagrammes : Kivio
    - Dessin vectoriel : Kontour (anciennement Killustrator)
    - Dessin bitmap : Krayon
    - Gestion des finances : KMyMoney
    - connexion aux bases de données : KMySql
  - gestionnaire de fichiers (Konqueror)
  - navigateur internet (Konqueror aussi)

  - client FTP : encore Konqueror ! Pour accéder à un compte FTP anonyme, il suffit de taper l'adresse du compte (par exemple `ftp ://ftp.truc.com`). Par contre, si tu as besoin de t'identifier (par exemple pour les comptes accueillant des sites web), il faut taper les adresses sous cette forme : `ftp ://utilisateur@ftp.truc.com`, où l'utilisateur est le login du compte ftp. Ensuite une boîte de dialogue s'affiche et on tape le mot de passe. Par exemple, pour accéder à mon site, je tape `ftp ://cbenz@ftp.tuxfamily.org` et je tape mon mot de passe dans la boîte de dialogue.
  - Mais Konqueror fait encore plein de choses : il intègre en plus des plugins qui permettent de générer des galeries de photos en HTML à partir d'un répertoire contenant des images, d'extraire à la volée des MP3 (et Ogg Vorbis) depuis les CD audio, etc...
- un bureau avec des icônes, une corbeille, des bureaux virtuels...
- intégration de divers serveurs, comme :
  - Arts pour la gestion du son
  - Cups pour la gestion des imprimantes
- gestion des plantages intégrée : on appuie sur Control+Alt+Echap puis le curseur se change en éclair (ou tête de mort selon la version). Ensuite il suffit de cliquer sur la fenêtre qui ne répond plus et elle disparaît. Cette combinaison de touches appelle le logiciel « xkill ». Si on n'est pas dans KDE, on peut s'en servir en l'utilisant, exactement de la même façon. Si un logiciel plante, une fenêtre nous prévient, nous demande si on souhaite faire un rapport de bogue, puis c'est fini. Le logiciel disparaît et on repart de plus belle. Pas de reset !
- le Panneau de Contrôle qui permet de personnaliser KDE à travers une soixantaine de modules de contrôle (un pour le réglage de la souris, un pour le clavier, un autre pour les e-mails, un autre pour les associations de fichiers aux applications... mais beaucoup ne sont là que pour donner des informations sur le système)
- un outil de développement complet (IDE - Integrated Developpement Interface) qui nous assiste de la création du projet jusqu'au paquetage final prêt à être distribué, en passant par la compilation, la création de l'interface graphique et même le débuggage. Ce logiciel s'appelle KDevelop.
- un look qui peut être complètement personnalisé : les thèmes, les jeux d'icônes, les boutons des barres d'outils, les menus, les fonds d'écran... tout est flexible en configuration.

Tu l'auras compris, KDE est très complet, et la version 3 est livrée avec plus de 1000 logiciels. On reconnaît les logiciels faisant partie de KDE car ils commencent pas la lettre K.

Pour les programmeurs, KDE est intéressant car il est très modulaire : une application pourra tout de suite afficher du HTML en appelant le KPart KHTML. On construit ses applications comme un puzzle, et les composants sont très puissants. Ensuite, il est très simple d'écrire des plugins pour Konqueror, des applets pour le tableau de bord... KDE utilise la bibliothèque graphique QT (désormais en version 3), qui est aujourd'hui totalement libre (mais payante pour des projets commerciaux, ce qui est tout à fait normal).

Adresse internet du projet KDE : `http://www.kde.org`. Ce site est vraiment complet, il est partagé en plusieurs autres sites (on y a accès à partir de celui que j'ai donné). Par exemple pour tout ce qui touche au multimédia, on va sur `http://multimedia.kde.org`, et pour trouver de nouveaux logiciels on va sur `http://apps.kde.com` (et il y en a, des logiciels, va voir !).

4.3.2 Gnome

Gnome signifie : « The GNU Network Object Model Environment ». Gnome est, comme KDE, un autre environnement de bureau pour GNU/Linux. Il possède les mêmes atouts, mais est basé sur une technologie complètement différente. En effet il utilise la bibliothèque GTK, qui est aussi totalement libre.

Gnome utilise plus de composants externes que KDE. Par exemple, il se base sur un gestionnaire de fenêtres qui existe déjà (comme Sawfish). Gnome propose :

- un tableau de bord (barre en bas de l'écran qui permet de placer des icônes, des menus, des tiroirs, des applets (petits programmes qui s'y intègrent) très facilement configurable
- tous les logiciels nécessaires à l'utilisation de l'ordinateur :
  - applications Internet
    - Balsa, Sylpheed, Evolution : lire, écrire et gérer ses e-mails
    - Gwget : gérer des téléchargements
    - XChat : pour chatter sur l'IRC
    - Gaim : version KDE du logiciel d'AOL pour chatter
    - Pam : consultation des newsgroups
  - applications d'administration/configuration
  - applications multimédia (lecteurs musique/vidéo...)
  - utilitaires (gravure des CD-Roms,...)
  - jeux
  - suite bureautique (GnomeOffice)
    - Traitement de texte : AbiWord (ressemble comme deux gouttes d'eau à Word)
    - Tableur : Gnumeric

    - Logiciel de présentations : Achtung
    - Logiciel de graphiques : Guppi (représente les données issues du tableur sous forme de camemberts, histogrammes. . .)
    - Logiciel de création de diagrammes : Dia
    - Dessin vectoriel : Sodipodi et Sketch
    - Dessin bitmap : The Gimp
    - Gestion des finances : GnuCash
    - connexion aux bases de données : Gnome-DB
  - gestionnaire de fichiers (Nautilus)
  - navigateur internet (Nautilus, Galeon)
  - client FTP (gFTP)
- un bureau avec des icônes, une corbeille, des bureaux virtuels. . .
- intégration de divers serveurs, comme :
  - esd pour la gestion du son
- gestion des plantages intégrée (bugzilla) : si un logiciel plante, une fenêtre nous prévient, nous demande si on souhaite faire un rapport de bogue, puis c'est fini. Le logiciel disparaît et on repart de plus belle. Pas de reset !
- le Panneau de Contrôle qui permet de personnaliser Gnome à travers plusieurs modules de contrôle
- un look qui peut être complètement personnalisé : les thèmes, les jeux d'icônes, les boutons des barres d'outils, les menus, les fonds d'écran. . . tout est flexible en configuration.

Note bien, et c'est très important, que l'on peut utiliser les logiciels Gnome depuis KDE et vice-versa ! C'est une question qui revient sans-arrêt, et on comprend facilement pourquoi. En effet, l'apparence graphique des logiciels KDE/Gnome est régie par deux bibliothèques différentes. Sous Linux on peut charger n'importe bibliothèque n'importe quand ! Donc pas de problème quant à charger QT depuis Gnome ou GTK depuis KDE.

4.3.3 Autres gestionnaires de fenêtres

Il existe, en dehors des environnements de bureau complets, des gestionnaires de fenêtres (windows managers en anglais). Ils sont généralement plus légers que KDE ou Gnome car leur boulot est simplement de traiter les fenêtres (leur barre de titre, leurs boutons de fermeture/agrandissement. . . en tout cas pas leur contenu). Les gestionnaires de fenêtres ne proposent pas non plus de logiciels comme le font KDE ou Gnome. Voici quelques gestionnaires de fenêtres :

- Blackbox
- AfterStep

- WindowMaker
- Sawfish
- Enlightenment
- Xfce
- etc...

4.4 Arborescence des répertoires

J'en vois déjà qui se demandent ce que c'est qu'une arborescence.

arborescence : en informatique, on trie les fichiers dans des répertoires. Imagine que tu es dans une salle de travail, avec des armoires contenant des pochettes cartonnées qui, elles, contiennent des feuilles. L'armoire correspond en informatique au disque dur, et sous Linux plus particulièrement au point « / », appelé *racine* de l'arborescence. Ensuite viennent les pochettes cartonnées. Tu l'auras compris, elles représentent les *répertoires*. Dans les pochettes on trouve des feuilles, ce sont les *fichiers*. Là où l'informatique va plus loin que cette représentation simpliste, c'est que tu peux mettre d'autres pochettes dans les pochettes existantes ! Ce sont les *sous-répertoires*. On peut donc imbriquer les répertoires à l'infini. L'arborescence est le mot qui désigne la manière dont sont organisés tous ces répertoires, c'est-à-dire comment ils sont imbriqués les uns dans les autres.

Maintenant que la notion d'arborescence est plus claire, on va s'intéresser à celle d'un système Linux normal. Entre les distributions certains détails peuvent changer, mais on va faire l'impasse. Pour l'info, j'utilise Mandrake 8.1 pour le moment. J'insiste sur le fait que normalement tout ce qui va suivre fonctionnera de la même manière sur n'importe quelle distribution. Je tape dans la console « `cd /` » pour aller à la racine de l'arborescence puis « `ls` » pour en afficher le contenu. On verra ces commandes dans le chapitre 5. Pour l'instant ne cherche pas trop à comprendre. Voici le résultat :

```
[christophe ~]$ cd /
[christophe /]$ ls
bin/   dev/  home/  mnt/  proc/  sbin/  usr/
boot/  etc/  lib/   opt/  root/  tmp/   var/
```

On remarque que sur la première ligne se trouve le caractère « ~ » (tilde). Il symbolise le répertoire maison de l'utilisateur. Lorsqu'on tape « cd / », on ordonne au shell (l'interpréteur de commandes) d'aller à la racine du système de fichiers (c'est comme l'armoire de tout à l'heure). Le caractère « ~ » se change donc en « / » pour nous indiquer où on se trouve. Je te conseille de jeter un œil à la figure 4.1 afin de mieux comprendre cette arborescence. Les anciens utilisateurs du DOS n'auront aucun mal à comprendre.

Chaque répertoire (bin, dev, etc...) est directement imbriqué dans « / ». Si je vais dans le répertoire « /usr » en tapant « `cd usr` », et que je demande d'afficher son contenu avec « `ls` », j'obtiens :

```
[christophe /]$ cd usr/
[christophe /usr]$ ls
bin/  etc/    include/  java/  libexec/  man@   share/  tmp@
doc/  games/  info/     lib/   local/    sbin/  src/    X11R6/
```

On voit que dans « /usr » il y a d'autres répertoires. On pourrait continuer à explorer un autre sous répertoire tel que share, mais je pense que ce serait inutile car tu as certainement compris. A toi de le faire pour t'amuser !

Maintenant qu'on a vu comment naviguer dans les répertoires en console, on va voir quelles sont les caractéristiques de chaque répertoire.

/ — c'est la racine du système de fichiers. Tous les répertoires suivants sont situés dans à la racine de l'arborescence.

/bin — contient les commandes de base du système comme cat, cp, mv, ls, ln, tar ou encore gzip. Voir le chapitre 5 pour comprendre ces commandes.

/boot — divers fichiers qui concernent le démarrage de l'ordinateur, notamment le chargeur de démarrage Lilo (ou Grub), le noyau et d'autres trucs. On n'y va jamais en pratique, sauf lorsqu'on copie un nouveau noyau fraîchement compilé, afin de l'ajouter au système...

/dev — contient tous les fichiers qui représentent les périphériques. En effet, sous Linux, chaque périphérique est représenté sous forme d'un fichier situé dans le dossier /dev.

Tous les périphériques y figurent : le clavier, la souris, les ports de communication... Voici un résumé de la façon de nommer les périphériques sous Linux :

- hdx représentent les disque durs (ou CD-Roms) sur le port IDE. On remplace x par a, b, c ou d. hda1 correspond au périphérique maître sur la première nappe IDE, hdb correspond au périphérique esclave sur la première nappe IDE, hdc correspond au périphérique maître sur la seconde nappe et hdd correspond au périphérique esclave sur la seconde nappe.

- fdx représentent les lecteurs de disquettes, où x est à remplacer par 0 ou 1. Ainsi, fd0 représente le premier lecteur de disquettes, et fd1 le second.
- ttySx représentent les ports série (x va de 0 à 1 généralement)
- lpx représentent les ports parallèles (x va de 0 à 1 si tu as 2 ports parallèles, sinon tu n'accèdes qu'à lp0)
- il y a beaucoup d'autres fichiers de périphériques, mais on ne va pas tous les lister ici

Par exemple, /dev/fd0 concerne le premier lecteur de disquettes. De même que pour /boot, on n'y va presque jamais pour une utilisation quotidienne de l'ordinateur. Voir la section 8.3 pour plus de renseignements sur l'accès aux périphériques.

/etc dans ce répertoire se trouvent plein de fichiers de configuration des logiciels GNU/Linux. Voir la section 4.4.1 pour avoir la liste des principaux fichiers de configuration à connaître.

/home c'est le répertoire qui contient les répertoires maison de tous les utilisateurs. Par exemple, mon login (identificateur de connexion) est « christophe », alors mon répertoire maison est /home/christophe. On n'est pas obligé de placer les répertoires maison des utilisateurs dans /home, mais c'est un standard. Il est bon de placer le répertoire /home dans une partition différente de la racine, afin de faciliter la mise à jour du système : on peut formater la partition / (contenant la racine) ou /usr (voir plus loin) sans toucher aux données personnelles ni aux réglages personnels des logiciels.

/lib les bibliothèques des programmes (ou librairies) se trouvent ici

/mnt contient par convention les répertoires qui permettent d'accéder aux lecteurs. Par exemple, « `ls /mnt/cdrom` » liste le contenu du cdrom qui est dans le lecteur. Voir la section 8.3 pour plus de détails.

/opt ce répertoire contient les programmes ajoutés en dehors de la distribution. En général, ce répertoire est situé sur une autre partition que celle qui contient la racine (comme /home) afin de faciliter les mises à jour du système (même raison que pour /home). De plus, chaque programme de /opt est dans son propre sous-répertoire, ce qui conduit à une bonne structuration (alors que tout est en vrac dans /usr/bin !)

Par exemple, si StarOffice n'est pas fourni par ta distribution (Mandrake la fournit), tu l'installeras dans /opt.

/proc contient divers fichiers virtuels (c'est la représentation sous forme de fichiers d'informations du noyau ; ce ne sont pas de vrais fichiers stockés sur le disque dur) contenant des informations sur les processus. Je rappelle qu'un processus est un programme qui est en train d'être utilisé, ou d'ailleurs en attente, programme qui peut être visible ou non à l'écran. Lire la page man de proc en tapant « `man proc` » dans une console. Voir la section 6.1 pour savoir comment voir les pages man.

/root c'est le répertoire maison du super utilisateur root. On aurait pu le mettre, il est vrai, dans « /home/root », mais on a préféré le mettre à part, et cela se comprend, vu que root est le chef des chefs.

/sbin contient les commandes utilisables seulement par root (celui qui a tous les droits), comme halt qui arrête l'ordinateur, fdisk qui partitionne le disque, etc... Tout le monde ne peut pas faire ces opérations, ce serait trop dangereux. Il existe des moyens d'autoriser certains utilisateurs à effectuer des tâches : les scripts « sudo ». Je n'en dirai pas plus ici...

/tmp c'est le répertoire « temporaire », c'est-à-dire que tout ce qui s'y trouve n'est pas très important. On n'y touche pas en théorie, ce sont les programmes qui l'utilisent pour stocker des fichiers de sauvegarde, ou pour faire diverses opérations, sans encombrer ton répertoire maison.

/usr répertoire très important ! Il contient tous les programmes utilisateur (/usr/bin), les données partagées (/usr/share) entre tous les programmes (icônes, fonds d'écran, sons, etc...), les documentations (/usr/share/doc)... c'est très complet !

/var contient les files d'attentes des imprimantes (/var/spool/lpd si tu utilises lpd pour imprimer), les emails reçus (/var/spool/mail) et diverses choses que je n'ai pas encore apprises.

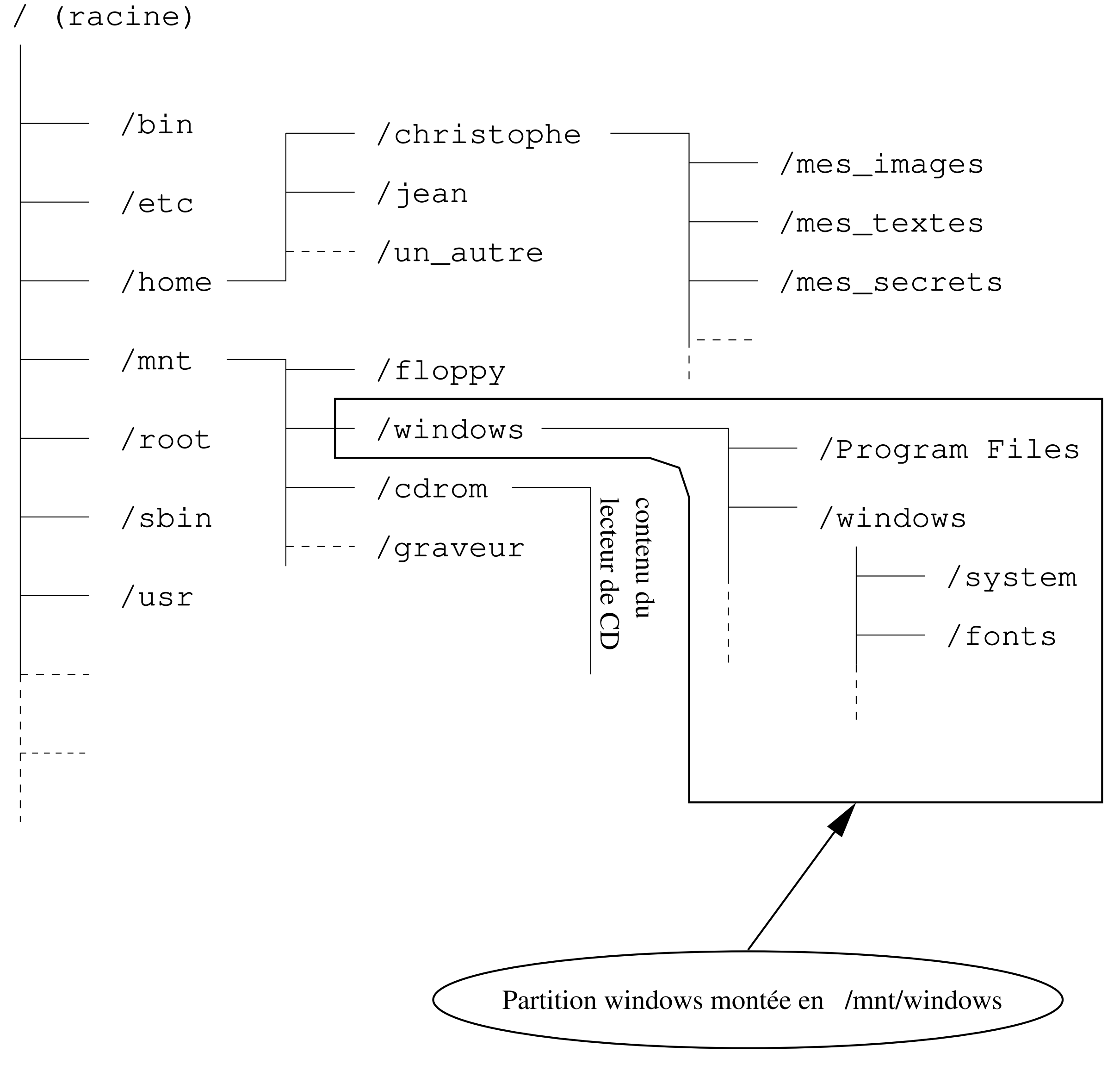
Cette organisation des répertoires est en passe d'être standardisée par le projet Linux Standard Base (LSB) qui définit le File Hierarchy Standard (FHS). Les distributions en général respectent cette arborescence, mais il y a de petites variantes de l'une à l'autre.

La figure 4.1 schématise l'arbre des répertoires.

4.4.1 Principaux fichiers de configuration

Linux est réglé par ses fichiers de configuration. Ce sont de simples fichiers texte qui contiennent ces réglages. Les « anciens » configurent leur système

FIG. 4.1 – Arborescence typique sous Linux



Tu peux voir que la partition de windows est montée sur le répertoire « /mnt/windows », et que la notion de lecteur C : ou D : etc... ne veut plus rien dire. Par exemple, pour accéder au contenu du lecteur de CDROMs, on n'a plus qu'à lister (faire afficher) les fichiers présents dans « /mnt/cdrom ». Bien sûr, on peut choisir de monter ses partitions autre part, mais par convention on utilise ce que j'ai mis.

en modifiant directement ces fichiers. C'est la façon de faire la plus rapide, mais le débutant a toujours peur d'effacer une ligne ou de se tromper dans la syntaxe. C'est pour cela qu'il existe, aujourd'hui, des interfaces graphiques (comme linuxconf ou webmin) qui génèrent pour toi le fichier de configuration. En général, on peut moins affiner les réglages avec les interfaces, mais elles ont l'avantage d'être très intuitives.

On distingue deux sortes de fichiers de configuration : les fichiers qui décrivent les réglages communs à tous (ils sont placés dans /etc) et les fichiers personnels (placés dans ton répertoire personnel) qui ne s'appliquent qu'à toi. C'est l'application de la notion de multi-utilisateurs, vue à la section 4.1. Personne à part le super utilisateur ne peut modifier les fichiers communs. C'est pour cela que les fichiers de /etc sont accessibles en lecture seule. Mais comme chacun doit pouvoir avoir le droit de configurer la machine comme il le souhaite, il peut enregistrer ses propres fichiers dans son répertoire personnel.

En fait, les programmes vont d'abord chercher les fichiers personnels puis, le cas échéant, vont prendre les fichiers communs. Parfois même, comme pour le fichier de configuration du shell Bash, le fichier personnel (`~/.bashrc`) *appelle* le fichier commun (`/etc/bashrc`). Dans ce cas le fichier personnel est une *extension* du fichier commun.

Dans ton répertoire personnel, les fichiers de configuration sont cachés. Pour les voir, il faut taper « ls -a » (on le verra plus tard, à la section 5.1). On ne les voit pas car ils commencent par un point, et le shell n'affiche pas les fichiers commençant par un point.

Nous nous intéressons aux fichiers communs, c'est-à-dire ceux qui se trouvent dans le répertoire /etc. En voici la liste des principaux :

/etc/rc.sysinit c'est le script de démarrage qui est exécuté en premier. Il appelle beaucoup des scripts qui suivent au démarrage de l'ordinateur. C'est dans ce script qu'on peut changer la variable du PATH, mais elle ne sera accessible qu'au prochain redémarrage. Il faut donc aussi changer la variable de façon interactive dans le shell en attendant le prochain redémarrage.

/etc/rc.local ce script est exécuté après tous les autres, lors du démarrage de l'ordinateur.

/etc/bashrc le fichier de réglage de bash. On peut notamment régler la variable PS1, qui correspond au prompt (invite de commandes) du shell.

/etc/profile sous Mandrake 8.x, exécute tous les fichiers de « /etc/profile.d/*.sh ». Notamment :

alias.sh les alias du shell

color_ls.sh définition des options d'affichage pour la commande « ls » (couleur…)

/etc/fstab association des périphériques de stockage (CD-Roms, disques durs et disquette…) et de leurs partitions avec leurs points de montage. Indique aussi les permissions et le système de fichiers. C'est dans ce fichier qu'on active le supermount (fonctionnalité permettant de pouvoir changer dynamiquement les CD-Roms sans avoir à les monter et démonter). Voici un exemple de fichier fstab, avec supermount :

```
/dev/hda1 /mnt/windows vfat iocharset=iso8859-15,umask=0,codepage=850 0 0
/dev/hda5 swap swap defaults 0 0
/dev/hda6 /home reiserfs notail 1 2 /dev/hda7
/ reiserfs notail 1 1
/dev/hdc1 /mnt/3go reiserfs notail 1 2
/dev/hdc2 /mnt/3go_fat vfat
iocharset=iso8859-15,umask=0,codepage=850 0 0
/mnt/cdrom /mnt/cdrom supermount
dev=/dev/cdrom,fs=iso9660,ro,exec 00
/mnt/floppy /mnt/floppy supermount
dev=/dev/fd0,fs=vfat 0 0
none /dev/pts devpts mode=0620 0 0
none /dev/shm tmpfs defaults 0 0
none /proc proc defaults 0 0
```

/etc/inittab décrit les niveaux d'exécution et règle le niveau d'exécution par défaut. Le système peut fonctionner sous différents niveaux d'exécutions : à chaque niveau d'exécution on peut associer ou non différents services. Un service est un programme (script) qui va fournir une fonctionnalité. On verra comment démarrer ou non les services à l'allumage de l'ordinateur plus loin.

Par convention, le niveau 3 signifie que l'ordinateur démarre en mode texte, le niveau 5 correspond au mode graphique. Le niveau 0 correspond à l'arrêt complet de l'ordinateur, et le niveau 6 correspond au redémarrage de l'ordinateur. Dans ce fichier, on règle justement dans quel niveau d'exécution on veut se trouver au démarrage de l'ordinateur. Généralement, on choisit entre 3 et 5, le plus souvent 5 pour arriver en mode graphique. Il paraît évident qu'il ne faut pas mettre 0 sinon l'ordinateur s'éteint dès qu'il a fini de démarrer, ni 6 sinon il redémarre après s'être allumé.

Chez moi j'ai choisi le niveau 5, et je vois cette ligne vers le début du fichier :

```
id :5 :initdefault :
```

La dernière ligne de inittab permet de choisir, au cas où on ait choisi le niveau 5, le gestionnaire de login (le programme qui permet de s'identifier vis à vis du système).

/etc/rc.d/init.d/ ce répertoire contient simplement tous les services installés. Les services sont donc des programmes s'exécutant en tâche de fond (arrière plan). Des exemples de services ? En voilà :
- mysql (base de données)
- httpd (serveur http, plus connu sous le nom de serveur web) Ce serveur porte le nom de Apache[1], c'est vraiment un serveur Internet très performant.
- anacron permet de lancer des travaux à intervalles réguliers ou à des dates précises (par exemple, faire une sauvegarde tous les lundis matin)

/etc/rc.d/rcx.d/ c'est une série de répertoires qui fonctionnent avec inittab. Dans le répertoire /etc/rc.d on trouve les sous-répertoires « `rcx.d` », où `x` est à remplacer par le niveau d'exécution. Dans les répertoires rcx.d, on trouve les services à lancer ou non pour chaque niveau d'exécution. Par exemple, on veut configurer les services à lancer au démarrage de l'ordinateur. On va alors aller dans /etc/rc.d/rc5.d (on suppose que le niveau d'exécution par défaut choisi est 5) :

```
[root ~]# cd /etc/rc.d/rc5.d
```

Ensuite, dans ce répertoire, on trouve des fichiers commençant par S et d'autres par K. Ceux qui commencent par S (start) seront exécutés au démarrage de l'ordinateur. Ceux qui commencent par K (kill) seront stoppés, au cas où ils existent[2]. Sinon ils ne seront tout simplement pas démarrés. Ces fichiers contiennent ensuite, après la lettre S ou K, un numéro : il représente l'ordre chronologique dans lequel ils sont exécutés. Le 01 sera démarré en premier et le 99 en dernier. On peut mettre les numéros qu'on veut (sauf dans le cas où un service a besoin d'un autre pour

[1] Il existe aussi Caudium, Zope...

[2] Dans le cas où on allume l'ordinateur, c'est sûr, aucun service n'est en exécution. Par contre, si on passe d'un niveau d'exécution à l'autre, c'est utile de stopper un processus qui serait actif dans le niveau d'exécution précédent.

fonctionner, alors il faut faire attention à l'ordre). Puis après ce numéro on voit le nom du service.

Ces fichiers sont justement des liens vers les fichiers de services présents dans /etc/rc.d/init.d/

/etc/inputrc configuration du clavier. Associe les touches du clavier un peu spéciales comme « Page Up/Down » ou les flèches à leurs fonctions réelles. Dans ce fichier, on peut aussi placer la variable « bell-style » à la valeur « visible » afin de désactiver le bip du système. Concrètement, on place la ligne suivante dans le fichier :

```
set bell-style visible
```

et au prochain redémarrage du shell, on verra l'écran clignoter au lieu d'entendre le bip. Pour redémarrer le shell, on le ferme avec Ctrl+D puis on en ouvre un autre. Pas besoin de redémarrer l'ordinateur tout entier !

/etc/lilo.conf configure Lilo, le chargeur de démarrage qui permet de choisir entre windows et Linux au démarrage de l'ordinateur. Ce fichier peut être généré par l'utilitaire « drakboot » si on a la distribution Mandrake, sinon on peut utiliser « linuxconf ».

Si tu utilises un autre chargeur de démarrage comme Grub, ne sais pas comment on le configure.

/etc/man.config configuration des pages man (manuel) : emplacement des pages man et des décompresseurs (gzip, gunzip, bunzip2...).

/etc/modules.conf définition des modules du noyaux à charger et de leurs paramètres.

/etc/modules ce fichier est appelé par rc.sysinit. Il permet d'ajouter les modules du noyau à charger systématiquement au démarrage de l'ordinateur, même s'il n'en a pas besoin. Normalement, le noyau charge dynamiquement les modules nécessaires, mais dans certains cas il faut les charger impérativement.

/etc/motd signifie « Message Of The Day » (message du jour) : c'est simplement le message qui est affiché à chaque fois qu'on lance le shell. L'administrateur peut le changer pour donner l'ordre du jour... mais pour la maison on préfère laisser ce fichier vide, car le message devient vite agaçant. On peut aussi en faire un lien vers le programme fortune, qui affiche une citation ou une blague à chaque fois qu'on l'exécute.

/etc/mtab contient les partitions et leur point de montage, en temps réel. fstab contient des données en absolu, alors que mtab change si le montage des périphériques change. mtab est affiché si on tape la commande « mount » toute seule. La notion de montage est expliquée à la section 8.3.

/etc/passwd ce fichier décrit les comptes utilisateurs, et contient les mots de passe cryptés.

/etc/group ce fichier décrit les groupes d'utilisateurs.

/etc/DIR_COLORS configuration des couleurs qu'affiche la commande « ls » dans le terminal en fonction du type des fichiers

4.4.2 Le répertoire maison

Chaque utilisateur « physique[3] » d'une machine se voit attribué un répertoire maison lors de la création de son compte (on dit aussi répertoire personnel). Il n'est cependant pas obligé d'en avoir un... mais ce n'est vraiment pas courant. Le répertoire maison est en fait un répertoire du disque dur qui appartient à un utilisateur, dans lequel il pourra faire ce qu'il veut. C'est là qu'on stocke ses documents, ses travaux... On peut définir des quotas afin que tel ou tel utilisateur n'ait pas droit à plus de x Mo pour son répertoire maison. Comme ça, sur un système comportant beaucoup d'utilisateurs, on est sûr que personne ne va saturer le disque dur.

Les répertoires personnels des utilisateurs sont placés *par convention* dans le répertoire `/home` suivi du nom de l'utilisateur. Par exemple, l'utilisateur « cacahuette » aura pour répertoire maison « `/home/cacahuette` ». Bien sûr, dans l'absolu, on peut placer les répertoire maison des utilisateurs n'importe où sur le disque dur, c'est à l'administrateur root de décider.

Pour accéder à son répertoire personnel il suffit de taper la commande « cd » tout court. Voici le résultat :

```
[christophe /]$ cd
[christophe ~]$ pwd
/home/christophe
[christophe ~]$
```

Bien sûr, c'est moi « christophe ». Ce qui peut te surprendre, c'est ce signe « ~ ». Explications : au départ je suis à la racine du système de fichiers (répertoire /). Ensuite, je tape « cd », ce qui a pour effet de me placer dans mon répertoire maison. Le tilde « ~ » symbolise tout simplement le

[3] un humain, tout bêtement !

répertoire maison. La commande pwd, on va la voir plus tard, dans la section 5.1. Ne panique pas, elle permet simplement de faire afficher le répertoire dans lequel on est, sous forme complète. On voit donc qu'on est bien dans « /home/christophe », mon répertoire personnel.

Le second rôle du répertoire maison est de permettre aux utilisateurs de *tout* personnaliser dans les logiciels, sans pour le moins du monde gêner les autres. Eh oui, qui n'a jamais pesté contre quelqu'un qui utilise la même machine que lui, sous prétexte qu'il a changé le fond d'écran, ou déplacé les icônes... C'est ridicule ! Sous GNU/Linux, étant donné que chacun a son répertoire personnel, le fameux utilisateur sauvegarde ses réglages dedans. C'est là qu'interviennent les fichiers cachés.

Lorsque tu changes les réglages d'un logiciel, ce dernier les sauvegarde dans un fichier *caché* situé dans ton répertoire personnel.

Fichier caché : Un fichier est normalement visible, c'est-à-dire que l'utilisateur va le voir lorsqu'il fera afficher la liste des fichiers, se trouvant à un endroit quelconque (on verra la commande « ls » qui permet cela plus loin). À l'inverse, un fichier caché ne sera pas affiché lors de l'affichage de la liste des fichiers. Par convention sur les systèmes Unix (dont Linux), les fichiers deviennent cachés lorsque leur nom commence par le caractère « `.` » (point).
De la même manière, on cache des répertoires entiers en faisant précéder leur nom d'un point.

En général, les fichiers de réglages des logiciels finissent par « rc ». Donc, si on applique cette nomenclature, on trouve tout de suite le nom du fichier de configuration du logiciel « vim » (le meilleur éditeur de textes, selon moi...) : « `~/.vimrc` ». Le tilde, je le rappelle, signifie « `/home/christophe` » (chez moi) puis vient le point, qui rend le fichier « `.vimrc` » caché. À la fin, on voit les deux lettres « rc », et on comprend que c'est un fichier de configuration.

De même, « `~/.bashrc` » est le fichier de configuration du shell « Bash ». Les réglages de KDE sont tous stockés dans un répertoire caché nomme « `.kde/share` ».

Dans la section précédente, on a vu que les fichiers de configuration se trouvaient dans le répertoire « /etc ». Tu vas me dire : « Alors, c'est dans /etc ou dans ~/ qu'on les met, ces fichiers ? ». D'abord je tiens à préciser que les fichiers de configuration sont générés automatiquement par les logiciels. Ce sont eux qui se chargent de les enregistrer au bon endroit. Parfois, il faut tout de même aller changer un paramètre à la main.

Ensuite, il faut savoir que les logiciels, la plupart du temps, lisent les fichiers de configuration généraux, qui se trouvent dans /etc, puis regardent si on en a un personnel dans son répertoire maison.

Chapitre 5

Les commandes fondamentales de Linux

Sommaire

Pour un utilisateur débutant, l'utilisation de la console texte et des commandes système est repoussant. Même les experts sous Windows auront peur de l'utiliser. Ils assimilent la console texte à une horrible torture pour leurs doigts, martelant leur clavier. Ils pensent qu'ils vont perdre du temps à taper les commandes, et qu'il vaut mieux utiliser la souris et des interfaces graphiques à la pelle. Si tu savais comme ils se trompent ! Bien sûr, si je te place devant la console sans que tu saches t'en servir, tu ne sauras rien faire. L'avantage des interfaces graphiques est sûrement d'éviter la phase d'apprentissage. On a l'impression de manipuler des objets concrets (icônes, menus. . .) et c'est ce qui fait que les débutants sont alléchés. Par contre, une fois qu'on sait se servir des commandes, on préfère largement les utiliser, tellement on gagne du temps : plus besoin de cliquer partout, d'attendre que les fenêtres apparaissent. . . on peut réellement programmer l'ordinateur le plus simplement du monde, en lui donnant une liste de travaux à effectuer, 10000 fois par exemple. Ce sont les scripts. Sous MacOS il existe AppleScript, un langage simple permettant de donner des consignes au système, sous Linux c'est la même chose. On va commencer par étudier plusieurs commandes séparément, puis on verra comment les encastrer pour créer de véritables petits programmes.

Il y a certaines commandes que tu ne peux pas ignorer. Sinon tu ne pourrais rien faire de vraiment intéressant, à moins de se borner à KDE ou Gnome (des environnements graphiques), ce qui serait dommage. Tu passerais à côté de tant de choses ! En effet, le fait de connaître les commandes de base permet de « travailler » bien plus vite, plus efficacement, et cela avec bien moins d'efforts que sous un environnement graphique.

Il existe une bonne quantité de commandes, mais toutes ne sont pas essentielles. La réaction que tous les débutants ont est quelque peu amusante : « Comment vais-je faire pour retenir toutes ces commandes ? ». En fait ce sont toujours les mêmes qui reviennent, il y en a peut-être une dizaine vraiment courantes. De plus, il suffit de taper « man truc » pour voir le manuel de la commande « truc ».

La plupart des commandes système sont dans le répertoire /bin, ce sont les commandes fondamentales, et d'autres dans /usr/bin, secondaires. Celles qui sont réservées à root sont dans /sbin ou dans /usr/sbin. On met dans /usr/bin ou /usr/sbin les commandes secondaires. Les commandes permettant de gérer directement le système sont appelées fondamentales, et les commandes servant à des tâches plus diverses sont appelées secondaires. Nous n'allons que détailler les commandes les plus courantes, présentes dans les répertoires que j'ai cité juste avant. Si tu veux de la documentation sur une autre commande que tu aurais vue dans un des répertoires cités au dessus, va voir sa page man ou - -help. (voir la section 6.1 pour savoir comment trouver de la doc sur ton propre disque dur).

Une commande est un petit programme (généralement écrit en C, tout comme le noyau Linux) qui sert à effectuer une tâche très précise, de façon parfaite, le plus rapidement possible. Il n'y a pas deux commandes fondamentales qui font la même chose. Les commandes sont entrées au clavier dans le shell, programme qui permet de les exécuter, de t'afficher les résultats... Le shell le plus courant sous GNU/Linux est Bash (Bourne Again SHell), toutes les distributions récentes le proposent par défaut. Zsh est également très performant.

Les commandes sont simples et sans fioritures. Leur puissance ne réside pas en elles-mêmes, mais dans la façon de les combiner. Elles fonctionnent ensemble, comme les wagons d'un train. Chaque commande prend en entrée des paramètres que tu lui fournis, et te renvoie en sortie le résultat de ses cogitations. Si ce résultat est donné en entrée à une deuxième commande, qui elle-même donne son résultat à une autre..., on arrive à créer une « super-commande » relativement complexe, composée de plusieurs maillons très simples à la base. Nous verrons cela plus loin, à la section 5.4, après avoir étudié chaque commande séparément.

Pour chaque commande, je vais donner les paramètres qu'il faut lui donner, et je montrerai de façon systématique les exemples d'utilisation. Il faut savoir deux dernières choses :

- ce que je vais mettre entre crochets dans les syntaxes est optionnel.
- un argument est un paramètre qui est indiqué à la commande afin qu'elle sache ce qu'on veut faire. Par exemple, la commande cp (copier) accepte au minimum deux arguments : le fichier original, et sa destination.

Allez, on ne fait pas la chochotte, et on suit le guide ! N'oublie pas que le mieux est d'ouvrir un terminal (ou de passer en console plein écran avec ctrl+alt+F1[1]) et d'essayer *par toi même* ces commandes.

5.1 Gestion des fichiers en général

5.1.1 Listage, taille, arborescence

ls (list)

C'est sans aucun doute la commande qu'on utilise le plus, et à toutes les sauces. Bon appétit, bien sûr !

[1]Une fois que tu es en console plein écran, appuie sur Alt+F7 pour retourner sous l'écran graphique.

Cette commande permet d'abord de lister le contenu d'un répertoire, c'est-à-dire faire afficher son contenu. Tape « ls » tout court et tu verras le contenu du répertoire dans lequel tu te trouves. En général quand tu ouvres un terminal, tu es dans ton répertoire home.

Cette commande admet plusieurs options. Pour s'en servir, il suffit de rajouter l'option voulue (voir la liste ci-dessous) juste après la commande, séparée par un espace.

ls -l (long) permet de lister les détails, comme les permissions sur les fichiers, le propriétaire, la date et bien sûr la taille des fichiers.

ls -S (size) permet de trier les fichiers par taille de façon décroissante.

ls -t (time) permet de trier les fichiers par date, toujours de façon décroissante.

ls -r (reverse) permet d'inverser l'ordre de tri des fichiers, et peut être combiné

Exemples :

```
[christophe ~/essai]$ ls
gros_fichier moyen_fichier petit_fichier
```

On constate qu'avec « ls » sans option on ne voit que le nom des fichiers présents dans le répertoire dans lequel on se trouve. Essayons l'option « -l » pour avoir les détails :

```
[christophe ~/essai]$ ls -l
total 4632
-rw-rw-r-- 1 christop christop 4713984 déc 21 12 :31 gros_fichier
-rw-rw-r-- 1 christop christop   20460 déc 21 12 :32 moyen_fichier
-rw-rw-r-- 1 christop christop    6820 déc 21 12 :32 petit_fichier
```

Cette fois-ci on a plus de détails. Les premiers caractères « -rw-rw-r- - » indiquent les permissions sur les fichiers. On les étudiera par la suite. Après, on voit le chiffre 1, qui est le nombre de liens physiques (1 dans ce cas puisque le fichier n'est pas cloné). De même, cette notion est pour plus tard. Ce qui nous intéresse, c'est « christop » : cela veut dire que le fichier m'appartient et appartient aussi à tous ceux de mon groupe (notion également abordée plus tard). Ensuite vient la taille du fichier. Stop ! La taille n'est pas lisible : elle est indiquée en octets, ce qui est difficile à estimer pour un Homme ! On demande donc d'afficher au format « humain » avec l'option « -h » , et toujours avec l'option « -l ». On utilise donc l'option « -lh » (ou « -hl » !) :

```
[christophe ~/essai]$ ls -lh
total 4.6M
```

```
-rw-rw-r-- 1 christop christop 4.5M déc 21 12 :31 gros_fichier
-rw-rw-r-- 1 christop christop  20k déc 21 12 :32 moyen_fichier
-rw-rw-r-- 1 christop christop 6.7k déc 21 12 :32 petit_fichier
```

La première ligne indique le total (en Méga octets) du contenu du répertoire. Et la taille des fichiers devient ainsi lisible : 4.5M signifie 4.5 méga octets, k signifie kilo octets. Ensuite on a la date du dernier accès au fichier (21 décembre) et l'heure (12 :31 ou 12 :32). Tu comprends maintenant pourquoi j'ai appelé ces fichiers ainsi (vu leur taille).

Équivalent DOS : dir

cd (change directory)

Comme on l'a vu plus tôt, la commande « cd » permet de se déplacer dans l'arborescence des répertoires. Sa syntaxe est très simple :

```
cd <répertoire à atteindre>
```

Après avoir appuyé sur « entrée », on se retrouve dans le répertoire à atteindre.

Pour retourner dans le répertoire précédent, c'est-à-dire celui dans lequel on se trouvait avant de s'être déplacé, on doit taper :

```
cd -
```

Si on tape encore « cd - », on se retrouvera à nouveau dans le répertoire à atteindre du début. C'est un va et vient.

Équivalent DOS : cd

pwd (print working directory)

Affiche le répertoire où l'on se trouve, chemin d'accès complet. Par exemple, si on se trouve dans `/usr/share/doc/` (répertoire standard des documentations), on verra ceci :

```
[christophe doc]$
```

On ne voit que « `doc` », le dernier élément du chemin d'accès `/usr/share/doc/`[2].

Si je tape `pwd` :

```
[christophe doc]$ pwd
/usr/share/doc
[christophe doc]$
```

[2]On peut changer la variable « `PS1` » du shell pour qu'il affiche plus que ça : éditer le fichier « `.bashrc` » et regarder dans le manuel de Bash.

le shell me répond que je suis dans le répertoire `/usr/share/doc/` et cette fois-ci je connais le chemin d'accès entier.

Cette commande est utile bien sûr au cas où on serait « perdu » mais surtout pour élaborer des scripts.

du (disk usage)

Permet de connaître l'espace disque qu'occupe un répertoire. En fait, « du <nom_répertoire> » affiche les détails au niveau de la taille des fichiers et fait la somme de la taille de tous les fichiers contenus dans le répertoire, et dans ses sous répertoires. Vraiment utile pour connaître l'occupation du disque. Exemples d'utilisation :

```
$ du
$ du /bin
$ du -s
affiche seulement la taille du répertoire
(récursivement), celle qui est normalement inscrite à
la fin du résultat de la commande du seule.
```

Équivalent DOS : pas de commande équivalente, on peut bidouiller avec la commande dir...

df (disk free)

Permet de savoir quel est le niveau de remplissage des partitions utilisées. Si je tape « df », j'ai :

```
Filesystem    Size   Used   Avail   Use%   Mounted on
/dev/hda7     5.0G   2.3G    2.6G    47%   /
/dev/hda6    1004M    56M    948M     6%   /home
/mnt/cdrom    4.2M   4.2M       0   100%   /mnt/cdrom
/dev/hda1     5.8G   1.9G    4.0G     2%   /mnt/windows
```

Explications : La $1^{\text{ère}}$ partition est pleine à 47%, la $2^{\text{ème}}$ partition , montée en /home, occupe 1004 Mo en tout, mais est presque vide (6%), la $3^{\text{ème}}$ partition est en fait le lecteur de CD-ROMs, on remarque que ce lecteur sera *toujours* plein à 100%, même si le CD-ROM est vide. Ceci est dû au fait que le CD-ROM est accessible en lecture seule. La dernière partition est celle de windows, située en premier sur le disque dur, d'où le nom /dev/hda1.

df -a permet de faire afficher en plus l'utilisation des disquettes, et autres systèmes de fichiers.

Équivalent DOS : aucun (héhé)

tree

Permet d'afficher un arbre du système de fichiers, de façon très élégante. Exemples d'utilisation :

```
$ tree
$ tree -L x
tronque l'arbre au niveau (Level) x
$ tree -s
affiche la taille de chaque fichier. On ne peut
malheureusement pas utiliser l'affichage « humain ».
$ tree -H
génère une page HTML avec même les liens !
```

Équivalent DOS : tree aussi, mais on ne le trouve pas dans toutes les versions, et depuis windows 9x, impossible de l'avoir...

5.1.2 Copie, déplacement, suppression

Bon jusque là on n'a vu que des commandes simples. Seulement, par la suite, il arrive qu'une commande soit plus longue à s'exécuter que ce que l'on voudrait. Pour arrêter une commande en cours d'exécution (tout comme n'importe quel programme), il faut appuyer sur CTRL+C.

cp (copy)

Permet de copier un fichier. Plusieurs cas sont possibles :

– copier un fichier vers un autre répertoire mais en gardant le nom d'origine :

  ```
  cp <nom du fichier> <répertoire de destination>
  ```

  exemple :

  ```
  cp <machin> /usr/bin/
  ```

– copier un fichier vers un autre répertoire en changeant le nom d'origine

  ```
  cp <nom du fichier> <répertoire de destination>/<nouveau nom du fichier>
  ```

  exemple :

  ```
  cp <machin> /usr/bin/truc
  ```

– copier un fichier en restant dans le même répertoire en changeant le nom d'origine

```
cp <nom du fichier> <nouveau nom du fichier>
```

exemple :

```
cp <machin> <truc>
```

- copier plusieurs fichiers vers un autre répertoire. Dans le répertoire de destination, on trouvera les clones des fichiers présents dans le répertoire d'origine, avec les mêmes noms respectifs que les originaux.

```
cp <noms des fichiers> <répertoire de destination>
```

exemple :

```
cp <machin> <truc> <chose> /usr/bin/
```

- copier un répertoire vers un autre de nom différent :

```
cp -r <nom du répertoire original> [nouvel emplacement]/<nouveau nom du répertoire>
```

exemple :

```
cp -r <répertoire>/ <nouveau_rép>/
```

Équivalent DOS : copy

mv (move)

Permet de déplacer un fichier. En fait, c'est comme la commande précédente cp sauf que l'original (ou les originaux) sont supprimés. Je ne vais donc pas détailler toutes les possibilités, réfère-toi à la commande cp, mais remplace cp par mv. Attention cependant : dans le dernier cas, pour copier un répertoire, ce n'est pas la peine de taper l'option -r. Cela donne :

```
mv <répertoire>/ <nouveau_rép>/
```

mv permet, par extension, de renommer des fichiers :

```
mv <ancien_nom> <nouveau_nom>
```

Si on veut faire en sorte qu'un fichier soit caché, il faut que son nom soit précédé d'un point :

```
mv <fichier> <.fichier>
```

Ainsi, le fichier nommé « `.fichier` » sera caché.

Équivalent DOS : move

rename (renommer) on vient de voir que la commande mv permettait de renommer les fichiers. Seulement essaie un peu de renommer tous les fichiers « .htm » en « .php » pour voir comme c'est compliqué. Il faut utiliser une boucle for avec un script sed... c'est beau du point de vue du connaisseur mais un peu trop compliqué pour le débutant. Je te propose donc la commande rename, spécialement dédiée au renommage des extensions des fichiers. Il suffit de taper :

```
rename .htm .php *
```

et tous les fichiers portant l'extension « .htm » se trouvent transformés en « .php ». Le premier argument « .htm » indique l'extension à modifier, le deuxième argument indique la nouvelle extension, et le troisième argument indique les fichiers à traiter. « * » signifie : traiter tous les fichiers.

Équivalent DOS : ren

rm (remove)

Permet de supprimer un ou plusieurs fichiers. Une fois supprimés, on ne peut plus y accéder, ils n'existent plus. Il n'y a pas de corbeille ou quoi que ce soit avec la commande rm. C'est pourquoi je te conseille de ne pas supprimer un fichier à l'aveuglette, déplace le plutôt vers un répertoire « poubelle ». Encore une fois, plusieurs cas sont possibles :

- Supprimer un ou plusieurs fichiers : (si tu ne veux en supprimer qu'un seul, ne mets qu'un seul fichier en argument !)

  ```
  rm <fichier> [fichier_2] [fichier_3]
  ```

  Tu vas voir que le shell te demande confirmation pour chaque fichier avant de l'effacer. Je trouve ça pénible d'appuyer sur o pour dire oui alors il me suffit de taper :

  ```
  rm -f <fichier> [fichier_2] [fichier_3]
  ```

  et alors le shell ne demandera pas de confirmation. Attention tout de même à ne pas aller trop vite avec -f (force).
- Supprimer un répertoire et tout ce qu'il contient :

  ```
  rm -r <répertoire>/ [autre_répertoire/]
  ```

  Là encore, si tu rajoutes l'option -f, tu n'auras pas besoin de confirmer.

  ```
  rm -rf <répertoire>/ [autre_répertoire/] ...
  ```
- On peut combiner les deux possibilités précédentes : supprimer fichiers et répertoires en une seule ligne de commande :

  ```
  rm -r(f) <répertoire>/ fichier
  ```

Équivalent DOS : del

Les Alias (complément) :

Un alias est un joli tour qu'on joue au shell : on crée virtuellement des commandes, et dans ce cas « ll » signifie bel et bien « ls -l » : on a tapé « alias ll='ls -l' » et dès qu'on tapera « ll », ce sera comme si on avait tapé « ls -l ». On peut sauvegarder tous nos alias (ou plutôt les redéfinir à chaque redémarrage du shell de façon automatique) en les inscrivant dans le fichier de login de Bash, se trouvant dans « ~/.bashrc ». Pour connaître *tous* les alias en cours, taper « alias » tout court. Il faut bien sûr tous les essayer. En pratique, les alias servent à placer des paramètres automatiquement au lancement des programmes, ou à définir de nouvelles commandes.

5.1.3 Compression

tar

Cette commande permet de rassembler plusieurs fichiers en un seul, dans le but de faciliter les sauvegardes. Il faut savoir que lorsque l'ordinateur sauvegarde des fichiers sur disquette, il va être bien plus rapide s'il doit sauvegarder un seul et gros fichier que plusieurs petits dont la somme ne dépasse pas la taille du gros. Ce n'est pas diablerie, c'est simplement dû au fait que chaque fichier doit être « ouvert », écrit, puis « fermé » sur la disquette (ou autre support). L'ordinateur doit écrire sur la disquette pour chaque fichier : « il va y avoir un fichier », écrire le contenu du fichier, puis « le fichier est terminé ». Rebelote pour le fichier suivant. On a donc à chaque fois deux étapes d'ouverture et de fermeture qui correspondent au bruit que fait le lecteur lors des sauvegardes. De ce fait, en concaténant tous les petits fichiers en un seul gros, on réduit ces deux étapes et la sauvegarde n'est que plus rapide.

Cependant, on utilise le plus souvent la commande tar de façon détournée, c'est-à-dire avec compression. En fait, au lieu d'obtenir un fichier « xxx.tar », on aura « xxx.tar.gz » (compression gzip) ou « xxx.tar.bz2 » (compression bzip2). Voir la section 8.2 pour plus de précision sur l'efficacité des différents types de compression.

Quelques exemples ? On a 5 fichiers : un, deux, trois, quatre et cinq.

- Pour *créer* un fichier archive compressé avec gzip, nommé « groupe.tar.gz », on tape

  ```
  tar cvzf <groupe>.tar.gz <un> <deux> <trois> [ENTRÉE]
  ```

  Le c signifie « créer » (une archive), le v (verbose) permet d'afficher les détails de l'opération à l'écran, le z permet de « gziper » l'archive tar

  et le f (fichier) nous permet de choisir un nom d'archive compressée, en l'occurrence groupe.tar.gz. L'ordre des lettres c, v, z et f n'a pas d'importance. On voit dans les documentations tous les ordres possibles, mais cet ordre là est quand même le plus courant, car il est logique : on veut créer, (afficher), gziper et mettre tout ça dans le fichier machin.tar.gz. Ensuite on dit quels fichiers y figureront, dans cette archive.
- De même, pour *créer* une autre archive compressée avec bzip2, nommée « groupe.tar.bz2 », on tape

```
tar cvjf <groupe>.tar.bz2 <un> <deux> <trois> [ENTRÉE]
```

  Remarque que le z dans le cvzf de tout à l'heure s'est transformé en j, car le j indique de compresser avec bzip2. Pourquoi un j, je l'ignore, c'est sûrement arbitraire.

Au fait, pour *ne pas* compresser, il suffit de ne pas utiliser l'option z ou j. Et n'oublie pas, tape « man tar » pour toutes les options possibles ! (voir la section 6.1 pour savoir comment trouver toi même de la documentation sur les commandes).

5.2 Avec les fichiers texte...

Cette section va être beaucoup plus amusante pour toi, je t'assure ! Tu vas vraiment pouvoir essayer ces commandes, et tu t'en serviras tout le temps.

Déjà, avant toute chose, autant dire que pour manipuler des fichiers texte, on est obligé de disposer de fichiers texte (incroyable, quelle perspicacité) ! Nous allons donc créer une petite poignée de fichiers texte, quatre ou cinq par exemple, afin de travailler avec. Au fait, un fichier texte est un fichier qui contient... du texte (je crois que certains se sentent encore une fois pris pour des ânes, désolé !). Eh bien voilà, tu peux lire la suite, pour savoir comment créer des fichiers texte !

5.2.1 Introduction

Pour créer un fichier texte, rien de mieux qu'un logiciel tout simple, qui va éditer des fichiers texte, et qui s'appelle... un éditeur de texte[3]. Pourquoi un « éditeur » de textes ? Parce que d'habitude on s'en sert pour modifier un texte déjà existant mais on peut bien évidemment en créer de toutes

[3] Si c'est déjà trop dur pour toi, reprends le document demain après une bonne nuit de sommeil.

pièces ! En plus, ça ferait vraiment débile comme nom « créateur de texte », on croirait que c'est le logiciel qui les crée, alors que c'est nous ! Non mais !

Il existe une multitude d'éditeurs de textes, c'est à peine s'ils sont dénombrables (j'exagère à peine...). Quelques exemples les plus connus ? Allons-y :

- « edit » sous Ms-DOs
- « bloc notes » sous windows
- « ultra edit »

Voilà ce que l'on connaît lorsqu'on émerge de la pouponnière de Monsieur Billou le filou Gâteux, expert en conditionnement moral, lavage de cerveau, découragement de l'utilisateur... Pour avoir une liste complète de tous les inconvénients qu'apporte cet individu, contactez votre revendeur, mais d'abord faites control+alt+suppr, car votre écran souffre à force de clignoter entre les deux écrans bleus.

Et lorsqu'on s'immerge dans le monde du logiciel libre, seulement deux éditeurs de texte émergent : les deux premiers (vi et Emacs) ; les autres sont secondaires :

- vi : je te présente blanc bonnet. (prononcer vi-aille et surtout pas véhi !)
  - vim : vi meilleur, c'est-à-dire qu'il est mieux, quoi ! Il permet notamment la coloration syntaxique, et l'indentation automatique des codes source (assistances à la programmation)
  - gvim : frontend pour vim
- Emacs : je te présente bonnet blanc . Emacs possède des extensions que ne possèdent pas V.I.
  - xemacs : frontend pour Emacs.
- joe : un autre éditeur
- kedit : l'éditeur de KDE
- gedit : l'éditeur de Gnome
- et plein d'autres comme GNotepad+, Cool Edit, Nedit, Yudit, kwrite...

En tout cas tous ces éditeurs sont largement plus performants que le bloc notes !

vi et Emacs se valent, et c'est très difficile de dire lequel est le meilleur. Personnellement, je préfère vi, mais je n'ai pas d'argument : je n'ai jamais essayé Emacs. La rumeur court que Emacs est quand même plus confortable et assisté que vi pour des gros travaux[4]. Je n'ai jamais fait de gros travaux et on ne va pas en faire, donc c'est décidé, on va utiliser vi. En plus, c'est l'éditeur de textes que j'utilise à l'IUT pour mes études, alors...

Il faut noter que vim est une extension de vi : ils ont les mêmes commandes, mais vim en possède plus, inclut plus de fonctionnalités... donc si tu as vim, préfère-le à vi. Il faut aussi savoir, pour l'aspect culturel, que vi et

[4] je l'ai lu dans Paris Match

Emacs existent depuis très longtemps, je crois depuis un peu plus de 20 ans ! C'est dire s'ils sont exempts de bugs aujourd'hui. Retiens que vi, est l'éditeur de texte avec lequel on va le plus vite : il suffit de frapper quelques touches et hop ! on obtient le résultat voulu. Pas besoin de la souris, c'est une perte de temps. Bien sûr celui qui ne connaît pas les indispensables raccourcis clavier trouvera que vi est trop dénudé et préfèrera la souris. Tu es prévenu !

Passons donc à la pratique. J'adore les légumes, et ce serait un plaisir de travailler avec eux. Tu vas donc taper :

```
vi carotte
```

pour lancer vi et créer le fichier nommé carotte. Maintenant que tu es dans vi, tu dois remplir le fichier avec les phrases que je vais te donner. Il suffit de se mettre en mode insertion avec la touche « i » et de taper « J'aime les carottes, surtout quand elles sont bien cuites. ». Pour sauvegarder puis quitter, tape « ESC » (ou Echap selon les claviers) pour sortir du mode insertion puis « `:wq` » suivi de « Entrée ». Tu te retrouves dans le shell, et il ne nous reste plus qu'à voir si le fichier « carotte » a bien été sauvegardé.

Je vais t'indiquer toutes les manipulations nécessaires à nos exemples pour utiliser vi mais je ne vais pas tout inventorier. Ce serait stupide de réécrire un manuel pour vi, car il en existe déjà énormément. Je te conseille de lire le mini livre des éditions O'Reilly sur vi, dans la série « précis et concis », ou de lire la page html très bien faite, en français, qui est un tutoriel pour vim. `http://www.student.info.ucl.ac.be:8080/HomePages/fleurial/mai.com/vim-tutor.html`

5.2.2 Affichage et navigation

Maintenant, on veut vérifier que carotte a bien été enregistré, en tapant « cat carotte ».

cat (concaténer)

Cette commande est fondamentale dans tout système Linux. Ses possibilités sont assez limitées, mais combinée à la magie du Shell, elle peut faire des merveilles :

- afficher le contenu d'un fichier quelconque (les fichiers texte par exemple) sur la sortie standard, donc l'écran de la console
- réunir plusieurs fichiers en un (des fichiers texte, mais aussi des sons wave !)
- numéroter rapidement les lignes d'un fichier
- extraire une image disque d'un disque dur, disquette, CD-ROM...

Revenons à nos carottes :

`cat <fichier>` : permet d'afficher le contenu d'un fichier, et donc de visualiser s'il s'agit d'un fichier texte

donc « cat carotte » devrait faire afficher « J'aime les carottes, surtout quand elles sont bien cuites. ».

`cat <fichier_1> <fichier_2>...` : permet d'afficher à la suite les fichiers mentionnés en argument.

Cette fois, pour pouvoir expérimenter, on a besoin d'un autre fichier que l'on nommera « patate ». Maintenant que tu sais le faire, tu crées avec vi un fichier « patate » qui contiendra le texte suivant : « Je n'arrive pas à éplucher les patates lorsqu'elles sont brûlantes. ». Si tu ne te rappelles plus comment on crée un fichier texte, va voir la sous-section 5.2.1.

On a donc deux fichiers nommés carotte et patate.

– Si tu tapes « cat carotte patate », tu auras :
```
J'aime les carottes, surtout quand elles sont bien cuites.
Je n'arrive pas à éplucher les patates lorsqu'elles sont
brûlantes.
```

– Si tu tapes « cat patate carotte », tu auras :
```
Je n'arrive pas à éplucher les patates lorsqu'elles sont brûlantes.
J'aime les carottes, surtout quand elles sont bien cuites.
```

C'est tout simple. Un dernier petit truc, une ruse de « mémé nez crochu qui connaît tout » : si tu tapes « cat fichier » suivi de « echo » pour insérer une ligne vide puis encore « cat fichier2 » etc... le tout redirigé vers un autre fichier, tu pourras concaténer des fichiers avec une ligne vide (ou autre motif comme des lignes étoilées par exemple) de manière superbe !

`cat -n <fichier_1>` : permet d'afficher, comme dans le premier cas, un fichier (on peut d'ailleurs en mettre plusieurs en argument), mais en numérotant les lignes.

`cat -v <fichier_1>` : Affiche également les caractères non imprimables (on ne les voit pas mais ils existent tout de même).

Équivalent DOS : type, mais type est bien moins intéressant.

more (plus)

Permet, comme cat, d'afficher un fichier texte (ou autre, encore une fois), mais à la différence près que si le texte dépasse l'écran, more va l'arrêter au bout du premier écran, et quand tu appuieras sur espace, tu verras la suite, etc... Nous passerons comme une flèche sur cette commande car la suivante est bien mieux.

Équivalent DOS : more

less (moins !)

DOS a more et Linux a less. On notera l'humour linuxien très modeste, qui a transformé more en less (plus en moins), car less est mieux que more !

less permet d'afficher un fichier et de naviguer dans ce fichier. S'utilise très simplement :

```
less <fichier>
```

Ensuite, avec les flèches du clavier, tu descends ou remontes les lignes, avec page up et page down tu fais défiler d'un écran entier. Et même les flèches droite et gauche fonctionnent, ce qui ne sert presque jamais...

Pour rechercher une chaîne de caractères dans le fichier, tape « `/` » puis la chaîne à rechercher. Ensuite, less t'amène à la première occurrence. Pour arriver à l'occurrence suivante, tape « `n` ». Pour retourner à l'occurrence précédente, tape « `N` ».

Pour aller au début du fichier, tape « `<` » et pour aller à la fin du fichier, tape « `>` ».

Pour sortir de less, tape « `q` ».

Équivalent DOS : aucun

Maintenant tu sais te débrouiller avec les fichiers texte, au moins en créer et en afficher. On va passer aux choses plus sérieuses.

echo

Affiche une chaîne de caractères à l'écran. « echo azerty » va retourner azerty ! Avec echo, on peut faire afficher des variables d'environnement comme le PATH (chemin par défaut dans lequel le shell va chercher en premier les programmes. C'est pour cela qu'on n'a pas besoin de taper /bin/ls pour ls !). Il suffit de taper « echo $PATH ».

Echo affiche des chaînes de caractères et cat des fichiers...

Quelque chose d'intéressant :

```
echo « La baleine est grosse » > baleine.txt
```

Cette commande crée un fichier texte appelé baleine.txt et qui contient une seule ligne : « La baleine est grosse ». On évite de passer par vi !

grep

Permet de rechercher les occurrences d'une chaîne de caractères dans un fichier (de façon très puissante). Utilisation vraiment de base :

```
grep <chaîne à rechercher> <fichier_1> [<fichier_2>]
```

L'option -i ignore la casse (ignore les différences entre les majuscules et les minuscules).

N'hésite pas à faire un grep - -help pour plus de renseignements. Tu peux aussi rechercher de la documentation sur les « expressions régulières sous Linux », car grep et d'autres commandes également les utilisent. Nous n'avons pas ici pour objectif de rentrer dans les détails.

Équivalent DOS : hmm... grep introduit le concept d'expression rationnelle (nous ne le développerons pas ici), qui est très riche et c'est sûrement la meilleure façon de décrire des expressions. Sous DOS, ces expressions rationnelles n'existent pas.

split (couper)

On peut avoir envie de découper un fichier en plusieurs morceaux plus petits, par exemple si le fichier d'origine est trop gros pour tenir sur une seule disquette. On veut donc découper dans ce cas le gros fichier en morceaux de 1400 Ko (la disquette est en théorie capable de recevoir 1440 Ko mais par précaution on ne l'utilisera pas en entier). Nous avons donc la commande split à notre disposition pour effectuer ce travail très simple :

```
split -b 1400k <gros_fichier>
```

Les petits fichiers générés s'appellent xaa, xab, xac... En général cette commande est utilisée pour découper des archives « .tar.gz » ou « .tar.bz2 ». Voir la section 5.1.3 pour plus de renseignements à propos des archives.

fold

Cette commande est très utile pour imprimer des fichiers texte. Les imprimante ne peuvent généralement pas dépasser 80 caractères par ligne. Si tu veux imprimer un texte qui comporte des lignes de plus de 80 caractères, il suffit d'utiliser la commande fold et elle te renverra à l'écran ton texte « coupé » au $80^{\text{ème}}$ caractère :

```
fold texte.txt
```

Le problème est que les mots sont coupés n'importe où : il suffit d'utiliser l'option « -s » et fold coupera les lignes *entre* les mots :

```
fold -s texte.txt
```

Et bien sûr, si tu veux imprimer, il faut renvoyer la sortie de fold vers la commande « lpr » :

```
fold -s texte.txt | lpr
```

5.3 Administration système

L'administration système est la discipline d'un administrateur système, qui est la personne qui s'occupe d'un ensemble d'ordinateurs. Cependant, même un ordinateur isolé fonctionnant avec un système d'exploitation multi-utilisateurs, même le mien ou le tien, a besoin d'un administrateur système. C'est une possibilité qui permet d'accroître de façon exponentielle la sécurité du système. Bon, je m'enlise, et je sais très bien que pour la plupart des gens, l'administrateur système est quelqu'un d'autre qu'eux par définition. C'est vrai : rien que le mot fait peur !

Alors soyons simples : j'ai parlé de l'utilisateur root depuis le début de ce document. Si tu t'es déjà identifié comme root, alors tu es l'administrateur système ! On l'appelle aussi le super-utilisateur[5]. Être administrateur système pour un poste isolé n'est pas si difficile : son rôle consiste simplement à attribuer les permissions aux utilisateurs sur les fichiers[6], déclarer les utilisateurs et les groupes auxquels ils appartiennent éventuellement, installer des programmes et régler le matériel... cela ne se fait pas tous les jours ! Et encore moins pour un ordinateur personnel. De toutes façons, root a le droit de *tout* faire, sans restrictions. C'est pour ça que c'est très dangereux de s'identifier comme root.

Nous allons donc voir à présent les commandes réservées à root, je rappelle qu'elles sont situées dans le répertoire /sbin ou /usr/sbin. J'arrête de donner les équivalents DOS car ils n'y en a plus aucun à partir d'ici.

su

Cette commande est essentielle car elle permet d'enfiler la peau d'un nouvel utilisateur. On s'en sert dans la plupart des cas pour devenir root depuis notre identifiant normal. Par exemple, je m'identifie toujours sous christophe, jamais sous root pour des questions de sécurité, et quand j'ai quand même besoin des privilèges root, je tape :

```
[christophe textes]$ su
Password : (je le tape)
[root textes]#
```

et voilà je suis root ! Le $ du shell s'est transformé en # (dièse) et christophe s'est transformé en root. Et là je suis capable de tout, et même du pire, comme

[5]N'oublie jamais que s'il y a un problème, il suffit de siffler super-utilisateur qui arrivera dans la seconde qui suit vêtu de sa cape et de son masque.

[6]à l'IUT les permissions permettent par exemple d'interdire aux élèves de lire les fichiers contenant les corrigés des exercices mais aussi de les autoriser à lire les énoncés.

effacer tout le système en entier sans aucune restriction. Ne te connecte sous root que lorsque tu es dans ton état normal !

Maintenant on va voir quelles commandes on peut utiliser, commandes qui nous étaient interdites jusqu'à présent. Attention, attention : on entre dans le cœur du concept multi utilisateurs et les windowsiens vont sentir leur ventre se retourner, s'écriant Samuéliquement[7] : « Mais c'est gé-ni-al !!! ».

5.3.1 Gestion des processus

Quand on veut utiliser un programme, la première chose que l'on fait est de l'exécuter. Soit on le choisit dans le menu de l'environnement graphique (menu K pour KDE, le pied pour Gnome, Démarrer pour windows), soit on clique sur son icône, soit on tape la commande dans la console. Bien sûr, le mieux est la console, surtout pour bien comprendre les choses.

Quand on exécute un programme, on crée par la même occasion un processus. Surtout, il ne faut pas entendre le mot au sens français, mais il faut juste penser que le noyau gère en même temps une liste de processus (le multi-tâches, le voilà !) et qu'un programme exécuté crée un élément de plus dans cette liste des processus.

Si un programme plante et se gèle (c'est-à-dire qu'il ne répond plus à aucune action de l'utilisateur), il suffit de « tuer » le processus qui lui est associé pour que le noyau libère toute les ressources (mémoire) qui étaient utilisées par le programme et ainsi le programme disparaît. Sous windows, le problème est que la plupart du temps le plantage d'un programme entraîne le plantage du noyau de windows ! Un comble !

ps (processus)

Cette commande permet de lister les processus en cours. ps tout seul n'affiche presque rien. Il faut lui faire cracher pour de bon la liste en utilisant les option a, u et x.

```
ps aux
ps auxw
```

L'ordre des lettres a, u, x et w n'a aucune importance, c'est simplement qu'il faut bien choisir une combinaison !

Le w permet de faire afficher en entier le nom du processus car parfois ça déborde à droite de l'écran. Par contre c'est pénible parce qu'il faut lire une ligne sur deux.

[7] Désolé Samuel, j'ai pas pu m'en empêcher !

Chaque processus listé a un numéro qui lui est associé : le PID (Processus IDentifier). Ce numéro est affiché dans la deuxième colonne à gauche. Le PID est important car il va nous servir directement à désigner un processus.

kill (tuer)

Tuer un processus n'est pas chose cruelle. C'est tout à fait normal comme procédé, lorsque le programme associé au processus refuse d'obéir. En tuant un processus on assainit le système.

Imagine qu'un programme plante. Mais pour de bon. Il ne réagit plus du tout. Alors tu ouvres un terminal, tu tapes « ps aux », tu repères le PID du programme qui plante, tu tapes

```
kill <PID>
```

et c'est fini, le programme disparaît définitivement. Si ce n'est pas le cas et qu'il est très coriace, envoie-lui le violent signal 9 avec

```
kill -9 <PID>
```

et là ses chances de survie sont quasi nulles. S'il résiste, retape « kill -9 <PID> » plusieurs fois.

Nous n'étudierons pas les signaux qu'il est possible d'envoyer aux processus, mais sache que le signal 9 est très violent : il permet de terminer un programme bien plus efficacement que le signal par défaut.

Attention tout de même, tu ne peux tuer que les programmes que *tu* as exécutés, pas ceux exécutés par root par exemple. A ce moment là, il suffit de faire « su » et tu peux tuer n'importe quel programme. L'utilisateur qui a lancé un processus est affiché dans la première colonne à gauche quand tu fais « ps aux ». Si tu vois 501 ou quelque chose, tape « cat /etc/passwd » et repère le numéro du genre de 501 pour voir qui est associé à 501. Eh oui, chaque utilisateur a lui aussi un numéro !

killall (tuer tous)

Tu trouves cela pénible de localiser le PID d'un programme ? Dans certains cas ne n'est pas la peine ! Si tu connais le nom du programme à tuer, tape

```
killall <nom du programme>
```

ou même le signal 9 (tue le processus de façon violente)

```
killall -9 <nom du programme>
```

et la vie de tous les programmes portant ce nom s'arrêtera là.

Quelquefois, cette commande ne fonctionne pas, et il faut utiliser « ps aux » suivi de « kill <PID> ».

5.3.2 Premier plan et arrière plan

Lorsque tu exécutes un programme en console, tu peux soit le garder au premier plan, et à ce moment là tu ne pourras plus rien faire (sauf changer de terminal), soit le mettre en arrière plan, et continuer le travail pendant l'exécution du programme.

Si tu tapes :

```
cp <gros fichier> ~/
```

tu devras attendre relativement longtemps que le gros fichier soit copié vers ton répertoire maison.

Par contre si tu tapes :

```
cp <gros fichier> ~/ &
```

la copie se fera en arrière plan et tu pourras continuer à taper d'autres commandes.

Tout va bien jusque là. Seulement, si tu ignorais la taille si importante du fichier et que, pendant la copie, tu en avais marre, tu aurais deux solutions :
– arrêter la copie avec Ctrl+c puis reprendre la copie en agrémentant la commande d'un &.
– mettre en pause la copie avec Ctrl+z et l'envoyer en arrière plan avec bg.

On va expliciter tout ça :

&

Ce n'est pas une commande mais une fonctionnalité du shell. Cependant j'ai pensé qu'il fallait placer & dans cette section pour mieux comprendre les commandes suivantes.

& permet, lorsqu'il est placé en fin de ligne de commande, de lancer un programme directement en arrière plan. Très pratique pour les programmes graphiques ou encore pour les programmes qui n'affichent rien mais qui sont longs. Essaie et tu comprendras :

```
gedit &
```

par rapport à :

```
gedit
```

jobs (travaux)

Affiche la liste des travaux en arrière plan. Normalement il n'y en a aucun. Par contre, si tu exécutes un programme avec & à la fin de la ligne, comme par exemple « gedit & », alors « jobs » va afficher :

```
[1]+ Running gedit &
```

Si tu veux faire revenir le processus gedit au premier plan, tape « fg 1 » puisque le numéro du job est 1. Si maintenant tu appuies sur Ctrl+z, il est mis en pause : gedit ne réagit plus du tout. A partir de là, on a le choix : soit on le remet au premier plan avec « fg » soit on le met en arrière plan avec « bg ».

bg (background)

Permet d'envoyer un travail en arrière plan.

```
bg <numéro de job>
```

Le numéro de job est facultatif, mais si tu ne mets rien, la commande bg traitera le dernier job traité.

fg (foreground)

Permet de faire revenir un travail au premier plan, ou encore de réactiver un programme qui serait mis en pause.

```
fg <numéro de job>
```

De même, le numéro de job est facultatif, mais si tu ne mets rien, la commande fg traitera le dernier job traité.

5.3.3 Permissions

Les fichiers ont des permissions, dans le but d'éviter les catastrophes, comme l'effacement des fichiers système importants ou encore le piratage des documents précieux. Je rappelle qu'on peut voir les permissions en tapant « ls -l » ou l'alias ll.

Il existe trois types de permissions : lecture (read ou encore « r »), écriture (write « w ») et exécution (execute « x »).

- Ces permissions, appliquées à des fichiers, signifient :
  - r : on peut lire le contenu du fichier
  - w : on peut modifier le contenu et le nom du fichier
  - x : on peut exécuter un fichier (à réserver pour les programmes, car exécuter un fichier texte par exemple n'a aucun sens !)
    Le droit d'exécution est nouveau pour un utilisateur de windows. C'est simple à comprendre. Bon, par exemple, les fichiers texte n'ont pas besoin d'être exécutés. On n'a besoin que de les lire ou de les écrire, car ce ne sont pas des programmes. Le terme exécuter n'est

pas à prendre au sens de l'assassinat ! « Exécuter un programme » veut dire qu'on décide à l'instant voulu que les instructions qui sont dans le programme vont être effectuées par l'ordinateur. Quand on lance un programme, on dit qu'on l'exécute. Un programme qui n'a pas le droit d'exécution ne pourra pas être transmis à l'ordinateur, il faudra lui donner ce droit d'exécution.

La permission x, appliquée à des répertoires, est légèrement différente : elle signifie qu'on peut se placer dans le répertoire en question.

Et là où c'est génial, c'est que ces permissions peuvent être différentes pour trois catégories d'utilisateurs : le propriétaire du fichier, le groupe propriétaire du fichier, ainsi qu'à tous les autres utilisateurs. Par exemple, pour un fichier texte, le propriétaire du fichier a le droit de lecture et d'écriture, le groupe du propriétaire du fichier a le droit de lecture seulement et les autres ne peuvent ni lire ni écrire. Dans ce cas le schéma « rwx » serait : « `rw-r-----` ».

chmod (change access mode)

Permet de changer les permissions accordées aux utilisateurs (3 classes : le propriétaire du fichier, son groupe et tous les autres utilisateurs) de deux façons différentes.

- Méthode des lettres :
  On désigne une catégorie d'utilisateurs
  - u pour le propriétaire du fichier (user)
  - g pour son groupe (group)
  - o pour tous les autres (others)
  - a pour tout le monde, c'est-à-dire u+g+o (all)

  puis le signe de l'opération
  - + pour ajouter une permission
  - - pour l'enlever
  - = pour ne pas changer (mais à ce moment là autant ne pas l'utiliser !)

  et pour finir quelle permission
  - r pour lecture (read)
  - w pour écriture (write)
  - x pour exécution (execute)

  On peut combiner plusieurs opérations, et on change de catégorie d'utilisateurs avec une virgule.
  Quelques exemples :

  ```
  chmod a+r+x,u+w <fichier(s)>
  ```

  permet d'attribuer à tout le monde les droits de lecture et d'exécution au fichier, et à l'utilisateur le droit d'écriture. Notons que si jamais les autres avaient déjà le droit d'écriture, cette commande ne le leur enlèverait pas. Il faudrait taper :

```
chmod a+r+x,u+w,o-w <fichier(s)>
```

pour être sûr que les autres utilisateurs ne peuvent pas modifier le(s) fichier(s).
Cette méthode est la plus intuitive mais on se rend compte que pour tout imposer, c'est-à-dire ne rien laisser comme avant comme permissions, il faut tout mentionner et c'est long à taper. La méthode des chiffres est meilleure dans ce sens. Par contre, pour une petite modification, comme donner le droit d'écriture au propriétaire, en se fichant du reste des permissions, cette méthode est la meilleure. Tu comprendras mieux en lisant la méthode des chiffres.

– Méthode des chiffres :
Cette méthode est plus intéressante pour imposer toutes les permissions d'un coup en trois chiffres seulement. Par contre elle est pénible pour modifier juste une seule permission (voir alors la méthode des lettres).
Il faut savoir que :
  – la permission d'exécution correspond à 1
  – la permission d'écriture correspond à 2
  – la permission de lecture correspond à 4

  Toute somme de ces trois chiffres correspond aux permissions des chiffres ajoutés. Par exemple, 3 correspond à la fois à la permission d'exécution et à celle d'écriture. Il faut avoir remarqué que le chiffre 3 ne peut s'obtenir *uniquement* par la somme de 1 et de 2 ! De même, 7 correspond à toutes les trois permissions ou encore 6 correspond à l'écriture et la lecture.
  Chaque catégorie d'utilisateurs (propriétaire, son groupe, et tous les autres) reçoit un chiffre selon les permissions qui lui sont accordées. On a donc à chaque fois, avec chmod, un groupe de 3 chiffres qui vient en argument. Par exemple, si on veut rendre un fichier texte lisible et modifiable par son propriétaire, seulement lisible donc pas modifiable pour le groupe et les autres, et que l'exécution soit interdite pour tout le monde... quel calvaire avec la méthode précédente ! Ici, il suffit de réfléchir une petite seconde : pour le propriétaire, on veut lire et écrire et pas exécuter, donc 6. Pour le groupe et les autres on ne veut que lire (donc pas écrire ni exécuter) donc 4 pour les deux. Le nombre à trois chiffres est 644. On tape donc :

```
chmod 644 <fichier(s)>
```

  Si on avait voulu faire la même chose avec la méthode des lettres, il aurait fallu taper :

```
chmod u+r+w-x,g+r-w-x,a+r-w-x <fichier(s)>
```

ce qui est bien plus long. Surtout que dans la plupart des cas, on utilise 644 pour un fichier non exécutable et 755 pour un fichier exécutable. A connaître et surtout comprendre : 644 et 755.

L'option -R permet d'appliquer chmod récursivement. Je rappelle que cela signifie que tous les sous répertoires et tout ce qu'ils contiennent sont traités, et pas seulement le répertoire en cours. Une option très utile.

chown (change owner)

Permet de changer le propriétaire d'un fichier. Pour pouvoir le faire, il faut être, bien sûr, soit le propriétaire soit root. Si je n'ai plus envie de travailler sur tel fichier, je déclare que son propriétaire est Gaston et non plus moi même.

```
chown <nouveau_propriétaire> <fichier(s)>
```

chgrp (change group)

Permet de changer le groupe d'un fichier. Les groupes sont utiles dans les systèmes avec beaucoup d'utilisateurs comme dans le scolaire, où on crée un groupe regroupant tous les profs, un autre tous les élèves de première année, un autre tous les élèves de seconde année, un autre tous les élèves en général...

On l'utilise ainsi :

```
chgrp <nouveau_groupe> <fichier(s)>
```

Les groupes sont définis dans le fichier de configuration « /etc/group ».

init Change de niveau d'exécution en cours. On les a rapidement évoqués à la section 4.4.1.

Par exemple, si je veux éteindre le mode graphique, je dois passer en niveau d'exécution 3 (c'est la convention) :

```
[root ~]# init 3
```

Si je veux repasser en graphique, je passe au niveau 5 (on remplace 3 par 5).

Mais cette commande n'est vraiment nécessaire que pour des opérations de maintenance. Ne t'en sers pas pour taper des commandes ! Si tu veux accéder à une console, soit tu en ouvres une depuis KDE ou Gnome, soit tu passes en plein écran texte avec Ctrl+Alt+F1 (et Alt+F7 pour retourner en mode graphique).

Cette commande n'est accessible qu'au root.

5.4 Les possibilités du shell

Elles sont simples à maîtriser, le plus intéressant est qu'en les combinant on peut faire des merveilles !

5.4.1 Les redirections

Les redirections sont très pratiques. La première chose qu'elles permettent est de capturer ce qui aurait dû être affiché à l'écran, et de le stocker dans un fichier. En fait, les commandes de Linux ne disent pas explicitement au système d'envoyer leur résultat à l'écran. Non, ce serait trop compliqué. Au lieu de cela, les commandes envoient leur résultat vers un fichier spécial. Ce fichier spécial est lui-même lu en permanence par le système, et c'est le pilote de la carte vidéo qui affichera le contenu de ce fichier spécial sur l'écran. On dit que ce fichier spécial est la sortie standard. Grâce à ce stratagème, les commandes s'adaptent à n'importe quel type de matériel, et il suffit seulement de programmer un nouveau pilote pour un nouveau matériel pour que tout le système fonctionne. Mais ce n'est pas le moment de parler de ça, on se focalise sur les redirections. La redirection de sortie demande à la commande exécutée d'envoyer son résultat vers un fichier différent du fichier spécial. Un exemple d'application : tu as un cdrom dans ton lecteur et tu veux stocker son contenu dans un fichier pour pouvoir le connaître plus tard sans avoir à l'insérer dans le lecteur, tape

```
tree>fichier
```

et pour vérifier tape « less fichier » ! Maintenant tu n'auras plus besoin d'insérer ton cdrom pour connaître son contenu.

Nous venons de voir la redirection de sortie. Il en existe d'autres, par exemple la redirection d'entrée. L'entrée standard sur un ordinateur est... le clavier ! Si on détourne l'entrée standard, c'est-à-dire si on lit les entrées depuis un fichier et non plus depuis le clavier, la commande fonctionnera sans qu'on ait besoin de taper quoi que ce soit. Par exemple :

```
mail machin@fournisseur.fr < email.txt
```

enverra un email et le corps de l'email sera email.txt, alors que sans la redirection d'entrée on doit taper le mail à la main (on doit même indiquer la marque de fin de fichier avec Ctrl+D). Ne panique pas si tu ne connais pas la commande mail, c'est juste un exemple.

Troisième redirection intéressante : la double redirection de sortie, notée $>>$. C'est comme la redirection de sortie simple, mais à la différence près qu'on rajoute au lieu de remplacer. Par exemple :

```
ls > fichier.txt
cd rep
ls >> fichier.txt
```

Au final, ce fichier contiendra les résultats des deux commandes ls, dans l'ordre chronologique. Si on avait omis la double redirection, la deuxième commande ls aurait écrasé le contenu du fichier, qui ne contiendrait à la fin plus que le résultat de la seconde commande ls.

C'est tout pour le moment, on n'a pas besoin d'en savoir plus pour débuter.

5.4.2 Les pipes

Au fait... prononcer « païpe » comme le mot « paille » puis rajouter « p ». Pipe veut dire tuyau en anglais. On en déduit que piper des commandes signifie tuyauter des commandes entre elles. C'est simple ! Imagine qu'une commande soit une pompe à essence (on voit déjà les utilisateurs de gazole rouspéter...) et qu'une autre commande soit une voiture (ou un autre véhicule fonctionnant aux hydrocarbures). On doit relier la pompe à la voiture. Bravo Mongolito.

Eh bien les commandes Linux c'est pareil. La première fournit quelque chose (du texte, des données, etc...) et la seconde utilise ces mêmes données comme entrée, comme paramètres. Au lieu que ce soit toi qui donnes les paramètres à la deuxième commande, c'est la première qui le fait. Et bien sûr, pour relier les deux commandes, on utilise... un tuyau ! Le caractère du clavier qui symbolise un tuyau est le caractère pipe (tiens tiens comme on s'retrouve), celui qui ne sert jamais, comme disent les windowsiens. Il se situe sur la touche 6, au dessus de la touche Y, et on l'obtient en tapant « Alt Gr » + 6. Un joli caractère comme ça : « | » apparaît.

Maintenant, l'illustration de tout ce baratin :

```
$ cat texte
```

Joli sonnet que j'ai écrit
Le printemps est là
Les fleurs ont fleuri
Et l'huissier n'est plus là.

Linux est très performant
Linus Torvalds est excellent
Et Richard Stallman vibre
Pour répandre le logiciel libre.

GNU, il y a consacré sa vie

Pour obtenir un système de luxe.
Je serais vraiment triste si
Je cessais d'étudier GNU/Linux,
Ô ! Perspectives horribles !
Ah ! L'huissier revient mais point de logiciels piratés il ne saisira.

P.S. : Chers lecteurs, ne m'envoyez plus de compliments pour ce poème, j'en reçois déjà trop !

```
$ cat texte|grep luxe
Pour obtenir un système de luxe.
```

Voilà ! Grep (commande déjà étudiée) a utilisé en entrée le texte (qui a été envoyé à grep et non à l'écran) et a affiché le résultat. Bon, cet exemple est mal choisi puisqu'on aurait pu faire : « `grep luxe texte` ». Autre exemple :

```
rpm -qa|less
```

là oui, le pipe est utile : rpm -qa liste les paquetages rpm installés et less permet de visualiser cette longue liste, sans qu'elle déborde de l'écran. Pas besoin de rediriger la sortie de rpm -qa vers un fichier et de faire less fichier !

J'espère que ce concept est compris.

5.5 Un petit topo sur les scripts...

Ah ! Les scripts...

Ils sont tellement utiles, agréables à concevoir, souples... que c'est à eux que j'aurais dû dédier mon sonnet !

Les scripts servent à programmer *très rapidement* des programmes interprétés (non compilés), afin de répondre à tes besoins. Par exemple, j'ai écrit des scripts permettant d'imprimer des documents sous la forme d'un livre (voir section 8.7), pour rechercher un fichier donné dans un lot d'archives RPM, pour copier un CD audio de façon automatique, pour dézipper un zip venant de windows dans un répertoire à part, prendre une photo avec mon appareil numérique et la mettre dans un répertoire en incrémentant le numéro de la photo, et c'est tout pour le moment. Toutes ces tâches sont effectuées de façon automatisée. Et je m'en sers souvent, de mes scripts.

L'avantage du script Bash est qu'on ne doit pas apprendre de syntaxe spéciale, car on programme en utilisant la même syntaxe que lorsqu'on entre les commandes dans le shell. Si tu connais le vieux langage de scripts Batch de MS-DOS, tu peux entrevoir ce qu'est le script Bash. Le script Bash est au Batch ce que le shell Bash est à MS-DOS, donc le script Bash est extrêmement plus puissant et plus pratique que le Batch.

L'inconvénient (en est-il vraiment un ? Je ne pense pas...) est que le script, comme son nom l'indique, ne se compile pas, donc il est relativement lent à l'exécution. Mais soyons francs avec nous même : on s'en fout ! C'est vrai, quoi : avec le script on automatise des tâches, on ne recherche en aucun cas la rapidité des calculs, des rendus, ou n'importe quoi d'autre *puisque* ce sont les logiciels auxquels on fait appel qui s'en chargent ! Savoir programmer un script, c'est juste savoir quel travail donner à quel moment, selon quelles conditions, à telle application ou telle autre. La rapidité est largement suffisante pour ce que l'on fait : utiliser d'autres programmes les uns après les autres. Par contre, il ne faut surtout pas envisager de programmer un logiciel de calcul mathématique poussé ou quoi que ce soit de ce genre avec le script Bash. D'ailleurs, c'est exactement le même problème pour tous les scripts : le javascript, le perl, même le Batch de MS-DOS, c'est-à-dire pour n'importe quel langage *non compilé*.

Commençons donc par faire notre premier script. Il doit répondre à la question suivante : « comment faire pour concaténer des fichiers texte en insérant entre eux une ligne de caractères - (tirets) ? ». Le problème est posé, je le décompose en sous-problèmes plus simples à résoudre : il faut afficher un fichier, afficher une ligne de tirets, afficher un autre fichier, etc... Mais on voit bien qu'il faut définir le moment où on s'arrêtera ! Eh bien, on s'arrête quand il n'y a plus de fichier texte, c'est-à-dire quand tous les fichiers texte ont déjà été lus, pardi !

Je résume mes sous-problèmes :

- afficher un fichier : c'est très simple, on connaît la commande « cat »
- afficher une ligne de caractères tirets : c'est aussi très simple, on connaît la commande « echo », qui affiche n'importe quelle chaîne de caractères
- par contre, plus difficile : il faut trouver un truc qui va à la fois nous permettre de répéter ces deux commandes successivement, mais aussi nous permettre de dire quand on s'arrête, ou plutôt de dire dans quelles conditions on fait notre affichage. Et ce truc plus difficile (excusez-moi amis programmeurs, mais il faut bien un peu de pédagogie !), ça s'appelle une boucle « for ». Pourquoi une boucle ? Parce qu'on a la notion de « tourner en rond ». L'instruction qui permet de boucler plusieurs instructions (les faire répéter) s'appelle for. Bien sûr, je ne sors pas cela de mon chapeau de magicien, je l'ai appris il y a bien longtemps, mais tu dois savoir qu'en cherchant sur les sites de documentation Linux (voir 6.2, en particulier Léa-Linux), et en cherchant dans ces sites de la documentation sur la programmation script shell, tu trouveras ton bonheur.

Voici le script, qui est bien petit :

```
for i in *.txt
do
  cat i
  echo -------------------
done
```

Explications :

- for i in *.txt : déclare une boucle for, en donnant à la variable « i », pour chaque tour de la boucle, le nom d'un fichier texte. « in *.txt » signifie que « i » prendra à chaque tour de boucle la valeur du fichier texte numéro 1, puis au tour suivant, la valeur du numéro 2... Un tour de boucle, si c'est pas clair, c'est en gros comme si l'ordinateur mettait son doigt sur chaque ligne et à chaque fois qu'il rencontre done, il remonte jusqu'au do (il ne cherche pas à comprendre, il est bête !). Et le fait de remonter s'appelle un tour de boucle.
- do : début de la boucle. Cette instruction est une balise, elle ne fait rien en apparence.
- cat i et echo - - - - - - : les instructions qu'on connaît déjà.
- done : balise qui annonce la fin de la boucle.

Bon, bon, bon... je sens vraiment que si je me plonge dans la partie programmation shell, ce sera trop long et en plus je suis en train de copier les documents qui m'ont servi à apprendre. Je préfère m'arrêter là et me consacrer au reste du document. La programmation est un domaine tellement vaste ! Va plutôt voir sur Internet, c'est bourré de documentation. Par exemple sur l'excellent site de documentation Léa-Linux, tu trouveras un très bon tutoriel pour programmer tes premiers scripts : `http://lea-linux.org/dev/shell\_script.php3`.

Ce document est consacré essentiellement à GNU/Linux, pas à la programmation.

Chapitre 6

Où trouver de la documentation

Sommaire

Eh bien, lis donc celle que tu tiens entre tes mains ! Non, je plaisante... GNU/Linux est le système d'exploitation le plus documenté au monde, et ses utilisateurs sont les plus solidaires entre eux. Il existe donc tout un tas de ressources, de documentations. On peut en trouver un peu partout.

6.1 Sur ton disque dur

C'est sûr que ça choque un peu pour un utilisateur de windows, de savoir que son disque dur contient de la documentation ! Mais je parle de la VRAIE documentation, pas les fichiers d'aide à la noix que l'on trouve sous windows en appuyant sur la touche F1 (certains sont quand même bien faits...). Il existe plusieurs sortes de documentations sur ton disque dur :

– les pages « manuel » : on a vu dans le chapitre 5 tout un tas de commandes, toutes très utiles. Parfois on ne se rappelle plus comment on les utilise. Pas de problèmes, les pages manuel[1] sont là ! Tu veux savoir comment on utilise la commande « `cp` » ? Tape « `man cp` » dans une console et ceci s'affichera :

  ```
  CP(1) Manuel de l'utilisateur Linux CP(1)
  NOM
       cp - Copier des fichiers.
  SYNOPSIS
       cp [options] fichier chemin
       cp [options] fichier... répertoire
  [etc]
  DESCRIPTION
  cp sert à copier des fichiers (et éventuellement
  des répertoires). On peut aussi bien copier un
  fichier donné vers une destination précise que
  copier un ensemble de fichiers dans un répertoire.
  [etc]
  ```

  Appuie sur les flèches du haut et du bas pour défiler (page up et page down fonctionnent) et sur « q » pour quitter la page man.
  Les pages man sont situées par défaut dans les sous-répertoires de `/usr/share/man/`
  Je me sers sans arrêt des pages man.
– Taper la commande suivie de « --help » affichera de l'aide. Par exemple, « cp --help » affichera autre chose que « man cp ». En fait, les pages man et les --help ne documentent pas de la même façon. --help est

[1] on dit plus couramment « pages man »

plutôt un résumé de l'utilisation et des options d'une commande, alors que man est plus complet. Parfois, on ne trouve que la page man d'une commande, parfois que --help, et parfois, ce qui est le mieux, les deux. Note que dans certains cas le fait de taper une commande toute seule affiche soit un message indiquant de taper « --help », soit affiche directement le contenu de « --help ». Par exemple, cp tout court nous demandera de taper « cp --help » pour en savoir davantage au sujet de cp alors que cdrecord tout court nous affichera « cdrecord --help ». De plus, si on tape une commande avec les mauvais paramètres, en général la commande nous explique ce qui ne va pas.
- Les pages Info sont un peu différentes des pages man. On les utilise en tapant `info [commande]`. Je me sers presque jamais des pages info.
- La commande `apropos [commande]` permet de faire afficher la liste de toutes les commandes ayant rapport avec la commande en argument. Je ne m'en sers presque jamais.
- Les HowTo (traduction : comment faire) : ce sont des documents HTML[2] très pratiques, te renseignant sur un point précis. Tu veux tout savoir sur le son sous Linux ? Consulte le HowTo Son (ou Sound HowTo). Attention tout de même à la date du document, car Linux évolue très vite ! Pour Mandrake, les HowTo sur le disque dur sont accessibles depuis la page HTML : `/usr/share/doc/HOWTO/HTML/fr/index.html` si tu les as installés. Pour savoir comment les installer, va voir le chapitre 8. Pour consulter les HowTo, ouvre la page index.html citée ci-dessus dans un navigateur HTML comme Konqueror, Mozilla, Netscape, ou encore dans le navigateur HTML *en mode texte* qui s'appelle Links. Les HowTo sont une énorme source d'informations ! À noter que les HowTo en anglais sont plus nombreux, il faut donc les consulter également.
- L'aide de KDE et de Gnome t'aident à te servir de... KDE et de Gnome ! Pas très utile, car KDE et Gnome sont très intuitifs.

6.2 Sur Internet

Internet c'est une bibliothèque gratuite ! On y trouve de tout. D'abord tu t'apercevras que les docs du disque dur demandent souvent d'aller voir sur Internet, ils proposent des références. Il y a des sites *incontournables* que tu dois visiter ! Je vais te donner la liste de ceux que j'ai repérés. Évidemment, c'est toujours pareil, tu ne vas pas y trouver *le* site qu'il te fallait absolument ! Je te le dis clairement : en général, les sites intéressants finissent par « .org ».

[2]mais aussi PostScript, PDF, et surtout DocBook, le format d'origine qui sert à générer les autres formats

Mais si tu ne trouve encore pas, va évidemment voir `http://www.google.fr/linux`, le meilleur moteur de recherche, avec en plus cette page spéciale pour rechercher des sites Linux !

6.2.1 Sites dédiés à la documentation

- Lea-Linux `http://lea-linux.org` : ce site est une mine d'or pour tout Linuxien ! Lea signifie « Linux Entre Amis ». Il contient un forum très intelligemment utilisé (peu de sujet bidon ni de disputes stupides), plein d'astuces, une pléiade d'articles très clairs sont disponibles, et le site est remis quotidiennement à jour. C'est un bon site pour poser tes questions dans le forum. En plus les gens de Lea Doc font un « Lea Book » au format PDF, qui est un condensé du site sous la forme d'un livre de 350 pages environ. De nouveaux Lea Books remis à jour sortent régulièrement. J'en ai imprimé un, tu n'as qu'à voir la section 8.7 pour savoir comment l'imprimer au format d'un livre, ce qui divise par 4 le nombre de feuilles de papier utilisées.
- Linux-Kheops `http://linux-kheops.com/doc/` : linux-kheops n'est pas entièrement dédié à la documentation, mais la partie /doc du site l'est. On y trouve des liens vers une pléiade de documents.
- Mandrake Campus, Mandrake Expert : voir 6.2.6.
- Sans-Linux `http://www.sans-linux.org/` : ce site est dédié aux réseaux informatiques sous Linux. C'est un site à visiter si tu veux un aperçu de ce que propose Linux pour le réseau (Linux, sans le réseau, c'est déjà formidable, mais si on s'en sert pour faire du réseau, alors là... c'est l'extase !). Le site n'est pas exhaustif, mais c'est un bon début. Avant, il s'appelait Linuxenrezo.

6.2.2 Informations / Quotidiens

- LinuxFR `http://www.linuxfr.org` : très bon site de news remis quotidiennement à jour, en français, traitant à peu près de tout.
- zlinux `http://zlinux.free.fr` : ce site est maintenu par un ami qui passe tous les soirs pas moins de 3 heures à récolter les dernières nouveautés (logiciels, informations...) pour le plaisir de ses visiteurs. Chacun peut participer en envoyant ses informations grâce à PHP/MySQL.

6.2.3 Magazines en ligne

Ce sont des magazines que l'on peut lire sur Internet.

– Linux Focus (français) `http://www.fr.linuxfocus.org/Francais/` : ce magazine parle de programmation, des logiciels Linux, de logiciel libre...

6.2.4 Sites personnels

– Alex's home page (français) `http://home.alex.tuxfamily.org/` : tiens, il est hébergé comme moi chez tuxfamily ! Cet hébergeur fonctionne avec des logiciels libres, et n'accueille que des sites en relation avec le libre. J'apprécie beaucoup cet hébergeur, car les administrateurs sont très proches et répondent vite à nos demandes. Alex's home page est un site destiné à un débutant, qui peut ainsi voir rapidement divers aspects de Linux (captures d'écrans, synthèses, interviews...).
– Linux story (français) `http://michael.briffoteaux.free.fr/` : voilà un site que j'ai aimé ! C'est l'histoire d'un débutant qui a décidé de mettre *toutes* ses expériences en ligne. Et il met, jour après jour, tous ses problèmes/solutions en ligne. C'est très complet, et on voit bien l'évolution.
– Oki666 `http://okki666.free.fr` : non ce n'est pas le diable ! Ce site contient énormément d'articles et de tutoriels. À visiter !
– Christian Casteyde `http://casteyde.christian.free.fr` : ce site est riche en documentations. On y trouve des renseignements sur les systèmes d'exploitation, sur les outils de développement GNU, et même un guide très complet d'installation, de configuration et d'utilisation de GNU/Linux. Christian Casteyde parle aussi des brevets logiciels et de leurs dangers.
– Benjamin Mauclaire `http://bmauclaire.free.fr/Informatique/informatique.html` : c'est le site d'un ami, qui est comme moi passionné de Linux. Il donne une liste de logiciels (surtout éducatifs), diverses documentations...

6.2.5 Logiciels

Sites « officiels »

– KDE `http://www.kde.org` : voir la présentation de KDE à la section 4.3.
– Gnome `http://www.gnome.org` : voir la présentation de Gnome à la section 4.3.
– LyX `http://www.lyx.org` : J'ai présenté ce logiciel à la section 7.1.

Sites proposant des éventails de logiciels

- RPM Find `http://www.rpmfind.net` : une base de données contenant tous les RPMs que l'on veut. Si tu veux obtenir un logiciel, il faut avoir plusieurs réflexes : voir la section 8.1. Au cas où tu es obligé de télécharger le RPM, il y est presque toujours.
- Gnu software map : `http://www.gnu.org` : regroupe tout un tas de logiciels, tous GNU.
- Dave Central : notre ami Dave a sélectionné les logiciels qu'il trouve les meilleurs
- Apps.Kde.org : `http://apps.kde.com` : une pléiade de logiciels spécialement prévus pour KDE. Il y a de plus en plus de logiciels grand public, enfin, des interfaces graphiques : des logiciels de gravure, des logiciels pour partitionner le disque dur, des logiciels pour traiter des sons, préparer des MiniDisc, des jeux... Tous ces logiciels existent mais ici on les trouve sous une peau magnifique : une interface graphique KDE. Pour les novices, je rappelle qu'un logiciel KDE peut tout à fait être exécuté depuis Gnome ou même n'importe quel gestionnaire de fenêtres, puisque ce sont les librairies KDE qui sont utilisées, pas l'environnement ! De même, un logiciel GTK (tout le monde dit logiciel Gnome) peut s'exécuter depuis n'importe quel gestionnaire de fenêtres.
- « Get more software » de Gnome : on y accède depuis `http://www.gnome.org` en cliquant en haut à gauche sur le lien « Get more software ». Regroupe plein de logiciels, dont la plupart sont bâtis sur le toolkit GTK.
- Linux center `http://www.linux-center.org/fr/` : ce site est un portail *très* complet ! On peut surtout télécharger des logiciels.
- Linux apps `http://www.linuxapps.com/` : un peu le même genre que Linux center, mais en anglais.
- Happy penguin `http://www.happypenguin.org` : Ce site rassemble une multitude de jeux pour GNU/Linux. Il y en a peut-être 500 !
- Lea-Linux `http://lea-linux.org` : ce site déjà décrit plus haut propose également un éventail de logiciels très bien répertoriés. Il y a des captures d'écrans et c'est appréciable.
- Sound & MIDI Software For Linux `http://www.bright.net/~dlphilp/linuxsound/` : ce site référence beaucoup de logiciels son et musique pour Linux. Attention tous ne sont pas terminés ni très soignés au niveau de l'interface (mais bon, ils recherchent la performance...).
- Le site des logiciels éducatifs libres `http://charger.tuxfamily.org/` : son nom parle de lui-même.
- Logiciels libres éducatifs `http://richtech.ca/cgi-bin/seul/seulview.`

pl : c'est un énorme liste de logiciels libres éducatifs, tous commentés.

6.2.6 Distributions

- Linux Mandrake http://www.linux-mandrake.org
  Mandrake a mis à disposition de ses utilisateurs (et même à ceux qui n'utilisent pas la distribution Mandrake) d'autres sites dédiés à l'apprentissage, à répondre aux questions...
  - Mandrake Campus http://www.mandrakecampus.org : on y apprend plein de choses. Ce sont des sortes de cours.
  - Mandrake Expert http://www.mandrakeexpert.org : on peut poser des questions et des « experts » nous répondent. Tout le monde peut s'inscrire en tant qu'expert, ce qui est très sympathique.
- Debian http://www.debian.org
- RedHat http://www.redhat.fr/
- SuSe http://www.suse.de/fr/
- Slackware http://www.slackware.org

6.3 LUGs

Un LUG est un « Linux User Group » : un groupe d'utilisateurs de Linux. Il y en a presque partout en France, au moins dans toutes les grandes villes. Dans le magazine Planète Linux, ils fournissent une liste des LUGs Français, avec emails et tout et tout, ainsi que sur le site de l'AFUL (Association Francophone des Utilisateurs de Linux et des Logiciels Libres) : http://www.aful.org

Contacter un LUG est le meilleur moyen de rencontrer du monde s'intéressant à Linux, de faire des soirées de démonstration ou d'install-party (on y installe Linux sur les ordinateurs des débutants et on leur explique), des bouffes, etc... C'est vraiment chaleureux.

Par exemple, celui de Digne s'appelle Linux Alpes. Je remercie ce LUG car c'est grâce à lui que j'ai découvert Linux. Ensuite, à Aix, il y a l'AXUL, à Marseille, le PLUG, etc... et chaque LUG a sa mailing list, pour se tenir au courant des rencontres.

6.4 Dans les livres

Je vais ici te donner mon avis sur les livres que j'ai achetés.

6.4.1 Livres généralistes

Je te conseille absolument d'acheter un livre qui parle de Linux sans se focaliser sur une certaine distribution, car ce genre de livre ne fait que décrire la distribution, et comme elle évolue sans cesse, le livre devient vite obsolète. Achète un livre qui parle de Linux en général, car les principes de base de Linux sont toujours les mêmes depuis le début.

Les livres des éditions O'Reilly sont tous très bien faits. « Le système Linux » est, je pense, le meilleur ouvrage sur Linux, car il parle de tout mais rentre assez dans les détails. Tous ceux qui l'ont l'apprécient énormément : je pense que c'est un achat que tu ne regretteras pas. Les auteurs (Matt Welsh, Matthias Kalle Dalheimer et Lar Kaufman) ne font pas leur livre à coups de captures d'écran qui prennent plein de pages et ne servent à rien du tout. Ce livre a été jugé excellent par tous les linuxiens que je connais.

6.4.2 Livres pour la programmation

Pour les scripts, il existe le livre de Christophe Blaess, c'est très complet et didactique à la fois. On y fait ses premiers pas en Bash, Tcl/tk, Perl,... mais cet ouvrage donne de bonnes habitudes en matière de programmation, il ne fait pas que présenter le langage. De plus, tous les scripts sont disponibles sur Internet, car ce serait trop long de les taper au clavier ! Ce livre est disponible aux éditions Eyrolles.

Christophe Blaess a écrit un autre livre sur la programmation système en C sous Linux. Je l'ai juste feuilleté en librairie, et il a l'air tout aussi bien que celui sur les scripts.

Au passage, Christophe Blaess est passionné de GNU/Linux, et il a traduit également pas mal de pages manuel en français. Tape « `man ls` » en console et va en bas de la page man : tu verras son nom ! Blague à part, il a beaucoup contribué au projet GNU.

6.5 Dans les magazines

Le monde des magazines informatique se partage en deux. On trouve les magazines Linuxiens et windowsiens. Entre les deux il en existe certains. Je ne veux pas être méchant, mais c'est dans le but de te fournir des renseignements les plus exacts possible que je vais faire la liste de tous les magazines que je connais, en donnant mon avis sur eux.

6.5.1 Les magazines windowsiens

- windows news : c'est le magazine que je trouve le plus « acheté » par windows ! C'est incroyable, ils vantent windows comme si c'était leur produit ! Où sont l'objectivité et la sincérité ? Chaque nouveau produit de Microsoft est attendu comme le messie.
- micro hebdo : eux aussi, ils recommandent des utilitaires très chers qui ne font que zipper des fichiers ! Et ils osent mettre une note de 4/5.... Chacun défend sa paroisse, mais lorsqu'on se présente comme magazine pour débutant, je pense qu'il vaut mieux traiter de *tout* ce que le débutant peut rencontrer au lieu de le focaliser sur les logiciels propriétaires. De plus, ils ont parfois confondu logiciel libre et freeware.

6.5.2 Les magazines « entre les deux »

- PC Max : lors de leurs premiers numéros ils ont écrit des articles sur Linux, très bonne initiative. Cependant ils se sont arrêtés. Pourquoi ?
- PC Team : les rédacteurs ont l'esprit Linux (pas complètement encore) mais le magazine parle de windows.
- Le virus informatique : c'est le canard enchaîné de l'informatique. Des rubriques par dizaines, aussi drôles et dénonciatrices les unes que les autres. Vraiment un magazine (sans pub en plus) intéressant. Magazine pro-Linux mais qui ne s'y consacre pas.
- Pirates Mag : comme son prédécesseur parle plus des attaques informatique, de la sécurité Internet... très intéressant aussi. Magazine pro-Linux mais qui ne s'y consacre pas.

6.5.3 Les magazines Linuxiens

- Planète Linux : le contenu du magazine n'est pas très riche mais à chaque fois on trouve deux cd roms bien remplis de programmes intéressants.
- Login : ce magazine parle d'autres systèmes d'exploitation comme amiga, beos, ou encore FreeBSD. Le problème est qu'il ne parle pas du tout du logiciel libre ! Il parle de Linux comme une simple alternative à windows, alors que c'est sa philosophie qu'il faut mettre en avant.
- GNU/Linux France Magazine : c'est le magazine que je préfère. Les articles parlent de sujets très intéressants, avec un sérieux inégalé. Le cdrom et le magazine sont très bien remplis. C'est le seul magazine, écrit par des passionnés, qui parle du projet GNU comme il se doit. Son titre exact est GNU/Linux France Magazine & Hurd. Ce qui est bien

avec ce magazine, c'est qu'il publie des articles de différents niveaux de difficulté. Il y en a certains que je ne comprenais pas et que je comprends aujourd'hui, d'autres que je ne comprends toujours pas... en tout cas c'est un magazine qu'on ne jette pas à la poubelle, car on peut le ressortir un an après et on le lira sous un nouvel angle. Il parle de tout : programmation, administration, jeux, graphisme, 3D...

Chapitre 7

Logiciels à connaître

Sommaire

7.1 Logiciels ayant servi à créer ce document

Ce document a été rédigé sous Linux, bien entendu. Ce serait le comble de l'avoir écrit sous windows ! Si tu veux toi aussi écrire des documents, lis la suite, je vais détailler tous les logiciels (libres) dont je me suis servi. Voir la section 6.2.5 pour avoir les adresses internet des logiciels cités.

LyX, version 1.1.6fix3 : c'est le logiciel qui m'a permis de taper ce livre, d'insérer les images aux bons endroits[1]. Tu te dis sûrement que LyX est comme word, tu dois penser qu'il est moins bien que word mais c'est faux ! LyX n'est pas un traitement de texte ordinaire, on ne l'appelle même pas « traitement » de texte mais plutôt « préparateur » de documents. LyX a une autre approche du texte que les traitements de texte comme word : en utilisant LyX, je ne m'occupe *que* du contenu, de la signification de mon document. J'indique quel rôle va jouer tel ou tel ensemble de mots. Pour faire un titre, rien de plus simple ! Je choisis « titre » dans la boîte en haut à gauche puis je tape le titre. Plus besoin de s'occuper de la taille des polices, du gras et tout le tintouin. LyX s'occupe de tout. C'est vraiment pratique pour rédiger des documents. Ensuite, il faut lancer le rendu (format DVI, PostScript, PDF, HTML ou même texte TXT). Pour visualiser le document, le format le plus approprié est le PostScript. Il suffit d'appuyer sur Ctrl+T pour lancer le rendu PostScript. LyX nous ouvre alors un logiciel de visualisation des fichiers PostScript.

Pour être exact, je vais un peu plus détailler le fonctionnement interne. En fait LyX génère un fichier texte à l'extension « .tex » qui est un code source pour LaTeX. LaTeX est un puissant logiciel de création de documents. Il se base sur un fichier texte d'extension « .tex » qu'il transforme en document PostScript. La plupart des gens, moi y compris, trouvent l'apprentissage des commandes LaTeX trop compliqué, c'est pour cela que LyX existe. LyX va donc générer pour nous le fichier « .tex »et c'est LaTeX qui, en arrière plan, produit le document. Quel travail en moins ! Bien sûr, si tu tapes les commandes LaTeX toi-même, tu peux faire bien plus de choses, mais LyX m'a suffit pour la création de ce document.

J'ai dit que LyX avait une approche différente. Avoue que lorsque tu utilises word, tu passes au moins le tiers de ton temps à chipoter sur la mise en page, à essayer d'imprimer une fois, corriger, imprimer encore, retoucher la taille de telle police... Eh bien avec

[1] Maintenant je tape directement le code LaTeX...

LyX (LaTeX en fait) plus de problèmes ! Plus besoin de s'occuper de vérifier si les titres sont tous à la même taille, s'ils sont en gras ou non, si un titre de section apparaît en bas de page et son contenu sur la page suivante, ou encore si les pages mentionnées dans la table des matières correspondent bien aux bonnes pages. LyX et LaTeX sont les meilleurs quant aux notes de bas de page, la numérotation automatique des chapitres, des sections, puisque c'est LaTeX qui s'occupe de tout... LaTeX existe depuis une dizaine d'années et il a été adopté par une multitude de personnes :

- très nombreux sont ceux qui ont écrit leur thèse avec LyX plutôt que word
- de nombreux éditeurs de livres ont choisi LaTeX pour leurs ouvrages (par exemple O'Reilly, ou la majorité des livres de mathématiques)
- les particuliers pour rédiger des documentations, des rapports, des courriers...

Le domaine de prédilection de LaTeX est l'impression de notations mathématiques. Tous les livres de maths l'utilisent, et les équations produites sont magnifiques. Bonne nouvelle, LyX interface très bien les mathématiques, et c'est en moins de 10 secondes que je tape cette équation :

$$\frac{10+\sqrt{x}}{(1+x)\,\frac{1}{\ln(e^{x})}} \times \int_{0}(e^{x})\,dx$$

Taper des textes est hyper facile avec LyX, c'est d'ailleurs pour ça qu'il a été créé. L'esprit du logiciel n'est pas le même. Si word a la mentalité windows, alors LyX et LaTeX ont la mentalité Linux. De plus, de par leur ancienneté, LyX et LaTeX sont dépourvus de bugs, et même si LyX plante, il sauvegarde ton fichier d'urgence et se ferme proprement. Lorsque tu ré-ouvriras ton document, LyX te dira : « Il existe une copie de sauvegarde d'urgence, voulez-vous la charger ? », et tu retrouveras ton document *exactement* dans l'état où il était lorsque LyX a planté. Même si tu n'avais pas sauvegardé depuis 10 minutes. Bravo !

Tu peux lire la section 8.4 pour savoir comment créer ton premier document PostScript/PDF sous LyX.

Xfig, version 3.2 : ce logiciel est super léger (comprendre : très rapide à réagir lors de son utilisation) et malgré le rejet que l'on peut ressentir lors des premières utilisations de par son aspect rustre, il est vraiment performant. Comme Dia (voir plus bas), il permet de dessiner des diagrammes, des figures ou encore des dessins vectoriels de façon très rapide et intuitive. En plus il est *beaucoup* plus stable que Dia. On sauvegarde ses images au format voulu, en passant du JPEG au PostScript et aussi le GIF, le PNG, le PDF et bien d'autres. Il ne reste plus qu'à les insérer dans LyX. J'ai dessiné l'arborescence avec XFig (figure 4.1 page 64).

Dia, version 0.86 : pareil que Xfig mais plus convivial, par contre il plante encore beaucoup. Les numéros de version parlent d'eux mêmes : entre 0.86 et 3.2, lequel est le plus stable ? C'est avec Dia que j'ai dessiné le schéma du disque dur, présent à la page 30 (quel formidable travail !).

QCad, version 1.4.6 : un logiciel de dessin technique épatant. Il est suffisant pour réaliser les dessins les plus courants, et je préfère m'en servir plutôt qu'AutoCad[2].

7.2 Logiciels triés par catégorie

Cette section est en permanente évolution.

7.2.1 Graphisme

GqView : un visualisateur d'images très rapide, tous formats, diaporama,...

XSane : un super logiciel pour scanner. Il fait tout, et je n'ai jamais vu un logiciel si complet sous windows. Il y a tout : histogramme, gamma, et plusieurs modes comme photocopie, fax, etc...

The Gimp : N'oublions pas le maestro de la retouche d'image, qui fait un beau pied de nez à « Photoshop » et « Paint shop pro » ! En effet, « The Gimp » (Gnu Image Manipulation Program) permet l'utilisation des calques graphiques, des masques, il est doté d'une palette de plus de 60 filtres, une palette de 25 outils que l'on peut configurer... Bien sûr tous les outils de traitement des couleurs sont présents : l'histogramme, les courbes de niveau, réglage du gamma de l'image... Il est adapté à la publication Internet : création d'animations au format GIF animé, compression des palettes de couleurs indexées et création d'images cliquables (image maps) grâce à un assistant. Il permet aussi de traiter des vidéos au format mpeg. Mais sa plus grande qualité est qu'il est

[2] aussi je fais des dessins industriels très simples

totalement configurable. On peut en effet écrire des « scripts-fu » combinés aux scripts Perl. Résultat : on peut relativement facilement créer ses propres filtres.

Étant donné que ce logiciel est libre, des personnes se sont lancées dans la réalisation de maintes documentations libres, la plus importante est « Grokking The Gimp » : un livre entier en libre téléchargement à l'adresse `http://www.gimp-savvy.com`. On peut aussi acheter la version papier. Il est en anglais, mais c'est vraiment un très bon ouvrage rempli de cas concrets (à télécharger et à consulter !).

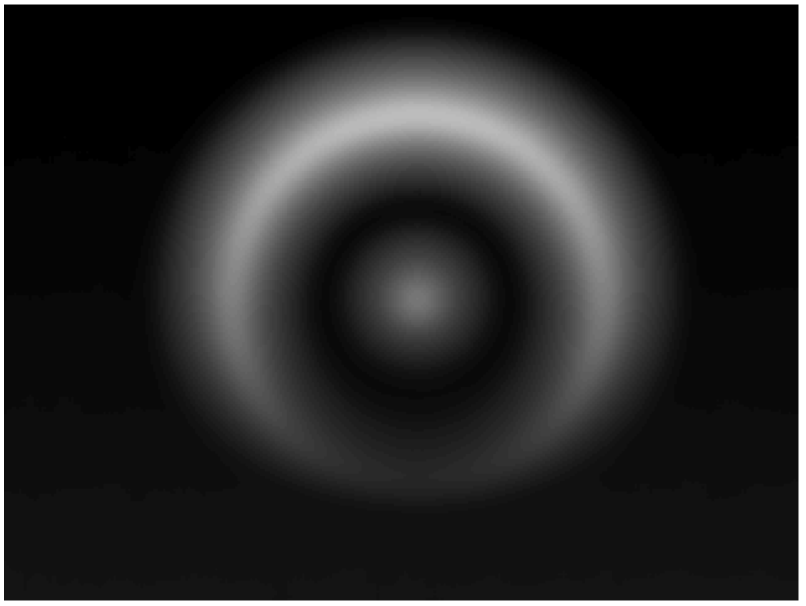
Photoshop est peut-être utilisé par la majorité, mais « The Gimp » est un outil à sa hauteur. Je pense même que, hormis le fait qu'il est libre, « The Gimp » est supérieur à Photoshop car il est bien plus intuitif (il faut un petit moment d'adaptation), mais surtout parce que lorsqu'on sait programmer, on peut créer des scripts incroyablement efficaces (voir les script-fu et scripts perl dans le menu extensions). De plus, les mises à jour (certes pas très fréquentes) apportent à *chaque fois* leur dose de *réelles* nouveautés.

Les experts de Photoshop trouveront « The Gimp » déroutant, mais lorsqu'on s'y est habitué, on le trouve plus pratique...

Blender : C'est le maestro de la modélisation/animation/interactivité/rendu 3D, qui est lui aussi, comme « The Gimp », à la hauteur du logiciel le plus répandu dans ce domaine : 3D $tudio max. En plus, il a une interface certes déroutante mais très pratique à la longue. Ce logiciel évolue presque tous les mois. Il est désormais un logiciel libre. Blender est lui aussi très documenté, et il te suffit d'aller voir sur le site `http://www.blender3d.com` pour voir tous les tutoriels. Cela te permettra d'acquérir en peu de temps les bases de Blender, et de pouvoir très vite réaliser tes premières images de synthèse. Ce n'est pas parce que 3D studio Max coûte plus de 5000 € qu'il est forcément meilleur ! Bien sûr, 3D studio max permet plus de choses que Blender mais ce sont des technologies non accessibles à l'utilisateur normal (motion capture par exemple, le fait d'animer des personnages virtuels reliés à des capteurs placés sur de vrais acteurs). De plus, Blender s'utilise très rapidement une fois qu'on a compris les principales touches du clavier, car avec Blender on ne perd pas de temps à aller chercher les fonctions dans les menus. Du point de vue du logiciel lui-même, tu seras étonné d'apprendre que Blender tient en 2 Mo seulement, alors que 3D studio Max occupe 30 Mo minimum. Incroyable, non ?

MoonLight Atelier : Un autre très bon logiciel de création 3D, qui ressemble plus à 3D studio max ou Softimage.

FIG. 7.1 – Carte de relief



Sis : Ce logiciel permet de créer des stéréogrammes. Un stéréogramme est une image que l'on doit regarder d'une façon spéciale afin de voir apparaître une forme en 3 dimensions. Bien sûr, si on la regarde normalement, on ne voit rien. Un stéréogramme est obtenu à partir d'une image en niveaux de gris. Voir la figure 7.1.

C'est cette forme que l'on verra ensuite, une fois que le stéréogramme sera créé. Les pixels les plus noirs apparaîtront au fond, et les plus blancs sortiront de l'image vers soi. Après ça, on choisit une texture, par exemple la figure 7.2.

et pour finir on lance le calcul du stéréogramme. Le calcul dans ce cas a pris moins de 5 secondes. Pour l'information, j'ai utilisé la ligne de commande suivante :

```
sis oval.tif stéréogramme.tif -m -t texture.tif
```

sachant que `oval.tif` est l'image en niveaux de gris (les autres fichiers tif n'ont pas besoin d'explications...).

On obtient le stéréogramme de la figure 7.3.

Pour voir apparaître le stéréogramme, il faut regarder plus loin que la feuille, comme si elle était transparente. Tu peux commencer à placer ta main devant ta figure et regarder plus loin : elle se dédouble. Fais pareil avec la feuille. En ajustant ton regard de façon à ce que les deux

FIG. 7.2 – Texture

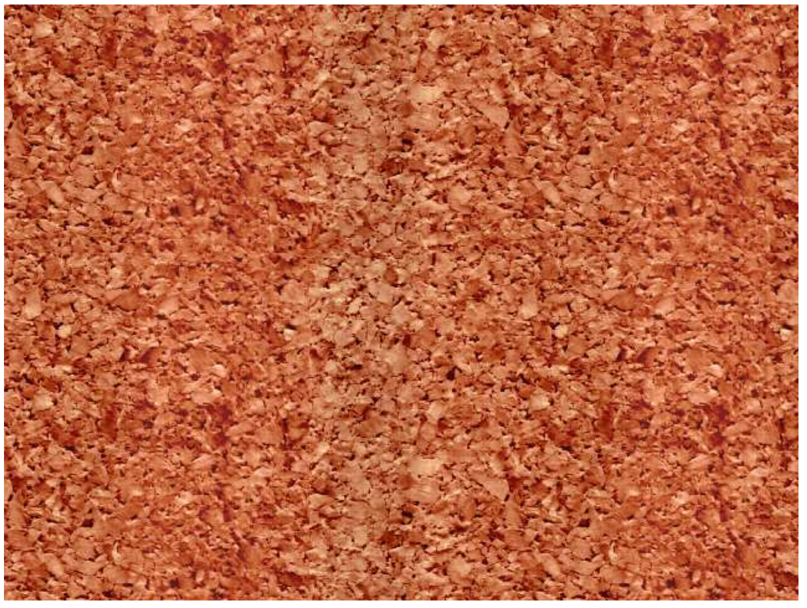
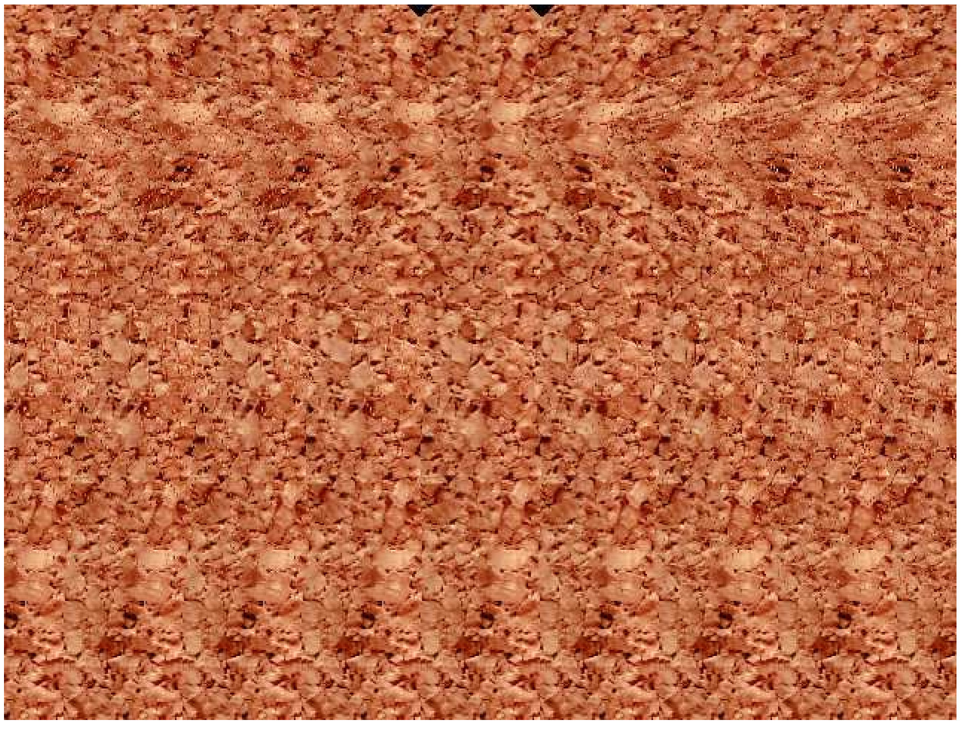


FIG. 7.3 – Stéréogramme



points noirs en haut du stéréogramme coïncident et se dédoublent en trois points, la forme apparaît. Si tu as l'habitude d'être ivre, ce sera plus facile pour toi...

C'est vrai que ça n'a rien à voir avec GNU/Linux, mais bon, c'est très intéressant.

7.2.2 Son et musique

Xmms : C'est l'équivalent de winamp pour Linux, sous le signe du logiciel libre bien sûr ! Il est doté de très nombreux plugins de visualisation, de traitement du son, etc...

Zinf : Ce logiciel (anciennement Freeamp) est remarquable ! Je viens de l'essayer aujourd'hui et je trouve qu'il ressemble beaucoup à musicmatch jukebox, du point de vue du tri des fichiers audio (par genre musical, par interprête...).

Brahms : c'est un séquenceur audio/midi pour KDE. Il ressemble au logiciel Cubase. Brahms est assez évolué aujourd'hui.

Audacity : édition/traitement de sons.

KWave : édition/traitement de sons.

Solfege : Ce logiciel te permet d'apprendre le solfège à travers divers modules, certains théoriques, d'autres pratiques : reconnaissance d'intervalles et d'accords, dictée musicale, chant, rythme, gammes,...

7.2.3 Montage vidéo

Broadcast2000 : Broadcast2000 est un logiciel de montage vidéo pas mal du tout. Les développeurs initiaux ont abandonné le projet pour des raisons obscures, et d'autres l'ont repris (voilà l'avantage du logiciel libre). Seulement, les distributions fournissent une ancienne version. Il faut aller télécharger la version améliorée (version Broadcast2000c-mf2) sur le site `http://www.physik.tu-muenchen.de/~jfranosc/download.html`. Ce logiciel permet aussi de faire du montage audio seul, dans la mesure où il est capable de faire du montage vidéo-audio.

7.2.4 Astronomie

Un site qui rencense des logiciels pour l'astronomie : `http://www.science.physique.free.fr/AUTRES/Astro\_linux/`

Xplns : un excellent planétarium en 2D.

KStars : un autre planétarium en 2D.

Stellarium : un planétarium en 3D. `http://stellarium.free.fr`

Celestia : simulation spaciale temps réel en 3D (voyage dans l'espace).`http://www.shatters.net/celestia/`

Open Universe : encore une simulation spaciale en 3D. `http://www.openuniverse.org/`

Xephem : cartographie et éphémérides astronomiques, très complet et puissant, nombreux catalogues. `http://www.clearskyinstitute.com/xephem/`

MIDAS : le traitement d'images utilisé par les professionnels, puissant mais piloté par macro-commandes. `http://nnsa.dl.ac.uk/MIDAS/`

IRAF : un autre traitement d'images possédant moultes bibliothèques dédiées aux différentes spécialités astronomiques. `http://iraf.noao.edu/projects/pc-iraf/pc-iraf.html`

7.2.5 Physique

Xoscope : pour analyser comme un oscilloscope les signaux qui entrent par la carte son. Le logiciel supporte même les cartes d'acquisition ProbeScope, disponibles dans les catalogues d'électronique.

Mek : logiciel de simulation de particules en mécanique de Newton.

Lum : étude de la lumière dans les prismes (optique)

gChamois : un logiciel de géométrie destiné à l'optique géométrique comme l'est Chamois(TM)

Grany : simulateur d'automates cellulaires pour les milieux granulaires. Ce que j'ai compris, c'est que je ne suis pas assez calé en physique pour savoir de quoi ça parle !

xspringies : modéliser les oscillateurs à ressort dans différents environnements (g, rebonds...)

7.2.6 Chimie

Kalzium : Dessine la table périodique des éléments de Mendeliev. Inclut un quizz, une barre temporelle (pour représenter la table à différentes époques), mise en couleurs selon la classe des éléments...

Xdrawchem : pour dessiner des molécules en 2D.

xem : simulation de dosages acido-basiques avec évolution des couleurs de/des indicateur(s) colorés choisis et traitement de ses propres mesures

7.2.7 Mathématiques

Maxima : un langage algébrique de haut niveau. Je ne m'en suis jamais servi.

Octave : un langage de haut niveau interactif pour les calculs numériques.

Mupad : excellent logiciel de maths ! C'est comme Maple, mais je préfère Mupad. Par contre Mupad n'est pas libre.

Kgeo : Logiciel de géométrie, un peu comme Cabri géomètre. On trace les figures, on définit des contraintes géométriques, on les anime,...

DrGenius : Autre logiciel de géométrie

KPercentage : pour les plus jeunes, aide à « sentir » la notion de pourcentage

KTimes : aide à apprendre ses tables de multiplication

KmPlot : logiciel de tracé de fonctions

GtkGraph : autre logiciel de tracé de fonctions

Mathplot : autre logiciel de tracé de fonctions

GnuPlot : logiciel de tracé de fonctions, plus complet que tous les autres, mais en ligne de commandes. Il produit des fichiers PostScript destinés à l'écran ou à l'imprimante.

Kfract : générateur de fractals

Scilab : logiciel libre de calcul numérique, similaire à MathLab. Développé par l'INRIA (Institut national de recherche en informatique et en automatique), il est vraiment très puissant. Il est aujourd'hui utilisé par des chercheurs, des scientifiques... On pourrait le mettre dans la catégorie physique.

R : logiciel mathématique servant à représenter des données sous forme de graphiques.

Scigraphica : logiciel de représentation graphique de données numériques. Il permet la 2D, la 3D et les coordonnées polaires. Clone de l'onéreux Origine(TM).

7.2.8 Sciences de l'ingénieur

KLogic : simulation de circuits de portes logiques, création de chronogrammes...

QCad : c'est un logiciel de DAO (Dessin Assisté par Ordinateur), doté d'une interface très agréable. Ce n'est pas la même que les AutoCAD, mais je la trouve très agréable.

7.2.9 Langues

KLatin : logiciel pour réviser son latin !

KLettres : pour les plus jeunes, apprentissage de l'alphabet et des syllabes

KVocTrain : assistant pour apprendre du vocabulaire étranger

7.2.10 Astrologie

KAstrolog : création du thème astral, avec tout un tas d'options. Ce logiciel a l'air très complet, mais je ne connais rien à l'astrologie !

7.2.11 Éducatifs

Pour les logiciels éducatifs libres, il faut aller voir ce site `http://charger.tuxfamily.org/` et aussi `http://richtech.ca/cgi-bin/seul/seulview.pl`.

KEduca : permet aux enseignants de créer des leçons (des sortes d'assistants dans lesquels on répond par les choix prévus) et les élèves utilisent ces fichiers de leçons. KEduca est un créateur/lecteur de fichiers de leçons !

KTouch : pour apprendre à taper au clavier avec ses 10 doigts.

7.2.12 Gestionnaires de fichiers

Konqueror : c'est l'outil à tout faire de KDE. Gestionnaire de fichiers, navigateur Internet, client FTP, et plein d'autres choses... Je le présente un peu mieux à la section 4.3.

Xwc : c'est comme l'explorateur windows. Si Konqueror te semble trop lent, essaie xwc, mais n'oublie pas d'essayer Krusader.

Rox : encore un gestionnaire de fichiers, lui aussi très léger.

Krusader : ce logiciel est extraordinaire ! C'est un gestionnaire de fichiers comme Midnight Commander (ou Norton Commander ou Windows Commander). L'écran est partagé en deux fenêtres et on copie/déplace les fichiers d'une fenêtre à l'autre. Il est très pratique : on peut créer ou décompresser une archive dans tous formats courants, créer un répertoire, voir un fichier (l'outil de visualisation apparaît en moins d'une seconde !), l'éditer, le copier ou le déplacer comme je viens de le dire. En plus, il utilise un terminal intégré (en fait c'est Konsole), des signets, on peut même le réduire dans la barre d'icônes à côté de l'horloge...

C'est un très bel exemple du logiciel libre typique , qui est super bien adapté à nos besoins parce qu'il a été créé par ses utilisateurs.

7.2.13 Finances

GnuCash : C'est l'un des meilleurs logiciels de gestion de comptes sous Linux. Il permet d'importer les fichiers Quicken.

GnoFin : Un autre logiciel du même genre.

KMyMoney2 : Logiciel de gestion financière au look KDE.

Emma : Ce logiciel a l'air intéressant. Je ne l'ai pas essayé.

7.2.14 Bureautique

Attention ici je décris des logiciels bureautiques dans lesquelles on travaille de manière WYSIWYG (What You See Is What You Get) : on manipule directement les objets (texte, images. . .) à l'écran, contrairement aux préparateurs de texte comme LaTeX dans lesquels on « programme » le texte. Pour les préparateurs de texte, voir la section 8.4 pour savoir comment faire pour créer son premier document sous LyX.

KOffice : Cette suite bureautique est détaillée à la section 4.3. Elle est totalement intégrée à KDE, ce qui fait qu'elle est très : on peut visualiser depuis Konqueror (le logiciel à tout faire de KDE !) n'importe quel fichier de KOffice. Par exemple, pas besoin de lancer KWord pour voir un fichier produit par ce logiciel : Konqueror permet de l'afficher (en fait il ne charge que le composant de KWord qui permet d'afficher les fichiers KWord).

KWord permet de lire les documents de Microsoft Word, à condition qu'ils ne soient pas trop complexes (éviter les wordarts, images cliparts. . .). KOffice en général n'est pas encore terminé, ce qui fait que les logiciels plantent encore trop souvent, lorsqu'on se met à effectuer des opérations trop complexes comme l'impression d'une page pleine d'images. . . en tout cas c'est le cas chez moi. Mais essaie toujours la dernière version du logiciel car sûrement qu'au moment où tu liras ce livre, les logiciels se seront améliorés.

Gnome-Office : Cette suite bureautique elle aussi est détaillée à la section 4.3. Elle aussi est en pleine évolution. De même que KOffice est intégré à KDE, Gnome-Office est fait pour fonctionner avec l'environnement Gnome.

AbiWord permet aussi de lire des fichiers Microsoft Word basiques, et Gnumeric ceux d'Excel.

StarOffice/OpenOffice : StarOffice est une suite bureautique commerciale très lourde[3]. Cependant elle est très complète, assure une totale compatibilité avec les document Microsoft Office[4],... Au départ, StarOffice était développé par StarDivision, avant d'être rachetée par Sun Microsystems (le plus gros concurrent de Microsoft). Afin de rivaliser de façon réaliste avec Microsoft Office, Sun a décidé de libérer le code source de StarOffice : c'est ainsi qu'est née OpenOffice. En fait StarOffice et OpenOffice sont les mêmes logiciels, sauf qu'ils sont développés en parallèle, et que StarOffice reste commercial, et que OpenOffice sera toujours libre (avec tous les avantages que cela apporte).

Cette suite bureautique se compose donc de :

* Traitement de texte : StarWriter/OpenWriter

* Tableur : StarCalc/OpenCalc

* Dessin vectoriel : StarDraw/OpenDraw

* Présentations : Impress

Il existe d'autres suites comme Applix ou SiagOffice, mais je ne les connais pas assez pour les présenter.

Scribus : ce logiciel n'est pas encore terminé, mais il est déjà très attrayant. C'est un vrai logiciel de PAO (Publication Assistée par Ordinateur), qui ressemble fort à Quark XPress. On travaille à base de cadres, qui peuvent être chaînés, on applique des maquettes...

Oleo : c'est un tableur pour terminal texte. Souvent on me demande un tel logiciel, alors je le mentionne... bien que ce ne soit pas une suite complète de logiciels.

7.2.15 Navigateurs Internet

Il existe beaucoup de navigateurs sous GNU/Linux. Certains sont propriétaires, d'autres sont libres.

Konqueror : on commence à le connaître, celui-là ! C'est le navigateur officiel de KDE.

Netscape/Mozilla/Galeon : la même chose que StarOffice/OpenOffice s'est passée pour Netscape. La société a libéré son code source tout en continuant à développer en propriétaire certaines parties du logiciel. Mozilla est donc la version libre de Netscape, et Galeon se base sur le moteur de Mozilla : c'est le navigateur de Gnome.

[3] lente à l'exécution et grosse en occupation sur le disque dur

[4] qui ne sont certes pas une référence en matière de standards ouverts, mais comme tout le monde s'en sert...

Opera : il n'est pas libre mais il est performant.

Lynx : c'est un navigateur internet en mode texte. Quel est l'intérêt ? Il est très rapide. Bien sûr il n'affiche pas les images et il faut passer par un logiciel externe comme xview. L'option -dump permet de copier la page dans un fichier texte (si on redirige la commande). Lynx ne supporte pas les frames ni les tableaux.

Links : c'est aussi un navigateur internet en mode texte, et en plus il supporte les frames et les tableaux. Il est également possible d'utiliser l'option -dump, comme pour Lynx.

Internet Explorer : non je plaisante. On est épargné.

7.2.16 Réseau

VNC : ce logiciel permet d'ouvrir une session sur un système d'exploitation depuis un autre. Par exemple, depuis Windows on peut ouvrir son compte sur une machine Linux et voir dans une fenêtre ce qu'on verrait sur la *vraie* machine Linux. Mais VNC ne se limite pas à Windows et Linux, et on peut l'utiliser pour un nombre incroyable de plateformes, par exemple Macintosh, PalmOS, et on a même droit à un client sous forme d'applet Java, ce qui permet d'ouvrir une session depuis un navigateur Internet.

7.3 Le cas des frontends

Sous Linux, les commandes régissent la plupart des choses. Certaines personnes préfèrent une fenêtre graphique à une commande. Des programmeurs ont donc créé des « frontends » (apparence graphique, mais attention, c'est une traduction à la noix) à ces commandes, qui sont juste des fenêtres avec des boutons, des cases à cocher…, mais le travail est exécuté en réalité, par derrière, par la commande, et non par le frontend. Il est donc très facile de créer un frontend car on ne reprogramme pas les fonctionnalités mais on programme l'apparence. C'est le cas de Klpd, qui est le frontend de lpd. Il faut bien avoir compris cette notion. Si tu cherches un « winzip », tu ne trouveras que des frontends à la commande tar. L'avantage est que, pour optimiser les performances, on optimise la commande, et tous les frontends sont du coup améliorés sans qu'on ait besoin de les reprogrammer !

7.4 Tableau d'équivalence logicielle windows/Linux

J'ai moi-même utilisé windows pendant très longtemps, *trop longtemps.* Je m'en veux car, d'une part, j'ai perdu un temps monstre pendant lequel j'aurais pu m'instruire autrement et également car je suis passé à côté de tas de choses, mais d'autre part je me dis que ce n'était pas de ma faute car personne ne m'avait montré un autre système. Je n'étais pratiquement pas au courant de l'existence de GNU/Linux. La plus grande difficulté que j'ai rencontrée a été de me bâtir une culture logicielle. Je m'explique : sous windows, si je te parle de créer une archive compressée, tu vas tout de suite penser aux fichiers zip et au logiciel « winzip » (à moins que tu n'utilises pas souvent l'ordinateur). En arrivant sous Linux, tu vas forcément chercher un équivalent de winzip. On peut généraliser ce comportement pour tous les logiciels : un windowsien cherchera à se reconstruire une logithèque proche de celle qu'il connaît déjà, un utilisateur de Macintosh fera la même chose. C'est une regrettable erreur ! Certains éditeurs exploitent cette malheureuse tendance qu'ont ces personnes : ils sortent une version pour Linux de leurs logiciels (non libres évidemment) afin de gagner quelques clients de plus. Le problème est que les personnes mal informées vont utiliser GNU/Linux avec des logiciels commerciaux, ce qui gâche tout (sauf si le logiciel commercial est le seul dans son domaine d'application). Utilisons plutôt les logiciels fabuleux du monde de GNU/ Linux, découvrons-les petit à petit. Certes, cette étape prend un temps considérable, il faut exécuter tous les programmes présents sur ton disque, les essayer et comprendre à quoi ils servent. Je vais donc te donner un tableau d'équivalence, pour commencer, en t'indiquant le nom du logiciel Linux correspondant à celui que tu connais sous windows. Les « ? ? ? » signifient qu'il n'existe pas de logiciel propriétaire équivalent au logiciel libre proposé. Bien sûr, on peut toujours (et encore. . .) recompiler un logiciel libre pour windows, puisqu'on a le code source, quand je dis logiciel windows, je parle de ceux qui sont propriétaires et que le grand public a l'habitude d'utiliser. LaTeX est disponible sous windows, LyX aussi, « The Gimp » aussi, Mozilla aussi, OpenOffice aussi. . .

Mais n'oublie pas qu'ensuite, il faudra un peu plus se cultiver à la sauce Linux et tu verras aussi que ce ne sera pas si difficile et que tu en redemanderas. Mieux : tu n'auras plus aucun mal à ne plus utiliser les anciens logiciels windows, tellement tu seras emballé par ceux de Linux, et tu te surprendras même à pester contre tes anciens « mauvais assistants de travail » que tu adorais tant, tellement les logiciels Linux sont adaptés à tes besoins. Je prends souvent l'exemple de GNUCash qui est un logiciel libre de gestion des comptes bancaires. Certes, un comptable professionnel ne l'utilisera sûrement pas car GNUCash n'est pas assez complet, mais moi j'en raffole ! Un windowsien pensera en tout premier à pirater « Microsoft money » ou « quicken » pour gérer ses comptes, car bien sûr, il ne veut pas payer pour un logiciel dont il n'exploitera même pas un dixième des capacités. Remarquons également que le windowsien a acquis au fil des âges un étrange comportement : il veut toujours posséder le meilleur logiciel et le plus cher de sa catégorie, même si ce logiciel est très lourd, alors qu'il peut se contenter d'un petit logiciel qui lui ira très bien. Il collectionne sur ses étagères des CD-Roms par dizaines qu'il exhibe à ses amis, se vantant de posséder une logithèque ultra complète. Il arrive même qu'il aie envie de copier un logiciel *juste pour l'avoir*, au cas où il en aurait besoin (ce qui n'est même pas sûr). Mauvais comportement ! S'il allait voir du côté du logiciel libre, il s'apercevrait qu'il existe une flopée de petits logiciels (qui sont très rapides et très légers d'exécution), dont GNUCash est le plus abouti, qui correspondent le mieux à ses besoins : entrée des transactions le plus facilement du monde, gestion de différents types de comptes, d'intérêts, de monnaies (même l'Euro €), et sortie des résultats en HTML, texte ou même LaTeX ! Il faut de manière générale privilégier un logiciel qui répond à nos attentes plutôt que le meilleur logiciel « en absolu ».

Voici donc ce tableau :

Logiciel libre	logiciel propriétaire	commentaires
tar, gzip, bzip2	winzip	voir remarques
GQview, Electric Eyes, view	Irfan view, acd see	très bon visualisateur d'images
The Gimp	Photoshop, Paint shop pro	The Gimp rassemble pratiquement *toutes* les fonctions de ses équivalents windows, et en rajoute d'autres bien pratiques
Xmms	winamp	Xmms, clone de winamp, est un lecteur multimédia universel
Arts Builder	virtual waves	synthétiseur de sons virtuel
Broadcast 2000	première, media $tudio pro	logiciel de montage vidéo et audio
Gnome-iconedit, Kiconedit	microangelo, etc	éditeur d'icônes
Ksnapshot, Xwpick	snag it, etc	capture d'écran (très complet)
Xmorph	winmorph, etc	logiciel de morphing
Gphoto	? ? ?	logiciel de communication avec appareil photo numérique
Xfig, Dia	? ? ?	logiciels ultra complets de création de diagrammes

Paint	paint	le Paint de Linux *surpasse* celui de window$ en fonctionnalités !
Open Office, Star Office, Word Perfect	micro$oft office, word perfect	Star Office permet de lire et d'écrire les fichiers micro$oft
Sketch, Kontour	adobe illu$trator	de très bons logiciels de dessin vectoriel
GnuCash, GnoFin, Kmymoney	micro$oft money, quicken	GnuCash est amplement suffisant pour gérer ses comptes personnels
Xoscope	wino$cillo	oscilloscope (grâce à la carte son ou carte dédiée)
GnuPlot, GtkGraph	plein de petits sharewares	tracé de courbes mathématiques
Kdevelop, Glade	vi$ual basic, visual C++...	outils de développement d'applications
Gmc, Nautilus, Konqueror, Rox, Krusader	explorateur windows	gestionnaires de fichiers
Konqueror, Mozilla, Netscape Navigator, Lynx, Links	internet explorer	navigateurs Internet
LyX, KlyX	? ? ?	traitement de texte d'une approche autre que celle des commerciaux. C'est avec LyX que j'ai rédigé ce document. LyX est extraordinaire, ne te fie pas à son apparence !
KreateCD, CD bake oven...	Nero, Winoncd	Ces logiciels sont des frontends aux quatre commandes de gravure : cdrecord, cdrdao, cdparanoia et cdda2wav.
Blender, Moonlight	3D $tudio max	Blender est le meilleur logiciel 3D que je connaisse. Il est si agréable... et c'est un chef-d'oeuvre de programmation : il tient en 2 mo !
Kgesture	Sensiva	Permet de lancer un logiciel par mouvement de la souris. On définit ses mouvements, puis l'ordinateur démarre le programme s'il détecte qu'on a bougé la souris de la même façon.

Chapitre 8

Tâches les plus courantes

Sommaire

Ce chapitre est presque vide pour l'instant, c'est à toi de m'envoyer des e-mails pour me demander quoi expliquer. De plus, il existe le « guide de survie du débutant sous Linux » (unique et très longue page HTML, sans aucune image) qui rassemble les tâches les plus courantes et bien plus. Va la trouver à l'adresse `http://www.medsyn.fr/perso/g.delafond/survie.htm`.

8.1 Installer un logiciel

Pour installer un logiciel sous Linux, on dispose de plusieurs approches. On peut récupérer soit le code source du logiciel (dans ce cas c'est à toi de le compiler), soit le logiciel déjà compilé (on dit qu'on télécharge les binaires). Bien sûr la première remarque est : « Pourquoi se casser la tête à compiler alors qu'on peut trouver le logiciel tout prêt ? En plus je n'y connais rien en programmation ». Bon, c'est vrai, la compilation rebute le débutant. Pourtant c'est simple, comme on va le voir. On va aussi se pencher sur la manière d'installer un logiciel déjà compilé (on dit aussi logiciel pré-compilé, bien que la particule « pré » ne signifie rien du tout). L'avantage de la compilation est que le logiciel sera plus performant après compilation sur ton ordinateur que si c'était quelqu'un d'autre qui l'avait compilé sur une autre machine. L'argument contraire est que le gain de performance est minime pour les programmes normaux (qui ne nécessitent pas énormément de puissance).

On comprend par contre qu'un logiciel qui nécessite le meilleur de ton ordinateur aura tout intérêt à être compilé sur ce dernier. C'est notamment le cas du noyau du système d'exploitation, sur lequel reposent *tous* les logiciels ! Imagine qu'il n'exploite pas toutes les capacités de ton ordinateur : tous les logiciels en souffriraient. C'est pour cela que les utilisateurs qui connaissent ce problème préfèrent compiler eux-mêmes leur noyau. Les débutants qui apprennent que leur interlocuteur a déjà compilé son noyau tombent des nues ! Ils le prennent pour un expert. C'est délicat, mais au final on a un petit bijou bien adapté à sa machine, qui fonctionne désormais à merveille.

De toute évidence, la compilation d'un logiciel est bien moins complexe que la compilation du noyau.

Donc revenons à nos moutons : tu rêves d'installer le logiciel « truc ».

8.1.1 Pas de compilation

Premier réflexe : *rechercher s'il se trouve sur les CD-Roms de ta distribution.* C'est inutile la plupart du temps d'aller chercher sur Internet. Chaque distribution range ses paquetages[1] dans un répertoire distinct, mais je peux

[1] Va voir la section 9.5 pour en apprendre plus sur les paquetages.

donner l'exemple de Mandrake : il faut regarder dans `/mnt/cdrom/Mandrake/RPMS` pour obtenir la liste des paquetages. Mandrake propose aussi les outils « urpm* » permettant de manipuler la « base de données » des RPMs disponibles. Par exemple, pour rechercher le logiciel « truc », on tape :

`$ su` (il faut être root pour installer quoi que ce soit)
`$ <mot de passe root>`
`# urpmi truc`

et on regarde le résultat. Soit on nous demande d'insérer le CD-Rom numéro x, soit on nous dit que ce paquetage n'est pas disponible. Dans le premier cas, on insère le bon CD-Rom et on appuie sur Entrée. On voit l'installation se dérouler en quelques secondes.

Si on n'a pas la chance d'avoir notre logiciel sur nos CD-Roms, on va sur Internet, sur le site `http://www.rpmfind.net`. On recherche le paquetage, et on télécharge celui qui correspond à notre distribution. Ce doit être quelque chose comme « truc-mdk-i586.rpm » pour une distribution Mandrake. Au passage, je ne traiterai pas des paquetages Debian car je n'ai jamais utilisé cette distribution. Je compte sur toi pour me renseigner au cas où tu connaisses la chose...

Une fois le paquetage téléchargé, on l'installe avec :

```
$ su
$ <mot de passe root>
# rpm -ivh truc-mdk-i586.rpm
```

et on admire le résultat. On n'a plus qu'à taper « truc » pour lancer le logiciel, je suppose.

Il est possible que le logiciel ait besoin de bibliothèques pour fonctionner. La majorité des bibliothèques se trouvent sur les CD-Roms de la distribution. Sinon, on recommence à télécharger. Lis la section 9.5 pour plus de détails sur les paquetages.

Pour désinstaller un RPM, il faut trouver son nom : on tape « `rpm -qa|less` » et on recherche son nom. Ensuite, on tape « `rpm -e nom_du_rpm` » et il se désinstalle. Si on veut désinstaller un RPM qui permet de faire fonctionner d'autres RPMs, il faut d'abord désinstaller ces autres RPMs. Je recommande quand même d'utiliser un logiciel graphique comme « rpmdrake » ou « kpackage » pour la gestion des RPMs. Le débutant sera plus rassuré.

8.1.2 Avec compilation

Dans ce cas, on doit passer par Internet. Il faut aller sur le site du logiciel pour télécharger le code source (ou sur SourceForge `http://www.`

`sourceforge.net` : on trouve énormément de codes sources). On récupère une archive « tar.gz » qu'on n'aura plus qu'à décompresser. Voir la section 8.2 pour savoir comment faire. Ensuite on entre dans le répertoire créé par la décompression. C'est le moment d'expliquer comment se compilent la majorité des logiciels libres. Il y a trois étapes :

- On vérifie si le logiciel contient le fichier « configure ». Si oui, suivre la procédure suivante. Sinon tu dois chercher comment compiler ton logiciel, en lisant sa documentation. Mais la plupart des logiciels utilisent la méthode suivante :
- On tape « `./configure` » : ce script vérifie la présence des logiciels et bibliothèques déjà installés dans le but de s'apercevoir s'il manque quelque chose. Par la même occasion, création du fichier « Makefile » qui est la recette de cuisine pour l'étape suivante. Si « ./configure » s'arrête avec une ligne du type « creating config.h » c'est bon. Si on te dit qu'il manque quelque chose, il faudra l'installer. Le plus souvent c'est des bibliothèques de développement : c'est pour cela qu'il faut installer le maximum de paquetages « devel » lors de l'installation de la distribution.

  S'il manque une librairie de développement, il suffit de taper « `urpmi libtruc-devel` » sous root et la librairie va être installée. Ensuite, en recommençant le « ./configure », on voit que cette fois il reconnaît la librairie.
- On tape « `make` » : cette commande va lire sa recette, le Makefile, et va se lancer toute seule dans la compilation du logiciel. S'il y a erreur, cela vient du code source, donc ce n'est pas de ta faute. Il ne faut pas se poser de questions : soit tu es programmeur et tu examines le code source, soit tu ne l'es pas et tu envoies en e-mail à l'auteur du logiciel pour lui signaler l'erreur. Envoie lui une copie des messages d'erreur pour qu'il comprenne de quoi il s'agit et normalement ça devrait marcher dans les prochaines versions du logiciel. Par contre si tout se passe bien il ne reste plus qu'à copier le logiciel dans les répertoires « système ». Réflexe : on passe en root avec « su ».
- On tape « make install » : install est une clause du Makefile qui dit quoi faire si tu tapes le mot « install » après la commande « make ». C'est justement le cas : cette clause dira toujours de copier les fichiers ainsi compilés dans « /usr/bin », « /usr/share » etc... C'est bien pour cela qu'il faut être root : on s'attaque aux répertoires interdits en écriture pour les utilisateurs normaux.
- Une fois « make install » effectué (il n'y a vraiment jamais d'erreur, à part des répertoires inexistants ou autre...) on n'a plus qu'à lancer le logiciel (le nom de la commande est souvent le nom du logiciel). On

n'a pas d'entrée dans les menus de KDE ou autre : la compilation ne s'intéresse pas à ça.
- On peut supprimer le répertoire qui a servi à compiler et aussi l'archive téléchargée (ou les stocker ailleurs que dans ton répertoire maison, pour ne pas l'encombrer).
- Certains logiciels proposent une désinstallation avec la commande « make uninstall ». Si c'est le cas mieux vaut garder le code source (ou seulemtent le Makefile) pour pouvoir désinstaller le logiciel. C'est vrai qu'avec les RPM la désinstallation est plus facile.

8.2 Décompresser toutes sortes d'archives

Et d'abord, c'est quoi une archive ? Si je te dis zip, tar.gz, tgz, ça te dis quelque chose ? Toujours pas ? Bon, alors je crache le morceau !

Une archive est comme un paquet cadeau. On y met plein de choses, on ficelle bien, et au final on a un paquet plus petit que tous les objets que l'on avait au début (c'est la compression). C'est un moyen d'économiser de la place, notamment sur Internet, car les temps de téléchargement sont encore longs aujourd'hui.

Créer une archive signifie qu'on prend en entrée les fichiers à archiver et qu'il ressort l'archive. L'archive est un autre fichier dont la taille est inférieure à la somme de tous les premiers. Désarchiver est l'action inverse : on prend en entrée le fichier d'archive et on récupère en sortie les fichiers d'origine. C'est l'ordinateur qui s'occupe de faire le travail en s'appuyant sur des algorithmes mathématiques. La seule chose qu'on doit lui dire, c'est le nom des fichiers d'origine (éventuellement le nom du fichier de destination).

Il existe plusieurs sortes d'archives, chacune étant plus ou moins compressée ou rapide à obtenir lors de sa création. Ce sont les deux critères majeurs qu'on prend en compte lorsqu'on veut créer une archive. Je vais énumérer les types d'archives les plus courants, mais sans entrer dans les détails. C'est juste pour la culture générale. Se référer à la section 5.1.3 pour plus d'informations sur les commandes qui compressent, celles qu'on utilise vraiment.

8.2.1 zip

Le format zip est le moins intéressant sous Linux. Bien sûr c'est celui que tout le monde utilise sous windows. On va le voir juste dans le cas où tu aurais des fichiers zip venant de windows.

Pour ziper des fichiers : la commande zip.

Pour déziper des fichiers : la commande unzip.

Note : ziper signifie « archiver au format zip » ! Ce sera de même pour la suite.

8.2.2 gzip

Le gzip est le format le plus utilisé car il est rapide et efficace.

Pour gziper (prononcer géziper) des fichiers : la commande gzip.

Pour gunziper (prononcer géheuneziper) des fichiers : la commande gunzip.

8.2.3 bzip2

Le bzip est bien plus efficace que le gzip (et donc beaucoup beaucoup mieux que le zip) mais il n'est pas encore très répandu. Cela peut s'expliquer par sa relative lenteur par rapport au gzip. En effet, une archive bzip2 est plus efficace au niveau de la compression, mais ce gain n'est pas gratuit : l'ordinateur doit travailler plus longtemps pour obtenir un meilleur résultat.

Le format bzip2 contient également une structure qui lui permet de s'auto-réparer au cas où il serait endommagé.

Pour bziper (prononcer béziper) des fichiers : bzip2.

Pour bunziper des fichiers : bunzip2.

8.2.4 tar

Cette commande ne sert pas à compresser des données. Pourtant, en étant le plus souvent utilisée de façon détournée, c'est à cela qu'elle sert tout le temps. On l'agrémente toujours d'une option qui ordonne d'utiliser gzip ou bzip2 pour compresser le fichier tar généré, et c'est bien gzip ou bzip2 qui fait le travail, pas la commande tar. A la base, elle sert simplement à concaténer plusieurs fichiers en un seul, les uns à la suite des autres, sans rien compresser. Voir 5.1.3 pour savoir comment on s'en sert.

8.2.5 Comparaison des formats

Comme rien ne vaut un tableau comparatif, j'en ai fait un. J'ai pris un répertoire qui contient 2.6 Mo de fichiers texte, PostScript et PDF. Je l'ai compressé en zip, tar.gz et tar.bz2.

	zip	tar.gz	tar.bz2
% processeur	92%	80%	94%
temps (s)	7.045 s	3.45 s	8.128 s
taille (Mo)	2.4 Mo	1.3 Mo	1.3 Mo
taille (Ko)	2 506 156 Ko	1 403 463 Ko	1 318 585 Ko
commande	zip -r rep.zip rep	tar cvzf rep.tar.gz rep	tar cvjf rep.tar.bz2 rep

Les mesures ont été faites dans les mêmes conditions, c'est-à-dire que le système n'était pas plus occupé lors d'une mesure que lors d'une autre.

Interprétation : le format zip est vraiment lamentable, puisqu'il va aussi lentement que le tar.bz2, alors qu'il occupe quasiment la même taille que l'original. Le tar.gz se situe entre des deux : avec un temps très rapide, il compresse relativement bien. Le tar.bz2 est encore plus lent que le zip mais on gagne un peu plus de compression par rapport au tar.gz.

C'est donc pour ça que le tar.gz est le plus utilisé.

Il existe d'autres formats de compression tels que le rar ou ace, mais je ne les ai pas encore testés de façon sérieuse.

8.3 Utiliser les périphériques amovibles

Les périphériques amovibles sont les lecteurs de disquette, de CD-Roms, les lecteurs ZIP, etc. . .

Il faut considérer que les lecteurs sont des partitions comme celles qui sont sur les disques durs. Avant, pour accéder aux lecteurs, on devait les monter (mount) et les démonter (umount) à chaque utilisation, et surtout démonter le lecteur de CD-Roms avant de l'éjecter. Aujourd'hui avec le « supermount », c'est automatique. Mandrake propose le supermount par défaut (sauf la version 8.1) alors qu'avec les autres distributions il faut l'activer séparément. Le fichier qui gère toutes les partitions est : `/etc/fstab`. Voici le mien :

```
/dev/hda1 /mnt/windows vfat user,exec,umask=0,codepage=850,iocharset=iso8859-1
0 0
/dev/hda5 swap swap defaults 0 0
/dev/hda6 / reiserfs defaults 1 1

/mnt/cdrom /mnt/cdrom supermount fs=iso9660,dev=/dev/cdrom
0 0
/mnt/floppyb /mnt/floppyb supermount fs=vfat,dev=/dev/fd1
0 0
```

```
   /mnt/floppy /mnt/floppy supermount fs=vfat,dev=/dev/fd0
0 0

   none /dev/pts devpts mode=0620 0 0
   none /proc proc defaults 0 0

   //Gros/partage /mnt/gros_partage_windows smbfs user,rw
0 0
   gros :/home/christophe /mnt/nfs nfs user,rw 0 0
```

Il y a un grand nombre de systèmes de fichiers. Ce fichier montre comment déclarer des disques en supermount. Normalement les distributions l'incluent pas défaut. On voit là où c'est encadré qu'il faudra aller dans le répertoire « /mnt/cdrom » pour en visualiser son contenu (taper « cd /mnt/cdrom » puis « ls »). Voir la figure 4.1 à la page 64 pour visualiser l'arborescence.

Détaillons le procédé de montage des disques. Il existe deux façons d'accéder aux lecteurs de disque amovibles (disquettes et cdroms) en écriture : en direct ou en différé. Y accéder en direct fait que le lecteur réagit tout de suite, alors qu'en différé il stocke ce qu'il a à faire puis il le fait au moment adéquat (on verra quand). Par exemple, tu décides de copier un fichier sur une disquette, puis un autre... et finalement tu ne veux pas que le premier y soit. En accès direct, tu feras travailler le lecteur de disquettes pour mettre le premier fichier, puis les autres, et tu le feras *encore* travailler pour supprimer le premier fichier. En différé, le lecteur ne travaillera pas du tout pour le premier fichier, puisqu'au final il retient que tu ne désires pas le garder. Belle économie !

Évidemment, pour profiter d'un accès en différé, tu dois dire au système d'exploitation : « OK j'ai fini d'utiliser la disquette, fais le bilan et écris ce qui doit être écrit ! ». Cette commande s'appelle « umount ». Elle s'utilise avec la commande « mount ». En fait, tu montes ta disquette, le système lit son contenu. Ensuite tu y travailles, sans surmener le lecteur, puis au moment de démonter la disquette, le système écrit ce qui doit être écrit. Voici la ligne qu'il faut voir dans /etc/fstab pour utiliser la disquette en accès direct :

```
/dev/fd0 /mnt/floppy vfat user 0 0
```

Le premier élément /dev/fd0 désigne le fichier qui représente le lecteur de disquettes, /mnt/floppy est un répertoire *choisi arbitrairement* par lequel tu accéderas à ta disquette. /mnt/floppy est un répertoire qui doit exister pour que le montage fonctionne. vfat est le système de fichiers utilisé pour la disquette. On dit toujours « disquette PC » mais c'est faux : on doit dire « disquette fat32 ». Je rappelle que sous Linux, on donne le nom de vfat à la fat32. On pourrait utiliser des disquettes formatées en reiserfs, mais comme la disquette est utilisée pour communiquer, mieux vaut utiliser le format fat32 de windows[2]. La commande utilisée pour monter une disquette est :

```
mount /mnt/floppy
```

C'est pas la mort tout de même. Bien sûr, la commande va lire la ligne énoncée juste au dessus et faire le rapprochement entre les deux /mnt/floppy. Si tu as changé /mnt/floppy dans le fichier /etc/fstab, alors il faut changer l'argument de la commande mount. Ensuite tu vas utiliser la disquette en accès différé, puis au moment de l'éjecter, il faudra penser à taper

```
umount /mnt/floppy
```

sinon la disquette ne sera pas totalement remise à jour. D'ailleurs sur les vieilles stations Unix, la commande umount éjectait les disquettes, et on ne pouvait pas les éjecter à la main (pas de bouton physique d'éjection, comme sur les Macintosh). On peut bien sûr accéder aux CD-Roms en accès différé, mais comme ils sont par définition en lecture seule, je ne vois pas trop l'intérêt. Voici tout de même la ligne à mettre dans /etc/fstab (basée sur le modèle de la ligne pour la disquette) :

```
/dev/cdrom /mnt/cdrom iso9660 user 0 0
```

Eh oui, le format de fichiers des CD-Roms est bien l'iso9660. Il faut bien vérifier que /dev/cdrom (c'est un lien) pointe bien vers ton périphérique de CD-Rom :

```
[christophe ~]$ ll /dev/cdrom
lrwxrwxrwx 1 root root 3 déc 21 11 :10 /dev/cdrom -> hdb
```

Chez moi /dev/cdrom pointe vers /dev/hdb, ce qui veut dire que c'est le périphérique esclave sur la nappe IDE n°1. On crée ce lien pour plus de commodité.

Examinons le cas de l'accès direct : dès que tu demandes la copie d'un fichier, même si tu décides de l'effacer tout de suite après (mais l'ordinateur ne peut pas le savoir !), le lecteur écrit ce fichier. L'avantage est que tu n'as pas à monter et démonter à la main la disquette. Il suffit de l'éjecter physiquement. La ligne à inclure dans /etc/fstab pour accéder en accès direct au lecteur a été citée plus haut, lorsque j'ai montré mon fichier /etc/fstab en entier.

8.4 Créer son premier document sous LYX

J'ai mis cette section dans les tâches les plus courantes car j'estime que n'importe doit pouvoir goûter aux joies de LYX et de LATEX. Je me sers quotidiennement de LYX tellement il me plaît.

[2]Eh oui j'en reviens pas mais je l'ai dit !

Bien que la documentation de LyX soit très bien faite, je me permets d'ajouter ce très court tutoriel, qui te montrera comment créer ton premier document PostScript ou PDF sous LyX. Créer un document PDF est quelque chose de fabuleux pour un utilisateur de Windows, qui pense qu'il est obligé d'utiliser le logiciel Adobe Acrobat pour fabriquer de tels fichiers. Pas du tout : sous Linux il existe plein de logiciels capables de le faire (en fait ils produisent du PostScript, que l'on convertit en PDF à l'aide du filtre ps2pdf).

Pratiquement tous les logiciels sont capables de produire une sortie PostScript, puisque c'est le format d'impression. Au lieu d'envoyer les données à l'imprimante, le logiciel les redirige vers un fichier PostScript. Sous KDE par exemple, tous les logiciels utilisent la même boîte de dialogue standardisée pour imprimer. Dans cette boîte de dialogue, on peut choisir d'imprimer le fichier ou de rediriger l'impression vers un fichier PostScript.

Pourquoi utiliser LyX comme préparateur de textes au lieu d'un autre logiciel ? La réponse est que LyX est si intéressant que ça aurait été un crime de ne pas te le présenter. Pour savoir quelle est la philosophie de LyX, lis la section 7.1, c'est *très* important : LyX ne fonctionne pas comme les autres traitements de texte.

Au début tu arrives dans LyX. Tu crées un nouveau fichier (Fichier-Nouveau !) et tu tapes quelque chose. Pour créer le titre, tu choisis « Titre » dans la boîte en haut à gauche de l'écran (pour l'instant il y a marqué « Standard »). Hop ! Le texte passe en plus gros, centré...

Appuie sur Entrée. Si tu appuies encore une fois, ça ne fonctionne pas et c'est normal. Entrée est la touche qui permet de changer de paragraphe, et on ne peut pas changer deux fois à la suite. C'est illogique. Ensuite, tape « Introduction » et choisis « Section » dans la boîte en haut à gauche. De même, le texte prend la forme d'un titre de section. Amuse toi avec les différents environnements (notamment les listes) et quand tu es prêt, appuie sur CTRL+T. Tu vas voir le résultat s'afficher dans une fenêtre. C'est le fichier PostScript.

Pour créer le fichier PDF, il faut rajouter une instruction dans la boîte de dialogue accessible par « Format ▹ Préambule LaTeX ». Rajoute dans cette boîte de dialogue la ligne suivante :

```
\usepackage{times}
```

Ainsi, les PDF générés utiliseront la police vectorielle times au lieu d'une police bitmap.

Sauvegarde ton document au format LyX, puis clique sur « Fichier ▹ Exporter ▹ PDF » (pas PDFLatex !) et voilà ! Dans ton répertoire maison tu trouveras un fichier PDF portant le nom du fichier sauvagardé au format LyX.

On peut visualiser ce PDF avec Acrobat Reader ou GhostView.

8.5 Convertir un fichier quelconque en fichier PostScript

Les fichiers PostScript (leur extension est par convention machin.ps) sont d'une souplesse incroyable. On peut en une ligne de commande extraire des pages d'un fichier vers un autre, inverser l'ordre des pages, mettre plusieurs pages en une et bien plus (voir notamment la section 8.7). Tout cela est réalisé grâce aux filtres contenus dans les paquetages « psutils » et, optionnellement, « a2ps ». On comprend donc que l'on veuille convertir un fichier html, PDF ou encore les images en PostScript. Tu l'auras compris, le PostScript est le format directement destiné à être imprimé.

Il existe de nombreux programmes pour convertir des fichiers en PostScript. On distinguera deux catégories :

8.5.1 Convertir un fichier texte en PostScript

On utilisera l'utilitaire « enscript » qui permet énormément de choses. On ne va pas tout détailler, mais par exemple pour convertir un fichier texte avec des lignes qui dépassent à droite de la feuille, et retourner en format paysage, on tape :

```
enscript texte.txt -B --word-wrap -R -o texte.ps
```

et le fichier texte.ps existe. Avec l'option « -E », enscript met en forme les codes source (C, C++, HTML) avec les mots réservés en gras…

8.5.2 Convertir un fichier graphique en PostScript

On utilisera l'utilitaire « convert » qui est très complet. Utiliser convert est très simple : imagine que tu veux convertir une image gif en jpeg, il suffit de taper

```
convert image.gif image.jpg
```

et c'est fini. Bien sûr on peut faire énormément plus de choses, comme jouer sur le contraste, le gamma, la luminosité, appliquer des effets, redimensionner… voir la page manuel très complète (man convert).

8.6 Convertir des images

Si tu as plusieurs images GIF et que tu veux les transformer en JPEG, c'est très simple. Il suffit d'exécuter ce petit script :

```
for i in *.gif
do
convert $i $i.jpg
rename .png.jpg .jpg *.png.jpg
done
```

Voilà ! On est en présence d'une simple boucle for, qui va traiter tous les fichiers gif (*.gif) les uns après les autres. Ensuite c'est le programme convert qui va détecter que tu auras mis « .jpg » après $i en deuxième argument, donc il va comprendre qu'il faut produire un fichier jpg. Le problème est que convert va garder l'extension .png et on se retrouvera avec des fichiers .png.jpg. On renomme donc les fichiers .png.jpg en .png avec la commande rename.

8.7 Imprimer un fichier PostScript sous la forme d'un livre

Souvent on trouve des fichiers PostScript (le format de fichiers texte le plus facile à traiter sous Linux) et on a envie de les imprimer. Seulement on se rend vite compte que le nombre de feuilles de papier qui sortent de l'imprimante défilent à une allure incroyable. J'ai donc écrit un petit script qui permet d'agencer les pages d'un document PostScript (ou autre, on verra comment convertir d'autres fichiers en PostScript) sous la forme d'un livre, comme ça il rentre 2 pages sur chaque feuille de papier, elles sont imprimées recto-verso et on peut tourner les pages.

Le fonctionnement est très simple : on tape « `lpr-book xxx.ps` » dans une console, une boîte de dialogue nous dit d'attendre que les feuilles recto sortent de l'imprimante, puis on les retourne et on les remet dans le bac à feuilles de l'imprimante. On appuie sur Entrée et le verso s'imprime. Il ne reste plus qu'à plier en deux les pages ainsi obtenues et on a notre petit livret. À propos, si tu imprimais ce petit survol de Linux ?

Voici donc ce script (je ne détaillerai que les commandes PostScript, il faut lire la section 5.5 pour plus de renseignements sur les scripts) :

```
# !/bin/bash
```

Cette première ligne sert à indiquer quel est l'interpréteur qui va exécuter le script. Dans ce cas, c'est Bash, qui se trouve dans le répertoire /bin. Normalement, les « # » servent à mettre des lignes de commentaires (non lues par l'interpréteur) mais sur cette première ligne, on doit mettre « #! » suivi de l'interpréteur.

```
tmp=/tmp/book.$$
usage(){
echo "syntaxe : lpr-book [fichier].ps"
exit -1
}
[ -z "$1" ]&&usage
psbook $1|psnup -2 -m20>$tmp
```

On crée un livre avec psbook, et on met deux pages en une avec psnup, en rajoutant une marge de 20 .je ne sais pas quelle unité.

```
psselect -o -r $tmp|lpr
```

On choisit d'imprimer les pages paires avec -o comme « odd ». dans l'ordre inverse avec -r comme reverse

```
dialog --msgbox "Attendez que l'imprimante s'arrête.
Remettez les feuilles de papier qui sont sorties,
sans les retourner, dans la réserve de papier de
l'imprimante. Appuyer Entrée pour continuer." 0 0
```

Afficher une boîte de dialogue, voir « man dialog » et « man gdialog »

```
psselect -e $tmp|lpr
```

On choisit d'imprimer les pages impaires avec -e comme « even »

```
rm -f $tmp
```

Fin du script !

Voilà, il suffit de copier-coller ce petit script dans un fichier texte `/usr/bin/lpr-book`, de le rendre exécutable avec `chmod 755 /usr/bin/lpr-book` (il faut être root pour le copier dans /usr/bin) et c'est installé. Si tu ne sais pas ce qu'est chmod, regarde la section 5.3.

8.8 Bien utiliser son clavier

Sous Linux le clavier est bien mieux exploité que sous windows. Voyons le cas des accents :

Si tu appuies sur « AltGr + μ » et ensuite une voyelle, tu la verras apparaître cette voyelle avec un accent grave. La touche μ est juste à côté de Entrée. Par exemple, je peux taper ces lettres : à, è, ì, ò, ù, À, È, Ì, Ò, Ù, sans quitter mon clavier. Les utilisateurs des Macintosh seront contents.

Si tu appuies sur la touche « windows » et en même temps sur la touche « 4 » (située en dessus du « e »), puis que tu relâches tout et que tu tapes ensuite une voyelle, tu auras un accent aigu. Exemples : á, é, í, ó, ú, Á, É, Í, Ó, Ú.

Fais de même avec la touche windows + « 3 » et ensuite une voyelle et tu auras un tréma : ä, ë, ï, ö, ü, Ä, Ë, Ï, Ö, Ü.

Mais beaucoup plus simple : caps lock permet de saisir les accents ! Sous windows, très souvent, on voit du texte comme ça : LE PR2SIDENT LOGE 0 L42LYS2E. Stupide, non ? En effet, le « e » accentué se trouve sur la même touche que le 2, et caps lock produit un 2... Rien de tout cela sous Linux ! Tu tapes é avec caps lock, et tu vois É à l'écran. Vraiment pratique.

Ensuite c'est bon de pouvoir taper les *vrais* guillemets français : « et ». On les obtient respectivement avec les combinaisons de touches AltGr+z et AltGr+x.

Pour taper le signe « copyright », on tape sur la touche « windows » et sur o en même temps. Ensuite, si on appuie sur c, on a ©. Si on appuie (après Win+o) sur e on obtient « $\frac{1}{2}$ ».

Chapitre 9

Résumé de l'installation d'une distribution

Sommaire

Ce chapitre traite un très vaste sujet : l'installation d'une distribution. La toute première fois c'est très difficile, *surtout si on fait n'importe quoi*, mais ensuite ça devient de plus en plus facile.

On va supposer que tu as déjà windows installé sur ta machine car c'est le cas le plus répandu : windows occupe tout le disque dur. De toute façon c'est bien plus facile d'installer Linux sur une machine sans windows donc on va étudier le cas le plus complexe (on a besoin de faire la section des préliminaires si windows est installé).

Installer Linux n'est pas si difficile qu'on le croit mais il faut suivre une méthode rigoureuse, sans rien oublier. Sinon ça rate, et je vais te donner des exemples de ratage que j'ai vécus à la section 10, afin que *tu* ne rates pas ton installation.

Je ne vais pas trop détailler, et tu ne trouveras pas d'informations hyper-précises, comme « clique sur le bouton xxx pour faire ceci » car cela dépend de chaque distribution. De plus, les manuels d'installation des distributions sont la plupart du temps clairs et précis, et je ne voudrais en aucun cas leur faire de l'ombre. Le manuel d'installation de la distribution est souvent placé sur le CD-Rom pour les gens (comme moi !) qui n'achètent pas le pack de la distribution mais qui l'« achètent » dans un magazine. Donc si je te dis de consulter le manuel, trouve-le sur le CD-Rom, au format HTML ou PDF.

Les grandes étapes :

9.1 Préliminaires indispensables

9.1.1 Re-partitionnement

Si windows est installé, on ne va pas, tu t'en doutes, installer Linux par dessus. On va pousser windows dans un coin du disque dur (voir la section 2.8) puis on aura le champ libre pour Linux.

Défragmenter absolument le disque sous windows avant toute chose, sinon certaines données seront forcément écrasées par Linux, ce qui serait dommage. Voir le cas de Samuel à la sous section 10.1.

9.1.2 Collecte de renseignements

On doit à présent collecter le maximum d'informations possible sur le matériel qui constitue ton ordinateur. Pour cela il faudra écrire tout ce que tu pourras trouver dans les propriétés systèmes du panneau de configuration windows.

Il faut savoir que Linux se contrefiche de la marque commerciale du matériel que tu possèdes. Linux veut savoir quel est le processeur (chipset) qui est utilisé par ton matériel. Le reste ne compte pas. Par exemple, j'ai acheté la très bonne carte son « Sound Blaster PCI 64 » mais le processeur qu'elle utilise est un « Ensoniq ES1371 » et c'est ça qui est important. Ce renseignement permettra au noyau de Linux de savoir quel module (sorte de pilote) charger en mémoire pour piloter la carte son. D'ailleurs ce module s'appelle « e1371.o », ce qui est très parlant. Comme ça, toutes les cartes à base de ce chipset fonctionneront sous Linux grâce à *un seul* fichier de 16 Ko seulement ! Voilà un système d'exploitation qui est propre et net.

Bien sûr, le débutant risque de s'affoler à la lecture de ces quelques lignes : la plupart du temps, le matériel est automatiquement détecté ! ! ! Mais il faut quand même faire la collecte d'informations au cas où... Il est très pénible d'avoir à recommencer plusieurs fois l'installation d'une distribution à cause d'un oubli, car on doit retourner sous windows pour chercher les informations manquantes, et reprendre à zéro l'installation !

J'en vois certains venir, qui se disent que sans windows on ne peut pas collecter ces informations, et qui essaient de me faire dire que windows est quand même bien pratique... balivernes ! On peut très bien obtenir tous ces renseignements par d'autres moyens : ouvrir son PC, et regarder sur les cartes électroniques le nom des chipsets. C'est ce que j'ai fait avec un copain car windows nous prend parfois tellement pour des idiots qu'il ne nous donne que le nom commercial du matériel, fourni par le pilote, et ne nous donne même pas le nom du chipset. Dans des cas comme celui-ci, on prend un tournevis et on ouvre le PC... Et vlan dans les dents, pour ceux qui essayaient de me faire dire quelque chose que je n'aurai pas dit.

Bon, on se calme, tout ça n'est pas obligatoire. Bien sûr tu n'as pas besoin d'ouvrir ton PC pour installer Linux !

Les renseignements les plus utiles sont :

- la taille du disque dur
- noter les partitions créées tout à l'heure
- les caractéristiques de la carte vidéo et de l'écran (fréquence de rafraîchissement maximale, minimale, nombre de couleurs et résolutions maximum...)
- de combien de mémoire RAM tu disposes
- as-tu du matériel SCSI ?
- bref, c'est mieux de savoir ce qu'il y a sous le capot de ta machine.

9.2 Booter le CD-Rom

Alors voici le moment d'insérer le premier CD-Rom de ta distribution dans le lecteur. En général, c'est le CD-Rom étiqueté numéro 1 qui permet d'installer la distribution. Le logiciel d'installation varie d'une distribution à une autre, mais toutes ont adopté le même système : le CD-Rom d'installation est « bootable », ce qui veut dire qu'il contient une zone spéciale (située sur le premier secteur du CD-Rom) contenant elle-même le programme d'installation. L'ordinateur, lors de la phase de démarrage, va lire le CD-Rom pour voir si par hasard un programme de boot se trouve sur le CD-Rom. Si ce n'est pas le cas, alors il démarre le système d'exploitation qui est sur le disque dur. Cet ordre de démarrage est paramétrable grâce au Bios : la petite puce qui gère le démarrage de l'ordinateur, avant même qu'il n'ait chargé le système d'exploitation.

Nous allons essayer de booter le Cd-Rom sans avoir besoin de trafiquer le Bios. Au moins si ça marche on ne se sera pas fatigué. Sinon, on se fatiguera, tant pis ! Donc : on place le premier CD-Rom dans le lecteur et on allume l'ordinateur. Si windows démarre comme d'habitude, c'est fichu, on va devoir régler le Bios pour qu'il regarde d'abord du côté du CD-Rom et non pas du côté du disque dur. Par contre si tu vois apparaître un programme qui t'étais inconnu jusqu'à ce jour, c'est gagné ! Voyons comment faire dans le « pire » des cas. En fait, c'est un jeu d'enfant si on sait déjà le faire... comme toujours !

Pour entrer dans le setup du Bios et ainsi le configurer, on appuie sur la touche dédiée - juste après avoir appuyé sur le bouton « marche » du PC. Cette touche varie d'un Bios à l'autre : F10, Del (ou Suppr), F2, Esc (ou Echap)... En fait, on commence par voir le modèle de la carte graphique pendant peut-être $\frac{1}{10}$ de seconde, puis le décompte de la mémoire RAM se déroule sous nos yeux, et au même moment, en bas de l'écran, on voit un message du type :

```
Appuyer sur Del pour entrer dans le Setup
```

ou, en anglais :

```
Press Del to enter Setup
```

Le « Setup » est le programme qui permet de modifier et de sauvegarder les réglages du Bios. La touche peut être Del (suppr en français), F1 ou encore F2.

Une fois arrivé dans le Bios, il faut se débrouiller pour faire en sorte de faire passer le CD-Rom *avant* le disque dur pendant la phase de démarrage. Souvent, c'est la disquette qui est avant le disque dur. Là je ne peux pas trop t'aider, car il existe une multitude de Bios différents. En général, c'est dans la rubrique « Boot settings ». Une fois que c'est réglé, quitte *en sauvegardant.*

Recommence le démarrage de l'ordinateur toujours avec le CD-Rom dans le lecteur. Normalement tu dois maintenant voir le programme d'installation de ta distribution.

Note spéciale pour les vieux ordinateurs : certains ordinateurs trop anciens n'ont pas la possibilité de booter avec le lecteur de CD-Roms. La solution est d'utiliser une disquette de démarrage. Si elle c'est pas fournie, il faut la créer. En général, on la fait sous Dos ou windows avec l'utilitaire « rawrite » qui est fourni souvent dans le répertoire « dosutils » du CD-Rom d'installation. Encore une fois, mieux vaut lire le manuel d'installation de ta distribution (sur papier ou à l'écran).

Lors du boot sur le CD-Rom, tu verras tout un tas d'informations défiler à l'écran, puis tu arriveras sur un écran graphique, sauf si ta distribution n'a pas d'installation graphique, comme Debian ou Slackware, ce qui n'est absolument pas un signe de faiblesse, au contraire. Cette procédure d'installation va te poser un nombre assez grand de questions, copiera les logiciels choisis, et tu auras enfin Linux sur ton disque dur. Tout ça sans redémarrer l'ordinateur une seule fois (on redémarre pour quitter la procédure d'installation une fois que tout est terminé).

9.3 Choix de la souris

L'étape suivante est destinée à choisir le type de souris utilisée. Sous Linux, c'est le serveur graphique XFree86 qui gère également les périphériques comme le clavier, la souris, l'écran ou même les tablettes graphiques. Il communique avec la souris suivant le protocole qu'utilise la souris. Il n'existe pas beaucoup de protocoles, les deux majeurs sont :

- les souris IntelliMouse de Microsoft (pour une fois cette firme mérite la majuscule car ses produits « matériels » sont très robustes).
- les souris Logitech
- etc...

Cette étape te demandera le modèle de ta souris et le programme d'installation trouvera tout seul le protocole à utiliser.

9.4 Partitionnement du disque

Cette étape a pour but de découper ton disque dur en plusieurs morceaux de façon à loger chaque système d'exploitation dans un de ces morceaux.

Voir la section 2.8 pour plus de détails, car je l'ai déjà expliqué.

9.5 Installation des paquetages

C'est l'heure de choisir les programmes que l'on veut installer ! Et pas question de penser comme avec windows : on trouve *tout ce qu'on veut* sur les CD-Roms de la distribution de GNU/Linux. Il suffit de choisir les programmes qu'on veut installer puis ils seront copiés.

Mais qu'est-ce qu'un paquetage ? Commençons par faire la critique de ce qui se passe lorsqu'on est sous windows. Si on veut installer un logiciel, on lance le traditionnel programme « Setup.exe » qui va afficher l'assistant d'installation du logiciel (le plus souvent « InstallShield »). Cet assistant va d'abord nous faire lire la licence du logiciel à installer (comme quoi on n'a pas le droit de le copier, que le logiciel ne nous appartient pas, et bla bla bla. . .), va nous faire choisir le répertoire dans lequel on installe le logiciel, puis l'emplacement des raccourcis dans le menu démarrer, et enfin va copier les fichiers du logiciel à installer dans le répertoire choisi. Évidemment on est forcé de redémarrer l'ordinateur après tout ceci, et dans le cas où on nous laisse le choix, le système n'est pas fiable si on n'a pas redémarré.

Pour prendre connaissance des points négatifs qui se cachent sous cette procédure d'installation des logiciels , il faut absolument comprendre ce qu'est une bibliothèque de programmes (on dit aussi une librairie, à cause d'une mauvaise traduction de l'anglais « library »). C'est très simple : lorsque plusieurs logiciels utilisent des fonctionnalités en commun, on regroupe ces fonctionnalités dans un autre programme : une bibliothèque. Par exemple aujourd'hui la majorité des logiciels savent gérer des images au format jpeg : ceci est dû au fait qu'il existe la bibliothèque « libjpeg ». Tout logiciel qui voudra utiliser des images au format jpeg n'aura qu'à appeler les fonctionnalités de la bibliothèque correspondante. Les bibliothèques ne servent à rien en elles-mêmes, mais ce sont les logiciels qui vont aller piocher des fonctionnalités à l'intérieur. Le but est bien sûr d'alléger ces logiciels, car ce serait très « lourd » si chaque programme réinventait la roue en incluant les mêmes fonctionnalités à chaque fois. Comme d'habitude, le monde se divise en deux catégories : ceux qui gèrent bien les bibliothèques, et ceux qui les gèrent mal.

Prenons l'exemple d'un logiciel qui a besoin d'une bibliothèque nommée « libtruc » (par convention sous GNU/Linux le nom des bibliothèques commence par « lib »). Imaginons également que ni le logiciel ni la bibliothèque ne sont installés. Le but est donc d'installer ce logiciel.

- Si on est sous windows, ce sera très simple, car le logiciel est fourni avec toutes les bibliothèques dont il a besoin. On cliquera sur « Setup.exe » et hop tout sera installé dans la foulée.
- Si on est sous GNU/Linux, on ne va déjà pas trouver un « Setup.exe » ! C'est là qu'interviennent les paquetages. On trouvera donc, soit sur les CD-Roms de la distribution, soit sur Internet, le paquetage du logiciel (un seul fichier) et le paquetage de la bibliothèque (un seul fichier aussi). Si on cherche à installer le logiciel alors que la librairie n'est pas installée, le système nous préviendra par un message qu'il faut *d'abord* installer la bibliothèque « libtruc » pour que le logiciel fonctionne. C'est donc à ce moment-là qu'on va installer la bibliothèque demandée. De la même façon, si la bibliothèque « libtruc » a besoin d'un autre paquetage pour fonctionner, le système nous le signalera, et il faudra d'abord installer les paquetages nécessaires. En remontant le cours des dépendances entre paquetages, on arrive enfin au moment où le logiciel est installé.

  Cela peut paraître compliqué (mais pas tant que ça), de toute façon il existe aujourd'hui des utilitaires comme « rpmdrake » qui nous simplifient considérablement la tâche, en nous permettant de cocher les logiciels que l'on veut installer, et en installant automatiquement *toutes* les dépendances, c'est-à-dire tous les paquetages nécessaires pour faire fonctionner le logiciel voulu.

Le système employé sous GNU/Linux est hyper sécurisé : on ne peut pas mettre son système en bazar. Prenons l'exemple de quelqu'un (malheureusement cela arrive très souvent) qui veut installer un logiciel sous windows. Comme d'habitude, il clique sur « Suivant » et « OK » plusieurs fois et au final il se retrouvera avec un ordinateur qui refuse de démarrer correctement. L'explication est très rapide : tous les logiciels sous windows ré-installent leurs bibliothèques lors de leur procédure d'installation. Si le logiciel est plus ancien que le système d'exploitation, il va installer une version de la bibliothèque plus ancienne que celle qui était présente. On se retrouve donc avec une vieille bibliothèque et un système récent. Le problème est que les autres logiciels *déjà présents* ont besoin de la version récente de cette bibliothèque. Tous ces logiciels refuseront de fonctionner et enverront un message « Mauvaise version de machintruc.dll ». Les utilisateurs de windows ont une fois au moins entendu parler des fameuses « dll » (dynamic linked library), la plupart du temps dans ce contexte-là. Et ils ne savent pas ce que c'est la plupart du temps. La solution que préconise Microsoft est de réinstaller windows : les versions récentes des bibliothèques seront ainsi réinstallées.

D'où la paranoïa qu'ont certains parents à refuser que leurs enfants installent des jeux sous prétexte qu'ils vont tout mettre à sac.

Sous GNU/Linux le système de paquetages est donc très propre. Chaque paquetage contient plusieurs informations :

- Le nom du logiciel ou de la bibliothèque
- La version du paquetage
- La description du paquetage : ce que permet de faire le logiciel (ou la bibliothèque)
- La liste des fichiers à installer
- L'emplacement des fichiers à installer
- Des règles de dépendance (quel paquetage est nécessaire à celui-ci)
- Un script à lancer après la copie des fichiers

Il existe plusieurs sortes de paquetages : les RPM et les DEB. Les RPM (RedHat Package Manager) ont été inventés par RedHat pour installer plus facilement leur distribution. Dans la même logique il existe les DEB qui présentent plus de possibilités, créés par Debian. La distribution Slackware fonctionne avec des archives au format « .tgz » (contraction de « .tar.gz »). Ce sont de simples archives compressées contenant un fichier indiquant les règles de dépendance et tout le reste.

Lors de l'installation de la distribution, on va choisir les paquetages à installer. Mieux vaut voir grand et installer les paquetages par défaut plutôt que d'avoir à les installer au compte-goutte par la suite. Les paquetages contenant « devel » dans leur nom sont des paquetages de développement (contenant le code source du paquetage « non devel »). C'est une très bonne idée de les installer car si on veut par la suite compiler un logiciel, il nous faudra installer les paquetages « devel » nécessaires.

Une fois qu'on a fait son choix, le programme d'installation de la distribution va effectivement installer ces paquetages.

Voici un petit complément sur la façon de nommer les fichiers RPM. Ce complément n'a rien à voir avec la procédure d'installation de la distribution, mais c'est intéressant de le savoir.

Souvent, tu verras des paquetages nommés « truc.noarch.rpm ». L'élément « noarch » signifie que ce n'est pas un logiciel qui va être installé mais des fichiers supplémentaires, comme des images, des sons, des fichiers de langage (pour traduire les logiciels)...

Ou encore, parfois on trouve l'élément « devel » dans le nom du RPM. Ça veut dire que ce RPM contient tout ce qu'il faut pour développer (c'est-à-dire programmer) un logiciel incluant la fonctionnalité proposée par le paquetage. Le plus souvent « devel » s'applique aux librairies. Par exemple, si tu veux développer un logiciel qui lit des fichiers MP3, il faudra installer le RPM « libmp3-devel-version.rpm », où « version » correspond au numéro de version du paquetage. Pour les programmeurs, ces paquetages sont les headers. Avec le paquetage devel, on pourra compiler le logiciel qui a besoin de lire des MP3. Et pour l'utiliser (après l'avoir compilé), on aura aussi besoin d'utiliser la librairie MP3 elle aussi compilée : on la trouve dans le paquetage qui a le même nom que le paquetage devel... sauf qu'on ne met pas le mot « devel ». Donc la librairie compilée MP3 sera dans le paquetage « libmp3-version.mp3 ».

Tu auras compris que les paquetages commençant par « lib » sont des librairies : ce sont des bouts de programmes qui sont très souvent utilisés par les logiciels. On regroupe les libriries par fonctionnalités : libmp3 pour lire les MP3, libjpeg pour décoder les images jpeg,...

9.6 Configuration

Cette étape varie d'une distribution à une autre : elle permet de configurer le réseau local, l'accès à Internet par modem (on entre les coordonnées du fournisseur d'accès)...

On configure l'imprimante en choisissant le pilote d'impression, puis on règle, on teste... C'est très différent d'une distribution à l'autre.

Ensuite on crée un compte pour chaque utilisateur. Je vais te donner l'éternel conseil : crée un compte pour toi, même si tu es le seul utilisateur de la machine, car il est « interdit » de travailler en root tout le temps. Chaque utilisateur bénéficie d'un répertoire maison de la forme : `/home/<nom>` et d'un mot de passe. Voir la section 4.4.2 pour plus de renseignements au sujet du répertoire maison.

9.7 Configuration de l'affichage

On règle le serveur XFree86 en lui donnant les caractéristiques de la carte graphique, la résolution, les couleurs... la plupart du temps c'est détecté automatiquement par le programme d'installation de la distribution.

9.8 Configuration des entrées de Lilo

Lilo (Linux Loader) est le petit programme qui se charge de te faire choisir entre Linux et windows au démarrage de l'ordinateur. La plupart du temps tout est prêt, mais on peut rajouter d'autres systèmes d'exploitation, et au démarrage, on pourra s'en servir.

9.9 Redémarrage

Et bien sûr, comme il faut bien essayer Lilo, on va voir si ça marche! Enlever CD-Roms et disquettes, redémarrer l'ordinateur et c'est fini! On profite de Linux.

On remarquera que Linux n'est pas un système friand de redémarrages. Combien de fois windows demande-t-il de redémarrer pendant l'installation? Trois fois, je crois! Et même pendant l'utilisation quotidienne, combien de fois faut-il redémarrer sous windows à la moindre installation d'un logiciel? Sans arrêt! Ce n'est vraiment pas une marque de fiabilité. Linux ne demande *jamais* de redémarrer, et certains serveurs (gros ordinateurs) sollicités sans arrêt, par exemple ceux qui sont chargés de nous envoyer les pages Internet, sur lesquels des tas de clients sont connectés en permanence... bref très chargés en travaux, n'ont pas redémarré depuis six ans, à Marseille! Je me permets de signaler que j'allume mon ordinateur le matin et que je l'éteins le soir, et parfois dans la journée j'ai compilé, installé, mis à jour des bibliothèques de programmes *sans avoir besoin de redémarrer une seule fois*, sans même parler de la stabilité du système qui ne plante jamais donc ne nécessite pas de redémarrer brutalement.

Chapitre 10

Ratages vécus

Sommaire

Eh oui, c'est classique, quand on ne s'y connaît pas, on rate et on pleure après. A moins d'être un génie, mais c'est très très rare. Comme disent les gens qui emploient les proverbes : on apprend en se trompant. Seulement, lorsqu'on est prévenu des pires bêtises, ou tout du moins des bêtises les plus stupides, il vaut mieux ne pas les faire, tu es d'accord avec moi, non ? Comme promis, je vais te raconter ce qui s'est passé les toutes premières fois où j'ai installé Linux (la première fois c'est toujours le plus difficile).

10.1 Comment Samuel a perdu une partie de windows

Pauvre Samuel ! A cause d'un stupide oubli (de notre part à tous les deux, en plus !) il a perdu une partie de windows. Explications :

Comme je l'ai déjà dit, windows utilise un système de fichiers, la FAT32 (qui fonctionne certes, mais qui n'est pas au top des performances) qui a la bonne idée d'écrire les informations n'importe où, en fait là où est la tête de lecture au moment de recevoir l'ordre d'écrire sur le disque dur. Ce qui fait que le disque dur se fragmente, et il faut donc le *défragmenter* régulièrement si on veut des performances optimales. Dérisoire !

Pour l'information, la majorité des systèmes de fichiers sous Linux, comme le ReiserFS, écrit les données au bon endroit, c'est-à-dire là où c'est le mieux au niveau de la fragmentation. On veut écrire un petit fichier, eh bien ReiserFS va l'écrire là où il y a juste la place pour lui (au cas où le disque dur serait utilisé comme un gruyère) et dans le cas le moins bon, il va grignoter de l'espace libre plus loin sur le disque au cas où il en reste, vers des espaces inhabités...

Mais revenons à nos moutons : Samuel et moi avions oublié de défragmenter la partition de windows avant d'installer Linux. Forcément, sachant que windows occupe à peu près toute sa partition (même si on ne l'utilisait presque pas) et que l'on a tronqué le disque dur en deux pour créer la deuxième partition étendue pour Linux, on a perdu la dernière moitié de windows !

Son disque dur est de 4 Go et windows utilisait 300 Mo à peu près. On pensait qu'on pouvait largement couper le disque dur en deux sans défragmenter, mais c'est faux !!! Même si windows utilisait dans ce cas *seulement* 10% du disque dur, monsieur « je me crois tout permis » s'était donné la peine d'occuper toute sa partition. Bonjour le gruyère : une donnée par ci, une autre par là, mais toutes les deux éloignées de centaines de kilomètres !

Et depuis ce jour je n'ai plus jamais oublié de défragmenter le disque dur avant d'installer Linux. Oui, je sais, c'est long, mais tu n'as qu'à le faire pendant la nuit ou pendant une journée de travail où tu ne serais pas chez toi... La défragmentation nécessite *en plus* que tu ne touches pas à l'ordinateur du tout, et elle peut durer des heures !

10.2 Comment Alexandre a vu disparaître 8 Go de son disque dur

En ayant utilisé divers logiciels DOS ou windows qui affirmaient savoir gérer les partitions du type Linux, sans parler du fdisk de windows qui ne sait rien faire à part la fat32 (ou 16, d'accord), je me suis mélangé les pinceaux et du coup une partition de 8 Go est devenue invisible. Je ne sais plus trop comment cela s'est produit, mais l'erreur vient du fait que je n'étais pas assez informé au sujet des partitions, mais aussi au sujet des logiciels à utiliser ou non.

Donc je recommande d'utiliser les logiciels qui sont fournis par la distribution Linux, et de ne pas t'amuser avec les sharewares sous DOS.

Ensuite il vaut mieux éviter avec certains sharewares de créer des partitions logiques car elles sont mal gérées.

10.3 Pourquoi mon père et Laurent ont été embêtés

lilo : c'est le chargeur de démarrage, le programme qui permet de choisir entre windows et Linux au démarrage. Il se loge dans le premier secteur du disque dur, qui est appelé le secteur d'amorçage ou MBR (Master Boot Record). Sans Lilo, on ne peut plus accéder à Linux sauf si on a créé une disquette de démarrage pour Linux.

Une constante : ne pas installer windows après Linux. Ils l'ont fait. Deux raisons :

- windows efface systématiquement le premier secteur du disque dur, celui qui contient Lilo. Comme par hasard ! On ne peut plus accéder à Linux. Il faut donc redémarrer sous Linux après avoir installé windows *avec une disquette de démarrage* et là : catastrophe car on a oublié de la créer, cette fameuse disquette. Heureusement que la distribution a prévu un programme sous DOS permettant d'en créer une, encore faut-il se rappeler exactement quelles partitions on a faites, car ce petit programme DOS de secours demande quelle partition Linux amorcer !

– Après avoir réinstallé Lilo depuis Linux, après avoir créé notre disquette de démarrage, parfois windows refuse toujours de s'amorcer. La solution est de réinstaller windows (en entier bien sûr !) et là on retombe dans le premier cas où cette réinstallation a encore viré Lilo. On ne peut plus à nouveau accéder à Linux, et ainsi de suite...

Au bout d'un moment (une journée de perdue) on se promet qu'on installera désormais windows en premier, au prix d'une défragmentation précédant l'installation de Linux.

Alors quelques conseils :

– imprimer le fichier /etc/fstab qui contient toutes les informations sur les partitions (on peut ainsi reconstituer la disquette de démarrage)
– noter à la main sur du papier les partitions créées lors de l'installation (par précaution)
– créer *systématiquement* une disquette de démarrage lors de la procédure d'installation de la distribution
– puis, bien sûr, *ne pas* installer windows après Linux.

Évidemment, si on ne peut pas faire autrement et que le mal est déjà fait, on essaye et on verra bien. C'est au petit bonheur la chance, ou plutôt au bon vouloir de Microsoft : acceptera-t-il de démarrer s'il reconnaît Linux ou nous fera-t-il croire que tous nos problèmes viennent de Linux ? (comme peuvent le suggérer parfois les lignes d'assistance au dépannage par téléphone)

10.4 Correction : une astuce pour installer windows après Linux

Cette astuce n'est pas de moi, mais de quelqu'un que j'ai rencontré à une réunion Linux dans le cadre du LUG d'Aix en Provence : l'AXUL.

Tu as installé Linux en ayant laissé une partition vide pour windows, de type fat32 (le nom est vfat sous Linux). Le problème précédent est que windows, lors de son installation, va détecter la partition fat32, mais va également voir d'autres partitions *non fat32*. Comme il voit la partition fat32 *déjà formatée*, il va s'imaginer que l'on a déjà installé windows, mais que celui-ci est souillé par d'autres partitions. Il croit que parce qu'on a installé windows (en plus c'est faux !) on n'a pas à installer autre chose. Il va dire « attention il n'y a pas que du fat32, le disque dur contient des partitions défectueuses, voulez-vous les formater ? ». Et voilà comment on perd Linux si on répond « oui ». Si on répond « non », windows refuse de s'installer. La ruse est de lui réserver une partition fat32 sans qu'il ne s'en aperçoive. Il faut donc, sous Linux, formater une partition en reiserfs (par exemple) et lui donner le type fat32 (vfat). Comme ça à l'installation de windows, il verra une fat32 *non formatée* donc pensera qu'on n'a pas windows (c'est le cas) et se comportera en vaillant chevalier : il nous la formatera et s'installera dessus.

Le problème est donc que si on réserve une partition déjà formatée windows croit qu'il est déjà installé. Si on réserve une partition non formatée (ou plutôt formatée autrement) il pensera qu'il arrive sur une terre non cultivée.

Il faut bien dissocier le formatage du type de partition. On peut assigner un type de partition tout en gardant le formatage précédent. Lorsqu'on formate une partition, il faut le faire selon le type de celle-ci pour que tout soit en harmonie. C'est justement cette harmonie qu'il faut briser pour réaliser cette astuce.

Chapitre 11

Divers

Sommaire

11.1 À propos de ce document

J'ai commencé à l'écrire début juillet 2001, en pensant faire un mémo des commandes les plus courantes et les plus utiles sous Linux. C'est cependant la partie que j'ai faite en dernier, tellement les autres chapitres étaient agréables à réaliser. J'ai finalement orienté mon travail vers une présentation générale de Linux, un survol comme l'indiquait le titre. D'ailleurs le titre original était : « Survol de Linux à la vitesse éclair destiné aux débutants mais aussi à ceux qui s'y connaissent, dans le but d'avoir un résumé sous les yeux ». Drôle de résumé !

J'ai écrit ce document dans plusieurs buts :
- aider les utilisateurs (débutants) à s'y retrouver dans le monde de Linux si différent de celui de windows ;
- transmettre mon expérience de débutant sous Linux aux autres utilisateurs, afin de rendre les premiers contacts moins difficiles ;
- combattre les préjugés qu'ont certains utilisateurs de windows ;
- combattre certaines tendances stupides qui poussent les gens à faire des dépenses inutiles ;
- et puis me faire plaisir !

Je n'ai en aucun cas pensé à me faire de l'argent avec ce document. J'espère qu'il t'aura apporté quelque chose. Si jamais tu es encore victime des préjugés que j'ai cités tout au long du document, essaye Linux, va voir des démonstrations de Linux dans les LUGs, par exemple. Si tu es toujours contre, alors je ne peux plus rien faire pour toi. Envoie-moi un e-mail pour m'expliquer !

Voir la section 7.1 pour savoir quels sont les logiciels libres que j'ai utilisés pour créer ce document.

Ce document est libre et gratuit. Il est sous licence FDL (Free Documentation license), cf `http://www.gnu.org/copyleft/fdl.html`. Tu peux le distribuer gratuitement à toutes les personnes que tu veux, et je t'y encourage. N'importe qui peut citer un ou plusieurs passages de ce document, mais il devra citer le titre du document et devra être en mesure de fournir le document en entier, et/ou d'indiquer l'endroit où l'on peut le télécharger en entier. (`http://cbenz.tuxfamily.org`)

11.2 À propos de l'auteur

J'ai 18 ans. Je suis un utilisateur des ordinateurs depuis l'âge de 5 ans (j'ai commencé avec un 8086 ainsi qu'un Amstrad, chez le voisin !). Je m'intéresse à Linux depuis à peu près un an mais je m'y suis mis pour de bon il y a quatre mois seulement[1]. Pourquoi si tard ? Eh bien parce que je pensais comme pensent les personnes qui sont caricaturées à la section 2.1. Je m'y suis mis et j'ai tout de suite accroché.

Je viens de windows et j'en étais encore relativement content il y a un an ! Mais je suis *encore plus* content de Linux, mais alors 100, 1000 fois plus ! J'espère que toi aussi tu vas faire comme moi et que tu seras aussi content ! J'ai réussi à convaincre mon père qui lui aussi hésitait à installer Linux. J'en parle tout autour de moi, et je me surprends à dire : « non merci, ce logiciel ne m'intéresse pas car j'utilise des logiciels libres » ou encore « je connais un bien meilleur logiciel sous GNU/Linux équivalent à celui que vous me proposez ». Je dis souvent « connaissez-vous les logiciels libres ? et GNU/Linux ? » puis j'enchaîne sur tous les avantages. Ce que je veux dire par là, c'est que Linux s'adresse à tout le monde, pas seulement aux gens qui connaissent windows sur le bout des doigts.

Depuis 6 mois maintenant je n'ai plus utilisé windows ! J'ai même envie de le chasser à grands coups de repartitionnement du disque dur, mais il faut bien le garder pour les autres personnes ayant accès à l'ordinateur (bien que je sois en train de les « convertir » à Linux !).

Note à l'attention des amoureux de windows :

> Oui, j'ai aimé windows, ce n'est pas un argument pour dire que windows est bien (si le dollar t'énerve, lis la section 3.2 pour comprendre). En fait, il faut savoir de quoi on parle, et surtout dans quel cadre. Disons-le tout net : je me place dans le cadre de l'informatique au sens noble, celle où tout est propre, net, précis et bien conçu, tu comprendras mieux ce que ça signifie en lisant ce document. Nous ne nous plaçons sûrement pas dans le cadre du presse-boutons (ou du clique-souris, devrais-je dire !). Si tu utilises ton ordinateur pour lire des vidéos DivX, pour copier des CD-ROMs avec « easy cd creator », aller glandouiller sur le net, ou encore taper des lettres dans word, tu dois te dire que windows est l'idéal ! *Par contre* tu ne feras pas d'informatique ! Et je ne joue pas sur les mots : tu feras du multimédia, de la bureautique, ou de la recherche d'informations. Mais tu ne connaîtras jamais les plaisirs de l'informatique pure : celle où tu crées, où tu donnes des ordres à ton « esclave » qu'est l'ordinateur. Je suis sûr que c'est l'ordinateur qui te mène par le bout du nez. Tu comprends ce que je veux dire ? Mon objectif dans ce document est de présenter l'informatique dans ce sens là. Et crois moi, si j'ai pris la peine de rédiger ce document, bénévolement, pendant trois

[1]Depuis que j'ai commencé à écrire ce document, ça fait un peu plus longtemps !

mois d'affilée, rien que pour le plaisir de faire fleurir l'envie chez quelques personnes de s'intéresser à cette informatique-là, de les voir partager l'immense plaisir, cet incroyable dépaysement que j'ai vécu, si je me suis décarcassé pour résumer tout ce que j'avais appris en deux ans de façon à ce qu'un débutant s'y retrouve, si j'ai enquêté sur les questions de débutants auprès de mes proches, bref si j'ai fait tout ce travail, c'est que ça vaut vraiment le coup pour toi d'ouvrir ton esprit et d'au moins t'intéresser à ce nouveau monde. J'aurai alors atteint mes objectifs. Je n'ai pas dit qu'il faut que tu te lances tête baissée, bien sûr il faut avoir les connaissances, le temps de le faire tranquillement, mais ça vaut vraiment la peine.

On pourrait dire que windows est bien dans le cadre d'une utilisation bête et méchante, avec des utilisateurs qui se fichent de l'éthique du logiciel, mais là, encore un fois, cet argument n'est pas valable ! Tout ce que j'ai dit (DivX, gravure, etc. . .) peut être fait sous Linux de façon plus performante, et pas plus compliquée ! Pas de chance pour les amoureux de windows. En fait Linux part sur les bases de l'informatique bien pensée, et les logiciels multimédia, bureautique, etc. . . ne sont plus que des extensions au système qui existe déjà. On ne peut pas dire, il est vrai, que Linux est le système idéal pour faire de la PAO ou de la création musicale, mais les logiciels sont justement en train d'être réalisés. Ces deux domaines sont encore manquants. Par contre aujourd'hui on peut *très bien* faire du graphisme, de la 3D, écouter de la musique, de la bureautique, lire des vidéos, graver des CD. . .

Je me demande aussi pourquoi certains s'acharnent à défendre Microsoft. Ils n'ont aucun intérêt à le faire. C'est un comportement de contemplation niaise, de béatitude. Bien sûr que Bill Gates a été un pionnier, bien sûr qu'il est un héros mythique ! En fait tout ça, c'est le reflet de l'image de marque que Microsoft s'est *lui-même* inventée. On dit toujours que sans Microsoft, l'informatique n'en serait pas arrivée à un tel point. C'est sûr que pour s'être démocratisée, l'informatique s'est démocratisée ! Mais est-ce dans la bonne direction ? Les utilisateurs sont-ils bien soignés ou au contraire profite-t-on d'eux ? Bill Gates a fait avancer l'informatique dans sa poche *avant tout*. Ne l'oublions pas. Dès le début il a racheté à grand coups de marketing des logiciels concurrents pour les revendre sous le nom de Microsoft. Ces logiciels n'étaient pas très performants mais les gens les achetaient déjà parce que « tout le monde les avait ». Ceux qui défendent Microsoft sont trop souvent mal informés sur ses pratiques, ils le vénèrent en fait. C'est stupide, il faut suivre une cure anti-propagande.

Il reste un dernier argument à ceux qui ne se sont pas mis à GNU/Linux : ils n'ont pas le temps de s'en occuper. On ne peut pas avoir des journées de 300 heures, c'est sûr. Par contre, on peut lire rapidement ce document et parler du logiciel libre autour de soi, aller voir un club Linux dans sa ville un soir en passant, bref, arrêter de penser de façon bornée. Pas le temps de s'intéresser à Linux n'entraîne pas l'idée que Linux n'est pas bien. « Je n'aime pas les épinards » ne signifie pas que les épinards ne sont pas comestibles.

Merci de méditer cela.

11.3 Remerciements

Je te remercie d'avoir lu ce document jusqu'au bout. Si tu es arrivé jusque là, c'est que tu as accroché. J'ai alors rempli mon contrat : t'ouvrir les yeux. Merci sincèrement, tu viens donc de me prouver que je n'ai pas travaillé pour rien.

Je remercie Samuel (un grand ami) pour avoir parcouru avec moi cette longue route depuis plus d'un an, et pour toutes ces après-midi lors desquelles nous avons échangés nos découvertes Linuxiennes si riches en astuces, qu'on a progressé et aimé Linux de façon incroyable, jusqu'à l'adoration aveugle et passionnée, comme on vénère un grand chef spirituel.

Également mon papa pour avoir encore et toujours posé les questions de débutants les plus naïves, qui m'ont aidé à bâtir mes arguments. Aujourd'hui il utilise Linux mais à cause de sa profession il est obligé de garder windows. Il est amené à utiliser des logiciels de traitement audio numérique, des séquenceurs (Cubase) et ce genre de logiciel, il est vrai, manque encore cruellement sous GNU/Linux.

Je remercie Alexandre (un autre grand ami) pour m'avoir demandé d'installer Linux chez lui, à l'époque où je n'y connaissais rien. Tout a foiré lamentablement, les 3/4 de son disque dur ont disparu, et cela m'a permis de m'obliger à m'intéresser de façon sérieuse au partitionnement des disques durs.

Je remercie, dans la lignée, tous les matériels qui n'ont pas fonctionné du premier coup, comme l'imprimante de mon grand-père, ce qui m'a aussi permis de me battre pour les faire fonctionner, et d'extraire de ces expériences la connaissance utile, pas celle du genre « clique et ça marche ».

Je remercie grandement toutes les personnes qui ont relu et corrigé les dernières imperfections : mon grand-père, Cédric Ferraris, Raphaël Stora, et tous les autres.

Remerciements spéciaux et pas du tout sincères à Monsieur ce cher et malino-débrouillardo-commerço-pourri pour avoir réussi à agrandir sa fortune personnelle de quelques sous et pour avoir fait du monde informatique ce qu'il en est aujourd'hui. Lui étant un informaticien de la première heure, époque que je n'ai malheureusement pas connue, je n'arrive toujours pas à le croire quand il dit qu'il est fier de ce qu'il a fait. Déjà à l'époque il faisait des coups de marketing douteux (comme commercialiser un logiciel déjà libre ou « voler » par un tour de passe-passe le système Dos de son ami pour en faire Ms-Dos) alors que d'autres informaticiens se concentraient sur leurs réalisations. Comme diraient les familles du terroir : rira bien qui rira le dernier, que le meilleur gagne, tel est pris qui croyait prendre, et l'arroseur finira par être arrosé.

Je remercie spécialement l'IUT d'Aix en Provence (département Informatique, là où je fais mes études) pour nous apprendre la programmation, l'utilisation de vi, etc... sous GNU/Linux.

Je remercie toutes les personnes du LUG de Digne : on s'est retrouvé plusieurs fois lors de « soirées Linux » au Cybercafé municipal. Ce sont eux qui m'ont entraîné vers GNU/Linux, merci à Nicolas, Boris, Jean-François et les autres[2].

11.4 Conclusion

Voilà, on arrive au bout du document, le moment est venu pour moi de te lâcher la main et te laisser aller par toi même. C'est donc le début de ton cheminement sous Linux, à propos duquel tu vas découvrir chaque jour plus de choses, et sans cesse tu auras le confort d'utiliser tous les logiciels que tu veux en restant dans la légalité. En cas de problème, ma boîte e-mail est toujours grande ouverte...

Nous n'avons même pas parlé des possibilités réseau sous GNU/Linux, c'est pourtant le meilleur système d'exploitation dans ce domaine. C'est drôle, les mauvaises langues disent toujours que Linux n'est bon que pour les réseaux... en tout cas j'ai réussi à écrire un petit livre sur GNU/Linux sans effleurer ce gigantesque domaine. Cela prouve donc que GNU/Linux sert aussi à autre chose !

[2] J'ai toujours peur d'en oublier...

Nous avons constaté de quelle manière a évolué un projet lancé par cet illuminé de Richard Stallman dans les années 1980 : une foule de personnes y a participé pendant une dizaine d'années, et Linus Torvalds, qui essayait de reproduire le grand système Unix qui tournait sur les serveurs de l'université dans laquelle il étudiait, a apporté une grande pierre à l'édifice. Il voulait créer un Unix pour PC, car à l'époque il n'existait que ces trucs du genre de ms-dos. Il s'est lancé, et après quelques mois de travail, il proposait son projet sur un newsgroups (groupes de nouvelles), en signalant que tout le monde pouvait le télécharger gratuitement et le modifier ! C'était en 1991. Son noyau a par la suite été choisi par les distributions[3]. Aujourd'hui, dix ans après, regarde où ça en est ! On se croirait dans SimCity (pour ceux qui ne savent pas ce que c'est, c'est un jeu dans lequel on doit construire et faire se développer une ville). Les personnes qui affirmaient que pour développer du logiciel costaud il fallait une entreprise modèle, hyper-hiérarchisée... se sont trompées ! Linus a travaillé non seulement à distance mais aussi sans grande organisation.

Ce projet n'avait aucun impact médiatique, ce qui montre que c'est donc bien grâce aux qualités du logiciel libre que GNU/Linux s'est développé. Ceux qui y ont participé, comme je le fais aujourd'hui avec cette minuscule contribution, l'ont fait afin de bénéficier d'un système d'exploitation qui leur va comme un gant, et non pas pour s'enrichir. Et pour le plaisir d'avoir participé au plus grand projet collaboratif de tous les temps. C'est ce point que pas mal de personnes n'arrivent pas à saisir, et répliquent « mais il faut bien les faire vivre, ces programmeurs ! ». C'est tout bêtement une question de choix entre la qualité (aussi bien fonctionnelle que morale), et la commercialisation. Nous avons montré tout au long de ce document que ces deux objectifs n'allaient pas si bien ensemble.

Oublions donc peu à peu le logiciel esclavagiste et utilisons, apprécions et faisons connaître les logiciels libres !

[3]Attention, ne pas croire que Linux n'est que le fruit du travail de Linus Torvalds !

Index