

Migrer vers une solution logicielle libre.

Pierre LO CICERO (p.locicero@netplus.ch) et Yoann LE BARS (le.farfadet.spatial@free.fr)

29 janvier 2004

Version1.1

Copyright (c) 2003 Pierre LO CICERO and Yoann LE BARS.

Pierre LO CICERO, Conseil en informatique, analyste-programmeur, administration de parcs informatiques.

Yoann LE BARS est étudiant en DEA — Diplôme d'Études Approfondies, diplôme universitaire de niveau bac +5, préliminaire au doctorat — d'analyse numérique et contributeur occasionnel à OpenOffice.org.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled « GNU Free Documentation License, » section 14, page 37.

Table des matières

| | | |
|----------|--|-----------|
| 1 | Préambule | 4 |
| 1.1 | Au sujet des références de cet article. | 4 |
| 1.2 | Présentation générale de l'article. | 4 |
| 1.3 | Remerciements. | 4 |
| 2 | Comparaison entre formats de fichiers ouverts et propriétaires. | 5 |
| 2.1 | Cas des formats de fichiers propriétaires. | 5 |
| 2.2 | Cas des formats de fichiers ouverts. | 6 |
| 2.3 | Conclusion. | 6 |
| 3 | Brève histoire du logiciel libre. | 7 |
| 4 | Les principes du logiciel libre. | 8 |
| 5 | Le système d'exploitation libre GNU/Linux. | 9 |
| 5.1 | Qu'est-ce qu'un système d'exploitation ? | 9 |
| 5.2 | Avant Linux. | 9 |
| 5.3 | Brève histoire de Linux. | 11 |
| 5.4 | Linux par rapport aux autres systèmes d'exploitations. | 14 |
| 5.5 | Les diverses distributions de Linux. | 15 |
| 5.6 | D'autres systèmes d'exploitations libres. | 16 |
| 6 | Pourquoi s'orienter vers les logiciels libres ? | 16 |
| 6.1 | Généralités. | 16 |
| 6.2 | Aspects pragmatiques. | 17 |
| 7 | Les principaux logiciels libres. | 19 |
| 7.1 | Solutions bureautiques et édition. | 20 |
| 7.2 | Editeur de textes. | 20 |
| 7.3 | Utilitaires pour Internet. | 20 |
| 7.4 | Outils de développement. | 20 |
| 7.5 | Graphisme. | 21 |
| 7.5.1 | Bitmap | 21 |
| 7.5.2 | Vectoriel | 21 |
| 7.5.3 | Image de synthèse. | 21 |
| 7.6 | Vidéo. | 21 |
| 7.7 | Son et musique. | 21 |
| 7.7.1 | Traitement du son | 21 |
| 7.7.2 | Séquenceur MIDI | 22 |

| | | |
|-----------|--|-----------|
| 7.7.3 | Synthèse sonore | 22 |
| 7.8 | Administration et réseau. | 22 |
| 7.9 | GNU/GPL sous Microsoft Windows | 22 |
| 8 | La délicate période de transition. | 22 |
| 9 | Exemple d'une migration de logiciels propriétaires vers logiciels libres. | 23 |
| 9.1 | Définition des besoins | 24 |
| 9.2 | Évaluation du matériel | 24 |
| 9.3 | Se renseigner | 24 |
| 9.4 | Choix de la distribution Linux | 24 |
| 9.5 | Installation proprement dite | 25 |
| 9.6 | Autres considérations | 25 |
| 9.7 | Exemple de migration en conservant Microsoft Windows | 25 |
| 9.7.1 | Installation de Microsoft Windows 2000 | 26 |
| 9.7.2 | Installation de Linux | 26 |
| 9.7.3 | Problème rencontré | 26 |
| 9.7.4 | Conclusions | 26 |
| 10 | Point de vue personnel des auteurs. | 26 |
| 10.1 | Yoann LE BARS | 27 |
| 10.1.1 | Au sujet des principes des logiciels libres. | 27 |
| 10.1.2 | Il ne s'agit pas d'une lutte mais de bon sens. | 28 |
| 10.1.3 | L'opportunité proposée par ce mode de fonctionnement. | 29 |
| 10.1.4 | Conclusion. | 29 |
| 10.2 | Pierre LO CICERO | 30 |
| 10.2.1 | Pratique | 30 |
| 10.2.2 | À propos de « morale » | 31 |
| 10.2.3 | Philosophique | 32 |
| 10.2.4 | En conclusion | 32 |
| 11 | Pour conclure. | 33 |
| 12 | Références pour compléter cet article. | 34 |
| 13 | Au sujet des logiciels cités dans l'article. | 36 |
| 14 | GNU Free Documentation License | 37 |

1 Préambule

Cet article a vu le jour suite à deux demandes exprimées sur la liste de diffusion `debutant@linux-mandrake.com`, à savoir fournir un argumentaire pour la migration d'un organisme de formation vers GNU/Linux, ainsi qu'un document utile aussi bien pour un chef d'entreprise que pour un utilisateur familial.

Les auteurs ont rassemblé les différents éléments à leur disposition, ceux qu'ils ont pu glaner au cours de leurs expériences, afin d'offrir à la communauté informatique dans son ensemble une synthèse objective et juste.

Toute personne ayant des informations à donner sur ce sujet ou voulant apporter des éléments est encouragée à prendre contact avec les auteurs. De plus, cet article est sous la licence libre de documentation GNU et, de ce fait, tout le monde est invité à la diffuser comme bon lui semble (voir partie 14 page 37).

Yoann LE BARS (`le.farfadet.spatial@free.fr`) maintient une base de données des sites mettant à disposition cet article, ainsi que de leurs webmestres, pour les tenir informés des nouvelles versions de ce document. Il demande à toute personne publiant cet article de bien vouloir le contacter afin de l'intégrer dans cette base de données, ceci bien sûr en respectant la loi « Informatique et Libertés, » c'est-à-dire en garantissant le droit de regard sur les informations données, qui ne seront pas transmises à un tiers.

1.1 Au sujet des références de cet article.

Le lecteur trouvera, section 12 page 34, de nombreux liens et des références bibliographiques. Chacune de ces références possède un numéro qui nous permet de les citer tout au long de l'article. Ainsi, par exemple, le reportage « Nom de code : Linux » porte le numéro [12] et c'est par ce numéro que nous y ferons référence dans tout l'article.

1.2 Présentation générale de l'article.

Avant tout, nous comparerons les formats de fichiers ouverts aux formats de fichiers propriétaires, et nous parlerons de l'histoire du logiciel libre, de sa philosophie, avant de se pencher sur le système d'exploitation libre GNU/Linux. Ensuite, il sera temps d'aborder les raisons d'adopter les logiciels laissant leurs codes sources librement consultables et modifiables. Les principales offres logicielles libres seront aussi abordées, quel que soit le système d'exploitation, ainsi qu'un cas concret de migration de Microsoft Windows vers GNU/Linux. En conclusion, les auteurs présenteront leurs points de vues personnels.

1.3 Remerciements.

Les auteurs remercient les relecteurs de cet article :

- Nadine LO CICERO, Monique URO, Claire HEURTEBISE, Isabelle DUVIGNAUX, Chantal et Arthur AGAPOFF, Lise et Dominique BONFANTI, pour l'important travail de correction orthographique ;
- Nicolas METROZ, instituteur et utilisateur exclusif de Microsoft Windows, pour sa relecture et ses conseils ;

- André SALAÛN, directeur d'école primaire, utilisateur de GNU/Linux depuis plusieurs années, pour ses précisions, précieux conseils ainsi que la revue de certaines formulations inadéquates ;
- Sophie GAUTIER, responsable de la partie francophone du projet OpenOffice.org, pour sa contribution à la correction orthographique et ses remarques sur la formulation et les licences ;
- Bertrand VUARNESON, dont l'utilisation de GNU/Linux est typiquement celle qualifiée de familiale, pour ses remarques pertinentes au sujet des formulations ;
- Bénédicte LEGUAY, novice en matière d'informatique, pour avoir apporté un point de vue extérieur sur cet article.

2 Comparaison entre formats de fichiers ouverts et propriétaires.

La portée de ce paragraphe va au-delà des seuls logiciels libres. En fait, il s'applique à tout individu ou entreprise utilisant un ou plusieurs ordinateurs. Il traite d'un problème très important : la communication, ainsi que de la pérennité des informations qui forment cette communication. Ceci est très largement inspiré de la conférence de Thierry STØHR (voir [6]).

Qu'est-ce qu'un format de fichier ?

Un format de fichier est tout simplement la forme ou, si l'on préfère, la structure dans laquelle des données vont être stockées d'abord dans la mémoire de l'ordinateur, puis dans un fichier. Ce format peut être ouvert, c'est-à-dire que tout un chacun peut avoir accès à ses spécifications et savoir, concrètement, quelle est la méthode utilisée pour stocker les données, ou alors propriétaire, c'est-à-dire que la méthode de stockage des informations n'est pas accessible.

2.1 Cas des formats de fichiers propriétaires.

Les formats propriétaires sont produits par des logiciels utilisant une méthode et une structure occulte. Il est donc absolument nécessaire de posséder le logiciel créateur afin de pouvoir relire le document.

Il va de soi que si l'on veut diffuser ce document, on ne peut le diffuser qu'à des personnes possédant le logiciel original. Cependant, les problèmes que cela induit sont bien plus importants :

- Les formats de fichiers peuvent évoluer d'une version à l'autre du logiciel les produisant. Pour peu que nous possédions une version postérieure à celle de nos interlocuteurs, ceux-ci ne pourront lire le document qu'à la condition expresse d'avoir acquis la nouvelle version du logiciel.
- La société éditrice de ce logiciel peut décider de ne plus assurer le support du format. On peut citer le cas de Microsoft Publisher, dont le format n'est pas compatible d'une version à l'autre ; ou encore celui Microsoft Quick Basic, un environnement de développement, pourtant un produit phare des années 80, qui plus est très utilisé professionnellement dont le format n'est plus lisible actuellement.
- À tout moment, une entreprise peut décider d'arrêter la production d'un logiciel, notamment pour des raisons de rentabilité ou d'évolution de la technologie.
- Il peut même arriver que la société, aussi importante soit-elle, disparaisse — le cas récent d'ENRON montre qu'une entreprise *a priori* prospère peut être démantelée du jour au lendemain.

- Une société peut être rachetée et, restructuration aidant, il est certain qu'une part de leur production sera abandonnée. Exemple Ami Pro, un traitement de texte qui fut très utilisé mais qui n'est désormais plus développé.

Quelles en sont les conséquences ?

Dans tous ces cas de figure, il est évident que le format diffusé ne sera plus supporté, et donc que les fichiers produits ne seront plus lisibles.

L'utilisation de formats propriétaires induit une dépendance forte envers les entreprises productrices de ce type de logiciels. Conserver l'accès aux fichiers dépend du bon vouloir voire de la bonne fortune de ces sociétés.

Quelques exemples de formats propriétaires : le format de Microsoft Publisher, celui de Quark X-Press, de Microsoft Word, de Microsoft Excel, de Quattro Pro.

La suite bureautique OpenOffice.org (voir section 7 page 19) par exemple est capable de lire les formats propriétaires de la suite Microsoft Office. Ceci, pourtant, n'est rendu possible que grâce à une phase d'ingénierie inverse, technique qui doit être ré-appliquée à chaque nouvelle version du logiciel.

2.2 Cas des formats de fichiers ouverts.

L'accès aux spécifications d'un format libre, permet l'interopérabilité, c'est-à-dire la possibilité, pour des logiciels différents de celui qui a produit le document, d'exploiter ce même document. Cela va au-delà d'un simple lecteur, il s'agit aussi de la possibilité offerte à n'importe quel organisme ayant un accès autorisé aux données contenues dans ces fichiers d'utiliser un logiciel différent du logiciel producteur.

Un format ouvert permettra à tous développeurs, à toutes entreprises, de concevoir et d'écrire un programme spécifique pour tirer parti de ce format de fichier. Créer des interactions entre les différents logiciels ou applications spécifiques, se fera de manière simple, rapide et efficace.

Même si dans le cas extrême où un format évolue de façon à ce que la compatibilité ascendante ne puisse plus être garantie, l'ancienne structure sera toujours accessible, et par voie de conséquence les données contenues dans la structure de ce format, aussi.

Quelques exemples de formats ouverts : HTML et XML (qui est un format structuré)(dans la mesure où le document respecte les spécifications W3C), PDF (Portable Document Format), PostScript (PS), rtf (Rich Text Form) lorsque le document respecte les normes, texte brut (txt), SGML (qui est un format structuré), T_EX ou L^AT_EX(qui sont des formats structurés), JPEG (compressé), MPEG (compressé), OGG (compressé), WAVE, TIF, AVI.

2.3 Conclusion.

Il est essentiel de comprendre qu'un logiciel propriétaire peut tout à fait utiliser des formats ouverts.

Mais une question se pose : pourquoi ces logiciels utilisent-ils exclusivement des formats qui sont propriétaires ?

La réponse est simple et semble unique : pour se rendre indispensables. C'est-à-dire obliger l'utilisateur à toujours acheter les nouvelles versions de ces mêmes logiciels pour pouvoir profiter de son travail passé, et ainsi s'assurer que tout un chacun doit obligatoirement utiliser leurs services.

Pourtant ce n'est pas la seule réponse :

En agissant ainsi, certaines entreprises produisant ces logiciels cherchent à s'assurer le contrôle des formats d'information. Non pour effectuer une censure — de toute façon assez hypothétique — mais bien pour le contrôle de l'information, par l'intermédiaire de ses vecteurs.

Sans forcer le sens de ce qui est dit plus haut, il s'agit bel et bien d'un piège dont la partie visible est purement économique et dont la partie invisible représente un réel danger.

Il faut rajouter ceci :

s'arroger le droit exclusif d'une technique (la structure du fichier), alors que celle-ci ne dépend que de l'intelligence humaine au service d'une conception rationnelle et pratique, est une erreur. Une structure n'est en général qu'une spécification pensée par rapport à un but donné. Dans la majorité des cas ces spécifications sont du domaine public parce que fondamentalement susceptibles d'améliorations. A bien y penser, le format de fichier fermé est même dangereux économiquement car cela n'aboutit qu'à construire une cage qui enferme ce type de format en le rendant incapable d'évoluer. L'incompatibilité des formats au fur et à mesure des évolutions logicielles, la dépense exagérée des énergies humaines ou financières trouvent, toutes, leurs sources dans une aberration de ce type.

3 Brève histoire du logiciel libre.

On fait souvent remonter l'histoire de l'informatique à la construction de l'ENIAC au cours de la deuxième guerre mondiale. Cette machine permettait d'automatiser les calculs. Ce premier ordinateur avait une structure suffisamment simple pour être parfaitement connue par les opérateurs. Cependant, très rapidement, les ordinateurs ont acquis une grande complexité. Deux problèmes concrets se sont alors posés :

- Rendre suffisamment aisée la création de nouveaux programmes.
- Conserver une certaine facilité d'utilisation.

Pour résoudre cela, il fallut concevoir un environnement propre à la machine : le systèmes d'exploitations 5, page 9. Il fallut de plus concevoir un langage formel qui puisse être compris par cette machine, le langage de programmation, permettant d'écrire des instructions aptes à être exécutées. Ce langage représente la première des interfaces entre l'homme et la machine et est appelé code source. Pour que la machine comprenne ensuite ce qui à été écrit, ce code source est compilé, c'est-à-dire traduit en binaire dans lequel toutes les instructions sont représentée sous la forme de 1 et 0.

A cette époque, le monde de l'informatique évoluait presque exclusivement dans celui de la recherche, fondamentalement habitué à partager ses découvertes. Même si le logiciel libre n'existait pas formellement de manière légale, c'était néanmoins une réalité indiscutée.

Cependant, au fur et à mesure des avancées techniques, et surtout de l'industrialisation de l'informatique, les enjeux économiques ont prévalu faisant évoluer les logiciels dans la direction d'objets purement commerciaux, vendus sous forme compilée, ou binaire, et donc non-modifiable. Le fait économique eut un impact paradoxal. Alors que les tâches confiées à l'ordinateur devenaient plus complexes, les projets et développements tendaient à s'isoler. Le risque de ralentissement, voire de stagnation, de ces projets ne pouvait devenir qu'un problème crucial.

Conscient des implications et des conséquences de cette fermeture, un jeune chercheur du laboratoire d'Intelligence Artificielle du MIT (Masachuset Institute of Technology), Richard

STALLMAN, créa la Free Software Foundation en 1983, dont l'objectif était de promouvoir un autre modèle, dans lequel les logiciels sont librement utilisables et modifiables par tout un chacun. Il démissionna du MIT en 1984 pour lancer le projet GNU — prononcez « gnou, » — projet de système d'exploitation libre proche du système UNIX (voir partie 5 page 9). La Free Software Foundation posa un cadre légal au logiciel libre.

Le projet GNU développa de nombreux outils mais connut, lui aussi, une période de stagnation, principalement due à un manque de ressources humaines, ce projet évoluant dans un cadre restreint et discret.

En 1991, en Finlande, un étudiant en informatique, Linus Benedict TORVALDS, allait changer la donne en créant le coeur d'un système d'exploitation qu'il nomma Linux.

Ce fut une aubaine pour le système GNU qui, en intégrant ce noyau, devint un système d'exploitation complet et fonctionnel tout en lui permettant d'élargir son audience. Depuis GNU suscite un intérêt sans cesse croissant de la part de la communauté mondiale des chercheurs et étudiants en premier lieu, puis de l'utilisateur professionnel et, enfin, de l'utilisateur de type familial.

4 Les principes du logiciel libre.

Comme indiqué précédemment, la Free Software Foundation créa un cadre légal au logiciel libre, en mettant notamment au point la licence *GNU Generic Public Licence* (GPL).

Ce cadre légal garantit les quatre libertés suivantes, telles que définies par Richard STALLMAN :

1. L'utilisateur a la liberté d'exécuter un logiciel libre pour quelque motif que ce soit ;
2. L'utilisateur a la liberté de modifier un logiciel libre afin qu'il corresponde mieux à ses besoins. Dans la pratique, pour que cette liberté prenne effet, il faut pouvoir accéder au code source, puisqu'opérer des modifications au sein d'un programme dont on ne dispose pas des sources du code est un exercice extrêmement difficile ;
3. L'utilisateur a la liberté de redistribuer des copies d'un logiciel libre, que ce soit gratuitement ou contre une somme d'argent ;
4. L'utilisateur a la liberté de distribuer des versions modifiées du programme, afin que la communauté puisse bénéficier de ses améliorations.

Ce cadre légal garantit le respect de la propriété intellectuelle de son ou ses auteurs, tout en veillant que nul ne puisse s'approprier la paternité d'un logiciel libre pour en faire un logiciel propriétaire

Logiciel libre ne signifie pas nécessairement gratuit. Certes, la liberté de diffuser le logiciel implique la liberté de le diffuser gratuitement et, de fait, tous les logiciels libres peuvent être récupérés et utilisés sans déboursier un centime. Il faut pourtant préciser qu'il est tout à fait possible pour une entreprise de vendre un tel logiciel de tout en restant en conformité avec la licence GPL. En ce sens, l'entreprise vend non pas le logiciel en tant que tel, mais surtout un service autour de ce logiciel.

Néanmoins il y'a des obligations imposées par la licence GPL :

1. Chaque logiciel distribué ou « vendu » doit être accompagné de son code source ;
2. Chaque logiciel libre doit être accompagné de la licence GNU originale ;

3. Les droits d'auteur de tout ou partie du logiciel sont préservés.

Nous reparlerons de cela dans les parties 6 page 16 et 7 page 19.

5 Le système d'exploitation libre GNU/Linux.

Présentons maintenant un élément majeur du logiciel libre : le système d'exploitation GNU/Linux. Avant d'aborder sommairement son histoire pour ensuite le mettre en perspective avec les autres offres, définissons rapidement ce qu'est un système d'exploitation.

5.1 Qu'est-ce qu'un système d'exploitation ?

Un système d'exploitation est composé de plusieurs éléments. Concentrons-nous sur ses deux composants principaux : le noyau et l'interface homme-machine.

Fondamentalement, un ordinateur est un ensemble d'éléments hétérogènes, les principaux étant :

- Un ou plusieurs processeurs, qui constituent la puissance de calcul de l'ordinateur ;
- De la mémoire, fixe (comme les disques durs) ou vive.

À quoi viennent s'ajouter de nombreux périphériques :

- Les écrans ;
- Le clavier ;
- Les imprimantes ;
- Etc.

Le noyau du système d'exploitation a pour objectif de gérer la communication entre tous ces éléments, faisant de l'ordinateur un tout cohérent.

Cependant, si l'ordinateur est désormais capable de communiquer avec ses différents constituants, le problème de la communication avec l'utilisateur reste entier. C'est là qu'intervient l'interface homme-machine, qui propose un moyen plus ou moins simple d'utiliser les ressources de la machine. Cette interface peut être en mode texte ou alors, comme sur tous les systèmes d'exploitations modernes, en mode graphique.

Le système d'exploitation le plus célèbre est Microsoft Windows, qui est un logiciel propriétaire.

5.2 Avant Linux.

Au début de l'informatique, les opérateurs étaient des techniciens connaissant parfaitement l'architecture de la machine. Le nombre d'états que pouvait prendre un ordinateur était suffisamment faible pour que les développeurs de programmes soient capables d'indiquer directement à l'ordinateur quoi faire. Au fur et à mesure des avancées techniques, les domaines de compétences faisant appel à l'ordinateur s'élargirent et il devint impossible de maîtriser entièrement les processus d'un ordinateur. C'est pour remédier à cela que furent développés les systèmes d'exploitations (ou autrement dit OS pour *Operating System*).

Dans les années 1970, la situation était toute autre que celle de nos jours et un mini ordinateur comme le PDP-11 fabriqué par Digital Equipment coûtait alors environ 65 000 dollars US. Le temps du micro-processeur devait donc être partagé et devait pouvoir exécuter plusieurs tâches.

En 1965, le MIT avait donc lancé le projet MULTICS, qui devait mettre au point un système d'exploitation répondant à cette demande. Cependant, ce projet n'aboutit pas. Le système n'était pas économiquement viable.

En 1969, Ken THOMPSON, chercheur au laboratoire AT&T Bell, désirait porter un jeu qu'il avait créé, « Space Travel, » sur un petit ordinateur PDP-7. Il n'y avait alors pas de système d'exploitation disponible pour cet ordinateur. Avec l'aide de Dennis RITCHIE, qui avait participé, tout comme Ken THOMPSON, au projet MULTICS, il écrivit alors un système d'exploitation nommé UNICS (qui deviendra UNIX). Ce système, qui héritait de certains principes de MULTICS, était d'ores et déjà multitâches et multiutilisateurs. Il reposait sur deux grands principes :

1. Tout est fichier. UNIX introduit le principe de répertoire, qui est finalement un fichier permettant de structurer le stockage d'autres fichiers. De même, les périphériques de l'ordinateur sont gérés comme des fichiers .
2. Chaque programme n'effectue qu'une seule tâche mais l'effectue au mieux. Le système inclut des mécanismes de communications entre programmes pour permettre d'accomplir des opérations complexes en combinant différents programmes.

La philosophie d'UNIX a été résumée ainsi par Doug Mac Ilroy, du même laboratoire que Ken Thomson et Dennis Ritchie :

« Rédigez des programmes qui remplissent correctement une tâche simple. Rédigez des programmes qui collaborent. Rédigez des programmes qui traitent des flots de textes car ce type d'interface est universel. »

Ken Thomson et Dennis Ritchie, membres de la communauté scientifique, diffusaient librement leur système d'exploitation — techniquement, il fallait la licence d'AT&T mais à l'époque ce n'était pas un problème. La situation évolua au début des années 1980, lorsqu'AT&T, réalisant le potentiel commercial d'UNIX, décida d'en fermer les sources. Cette situation poussa Richard Stallman à créer la Free Software Foundation et à lancer le projet GNU.

Son objectif était en premier lieu de créer un système d'exploitation libre, disponible à la fois sous forme binaire (compilée) et sous forme de code source. Puis de développer d'autres logiciels sur le même modèle, pour chaque tâche que l'on peut demander à un ordinateur. Richard Stallman pensait d'abord écrire un noyau. Cependant il a fallu préalablement développer d'autres éléments — un compilateur (GCC), une interface en mode texte (bash), entre autre. Le projet pour créer le noyau de ce futur système d'exploitation fut lancé sous le nom de HURD. Cependant, plutôt que de copier le fonctionnement d'UNIX, HURD se veut un système radicalement nouveau, tout en étant conforme au paradigme UNIX et compatible avec lui. Le développement du HURD est particulièrement long et, pendant longtemps, les outils GNU existaient sans qu'aucun noyau ne soit présent pour ce futur système d'exploitation. À l'heure où nous écrivons ces lignes, le projet HURD est fonctionnel mais pas encore suffisamment stable et complet pour être utilisé en production. Son développement se poursuit.

Avant de parler plus précisément de Linux, notons que le Professeur Andrew S. TANNENBAUM a créé, en 1987, un mini noyau de système d'exploitation, compatible avec UNIX, baptisé Minix. Minix est une superbe création mais son auteur le maintient délibérément à un niveau minimum, afin qu'un étudiant puisse entièrement l'appréhender en un semestre de cours. De plus, sa licence ne permet pas d'en diffuser un code source modifié. Pour toutes ces raisons, Minix n'est pas exploitable en tant que noyau pour un système en production.

5.3 Brève histoire de Linux.

Le 25 août 1991, le message suivant fut envoyé sur le newsgroup dédié à Minix :

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -
I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).
I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)
Linus (torvalds@kruuna.helsinki.fi)
PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

Que l'on peut traduire ainsi :

Sujet : qu'aimeriez-vous le plus voir dans Minix ?

Résumé : petit sondage pour mon nouveau système d'exploitation.

Salut à tous les utilisateurs de Minix.

Je réalise un système d'exploitation (libre) pour les clones AT à base de processeurs 386 et 486 (un simple loisir, il n'aura ni la dimension ni le côté professionnel de GNU). L'idée mijote depuis avril et commence à prendre forme. J'aimerais savoir ce qui vous a plu ou déplu dans Minix car mon système d'exploitation s'en inspire (même organisation physique du système de fichiers, pour des raisons pratiques, entre autres).

J'y ai déjà porté bash (1.08) et gcc (1.40) et tout semble fonctionner. Je devrais donc disposer de quelque chose d'exploitable dans les mois à venir et j'aimerais donc connaître les fonctionnalités susceptibles d'intéresser le plus grand nombre. Toutes les suggestions sont les bienvenues mais je ne promets pas de toutes les implanter. :-)

Linus

P.S. : oui, il est indépendant de tout code provenant de Minix et dispose d'un système de fichiers multi-thread. Il n'est PAS portable (utilisation de l'ordonnanceur 386, etc.) et ne gèrera probablement jamais autre chose que les disques durs AT car ce sont les seuls dont je dispose. :-)

Linus Benedict TORVALDS annonçait qu'il était en train de mettre au point un noyau de système d'exploitation. Contrairement au cas de HURD, il s'était attaché à reproduire un système analogue à UNIX plutôt que de créer un système novateur. Les qualités de programmeur

de Linus Benedict TORVALDS lui permirent de développer, relativement rapidement, un système parfaitement exploitable, ce d'autant plus qu'une armée de développeurs enthousiaste vint le rejoindre. Ajoutons le fait qu'il plaça son noyau sous licence GPL. Pour GNU, c'était une aubaine car ce noyau permettait de faire de GNU un système parfaitement utilisable. Il s'agit du système GNU basé sur un noyau Linux, en abrégé GNU/Linux.

Lorsque l'on dit que le noyau Linux fut développé relativement rapidement, il convient de tempérer : ce fut plutôt rapide pour un projet d'une telle ampleur mais prit tout de même à peu près quatre ans car la publication des premiers codes sources de Linux (la version 0.01) eut lieu en septembre 1991, tandis que la première version considérée comme suffisamment stable pour être exploitable en production (la version 1.0) n'est sortie qu'en mars 1994.

Le succès de Linux dépassa son auteur et Linus Benedict Torvalds se retrouva catapulté au coeur d'un projet beaucoup plus ambitieux qu'il ne l'avait envisagé au départ. A l'heure actuelle, GNU/Linux a été porté sur bien d'autres architectures que les compatibles IBM-PC, puisqu'il fonctionne, entre autre, sur PowerPC ou les stations de travail SPARC pour n'en citer que quelques unes. Linux est, de plus, devenu part intégrante du projet GNU.

Les diverses qualités de GNU/Linux l'ont rendu de plus en plus populaire. À ce sujet, une anecdote nous semble révélatrice.

En 1997, James Cameron sort dans les salles de cinéma le film Titanic, qui sera en 1998 couronné par plusieurs Oscars. Quel rapport avec GNU/Linux ?

Devant l'ampleur et le coût du projet de Titanic, les financiers retirèrent purement et simplement leurs investissements, contraignant James Cameron à investir sa propre fortune afin de terminer un film auquel il tenait.

Il est évident que l'économie fut de rigueur à tous les niveaux de la production. Comment mettre au point les effets spéciaux en images de synthèses, les maquettes 3D du bateau ?

Après avoir envisagé plusieurs possibilités propriétaires, il se tourna finalement vers GNU/Linux. Toutes les images virtuelles furent rendues sous une « ferme de calcul, » composée d'une centaine d'ordinateurs d'occasion sur lesquels GNU/Linux était installé. La production du film avoua être étonnée par la grande stabilité du système.

Si on connaît le succès final du film, on sait moins que cette annonce fit l'effet d'une bombe. Les diffuseurs de solutions propriétaires, qui parlaient de GNU/Linux comme d'un système de bricoleur ou d'un assemblage de bouts de ficelles, furent dépassés. Le grand public, lui, venait de découvrir une alternative plus que viable aux solutions propriétaires, une solution efficace, stable, de haute technologie et, surtout, accessible quelque soit l'état des bourses.

Il est d'ailleurs curieux de penser que le malheureux naufrage du Titanic, provoqué par l'orgueil technologique de l'époque, qui imposait une ségrégation drastique au niveau social, présida à l'émergence d'une technologie formidable issue de personnes travaillant humblement dans l'ombre et disponible à tout un chacun.

Depuis, le nombre d'utilisateurs de GNU/Linux a été croissant, dépassant les frontières du milieu scientifique proprement dit, pour atteindre dans un premier temps les entreprises, notamment pour tout ce qui concerne les architectures réseau. De plus, l'effort considérable fourni par les développeurs au niveau de la convivialité et des interfaces graphiques a permis la percée de Linux dans l'informatique familiale.

Parallèlement, les utilisateurs ne se rendirent pas compte qu'ils utilisaient GNU et crurent simplement utiliser Linux. Pour ceux qui se sont engagés dans le projet GNU, il était devenu difficile d'attirer l'attention des utilisateurs de GNU/Linux sur les problématiques éthiques

soulevées par les logiciels libres. On peut se demander s'il ne s'agit pas là d'une victoire de GNU/Linux car il semble finalement que son éthique aille maintenant de soi pour les utilisateurs. Il s'agirait d'une sorte d'acceptation tacite.

5.4 Linux par rapport aux autres systèmes d'exploitations.

Comme déjà dit, de nombreuses qualités intrinsèques assure la pérennité de GNU/Linux :

1. GNU/Linux répond au paradigme UNIX. UNIX représente le standard dans le milieu de la recherche et pour les gros systèmes. GNU/Linux propose de faire fonctionner un système d'exploitation proche de ce système sur les stations de travail
2. Linus Benedict TORVALDS est très regardant avant d'ajouter une nouvelle fonctionnalité au noyau, ce qui rend le système très stable.
3. GNU/Linux est un système orienté réseau, qui hérite de la tradition d'UNIX.

Faut-il dire que GNU/Linux est aujourd'hui un acteur majeur du monde informatique et que son usage tend à se généraliser. Citons entre autre le cas d'IBM, qui a annoncé, il y a deux ans, que désormais toutes ses machines supporteraient ce système d'exploitation. IBM est allé plus loin en indiquant que c'était d'ailleurs le seul système pouvant convenir à toute sa gamme, depuis le PC personnel jusqu'au plus puissant ordinateur (la « ferme de calcul » de la compagnie Shell comprenant 1024 serveurs IBM sous GNU/Linux). On peut aussi citer la chaîne de magasin Auchan en France, qui a entièrement migré sous GNU/Linux.

En ce qui concerne l'éthique du logiciel libre, que dire d'une entreprise comme Matra, active aussi bien dans l'ingénierie que de dans la construction de voitures ou des fusées Ariane ? Matra a mis les sources de son logiciel phare Cascade à disposition de la communauté du libre sous le nom d'Open-Cascade. De leurs propres dires, les possibilités et qualités de leur logiciel ont fait un bond en avant sans commune mesure.

GNU/Linux a beaucoup évolué et évolue encore beaucoup. De nos jours, il faut reconnaître que GNU/Linux, avec les interfaces graphiques telles que KDE ou GNOME, est tout aussi simple d'utilisation que Microsoft Windows XP ou Mac OS X. On peut noter que Mac OS X est un système d'exploitation basé sur UNIX — plus précisément la version BSD, développée à l'université de Berkley.

Du point de vue de la gestion de la machine, GNU/Linux nécessite une compétence d'administration de système UNIX. Cette compétence semble souvent plus difficile à acquérir qu'une compétence d'administration sous Microsoft Windows. Ceci est probablement dû au fait qu'UNIX et les systèmes en dérivant étaient jusqu'ici moins connus — au moins du grand public, — ainsi qu'à la profusion d'options de ce système d'exploitation.

Cela est dû également au fait qu'à une époque GNU/Linux était assez délicat à installer et à configurer. Cependant, d'énormes progrès ont été faits à ce niveau et une distribution telle que Mandrake Linux peut être installée par un novice complet en une demi-heure, sachant que le système se configure automatiquement, qu'il est immédiatement exploitable et permet, de manière automatisée, de faire cohabiter sur une même machine Microsoft Windows et GNU/Linux. Si, jusque vers l'an 2000, GNU/Linux était encore difficile d'accès pour l'utilisateur familial, il est aujourd'hui éminemment adapté à l'utilisateur dit « lambda », de par sa stabilité et sa robustesse, son utilisation courante étant un modèle de simplicité.

Il est vrai néanmoins que les fabricants ne développent pas toujours de pilote pour GNU/Linux et donc qu'il arrive que certains périphériques ne soient pas reconnus. Cependant, assez réguliè-

rement, des développeurs indépendants prennent en charge l'écriture d'un pilote de périphérique. De fait, les cas de matériels non supportés sont de plus en plus rares. Il reste, cependant, important, avant d'installer GNU/Linux, de s'assurer que son matériel est bien supporté, même si c'est généralement le cas.

Le milieu des administrateurs réseaux ou de parcs informatiques a remarqué qu'un système fonctionnant sous GNU/Linux est généralement très rationnel, structuré, plus stable à long terme et plus sûr que certains systèmes grand public. Car si, à « très court terme », certains de ces systèmes semblent représenter un gain de temps et de ressources humaines, ils se révèlent, sur le moyen et long terme, totalement distancés.

Ce qui est surtout frappant dans le système GNU c'est sa versatilité car il peut aussi bien combler l'administrateur réseau professionnel et exigeant que l'utilisateur domestique, sans oublier bien sûr toutes les nuances entre ces deux extrêmes : les utilisateurs professionnels dans le domaine du graphisme, de la programmation, du multimédia, etc.

5.5 Les diverses distributions de Linux.

Si le code source de Linux et des outils GNU est librement disponible, construire un système d'exploitation cohérent à partir de ces éléments épars est une tâche assez délicate, ne serait-ce que parce qu'il faut connaître les divers éléments nécessaires à la mise en place d'un système d'exploitation parfaitement utilisable, sans parler des logiciels que l'on peut y utiliser.

Pour permettre à tout un chacun d'utiliser GNU/Linux, des distributions ont été conçues, c'est-à-dire un ensemble logiciel cohérent permettant d'installer plus ou moins facilement le système sur un ordinateur ou un parc d'ordinateur. Passons donc en revue les distributions les plus courantes :

- AbulEdu, une distribution développée à l'origine pour l'éducation nationale française ;
- Debian Linux, une distribution entièrement libre, particulièrement stable et donc très apprécié des administrateurs système ;
- Gentoo, une distribution particulièrement paramétrable, très à jour, permettant de très grandes optimisations du système et de plus totalement libre, quoique assez difficile et longue à installer car proposant de compiler toutes les sources des programmes que l'on installe spécialement pour sa machine ;
- Knoppix, une distribution vive, c'est à dire totalement utilisable sans n'avoir rien à installer sur son disque dur car accessible sous la forme d'un CD-ROM (ou une clef USB) avec lequel la machine peut s'initialiser. Elle est de plus totalement libre ;
- Linux From Scratch (LFS), ce qui signifie Linux à partir de rien. Peut-on vraiment parler de distribution ? En fait, LFS est plus un regroupement d'utilisateurs qui construisent leurs propres systèmes d'exploitations, permettant d'être tenu au courant des évolutions et donnant toutes les informations pour installer sa propre version de GNU/Linux. Cette distribution est bien sûr entièrement libre ;
- Mandrake Linux, une distribution particulièrement conviviale ;
- Red Hat Linux, leader du marché des UNIX, il s'agit d'une distribution orientée serveur ;
- Slackware Linux, distribution historique, austère mais robuste et particulièrement paramétrable ;
- SuSE Linux, distribution d'origine allemande.

Ceci pour les distributions majeures. L'extrême souplesse du noyau, la publication de son code source, la possibilité pour chaque utilisateur de le recompiler avec des options très ciblées,

l'offre logicielle gigantesque, permettent aussi la création de distributions particulières, affectées à des tâches bien précises. De plus, pour la plupart d'entre elles, elles peuvent s'exécuter directement depuis un CD amorçable sans absolument nécessiter une installation sur le disque dur :

- eMovix, GeexBox, distributions destinées exclusivement à la diffusion multimédia vidéo et audio ;
- Dyne : Bolic, distribution orientée sur la création multimédia video et audio, accompagnée d'un noyau optimisé en ce qui concerne les temps de latence et l'accélération 3D ;
- Free-OS, distribution orientée serveur, comprenant un environnement d'administration en ligne particulièrement réussi et une installation et mise en oeuvre extrêmement simples ;
- FREEDUC, une distribution vive entièrement libre pourvue de nombreux logiciels à but éducatif, développée au lycée Jean BART de Dunkerque.

5.6 D'autres systèmes d'exploitations libres.

Le logiciel libre ne s'arrête pas au monde GNU. Il existe d'autres systèmes d'exploitations libres, quoique n'étant pas sous licence GPL mais sous d'autres licences semblables. Citons donc Free BSD, une version libre de l'UNIX développée à l'université de Berkley, Net BSD, QNX ou BeOS.

6 Pourquoi s'orienter vers les logiciels libres ?

Nous parlerons d'abord des nouveautés dans la façon de concevoir un logiciel, avant d'aborder les intérêts que peuvent retirer les entreprises comme les particuliers des logiciels libres.

6.1 Généralités.

Les points soulevés ici concernent tous les acteurs du monde informatique, que ce soit dans le domaine économique, celui du génie logiciel ou de la recherche.

Jusqu'à il n'y a pas si longtemps, l'informatique était un domaine plus ou moins cantonné entre les mains de spécialistes de type scientifique ou de grosses multinationales. Dans certains domaines très ciblés, cela restera le cas. Cependant, l'utilisation de l'outil informatique s'est désormais répandue dans tous les corps de métiers, ainsi que dans de nombreux domaines considérés comme du loisir. L'informatique n'est plus, dans la majorité de ses propres domaines d'applications, une technique destinée aux seuls spécialistes. Au contraire, l'informatique est devenue un métier à part entière : qu'il s'agisse de la programmation ou l'administration de systèmes. Elle propose un ensemble de techniques s'appliquant sur des matériaux (les logiciels et le matériel physique constituant l'ordinateur).

Somme toute, l'informatique fournit un outil adapté à certaines tâches. La complexité de l'offre logicielle n'est qu'apparente dans le monde concret. Un individu, selon sa fonction, n'aura besoin que d'un traitement de texte, un autre d'un logiciel d'images de synthèses. Il est donc illusoire pour un ou quelques-uns des acteurs de cette industrie de concentrer la totalité du domaine d'activité informatique mondial entre leurs mains. C'est tout simplement impossible car la somme des secteurs s'appuyant sur des solutions informatiques est, comme nous l'avons vu, bien trop vaste pour qu'une entreprise, aussi importante soit-elle, puisse proposer une offre pertinente pour chacun d'entre eux.

L'aventure du logiciel libre a donné une nouvelle dynamique à l'informatique : le travail coopératif. On peut le définir ainsi : le travail d'un seul pour le bien de tous, le travail de tous pour le bien d'un seul.

Alors que tous les développements étaient conçus anciennement de manière pyramidale, avec les déboires que l'on connaît (passage de l'an 2000 entre autres), le développement coopératif a permis au génie logiciel d'arriver de manière fulgurante aux buts qui lui étaient assignés.

Du point de vue économique, cela ne demande que peu de moyens pour d'étonnants résultats, chacun apportant soit une base soit des améliorations, en fonction de ses compétences, pour un même projet. Il s'agit donc d'une entraide mutuelle où chacun apporte des éléments que tous pourront utiliser, une collaboration permettant de construire un édifice qu'il aurait été bien difficile à mettre en place en concentrant les développements.

Cela ne s'est pas fait sans mal, les principales critiques portant sur l'apparente anarchie des développements, qui, il faut bien le dire, est un des pièges du développement coopératif. Mais la vitalité de la communauté du logiciel libre a permis, en tenant compte de manière objective de ces critiques, de réagir très rapidement en se donnant les moyens d'accomplir sa tâche : fortement structurer de manière immédiate ce possible désordre. Tous ceux qui travaillent en entreprise connaissent les problèmes de synchronisation intrinsèques à chaque projet mettant en œuvre plusieurs acteurs. Dans le cas qui nous occupe, il fallait synchroniser le projet non pas dans un microcosme particulier mais bien au niveau d'acteurs disséminés dans le monde entier. Des outils ont alors été créés et offerts à la communauté mondiale, tels que CVS.

Ceci eut été impossible dans le cadre d'une entreprise de moyenne importance au vu des ressources que cela requiert. Le projet GNU l'a fait et l'a offert à tout un chacun. Des projets comme XFree86, Apache, Linux, Samba, GCC, difficile de tous les citer, ont atteint un perfectionnement et une stabilité qui forcent l'admiration. Beaucoup l'on compris comme IBM, Apple ou Sun Microsystem, qui sont des géants de l'industrie informatique.

Il n'est pas difficile d'imaginer les bénéfices qu'une PME peut en tirer, puisqu'elle bénéficiera de la même opportunité en soumettant son projet à la communauté du logiciel libre.

Un des auteurs a travaillé dans une startup (4 personnes dont un développeur matériel). Grâce au développement coopératif, un organisateur de poche de nouvelle génération a pu être fonctionnel en moins de 8 mois, matériel, système d'exploitation et interface graphique compris alors que c'était un projet énorme, révolutionnaire et jusqu'alors jamais développé. Un tel projet n'aurait pas pu voir le jour sans la communauté du libre.

6.2 Aspects pragmatiques.

En observant les quatre libertés du logiciel libre, on constate immédiatement que l'on s'affranchit des restrictions des licences propriétaires, qui limitent souvent le cadre d'utilisation d'un logiciel et qu'il faut renouveler régulièrement. On comprend sans peine qu'il y ait une diminution des coûts à moyen terme — à court terme il y a généralement une diminution globale également mais nous y reviendrons dans la partie 8, page 22. Ceci dit, il faut voir aussi que le logiciel libre propose une conception radicalement différente du rapport au logiciel.

En effet, lorsque l'on utilise un logiciel libre, ce logiciel devient le sien, logiciel que l'on peut faire évoluer comme on l'entend. Plus encore, on devient membre de la communauté d'utilisateurs de ce logiciel, ce qui nous semble essentiel. Si on fait parfois remarquer que le support d'un logiciel libre peut être hasardeux car dépendant de son développeur qui prend sur son temps libre, c'est oublier d'une part que d'autres développeurs viennent généralement

rejoindre l'initiateur d'un projet — il est même courant qu'un logiciel libre soit maintenu par une équipe ne comportant plus aucun des développeurs initiaux — et de plus la communauté des utilisateurs de ce logiciel représente un support souvent très réactif car il n'est pas rare que l'on ait la réponse à son problème dans l'heure. Devenant membre de la communauté, on est soi-même invité à aider les autres utilisateurs, ce qui se fait souvent naturellement.

La communauté nous invite à envoyer un écho si jamais on trouve une erreur dans un logiciel. C'est d'ailleurs avantageux pour soi-même, proche du support, car on voit ainsi plus souvent corrigés les bogues que l'on rencontre dans l'utilisation d'un logiciel que dans le cas d'un logiciel propriétaire.

Ceci dit, il est clair qu'il n'est absolument pas nécessaire d'être informaticien ou un utilisateur avancé pour utiliser un logiciel libre, ni non plus pour aider la communauté, d'ailleurs. Nous voyons donc que le logiciel libre permet une responsabilisation de l'utilisateur, qui passe de consommateur à acteur mais ce sans contraintes absolues ni obligations : l'attitude de pur consommateur restant toujours possible et légale.

Certains font parfois remarquer que le développement d'un logiciel libre peu s'arrêter. Certes mais une entreprise peut-être rachetée ou faire faillite et donc abandonner le suivi des logiciels qu'elle développe. Cela n'est pas un arguments valable. Lorsque le développement d'un logiciel libre s'arrête, le code source reste disponible et donc reprendre ou continuer le développement de ce logiciel reste possible.

On peut parfaitement illustrer cela avec un cas extrême comme celui du logiciel d'images de synthèse Blender.

Développé en entreprise et pour ses besoins propres, Blender a toujours été offert gratuitement à la communauté par son concepteur. La qualité de ce logiciel, qui n'est pas loin de concurrencer les poids lourds de l'infographie l'a fait adopter par bon nombre d'utilisateurs, quand bien même il n'était pas libre *stricto sensu*, son code source n'étant pas disponible.

Ton ROSENDAAL, unique concepteur et programmeur, quelque'un d'une générosité extraordinaire, a voulu proposer un mode spécifique, afin de rentabiliser son travail après la dissolution de l'entreprise de graphisme dans laquelle il travaillait. Des investisseurs se sont très rapidement présentés. Blender était toujours distribué gratuitement mais avec un retard d'une ou deux versions, alors que la version la plus récente était vendue.

Il s'agissait de toute évidence d'un moyen terme. Tout le monde y trouvait son compte. Cependant, les difficultés financières se sont rapidement accumulées et l'entreprise de Ton Rosendaal fut mise en faillite. Le code source appartenant aux investisseurs, la communauté du libre était catastrophée mais est immédiatement intervenue.

Une souscription pour le rachat du code de Blender a été lancée sous la condition que le code, une fois racheté, passerait sous licence GNU/GPL. 100 000 euros étaient requis et c'est finalement 130 000 euros qui furent versés en moins de deux mois, chacun y allant selon ses moyens.

Blender est aujourd'hui sous licence GNU/GPL et sa vitalité nous vaut une mise à jour et de nouvelles fonctionnalités tous les deux mois environ. Quant à la qualité du logiciel, elle est exceptionnelle.

Le « logiciel libre » a souvent été, par le passé, associé à « petit logiciel rébarbatif et avec peu de fonctionnalités ». Citons par exemple la comparaison entre Gimp (logiciel de retouche photo) et Paint, livré en standard avec Windows, les mettant sur un pied d'égalité. A l'époque où cette comparaison fut faite, Gimp était déjà pleinement fonctionnel et soutenait la comparaison avec

certain poids lourd de l'infographie. Quoiqu'il en soit, cette image est totalement fautive : dans le cas de Gimp, cité plus haut, il suffit de rappeler qu'il est utilisé à outrance par l'industrie du cinéma dans la retouche des images en post-production et ce avec une version de base datant de 5 ans. Cette version, appelée aujourd'hui CinePaint, poursuit son évolution parallèlement à la version Gimp de retouche photo et est continuellement adaptée aux besoins spécifiques de l'industrie du cinéma.

Les logiciels libres sont aussi conviviaux que les logiciels propriétaires. De plus, ces logiciels sont développés par des passionnés, souvent très au fait des progrès techniques et théoriques. Dire que ces logiciels sont d'une grande qualité tient du pléonasme. On peut ajouter que l'interaction constante avec les utilisateurs leur permet de rester au plus proche de leurs demandes spécifiques.

Dans le cadre d'organismes de formations, un argument de plus vient s'ajouter : ceux qui suivent les différentes formations auront tendance à vouloir mettre en pratique leurs compétences fraîchement acquises. Pour ce faire, ils chercheront à se procurer les logiciels utilisés au cours de la formation. Les licences des logiciels propriétaires sont souvent très onéreuses — pensons par exemple à l'étudiant qui voudrait simplement acheter Microsoft Windows et Microsoft Office, ce qui lui reviendrait à plus du tiers du prix d'une machine de base. Pour peu que la formation soit assez complète — c'est-à-dire présente un large éventail de logiciels — ou bien présentant des logiciels un peu pointus — donc souvent très chers, — autant dire que l'on pousse les étudiants au piratage. A l'inverse, on a la liberté de diffuser un logiciel libre. De fait, il est aisé, pour l'organisme formateur, de fournir une copie des logiciels étudiés au cours de la formation de manière parfaitement légale. Une objection est parfois soulevée : il peut être difficile de vendre la compétence acquise lorsqu'il y a écrit sur un CV « OpenOffice.org », par exemple, en lieu et place de « Microsoft Office ».

Faisons remarquer qu'une formation bien faite enseigne des méthodes de travail et des compétences afin d'être à même de s'adapter à un vaste panel de situations et non pas sur quelle icône cliquer dans tel logiciel spécifique — dans notre exemple, la formation doit donner une compétence en bureautique plutôt que d'enseigner « Microsoft Office » ou « OpenOffice.org ». Enfin, le but de toute bonne formation doit, *a priori*, être de présenter différentes approches pour résoudre un travail donné, elle ne devrait donc négliger aucun des outils disponibles.

7 Les principaux logiciels libres.

Lorsque l'on parle de logiciels libres, on aurait tort de croire qu'ils se limitent à GNU/Linux ou UNIX : il n'en est rien. Les portages sont légions sous d'autres systèmes, comme Microsoft Windows et Mac OS et ce jusque dans les interfaces graphiques. Ceci principalement parce que les codes sources sont aisément disponibles, ce qui permet aux développeurs de porter les logiciels sur leurs plateformes de prédilection. De plus, certains outils, tel l'environnement Cygnus, permettent de faire fonctionner et de compiler bon nombre d'applications GNU/Linux ou UNIX afin qu'ils soient utilisables sous Microsoft Windows.

Au cours de sa pratique professionnelle dans une entreprise travaillant presque exclusivement sous Microsoft Windows, Pierre LO CICERO n'a jamais compté le nombre de fois où l'environnement de Cygnus, faisant fonctionner des programmes comme bash, find, grep et autres, lui permirent de gérer des problèmes que MS-DOS n'était pas à même de gérer de manière simple.

Le nombre de logiciels libres est extrêmement vaste, et c'est peu de le dire. Ce qui suit n'est qu'un très bref aperçu. Pour une liste exhaustive de l'offre logicielle GNU/Linux en correspon-

dance avec les logiciels fonctionnant sous Microsoft Windows, se référer à [9].

7.1 Solutions bureautiques et édition.

S'il ne faut en citer qu'une, il s'agit bien entendu d'OpenOffice.org. Cette suite bureautique est aujourd'hui reconnue comme une alternative sérieuse à Microsoft Office, tant pour un particulier que pour un professionnel. OpenOffice.org fonctionne sous Microsoft Windows, Mac OS X, GNU/Linux, FreeBSD et Solaris.

Scribus est un logiciel de mise en page en pleine expansion.

7.2 Editeur de textes.

Emacs et XEmacs (une version du premier utilisant des icônes) sont deux éditeurs de textes aux fonctionnalités impressionnantes qui tiennent presque lieu d'un système d'exploitation complet à lui tout seul. Certains utilisateurs font absolument tout avec Emacs : édition de texte, client courriel, navigation web, écriture de code C/C++, Java, Fortran, HTML, VRML, etc., avec analyse syntaxique, compilation, débogage... Emacs existe sous tout système de type UNIX (ce qui inclu bien sûr GNU/Linux) ainsi que sous Microsoft Windows et Mac OS X.

Citons aussi BEdit sous Mac OS et Jext, entièrement écrit en Java et donc indépendant de la plateforme.

7.3 Utilitaires pour Internet.

- Mozilla est un navigateur respectant parfaitement les standard du W3C (organisme gérant les standards d'Internet) ainsi qu'un client courriel de grande qualité, auquel on peut adjoindre des dictionnaires. Il existe sous Microsoft Windows, GNU/Linux et Mac OS ;
- Opera est un navigateur Internet ;
- Pine est un client courriel particulièrement léger, fonctionnant sur tout système UNIX, dont GNU/Linux ;
- Kmail est un client courriel fonctionnant sous GNU/Linux uniquement ;
- Evolution, sans doute le client courriel le plus proche de Microsoft Outlook dans sa version professionnelle, fonctionnant sous GNU/Linux uniquement ;
- Konqueror est un navigateur web fonctionnant sous GNU/Linux uniquement.

7.4 Outils de développement.

Le compilateur gcc est devenu un standard dans l'industrie. De fait, tout les développeurs sous UNIX l'utilisent. C'est une collection de compilateurs permettant de compiler le C, C++, Objective C, Ada, Fortran et Java. Il propose de nombreuses options d'optimisations automatiques et de débogage. C'est véritablement un excellent compilateur, qui de plus est régulièrement mis à jour. Il fonctionne sous tout système de type UNIX (donc y compris GNU/Linux). Un équivalent de gcc, entièrement libre, existe pour Microsoft Windows : cygwin.

Le débogueur gdb et ses frontaux graphiques ddd et kgdb permettent de faciliter le débogage dans toutes sortes de langages.

Signalons deux bibliothèques en C pour le développement d'interface graphique : GTK, dont l'outil GLADE permet de réaliser une interface de manière graphique et QT, dont l'outil

QT-Develloper est l'équivalent de GLADE. Ces bibliothèques sont compatibles GNU/Linux et Windows.

Sous GNU/Linux, KDevelop propose un environnement de développement proche des environnements de la société Borland et s'appuyant sur gcc, Qt, QT-Develloper, gdb et kgdb (ainsi que l'éditeur de texte Kate).

Anjuta, environnement de développement proche de Visual C++.

Quanta Plus (Qt), Screem (GTK), environnements de développement HTML, PHP, Javascript.

Signalons aussi wxWindows, qui est une bibliothèque permettant de s'affranchir de la barrière de la programmation de l'environnement graphique entre Linux, Windows et Mac OS.

7.5 Graphisme.

7.5.1 Bitmap

The GIMP est un outil de retouche d'image et de création bitmap qui s'approche de Photoshop. Fonctionne sous GNU/Linux, Windows et Mac OS.

7.5.2 Vectoriel

Sketch, Sodipodi, OpenOffice Draw, Dia pour le dessin vectoriel.

7.5.3 Image de synthèse.

- Blender environnement de modélisation et de rendu, similaire à 3D Studio, possédant de plus un puissant système d'édition vidéo, ainsi qu'un langage de script basé sur python ;
- KPovModeler, environnement de modélisation destiné à être rendu sous le moteur POV-Ray ;
- White Dune, modeleur orienté VRML ;
- ArtOfIllusion, modeleur qui a la particularité d'être entièrement écrit en Java.

7.6 Vidéo.

- Kino est un logiciel de montage vidéo fonctionnant sous GNU/Linux ;
- Avidemux, le couteau suisse de la vidéo ;
- Cinelera, environnement de montage vidéo non linéaire ;
- Blender et son excellent module de montage vidéo ;
- Jahshaka un environnement OpenGL pour les effets spéciaux et la post-production.

7.7 Son et musique.

On peut citer de nombreux logiciels de traitement sonore fonctionnant sous GNU/Linux :

7.7.1 Traitement du son

- Ecasound, retouche de son professionnelle ;
- ReZound (un clone de coolEdit ou SoundForge) ;

- Ardour (très avancé et prometteur mais encore en développement) ;
- Audacity, fonctionnant sous GNU/Linux, Windows et Mac OS.

7.7.2 Séquenceur MIDI

- Jazz++, séquenceur MIDI similaire à Cubase ;
- Brahms, Rosegarden, MuSe.

7.7.3 Synthèse sonore

amSynth, ALSA Modular Synth, Bristol, Csound, FluidSynth, jMax, RTSynth, RX/Saturno, SpiralSynth Modular, Ultramaster Juno6, VSTserver ou ZynAddSubFX

7.8 Administration et réseau.

GNU/Linux, héritant de la tradition d'UNIX, est un système d'exploitation orienté vers le réseau. De fait, l'offre en ce qui concerne les logiciels, serveurs et clients d'administration réseau ainsi que tous les outils de surveillance est particulièrement importante. Référez vous à l'URL de référence [9].

7.9 GNU/GPL sous Microsoft Windows

Il faut faire mention du portage des logiciels libres sous un système comme Microsoft Windows, ainsi que des développements originaux sous ce système d'exploitation. Gimp, Jahshaka, Blender, Audacity fonctionnent parfaitement sous Microsoft Windows ainsi qu'une grande partie des utilitaires en ligne de commande, de même pour le gestionnaire de fenêtres XFree86 et GCC par l'intermédiaire de Cygwin. Les environnements graphiques Gnome et KDE. Le couteau suisse de la vidéo Avidemux a aussi son pendant sous windows avec VirtualDub ainsi que les codecs Ogg (un codec similaire à MP3, libre car basé sur un autre algorithme, dont la qualité dépasse celle du MP3 à poids égal de fichier), XVID (implémentation libre du MPEG4). Cette liste n'est pas exhaustive mais, si l'on prend la peine de rechercher cette catégorie de logiciels pour Microsoft Windows et développés sous licence GPL, le nombre de solutions est surprenant.

Typiquement, sous Microsoft Windows, l'utilisateur pourra, à profit, remplacer la combinaison Microsoft Office + Microsoft Outlook + Internet Explorer par OpenOffice.org + Mozilla.

8 La délicate période de transition.

Il est important de parler de la période de transition. Force est de reconnaître que c'est une période délicate : d'abord parce qu'elle requiert un changement d'habitude, ensuite à cause des formats propriétaires dont nous avons parlé. Il faut s'assurer de la possibilité de relire les fichiers déjà produits, par exemple avec OpenOffice.org pour ce qui est de lire les formats de la suite Microsoft Office. Remarquons que le format de fichier exécutable est également un format de fichier et que donc, à ce titre, il en existe également des propriétaires — tels que celui de MS-DOS — et des libres comme ceux de GNU/Linux (a.out et ELF). Notons d'abord que Microsoft

a d'ores et déjà annoncé que la prochaine version de Windows (Long Horn) ne supportera plus le format exécutable de MS-DOS.

Si l'on conserve le même système d'exploitation, on peut généralement utiliser ses applications spécifiques et opter en même temps pour des logiciels libres.

Il est certain que les applications spécifiques développées en interne peuvent poser un problème au cours d'une migration pour un nouveau système d'exploitation ou pour un logiciel utilisant un langage de macro-commandes différent de celui utilisé auparavant. Il faut souvent envisager la réécriture de telles applications, ce qui peut augmenter le coût à court terme. Toutefois, les entreprises possèdent généralement le code source des applications spécifiques qu'elle développe. Ceci peut de réécrire des applications développées dans un langage devenu obsolète vers un langage plus récent, améliorant ainsi la pérennité du logiciel car il ne faut pas se leurrer : il faut de toute façon rerédiger de temps en temps ses applications. De fait, le faible coût des logiciels libres peut être en partie contrebalancé par les coûts de développements et de formation. Cependant, il faut être bien conscient que le vrai prix d'un logiciel se mesure sur la durée.

L'IDA (*Interchange of Data between Administrations*), programme de la Commission Européenne chargé d'améliorer l'interopérabilité des réseaux entre les administrations les plus sensibles, a récemment conclu (voir [14]) que, en se basant sur une période de cinq ans et en prenant en compte les mises à jours, l'adoption de solutions en logiciels libres revient jusqu'à deux fois moins chère que des applications propriétaires fermées. Ceci, bien sûr, dans le cas d'une entreprise, puisque l'utilisateur individuel voit, lui, pour ainsi dire disparaître tout coût logiciel.

Une méthode de migration classique est d'agir en douceur : d'abord conserver le même système d'exploitation mais utiliser par ailleurs des logiciels libres — OpenOffice.org et Mozilla en tête — et, au besoin, ainsi préparer la migration vers un nouveau système d'exploitation. Il est d'ailleurs possible de faire non seulement cohabiter des machines ayant des systèmes d'exploitations différents dans un réseau mais également de faire cohabiter plusieurs systèmes d'exploitations sur une même machine. Il est parfaitement possible d'utiliser des logiciels libres sans changer de système d'exploitation.

Le principal problème d'une migration est, en quelque sorte, un problème psychologique : les utilisateurs ont parfois l'impression d'être, dans un sens, punis, de recevoir une charge de travail supplémentaire qui consiste à apprendre à connaître de nouveaux outils. Ils ont souvent peur de voir leurs compétences dévaluées. Les deux maîtres mots, au court de la période de transition, doivent donc être : communication et formation.

9 Exemple d'une migration de logiciels propriétaires vers logiciels libres.

Nous nous contenterons dans cet article de conseils pour une migration dans le cadre d'une machine personnelle. Les conseils pour une migration industrielle découlent directement de la première et sont trop complexes et divers pour être abordés dans le cadre de ce document. Cependant, tous les outils adaptés à l'industrie, comme, par exemple, des serveurs performants, sont présents sur les 3 CD-ROM de la distribution de GNU/Linux utilisée. Il en est ainsi pour la plupart des distributions.

9.1 Définition des besoins

Si l'on envisage le passage de Microsoft Windows à GNU/Linux, la première chose à faire est de définir les besoins : qu'est-ce que je veux faire de mon ordinateur ? Ensuite, il convient de hiérarchiser ces besoins. Est-ce une station de travail destinée au travail de bureau, au développement de logiciels, aux jeux, à la lecture multimédia ou à la création multimédia ? Ou tout cela en même temps ? Quels sont les logiciels dont je ne peux me passer ? Ont-ils leurs correspondances sous GNU/Linux ? Ce sont les premières questions que l'on doit se poser.

9.2 Évaluation du matériel

Dès lors, il faut se pencher sur le matériel dont on dispose : scanner, carte son, carte TV, etc. Au niveau du matériel, la coopération des constructeurs est très diverse et si la palme peut-être donnée à NVidia (fabriquant de cartes vidéo), il n'en est pas de même pour tous les constructeurs. Il est même certains constructeurs à éviter comme Pinnacle.

Ensuite il faut définir la machine elle-même : est-ce une machine existante ou une nouvelle machine ?

Si c'est une nouvelle machine, il faut préférer lors de son achat les intégrateurs qui permettent un choix des composants. Mais, dans tous les cas, il faut chercher, sur Internet, les problèmes pouvant exister avec certains matériels ou quel est le matériel adéquat à demander à son intégrateur lors de l'achat d'une nouvelle machine. S'il s'agit d'une machine de marque (mis à part IBM chez qui toute la gamme supporte actuellement GNU/Linux), demander au vendeur une confirmation du support matériel sous GNU/Linux mais préférez un intégrateur. La machine n'est souvent pas plus chère — toute la gamme de prix est possible et les intégrateurs savent mieux en général de quoi ils parlent. D'autre part, vous serez sûrs de ne pas avoir une machine de marque verrouillée.

9.3 Se renseigner

Une fois ce travail préliminaire effectué, vous aurez une idée claire de ce que vous voulez, vous saurez s'il faut conserver momentanément Microsoft Windows et installer le système GNU/Linux sur une autre partition ou si vous pouvez y aller franchement avec une machine uniquement sous GNU/Linux. Dans tous les cas, inscrivez-vous sur une liste de discussion vivante et posez la question. Il est très rare qu'une question n'ait pas sa solution. Posez la question de manière précise en donnant les caractéristiques de votre matériel et de vos besoins. Vous serez agréablement surpris. Ces communautés d'utilisateurs sont très vivantes et vous aiguilleront très rapidement sur les bons choix.

9.4 Choix de la distribution Linux

Choisissez une distribution conforme à votre niveau informatique. Si vous êtes débutants, opter pour une distribution Slackware ou Debian risque de se solder par un échec, alors qu'à l'inverse une Mandrake vous permettra presque à tous les coups de vous en sortir sans trop de difficultés.

De plus, ces logiciels ne répondant pas aux mêmes impératifs commerciaux que les logiciels propriétaires, ne vous dites pas : comme je prends une Mandrake (plus facile d'accès) et non pas une Slackware (largement plus difficile d'accès) j'en aurais moins ! C'est faux.

Toutes les distributions, si elles ne se ressemblent pas dans la mise en œuvre, possèdent les mêmes composants et programmes. La différence n'est pas dans « le mieux et le moins bien » mais dans la facilité ou la complexité de la mise en œuvre. Des distributions comme la Slackware ou Debian requièrent simplement des connaissances plus profondes du système.

9.5 Installation proprement dite

Enfin, lors des 20 minutes d'installation (pour une Mandrake), prenez les options standards proposées par l'installation. Vous pourrez toujours, par la suite, affiner selon vos besoins. Mais dites-vous bien que seront déjà installés l'équivalent de Microsoft Office, de Photoshop, idem pour les logiciels de gravure de CD (si vous possédez un graveur) et de tous les programmes d'utilisation courante (quelques centaines au bas mot).

9.6 Autres considérations

Un rapide calcul financier de cette installation, pourtant basique, sous Linux :

En considérant que vous ayez acheté un pack Mandrake, ce qui n'est pas une obligation, vous pouvez le télécharger gratuitement, dans un commerce afin de disposer du bon millier de pages de documentation, vous aurez dépensé environ 80 euros.

A l'installation, identique vous compterez environ 200 euros sur un système propriétaire, uniquement pour le système d'exploitation. Dès l'installation de certains logiciels tel que Microsoft Office afin de disposer d'un traitement de texte (Word), d'un tableur (Excel), d'un logiciel de traitement photo de la trempe de Photoshop, la somme de 2000 euros sera dépassée. Si l'on y ajoute des logiciels plus spécialisés comme Blender (3D) ou Jahshaka (post-production vidéo), le calcul ne justifie même plus le temps que l'on prend à le faire. Au bout des 20 minutes d'installations d'une distribution telle que Mandrake, tous ces logiciels seront présents. Ensuite vous n'aurez qu'à sélectionner dans les dizaines de milliers de programmes divers ceux qui vous seront utiles : lecteurs multimédia spécifiques, traitement du son, traitement de la vidéo, développement, serveurs de fichiers, serveur web, outils d'administration...

9.7 Exemple de migration en conservant Microsoft Windows

Voici un exemple concret de migration exécutée récemment par Pierre Lo Cicero (toutes les spécifications de ce matériel peuvent être trouvées sur Internet) :

- Boitier Soltek Qbic 3000
- Processeur Intel 4 2400Mhz
- Carte réseau Realtek intégrée, carte son AC3 intégrée.
- HDD de 80 GO.
- Scanner à plat Mustek Bear@Paw 1200 TA
- Carte vidéo onboard désactivée et remplacée par un NVidia MX400 avec sortie TV .
- Imprimante HP Deskjet 845 sur le réseau

En l'occurrence, cette machine était neuve et vide de tout systèmes.

9.7.1 Installation de Microsoft Windows 2000

Non pas que j'en ai l'utilité, moins de 5 heures par mois mais en prévision d'un traitement possible d'informations en format propriétaire. Il est clair que la machine permettra de lancer, au démarrage, indifféremment GNU/Linux ou Microsoft Windows. Le logiciel permettant ce choix (Lilo) s'exécute au démarrage de la machine, avant de charger un quelconque système d'exploitation. Microsoft Windows a la manie délibérée de supprimer ce programme. (ce n'est pas bien grave, la réinstallation étant simple, mais ça agace). Pour se simplifier la vie, donc, il est préférable d'installer Microsoft Windows en premier.

Disposant d'un disque de 80 GO, 5GO sont attribués pour Microsoft Windows 2000, formaté en FAT32, ce qui permet une lecture facilitée de cette portion du disque par GNU/Linux. Le reste du disque n'a pas besoin d'être partitionné, l'installation de Linux s'en chargeant parfaitement.

Cette installation a duré 1 heure 15. Elle ne comprend pas celle de pilotes comme ceux de la carte vidéo, de la carte son ou de l'USB. Ceux-ci nécessitent une recherche et un téléchargement sur le web. Pour un système moyennement complexe, on peut compter la journée entière (logiciels et drivers).

9.7.2 Installation de Linux

Partitionnement Linux et rajout d'une partition FAT32 depuis l'installateur Linux, et formatage.

Installation fonctionnelle de Mandrake Linux 9.1 standard effectuée en 35 minutes.

Tous les logiciels d'utilisation courante se trouvaient installés, traitement de texte (OpenOffice.org, équivalent de Microsoft Office), logiciel de retouche photo GIMP, scanner (5 minutes pour télécharger et installer le firmware adéquat), lecteurs multimedia, graveur de CD, navigateur web, client courriel, environnement de développement C/C++, KDE et Gnome ; outil de communication comme LICQ, outils de PAO comme Kile, compilateur et lecteur PDF... Je n'ai pas compté le nombre d'applications mais il y en a certainement bien plus d'une centaine.

9.7.3 Problème rencontré

A l'exception du réseau tout fonctionnait parfaitement, la puce Realtek possédant une particularité. La solution a été finalement trouvée en 2 heures et il a fallu 1 heure pour mettre par écrit cette solution afin qu'elle soit postée et puisse servir à toute la communauté.

9.7.4 Conclusions

Il est vrai que cela a été fait en disposant peut-être d'une certaine expérience mais à expérience égale, il faut un temps nettement supérieur sur un autre système, et je ne compte pas la rédaction d'un document utile à tous et publié sur internet.

10 Point de vue personnel des auteurs.

Les points de vue personnels des auteurs sont d'une teneur subjective mais tendent à montrer quelles peuvent être les motivations particulières d'un utilisateur de logiciels libres.

10.1 Yoann LE BARS

Pour moi, il est un peu étrange de parler de migration : ma découverte des systèmes UNIX et donc de GNU/Linux remonte à mon entrée dans l'enseignement supérieur, en 1997. Auparavant, j'avais presque exclusivement utilisé MS-DOS et n'avait pas installé Microsoft Windows 95. Il faut dire que je faisais principalement de la programmation et du jeu, activités classiques d'un gamin qui avait découvert l'informatique dans le milieu des années 80. Depuis 1997, je n'utilise plus que des systèmes de type UNIX, si l'on excepte une machine qui a, un temps, tourné sous Microsoft Windows 98 et dont mon utilisation cumulée doit monter à, au plus, une cinquantaine d'heures. De fait, pour moi, la migration a été de MS-DOS à UNIX puis GNU/Linux, autant dire qu'elle ne fut pas trop difficile. Je n'ai donc pas vraiment le point de vue de quelqu'un venant d'un autre système et l'utilisation de logiciels libres tout comme la participation à leur développement m'est assez naturelle.

Il est courant qu'un utilisateur de Microsoft Windows se tourne vers GNU/Linux par lassitude. Il me semble pourtant que les raisons pour utiliser des logiciels libres sont bien plus profondes qu'une quelconque fatigue. Les logiciels libres proposent un point de vue sur les œuvres de l'esprit, point de vue qui semble n'être que du bon sens. Ce point de vue propose des opportunités inconnues du système propriétaire. Examinons cela, la conclusion venant d'elles-mêmes.

10.1.1 Au sujet des principes des logiciels libres.

Ce qui me semble essentiel dans les logiciels libres n'est pas leur gratuité mais leur éthique sous-jacente. En effet, les logiciels libres rappellent que le savoir ne doit pas être possédé et retenu par quelques entités mais, au contraire, doit circuler librement. Les programmes informatiques ne sont pas des biens matériels mais des œuvres de la pensée et, en tant que telles, ils ne peuvent ni ne doivent être entravés dans leur circulation.

Aux États-Unis, la durée normale des droits d'auteurs est de soixante-dix ans. Passé ce délai, une œuvre doit normalement tomber dans le domaine public. Cependant, Disney, qui devrait voir Mickey tomber dans le domaine public, ne l'entends pas de cette oreille et cherche à en allonger la durée. Ceci s'étendrait à toutes les œuvres, dont de nombreuses n'ont pas une notoriété suffisante pour pouvoir perdurer sans être dans le domaine public. Tout un pan du patrimoine d'une nation risque donc d'être sacrifié pour le seul bénéfice d'un consortium, alors que Disney pourrait fort bien s'assurer des revenus avec ses œuvres d'une manière différente.

À l'heure actuelle, la tendance est à la rétention du savoir, des œuvres ou des logiciels (entre autres). Certains groupes ne voient dans les créations humaines que des moyens de s'enrichir. La structure mise en place par les logiciels libres, au contraire, montre qu'il est possible d'œuvrer pour le bien de tous et pourtant de bien en vivre. Voire très bien puisque certains développeurs de Linux, par exemple, sont devenus millionnaires.

Les logiciels libres peuvent donc servir d'exemple, montrant que le bien du patrimoine de l'humanité peut être préservé dans le cadre de notre société. Plus encore, ils montrent que solidarité et entraide ne sont pas de vains mots, ne sont pas seulement des vues de quelques idéalistes mais des valeurs ayant toujours un sens.

10.1.2 Il ne s'agit pas d'une lutte mais de bon sens.

Par contre, je constate certains amalgames, le principal étant que GNU/Linux a été créé pour remplacer Microsoft Windows. Je pense que cela est dû au fait que les logiciels libres ont désormais quitté le cercle restreint des informaticiens et des scientifiques pour se répandre vers de nombreux utilisateurs, certains assez peu au fait des techniques informatiques mais souvent politiquement assez engagés, en tout cas pour qui la notion de partage est essentielle. Le fait que des non-spécialistes se soient appropriés les logiciels libres, qu'ils y aient trouvé une résonance avec leurs préoccupations, est une bonne chose et c'est une victoire pour ces logiciels. Toutefois, il me semble intéressant de rappeler comment l'idée de logiciels libres est apparue.

Comme nous l'avons déjà présenté plus haut, jusque dans les années soixante-dix, l'informatique était cantonnée aux milieux de la recherche et de certaines industries de pointe. A cette époque, la règle était de partager son travail, créant une émulation permettant d'obtenir les meilleurs résultats possibles. Cette situation changea au début des années quatre-vingts, avec l'intrusion de la logique industrielle dans l'informatique. C'est bien le fait de ne plus pouvoir partager ses trouvailles, ce qui à l'évidence était un frein à l'innovation, qui poussa Richard STALLMAN à initier le mouvement du logiciel libre. Il me semble important de noter que Richard STALLMAN était un universitaire et qu'il s'est attaché à créer un système d'exploitation proche d'UNIX.

La puissance des ordinateurs personnels actuels, qu'ils soient compatibles IBM-PC ou Macintosh, nous permet mal de nous souvenir combien les choses étaient tranchées à l'époque : il y avait d'un côté l'informatique personnelle, peu puissante et principalement orientée vers le loisir et de l'autre l'informatique professionnelle. UNIX était alors un système d'exploitation dédié au gros système, un standard, certes mais dans le monde universitaire et de l'industrie. De plus, Microsoft Windows n'existait pas. De fait, l'objet des logiciels libres n'était alors pas de remplacer le système d'exploitation désormais le plus répandu sur les ordinateurs personnels mais bien de permettre au développement informatique de suivre la voie qu'il avait jusqu'alors toujours suivi et qui semble à Richard Stallman, comme à beaucoup d'autres dont moi, le meilleur garant de la constance dans l'innovation. De même, lorsque Linux est sorti, Microsoft Windows n'en était qu'à sa version 3 et n'était alors qu'une surcouche de MS-DOS.

En fait, toute la philosophie des logiciels libres me semble assez pragmatique, puisqu'elle est là pour régler des problèmes que l'on n'arrive pas à régler sans eux. Sans les logiciels libres, le développement des nouvelles technologies resterait entre les mains de quelques sociétés, avec tous les risques de stagnation et les difficultés pour l'enseignement que cela implique. Lorsque Linus Benedict TORVALDS développa Linux, il l'a fait parce qu'il n'y avait pas de système UNIX sur les compatibles IBM-PC (AT), sauf à un prix exorbitant et pour des résultats plus que moyens. La lutte de David (les logiciels libres) contre Goliath (Microsoft) n'est pas vraiment le but initial et n'est finalement qu'une conséquence, parce que Microsoft représente maintenant exactement ce contre quoi le logiciel libre a été créé. Il n'en a pas toujours été ainsi.

Ce qui me semble essentiel dans la création du mouvement du logiciel libre, c'est le fait qu'il rappelle que la connaissance se base toujours sur les travaux de ses prédécesseurs. Si nos sociétés sont arrivées au stade auquel elles sont parvenues à l'heure actuelle, c'est parce qu'elles ont sut tirer les leçons de leur héritage. Verrouiller d'une quelconque façon les savoirs ne peut, de toute évidence, qu'être un facteur de stagnation, voire de régression au vu du risque de la disparition de certaines expériences passées.

10.1.3 L'opportunité proposée par ce mode de fonctionnement.

Ce qui est gratifiant pour tous ceux qui se sont investis et s'investissent encore dans le logiciel libre, c'est de voir que de plus en plus de sociétés adoptent ce schéma de développement, certaines parmi les plus grands acteurs du monde de l'informatique, tels que Sun Microsystem ou IBM, les poussant à créer un nouveau modèle économique, basé sur le service, qui fonctionne : IBM et Sun Microsystem ont connu une période de contre-performances dont la fin correspond avec l'implication dans les logiciels libres. Ceci prouve d'abord que les enjeux commerciaux sont suffisamment grands pour que d'importantes entreprises y trouvent leur compte. Ensuite, cela signifie que la logique commerciale n'est pas incompatible avec l'enrichissement commun.

A côté de cela, la structure mise en place par les logiciels libres permet un énorme gain de temps et d'argent. En effet, lorsque je développe en accord avec les principes du libre, je peux m'appuyer sur tout ce qui a déjà été fait par d'autres — et la libre disponibilité des sources permet une amélioration constante des logiciels libres — mais, de plus, j'ai la satisfaction de produire quelque chose qui pourra être utile à n'importe qui.

La structure des logiciels libres propose une chance unique pour l'utilisateur, celle de redevenir le propriétaire de son matériel et des fichiers qu'il produit mais aussi propose de remanier en profondeur l'idée de l'informatique, pour la voir désormais comme une compétence, un savoir, pour lequel on propose un service. Cette structure propose non seulement de libérer les utilisateurs de logiciels mais aussi d'autres domaines, en suivant cet exemple. En effet, elle rappelle que ce qui est savoir ne s'épuise pas dans le partage mais au contraire ne peut s'enrichir véritablement que de cette façon.

Savoir quelle est la plate-forme (le système d'exploitation) n'est absolument pas essentiel. D'ailleurs, ceci n'a rien d'une lutte. L'essentiel est de voir que la mise en commun permet d'aller plus loin et ne peut être que bénéfique à tous, qu'ils soient utilisateurs individuels ou professionnels.

10.1.4 Conclusion.

Bien souvent, on oppose aux logiciels libres le fait que notre société est bâtie pour l'enrichissement personnel, qui ne peut se faire qu'au détriment de celui d'autrui. Je crois que la réussite avérée de ce mode de fonctionnement est le déni le plus flagrant de cette pensée : au sein d'un système bâti pour l'égoïsme, la notion de partage parvient, bien que l'on tente de l'entraver — je fais référence à l'affaire des brevets logiciels qui a récemment fait polémique au sein du gouvernement européen et qui n'est d'ailleurs pas finie, — est capable de proposer un mode de fonctionnement qui fonctionne et est bénéfique à tous. De toute façons, je pense que c'est dans l'air : le monde de la recherche a toujours eut l'habitude de partager son travail. Les logiciels libres ont proposé une méthode viable pour reproduire ce principe sur des objets commerciaux. D'autres domaines commencent à suivre cet exemple, puisque certains artistes rendent leurs œuvres disponibles selon un modèle s'inspirant de celui mis en place par la Free Software Foundation. Je pense que d'autres secteurs d'activités auraient tout à gagner à suivre ces exemples. Pour en revenir aux logiciels, le mouvement est initié et il perdura, avec ou sans nous.

10.2 Pierre LO CICERO

Mes premières expériences avec Linux ont été motivées par la curiosité ainsi qu'un certain ras-le-bol. Il s'agissait à l'époque (1991 ou 92 si je me souviens bien) d'une Slackware que je n'ai jamais su faire fonctionner. L'utilisation quotidienne n'a démarré qu'avec une RedHat 5, un peu plus tard. Contrairement à Yoann, j'ai à un moment choisi de migrer sous Linux. Quelles ont donc été les motivations pour choisir un système tel que celui-ci ?

Elles sont de trois ordres :

10.2.1 Pratique

Ayant débuté en informatique sous DOS, avec comme outil de programmation le BASIC puis plus tard le QUICKBASIC, ce fut surtout une période de découverte. Désireux de poursuivre, j'ai continué en suivant les améliorations au fur et à mesure de leur progression.

Après un bref passage (très bref) sous Windows version 2, enfin est arrivé la version 3.0 puis 3.1. L'interface graphique aidait bien aux quelques manipulations nécessaires et simplifiait la gestion de la machine. Dans l'utilisation courante, j'utilisais principalement deux logiciels : 3D Studio version 3 et, après un rapide passage par Visual Basic 1, Visual Basic 3 (la version 2 n'ayant jamais existé). En ce qui concerne le C/C++ rien sauf Borland, l'environnement de Microsoft, depuis l'ex-programmation sous DOS, ressemblant à un cauchemar. Ce fut aussi l'époque des premières déconvenues avec le DOS 6.0, qui « giclait » les disques durs mieux qu'il ne les gérait et surtout, ce qui m'a agacé au plus haut point, sortir son porte-monnaie pour avoir une mise-à-jour, la 6.1. Une amélioration à peine perceptible qui obligeait quiconque l'utilisait à passer à la 6.2.

La deuxième déconvenue et celle-là fut financièrement de taille : l'achat de 3Dstudio 3 (3 600 francs suisses à l'époque) m'avait coûté « chaud. » J'envisageais, dès la sortie de win95, la mise-à-jour en 3DSmax 1. Cependant, il s'est avéré que le vendeur de ma première version m'avait livré une version de démonstration, certes complète mais destinée aux points de vente uniquement. Malgré les divers contacts avec Autodesk, la production des pièces comptables, je n'ai eu comme proposition que la possibilité d'acheter une version complète de 3DSmax ou de faire un procès à mon vendeur, qui entre-temps avait fait faillite.

Win95, défini à cor et à cris comme un système 32 bits, en avait perdu 16 au passage (il faudra attendre Win2000 et XP pour cela). L'introduction du réseau intra et extra net n'a fait que compliquer les choses. Impossible d'y accéder sauf en disposant de finances exagérément bien fournies.

Les errements de Microsoft, comme sous NT 4 le service pack possédant des « Cheat Codes » disponibles uniquement par les services d'information de l'armée américaine, la difficulté, professionnellement parlant, de développer des solutions qui soient stables, avec une compatibilité ascendante garantie, le mépris affiché par Microsoft pour ses utilisateurs, tout cela, finalement, n'a abouti qu'à une seule conclusion : ce genre de produit n'est pas viable.

Responsable de l'informatique dans l'entreprise où je travaillais en 1998, l'introduction de Linux à été une véritable bouffée d'air frais, à la satisfaction de tous les utilisateurs et clients qui disposaient enfin d'outils fiables. La sortie de XP, l'élaboration par Microsoft du projet Palladium renommé aujourd'hui NGSCB, voir [17], [18] et [19], pour ma part, a mis le coup de grâce à l'utilisation de ce genre de produits, bien trop dangereux et sans aucun doute précurseurs d'un monde semblable à certains films cauchemardesques inspirés du roman « 1984. »

Je peux l'affirmer : dans tous les domaines, Linux m'a simplifié la vie.

10.2.2 À propos de « morale »

Être traité de pirate, voire de criminel, si j' utilise la copie d'un logiciel a contribué à ce que je me penche sur ce problème particulier qui justifie de la part de certains éditeurs un verrouillage des informations, des connaissances mais aussi surtout un verrouillage économique

Qu'est-ce qu'un pirate, quand y a-t-il un tort véritable ? On peut se poser la question. Si je fais une copie d'une ceinture Cartier, que je la vends comme du Cartier, oui je suis un pirate car, manifestement :

1. je trompe l'acheteur ;
2. je m'enrichis aux dépends d'autrui.

Voilà la définition d'un pirate. Mais si j'utilise la ceinture Cartier qu'un ami m'a prêtée ou donnée, en suis-je un ? Certainement non car où ai-je causé un tort ? Où me suis-je enrichi aux dépends d' autrui ?

Pour le reste, je donnerais la parole à André SALAÜN car je souscris à son analyse :

« Pourtant lorsqu'il s'agit d'un logiciel, la simple utilisation, même temporaire, tombe sous le coup de la Loi ! Dans le même temps les sociétés, organismes ou fondations en émanant, défendent dans leur argumentaire anti-copie que le logiciel est un produit comme un autre ce pour quoi on doit l'acheter. Certes ! Mais alors pourquoi ne peut-on le prêter ? Car il n'est pas comme un autre ? C'est une oeuvre intellectuelle et son utilisation doit être soumise à un contrôle. Ah ! Ça se complique !

Mais alors pourquoi puis-je prêter en toute légalité un livre voire un cd à un ami sans qu'il paie pour l'écouter ou le lire ? On développe à partir d'ici des arguments spécieux qui consistent à tenter de démontrer que le logiciel est un produit comme un autre (on le vend) tout en ne l'étant pas (le logiciel acheté n'est pas la propriété de l' acheteur).

La vérité est bien sûr ailleurs : en fait dans l'analyse (grâce entre autres à l'informatique) comptable et dans ce concept économique devenu aberrant qu'est *le manque à gagner*. Si cette idée peut se justifier concernant une fermeture imprévue d' entreprise pour des causes diverses, il a aujourd'hui été porté à son comble en incluant *ce qu'on pourrait gagner en plus si les conditions sociales, politiques et économiques étaient idéales pour l'entreprise, le groupe ou la communauté dont on défend les intérêts*. Autrement dit si la société dans laquelle elle fonctionne n'était organisée qu'en vertu de l'intérêt de la dite entreprise. Ceci est en fait purement rhétorique mais engendre un lobbying d'une agressivité peu commune masquée derrière des valeurs morales de bon aloi : le vol, le piratage voire l'anti-patriotisme dans les cas les plus graves. Les termes utilisés sont d'ailleurs issus du même registre moral, que ce soit du côté des puissances totalitaires ou du côté des groupements internationaux d'intérêts privés : on appelle cela de la propagande !

Il s'agit à partir de là de construire (dans les pays totalitaires) ou d'influencer (par le lobbying, dans les pays démocratiques) un arsenal législatif conforme à ses propres intérêts, dût-il se faire au détriment de l'intérêt collectif (concernant le droit) ou individuel (concernant les libertés). On recouvre le tout d'un discours hautement moral, voire moralisateur et le tour est joué ! On a dénoncé ce phénomène sous le nom de "Stalinisme" dans certains pays, les sociologues parlent de "libéralisme stalinien" en ce qui concerne nos démocraties !

La *communauté du libre* (si tant est qu'elle existe et de toute façon pas sous forme d'organisation monolithique) est quant à elle beaucoup plus pragmatique et reconnaît volontiers qu'un

auteur doit conserver ses droits et une entreprise le bénéficie de son travail industriel et de ses investissements. Mais pour cela point n'est besoin de créer un statut particulier pour ce produit qu'est le logiciel. En tant que produit industriel il est promu (et défendu) par les lois communes sur le commerce et l'industrie, en tant qu'œuvre de l'esprit il l'est par le droit d'auteur. Ainsi chacun bénéficie des fruits de son travail (la licence GPL reconnaît et est issue du droit de l'auteur) et la diffusion de ces nouvelles connaissances n'est pas bridée par une appropriation abusive uniquement liée à la puissance financière comme l'est la notion de brevet !

Si la *communauté du libre* exprime une quelconque *morale* elle pourrait se traduire ainsi : **le droit de tous et de chacun sans renier le bien commun**. Il n'y a ici rien de bien révolutionnaire ou d'idéaliste... »

10.2.3 Philosophique

Il y a bien des choses qui sont inaliénables à l'être humain. Il ne s'agit pas de « droits » ni de « devoirs, » pour la simple et bonne raison que c'est intimement lié à l'être humain et est indissociable de sa nature. Il s'agit plus ici de la soif de connaissance d'une part et la transmission de la connaissance acquise d'autre part. Tout homme qui voudrait brider cela bride l'être humain pour le ramener à à peine plus que l'animal. Une telle pensée, quoique la méthode soit différente, aboutit aux mêmes résultats que certaines théories contemporaines désastreuses, réduisant l'être humain à une fourmi contrainte à mendier aux marchands jusqu'à l'air qu'elle respire.

10.2.4 En conclusion

L'informatique est le métier que je pratique, donc une part importante de ma vie. Lorsque je dis métier, cette dénomination est en fait relativement récente. L'informatique est bien, au XXI^e siècle, à la jonction entre « quelque chose de vague » et un métier véritable, peu différent de celui d'un charpentier ou d'un maçon dans les connaissances très fines qu'il requiert, connaissances spécifiques que certains ont voulu mettre « sous le boisseau » afin de s'en arroger des bénéfices substantiels. J'ai trouvé dans GNU/Linux un outil que je considère comme éminemment utile et efficace. Pour exprimer cela, je prendrais une image :

Les cathédrales, qui 700 ans plus tard, sont encore un objet d'admiration ont été bâties sur le même modèle que ce qui est exposé plus haut, c'est-à-dire par les gens de la cité, amenant chacun leur propre pierre à l'édifice. Les motivations certes étaient diverses et sans commune mesure mais la méthode exactement la même. On ne se souvient certainement pas de la « marque » du marteau ou de la scie qui a servi, pas plus que du nom de l'artisan bâtisseur d'ailleurs. Pourtant ces cathédrales sont bien là et témoignent d'un art (« techne » en grec signifiant à la fois art, technique, métier) et surtout d'une méthode analogue.

De plus, GNU/Linux me libère du fait de devoir « voler, » vol supposé ou de fait, une connaissance. En définitive, je trouve particulièrement exaltant, au XXI^e siècle, de vivre ce qui a déjà été exprimé il y a longtemps :

« Nous sommes comme des nains portés sur les épaules de géants. Nous voyons plus de choses que les anciens et de plus éloignées, non par la pénétration de notre propre vue ou par l'élévation de notre taille mais parce qu'ils nous soulèvent et nous haussent de toute leur hauteur gigantesque. »

Bernard de Chartres, XII^e siècle.

11 Pour conclure.

Nous pensons avoir exprimés des arguments essentiels concernant l'intérêt que représente la migration vers un système libre, ainsi que les avantages que l'on peut en retirer. Nous avons volontairement mis de côté les aspects idéologiques et politiques que certains voudraient y attacher, ce genre de considérations étant hors sujet. Nous espérons juste avoir réussi à donner des arguments valables aussi bien à l'utilisateur familial qu'à l'entrepreneur, l'industriel ou l'homme politique, à chacun d'en faire un usage adéquat.

Comment terminer ce document ?

Nous avons parlé de licences libres GNU/GPL. Il est honnête de citer quelques extraits d'une licence appartenant à un de ses détracteurs, en l'occurrence un acteur majeur de l'informatique : Microsoft.

Microsoft EULA : End User Licence Agreement : (cf. Microsoft XP Professional)

- « [...]Vous autorisez Microsoft et Microsoft Corporation à collecter et à utiliser les informations techniques recueillies de quelque manière que ce soit [...] »
- « [...] Microsoft, Microsoft Corporation et leurs affiliés peuvent divulguer ces informations à des tiers[...] »
- « [...]Vous reconnaissez et acceptez que Microsoft, Microsoft Corporation ou ses filiales puisse automatiquement vérifier la version du LOGICIEL et/ou des composants que vous utilisez et puisse fournir des mises à jour ou des corrections au LOGICIEL qui seront automatiquement téléchargées sur votre Ordinateur [...] »

À chacun de voir s'il est, pour lui-même, son entreprise ou son département, acceptable de laisser de telles possibilités à un tiers.

12 Références pour compléter cet article.

Voici quelques références qui ont permis la rédaction de cet article ou qui permettent d'aller plus loin.

Références

- [1] <http://www.gnu.org/home.fr.html>, la partie française du site du projet GNU et de la Free Software Foundation.
- [2] <http://fr.openoffice.org/Marketing/mateexpo/Documents/index.html>, une adresse où on trouvera plusieurs documents au sujet de la migration de Microsoft Office vers OpenOffice.org, notamment le mémoire de Frédéric LABBÉ.
- [3] <http://fr.openoffice.org>, la partie française d'OpenOffice.org.
- [4] Sophie GAUTIER, Christian HARDY, Frédéric LABBÉ, Michel PINQUIER : *OpenOffice.org 1.1 efficace*, Eyrolles. Un livre qui présente non seulement comment utiliser au mieux OpenOffice.org mais aussi comment trouver l'information, comment déployer ce logiciel à petite comme à grande échelle et donne des informations sur la migration de Microsoft Office à OpenOffice.org.
- [5] www.mozilla.org Le site officiel de Mozilla.
- [6] *Linux Magazine 54 (octobre 2003.)* Les conférences sur les logiciels libres de l'EPPLUG (association des utilisateurs de Linux de la région Picarde) y ont été publiées sous forme vidéo dans le CD-ROM d'accompagnement de ce magazine.
- [7] Cyril NOCTON : *Mandrake Linux 9.1*, Micro Application. Présentation du système GNU/Linux par l'intermédiaire de Mandrake Linux 9.1.
- [8] <http://www.li.org/linuxhistory.php>, un site sur lequel Linus Benedict Torvalds commente les messages échangés sur comp.os.minix.
- [9] <http://www.om-conseil.com/sections.php?op=viewarticle&artid=42>, site présentant un large choix de logiciels libres sous GNU/Linux et ses équivalents sous Windows.
- [10] <http://lea-linux.org/> Le site de Léa (Linux Entre Amis), qui donne des conseils pour utiliser GNU/Linux.
- [11] <http://www.aful.org> Le site de l'AFUL (Association Francophone des Utilisateurs de Linux et de logiciels libres).
- [12] Hannu PUTTONEN : *Nom de code Linux*. Le 14 mars 2002, Arte a diffusé ce reportage sur Linux, parlant aussi des logiciels libres d'une manière générale.
- [13] Jean-Marie GOUARNÉ http://solutions.journaldunet.com/0301/030115_trib_gouarne.shtml#1
Un article datant du 15 janvier 2003, qui fait le point sur les vrais enjeux et les erreurs courantes au sujet de la bureautique libre.
- [14] <http://europa.eu.int/ISPO/ida/jsps/index.jsp?fuseAction=home> Site de l'IDA (*Interchange of Data between Administrations*).
- [15] Rosaire AMORE : *Pourquoi les logiciels libres ?*, Linux Pratique n°21 (janvier-février 2004).
Un article présentant les risques de la politique hégémonique de Microsoft.
- [16] <http://sources.redhat.com/cygwin/>, adresse où l'on peut télécharger cygwin.

- [17] <http://www.laboratoire-microsoft.org/articles/win/ngscb/2/>, concernant le projet Palladium.
- [18] <http://www.zdnet.fr/techupdate/infrastructure/0,39020938,2121755,00.htm>, portant sur le même sujet.
- [19] <http://linuxfr.org/2002/10/07/9882.html>, traitant des risques encourus si nous laissons le projet Palladium arriver à terme.
- [20] http://www.vecam.org/article.php3id_article=270, à propos du biopiratage et de la propriété intellectuelle.

13 Au sujet des logiciels cités dans l'article.

Microsoft Windows, Microsoft Office, Microsoft PowerPoint, Microsoft Outlook, Microsoft Publisher, Microsoft Word, Microsoft Excel, Microsoft FrontPage, Windows Media Player, Quick Basic and MS-DOS are trademarks or registered trademarks of Microsoft Inc. Photoshop is a trademark of Adobe Inc.

Linux is registered trademark of Linus Torvalds.

Sun StarOffice and Java are trademarks or registered trademarks of Sun Microsystems Inc.

Mac OS is registered trademark of Apple Computer Inc.

Mandrake et Mandrake Linux sont des marques déposées par MandrakeSoft SA.

Debian is a registered trademark of Software in the Public Interest, Inc.

Slackware® is a registered trademark of Slackware Linux, Inc

KDE and K Desktop Environment are trademarks of KDE

GNOME is trademark of the GNOME Foundation.

14 GNU Free Documentation License

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

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

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

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

1. APPLICABILITY AND DEFINITIONS

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

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

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

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

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

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

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats

include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

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

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

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

2. VERBATIM COPYING

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

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

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

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

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

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

4. MODIFICATIONS

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

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions

- (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
 - C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
 - D. Preserve all the copyright notices of the Document.
 - E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
 - F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
 - G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
 - H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - N. Do not retitling any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or

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

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

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

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

5. COMBINING DOCUMENTS

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

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

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled

"History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

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

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

7. AGGREGATION WITH INDEPENDENT WORKS

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

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

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a

translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

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

10. FUTURE REVISIONS OF THIS LICENSE

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

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

ADDENDUM: How to use this License for your documents

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

Copyright (c) YEAR YOUR NAME.

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

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

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

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

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