

**Comparaison et justification économique des  
solutions basées sur des logiciels libres en  
matière de politique informatique pour les  
serveurs d'entreprises par rapport aux  
solutions propriétaires traditionnelles.**

Jury:  
Directeur:  
Wilfried NIESSEN  
Rapporteurs:  
Alain DUBOIS  
Yves WESCHE

Mémoire présenté par  
**Laurent RICHARD**  
En vue de l'obtention du diplôme de  
Licencié en Sciences Commerciales  
et Financières  
Orientation Finances  
Année académique 2001/2002

Copyright (c) 2002 Laurent RICHARD. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

## Remerciements

Je remercie Monsieur NIESSEN pour avoir accepté de diriger ce mémoire et Monsieur WESCHE pour ses explications techniques et pratiques sur les logiciels libres.

Je tiens à remercier tout particulièrement Mademoiselle Nathalie SAMSON pour son aide précieuse, la correction de ce mémoire et la pertinence de ses remarques qui m'ont permis de bénéficier d'un oeil critique et de vulgariser ce mémoire.

Je remercie aussi Monsieur Axel KRAUSCH et Réginald RATZ pour leurs heures passées à la correction de ce mémoire et leur soutien moral.

De plus, je tiens à remercier Messieurs Nicolas PETTIAUX et Alexandre DULAUNOY pour leurs précisions techniques, leurs conseils quant aux points à aborder dans ce mémoire, leurs propositions de livres et articles à lire et leurs avis d'experts de la question GNU/Linux, des logiciels libres et des libertés informatiques.

Je tiens à exprimer aussi ma plus profonde gratitude aux membres du groupe d'utilisateurs Linux de Liège (LiLiT) qui m'ont soutenu ainsi que les diverses personnes sur les forums d'Internet qui ont téléchargé mon mémoire et ont contribué par leurs commentaires à l'exactitude du contenu de celui-ci.

Je remercie en outre toutes les personnes qui, de près ou de loin, ont contribué à l'élaboration du mémoire.

# Avant-propos

Pour illustrer la comparaison entre les logiciels libres et les logiciels propriétaires, j'ai décidé de m'en tenir à certains logiciels connus. Les deux plus connus étant Windows de Microsoft et GNU/Linux. J'ai préféré m'en tenir là plutôt que de me perdre dans des comparaisons n'ayant plus de caractère précis car les logiciels propriétaires sont très divers, cependant je me suis attaché à rester le plus général possible. Dans les pages qui vont suivre, d'autres noms seront abordés mais cela sera très peu fréquent.

Tout au long de ce mémoire, le terme GNU/Linux sera utilisé pour représenter tout le système d'exploitation et les applications fonctionnant grâce à celui-ci. La terminologie exacte du terme Linux fait référence au noyau seul.

De même, dans un but de clarté et pour ne pas entrer dans des conceptions philosophiques, j'utiliserai indifféremment les termes « logiciels libres » ou « Open Source » pour désigner la même chose. En effet, certaines personnes pourraient argumenter que certains logiciels propriétaires ont ou pourraient avoir un code source accessible. Dans l'idée de ce mémoire, le terme Open-source n'a d'autre signification que celle de logiciels libres ayant par définition leur code source ouvert, et non la traduction littérale du terme anglais Open Source.

En parlant de logiciels libres, je ne parle pas de n'importe quels logiciels libres car nombreux sont ceux qui se trouvent toujours à phase de développement assez peu avancée. Il est donc impensable de comparer un logiciel qui n'a pas encore été suffisamment testé et qui n'a pas encore atteint un niveau satisfaisant pour répondre efficacement à la demande du marché. Dans ce mémoire, je m'en tiendrai aux logiciels libres ayant atteint un seuil de maturité suffisant, comparables aux solutions traditionnelles en termes de fonctionnalité, et donc aptes à répondre aux demandes du marché et de la société souhaitant acquérir un logiciel neuf ou remettre en cause ses logiciels actuels.

Je ne m'attacherai pas non plus à comparer des logiciels qui sont gardés secrets par les entreprises pour des raisons d'avantages compétitifs. En effet, dans des cas bien précis, il est pas possible de changer des logiciels ou de les rendre publics sans porter atteinte à la continuité des revenus de l'entreprise. Je développerai donc ici mon argumentation sur des logiciels qui permettent à l'entreprise de s'améliorer sans pour autant perdre de l'argent car, bien sûr, l'idée est de réaliser le meilleur choix dans une optique d'optimisation des investissements et des performances.

# Plan du mémoire

Introduction

Chapitre 1: Les logiciels libres, Open Source et Linux

- 1.1 Définition
- 1.2 Historique GNU/Linux et \*BSD
- 1.3 GNU/Linux, ce concurrent de Microsoft
- 1.4 Le concept de logiciel libre
- 1.5 Quelques caractéristiques
- 1.6 Les licences d'utilisation
- 1.7 L'importance de l'ouverture des sources
- 1.8 Le cas du logiciel libre dans l'Administration française
- 1.9 Quel futur pour de tels logiciels?

Chapitre 2: Business modèles

- 2.1 Quelques modèles
- 2.2 Quelques exemples

Chapitre 3: Les logiciels libres contre les logiciels propriétaires

- 3.1 La situation actuelle du logiciel libre et Open Source en entreprises
- 3.2 Quels avantages pour les sociétés utilisatrices?
  - 3.2.1. Compatibilité (matérielle, protocoles, ...)
  - 3.2.2. Portabilité
  - 3.2.3. Performance
  - 3.2.4. Fiabilité
  - 3.2.5. Sécurité
  - 3.2.6. Pérennité
  - 3.2.7. Indépendance des vendeurs
  - 3.2.8. Coûts des formations
  - 3.2.9. TCO

- 3.3 Avantages apparents des logiciels propriétaires
- 3.4 Quels sont les apports des logiciels Open Source et propriétaires à l'entreprise?
- 3.5 Comparaison chiffrée entre des solutions de type Microsoft et de type Linux

Chapitre 4: Le cas de l'Administration péruvienne.

Conclusions

Lexique

# Introduction

L'industrie du logiciel connaissait, depuis les années 60 et plus encore avec l'arrivée d'Internet, un intérêt croissant à fournir les entreprises en outils de gestion et de vente des produits dans des formes les plus diverses.

Avec le temps, toute cette industrie s'est finalement concentrée en quelques acteurs fournissant des programmes associés de licences d'exploitation diverses et de droits de copyright sur leur création pour empêcher toute récupération. Ce sont les logiciels propriétaires. Ce modèle a permis à des entreprises de gagner beaucoup d'argent, de s'agrandir, de racheter des sociétés prometteuses, ... et ainsi de contrôler une grande partie du secteur.

Face à cela, une communauté de développeurs reliés entre eux via Internet créent des programmes pour lesquels ils permettent à tout un chacun de voir le code source et proposent de l'améliorer si la personne le désire et de remettre leur amélioration à la disposition de la communauté. C'est un type de développement totalement différent et formant un nouveau modèle économique basé sur le partage. Les produits de ce modèle économique se nomment « logiciels libres ».

Ceux-ci ont connu ces derniers temps une croissance fulgurante grâce à leur système de développement mais surtout grâce à la qualité et l'intérêt des solutions créées, et ce à l'encontre du schéma traditionnel des logiciels propriétaires.

Depuis quelque temps, les principaux acteurs du monde informatique se passionnent pour les logiciels libres. En effet, IBM a décidé d'investir plus d'un milliard de dollars dans Linux en créant des départements de développement interne. Hewlett-Packard a engagé Bruce Perens, éminent gourou du libre et un des fondateurs du concept de open source, dans le but de définir leur stratégie Linux.

De plus en plus de serveurs commencent à être commercialisés avec Linux pré-installé chez ces différents acteurs.

Qu'attendre de cette nouvelle vague qui semble bousculer les habitudes propriétaires des Unix et Microsoft déjà présents sous différentes formes dans les entreprises?

Quels avantages en termes de coûts, de gains et de fonctionnement mon entreprise peut-elle trouver à envisager une telle possibilité?

Pour répondre à ces principales interrogations, j'ai subdivisé ce mémoire en plusieurs chapitres:

- un chapitre présentant Linux et les autres logiciels libres;
- un chapitre pour montrer différentes façons d'utiliser GNU/Linux de façon stratégique selon des exemples réels;
- un chapitre de comparaison des logiciels propriétaires et libres au niveau du fonctionnement, des attentes concernant le rendement et des tâches à effectuer ainsi qu'une comparaison chiffrée;
- un chapitre montrant l'intérêt des logiciels libres par rapport aux logiciels propriétaires dans le cadre d'un exemple réel de l'administration péruvienne.



# **Chapitre 1: les logiciels libres, Open Source et Linux**

## **1.1 Définition**

Si on en reste à une définition assez formelle, le logiciel se définit comme « l'ensemble des programmes, procédés et règles et éventuellement de la documentation, relatif au fonctionnement d'un ensemble de traitements de données » <sup>1</sup>. Plus généralement, on pourra réduire cela à un ensemble d'instructions permettant à un utilisateur d'obtenir un certain résultat de la part de l'ordinateur.

## **1.2 Historique GNU/Linux et \*BSD**

UNIX a vu le jour sous la forme d'un projet de recherche dans les laboratoires d'AT&T Bell en 1969. En 1976, il a été distribué gratuitement aux universités. Il a donc servi de base à plusieurs cours sur les systèmes d'exploitation ainsi qu'à des projets de recherche académique.

L'UNIX de Berkeley a vu le jour en 1977 lorsque le CSRG (Computer Systems Research Group) de l'université de Berkeley, en Californie, a vendu le code d'AT&T sous licence. Les versions de Berkeley (appelées BSD, pour Berkeley Software Distribution) ont été distribuées à partir de 1977.

Unix est l'un des systèmes d'exploitation les plus appréciés, notamment en raison du grand nombre d'architectures prises en charge. Tout d'abord développé en tant que système d'exploitation multitâche pour des mini-ordinateurs et des gros systèmes au début des années septante, il est aujourd'hui omniprésent en dépit d'une interface en ligne de commande parfois déroutante pour les habitués des interfaces graphiques et d'un manque d'une réelle standardisation.

Unix est populaire au point que de nombreux programmeurs le considèrent comme le seul système d'exploitation viable et lui vouent un culte quasi religieux. Il existe des versions d'Unix pour une vaste gamme de machines, depuis l'ordinateur personnel jusqu'au super-calculateur. Néanmoins, la

---

<sup>1</sup> <http://www.ccip.fr/irpi/faq/logiciel/definition.htm>

plupart de ses implémentations pour PC restèrent longtemps onéreuses.

On comprend alors que certains ont mis la main à la pâte afin de réaliser une version d'Unix librement diffusable. Ainsi, un dénommé Linus Torvalds, alors étudiant à l'université d'Helsinki (Finlande), développa la première version de Linux. De nombreux programmeurs et spécialistes d'Unix participent aujourd'hui activement à l'optimisation du noyau Linux grâce au réseau mondial Internet. En outre, toute personne compétente désireuse de contribuer à l'évolution du système est invitée à y participer. Le noyau de GNU/Linux n'utilise aucun code issu d'une version commerciale d'Unix ou de toute autre source propriétaire et la plupart des programmes disponibles pour GNU/Linux sont conçus dans le cadre du « projet GNU »<sup>2</sup>, qui vise à constituer un système complet librement diffusable.

Linus Torvalds s'est inspiré de Minix, modeste système Unix développé par Andy Tanenbaum.

Linus Torvalds se proposait avant tout de maîtriser la commutation de tâches en mode protégé sur le processeur 80386. Tout fut réalisé en langage Assembleur. L'avènement de la version 0.01 ne fit l'objet d'aucune notification préalable. Elle n'était même pas exécutable et l'archive, qui ne contenait que le squelette des sources du noyau, imposait l'accès à un système Minix pour compiler et découvrir Linux.

Le 5 octobre 1991, Linus délivra la toute première version « officielle » de Linux (0.02). A défaut de mieux, le système était à présent capable d'exécuter bash (le Bourne Again Shell de GNU) et gcc. En effet, les premières moutures de Linux étaient surtout destinées aux hackers et se focalisaient sur le développement du noyau; ce qui explique l'absence de service d'assistance technique, de documentation, etc. Dans une moindre mesure, la communauté GNU/Linux considère encore aujourd'hui ces aspects comme très secondaires en regard de la « vraie programmation »: le développement du noyau.

Une fois diffusée la version 0.03, Linus passa directement à la version 0.10, au vu du développement rapide. Puis, après plusieurs autres révisions, Linus, qui présentait que Linux serait bientôt prêt pour une version « officielle », gonfla à nouveau le numéro qui passa alors à 0.95 (par convention, le numéro de version 1.0 n'est attribué qu'après l'obtention d'un programme théoriquement complet et dénué de « bugs »). Dix-huit mois plus tard (fin 1993), Linus apportait la touche finale à la version 0.99.pl14 du noyau, s'approchant de 1.0 de manière apparemment

---

<sup>2</sup> <http://www.gnu.org>

asymptotique. Certains ne croyaient plus que la version 1.0 verrait le jour. Elle existe pourtant, et a tant et si bien évolué que la version stable actuelle est la version 2.4.19.

La version d'Unix issue de Berkeley (dite « BSD ») a, elle aussi, joué un rôle important, quoique davantage à travers la mise à disposition de ses fameux outils que lors de la création de Linux. Beaucoup d'utilitaires fournis avec les distributions du système GNU/Linux proviennent de portages de programmes BSD. A vrai dire, la partie réseau du code de Linux a été développée en totalité (deux ou trois fois, en fait), mais ses commandes émanent en droite ligne de BSD.

A ce jour, GNU/Linux est un système Unix complet capable de prendre en charge le serveur graphique X Window, les protocoles TCP/IP et UUCP, mes éditeurs de texte Emacs et vi, le courrier électronique, les forums Usenet et bien d'autres utilitaires et fonctions. Rares sont les logiciels libres célèbres qui n'ont pas encore été portés sous GNU/Linux, et il serait intéressant de voir plus de grands éditeurs proposer des applications commerciales. Ce système reconnaît aujourd'hui beaucoup de périphériques, ce qui n'était pas le cas dans ses premières versions. A de multiples occasions, les machines dotées de Linux ont affiché des performances comparables aux stations de travail de milieu de gamme de Sun Microsystems et Digital Equipment Corporation. Qui aurait pu imaginer que cet Unix minimaliste gagnerait un jour une telle ampleur et un tel rayonnement?

Linux a été reconnu conforme au standard POSIX (.1 et .2) grâce aux efforts de la société Lasermoon. Au-delà de sa valeur symbolique, une telle reconnaissance accélère la propagation de GNU/Linux au sein des sociétés commerciales mais aussi de l'administration qui exige la conformité POSIX de la plupart des systèmes qu'elle utilise (MS-Windows compris, comme en témoignent les aménagements apportés à la version NT par Microsoft).

### **1.3 GNU/Linux, ce concurrent de Microsoft**

GNU/Linux est un système d'exploitation, qui agit en tant que service de transmission entre le hardware (le matériel physique d'un ordinateur) et les softwares (les applications qui utilisent le matériel) d'un système informatique.

Pour comprendre GNU/Linux, il faut d'abord comprendre UNIX. Or, ce système d'exploitation est probablement le plus versatile et le plus populaire parmi ceux qui occupent actuellement le marché des serveurs et des stations de travail scientifique et à hautes performances.

Le projet Linux a commencé avec l'objectif de construire une version fonctionnelle d'UNIX pour les ordinateurs basés sur les processeurs Intel, autrement dit, les ordinateurs compatibles avec les PC que tout un chacun connaît aujourd'hui.

Ce système a été mis sur pied par des centaines de programmeurs dans le monde entier. L'objectif était de créer un clone de UNIX, mais exempt de droits de licences et utilisable par tous.

Au commencement, Linux est un noyau de système d'exploitation qui a été créé comme passe-temps par Linus Torvalds. C'était un projet d'étudiant qui n'était rattaché à aucune organisation ni aucun but philanthropique. Linus au début n'avait pas l'intention de créer un système d'exploitation libre. Il s'est avéré qu'il a choisi la licence GPL pour son noyau et que celui-ci a intégré le projet de la GNU Foundation dont les outils tel le compilateur gcc ont donné à Linux des possibilités et une ampleur mondiale. Linus s'intéressait beaucoup à Minix, un petit système Unix créé par un professeur d'architecture système néerlandais: Andrew S. Tanenbaum. Il a donc décidé de développer un système capable de dépasser les normes de Minix.

La version complète actuelle est la 2.4.19 (sortie en août 2002); le développement est toujours en cours. Linux est développé sous la licence GPL de la GNU et son code source est librement disponible à tout un chacun. Cependant, cela ne signifie pas que GNU/Linux et ses distributions sont gratuites. Les sociétés peuvent demander de l'argent pour fournir GNU/Linux et des logiciels libres aussi longtemps que le code source demeure disponible.

GNU/Linux peut être utilisé dans une grande variété d'objectifs comprenant la gestion de réseau, le développement de logiciels, et comme plate-forme pour les utilisateurs. GNU/Linux est souvent considéré comme une excellente alternative, peu coûteuse par rapport à d'autres systèmes d'exploitation plus chers. En raison de la nature même de la fonctionnalité de GNU/Linux et de sa disponibilité, il est devenu mondialement connu et un grand nombre de programmeurs de logiciels ont téléchargé le code source de Linux et l'ont adapté pour satisfaire leurs besoins individuels.

Linux a une mascotte officielle, le pingouin Tux, qui a été choisie par Linus Torvalds pour représenter l'image qu'il associe au système d'exploitation qu'il a créé.

## 1.4 Le concept de logiciel libre

L'immense majorité des logiciels vendus dans le commerce par les éditeurs comme Microsoft, Lotus, Oracle ou SAP, sont distribués en version « exécutable », alors que les logiciels libres sont fournis avec leur « code source ».

Source, exécutable? Une petite explication de ces termes s'impose pour éclairer un peu plus sur les principales différences entre logiciels propriétaires et logiciels libres quant au fond. On peut considérer le code source d'un logiciel comme la partition de musique de celui-ci, et le code exécutable comme la même musique en version enregistrée. Une partition peut être lue par n'importe qui, modifiée par n'importe qui souhaitant étoffer ou améliorer la mélodie. Mais par contre, avec une version gravée, il n'est possible que d'écouter le résultat obtenu sans pouvoir modifier quoi que ce soit.

Le passage d'une version à l'autre s'opère grâce à une « compilation » qui traduit le code source (des lignes écrites en usant des langages de programmation maîtrisés par les programmeurs) en code exécutable (suite binaire uniquement compréhensible par l'ordinateur). Donc, avant d'arriver dans le commerce ou dans l'entreprise, une version en code source d'un logiciel est toujours construite par l'éditeur. Les logiciels libres sont livrés directement sous cette forme (à charge pour les utilisateurs de procéder eux-mêmes à la compilation ou de la faire réaliser par des intermédiaires), tandis que les éditeurs de logiciels propriétaires vendent uniquement le code « exécutable », le reste étant considéré comme secret industriel et non divulgué. Se définissant par opposition au logiciel « propriétaire », le logiciel « libre » a pour but de préserver la liberté des utilisateurs d'exécuter, de distribuer, d'étudier, de modifier et d'améliorer le logiciel au travers de son code source, l'ensemble des lignes de code qui, une fois compilées en langage machine, donneront le logiciel fini. Ces logiciels sont d'ailleurs généralement distribués sous forme de lignes de code que l'utilisateur configurera selon ses attentes puis compilera au moment de l'installation.

Il existe 4 niveaux de libertés que chaque logiciel qui se dit libre doit respecter<sup>3</sup>:

- la liberté d'exécuter le programme pour tous les usages, ce qui est le degré minimum de liberté que l'on peut attendre d'un logiciel;
- la liberté d'étudier comment le programme fonctionne et de l'adapter à ses besoins;
- la liberté de redistribuer des copies sans aucune contrainte;
- la liberté d'améliorer le programme et de diffuser ses améliorations publiquement de telle sorte que la communauté tout entière en bénéficie.

La liberté de modifier et de distribuer des logiciels libres sans payer de droits d'auteur est fondée sur une licence. Celle-ci représente un contrat de type copyright (comme n'importe quelle licence propriétaire) mais qui confère ces avantages ou plutôt ces droits à l'utilisateur. Les licences les plus connues sont le contrat GPL (GNU Public licence) et la licence de type BSD (Berkeley).

Pour comprendre la légitimité du logiciel libre, il est nécessaire d'étudier les différentes licences sur lesquelles se base l'économie logicielle dans son ensemble.

Ensuite, la définition d'Open Source est apparue, elle a pour but de compléter celle qui existait avant. Cette nouvelle définition est équivalente à celle de « Free software » et peut être traduite en français par logiciel libre aussi, mais à l'inverse de la définition de la FSF Foundation (basée sur les libertés), celle-ci est plus orientée sur la vision économique. En effet, tel que le montre son explication sur le site<sup>4</sup>, celle-ci décrit précisément ce qui peut se faire et les conditions à respecter. Personnellement, je la trouve meilleure car elle présente les conditions en termes de distributions et de conséquences plutôt que en terme de libertés ce qui n'est pas très représentatif ni même compréhensible pour une personne étrangère à l'idée de libertés prônée par la FSF.

Certaines personnes désirent ardemment différencier les deux définitions pour des raisons de vision personnelle des logiciels libres, mais dans ce mémoire, j'utiliserai l'une ou l'autre de ces appellations sans distinction.

---

3 <http://www.fsf.org/philosophy/free-sw.fr.html>

4 <http://www.opensource.org/docs/osd-francais.php>

## 1.5 Quelques caractéristiques

Je vais définir les caractéristiques des logiciels libres en montrant les deux grandes oppositions par rapport aux logiciels propriétaires.

1. Le droit primordial est celui du consommateur/utilisateur par opposition à celui du fournisseur. Ce qui est l'inverse du modèle propriétaire.

En effet, les améliorations font suite aux commentaires, problèmes, ajouts ou même intérêt des consommateurs. C'est donc un schéma réactif et non pro-actif comme dans le cas des logiciels propriétaires qui sont conçus en circuit fermé et donc laissent moins de place aux idées et conseils des utilisateurs. Souvent avec les logiciels propriétaires, le client doit se plier aux évolutions et au bon vouloir de l'éditeur qui peut changer fréquemment la présentation et les habitudes pour appeler des fonctions (localisation dans les menus, reclassement dans d'autres catégories, ...). Les logiciels libres offrent une plus grande continuité concernant l'utilisation des fonctions.

2. On a le droit de faire ce que l'on veut avec l'application, de l'adapter et de la redistribuer.

## 1.6 Les licences d'utilisation

Dans le cadre juridique actuel, la licence sous laquelle un programme est distribué définit exactement les droits détenus par ses utilisateurs. Par exemple, dans la plupart des programmes propriétaires, la licence retire les droits de copier, de modifier, de prêter, de louer, d'utiliser sur plusieurs machines, etc. En fait, les licences indiquent habituellement que le propriétaire du programme est la compagnie qui l'édite, et que celle-ci ne vend que les droits restrictifs d'utilisation du logiciel.

### 1.6.1 Les licences libres

Dans le monde du logiciel libre, la licence sous laquelle un programme est distribué a également une importance primordiale. Il est à noter que même dans le cas des licences libres, l'auteur du programme reste propriétaire. Seule la licence est concédée. Habituellement, les conditions indiquées dans les licences du logiciel libre sont le résultat d'un compromis entre plusieurs objectifs qui sont, dans un certain sens, contradictoires. Parmi ces conditions, on peut citer:

- garantir quelques libertés de base (redistribution, modification, utilisation) aux utilisateurs;
- assurer quelques conditions imposées par les auteurs (citation de l'auteur dans les travaux dérivés, par exemple);
- garantir que les travaux dérivés seront également des logiciels libres.

Les auteurs peuvent choisir de protéger leur logiciel avec différentes licences selon le degré avec lequel ils veulent accomplir ces buts, et les détails qu'ils veulent assurer. En fait, les auteurs peuvent (s'ils le désirent) distribuer leur logiciel sous différentes licences par différents canaux (et différents prix). Donc, l'auteur d'un programme choisit habituellement très soigneusement la licence sous laquelle il sera distribué. Et les utilisateurs, particulièrement ceux qui redistribuent ou modifient le logiciel, doivent soigneusement étudier sa licence.

Heureusement, bien que chaque auteur puisse utiliser une licence différente pour ses programmes, le fait est que presque tous les logiciels libres utilisent une des licences habituelles (GPL, LGPL, artistique, BSD-like, MPL, etc.), parfois avec de légères variations. Pour encore simplifier les choses, des organismes apparus récemment ont défini quelles étaient les caractéristiques qu'une licence de logiciel devait posséder si elle voulait être qualifiée de licence pour logiciels libres.

Le GNU Project donne une définition du logiciel libre qui fait souvent foi pour départager les vraies et les fausses licences libres.



Les différences entre les licences libres se situent habituellement dans l'importance que l'auteur donne aux points suivants:

- protection du caractère ouvert: quelques licences exigent en effet que le redistributeur maintienne la même licence que celle d'origine, et par conséquent, les droits du destinataire sont les mêmes (et cela que le logiciel soit reçu directement de l'auteur ou de n'importe quelle partie intermédiaire).
- protection des droits moraux: dans beaucoup de pays, la législation protège certains droits moraux, comme la reconnaissance de l'auteur. Certaines licences assurent également la protection à ce niveau là, ce qui permet de contourner certaines législations locales.
- protection de certains droits de propriété: dans certains cas, l'auteur original (celui qui a initialement créé le logiciel) a quelques droits supplémentaires, qui dans un certain sens sont un genre de droits de propriété.
- compatibilité avec des licences propriétaires: quelques licences sont conçues de sorte qu'elles soient complètement incompatibles avec le logiciel propriétaire. Par exemple, elles peuvent interdire de redistribuer n'importe quel logiciel étant le résultat d'un mélange de logiciel couvert par la licence et d'un logiciel propriétaire.
- compatibilité avec d'autres licences libres: quelques licences libres ne sont pas compatibles l'une l'autre, parce que les conditions de l'une ne peuvent pas être remplies si les conditions imposées par l'autre sont satisfaites. Dans ce cas-ci, il est habituellement impossible de mélanger des parties de code provenant des logiciels dont les licences sont incompatibles.

Voici les licences les plus connues et utilisées dont s'inspire la majorité des autres licences:

- BSD (Berkeley Software Distribution). La licence BSD couvre, entre autres, les versions des distributions de types \*BSD (Berkeley Software Distribution). C'est un bon exemple d'une licence « permissive » qui n'impose presque aucune limite à ce que l'utilisateur peut faire avec le logiciel, y compris faire payer les clients pour des distributions binaires, sans l'obligation d'inclure le code source. En résumé, les redistributeurs sont autorisés à faire presque n'importe quoi avec le logiciel, y compris l'utiliser pour des logiciels propriétaires. Les auteurs veulent seulement que leur travail soit identifié. Dans un certain sens, cette restriction assure aux auteurs une certaine quantité de marketing gratuit (car il ne coûte pas

d'argent). Il est important de noter que ce genre de licence n'inclut aucune restriction ayant pour but de garantir que les améliorations futures apportées au logiciel restent en source ouverte. Cette licence se trouve en annexe.

- GPL (GNU General Public License). La GPL a été soigneusement conçue pour favoriser la production d'un logiciel plus libre, et cela grâce à l'interdiction explicite de certaines actions sur le logiciel pouvant mener à l'intégration du logiciel en GPL dans des programmes propriétaires. La GPL est basée sur la législation internationale à propos du copyright, qui assure son applicabilité. Les caractéristiques principales de la GPL sont les suivantes:
  - elle permet la redistribution binaire, mais seulement si la disponibilité de code source est également garantie;
  - elle permet la redistribution des sources (et l'impose en cas de distribution binaire);
  - elle permet la modification sans restriction (si le travail dérivé est également couvert par GPL);
  - elle permet aussi l'intégration complète d'un autre logiciel mais cela uniquement s'il est également couvert par la GPL.

Note: La licence GPL se trouve également en annexe.

### **1.6.2 Etude d'une licence non-libre: la licence Sun**

Sun Microsystems a libéré certaines de ses technologies centrales de logiciel sous une nouvelle licence, la Sun Community Source License (SCSL). Il ne s'agit pas d'une licence libre, mais elle essaye d'imiter certaines des caractéristiques de cette dernière. En fait, les documents où Sun explique et fournit son raisonnement pour leur schéma de licence incluent les arguments habituels en faveur de la licence libre (bien qu'appliqués à la SCSL). Cependant, les points manquants (ceux qui feraient qu'elle serait vraiment une licence libre) ont (selon moi) pour conséquence, d'un point de vue pratique, que la plupart des avantages du logiciel open-source ne sont pas applicables aux logiciels qui sont couverts par la SCSL.

Certains des problèmes les plus connus (d'un point de vue open-source) avec la SCSL sont:

- l'obligation d'envoyer à Sun Microsystems toutes les modifications de code si elles doivent être distribuées. C'est Sun qui décidera si ces modifications sont redistribuables ou pas. Il n'est pas possible de créer même des variations « expérimentales » du code. Cette impossibilité est due aux clauses couvrant les tests de compatibilité qui doivent être passés.
- Sun utilise en fait deux licences pour son code. L'une d'elles, appliquée à la recherche et au développement interne, n'exige aucun honoraire. Mais l'autre, pour le déploiement commercial, exige que des honoraires leur soient payés. Le montant exact de ces honoraires sera décidé par Sun qui a donc une maîtrise totale sur le code développé par des contributeurs.
- pour les développeurs open-source (s'ils sont d'accord sur la SCSL et qu'ils téléchargent le code Sun), il est interdit d'utiliser le produit de leur développement ou une technologie semblable avec une licence open-source.

D'un point de vue stratégique, n'importe quelle société ou groupe de développement utilisant ou construisant sur le code couvert par la SCSL donne à Sun un contrôle important sur son développement et même sur ses plans de marketing (ce qui est habituellement considéré comme indésirable).

La communauté du logiciel libre a déjà réagi à cette licence avec le lancement de plusieurs projets visant à recréer la technologie couverte par la SCSL dans un environnement n'utilisant pas de code source Sun.

En général, pour un projet commercial, il sera beaucoup plus commode d'utiliser, par exemple, la GPL au lieu de la SCSL. Le logiciel résultant tirera bénéfice de tous les avantages qu'il pourrait avoir avec la SCSL, mais le développeur aura la pleine maîtrise du programme de développement, de la politique de distribution, et de la voie marketing.

## 1.7 L'importance de l'ouverture des sources

Le code source propriétaire, protégé par loi du copyright, joue un rôle central en permettant « la puissance architecturale de conception » du logiciel. Initialement, le copyright est apparu pour protéger et récompenser l'expression créatrice dans des medias imprimés et ainsi assurer sa diffusion publique. Mais le copyright est souvent utilisé avec abus comme outil légal pour limiter l'expression créatrice et contrôler les limites de la concurrence sur un marché donné. Le vendeur peut empêcher des utilisateurs de modifier les fonctionnalités d'un produit de logiciel, par exemple, ou limiter son interopérabilité avec les autres logiciels et matériels. Le vendeur peut empêcher des utilisateurs de personnaliser le produit pour convenir à leurs propres besoins. Le commerçant peut ignorer les anomalies connues et d'autres imperfections de produit, et ainsi forcer les utilisateurs à accepter les caractéristiques de conception non désirées.

Historiquement, les développeurs de logiciels ont cherché à préserver ces prérogatives (et ainsi leurs avantages relatifs par rapport aux concurrents et aux consommateurs) en protégeant vigoureusement leur code source, le modèle interne d'un produit de logiciel servant de base au code binaire ou « binaries » (la manifestation visible par l'utilisateur d'un produit de logiciel qu'un ordinateur exécute). Aussi longtemps que le code source propriétaire peut être tenu comme secret commercial protégé par la loi de copyright, les vendeurs gardent une plus grande mainmise que les utilisateurs sur les limites de la conception, le contrôle de la qualité, l'innovation et l'utilisation des logiciels.

Le contrôle important du code source au point de vue propriétaire peut être, à certains égards, un contrôle diminuant la souveraineté, la responsabilité et le choix du consommateur. Dans beaucoup de cas, il a pour conséquence l'anti-compétitivité. Cela est particulièrement pernicieux vu que, sur les marchés à évolutions rapides (concernant les produits techniquement complexes), ni les consommateurs ni les régulateurs anti-trust ne sont vraiment équipés pour défier de tels abus.

Le mouvement créé par le logiciel libre offre une base radicalement nouvelle pour la concurrence, l'innovation et l'importance du consommateur dans le marché. Propulsé par des forces convergentes, ce mouvement a été catapulté sur l'avant-scène de l'actualité au cours de l'année 2001. Ce modèle « user-driven », créé par une vaste communauté d'informaticiens de par le monde, rivalise avec le logiciel propriétaire au niveau de la qualité, de la fiabilité, de la flexibilité et du prix. Il appartient de façon effective à la communauté des utilisateurs et leur offre donc une base bon marché, souple et

durable pour le développement de solutions pour répondre à leurs besoins.

De même que le mouvement coopératif a permis aux ouvriers et aux consommateurs d'affirmer un plus grand contrôle de leurs vies économiques et personnelles, le nouveau mouvement des logiciels libres ouvre de nouvelles voies d'autodétermination. Cette approche permet de passer outre les inefficacités de coût, les barrières à l'innovation, la manipulation du consommateur et les rigidités de conception qui caractérisent beaucoup de marchés de logiciel propriétaire.

## **1.8 Le cas du logiciel libre dans l'Administration française**

La signification des logiciels libres pour l'Administration Publique est assez particulière.

Le logiciel, en terme générique, représente une matière singulière et non un produit au sens usuel du terme: la valeur de ce logiciel augmente avec l'usage et il est possible de le partager sans perte.

Le logiciel libre permet de développer un marché local de services basés exclusivement sur des compétences humaines en lieu et place de dépenses sur des licences.

Le logiciel libre (communément appelé LL ou OSS/FS en anglais) supporte des objectifs de service public et d'intérêt général:

- accès public par tous;
- meilleur usage de l'argent dépensé;
- indépendance par rapport aux vendeurs;
- réutilisation et mutualisation des ressources.

Mr Thierry Carcenac, député du Tarn, a rédigé un rapport à destination du Premier Ministre français sur les méthodes et les moyens de rendre l'Administration française électronique et citoyenne.

Il y développe différents points de l'organisation que pourrait adopter cette administration dont celui des logiciels libres.

Pour un certain nombre de tâches, les logiciels libres se sont révélés fiables, performants, sécurisés et financièrement compétitifs. Dès lors, comment encourager la prise en compte de ce nouveau modèle économique dans l'administration?

Un logiciel libre garantit quatre libertés:

1. celle d'étudier le fonctionnement du programme et de l'adapter à ses besoins;
2. celle de le copier et de le diffuser (ce qui est interdit avec les programmes commerciaux);
3. celle de l'améliorer pour en faire profiter la communauté;
4. celle de placer sous licence « libre » l'ensemble des développements réalisés par ou pour les administrations.

A la suite d'une analyse détaillée des postes de coûts et de gains, le recours accru aux logiciels libres par les administrations semble souhaitable selon lui.

| Type de coût                            | Coût du libre                            | Commentaires   |
|---|--|--|
| Installation                            | Plus cher qu'un logiciel propriétaire.   | La multiplicité des variantes rend la chose plus complexe. |
| Insertion dans le Système d'Information | Moins cher.                              | Le respect des standards ouverts assure un coût moindre.   |
| Matériel                                | Moins cher.                              | Les logiciels libres se contentent d'un matériel moindre.  |
| Licences                                | Moindre.                                 | Coût quasiment peu près nul.                               |
| Sécurisation                            | Moins cher.                              | Le mode de développement assure la sécurité.               |
| Formation                               | Moins cher sauf les coûts de transition. |  |

| Type de coût | Coût du libre | Commentaires  |
|--------------|---------------|---|
| Assistance   | Plus cher.    | En cours de constitution, donc onéreuse actuellement. |
| Evolution    | Moins cher.   | Evolution indépendante des logiciels.                 |
| Archivage    | Moins cher.   |   |

En effet, le caractère ouvert et public du code source des logiciels libres permet d'en améliorer la pérennité et la sécurité. Ces logiciels respectent particulièrement bien les standards d'Internet vu qu'il n'y a personne pour les contrôler, ni pour en tirer un avantage financier. Le caractère coopératif du développement permet de nombreux tests et garantit ainsi la robustesse des logiciels conçus. De surcroît, tout le code étant public, revu et corrigé par de nombreuses personnes appartenant à diverses organisations, il est particulièrement difficile d'y introduire une porte dérobée permettant, par exemple, de s'introduire dans la machine faisant fonctionner le logiciel.

Ces arguments ne conduisent naturellement pas à un acte législatif. En revanche, Mr Carcenac (député du Tarn), souhaite affirmer au plus haut niveau de l'État l'obligation, pour l'Administration, d'utiliser des standards de communication ouverts aussi bien au niveau de la sauvegarde des données que de leur transport.

Voici certaines de ses propositions...

- placer sous licence libre l'ensemble des développements réalisés par ou pour les administrations et les rendre disponibles sur le Web;
- publier sur le Web l'ensemble des cahiers des charges (CCTP) utilisés par les administrations dans les réalisations informatiques par des prestataires externes;
- aborder de manière systématique la question des standards ouverts et des logiciels libres lors des négociations des budgets informatiques des ministères.

## 1.9 Quel futur pour de tels logiciels?

### 1.9.1 De l'enrichissement des logiciels libres

La majorité des logiciels libres n'est pas développée dans un objectif commercial mais bien pour répondre aux besoins des utilisateurs. Les logiciels libres constituent depuis 1994 un modèle économique pour un certain nombre de sociétés éditrices de logiciels. Cette constatation peut paraître paradoxale vu l'ouverture et la gratuité de fait des logiciels libres, car la stratégie de ces acteurs n'est pas basée sur la vente de licences à l'inverse des éditeurs de logiciels propriétaires.

#### La vente de services avant tout

Ces acteurs majeurs du logiciel libre et de sa promotion commerciale existent depuis 1994 lorsque la société américaine Red Hat Software décida d'adapter et de distribuer une compilation des sources de GNU/Linux disponibles sur Internet. Depuis, la société Red Hat a été rejointe par de nombreuses autres sociétés dont les principales sont SuSE (Allemagne), Mandrake (France), Caldera, Turbolinux ou encore RedFlag Software pour la Chine. En fait, il existe une « infinité » de distributions plus ou moins dérivées de grands noms comme Red Hat et adaptées à un usage ou à un pays spécifique. Ces distributions respectent la GPL et se trouvent donc librement adaptables, modifiables et même revendables par tous. Il est tout à fait possible de récupérer une distribution existante, de lui ajouter quelques packages logiciels ainsi que quelques logos personnels, et de la revendre sous son propre nom. D'ailleurs, parallèlement à la vente de distributions sous forme de packs (CD-roms d'installation, manuel et aide en ligne), ces sociétés permettent toujours de télécharger gratuitement leurs versions les plus récentes. Ne possédant pas la propriété de leur vente, elles n'ont pas à en limiter la distribution. Au contraire, leurs sources de revenus sont principalement basées sur la vente de services associés où leur expertise peut être monnayée. Cette rémunération peut aller du soutien offert aux acheteurs des boîtes logiciels vendues en magasins (ceux-ci payent plus un service que les logiciels eux-mêmes), à la mise en place et au déploiement de solutions particulièrement volumineuses basées sur Linux dans les entreprises. Des acteurs comme SuSE, Red Hat ou Turbolinux ont conclu de nombreux accords de partenariats avec les plus grands fabricants de matériel informatique afin de permettre à ces derniers d'enrichir leur offre de solutions logiciels libres. Red Hat est par exemple lié à IBM, Compaq, Dell, HP, Intel et bien d'autres.



Si les perspectives de profit demeurent moins importantes qu'avec le système de copyright des logiciels propriétaires, toute société compétente peut proposer une offre complète de logiciels et de services identiques. Le besoin de services sur lequel se base leur stratégie et leur revenus est bien réel comme le rappelle Bob Young, cofondateur de RedHat:

« Il faut bien se rendre compte que Linux a commencé avec les « early adopters » , c'est à dire avec des étudiants ou des ingénieurs qui n'ont absolument pas besoin de support ou de services. C'est pourquoi beaucoup d'observateurs ont considéré que Linux était un marché où il n'y avait pas d'argent à gagner. Les choses changent aujourd'hui car ces «early adopters» ne sont que 10% des clients potentiels. Les autres 90% ont besoin de services et de support, ils le paieront si Linux répond à leurs besoins et permet d'améliorer le TCO (coût total de possession). »

(interview de Bob Young, co-fondateur de RedHat, au magazine Linux+, septembre 2000, p10-11)

### Un enrichissement collectif

Les logiciels libres sont, pour ces éditeurs, un moyen de s'affranchir de la dépendance envers d'autres sociétés et de pouvoir apporter librement leurs compétences. Aucun frein (comme le paiement de royalties ou encore le manque d'informations sur le fonctionnement du système) ne vient les subordonner aux décisions conceptuelles ou stratégiques d'autres acteurs. Cette liberté permet une compétition plus saine entre les éditeurs de logiciels. Le succès de ceux-ci ne dépend que de la valeur ajoutée (produite en raison de leurs améliorations qui iront par la suite enrichir la communauté du logiciel libre dont ils profiteront eux-mêmes en retour). Par exemple: la firme Red Hat est à l'origine du « Red Hat Package Manager » qui facilite l'installation de logiciels compilés (cf. format rpm). De nombreuses autres distributions ont repris cet outil à leur tour, popularisant le format développé par RedHat.

« On nous oppose également à Debian ou à Mandrake. C'est un non-sens. Plus Debian sera fort, plus nous le serons et vice versa. N'oubliez pas que nous sommes dans l'Open Source où les uns voient et critiquent le travail des autres »

(interview de Bob Young, co-fondateur de RedHat, au magazine Linux+, septembre 2000, p10-11)

### Des perspectives très larges

Ces sociétés cherchent également à se diversifier pour conquérir de nouveaux marchés dans lesquels le logiciel libre possède un potentiel certain. Par exemple, les Internet Appliances

(applications embarquées) ainsi que tous les types d'organiseurs personnels reliés à Internet et promis à un fort développement, pourront tirer avantage de GNU/Linux par ses faibles besoins en ressources, sa forte adaptabilité et l'absence de coûts de licences au niveau du prix de vente de ces produits. Les solutions commerciales basées sur les logiciels libres semblent donc tout à fait viables, même si elles obéissent à un modèle économique différent.

### Le rôle économique des logiciels libres

En plus de réduire la dépendance stratégique des entreprises, les logiciels libres peuvent également avoir un impact important sur l'économie et l'emploi, souvent bien supérieur à ceux des logiciels contrôlés par des acteurs propriétaires. Au même degré que les distributeurs de solutions basées sur GNU/Linux, les logiciels libres sont créateurs d'emplois décentralisés de services (PME). De plus, de nombreuses petites sociétés se créent dans la maintenance ou dans le développement personnalisé. En supprimant le coût des licences, le logiciel libre autorise une marge supplémentaire en termes d'adaptation, de réactivité et de fiabilité. La simple économie en licences non payées peut permettre de payer des ingénieurs pour adapter le logiciel libre aux besoins de l'entreprise mais aussi pour former les utilisateurs. Ainsi le logiciel libre (adapté sur place) est à l'origine d'une création d'emplois locaux supérieure à celle du logiciel propriétaire importé. Arkane Média<sup>5</sup>, société prestataire de services en logiciels libres basée à Strasbourg, a ainsi reçu le prix « Start-up de l'année » lors des Trophées de la Nouvelle Economie <sup>6</sup>.

Ce schéma reste valable au niveau national pour des pays dont les ressources financières sont limitées, mais qui disposent d'une main d'oeuvre bien formée ou pouvant l'être.

Le logiciel libre semble donc tout à fait viable comme modèle économique pour des entreprises éditrices de logiciels.

---

5 <http://www.arkane-media.net>

6 <http://www.jdnet.fr/lille/010131arkane.shtml>

## 1.9.2 Les limites de l'approche propriétaire

La production de logiciels du point de vue propriétaire obéit à des règles économiques très particulières. Les coûts se concentrent essentiellement sur la conception et ne concernent plus la production et la distribution que de façon marginale. Cette concentration des coûts aboutit à des phénomènes de concentration très importants. Ainsi, la tendance au monopole dans l'économie logicielle paraît bien plus élevée qu'ailleurs. Ces phénomènes ont pour effet logique d'accroître la dépendance des utilisateurs, mais aussi de ralentir le progrès technologique à long terme. Voyons en détail les règles et limites de la production logicielle propriétaire.

### L'importance majeure des investissements

Une caractéristique importante des biens immatériels (comme les logiciels) est le rapport élevé entre les coûts fixes d'investissement (plus humains que matériels) afin de créer le premier exemplaire d'un bien ainsi que le coût marginal de la production. Concevoir un logiciel complexe demandera d'importants investissements humains et coûtera beaucoup plus cher que de le diffuser par la suite. Avec, d'une part, la progression très importante du taux d'équipement informatique ces dernières années et, d'autre part, le développement d'Internet et des CD-ROM, le phénomène s'est exacerbé par la possibilité de communications massives, ultra-rapides et même interactives permettant une grande variété de protocoles et donc une diffusion à des coûts de plus en plus marginaux. Par conséquent, pour beaucoup de biens immatériels, nous ne pouvons même plus parler de coût de revient. En effet, le coût marginal de la production et de la distribution peut pratiquement être considéré comme nul (si l'on fait abstraction du service après-vente et des « hotlines » pouvant être indépendants de la production et de la distribution).

Cette situation conduit dans de nombreux cas à des phénomènes de concentration (allant jusqu'au monopole) qui entravent le bon fonctionnement des mécanismes du développement économique et technologique et qui conduisent à une gestion purement financière des ressources scientifiques, technologiques ou culturelles. Le tout s'accompagne de nombreux effets néfastes motivant le procès anti-trust contre Microsoft depuis 1998.

### De la concentration au contrôle des standards

Sans revenir sur les nombreux phénomènes qui soutiennent la tendance à la concentration et au monopole dans les secteurs industriels, l'un d'entre eux s'avère particulièrement influant au niveau

de l'industrie du logiciel en raison du rôle primordial qu'y jouent les mécanismes de modularité dans la constitution de systèmes très variables et très complexes.

En effet, dès qu'une plate-forme (x86, Mac, Sparc, ...) domine le marché, tout créateur commercial de produits (matériel ou logiciel) liés aux plates-formes s'adaptera de préférence à la plate-forme dominante pour des raisons évidentes de rentabilité. Il en est ainsi pour les fournisseurs de composants pour PC qui fournissent systématiquement les pilotes pour les plates-formes logicielles Microsoft et plus rarement pour les autres (certains modems sont même conçus pour ne fonctionner que sous Windows). Un autre exemple plus flagrant est celui des éditeurs de logiciels d'application qui, lorsque leur marché reste concurrentiel, n'ont souvent pas les moyens (ou parfois même l'intérêt économique) de développer pour un autre objectif que la plate-forme la plus répandue du marché. Cet effort renforce son caractère dominant vu que les autres plates-formes, possédant alors moins d'applications, présentent moins d'attrait. Certains outils comme Direct 3D pour les jeux permettent d'encourager le développement sur une seule plate-forme, même si d'autres alternatives facilement portables (comme OpenGL) existent.

Ce phénomène se renforce par la protection légale ou technique des interfaces rendant difficile, à cause d'un manque d'information, le développement par d'autres sociétés de logiciels équivalents compatibles sur les plates-formes ainsi négligées. Il est, par exemple, très difficile d'exploiter une encyclopédie prévue pour Windows sur une plate-forme Unix bien que cela ne devrait en principe poser aucun problème technique.

### Les effets de cette situation

Cette situation entrave généralement le progrès technologique à long terme. Une fois la concurrence disparue, le seul producteur restant n'a plus véritablement intérêt à investir pour améliorer ses produits. Le contrôle d'une technologie par une seule société implique un petit nombre de professionnels engagés dans l'amélioration de cette technologie. La recherche universitaire et l'enseignement sont alors entravés ou contrôlés par la rétention de l'information. De plus, la diversité réduite des développements limite considérablement les possibilités de progrès par une évolution concurrentielle à la manière des distributions GNU/Linux. Ainsi, la vulnérabilité générale du tissu technologique aux agressions augmente. (l'année dernière, quelques virus célèbres ont, par exemple, pu infecter un nombre très élevé de machines par l'exploitation d'une faille du logiciel de messagerie Outlook)

Les inconvénients de cette situation sont nombreux dans le cadre d'une exploitation en entreprise. Le fait de ne pouvoir disposer que d'un fournisseur unique en matière de solutions logicielles crée une situation de dépendance au niveau des prix et des services. Il en va de même pour la stratégie à long terme de l'entreprise pouvant dépendre des décisions de son unique fournisseur. Techniquement, la non disponibilité des codes sources (ou leur prix excessif) limite fortement ou même interdit aux sociétés clientes toute utilisation et tout service personnalisé quel que soit le niveau (la maintenance, la sécurisation, le portage sur de nouvelles plates-formes ou encore l'adaptation à des besoins spécifiques). En fait, la société cliente contrôle mal la qualité et la pérennité de son investissement.

## **Chapitre 2: Business modèles**

### **2.1 Quelques modèles**

Eric Raymond, un auteur phare du mouvement du logiciel libre, a écrit quelques essais (disponibles sur Internet) afin d'expliquer le phénomène et les avantages des logiciels libres. Dans une intention de meilleure intégration du contexte, vous trouverez ci-dessous un résumé de l'un de ces ouvrages (Le chaudron magique<sup>7</sup>) qui pose la problématique de l'ouverture du code du programme. Il s'agit en fait d'une façon détournée de montrer l'intérêt que les entreprises pourraient en retirer. M. Raymond insiste aussi sur les différents modèles économiques pour lesquels il propose des exemples au travers d'entreprises ayant ouvert leur code et ayant opté pour le « bon choix ».

#### **2.1.1 Les modèles économiques financés par la valeur d'utilisation**

Il est nécessaire de remarquer (dans la distinction entre la valeur d'utilisation et le prix d'acquisition) la menace planant sur le prix d'acquisition par l'ouverture des sources. Par contre, la valeur d'utilisation demeure identique.

Si la valeur d'utilisation pilote le développement de logiciels plus que le prix d'acquisition, et si le développement à sources ouvertes est effectivement plus efficace et plus direct que le développement à sources fermées, alors nous devons nous attendre à découvrir des circonstances pour lesquelles la valeur d'utilisation attendue finance seule le développement à sources ouvertes de manière durable.

Grâce à cela, nous pouvons identifier au moins deux modèles importants où seule la valeur d'utilisation finance les salaires des développeurs à plein temps sur des projets à sources ouvertes.

---

7 [http://www.linux-france.org/article/these/magic-cauldron/magic-cauldron-fr\\_monoblock.html](http://www.linux-france.org/article/these/magic-cauldron/magic-cauldron-fr_monoblock.html)

## 2.1.2 Les modèles exploitant une valeur d'acquisition indirecte

Il existe deux manières de gagner des marchés de services liés au logiciel qui capturent ce qu'on pourrait appeler une valeur d'acquisition indirecte. Dans ce domaine, nous pouvons dénombrer cinq modèles connus et deux modèles pressentis (Note: d'autres pourront être développés à l'avenir).

### 1. Vendre à perte pour se positionner sur un marché

Dans ce modèle, des logiciels aux sources ouvertes sont utilisés pour créer ou pour maintenir une position stable dans un marché où un logiciel propriétaire engendre un flux financier direct.

Dans la variante la plus fréquente, le logiciel client, aux sources ouvertes, favorise la vente d'un logiciel serveur ou des revenus pour les inscriptions et la publicité associée à un site portail.

Commentaire: en ouvrant le code source de leur navigateur, qui remportait encore les faveurs de beaucoup, la société Netscape a soudainement ôté à Microsoft la possibilité de contrôler entièrement le marché des navigateurs. Netscape s'attendait à une collaboration dans le cadre d'un projet de développement aux sources ouvertes (« Mozilla ») qui accélérerait la mise au point et le « débogage » de leur navigateur. De plus, il espéraient que l'IE (Internet Explorer) de Microsoft serait réduit à les suivre à la trace, définitivement empêché de redéfinir seul la norme HTML.

Cette stratégie a fonctionné. En novembre 1998, la société Netscape commença à reprendre des parts de marché à IE. Quand, au début de l'année 1999, la société AOL a racheté Netscape, l'avantage sur la concurrence conféré par le projet Mozilla était si évident que l'un des premiers engagements publics d'AOL fut de continuer à encourager le projet Mozilla, bien qu'il en soit encore au stade alpha.

### 2. Gel des gadgets

Ce modèle concerne les fabricants de matériel. Dans ce contexte, nous entendons par matériel toute l'électronique, des cartes Ethernet ou autres cartes périphériques aux systèmes informatiques complets. Les pressions du marché ont forcé les sociétés de matériel à écrire et à maintenir du logiciel (des pilotes de périphériques aux

systèmes d'exploitation complets en passant par les outils de configuration). Toutefois, le logiciel n'est pas réellement source de bénéfices. Il s'agit d'un surcoût rarement négligeable.

L'effet à long terme de l'ouverture des sources est particulièrement important dans le cadre du « gel des gadgets ». Une assistance technique est produite et fournie pour les produits matériels durant une période finie après laquelle les consommateurs sont livrés à eux-mêmes. Mais, s'ils ont accès au code-source du pilote et s'ils peuvent le corriger selon leurs besoins, il est envisageable qu'ils seront heureux de choisir la même société pour leurs futurs achats.

Un exemple marquant du gel des gadgets fut la décision de la société Apple Computer, mi-mars 1999, d'ouvrir les sources de Darwin, le coeur de leur système d'exploitation MacOS X.

### 3. Donner la recette et ouvrir un restaurant

Dans ce modèle, les sources d'un logiciel sont ouvertes pour permettre l'ouverture d'un marché, non pas pour un logiciel fermé mais bien pour des services.

Il s'agit de la démarche de la société Red Hat et d'autres distributions GNU/Linux. Ils ne vendent pas le logiciel (les bits) lui-même, mais bien la valeur ajoutée par l'assemblage, les tests d'un système d'exploitation garanti (ne serait-ce qu'implicitement) comme étant de qualité commerciale et compatible avec d'autres systèmes d'exploitation portant la même marque. D'autres éléments de leur proposition permettent une assistance technique gratuite à l'installation et la fourniture d'options pour des contrats d'assistance étendus.

Un cas de figure récent et très instructif fut celui de Digital Creations, société de conception de sites Web créée en 1998 et spécialisée dans les bases de données complexes et les sites de transactions. Leur outil principal est un logiciel de publication objet désormais appelé Zope qui a connu différents noms de par le passé.

Un autre exemple est celui de la société E-smith, Inc qui vend des contrats d'assistance technique pour un logiciel serveur clef en main pour Internet à source ouverte, un GNU/Linux personnalisé. L'une des responsables, décrivant la masse de



téléchargements de leur logiciel sur leur site, déclare « La plupart des sociétés parleraient de piratage de logiciel. Nous considérons qu'il s'agit de libre marketing.»

#### 4. Accessoires

Dans ce modèle, des accessoires pour du logiciel Open Source sont vendus couvrant toute la gamme, des objets (comme les tasses et les t-shirts) aux documentations éditées et produites par des professionnels. La firme O'Reilly & Associates, editrice de nombreux ouvrages de référence sur le logiciel Open Source, fournit un bon exemple de société en accessoires. En réalité, la société O'Reilly emploie et assiste financièrement des hackers bien connus dans le monde de l'Open Source (comme Larry Wall et Brian Behlendorf) afin de se construire une réputation dans le marché qu'elle s'est choisi.

#### 5. Libérer l'avenir, vendre le présent

Dans ce modèle, le logiciel est publié sous forme de binaires et les sources sous une licence fermée avec une date limite à laquelle les restrictions prendront fin. Vous pouvez, par exemple, écrire une licence autorisant la libre redistribution, interdisant l'utilisation dans un but commercial sans acquittement d'obole, et garantissant que le logiciel sera diffusé selon les termes de la GPL un an après sa sortie ou en cas de banqueroute du vendeur.

Le principal inconvénient de ce modèle est que les restrictions et le caractère fermé des premières versions de la licence ont tendance à inhiber la revue par les pairs et la participation aux premiers stades du cycle du produit, là où précisément ils sont le plus requis.

#### 6. Libérer le logiciel, vendre la marque

Un modèle économique encore à l'état de théorie. Les sources d'une gamme de logiciels sont ouvertes, une batterie de tests ou un ensemble de critères de compatibilité sont mis en place, et une marque certifiant que leur implémentation de ces techniques est compatible avec tous ceux qui font état de cette même marque est vendue aux utilisateurs. (voici la manière dont la société Sun Microsystems devrait gérer Java et Jini)

### 7. Libérer le logiciel, vendre le contenu

Un modèle économique théorique. Imaginez un service d'inscription dans le style d'un centre d'informations. La valeur ne réside ni dans le logiciel client ni dans le serveur, mais bien dans la fourniture d'une information fiable et objective. Ainsi, le logiciel est libéré et des inscriptions au contenu sont vendues. Dans la continuité du travail des hackers pour assurer le portage vers de nouvelles plates-formes et pour améliorer le logiciel de diverses manières, votre marché se développera en conséquence. (voilà la raison pour laquelle la société AOL devrait libérer les sources de son logiciel client)

## 2.2 Quelques exemples

### 2.2.1 Le cas Apache: partage des coûts

Supposons que vous travailliez pour une société présentant le besoin critique d'un serveur Web de haute fiabilité et capable de traiter des volumes conséquents. Votre serveur devrait alors fonctionner 24 heures sur 24, 7 jours sur 7, mais aussi être rapide et personnalisable.

Comment allez-vous procéder pour arriver à vos fins? Vous pouvez choisir l'une des trois stratégies suivantes:

- acheter un serveur Web propriétaire. Dans ce cas, vous faites le pari que le cahier des charges du vendeur correspondra au vôtre et que ce dernier aura la compétence technique pour implémenter le serveur correctement. Même en ces deux assertions, il est probable que la personnalisation du produit montre ses limites. Vous ne pourrez le modifier qu'à travers les points d'accès fournis par le vendeur. Ce choix du serveur Web propriétaire n'est pas le plus séduisant (pour la plupart des décideurs).
- développer votre propre serveur. Il ne s'agit pas d'une option à écarter a priori. Les serveurs Web ne sont pas très compliqués (à l'inverse des navigateurs) et un serveur spécialisé peut très bien être incomplet et humble. En suivant cette voie, vous

obtiendrez exactement les fonctionnalités et la personnalisation souhaitées, mais il vous faudra payer ces avantages en temps de développement. Votre société aura peut-être des problèmes le jour où vous la quitterez (pour la retraite ou pour un autre emploi).

- rejoindre le groupe Apache. Le serveur Apache a été mis au point par un groupe de webmasters reliés par Internet, qui ont compris la nécessité d'unir leurs efforts et d'améliorer un code commun plutôt que d'engager un grand nombre d'efforts de développement parallèles. De cette manière, ces personnes ont pu réunir à la fois les avantages du développement d'un serveur personnel et le puissant effet de « débogage » grâce à la revue par un grand nombre de pairs travaillant en parallèle.

Les avantages du choix d'Apache sont certains. Nous pouvons en mesurer l'impact en lisant tous les mois le sondage du site Netcraft<sup>8</sup>, montrant depuis sa mise en place une progression inéluctable du serveur Apache au détriment de tous les serveurs propriétaires. En juin 1999, ce serveur et ses dérivés dirigeaient 61% des serveurs Web de par le monde. Sans propriétaire légal, sans promotion et sans société de services offrant des contrats de maintenance.

Nous pouvons généraliser l'histoire d'Apache à un modèle où les utilisateurs de logiciels trouvent leurs avantages dans le financement d'un développement à sources ouvertes. Cette approche leur fournit un produit meilleur à un prix plus faible.

## 2.2.2 Le cas Cisco: étaler les risques

Voici quelques années, deux programmeurs de la société Cisco ont reçu pour mission d'écrire un système de files d'impression à l'intention du réseau d'entreprise de leur société.

Le duo a proposé des modifications astucieuses au logiciel standard de files d'impression sous Unix ainsi que quelques scripts d'interface s'acquittant de la tâche. Toutefois, un petit problème est apparu.

---

<sup>8</sup> <http://www.netcraft.net>

En effet, aucun de ces deux programmeurs n'allait rester employé éternellement dans cette entreprise.

Voilà pourquoi ils ont rencontré leur patron et l'ont convaincu d'autoriser la publication du logiciel de files d'impression sous une licence à sources ouvertes. Ils ont fondé leur argumentation sur le fait que la société Cisco n'y perdrait rien en termes de prix d'acquisition et avait donc tout à y gagner.

En encourageant la croissance d'une communauté d'utilisateurs et de co-développeurs issus de nombreuses sociétés, la société Cisco pouvait se prémunir efficacement contre la perte des développeurs originaux du logiciel.

Il est possible de généraliser l'histoire de Cisco à un modèle dans lequel les sources ouvertes n'ont pas plus l'intention de réduire les coûts que d'étaler les risques. Toutes les parties pensent que l'ouverture des sources et la présence d'une communauté qui collabore, suite à un financement par de nombreux flux indépendants, fournit une assurance économiquement appréciable et suffisante pour lui dégager des crédits.

### **2.2.3 Le cas Open-Cascade**

Open-CASCADE est une filiale de Eads Matra Datavision, fondée en janvier 2001. Cette équipe de 100 personnes (80 réalisateurs inclus) travaille en France et en Russie. Leur mission est de fournir des services ainsi que le support pour les utilisateurs, les éditeurs de logiciels et les chercheurs industriels pour leurs projets de développement basés sur le logiciel Open-CASCADE 3d (logiciel libre d'entreprise) chargé de modéliser des composants.

Les compagnies industrielles obtiennent un avantage concurrentiel via les applications contenant leur savoir-faire spécifique. Dans beaucoup de cas, ce besoin spécifique n'est pas couvert par le logiciel commercial de CAO ou, s'il est couvert, est difficile ou cher à la mise en application.

Par contre, avec le logiciel Open-Cascade ...

- il leur est possible de développer leur propre application et de se fonder sur le support cost-effective de l'équipe d'Open-CASCADE;
- il est possible de faire appel aux experts d'Open-CASCADE pour le développement, le déploiement et l'entretien de leur application spécifique.

Dans les deux cas, la solution d'Open-CASCADE leur apporte:

- une intégration facile de leur logiciel avec des systèmes DAO (Dessin assisté par ordinateur) génériques de ce logiciel;
- tous les avantages d'un outil d'Open Source, comme:
  - la continuité de produit;
  - l'absence de contrainte de logiciel de boîte noire;
  - le prix réduit: aucun honoraire d'exécution, à l'exception des services dont ils ont besoin;
  - la possibilité de distribuer leur application sans contrainte d'autorisation;
  - l'accès au code source d'Open-CASCADE permettant un choix particulièrement large des possibilités de projet de recherche;
  - l'échange en ligne à [www.opencascade.org](http://www.opencascade.org) fournissant une plateforme idéale pour la diffusion des résultats de leurs développements;
  - des solutions rentables de support d'Open-CASCADE, ayant été conçues pour répondre aux besoins spécifiques de l'environnement de la recherche.

Plusieurs grandes entreprises industrielles et plusieurs entreprises de recherches ont accordé leur confiance à Open Cascade:

- BMW
- U.C. Berkeley
- Flamingo
- etc

### 2.2.4 Le cas du Bundestag <sup>9</sup>

Le Bundestag, la Chambre du Parlement allemand, analyse l'éventualité de se séparer du système d'exploitation Windows en faveur de GNU/Linux dans un souci partiel de sécurité. Une commission parlementaire va devoir décider quel système d'exploitation utiliser lorsque le moment de remplacer Windows NT sera venu (NT est l'OS actuellement en service sur les machines du Bundestag); selon Volker Schroer, un conseiller social-démocrate au Bundestag.

« Le problème de base est la fin du support NT de Microsoft à la fin de l'année 2002. Il est possible de continuer à l'utiliser pendant un certain temps, mais vous devrez l'upgrader ou alors changer d'OS (Operating System ou système d'exploitation.) » signale-t-il. « Les seules options à l'étude sont une autre version de Windows ou GNU/Linux. Le comité examine la proposition de commuter les ordinateurs du Bundestag (plus de 5000) vers GNU/Linux afin d'en améliorer (éventuellement) la stabilité et la sécurité, mais aussi pour réaliser une opération rentable. ».

D'après les études du comité, mettre GNU/Linux en application dans tout le gouvernement fédéral pourrait aboutir à une épargne de 125 millions d'euros. Parmi les arguments en faveur de GNU/Linux: des rapports selon lesquels certaines versions de Windows contiendraient des backdoors (trou de sécurité créé intentionnellement à des fins d'espionnage) conçues pour permettre un accès aux données des utilisateurs par la NSA (les services secrets américains).

Des rapports concernant certaines backdoors ont déjà été mentionnés dans les médias l'année passée. Selon ceux-ci, les militaires allemands éliminaient les logiciels Microsoft des ordinateurs dits sensibles. (rapports niés par la suite par le ministère de la défense; cf. « German army denies report on Microsoft ban »<sup>10</sup>)

---

9 [http://www.idg.net/ic\\_712902\\_1794\\_9-10000.html](http://www.idg.net/ic_712902_1794_9-10000.html)

10 <http://www.jdnet.fr/lille/010131arkane.shtml>

## **Chapitre 3: Les logiciels libres contre les logiciels propriétaires**

### **3.1 Situation actuelle du logiciel libre et Open Source en entreprises**

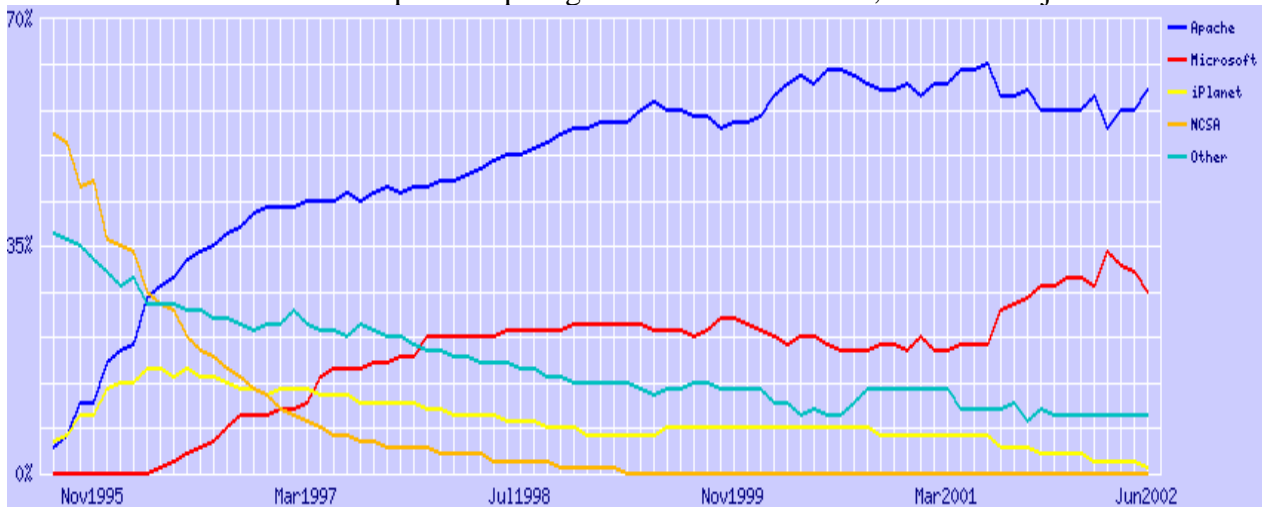
Beaucoup de gens pensent qu'un produit est gagnant uniquement s'il a une part de marché significative. Il s'agit d'une opinion très commune, mais il y a une raison à celle-ci: les produits ayant de grandes parts de marché obtiennent des applications, des utilisateurs qualifiés, et un élan qui réduit le risque futur. Certains auteurs arguent que OSS/FS ou GNU/Linux ne font pas partie du courant principal, mais vu leur utilisation répandue, ces considérations reflètent le passé et non le présent. Il existe d'excellents indicateurs montrant que l'OSS/FS possède des parts significatives dans de nombreux marchés:

1. Apache est le premier serveur Internet depuis avril 1996. *Netcraft's statistics on webservers*<sup>11</sup> a démontré qu'Apache domine le marché public des serveurs Internet depuis qu'il est devenu numéro un en avril 1996. Par exemple en juin 2002, Netcraft a recensé tous les sites Internet qu'il a pu trouver (38.807.788 sites) et a prouvé que Apache possède 59,67% du marché, Microsoft 28,96%, Zeus 2,06% et iPlanet (alias Netscape) 1,77%.

---

<sup>11</sup> <http://www.netcraft.net/survey>

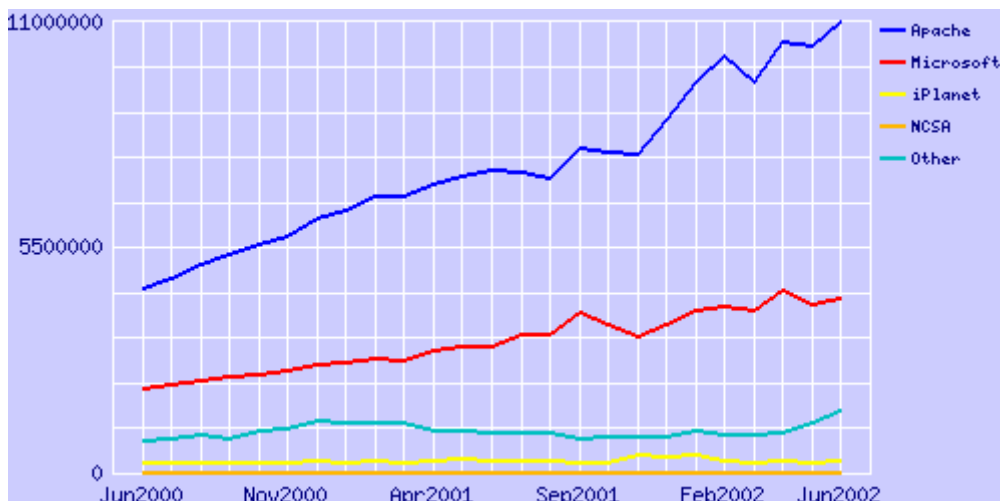
Part de marché pour les plus grands serveurs Internet, août 1995 - juin 2002



Sources: [www.netcraft.net](http://www.netcraft.net)

Plus récemment, Netcraft a essayé de décompter les sites « inactifs ». En effet, beaucoup de sites ont été ouverts uniquement pour « prendre de la place » mais ne sont pas activement utilisés. En comptant uniquement les sites « actifs », Apache s'améliore encore: en juin 2002, Apache occupait 64,42% du marché, Microsoft 24,93%, iPlanet 1,66%, et Zeus 1,34%.

Parts de marché pour les serveurs Internet actifs, juin 2000 - juin 2002



Sources: [www.netcraft.net](http://www.netcraft.net)



Le même résultat a été obtenu indépendamment par E-soft<sup>12</sup>. Leur rapport publié en avril 2002 a examiné 5.940.265 serveurs Internet et a constaté qu'Apache était premier (65,80%), Microsoft IIS (Internet Information Server) étant deuxième (25,30%). Evidemment ces résultats évoluent sans arrêt, les derniers chiffres d'études sont consultables chez Netcraft et E-soft.

- Deux études de Netcraft en 2001 (enquêtes de juin 2001 et de septembre 2001) ont constaté que GNU/Linux est le système d'exploitation numéro deux au niveau des serveurs Internet en comptant par adresse IP (et que ce système avait gagné de considérables parts de marché depuis février 1999). Chaque connexion à un réseau accessible au public est liée à une seule adresse Internet appelée adresse IP, et cette étude utilise les adresses IP (leur enquête de serveur Internet comptant par IP et non par nombre de machines physiques). Une précision: Netcraft date ses informations fondées sur les études sur les serveurs Internet (Note: je ne parle pas de la date de publication), et ils ne rendent compte que des informations datant d'un ou plusieurs mois. Ainsi, l'enquête datée de juin 2001 a été publiée en juillet et concerne les résultats d'études sur les systèmes d'exploitation de mars 2001, alors que l'enquête datée de septembre 2001 a été publiée en octobre et concerne les résultats de juin 2001. Voici les résultats d'enquête de Netcraft:

| <b>Système d'exploitation</b> | <b>Pourcentage (Mars)</b> | <b>Pourcentage (Juin)</b> | <b>Versions concernées</b>  |
|-------------------------------|---------------------------|---------------------------|---|
| <b>Windows</b>                | 49.2%                     | 49.6%                     | Windows 2000, NT4, NT3, Windows 95, Windows 98.                       |
| <b>GNU/Linux</b>              | 28.5%                     | 29.6%                     | GNU/Linux   |
| <b>Solaris</b>                | 7.6%                      | 7.1%                      | Solaris 2, Solaris 7, Solaris 8.                                      |
| <b>BSD</b>                    | 6.3%                      | 6.1%                      | BSDI BSD/OS, FreeBSD, NetBSD, OpenBSD.                                |
| <b>Autres Unix</b>            | 2.4%                      | 2.2%                      | AIX, Compaq Tru64, HP-UX, IRIX, SCO Unix, SunOS 4 et autres.          |
| <b>Autres non-Unix</b>        | 2.5%                      | 2.4%                      | MacOS, NetWare, Systèmes d'exploitations propriétaires d'IBM.         |
| <b>Inconnu</b>                | 3.6%                      | 3.0%                      | Non identifié par le détecteur de système d'exploitation de Netcraft. |

12 [http://www.securityspace.com/s\\_survey](http://www.securityspace.com/s_survey)

Tout dépend de ce que vous voulez mesurer bien entendu. Beaucoup de BSD (Free BSD, Net BSD, et Open BSD) sont aussi des OSS/FS. Une part d'au moins 6,1% devrait donc être ajoutée aux 29,6% de GNU/Linux pour déterminer le pourcentage de systèmes d'exploitation OSS/FS utilisés comme serveurs Internet. Il semble donc qu'approximativement un tiers des ordinateurs des serveurs Internet utilise un système d'exploitation OSS/FS. On retrouve aussi des différences régionales. Par exemple, nous pouvons constater que GNU/Linux arrive devant Windows en Allemagne, Hongrie, République Tchèque et en Pologne.

Les sites Internet connus utilisant GNU/Linux incluent ceux de la Maison Blanche, Google et Rackspace. D'autres sites web utilisent également des systèmes d'exploitation OSS/FS comme Yahoo et Sony Japan qui utilisent Free BSD.

En rassemblant tous les systèmes Unix ou similaires on arrive à un total de 44,8% (comparés aux 49,2% de Windows) en mars 2001.

Comme exposé par Netcraft, Apache fonctionne avec des systèmes Unix variés et fait fonctionner plus de sites que Windows. Il est fortement développé comme hôte des compagnies et ISP qui s'efforcent de faire fonctionner un maximum de sites sur un seul ordinateur afin d'en réduire les coûts.

3. GNU/Linux est le système d'exploitation numéro un des serveurs Internet (en comptant par noms de domaine) selon une étude de 1999 portant principalement sur les sites européens et les sites d'éducation. La première étude que j'ai découverte et qui examinait la pénétration du marché par GNU/Linux est une enquête faite par Zoebelin<sup>13</sup> en avril 1999. Cette étude montre que, du total des serveurs Internet en 1999 [exécutant au moins ftp, news, ou http (www)] se trouvant dans une base de données de noms qu'ils utilisaient, le système d'exploitation numéro un était GNU/Linux (à 28,5%). Il est important de noter que cette étude, la première que j'aie découverte pour essayer de répondre aux questions concernant les parts de marché, utilisait des bases de données existantes sur les serveurs de « .edu » (domaine de l'éducation) et une base de données RIPE (qui recouvre l'Europe, le Moyen Orient, une partie de l'Asie et de l'Afrique), ce qui veut dire qu'il ne s'agit pas d'une étude prenant compte « l'entièreté d'Internet » (par exemple elle omet « .com » et « .net »). Cette

---

13 <http://www.leb.net/hzo/ioscount>

étude est réalisée par nom de domaine (le mot que vous tapez dans un navigateur pour trouver un site) plutôt que par adresse IP. Ce qui implique que le décompte diffère de celui de Netcraft, pour son étude sur les systèmes d'exploitation de juin 2001. Notons que cette étude a pris en compte les serveurs fournissant des services ftp et news (n'étant pas uniquement des serveurs Internet).

Voici les différents systèmes repris dans l'étude:

| Part de marché   | Système d'exploitation | Versions  |
|------------------|------------------------|---|
| <b>GNU/Linux</b> | 28.5%                  | GNU/Linux   |
| <b>Windows</b>   | 24.4%                  | Toutes versions combinées (incluant 95, 98, NT).  |
| <b>Sun</b>       | 17.7%                  | Sun Solaris ou SunOS.                             |
| <b>BSD</b>       | 15.0%                  | BSD Family (FreeBSD, NetBSD, OpenBSD, BSDI, ...). |
| <b>IRIX</b>      | 5.3%                   | SGI IRIX  |

Une partie de la famille BSD est OSS/FS aboutissant à un total du système d'exploitation OSS/FS encore plus élevé. Si plus des 2/3 des BSD sont OSS/FS, alors la part d'OSS/FS devrait être d'environ 40%. Les partisans des systèmes Unix ou similaires noteront que la majorité (environ 66%) fonctionnaient avec de tels systèmes, alors que seulement 24% agissaient avec une variante de Microsoft Windows.

4. GNU/Linux est le deuxième système d'exploitation des serveurs vendu en 1999 et 2000, et présente la croissance la plus forte. Selon une étude IDC de juin 2000<sup>14</sup> sur les licences de 1999, 24% du total des serveurs installés en 1999 (en comptant les services Internet et Intranet) fonctionnaient sous GNU/Linux. Windows NT arrivait premier avec 36%. Tous les systèmes Unix mis ensemble totalisaient 15%. Une nouvelle fois, depuis que certains systèmes Unix sont devenus des systèmes OSS/FS (par exemple Free BSD, Open BSD, et Net BSD), le nombre de systèmes OSS/FS a dépassé celui de GNU/Linux. Notez que tout dépend de ce que vous désirez calculer. 39% de tous les serveurs installés depuis cette étude étaient des systèmes Unix ou similaires (totalisant 24% + 15%), ce qui fait des serveurs utilisant un système Unix ou similaire les numéro un au niveau des parts de marché (si vous

14 <http://www.idc.com/itforecaster/itf20000808.stm>

regroupez les chiffres de GNU/Linux et d'Unix).

IDC a réalisé une étude similaire le 17 janvier 2001 intitulée « Server Operating Environments: 2000 Year in Review »<sup>15</sup>. Au niveau des serveurs, Windows est arrivé à 41% des ventes des systèmes d'exploitation pour les nouveaux serveurs en 2000 et s'est accru de 20%. Cependant, GNU/Linux est parvenu à 27% et augmenta encore de 24%. Les autres principaux systèmes Unix comptaient alors 13%.

Les données de ce type ont inspiré des rapports tels celui d'IT-Director du 12 novembre 2001 « Linux on the desktop is still too early to call, but on the server it now looks to be unstoppable. »<sup>16</sup>.

Toutefois, ces mesures ne prennent pas en compte tous les systèmes des serveurs installés cette année. Certains systèmes Windows (installés illégalement) n'ont pas été payés et les systèmes d'exploitation OSS/FS comme GNU/Linux et BSD sont souvent téléchargés et installés sur plusieurs systèmes (puisque'il est légal et gratuit de le faire).

5. Le « November 2001 edition of the Evans Data International Developer Survey Series »<sup>17</sup> rapportait, lors d'interviews de plus de 400 promoteurs (représentant plus de 70 pays) sur le système d'exploitation qu'ils pensaient utiliser pour la plupart de leurs applications l'année prochaine, que 48,1% des promoteurs internationaux et 39,6% des Nord-Américains avaient l'intention de diriger la plupart de leurs applications vers GNU/Linux alors que moins d'un tiers de la communauté des promoteurs internationaux utilisaient GNU/Linux l'année précédente (pour leurs applications). L'étude montrait aussi que 37,8% de la communauté internationale des promoteurs et 33,7% des promoteurs nord-américains avaient déjà écrit des applications pour GNU/Linux, et que plus de la moitié de ceux repris dans l'étude avaient suffisamment confiance en GNU/Linux pour l'utiliser pour des applications portant sur des missions exigeantes.
6. Microsoft a commandité sa propre enquête pour « prouver » que GNU/Linux n'est pas largement utilisé, mais de gros défauts apparaissaient dans cette enquête. Microsoft a sponsorisé un rapport du groupe Gartner<sup>18</sup> annonçant le fonctionnement sous GNU/Linux de

---

15 [http://www.computer.org/computer/homepage/june/ind\\_trends/index.htm](http://www.computer.org/computer/homepage/june/ind_trends/index.htm)

16 <http://www.it-director.com/article.php?id=2332>

17 [http://www.businesswire.com/cgi-bin/f\\_headline.cgi?bw.111301/213170209](http://www.businesswire.com/cgi-bin/f_headline.cgi?bw.111301/213170209)

18 <http://www.zdnet.com/zdnn/stories/news/0,4586,2772060,00.html>

seulement 8,6% des serveurs installés aux U.S.A. pendant les trois quarts de l'année 2000. Cependant, Microsoft, étant sponsor de l'enquête, avait une entière liberté pour créer des chiffres très bas qui sont bien différents de ceux de l'étude IDC sur le même sujet. Kusnetzky d'IDC explique que le groupe Gartner a probablement utilisé une définition très étroite de ce qu'est un serveur installé. Par exemple, il n'est pas tout à fait objectif de ne prenant en compte que les serveurs utilisant officiellement GNU/Linux. « Nos recherches ont montré que la plupart des utilisateurs n'obtiennent pas GNU/Linux en l'achetant. Nous avons constaté que 10 à 15% des adoptions de GNU/Linux venaient de copies déjà installées. Pour chaque copie payée de GNU/Linux, on peut avoir une copie gratuite qui peut elle-même être copiée 15 fois. » De plus, il est très difficile d'acheter un nouvel ordinateur x86 sans un système d'exploitation Microsoft (les contrats de Microsoft avec les fabricants l'exigent), ce qui n'implique pas l'utilisation de ses systèmes. Gartner explique qu'il a effectué des interviews pour contrer ce problème, mais le résultat final de sa recherche (comparé aux faits réels) suggère qu'il n'a pas pu l'éviter. Gartner rapporte, par exemple, qu'il n'existe aucune utilisation de GNU/Linux dans le domaine des super-ordinateurs. Pourtant, GNU/Linux est largement utilisé dans des clusters parallèles sur beaucoup de sites scientifiques, y compris sur des sites de grande envergure. Beaucoup de ces systèmes ont été « faits maison », cela montre que la méthode de Gartner définissant une « installation » ne présente qu'un lointain rapport avec les installations en fonctionnement. L'article du Register « No one's using Linux »<sup>19</sup> (ainsi que son compagnon « 90% Windows.. »<sup>20</sup>) en parle plus longuement. En bref, l'enquête commanditée par Microsoft rapporte des chiffres très bas, mais qui demeurent bien suspects.

7. En 1999, GNU/Linux présentait au moins 80% du chiffre d'installations de MacOS sur Apple. Selon une étude IDC de juin 2000<sup>21</sup> à propos des licences de 1999 (5% pour MacOS, 4,1% pour GNU/Linux), on constatait au moins autant d'installations par des clients pour GNU/Linux qu'il y en avait pour MacOS.

Actuellement, GNU/Linux est un concurrent relativement nouveau dans le marché des systèmes d'exploitation clients et possède une petite pénétration du marché. Cette constatation n'est pas surprenante: contrairement à la richesse des applications des serveurs

---

19 <http://www.theregister.co.uk/content/4/19662.html>

20 <http://www.theregister.co.uk/content/4/19661.html>

21 <http://www.idc.com/itforecaster/itf20000808.stm>

et des réalisateurs, GNU/Linux a relativement peu d'applications OSS/FS compatibles et beaucoup de ces applications sont encore en cours d'élaboration. Il existe des applications commerciales compatibles (par exemple: le Word Perfect de l'éditeur Corel), mais elles ne sont pas OSS/FS et sont également disponibles sur d'autres plates-formes. Microsoft Office représente quasiment un monopole contrôlant le marché des applications « office » et n'est pas disponible sur GNU/Linux (Note: le deuxième plus gros producteur d'applications « office » est Corel mais, depuis que Microsoft en est partiellement propriétaire, il ne peut plus être réellement considéré comme concurrent). Des applications office OSS/FS sont en cours d'élaboration: Open Office<sup>22</sup> (le projet libre du produit Star Office de Sun Microsystems) vient de sortir en version 1.0 comme OSS/FS, et les alternatives incluent AbiWord, Gnumeric, et la suite Koffice. De plus, le coût de la mise en place de tous ces logiciels face au monopole établi par Microsoft fait qu'il est difficile pour les clients d'utiliser un autre système que Windows et Office.

Pourtant, en dépit du peu d'applications OSS/FS clientes (compatibles), une étude IDC<sup>23</sup> a déterminé sur 98,6 millions d'installations de systèmes d'exploitation clients en 1999 (parmi lesquels Windows 3.X, 95 et 98 comptaient pour 66%), Windows NT représentait 21%, MacOS 5% et GNU/Linux 4%. Il sera intéressant d'observer l'évolution de ces chiffres vers 2003. Nous pouvons déjà constater que certaines personnes anticipent également cette évolution: Richard Thwaite, directeur d'IT pour Ford Europe, expliquait en 2001<sup>24</sup> qu'il avait comme objectif un logiciel open source, et qu'il s'attendait à ce que l'on y parvienne (Richard Thwaite contrôle 33000 ordinateurs, donc son intervention ne serait pas négligeable). Nous pourrions considérer qu'il ne s'agit que d'un stratagème pour négocier avec Microsoft, mais ce genre de démarche ne fonctionne que si elle est crédible.

8. Business plan pour augmenter l'utilisation de GNU/Linux. Une étude de Zona Research a démontré que plus de la moitié des grandes entreprises interrogées s'attendaient à des augmentations de plus de 25% du nombre d'utilisateurs de GNU/Linux dans leur entreprise, et plus de 20% s'attendaient à une croissance de plus de 50%. Dans les petites compagnies, plus d'un tiers pensaient que l'utilisation de GNU/Linux augmenterait de 50%. Les facteurs

---

22 <http://www.openoffice.org>

23 <http://news.cnet.com/news/0-1003-200-1546430.html>

24 <http://www.silicon.com/bin/bladerunner?30REQEVENT=&REQAUTH=21046&14001REQSUB=REQINT1=454>

les plus importants motivant ces décisions étaient la fiabilité, les prix plus bas, la rapidité des applications, et l'adaptabilité. En voici les chiffres:

| Utilisation attendue de GNU/Linux | Petites entreprises | Moyennes entreprises | Grandes entreprises | Total |
|-----------------------------------|---------------------|----------------------|---------------------|-------|
| Croissance de 50%                 | 21.0%               | 16%                  | 19.0%               | 19%   |
| Croissance de 10-25%              | 30.5%               | 42%                  | 56.5%               | 44%   |
| Pas de croissance                 | 45.5%               | 42%                  | 24.5%               | 36%   |
| Réduction                         | 3.0%                | 0%                   | 0%                  | 1,00% |

Vous pouvez trouver plus d'informations sur cette étude dans « The New Religion: Linux and Open Source »<sup>25</sup> et dans l'article d'Info World du 5 février 2001 « Linux lights up enterprise: But concerns loom about OS vendor profitability. ».

9. Le top 1000 global des « Internet Service Providers » (ISP) s'attend, selon une étude<sup>26</sup> d'Idaya conduite de janvier à mars 2001, à ce que l'utilisation de GNU/Linux augmente de 154%. De plus, environ les deux tiers (64%) des ISP considèrent que l'Open Source software rencontre les besoins des entreprises en ce qui concerne le niveau de leurs applications et est donc comparable au logiciel de propriété industrielle. La firme Idaya produit des logiciels OSS/FS; il est donc nécessaire de garder cela à l'esprit comme étant un parti pris potentiel.

10. IBM a découvert une croissance de 30% dans le nombre des applications GNU/Linux au niveau des entreprises pendant une période de six mois s'achevant en juin 2001. Il fut un temps où il était commun de dire: « Il n'y a pas suffisamment d'applications fonctionnant sous GNU/Linux au niveau de l'utilisation par les entreprises. ». Toutefois, selon une étude<sup>27</sup> de la firme IBM, il existe plus de 2300 applications GNU/Linux disponibles chez IBM (avec un accroissement de 30% en six mois) et les premières industries indépendantes vendant du logiciel. Un rapport spécial<sup>28</sup> de « Network Computing » sur Linux pour l'entreprise présente certains avantages et certains inconvénients de GNU/Linux. Ce document présente de nombreux points positifs sur les applications GNU/Linux pour les entreprises.

25 <http://www.zdnet.com/eweek/stories/general/0,11011,2651826,00.html>

26 [http://www.idaya.co.uk/news/newsdesk/iday\\_a\\_ispsurvey.phtml?member=ea1df79a5918559b4a8ce19ebbf22b0](http://www.idaya.co.uk/news/newsdesk/iday_a_ispsurvey.phtml?member=ea1df79a5918559b4a8ce19ebbf22b0)

27 [http://www.businesswire.com/cgi-bin/f\\_headline.cgi?bw.062701/211782585&ticker=IBM](http://www.businesswire.com/cgi-bin/f_headline.cgi?bw.062701/211782585&ticker=IBM)

28 <http://www.networkcomputing.com/1224/1224f1.html>

11. Une étude TechRepublic intitulée « Références, Tendances et Prévisions: rapport Linux »<sup>29</sup> datant de l'année 2001 a constaté que 46,6% des professionnels IT étaient confiants dans le soutien apporté à GNU/Linux par leurs organisations. Ce pourcentage représente un taux de confiance plus élevé que pour n'importe quel autre système d'exploitation à l'exception de Windows. Vu la domination du marché, il n'est pas surprenant de constater que la majorité de ces professionnels étaient sûrs du soutien que leurs organisations pourraient apporter aux différentes versions de Windows (pour Windows NT le chiffre s'élève à 90,6% et pour Windows 2000 à 81,6%).

Cependant, GNU/Linux arrivait en troisième position avec 46,4%. La moitié des sondés (environ) répondaient que leurs organisations étaient déjà confiantes dans leur capacité à soutenir GNU/Linux. GNU/Linux devance donc d'autres produits connus de longue date incluant Unix (42,1%), Novell Netware (39,5%), Sun Solaris (25,7%), et Apple (13,6%). TechRepublic suggérait qu'il y avait plusieurs raisons possibles à ce résultat étonnamment élevé:

- GNU/Linux est considéré comme étant une technologie « ascendante ». Beaucoup de professionnels IT sont déjà en train de l'étudier et d'apprendre son fonctionnement vu sa commercialisation à court terme.
- beaucoup de professionnels IT utilisent déjà GNU/Linux à domicile, permettant à GNU/Linux une entrée dans les organisations professionnelles.
- vu la similarité de GNU/Linux avec Unix, les professionnels IT compétents dans l'utilisation d'Unix peuvent aisément passer à GNU/Linux.

TechRepublic suggère un inventaire des aptitudes du personnel au niveau des cadres IT parce qu'il existe une grande probabilité que leur organisation soit déjà apte à soutenir GNU/Linux s'ils s'en sont déjà servis.

12. Le programme OSS/FS Sendmail demeure le principal serveur e-mail d'Internet. Une étude<sup>30</sup> réalisée entre le 27/09/2001 et le 3/10/2001 par D.J. Bernstein sur un échantillon aléatoire d'un million d'adresses IP reliées avec succès à 958 serveurs SMTP (e-mail) a démontré qu'« Unix Sendmail » occupait la première place du marché (42% des serveurs e-mail), suivi par « Windows Microsoft Exchange » (18%), « Unix Qmail » (17%), « Windows Ipswitch

---

29 <http://www.zdnet.com.au/newstech/os/story/0,2000024997,20261699,00.htm>

30 <http://cr.yip.to/surveys/smtpsoftware6.txt>



Email » (6%), « Unix smap » (2%), « UNIX Postfix » (autrefois « Vmailer » 2%) et « Unix Exim » (1%). Notons toutefois que l'outil de Bernstein est l'un des concurrents de « Sendmail » (qmail). Sa position est donc un gage de crédibilité et d'objectivité. « Qmail » se différencie d'un OSS/FS par le fait que les dérivés de qmail ne sont pas librement redistribuables. « Qmail » est à « source visible » ce qui entraîne une confusion avec les OSS/FS par de nombreuses personnes. Néanmoins, « Sendmail », « Postfix » et « Exim » appartiennent tous aux OSS/FS. Le programme leader (« Sendmail ») possède un nombre d'installations correspondant au double de son concurrent le plus proche.

13. Une enquête faite dans le deuxième trimestre de l'année 2000 a prouvé que 95% de tous les « reverse-lookup domain name servers » (DNS) utilisaient le programme « Bind », un autre produit OSS/FS. Internet a été construit à partir de composants d'infrastructure invisibles en majeure partie, ceci incluant les serveurs de noms de domaine qui traduisent des noms de machines humainement lisibles (comme « yahoo.com ») en adresses numériques. Les machines publiquement accessibles supportent généralement les « reverse lookups » qui convertissent les nombres en noms. Pour des raisons historiques, ce programme a été mis en application en utilisant le domaine caché « in-addr.arpa ». En étudiant le domaine « in-addr », il est possible d'observer de l'intérieur le support d'Internet dans sa globalité. Bill Manning a étudié<sup>31</sup> le domaine in-addr et est arrivé à la conclusion que 95% des noms de serveurs (durant le deuxième trimestre 2000) accomplissent cette tâche importante de l'infrastructure d'Internet grâce à une version de « bind » qui est un programme OSS/FS.

---

31 <http://www.isi.edu/~bmanning/in-addr-versions.html>

## 3.2 Quels avantages pour les sociétés utilisatrices de logiciels libres?

### 3.2.1 Compatibilité (matérielle, protocoles, ...)

#### Normes de programmation

Son code source libre représente, de par sa définition, le noyau Linux et ne peut être utilisé à des fins propriétaires.

En effet, les termes de la GPL restent stricts dans ce domaine. Une entité quelconque ne peut apporter des modifications au noyau Linux pour l'utiliser de manière externe sans en rendre le code source correspondant publiquement disponible (sous peine de sanctions pénales).

Ces modifications réduisent à néant toute tentative de « détournement » du noyau pour en produire une variante propriétaire. La communauté des utilisateurs et des développeurs n'a aucun intérêt à créer des normes spéciales et des protocoles propriétaires. Ainsi le système d'exploitation intègre naturellement les standards industriels reconnus.

Nous pourrions penser que cette intégration n'est que pure théorie mais GNU/Linux est aujourd'hui un système d'exploitation compatible à la norme POSIX et ses sous-systèmes constitutifs demeurent en accord avec toutes les normes standards comme ANSI, ISO, ITEF et W3C. Cependant, la demande de certification est une démarche séparée et non gratuite. La communauté GNU/Linux se refuse à payer des organismes de normalisation pour des certifications ne lui apportant pas beaucoup d'intérêt. En conséquence, GNU/Linux est conforme avec de nombreuses normes sans réelle certification.

Etrangement, alors que GNU/Linux est en accord avec les principales normes industrielles standards, il existe toujours un manque de standardisation entre les différentes distributions GNU/Linux. Ces différences sont mineures car toutes les distributions ont libre accès aux sources du noyau Linux dans son intégralité. Ces distributions diffèrent juste au niveau des applications qu'elles joignent à leur distribution, par les utilitaires d'installation et par les emplacements des différents fichiers-système.

Un projet nommé « Linux Standard Base » a été créé et vise à unifier toutes les distributions selon le respect de quelques critères comme les emplacements des répertoires destinés aux fichiers système. Il s'agit d'une sorte de charte que tout le monde peut télécharger et sur laquelle on peut méditer, réfléchir, mais aussi proposer des amendements. L'aboutissement de ce projet est dans

l'intérêt des utilisateurs puisqu'il conduira à la polyvalence et l'uniformisation de l'expérience des utilisateurs. Les 9 distributions suivantes ont été certifiées conformes à la norme LSB: Caldera, Debian, Mandrake, MontaVista, Red Hat, SuSE, SOT, Sun et TurboLinux.

### Normes de réseaux et de protocoles

Linux n'est pas en reste en ce qui concerne les standards des protocoles pour les réseaux Internet, Intranet, etc.

En effet, Linux gère totalement les protocoles principaux et moins répandus (TCP/IPv4 mais aussi IPv6 depuis 1999, IPX/SPX, AppleTalk, X.25, Ethernet, Token Ring, fibre optique, frameRelay, ISDN et bien d'autres).

Cette prise en charge rend GNU/Linux semblable aux autres systèmes d'exploitation si nous le comparons au point de vue communication avec d'autres ordinateurs.

GNU/Linux est également capable de voir et modifier les fichiers Windows sur des partitions FAT32 ou NTFS (Windows est incapable de voir les partitions EXT2 et EXT3 de GNU/Linux).

Par le jeu d'émulateurs ou de machines virtuelles, il est parfois possible de faire tourner des applications Windows sous un environnement GNU/Linux. Ce principe va utiliser beaucoup de ressources système pour exécution mais c'est possible sans trop de difficultés.

### Normes de matériel

GNU/Linux est compatible avec quasi tous les processeurs existants. Cette possibilité permet une meilleure uniformisation de l'ensemble des ordinateurs d'un réseau ou d'une entreprise. En effet, les outils informatiques sont multiples et il est souvent frustrant de devoir choisir une seule architecture informatique comme uniformisation uniquement à cause du système d'exploitation ne reconnaissant pas les autres architectures. Le problème a été résolu avec GNU/Linux. En effet, GNU/Linux est capable de reconnaître les partitions et les ordinateurs distants quel que soit leur système d'exploitation. Avec le logiciel SAMBA, GNU/Linux peut se muter en serveur NetBIOS mais peut également être installé sur toutes les machines. Si le système GNU/Linux satisfait un dirigeant d'entreprise, trouver une distribution GNU/Linux adaptée à l'architecture de n'importe quel ordinateur est totalement réalisable. Cela va du processeur Motorola qui se trouve dans les PDA (Personal Digital Assistant) aux gros serveurs Sparc.

Les processeurs capables d'exécuter GNU/Linux sont: Intel x86 et Pentium, Motorola, Sparc et UltraSparc, Itanium, Alpha, ARM, les processeurs embarqués, les systèmes en temps réel mais aussi les AS/400.

### 3.2.2 Portabilité

Quelques considérations intéressantes au sujet de la portabilité, un terme simple mais avec des implications multiples:

GNU/Linux et FreeBSD (deux systèmes d'exploitation libres couramment utilisés) supportent un éventail de plate-formes. Beaucoup utilisent le terme « adaptabilité » pour répondre à la question: « Pouvez vous utiliser le même système de logiciels pour des petits mais aussi de grands projets? » Nombreux sont ceux qui voudraient commencer par un système modeste mais avec la possibilité d'accroître le système selon l'exigence des besoins sans modification onéreuse. Sur ce point, les OSS/FS sont imbattables. Vu que de nombreuses personnes peuvent rencontrer des problèmes d'adaptabilité, mais aussi vu que son code source peut être optimisé pour chaque plate-forme, l'adaptabilité de nombreux produits OSS/FS reste étonnante. Observons spécifiquement le système GNU/Linux. GNU/Linux fonctionne sur des PDA, sur du hardware désuet, sur du matériel PC moderne, sur une douzaine de chipsets différents, sur des mainframes, sur des clusters et sur un certain nombre de super ordinateurs. GNU/Linux peut être utilisé pour le traitement parallèle massif: une approche commune pour ce traitement est l'architecture Beowulf. Citons Sandia « Cplant » qui fonctionne sur un ensemble de systèmes exécutant GNU/Linux. Il s'agit du quarante-deuxième ordinateur le plus performant au monde en juin 2001 (numéro 42 sur la liste PRINCIPALE de 500 ordinateurs géants, juin 2001<sup>32</sup>). Il existe également un prototype de mise en place de GNU/Linux sur une montre-bracelet.

Ainsi, il est possible d'acheter un petit système GNU/Linux et de l'accroître lorsque vos besoins augmentent. En effet, vous pouvez remplacer le matériel, ou les processeurs par d'autres possédant une plus grande vitesse d'horloge, ou les architectures par d'autres très différentes au niveau CPU, et cela sans changer de système d'exploitation. Les Windows CE/ME sont réduits à de petites plate-formes et travaillent seulement sur les systèmes x86. Beaucoup de systèmes Unix (tels que Solaris)

---

32 <http://www.top500.org/list/2001/06>

s'adaptent bien à de grandes plate-formes spécifiques mais pas aussi bien aux petites plate-formes répandues. Les systèmes OSS/FS font partie donc des programmes les plus adaptables.

### 3.2.3 Performance

Comparer les performances de GNU/Linux et de Microsoft Windows sur un matériel équivalent a souvent donné des résultats controversés parce que fondés sur différentes hypothèses. Selon moi, les OSS/FS ont au moins prouvé qu'ils sont compétitifs et parfois leur supériorité dans de nombreuses circonstances.

Les comparaisons sur les performances dépendent des attentes et de l'environnement. Ainsi la meilleure référence est celle que vous trouverez vous-même dans l'environnement de votre choix. Si vous ne procédez pas de cette manière, vous chercher à utiliser des mesures objectives car il est aisé de créer des mesures faussées.

Voici tout d'abord quelques études récentes suggérant que certains systèmes OSS/FS battent leur concurrence propriétaire dans certaines circonstances:

1. L'article de PC magazine datant de novembre 2001: « Performance Tests: File Server Throughput and Response Times »<sup>33</sup> a constaté que GNU/Linux et Samba surpassaient de manière significative Windows 2000 Server utilisé comme serveur de fichiers avec les protocoles réseau de Microsoft. Cette constatation était valable indépendamment du nombre de clients simultanés (jusqu'à 30 clients testés), et cela se vérifiait sur l'ensemble des ordinateurs utilisés (Pentium II/233MHz avec 128 MB de RAM, Pentium III/550MHz avec 256 MB de RAM, et Pentium III/1GHz avec 512 MB de RAM, où MB signifie megabytes ou  $10^6$  bytes <sup>34</sup>). En fait, plus les machines devenaient performantes plus la différence était prononcée. Dans le cas du plus grand nombre de clients sur le matériel le plus rapide, le débit de GNU/Linux était d'environ 130 MB/sec contre les 78MB/sec de Windows (GNU/Linux était 78% plus rapide).

---

33 <http://www.pcmag.com/article/0,2997,s%253D25068%2526a%253D16554,00.asp>

34 <http://physics.nist.gov/cuu/Units/binary.html>

2. Au cours des tests de performance réalisés par le magazine Sys Admin, GNU/Linux bat Solaris (sur Intel), Windows 2000 et FreeBSD. L'article « Which OS is Fastest for High-Performance Network Applications? »<sup>35</sup> (de l'édition de juillet 2001 du magazine Sys Admin<sup>36</sup>) a examiné des systèmes à rendement élevé et a constaté que GNU/Linux bat ses concurrents: Solaris (sur Intel), FreeBSD (un système OSS/FS), et Windows 2000. Les testeurs ont intentionnellement exécuté les systèmes de façon non-optimisée si ce n'est qu'ils ont augmenté le nombre de connexions TCP/IP simultanées (qui sont nécessaires pour le test des applications à processus multiples et asynchrones). Ils ont utilisé les dernières versions des systèmes d'exploitation ainsi que la même machine et ont enregistré les résultats (par système d'exploitation) de deux essais de performance différents.

Les développeurs de FreeBSD se sont plaints au sujet de ces essais notant que FreeBSD par défaut augmentait la fiabilité (pas la vitesse) et que n'importe qui ayant un besoin significatif de performance procéderait d'abord à des réglages. Alors, Sys Admin refit des essais pour FreeBSD<sup>37</sup> après optimisation. Un des changements opérés était de passer au support «asynchrone» rendant le système plus rapide (avec augmentation du risque de perte de données en cas de panne de courant). Il s'agit du support par défaut dans GNU/Linux, et il est facile à changer dans FreeBSD. La modification est très petite et raisonnable. Toutefois, d'autres changements ont été effectués. Par exemple, 17 patches du noyau FreeBSD ont été découverts et compilés, puis ont été utilisés. Les autres systèmes d'exploitation n'ont pas eu la chance d'être parfaitement paramétrés et donc, comparer des OS non optimisés à un FreeBSD optimisé ne conduit pas à des conclusions probantes.

Leur test « monde réel » a mesuré avec quelle rapidité de grandes quantités d'e-mails pourraient être envoyées en utilisant leur serveur d'envoi (MailEngine). Aucune différence ne survenait jusqu'à 100 envois simultanés, mais à mesure que le nombre augmentait, les systèmes commencèrent à montrer des différences significatives dans leur cadence d'envoi d'e-mails. GNU/Linux était nettement plus rapide que ses concurrents avec 500 envois simultanés, excepté FreeBSD optimisé. FreeBSD optimisé avait des performances semblables à GNU/Linux pour l'exécution de 1000 envois simultanés ou moins. Puis il réalise une pointe avoisinant 1000-1500 connexions simultanées, suivie d'une diminution

---

35 <http://www.samag.com/articles/2001/0107/>

36 <http://www.sysadminmag.com/>

37 <http://www.samag.com/articles/2001/0108/>

constante (non subie par GNU/Linux) de sa performance relative. FreeBSD optimisé a eu des problèmes au moment de dépasser les 3000 envois simultanés. Avec 1500 envois simultanés, GNU/Linux envoyait 1,3 millions d'e-mails par heure tandis que Solaris en envoyait approximativement 1 million, Windows 2000 et FreeBSD non optimisé environ 0,9 millions.

3. GNU/Linux a produit avec TUX (Web serveur/accélérateur) de meilleures valeurs SPEC que Windows/IIS (Internet Information Server) dans de nombreux cas et cela même sur des configurations inférieures. Le consortium SPEC<sup>38</sup> est une organisation essayant de donner des « benchmarks » impartiaux. Nous pouvons comparer Microsoft Windows et GNU/Linux au niveau des résultats SPECweb99 (qui mesurent les performances de serveurs web) avec un matériel identique (les deux systèmes d'exploitation ayant subi la même quantité d'efforts d'optimisation de performance). Malheureusement, les choses peuvent se compliquer: nous ne pouvons que rarement tester les deux systèmes d'exploitation avec le même équipement. Ainsi, lors du test conduit le 13 juillet 2001, les configurations testées différaient, par exemple, en utilisant un nombre différent de disques durs ou en incluant quelques disques durs plus rapide. En utilisant tous les résultats disponibles au 13 juillet 2001, il existait trois configurations de matériel, toutes de Dell [utilisant GNU/Linux (par emploi du Web server/accelerator TUX) et Windows (par emploi IIS)] sur un matériel de base identique.

---

38 <http://www.spec.org/>

Voici les résultats SPECweb99 en date du 13 juillet 2001, notant des différences de configuration:

| Systeme  | Résultats SPEC Windows   | Résultats SPEC Linux                  |
|--|--|---------------------------------------|
| Dell PowerEdge<br>4400/800, 2 800MHz<br>Pentium III Xeon | 1060 (IIS 5.0, 1 network controller)   | 2200 (TUX 1.0, 2 network controllers) |
| Dell PowerEdge<br>6400/700, 4 700MHz<br>Pentium III Xeon | 1598 (IIS 5.0, 7 9GB 10KRPM drives)  | 4200 (TUX 1.0, 5 9GB 10KRPM drives)   |
| Dell PowerEdge<br>8450/700, 8 700MHz<br>Pentium III Xeon | 7300/NC (IIS 5.0, 1 9Gb 10KRPM and 8 16Gb 15KRPM drives) then 8001 (IIS 5.0, 7 9Gb 10KRPM and 1 18Gb 15KRPM drive) | 7500 (TUX 2.0, 5 9Gb 10KRPM drives)   |

La première ligne (le PowerEdge 4400/800) n'est pas vraiment probante. Le système d'IIS présente des performances inférieures. Mais, ce programme possède un contrôleur de réseau. Alors que le système de TUX en a deux. Donc, ce système aurait présenté de meilleures performances en cas d'utilisation de deux connections réseau.

La deuxième ligne (le PowerEdge 6400/700) démontre que GNU/Linux additionné de TUX restent les plus performants. Le système d'IIS était doté de deux disques durs supplémentaires (afin d'améliorer les performances du système). Pourtant, le système TUX a réalisé des performances plus de deux fois supérieures à celles d'IIS.

La dernière ligne pour le PowerEdge 8450/700 reste bien plus complexe. D'abord, les lecteurs des deux systèmes sont différents. Ce qui devrait offrir à IIS le rendement le plus élevé vu que la vitesse de transfert est certainement plus importante.

Ainsi, la configuration GNU/Linux additionnée du système TUX (bien que testée sur un matériel inférieur) a présenté de meilleures performances en certaines occasions. Etant donné que d'autres facteurs peuvent être ajoutés, l'ensemble demeure difficile à juger. Dans certaines situations, un meilleur matériel peut produire une performance moins satisfaisante si un autre facteur non enregistré a un effet significatif. Si tout va bien à l'avenir, il y aura beaucoup d'essais sur de semblables configurations.



Notons que le logiciel TUX est fréquemment destiné à être utilisé comme «web accelerator» dans des systèmes où il manipule rapidement des demandes simples. Les requêtes plus complexes sont acheminées vers un autre serveur (habituellement Apache). Les chiffres de TUX ont été cités dans ce travail car il s'agit de chiffres de performances récents.

4. Ed Bradford (manager du Microsoft Premier Support pour IBM Software group) a édité en octobre 2001 l'étude: « Pipes in Linux, Windows 2000, and Windows XP »<sup>39</sup> dans laquelle il examine la performances des pipes, un mécanisme commun de communication entre les processus de programmes. M. Bradford considère que les pipes dans Red Hat 7.1 (avec le kernel Linux 2.4.2) présentaient un pic I/O d'environ 700 MB/sec avec une moyenne à 100 MB/sec pour les blocs de très grande taille. En revanche, Windows 2000 a réalisé une pointe à 500 MB/sec avec un état d'équilibre de 80 MB/sec pour les blocs de grande taille. Windows XP Professionnal (version d'évaluation) était particulièrement décevant. Sa cadence de I/O était seulement de 120 MB/sec, avec un équilibre à 80 MB/sec (tous sur le même matériel et tous utilisant une interface utilisateur graphique).

En février 2002, Ed Bradford édita « Managing processes and threads »<sup>40</sup>, dans lequel il a comparé les performances de Red Hat Linux 7.2, de Windows 2000 Advanced Server et de Windows XP Professional, tous s'exécutant sur un Thinkpad 600X avec 320MB de mémoire. Linux est arrivé à créer 10.000 processus indépendants par seconde alors que Windows 2000 Advanced Server n'en contrôlait pas tout à fait 5.000 et que Windows XP Professional en a seulement créé 6.000. Au niveau de la création de processus, Linux a contrôlé 330 processus/seconde alors que Windows 2000 Advanced Server en contrôlait moins de 200 et Windows XP Professional moins de 160.

5. Fastcenter<sup>41</sup> a mené des tests de performance de base de données en utilisant Oracle sur exactement le même matériel (doubles processeurs de 1 GHz, 2 Gigabytes de RAM). Ils ont comparé Microsoft Windows 2000 Server (version 5.0.2195, build 2195, sans patch) à SuSE Enterprise Linux Server 7 (installation standard). Les deux systèmes d'exploitation ont fait l'objet d'installations « courantes » (cf. les benchmarks de FastCenter pour plus d'information). Windows 2000 Server instance build a seulement réalisé 70% de la

---

39 <http://www.ibm.com/link/developerworks/?open&t=grl,l=252,p=pipes>

40 <http://www.ibm.com/link/developerworks/?Open&t=grl,l=252,p=mgth>

41 <http://www.fastcenter.com/>

performance de GNU/Linux. Une fois installé, Microsoft Windows 2000 server réalisa en moyenne 85% du débit de SuSE Linux. Gardons également en mémoire l'association de Fastcenter avec la SuSE Corporation (un distributeur de Linux). Pour plus d'information, veuillez vous référer à l'article de Fastcenter.

6. Une récente comparaison de serveurs de bases de données réalisée par IBM a présenté un très net avantage, au niveau des performances, des solutions sous Linux RedHat par rapport à celles sous Windows 2000.<sup>42 43</sup>

### 3.2.4 Fiabilité

Il existe plusieurs anecdotes sur la fiabilité des OSS/FS, mais finalement de nombreuses données la confirment.

1. Les applications équivalentes d'OSS/FS sont, selon une étude de 1995, plus fiables. Le «Fuzz Revisited» de l'année 1995<sup>44</sup> a mesuré la fiabilité de ces systèmes en alimentant des programmes avec des caractères aléatoires et en déterminant lesquels ont résisté aux crashes.

De plus, il est également intéressant de comparer les résultats des tests des systèmes commerciaux aux résultats des tests des « freeware » GNU et de GNU/Linux. Les sept systèmes commerciaux dans l'étude de 1995 présentent un pourcentage de panne de 23%, contre 9% pour GNU/Linux et 6% seulement pour les utilitaires de GNU. Il est raisonnable de se demander pourquoi un groupe de programmeurs globalement dispersé, sans support officiel de test ou de programmation peut produire un code plus fiable (selon cette mesure au moins) que le code produit commercialement. Même si vous ne considérez que les utilitaires qui étaient fournis par GNU, utilisant un autre noyau que Linux, ou GNU/Linux, les pourcentages de panne de ces deux systèmes sont plus avantageux que les autres systèmes.

« An Empirical Study of the Robustness of Windows NT Applications Using Random Testing »<sup>45</sup>, un article de l'année 2000, a constaté qu'ils pouvaient faire tomber en panne

---

42 <http://www.zdnet.com/filters/printerfriendly/0,6061,2760874-2,00.html>

43 <http://www.tpc.org/tpch/results/h-ttperf.idc>

44 <http://www.cs.wisc.edu/~bart/fuzz/fuzz.html>

45 <http://www.cs.wisc.edu/~bart/fuzz/fuzz.html>

21% des applications testées avec des applications GUI de Windows NT, suspendre 24 autres pour cent des applications, et faire tomber en panne ou arrêter toutes les applications testées en les soumettant à des messages aléatoires.

Bien que cette expérience ait été faite en 1995, aucune amélioration apportée depuis lors ne peut suggérer que le logiciel propriétaire soit devenu meilleur que l'OSS/FS. En effet, depuis 1995 il y a eu un intérêt et une participation accrues pour les OSS/FS avec pour résultat un nombre accru de personnes examinant et améliorant la fiabilité des programmes OSS/FS.

Les programmes OSS/FS ne représentent pas pour autant le remède miracle aux problèmes de fiabilité. Un logiciel en phase de test reste toujours sujet à des « bugs » éventuels.

2. GNU/Linux est plus fiable que Windows NT, selon une expérience de dix mois de Zdnet<sup>46</sup>. ZDnet a mené un test de dix mois sur la fiabilité en comparant les systèmes OpenLinux de Caldera, Red Hat Linux, et Windows NT Server 4.0 de Microsoft avec Service Pack 3. Chaque système a utilisé un matériel (CPU unique) identique. De plus, des requêtes réseau ont été envoyées en parallèle à chaque serveur pour des services Internet standard, des services d'impression, et des services d'archivage. Le résultat fut que le serveur NT tomba en panne en moyenne une fois toutes les six semaines, chaque panne nécessitant environ 30 minutes de réparation. Ce résultat n'est pas mauvais en soi, mais ni l'un ni l'autre des serveurs GNU/Linux ne s'est jamais planté. Cet article de ZDnet réalise également un excellent travail en identifiant des faiblesses de GNU/Linux.
3. GNU/Linux semble aussi plus fiable que Windows NT selon une expérience annuelle de Bloor Research. Bloor Research<sup>47</sup> possédait les deux systèmes d'exploitation sur des machines Pentium relativement anciennes. En l'espace d'un an, GNU/Linux ne tomba en panne qu'une seule fois en raison d'un défaut matériel (problèmes de disque), réparé en 4 heures, lui donnant une disponibilité mesurée de 99,95 pour cent. Windows NT est tombé en panne 68 fois suite à des problèmes de matériel (disque), de mémoire (26 fois), de gestion de fichiers (8 fois), et d'un certain nombre de problèmes divers (33 fois). Le total des réparations a atteint 65 heures et a donné une disponibilité de 99,26 pour cent. Il est toutefois étrange que l'unique problème de GNU/Linux et un certain nombre de problèmes de Windows aient été dus à des problèmes matériels. Nous pourrions penser à priori que le

---

46 <http://www.zdnet.fr>

47 <http://gnet.dhs.org/stories/bloor.php3>

matériel utilisé pour tester Windows était plus mauvais, ou encore que GNU/Linux a fait un meilleur travail en évitant et en contenant les pannes de matériel. De toute façon, la panne de gestion de fichiers ne peut être imputée qu'à Windows. Les problèmes divers semblent être également causés par Windows.

4. Selon une évaluation suisse portant sur trois mois, les sites utilisant le logiciel de serveur Web IIS de Microsoft passent plus du double de temps (en moyenne) offline que ceux utilisant Apache. Il s'agit des résultats de l'analyse <sup>48</sup> de Syscontrol AG à propos de l'uptime de sites web (annoncé le 7 février 2000) . L'étude fait état d'un test de plus de 100 sites Web suisses, pendant un trimestre, qui furent contrôlés depuis quatre emplacements différents toutes les cinq minutes. Vous pouvez consulter leurs résultats et leurs conclusions en allemand <sup>49</sup>. Vous trouverez ci-dessous l'ensemble des données sous forme de durée moyenne en heures par mois de la panne (le temps pendant lequel le site était inaccessible), plus une moyenne sur trois mois:

| <b>Downtime</b>  | <b>Apache</b> | <b>Microsoft</b> | <b>Netscape</b> | <b>Autres</b> |
|------------------|---------------|------------------|-----------------|---------------|
| <b>Septembre</b> | 5.21          | 10.41            | 3.85            | 8.72          |
| <b>Octobre</b>   | 2.66          | 8.39             | 2.80            | 12.05         |
| <b>Novembre</b>  | 1.83          | 14.28            | 3.39            | 6.85          |
| <b>Moyenne</b>   | 3.23          | 11.03            | 3.35            | 9.21          |

Il est difficile de ne pas remarquer qu'Apache (le web server OSS) a obtenu les meilleurs résultats sur une moyenne de trois mois (avec également de meilleurs résultats mensuels). En effet, le mois le plus décevant d'Apache était meilleur que le mois le plus satisfaisant de Microsoft. Selon moi, la différence entre Netscape et Apache n'est pas toujours significative statistiquement. Toutefois, ceci nous permet de constater que la solution libre OSS/FS (Apache) possède une fiabilité au moins aussi bonne que la solution propriétaire la plus fiable. L'enquête montre que tout cela ne peut être seulement imputable à un défaut de qualité du logiciel vu qu'il existait de nombreux Microsoft IIS présentant de courtes interruptions simultanément chaque jour (suggérant ainsi des relancements réguliers).

Cependant, nous pouvons toujours nous poser la question suivante: « Pourquoi les sites d'IIS

48 <http://www.syscontrol.ch/e/news/Serversoftware.html>

49 <http://www.syscontrol.ch/e/SWePIX/SWePIXe.html>

ont-ils exigé beaucoup plus de relancements réguliers que les sites utilisant Apache? » Chaque panne, même planifiée à l'avance, a pour conséquence une suspension du service (et pour des sites d'e-commerce, une perte potentielle de ventes).

5. Selon une étude de Netcraft, en date du 10 mars 2002, les OSS/FS se placent en bonne position dans le classement des 50 sites présentant le meilleur uptime. 90% de ces sites utilisent Apache et 82% sont exécutés sur les systèmes d'exploitation OSS/FS. Netcraft conserve une trace des 50 sites les plus souvent demandés (avec les plus longs uptimes) à <http://uptime.netcraft.com>. En date du 10 mars 2002, on peut constater que 90% (45/50) des sites utilisaient le logiciel Apache. Un serveur web de ce site était inconnu, les quatre autres n'utilisaient pas Apache. Dans ces quatre n'utilisant pas Apache, aucun n'utilisait Microsoft IIS. 82% (41/50) fonctionnaient sur un système d'exploitation Open Source et les meilleurs uptimes n'étaient obtenus qu'avec des systèmes d'exploitation à base Unix (Aucun système Windows n'a été repris parmi les meilleurs uptimes).

Comme toutes les études, celle-ci présente des faiblesses (débatues dans les FAQ de Netcraft). Leurs techniques d'identification de serveur web et des systèmes d'exploitation peuvent être critiquées. Seuls les systèmes pour lesquels Netcraft a envoyé beaucoup de demandes sont inclus dans l'étude (il ne s'agit donc pas de l'entièreté des sites dans le monde). Tout site questionné à travers le formulaire de requête « what's that site running » sur Netcraft.com est ajouté à l'échantillonnage de sites testés de manière régulière. Netcraft ne surveille en aucun cas les quelques 22 millions de sites qu'il connaît et cela pour des raisons de performance. De nombreux systèmes d'exploitation ne fournissent pas d'informations sur l'uptime et ne peuvent de ce fait pas être inclus: AIX, AS/400, Compaq Tru64, DG/UX, MacOS, NetWare, NT3/Windows 95, NT4/Windows 98, OS/2, OS/390, SCO UNIX, Sony NEWS-OS, SunOS 4, et VM. Ainsi, ce compteur d'uptime ne peut inclure que les systèmes fonctionnant sur BSD/OS, FreeBSD (excepté la configuration par défaut dans ses version 3 et ultérieures), les versions récentes HP-UX, IRIX, GNU/Linux Kernel 2.1 et ultérieur (sauf sur les systèmes à processeur Alpha), MacOS X, les versions récentes de NetBSD/OpenBSD, Solaris 2.6 et ultérieures et Windows 2000. Remarquons que des systèmes Windows NT ne peuvent être inclus dans cette étude (vu que leurs uptimes ne pourraient être comptés) mais les systèmes Windows 2000 le sont.

Notons que HP-UX (l'Unix de chez Hewlett-Packard), de nombreuses versions de GNU/Linux, Solaris et les versions récentes de FreeBSD se remettent à zéro après 497 jours comme si la machine avait été relancée. Ainsi, nous sommes dans l'incapacité de voir un HP-UX, beaucoup de systèmes GNU/Linux, ou Solaris avec un uptime au-dessus de 497 jours. Toujours, cette étude compare Windows 2000, GNU/Linux (jusqu'à 497 jours habituellement), FreeBSD et plusieurs autres systèmes d'exploitation sur un vaste nombre de serveurs de Web et ce sont OSS/FS qui se distinguent à nouveau.

En outre, recueillir des données dans des circonstances réelles prend du temps et cela constitue un véritable problème. Néanmoins, les résultats disponibles suggèrent que les OSS/FS présentent « une tendance manifeste à la fiabilité ».

### 3.2.5 Sécurité

Mesurer de manière quantitative la sécurité est malaisé. Cependant, en voici quelques tentatives. Celles-ci suggèrent que les OSS/FS demeurent souvent supérieurs aux systèmes propriétaires. Concentrons-nous en particulier sur la comparaison entre les OSS/FS et les systèmes Windows.

1. L'assurance contre les « crackers » de J.S. Wurzler Underwriting Managers<sup>50</sup> coûte 5 à 15% de plus en cas d'utilisation de Windows au lieu d'Unix ou de GNU/Linux pour Internet. Au moins une compagnie d'assurance a indiqué que Windows NT semble moins sécurisé que des systèmes Unix ou GNU/Linux, cela résultant en des primes plus élevées pour les systèmes basés sur Windows. Il est souvent difficile de découvrir quand une compagnie a été infiltrée avec succès. Les compagnies ne désirent souvent pas divulguer une telle information au public pour diverses raisons. En effet, si les consommateurs ou les associés perdaient la confiance en une compagnie, la perte résultante pourrait être bien plus importante que celle due à l'attaque initiale. Néanmoins, les compagnies d'assurance assurant contre l'infiltration de crackers sont en droit de réclamer l'obtention de telles informations (comme un état d'assurance) et peuvent calculer de futures primes sur base de cette connaissance. Selon CNET<sup>51</sup>, J.S. Wurzler Underwriting Managers a été un des premiers

---

50 <http://www.jszum.com>

51 <http://www.cnet.com>

courtiers à avoir proposé une assurance contre les crackers. Walter Kopf, vice-président de cette société d'assurance, signale que « nous avons découvert que la possibilité d'une perte augmente en utilisant le système NT ». Il déclara également que la décision se base sur les résultats de centaines d'évaluations de sécurité que la compagnie a réalisées auprès de ses clients PME pendant les 2 dernières années.

2. La plupart des sites Web attaqués sont hébergés par des machines Windows. De plus, les sites Windows sont attaqués d'une façon disproportionnée par rapport à leur part de marché. Une autre manière de considérer la sécurité est justement de regarder les systèmes d'exploitation employés par les sites Web attaqués et de comparer à leur part de marché. Un « site attaqué » est un site dans lequel quelqu'un s'est introduit et dont le contenu a changé (habituellement d'une façon assez évidente vu que des modifications subtiles ne sont pas souvent enregistrées). Il est souvent très difficile pour les victimes de cacher le fait qu'une personne a réussi à pirater leur système avec succès. Historiquement, cette information a été collectée par Attrition.org. Un résumé peut être trouvé dans article de James Middleton <sup>52</sup> avec les données réelles trouvées sur le site Web d'Attrition.org <sup>53</sup>. Les données d'Attrition.org ont prouvé que 59% des systèmes piratés avec succès pendant la période d'août 1999 à décembre 2000 tournaient sous Windows, 21% sous Linux, 8% sous Solaris et 6% sous d'autres OS. Ainsi, les systèmes sous Windows ont été presque trois fois plus fréquemment attaqués que les systèmes sous GNU/Linux. Ces attaques seraient logiques s'il existait trois fois plus de systèmes Windows mais, quels que soient les chiffres utilisés, cette argumentation ne tient pas la route.

Naturellement, tous les sites ne sont pas piratés via leur web server ou leur OS. Beaucoup sont cassés par des mots de passe exposés, par une mauvaise programmation d'application Web et ainsi de suite. Mais, en ce cas, pourquoi demeure-t-il une si grande différence dans le nombre d'attaques basées sur le système d'exploitation? Quelques autres raisons pourraient être mises en avant mais ceci suggère la possibilité d'une meilleure sécurité pour les OSS/FS. Attrition.org a décidé d'abandonner le maintien de cette information suite à la difficulté de suivre le volume important de sites piratés car il s'est avéré que le traitement de de cette information ne leur serait plus possible. Cependant, le site defaced.alldas.de a

---

52 <http://www.vnunet.com/News/1116081>

53 <http://attrition.org/mirror/attrition/os-graphs.html>

décidé d'assurer la reprise de ce service. Leurs récents résultats prouvent que cette tendance s'est poursuivie. Le 12 juillet 2001, ce site signale que 66,09% de sites attaqués utilisaient Windows, comparés à 17,01% pour GNU/Linux, sur un total de 20.260 sites web piratés.

3. La base de données de vulnérabilité de Bugtraq<sup>54</sup> suggère que le système d'exploitation le moins vulnérable est un OSS/FS et la totalité des systèmes d'exploitation OSS/FS dans son étude semblaient moins vulnérables que des Windows en 1999-2000. Une approche pour analyser la sécurité reste l'utilisation d'une base de données de vulnérabilité. Une analyse d'une base de données constitue la page de statistiques de base de vulnérabilité de Bugtraq. En date du 17 septembre 2000, voici le nombre de vulnérabilités pour certains systèmes d'exploitation principaux:

| Systèmes d'exploitation | 1997 | 1998 | 1999 | 2000 |
|-------------------------|------|------|------|------|
| Debian GNU/Linux        | 2    | 2    | 30   | 20   |
| OpenBSD                 | 1    | 2    | 4    | 7    |
| Red Hat Linux           | 5    | 10   | 41   | 40   |
| Solaris                 | 24   | 31   | 34   | 9    |
| Windows NT/2000         | 4    | 7    | 99   | 85   |

Pour rester critique, nous ne devrions pas prendre ces chiffres trop au sérieux. Certaines vulnérabilités sont plus importantes que d'autres (certaines ne peuvent fournir qu'un service réduit en cas d'exploitation ou ne sont vulnérables qu'en des circonstances peu probables) et seules certaines sont activement exploitées (tandis que d'autres ont été déjà résolues avant exploitation). Les systèmes d'exploitation Open Source tendent à inclure beaucoup d'applications, habituellement vendues séparément dans le cas de systèmes propriétaires (Windows et Solaris inclus). Red Hat 7.1 inclut, par exemple, deux systèmes de base de données relationnelles, deux traitements de textes, deux tableurs, deux serveurs Web et un grand nombre d'éditeurs de textes. En outre, dans le monde Open Source, les vulnérabilités sont discutées publiquement. Ainsi, des vulnérabilités peuvent encore être identifiées pour le logiciel à l'étude (par exemple, les logiciels « beta »). Ceux ayant de petites parts de marché

---

54 <http://www.bugtraq.org>



sont susceptibles d'une analyse moins poussée. Le qualificatif de « petite part de marché » n'a pas lieu d'être avec GNU/Linux, naturellement, vu qu'il a déjà été établi que GNU/Linux est l'OS de serveur n°1 ou n°2 (selon la façon dont vous les comptez). Ceci prouve clairement que les trois systèmes d'exploitation libres énumérés (Debian GNU/Linux, OpenBSD, et Red-Hat Linux) ont fait bien mieux, par cette mesure, que des logiciels Windows en 1999 et en 2000. Même si une distribution bizarre de GNU/Linux était créée explicitement pour reproduire toutes les vulnérabilités actuelles dans n'importe quelle distribution principale de GNU/Linux, cette distribution (intentionnellement mauvaise) de GNU/Linux serait tout de même même meilleure que Windows (Elle aurait connu 88 vulnérabilités en 1999, contre 99 dans les Windows). Les meilleurs résultats sont ceux de OpenBSD, un système d'exploitation libre qui a été, pendant des années, spécifiquement concentré sur la sécurité. Le nombre moins important de faiblesses pourrait être attribué au fait de son déploiement plus restreint, mais l'explication la plus simple reste qu'OpenBSD s'est concentré fortement sur la sécurité et l'a réalisée mieux que quiconque.

4. Une analyse plus récente réalisée par John McCormick dans le Tech Republic<sup>55</sup> compare certaines vulnérabilités de Windows et de Linux. Il s'agit d'une analyse intéressante, prouvant que même si Windows NT sort premier au niveau du nombre de vulnérabilités en l'an 2000, Windows 2000 se retrouve au « milieu du classement » (avec quelques systèmes GNU/Linux ayant plus, et d'autres moins de vulnérabilités). Cependant, il apparaît que dans ces derniers, des anomalies de logiciels fonctionnant sous GNU/Linux ont été comptées comme des erreurs du système d'exploitation GNU/Linux, alors que les anomalies dans des applications Windows n'ont pas été considérées comme des anomalies du système d'exploitation Windows. S'il en va ainsi, il ne s'agit pas vraiment d'une comparaison juste. Comme je le disais précédemment, les distributions typiques de Linux comprennent un grand nombre d'applications qu'il faut acheter séparément dans le cas des outils Microsoft.

---

55 <http://www.techrepublic.com>

5. Un autre point de repères est une liste sur le site SecurityPortal <sup>56</sup> reprenant le temps que les constructeurs ont mis pour résoudre les vulnérabilités. Voici leur conclusion:

*Leur table des données pour 1999:*

| <b>Vendeur</b>   | <b>Nombre total de jours pour résoudre</b> | <b>Nombre de vulnérabilités</b> | <b>Nombre de jours/vulnérabilités</b> |
|------------------|--|---------------------------------|---------------------------------------|
| <b>Red Hat</b>   | 348  | 31                              | 11,23                                 |
| <b>Microsoft</b> | 982  | 61                              | 16,1                                  |
| <b>Sun</b>       | 716  | 8                               | 89,5                                  |

En clair, ce tableau utilise une méthode différente de celle du tableau précédent afin de compter des problèmes de sécurité. Des trois systèmes mentionnés, Solaris de Sun a présenté le moins de vulnérabilités, mais il fut celui qui a été le plus lent à répondre aux problèmes de sécurité. Red Hat demeurerait le plus rapide à résoudre les problèmes de sécurité et se plaçait en deuxième position parmi ces trois systèmes. Il est intéressant de noter que le système d'exploitation d'OpenBSD a eu moins de vulnérabilités enregistrées que les autres. En résumé, avoir un système d'exploitation propriétaire ne signifie pas que vous êtes plus sécurisé.

6. Apache possède un meilleur « security record » que l'IIS de Microsoft (comme mesuré par des comptes-rendus de vulnérabilités sérieuses). L'article publié sur ZDNet le 20 juillet 2001, et ayant pour titre « Apache avoids most security woes »<sup>57</sup>, examina les problèmes de sécurité remontant jusqu'à Apache 1.0. Ils découvrirent que l'annonce du dernier problème de sécurité d'Apache remontait à janvier 1997 (un cas où les attaquants à distance pouvaient exécuter un code arbitraire sur le serveur). Un ensemble de problèmes moins sérieux a été trouvé et résolu en janvier 1998 avec Apache 1.2.5. Dans les trois ans et demi qui ont suivi, les seuls problèmes à distance de la sécurité d'Apache ont été une poignée de problèmes de « denial-of-service » et de fuites d'informations (les attaquants pouvaient consulter des fichiers ou des listes de répertoire auxquels ils n'ont pas accès normalement).

<sup>56</sup> <http://www.securityportal.com/>

<sup>57</sup> <http://www.zdnet.com/eweek/stories/general/0,11011,2792860,00.html>

Une vulnérabilité dans IIS est plus dangereuse qu'une vulnérabilité équivalente dans Apache car Apache suit sagement la bonne pratique en matière de sécurité. IIS fut conçu de telle sorte qu'une personne réussissant à prendre le contrôle d'IIS peut s'assurer le contrôle du système entier, exécutant des actions telles que la lecture, la modification ou la suppression de n'importe quel fichier sur le système.

En revanche, Apache est installé par défaut avec très peu de privilèges. Ainsi, même si on prenait le contrôle d'Apache, cela ne donnerait aux attaquants que relativement peu de possibilités. Par exemple, pirater Apache ne donne pas aux attaquants le droit de modifier ou d'effacer la plupart des fichiers. Ceci n'est pas toujours un avantage. Evidemment, un attaquant peut trouver une autre vulnérabilité pour obtenir l'accès complet, mais un système Apache présente plus de résistance à un attaquant qu'un système IIS.

L'article prétend qu'il existe quatre raisons pour lesquelles la sécurité d'Apache est supérieure, trois d'entre elles sont de bonnes pratiques en matière de sécurité. Apache installe peu d'extensions de serveur par défaut (approche « minimaliste »). Tous les composants du serveur sont exécutés en tant qu'utilisateur restreint, et tous les paramètres de configuration sont centralisés (les rendant faciles d'accès pour des administrateurs désirant s'informer sur la situation). Cependant, l'article prétend également qu'une des principales raisons pour lesquelles Apache est plus sécurisé que IIS est l'excellent contrôle de son code source pour les fichiers serveurs importants, une tâche facilitée par le fait d'être un OSS/FS. Nous pourrions donc considérer que l'OSS/FS encourage de bonnes pratiques en matière de sécurité.

La question n'est pas ici de savoir si un programme particulier est invincible (ce qui n'a aucun sens!). L'objectif est de savoir quel système est le plus susceptible de résister à de futures tentatives d'attaques sur base des performances précédentes. Il est certain qu'Apache présente un bien meilleur historique en matière de sécurité que IIS, de sorte que le groupe Gartner décida de faire une recommandation peu commune (décrite ci-dessous).

7. IIS a été attaqué 1.400 fois plus fréquemment qu'Apache en 2001. En outre, les ordinateurs sous « Windows Plus » (Windows avec l'extension Plus) ont été attaqués plus souvent que toutes les versions d'Unix. Le co-fondateur et CEO de SecurityFocus, Arthur Wong, a réalisé une analyse des diverses vulnérabilités et attaques (basées sur des données de

SecurityFocus) dans l'article de février 2002: « RSA: Security in 2002 worse than 2001, exec says »<sup>58</sup>. IIS a été attaqué à 17 millions de reprises, alors qu'Apache n'a été attaqué que 12.000 fois. Il s'agit, certes, d'une comparaison choquante vu qu'il existe (sensiblement) plus de systèmes Apache que de IIS. En l'an 2001, les systèmes Windows ont été attaqués 31 millions de fois tandis que certains systèmes d'Unix étaient attaqués 22 millions de fois. Veuillez vous référer à cet article pour plus d'information.

8. L'ISS de Microsoft possède un si faible niveau de sécurité qu'en septembre 2001, le groupe Gartner a recommandé aux entreprises<sup>59</sup> ayant été touchées par les virus Nimda et Code Red d'analyser immédiatement les alternatives à IIS, en ce compris une possible migration de leurs applications Web vers des logiciels de serveurs Web d'autres vendeurs tels iPlanet et Apache. Quand bien même ces serveurs Web ont nécessité des patches de sécurité, ils possèdent de bien meilleurs historiques de sécurité que IIS et ne sont pas soumis aux attaques répétées d'un grand nombre de virus et de vers.

Dans un document, le Gartner Group<sup>60</sup> discute des futurs impacts du Code Red. En juillet 2001, Computer Economics (une société de recherche) estimait la dépense des entreprises du monde entier à 1,2 milliards de dollars pour réparer les failles potentiellement exploitables dans leur système par Code Red. Pour rappel, Code Red est conçu pour s'attaquer uniquement aux systèmes IIS, les systèmes tels qu'Apache étant immunisés. Pour être totalement objectif, Gartner a correctement noté que le problème n'est pas simplement qu'IIS présente des vulnérabilités. Une partie du problème demeure que les entreprises utilisant IIS ne conservent pas leur sécurité à jour. Gartner s'est alors ouvertement demandé pourquoi il en était ainsi. Toutefois, Gartner pose également la question suivante: « Pourquoi les produits logiciels de Microsoft continuent-ils à fournir des ouvertures aisément exploitables pour de telles attaques? » Cette question était prémonitoire, puisque peu après, le virus Nimba attaquait aussi ISS, Microsoft Outlook ainsi que d'autres produits Microsoft

9. Les virus informatiques sont plus répandus sur Windows que sur n'importe quel autre système. L'infection par des virus coûta une somme très importante aux utilisateurs des différents Windows de Microsoft. Selon une récente estimation, le virus « The LoveLetter »

---

58 <http://www.cnn.com/2002/TECH/internet/02/25/2002.security.idg/index.html>

59 <http://news.com.com/2009-1001-273288.html?legacy=cnet&tag=nbs>

60 <http://www.gartner.com/DisplayDocument?id=336339>

coûta à lui seul 960 millions de dollars en coûts directs et 7,7 milliards de dollars dans la productivité perdue. De plus, les ventes de logiciels anti-virus se montent à presque un milliard de dollars annuellement. (« Analysis of the Impact of Open Source Software »<sup>61</sup> du Dr Nic Peeling et du Dr Julian Satchell inclut une analyse des diverses sources de données sur les comptages de virus dont voici un extrait.)

[ ... ] Pourquoi les Windows sont-ils vulnérables d'une façon si disproportionnée? Il existe trois raisons à cela, une sociale et deux techniques. Windows est de loin la cible la plus attrayante pour des auteurs de virus simplement parce que son utilisation est largement répandue. Pour propager un virus, celui-ci doit se transmettre à d'autres ordinateurs susceptibles de le recevoir. En moyenne, chaque infection doit causer au moins une infection supplémentaire. Le grand nombre de machines sous Windows permet d'arriver plus facilement à ce seuil. Par ailleurs, Microsoft a fait un certain nombre de choix de conception au cours des années (par exemple l'exécution des macros de démarrage dans Word, l'exécution des pièces jointes dans Outlook, le manque de protection en écriture dans les répertoires systèmes dans Windows 3.1/95/98) et a ainsi permis l'exécution de code non fiable, et ceci a fait de lui une cible très facile.[...]

Bien qu'il soit possible d'écrire un virus pour des systèmes OSS/FS, leur conception rend la propagation des virus plus difficile. Il s'avère que les développeurs d'OSS/FS tendent à opter pour des choix de conception limitant les dommages des virus (peut-être, en partie, parce que leur code est sujet à l'inspection et aux commentaires publics). Par exemple, les OSS/FS ne supportent généralement pas les macros au démarrage ni l'exécution des pièces jointes aux mails pouvant être contrôlés par les pirates. Les systèmes d'exploitation OSS/FS principaux (tels que GNU/Linux et \*BSDs) ont toujours été protégés en l'écriture au niveau des répertoires système.

---

61 [http://www.govtalk.gov.uk/documents/QinetiQ\\_OSS\\_rep.pdf](http://www.govtalk.gov.uk/documents/QinetiQ_OSS_rep.pdf)

### **3.2.6 Pérennité**

La disponibilité des sources garantit aux utilisateurs la pérennité des logiciels qu'ils utilisent. L'abandon du support du logiciel par la société éditrice n'est donc pas à craindre. N'importe qui peut en effet reprendre le code source pour son propre usage selon les limitations de la licence. Ce qui fait que n'importe quel groupe d'utilisateurs intéressé peut reprendre le travail d'évolution d'un logiciel libre; c'est tout à fait possible et assez courant.

### **3.2.7 Indépendance des vendeurs**

En cas de désaccord avec la société qui fournit des développeurs pour ajouter des fonctionnalités à un logiciel, par la disponibilité du code, il est possible grâce à la disponibilité du code de donner ce code à une autre entreprise qui pourra remplacer la précédente. En effet, le code n'étant pas fermé et donc accessible, il est simple de le récupérer et ainsi d'être indépendant de tout fournisseur.

### **3.2.8 Support et formation**

#### L'offre de formation Linux

La base de Linux reste avant tout un système Unix. De ce fait, la conversion des organismes assurant des formations Unix pourra se faire rapidement et sans trop de difficultés. Séparons trois sortes de formation:

- la formation technique des administrateurs;
- la formation des utilisateurs;
- la formation stratégique de l'encadrement.

#### La formation technique des administrateurs

Comme indiqué plus haut, Linux possède une base Unix. De ce fait, pour les techniciens gérant de telles infrastructures, il ne s'agira que d'une simple mise à jour des connaissances.

En ce qui concerne la disponibilité de telles personnes, j'ai pu constater que les grandes écoles formant les gradués et les licenciés en informatique de demain sont déjà pourvues de cours spécifiques à ce sujet. Les personnes compétentes qui sont ou seront disponibles dans un avenir proche n'est donc plus à prouver.

Au niveau des formations spécifiques (formation comme administrateur réseau, de serveurs Web, développeur, ...), il suffit de s'orienter vers les organismes assurant des formations Unix. Dans la plupart des cas, celles-ci dispenseront également des formations Linux.

De plus, le système de développement de GNU/Linux via Internet invite les développeurs à partager leurs connaissances via le web, permettant également un système de formation à moindre coût.

### La formation des utilisateurs

Même si cela sort légèrement du contexte du mémoire, je désire aborder brièvement la formation des utilisateurs car celle-ci me semble primordiale.

Dans le cas d'une utilisation de GNU/Linux uniquement sur des serveurs d'entreprises, une modification des systèmes déjà installés sur les postes de travail des utilisateurs n'est pas nécessaire. En effet, les systèmes d'exploitation sont capables de coexister puisque les informations échangées entre ces systèmes se basent sur des standards (sinon, comment expliquer l'utilisation de serveurs Unix avec des ordinateurs du personnel sous Microsoft Windows). Les utilisateurs ne sont en aucun cas perturbés dans leurs habitudes. Il en ressort qu'aucune formation n'est nécessaire pour les utilisateurs.

Si on désire utiliser également quelques postes sous GNU/Linux, une formation légère est à prévoir. GNU/Linux offrant de nombreuses interfaces graphiques, on peut opter pour celle qui ressemble le plus à MacOS ou celle étant la plus proche de l'environnement de départ pour réduire l'écart opératoire.

### La formation stratégique

En se libérant des licences des anciens systèmes d'exploitation, l'entreprise devra mettre en place une politique d'intervention et de participation dans le mouvement du logiciel libre.

- Les développeurs vont peut-être modifier les logiciels utilisés. Devons nous les laisser remettre en circulation la version modifiée obtenue dans le cadre du travail pour lequel ces salariés sont rémunérés?
- Si l'entreprise a passé un contrat avec un prestataire Linux, dans quelles conditions le personnel de l'entreprise est-il autorisé à se procurer des logiciels complémentaires ou des mises à jour par ses propres moyens?
- Si une campagne de désinformation marketing engendre des pressions sur certains décideurs, serez-vous capable de remettre de l'ordre dans les esprits? En effet, il faut toujours s'attendre, suite à de telles décisions, à des critiques et à des attaques de la part de vos fournisseurs qui pourraient voir leur chiffre d'affaire diminuer. Tout ceci semble naturel mais vous devez bien vous rendre compte que vos intérêts ne correspondent pas toujours à ceux de vos fournisseurs.

### L'offre de support Linux.

Au niveau d'un prestataire de support, les logiciels libres apparaissent comme une aubaine lui permettant de pouvoir réagir rapidement avec ses techniciens. En effet, vu le libre accès aux codes sources, modifier soi-même et très rapidement les programmes reste du domaine du possible, ce qui est d'autant plus satisfaisant pour le client.



Vous trouverez ci-dessous les adresses Internet de quelques sociétés belges, françaises et internationales:

- Mind - Leuven - <http://www.mind.be>
- Arafox - <http://www.arafox.com>
- Alcove - <http://www.alcove.fr>
- Atrid - <http://www.atrid.fr>
- Easter-eggs - <http://www.easter-eggs.com>
- ESR - <http://www.esr.fr>
- Learning Tree International - <http://www.learningtree.com/fr>

Ci-dessous, quelques constructeurs de hardware et de software donnant sur leur site Internet, de multiples informations ou programmes compilés pour Linux:

- 3COM
- Adobe
- Borland
- Bull
- Computer Associates
- Cisco
- Compaq - Hewlett Packard
- Dell
- Fujitsu
- IBM
- Intel
- Lotus
- Oracle
- SAP
- Siemens
- Silicon Graphics (SGI)
- Sun
- Unisys

### 3.2.9 TCO

Le coût total de propriété (TCO) est un paramètre important. Le fait qu'un produit soit bon marché à l'achat importe peu s'il devient fort coûteux par la suite en mises à jour et frais divers.

En fait, quel que soit le produit utilisé ou plébiscité, il est possible de trouver une étude afin de vérifier s'il possède le plus bas TCO en certaines circonstances. Evidemment, les firmes Microsoft et Sun <sup>62</sup> fournissent des études prouvant qu'elles possèdent toutes deux le TCO le plus bas.

En bref, le calcul du plus petit TCO dépend de votre environnement et de vos besoins. Pour déterminer le TCO, vous devez identifier tous les facteurs importants de coût (le modèle du « cost driver ») et estimer leur part. N'oubliez pas les coûts « cachés » comme les coûts de gestion, les coûts de mise à niveau, le support technique et ainsi de suite.

Les OSS/FS possèdent un certain nombre d'avantages en terme de coût dans diverses catégories qui, dans de nombreux cas, auront comme conséquence un plus petit TCO.

1. Les OSS/FS sont meilleur marché en frais d'acquisition mais pas gratuits. Le « free » dans le terme « free software » se rapporte à la liberté, pas au prix. Vous continuerez de dépenser de l'argent pour la documentation, le support, la formation, la gestion du système et ainsi de suite, juste comme vous l'auriez fait avec les systèmes propriétaires. Dans de nombreux cas, les programmes actuels dans les distributions OSS/FS peuvent être acquis gratuitement par téléchargement. Cependant, nombreux sont ceux qui voudront payer une somme modique à un distributeur pour obtenir des « packages » bien intégrés sur CD-ROM, de la documentation sur papier ainsi qu'un support. Il n'empêche que les OSS/FS restent moins chers en termes de coûts d'acquisition.

Observons, par exemple, certaines des différences de prix par essais de configuration d'un serveur. Des objectifs différents impliqueraient des composants différents. Les prix proviennent de « Redcorp.be » et datent de juillet 2002 (arrondis à l'euro près).

---

62 <http://www.sun.com/servers/workgroup/tco/metastudy.html>

Ce tableau en est un exemple.

|                            | Microsoft Windows<br>2000           | Red Hat Linux                                    |
|----------------------------|-------------------------------------|--|
| Système d'exploitation     | € 1.052 (inclut 5 licences clients) | € 223 Professional (nombre de licences illimité) |
| Serveur d'Email            | € 1.346 (inclut 5 licences clients) | inclus (nombre de licences illimité)             |
| Serveur de Base de données | € 1.537 (inclut 5 licences clients) | inclus (nombre de licences illimité)             |
| Développement C++          | € 110                               | inclus   |

Fondamentalement, « Microsoft Windows 2000 » (5 licences clients) présente un coût de 1052 euros. Leur serveur d'email « Microsoft Exchange » (accès pour 5 clients) coûte 1346 euros, leur serveur de base de données « SQL Server 2000 » 1537 euros (avec 5 licences d'accès client), et leur suite de développement Visual C++ coûte 110 euros. Red Hat Linux 7.3 (une distribution largement employée de Linux) affiche un coût de 223 euros pour la version professionnelle (les CD d'applications, les divers guides, le CD d'applications spécial serveur et une assistance téléphonique et online de 60 jours). Quoi qu'il en soit, le coût de la Red Hat version professionnelle inclut toutes les fonctionnalités suivantes: web server, serveur d'e-mail, serveur de base de données, C++, et plus. Un serveur web public avec Windows 2000 et un serveur de base de données pourrait coûter 2589 euros (1052 + 1537) contre 223 euros avec Red Hat Linux, alors qu'un serveur d'intranet avec Windows 2000 et un serveur d'e-mail pourrait coûter 2398 euros (1052 + 1346) contre 223 euros avec Red Hat Linux.

Le système GNU/Linux est toujours livré avec un nombre illimité de licences. Le nombre de clients que vous utiliserez réellement dépendra de vos besoins. Néanmoins, preuve est faite que les produits serveur de Microsoft coûtent des milliers d'euros de plus par serveur que le système équivalent GNU/Linux.

Une autre analyse détaillée comparant les coûts initiaux de GNU/Linux par rapport aux Windows sera développée plus loin sur base de critères en termes d'utilisateurs.

2. Les coûts de mise à niveau sont généralement moins importants, y compris à long terme, pour des systèmes OSS/FS. Par exemple, l'évolution d'un système de Microsoft coûtera généralement environ la moitié de l'achat initial. Le pire serait que l'utilisateur soit à leur merci au niveau du coût à long terme vu que le fournisseur est le seul à pouvoir assurer la maintenance de ses produits (voir Microsoft Turns the Screws<sup>63</sup>). En revanche, les systèmes GNU/Linux peuvent être téléchargés (gratuitement) ou achetés à nouveau pour disposer de la nouvelle version (généralement pour moins de 250 euros). De plus, la mise à niveau simple peut être utilisée sur chaque système n'incluant pas le support technique. Ceci dit, le support technique peut être concurrencé (situation qui n'est pas possible pour le logiciel propriétaire). Si vous n'appréciez pas votre fournisseur GNU/Linux (en raison de leur coût, par exemple), vous avez la possibilité de changer.
3. Les OSS/FS peuvent souvent être utilisés sur un matériel plus ancien plus efficacement que les systèmes propriétaires. Il en résulte des coûts plus faibles en matériel avec suppression possible du besoin en nouveau matériel (parce que les machines inutilisées peuvent reprendre une fonction). Par exemple, les conditions minimales pour Microsoft Windows 2000 Server (selon Microsoft)<sup>64</sup> sont un processeur compatible Pentium (133 mégahertz ou supérieur), 128 MB de RAM minimum (avec 256 MB comme « minimum recommandé »), et un disque dur de 2 GB avec au moins 1,0 GB libre. Selon Red Hat<sup>65</sup>, Red Hat Linux 7.3 exige au minimum un processeur Pentium (200 mégahertz est recommandé), 32MB en mode texte et 64 MB en mode graphique (96 MB recommandé) et un espace de disque dur de 650MB (2,5 GB recommandés).

Dans l'édition d'août 2001 de Scientific American, l'article: « The Do-It-Yourself Supercomputer »<sup>66</sup> explique la manière dont les chercheurs ont établi une plate-forme de calcul puissante avec un grand nombre d'ordinateurs obsolètes, jetés, inutilisés et GNU/Linux.

4. Au fil de l'augmentation du nombre et/ou des performances des systèmes et du matériel, la différence en terme de coûts initiaux et de mise à niveau devient de plus en plus substantielle. Avec l'augmentation du nombre de serveurs, les solutions propriétaires

---

63 <http://osopinion.com/perl/story/9849.html>

64 <http://www.microsoft.com/windows2000/server/evaluation/sysreqs/default.asp>

65 <http://www.redhat.com/software/linux/technical/>

66 <http://www.sciam.com/2001/0801issue/0801hargrove.html>

deviennent plus coûteuses. D'abord, beaucoup de systèmes propriétaires (Microsoft y compris) vendent des licences par client. Cela signifie que même si votre matériel peut supporter un nombre plus important de clients, vous devrez payer plus pour utiliser le matériel acheté. Deuxièmement, si vous désirez utiliser un plus grand nombre d'ordinateurs, vous devrez payer plus de licences aux niveau des systèmes propriétaires. En revanche, en ce qui concerne les distributions GNU/Linux, il est possible d'installer autant de copies que vous le désirez sans aucun surcoût.

Selon Network World Fusion News <sup>67</sup>, Linux est de plus en plus utilisé dans les soins de santé, les finances, les opérations bancaires et dans la vente au détail en raison de ses avantages de coût lors de l'installation d'un grand nombre de sites et de serveurs identiques. Selon leurs calculs, pour un déploiement de 2.000 sites, SCO UnixWare coûterait 9 millions de dollars, Windows 8 millions de dollars et et Red Hat Linux 180 dollars.

5. Il existe beaucoup d'autres facteurs dans le TCO, mais il est difficile de classer leurs effets. Il est généralement difficile aussi de quantifier ces effets. Les personnes pro-windows prétendent qu'il est moins cher et plus facile de trouver un administrateur système pour Windows qu'un administrateur système pour Unix/GNU/Linux. Les personnes pro-GNU/Linux et Unix prétendent que moins d'administrateurs système sont nécessaires par rapport à Windows (du fait de la facilité d'automatisation de la gestion et de la plus grande fiabilité des systèmes). Se mettre en règle au niveau des licences d'exploitation et de l'administration des licences peut s'avérer coûteux pour les systèmes propriétaires (un coût qui n'est pas applicable aux OSS/FS).
6. En de nombreuses circonstances, les économies peuvent être substantielles. Par exemple, une économie excédant 250.000 dollars par an a été enregistrée par 32% des CTO des entreprises interrogés lors d'une enquête menée en 2001 par InfoWorld<sup>68</sup>. 60% de ces CTO ont économisé plus de 50.000 dollars annuellement.

Le 27 août 2001, InfoWorld (en pages 49-50) a fait état d'une étude réalisée sur quarante CTO membres du réseau d'« InfoWorld CTO ». Les CTO ont affirmé utiliser ou projeter d'utiliser des OSS pour serveurs Web (65%), systèmes d'exploitation pour serveur (63%), serveurs d'applications Web (45%), tests de développement d'applications (45%) et systèmes

---

<sup>67</sup> <http://www.nwfusion.com/news/2001/0319specialfocus.html>

<sup>68</sup> <http://www.infoworld.com>

d'exploitation de bureau (38%).

InfoWorld a résumé l'étude de cette manière: « Début 2000, il semblait que le logiciel Open Source n'était pas utilisé pour des tâches critiques... Ainsi, une grande majorité des cadres de sociétés IT utilisent ou vont utiliser les systèmes d'exploitation et les serveurs Web OSS pour leurs applications d'entreprise. »

7. Plusieurs organismes ont enregistré une épargne significative en utilisant OSS/FS. Voici certains exemples de tels organismes:

- l'article « Linux as a Replacement for Windows 2000 »<sup>69</sup> est un exemple d'analyse comparant Red Hat Linux 7.1 à Windows 2000. Dans le cas du client, l'utilisation de Linux au lieu de Windows 2000 a permis d'économiser 10.000 dollars. L'auteur possède un « background » Windows/DOS. Après la réalisation d'un projet Linux intensif basé sur une administration du système à partir de shells et de lignes commande pendant plusieurs mois, cet article signale que « vous serez étonné de ce que peut vous rapporter chaque dollar investi dans un logiciel Open Source ».
- le Vice-Président IT d'Intel, Doug Busch, a enregistré une économie de 200 millions de dollars en remplaçant ses serveurs Unix coûteux par des serveurs meilleur marché exécutant GNU/Linux.
- le site Amazon.com a pu économiser 17 millions de dollars<sup>70</sup> dans des dépenses technologiques pendant un trimestre en grande partie en raison d'un changement pour Linux. Amazon a dépensé 54 millions de dollars pour la technologie et les dépenses de contenu durant son troisième trimestre comparés aux 71 millions de dollars pendant le même trimestre l'année précédente. Par ailleurs, les cadres s'attendaient à un coût technologique plus élevé en raison d'une diminution de 20% d'une partie des ventes nettes cette année.

L'étude du TCO de Microsoft n'est pas utile comme point de départ pour établir le TCO de n'importe quelle entreprise. Leur étude illustre le TCO moyen des sites utilisant les produits Microsoft comparé aux TCO moyens des sites utilisant les systèmes Sun, mais bien que les systèmes Microsoft coûtent 37% moins cher lors de l'acquisition, le système Solaris manipule de

---

69 <http://www.robval.com/linux/desktop/index.asp>

70 <http://news.com.com/2100-1001-275155.html?legacy=cnet&tag=owv>

plus grandes bases de données, plus d'applications émettant des requêtes vers ces bases de données, 63% plus de connexions simultanées, et 243% plus de « hits » par jour. En d'autres termes, les systèmes Microsoft qui effectuent moins de travail sont également moins chers. Il ne s'agit pas d'un point de départ intéressant si nous utilisons le TCO comme aide pour déterminer le système à acheter. Pour réaliser une comparaison valable via le TCO, nous devons comparer les TCO des systèmes exécutant tous les deux le travail motivant l'achat de ce matériel. Une analyse en deux parties par Thomas Pfau (référez-vous aux parties 1<sup>71</sup> et 2<sup>72</sup> ) examine ce matériel ainsi que beaucoup d'autres imperfections dans cette étude.

Une fois de plus, le TCO importe, et non certaines catégories de coût. Au vu de ces différences, dans de nombreuses situations, les OSS/FS possèdent un plus petit TCO que les systèmes propriétaires. Par ailleurs, l'installation d'OSS/FS réclame plus de temps mais, de nos jours, certains systèmes OSS/FS peuvent être achetés pré-installés. Certains prétendent que les coûts d'administration du système sont plus élevés mais des études comme celle de Sun suggèrent que dans de nombreux cas, les coûts d'administration sont inférieurs pour les systèmes « Unix-like » (ou au moins pour les systèmes Sun). Par exemple, sur les systèmes « Unix-like », automatiser des tâches semble plus facile vu la possibilité d'éviter d'utiliser un GUI (Graphical User Interface). Ainsi, avec le temps, beaucoup de tâches manuelles peuvent être automatisées (réduisant de ce fait le TCO).

En bref, il est souvent difficile de compenser les coûts déraisonnablement élevés d'une solution propriétaire par leurs avantages propres quand il existe un produit OSS/FS stable concurrent pour la fonction donnée.

Les OSS/FS ont-ils donc toujours le plus bas TCO? Non! Comme mentionné à de nombreuses reprises dans ce travail, tout dépend de leur utilisation. Pourtant, l'idée que les OSS/FS possèdent toujours un TCO plus élevé est erronée.

---

71 <http://fud-counter.nl.linux.org/tech/TCO.html>

72 <http://fud-counter.nl.linux.org/tech/TCO2.html>

### **3.3 Avantages *apparents* des logiciels propriétaires**

#### Investissements et importance de l'infrastructure ICT existante

Des investissements sont en phase de concrétisation. De grandes sommes d'argent entourent généralement de telles dépenses. Ainsi, remettre en question le temps de préparation, les nombreuses « heures humaines » pour préparer et parfois former le personnel à un changement d'habitudes de travail n'est pas chose évidente. De plus, les dépenses consenties pour les mises en place de projets resteraient alors sans rendement réel vu que l'ensemble de la politique a, jusqu'à présent, été centré sur les logiciels actuellement utilisés sur le lieu de travail et généralement propriétaires.

Ce cas de figure assez fréquent pose la question de l'investissement en général et des conséquences de celui-ci sur la politique future de l'entreprise.

#### Engagements contractuels

Considérons à présent le cas des accords pré-établis ou des contrats de maintenance, de consultance, de mise à jour passés avec les fournisseurs.

Ces engagements jouent un rôle primordial au sein de n'importe quelle entreprise puisqu'ils garantissent le recours à une tierce personne. Cela englobe également tout ce qui se rapporte à l'utilisation des logiciels. En effet, les négociations pour le prix des licences sont souvent liées à des conditions d'utilisation et à des contrats annexes ne pouvant être aisément rompus.

Dans un cas comme dans l'autre, l'entreprise est plus difficilement attirée vers les logiciels libres ou ne fait pas le pas de changer, parce que cela engendrerait soit des conséquences judiciaires, soit des problèmes dans le cadre de futures négociations avec de tierces personnes.

#### Crainte de la réduction de budget

Il s'agit d'un sujet un peu tabou mais il ne peut être ignoré. Selon de nombreuses interviews réalisées auprès d'un grand nombre de responsables informatiques, les budgets informatiques sont,



selon eux, alloués après de nombreuses négociations avec la direction. Le département informatique désire (au maximum) pouvoir compter sur des budgets importants. Par ailleurs, les logiciels propriétaires permettent à ces départements de pouvoir utiliser cet argent et, ainsi, de pouvoir garantir des budgets relativement égaux pour l'année suivante. De plus, on ne critiquera jamais quelqu'un pour avoir choisi un produit qui fait partie des produits dominants du marché.

Cela s'explique par une sécurité d'excuse toute trouvée pour le département informatique responsable de la politique de l'entreprise.

D'un coté, il assure ses subsides et, de l'autre, il ne pourra faire l'objet de critiques parce qu'il ne prend pas le risque d'opter éventuellement pour une solution moins populaire bien que meilleure mais qui, en cas d'échec, risquerait de mettre à mal la capacité du département à réaliser des choix stratégiques.

### Manque de systèmes préinstallés

Dans de nombreux cas actuels, principalement sur les stations de travail, les ordinateurs arrivent avec un système d'exploitation et/ou des applications pré-installées sur le disque dur ou ces applications sont réalisées sur site par des spécialistes du vendeur.

Cette pratique est courante de nos jours mais trop peu de sociétés vendant du hardware, des ordinateurs pré-construits ou des solutions de connectivité installent des logiciels libres. Progressivement, nous voyons apparaître, dans les magazines spécialisés, des publicités pour de tels produits.

Face à un tel déséquilibre, la société cliente ne voit souvent pas l'intérêt de tout désinstaller voire même d'installer tout par ses propres moyens alors qu'elle peut disposer de solutions toutes prêtes et installées par les soins de personnes extérieures présentant deux qualités:

- elles ne dépendent pas de l'entreprise cliente et, donc, ne font pas perdre de temps à des employés en interne qui peuvent dès lors vaquer à d'autres occupations.
- l'entreprise vendeuse est payée à titre forfaitaire ou quasi-forfaitaire. S'il se présente un problème technique (matériel ou autre) lors de l'installation, cela n'aura que peu d'influence sur le prix déboursé par l'entreprise.

Envisager d'autres solutions reviendrait à monopoliser le temps de personnes en interne, possédant un agenda pouvant être assez chargé, et ce sans avoir la moindre idée du temps réel nécessaire pour réaliser le travail.

### FUD (Fear, uncertainty and doubts)

La peur due à la mauvaise connaissance des logiciels libres est un autre avantage en faveur des logiciels propriétaires. Souvent, les logiciels libres sont en position d'« outsiders » sur leur marché face aux solutions propriétaires. L'ensemble amène une crainte (pas nécessairement justifiée) de la part des décideurs à l'intérieur de la société.

En effet, l'entreprise a une bonne connaissance des outils utilisés mais, malheureusement, elle craint ce qu'elle ne connaît pas. Il s'agit d'une réaction tout à fait normale.

Cette réaction implique des doutes et des interrogations. Au delà de ces interrogations se pose le problème du risque. L'entreprise va-t-elle risquer une perte de temps et peut-être d'argent pour des solutions somme toute douteuses et risquées dès le départ?

Il est difficile de répondre mais beaucoup d'entreprises ne franchissent pas le pas; ou il s'agit alors de projets internes aux départements qui sont mis en test à l'intérieur de l'entreprise, parfois à l'insu des décideurs et/ou sans budget, pour ensuite ressortir et être adoptés comme outils de l'entreprise.

### Manque *apparent* de recours

N'importe quelle entreprise ne disposant pas de compétences nécessaires dans son personnel ou même ne désirant qu'assurer le bon fonctionnement de son parc informatique va se mettre en quête de recours (help-desk, hot-line, dépannage rapide,...).

Ces services restent assurés, dans de nombreuses solutions propriétaires, vu que:

- tout le principe du logiciel propriétaire réside dans le fait que le client n'a pas accès aux sources du programmes;
- le client ayant ou n'ayant pas les compétences, est impuissant face au problème qui se pose;
- la demande est importante puisque le logiciel est répandu et souvent situé dans une niche;
- cela apporte parfois un profit supplémentaire pour le fabricant qui se positionne comme le spécialiste de son programme.

En ce qui concerne les logiciels libres, les solutions de recours sont moins nombreuses. En effet, ceux-ci sont développés fréquemment par de nombreux développeurs partout dans le monde. Ce développement ne permet pas souvent de convaincre l'entreprise du caractère contractuel de la relation qu'elle voudrait avoir avec une entité. De fait, l'entreprise de développement auprès de laquelle nous pourrions demander conseil est, pour ainsi dire, inexistante. De plus, peu d'entreprises acceptent de faire de la maintenance pour ces logiciels vu que de nombreux logiciels libres sont régulièrement supplantés par des logiciels propriétaires mieux connus, même si de telles entreprises apparaissent de plus en plus, en raison du marché émergent.

### Problèmes éventuels d'interopérabilité (hardware et software)

Souvent également, le problème d'interopérabilité survient. Nombreux sont ceux qui pensent que le manque de renommée ou le fait de ne pas être « leader » de son marché rend les programmes non compatibles et non standard, et donc pas interopérables.

De même, un doute pourrait survenir quant à l'interaction avec les logiciels installés sur les stations de travail. Il existe un doute quant à savoir s'il sera possible d'utiliser des logiciels libres pour travailler et pour obtenir les informations requises.

Veillez donc vous référer au sous-chapitre 3.2.1 de ce mémoire pour vous rendre compte qu'il ne s'agit pas d'un problème réel mais bien d'un préjugé.

### Ensemble apparemment réduit d'applications orientées pour le secteur d'activité

Ici se pose un problème assez fréquent même les applications sont exécutées en grosse partie sur les stations de travail. En effet, les programmes spécifiques à une profession (CAO, ...) sont généralement très pointus. De ce fait, les alternatives libres ne sont pas toujours suffisantes, quand elles existent.

### Défaut de département marketing et de larges dépenses en ce sens

Comme indiqué précédemment, beaucoup de logiciels libres n'émanent pas de sociétés. De ce fait, le marketing reste inexistant pour de tels logiciels. Ce manque de marketing entraîne une meilleure connaissance des produits propriétaires.

Les logiciels libres sont principalement connus par la qualité de leur code et par son efficacité pour le travail demandé.

Les logiciels propriétaires possèdent un avantage certain vu qu'ils possèdent une approche marketing atteignant directement les entreprises qui ne sont plus enclines à envisager l'usage d'autres logiciels.

## **3.4 Quels sont les apports des logiciels Open Source et propriétaires à l'entreprise?**

### **3.4.1 De multiples avantages...**

Le mode de développement des logiciels libres semble être particulièrement compétitif malgré son apparente anarchie. Il est possible de relever les caractéristiques techniques suivantes vis-à-vis des logiciels de développement Open Source: des fonctionnalités avancées, un maximum de rentabilité des efforts fournis et du temps investi, une amélioration « darwinienne » aboutissant à une efficacité maximum, une très grande fiabilité, un minimum de « bugs », un respect supérieur des standards, une garantie de fonctionnement optimale (en cas de suivi maîtrisé), une liberté de choix supérieure pour les utilisateurs et finalement la pérennité des solutions choisies. Voyons ces différents points plus en détail:

#### Fonctionnalités

Les logiciels libres sont souvent réalisés par des personnes passionnées par un sujet donné, par les fonctionnalités particulières d'un programme, ou par l'utilité du programme pour eux-mêmes. Par

conséquent, ces logiciels disposent de fonctionnalités avancées vu l'intérêt et le désir des personnes impliquées disposer de nouvelles fonctions répondant à leurs besoins et attentes dans leurs domaines respectifs. A contrario, les logiciels propriétaires ont tendance à faire évoluer des techniques déjà présentes ou à venir, les nouveautés en termes de fonctionnalités restant plus restreintes. Les dernières versions de Windows 98 sont, par exemple, encore basées sur Ms-Dos (datant de 1981). Les dernières versions des produits Windows ont été écrites intégralement même si nous retrouvons du code \*BSD caché dans leur code. D'autre part, les logiciels libres sur le marché peuvent servir de base fiable pour les personnes et les sociétés qui désirent ajouter des fonctionnalités spécifiques n'ayant pas encore été couvertes. Ces logiciels peuvent recevoir de nouvelles fonctionnalités n'ayant jamais été introduites dans un logiciel fermé (ou propriétaire) à cause de la trop forte spécificité de la demande.

### Rentabilité

Ce phénomène permet une grande économie de temps et de ressources vu que ceux qui veulent ajouter une nouvelle fonctionnalité peuvent réutiliser tous les codes sources existants. La perte d'énergie dans le développement est minimisée par les fréquentes mises à jour des améliorations. Il est en effet plus simple de repartir d'un logiciel libre existant et de lui ajouter la fonctionnalité recherchée que de repartir à zéro et d'écrire un logiciel complet. Cette démarche a une incidence importante sur la rentabilité de l'entreprise par rapport aux produits commerciaux au code source protégé.

### Efficacité

Par l'ouverture des sources et la possibilité de les modifier, les logiciels libres permettent la contribution de tout volontaire. Ces contributions portent parfois sur d'infimes parties du logiciel et se font par des personnes différentes dans le monde entier, sans autre rapport entre elles que la base de code commune. Ceci permet l'exploration de différentes solutions techniques. La meilleure solution est généralement retenue alors que l'industrie logicielle ne peut se permettre une telle recherche. Sur le long terme, grâce à la sélection naturelle (« darwinienne » pour certains) des solutions techniques, les logiciels libres se révèlent être les plus performants et les plus efficaces parce qu'ils permettent aux utilisateurs de faire du programme ce qu'ils en attendent réellement.

### Fiabilité

Grâce à l'ouverture des sources, tout utilisateur sachant le faire, a la possibilité de lire le code source et de corriger les erreurs éventuelles qu'il peut détecter. Rares sont les erreurs pouvant passer au travers de ce filtrage continu auquel tous les utilisateurs actifs participent. Les logiciels libres atteignent donc une grande fiabilité en un minimum de temps. Si une personne tentait de mettre en place des « backdoors » de sécurité, la probabilité de les découvrir et de le signaler est quasiment certaine vu que ces personnes n'ont aucun lien d'intérêt entre elles. Cette sécurité n'existe pas au niveau des logiciels propriétaires, dont il est possible d'imposer l'usage aveugle à une personne sans que la constatation d'erreur soit relevée.

### Compatibilité avec les standards

Les développeurs de logiciels libres favorisent le respect des standards. En effet, les standards seuls garantissent une parfaite interopérabilité avec d'autres logiciels. Personne n'a intérêt, dans le monde du libre, à utiliser des protocoles incompatibles ou des formats de fichiers non-standards puisque les sources sont ouvertes et qu'il est impossible d'utiliser les techniques de rétention d'informations classiques dans le but de gagner des parts de marché. Les logiciels libres manipulent donc leurs données sous des formats standards permettant de les récupérer et de les traiter avec d'autres logiciels de manière fiable et à moindre coût.

### Garantie de fonctionnement

Professionnellement, la possibilité de modifier les sources garantit le fonctionnement des logiciels. En effet, avec les logiciels propriétaires, les utilisateurs sont totalement dépendants des sociétés éditrices si un problème survient. Les contrats de service classiques sont non seulement chers et, souvent, inefficaces car la correction d'un « bug » est souvent conditionnée à l'attente de la version suivante (et à son achat).

Au contraire, les logiciels libres, plus réactifs, permettent une correction immédiate. De plus, ces logiciels permettent également aux utilisateurs de choisir la solution à leur problème. Ils peuvent donc le résoudre eux-mêmes (s'ils en ont les moyens) ou sinon louer les services d'une société spécialisée assurant un fonctionnement optimal. Dans les deux cas, les utilisateurs acquièrent une meilleure assurance du bon fonctionnement de leurs logiciels.

### Garantie de la liberté

La disponibilité des sources garantit en permanence la liberté de tout utilisateur. Il n'est pas possible, dans un logiciel en « open source », d'inclure des fonctionnalités cachées dans le but de restreindre les libertés individuelles ou de collecter des informations sur les utilisateurs (Ce que nous appelons le « spyware »). Du fait du respect des standards, les logiciels libres n'utilisent pas des formats de fichiers non documentés (comme, par exemple, les différentes versions de Microsoft Word) ou des protocoles de communication propriétaires. Ils garantissent donc la libre circulation des informations et la liberté d'expression de chacun et ce quel que soit son équipement. De plus, les logiciels libres proposent régulièrement une ou plusieurs alternatives pour chaque type de logiciels, permettant une liberté de choix à leurs utilisateurs.

### Pérennité

Le fait que les sources soient ouvertes à tout un chacun permet aux utilisateurs d'être certains de toujours pouvoir utiliser, et même modifier leurs logiciels, et cela même si la société qui les a édités les abandonne.

### **3.4.2 ... et quelques limites**

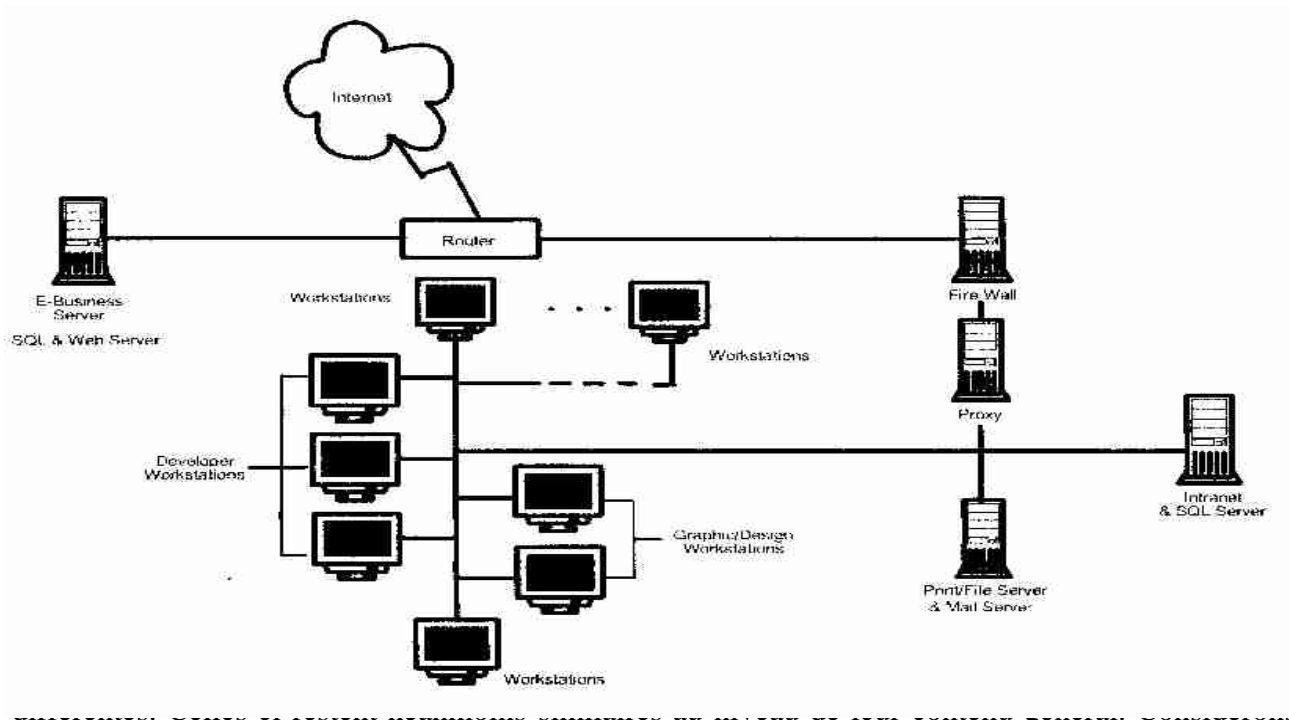
Comme nous avons pu le constater, l'ouverture des sources, l'importance de la base d'utilisateurs et la motivation de ceux-ci sont des éléments fondamentaux dans la rapidité de correction et d'adaptation d'un logiciel libre. Néanmoins, le dernier facteur constitue une limite dans le domaine des logiciels à diffusion réduite (ou hautement spécialisés), dont la faible base d'utilisateurs n'aboutit pas toujours à une amélioration coordonnée et efficace. Si l'ouverture des sources permet un développement personnalisé, la base de départ n'existera pas nécessairement dans le cas de logiciels destinés à un public restreint. Cependant, cela n'empêchera pas le développement d'une version au code source ouvert, par exemple par un prestataire de services spécialisés; mais les avantages du modèle coopératif de développement seront bien moins importants (même s'ils ne sont pas nuls).

Un exemple typique est fourni par les logiciels utilisés dans le monde médical ou bancaire pour lesquels les offres commerciales propriétaires semblent plus adaptées vu les exigences spécifiques de ces secteurs. De même, certains types de logiciels restent encore négligés par les programmeurs

« open source » vu leur relatif manque d'intérêt (comme les logiciels de comptabilité, d'édition sonore ou vidéo ou encore de modélisation 3D) et cela en dépit de l'existence de certaines ébauches (comme le logiciel de modélisation 3D Blender). Le succès du logiciel Apache s'explique également par l'intérêt que lui a porté la communauté des développeurs pour laquelle il constituait un modèle de performance, de sécurité et de configurabilité optimale recherchée par beaucoup dans le cas d'un serveur Internet.

Si le mode de développement des logiciels libres a su conduire à des résultats plus que concluants au niveau technique, comment pouvons-nous accorder notre confiance à des produits qui seront peut-être délaissés par les développeurs après quelques années parce qu'ils ne les intéressent plus?





des organisations standards ayant respectivement 50, 100 et 250 personnes à leur service. Chacune d'elles est équipée d'ordinateurs dont le nombre et l'affectation seront décrits plus loin.

Ne considérons pas que cette comparaison de prix est sans faille. Néanmoins, elle pourrait introduire un débat à propos des prix et pourrait attirer l'attention des lecteurs n'ayant pas encore envisagé cette approche.

Pour cette comparaison, appuyons nous sur une architecture standard telle que le montre le schéma page suivante.

Sources: <http://www.cyber.com.au>

Celui-ci représente le premier cas (avec un personnel s'élevant à 50 personnes). Le second cas présentera deux fois plus de serveurs de fichiers et d'impression. Enfin, le dernier en présentera cinq fois plus.

Supposons que la solution libre et celle de type Microsoft utilisent le même matériel (le prix de

celui-ci étant omis), la même infrastructure de réseau ainsi que les mêmes câblages. On suppose qu'aucun logiciel n'est pré-installé car l'offre fluctue selon le fournisseur. Ces solutions seront installées par des professionnels payés pour installer et pour configurer les systèmes, les serveurs et les stations de travail. Donc, ce coût peut être considéré comme identique de part et d'autre et peut donc être considéré comme ne devant pas intervenir dans le calcul comparatif du coût des deux solutions.

Dans la réalité, il est possible d'utiliser des solutions mixtes même si les vendeurs de Microsoft nous le déconseillent. De plus, de nombreux logiciels libres sont disponibles sous Windows.

Voici donc les produits qui seront installés sur les ordinateurs pour la solution Linux:

|  | Prix  | Commentaires   |
|--|---|--|
| Red Hat 7.3 Pro<br>dans des boîtes officielles | 222,73 €  | Distribution GNU/ Linux standard orientée réseaux et pouvant convenir pour des stations de travail comme pour des serveurs.  |
| Apache (Serveur Web)                           | Inclus dans les distributions GNU/Linux ou téléchargeable gratuitement depuis Internet. | Un serveur web efficace et extensible qui est leader du marché grâce à sa réputation et sa large diffusion sur les serveurs Web des entreprises.                             |
| Squid (Serveur Proxy)                          | Inclus dans les distributions GNU/Linux ou téléchargeable gratuitement depuis Internet. | Un serveur proxy haute performance et particulièrement configurable.   |
| PostgreSQL                                     | Inclus dans les distributions GNU/Linux ou téléchargeable gratuitement depuis Internet. | PostgreSQL est robuste et s'avère être l'une des meilleures bases de données du libre, même si moins connue du grand public que MySQL. Base de données-objets relationnelle. |
| Iptables (Firewall)                            | Inclus dans les distributions GNU/Linux ou téléchargeable gratuitement depuis Internet. | Paquetage puissant et excellent contrôle des filtres.  |
| Sendmail ou Postfix (serveur Mail)             | Inclus dans les distributions GNU/Linux ou téléchargeable gratuitement depuis Internet. | Sendmail est un serveur d'e-mail puissant et flexible occupant 80% du marché des serveur mail sur Internet. Postfix est également un bon serveur mail sécurisé et rapide.    |

|   |   |   |
|---|---|---|
| Kdevelop (Environnement de développement)   | Inclus dans les distributions linux ou téléchargeable gratuitement depuis Internet.               | Un environnement de développement riche qui supporte le C et le C++   |
| The Gimp (Graphisme)                        | Inclus dans les distributions linux ou téléchargeable gratuitement depuis Internet.               | Puissant programme de graphisme et de manipulation d'images de plus en plus célèbre.                                      |
| Open Office (Suite bureautique)             | Inclus dans les distributions linux ou téléchargeable gratuitement depuis Internet.               | Une suite bureautique complète de style Microsoft Office, compatible, fonctionnant sur plusieurs systèmes d'exploitation. |
| The Exchange Project (Système d'e-commerce) | Disponible à l'adresse: <a href="http://theexchangeproject.org">http://theexchangeproject.org</a> | Un serveur facile à gérer grâce à son interface graphique.  |

NB: Vu que Linux est généralement considéré comme sécurisé et immunisé contre les virus et particulièrement contre les virus sous Windows, nous n'ajouterons pas de programme anti-virus spécifique.

En ce qui concerne la solution de type Windows, voici les produits retenus:

|   | Prix        | Fournisseur | Commentaires   |
|---|-------------|-------------|--|
| Norton Anti-virus 2002                              | 39,37 €     | Symantec    |  |
| Microsoft Internet Information Server (serveur Web) | gratuit     | Microsoft   | Inclus dans la version 2000 Server et NT de Windows                          |
| Microsoft Windows 2000 Advanced Serveur             | 4.665,00 €  | Microsoft   | Avec 25 licences   |
| Microsoft Commerce Server                           | 12.407,91 € | Microsoft   | Il s'agit d'une licence par ordinateur incluant un serveur SQL.              |
| Microsoft ISA 2000 Server                           | 1.550,15 €  | Microsoft   | Licence par ordinateur. Inclut le firewall ainsi qu'un serveur proxy.        |
| Microsoft SQL Server                                | 5.028,32 €  | Microsoft   | Licence par ordinateur.  |
| Microsoft Exchange 2000 Std                         | 1.345,74 €  | Microsoft   | 5 licences-utilisateurs. La licence supplémentaire étant à 542 € (pack de 5) |
| Microsoft Windows XP Professionnel                  | 333,69 €    | Microsoft   |  |
| Microsoft Visual Studio 6.0                         | 981,29 €    | Microsoft   |  |
| Microsoft Office XP Standard                        | 540,51 €    | Microsoft   |  |
| Adobe Photoshop                                     | 923,01 €    | Adobe       |  |

### 3.5.1 Scénario 1: 50 utilisateurs

- 45 stations de travail standards;
- 3 stations de travail pour développeurs;
- 2 stations de travail pour graphistes;
- 1 serveur mail;
- 1 serveur de fichiers et d'impression;
- 1 serveur Firewall/Proxy;
- 1 serveur Intranet & serveur SQL;
- 1 serveur E-business (SQL + serveur Web);

#### Coût de la solution Linux

|  | Nombre de copies              | Prix            |
|--|-------------------------------|-----------------|
| Distribution Linux (ex. Red Hat)               | 1 copie nécessaire.           | 222,73 €        |
| Apache (Serveur Web)                           | Fournis avec la distribution. | 0,00 €          |
| Squid (Serveur Proxy)                          | Fournis avec la distribution. | 0,00 €          |
| PostgreSQL                                     | Fournis avec la distribution. | 0,00 €          |
| Iptables (Firewall)                            | Fournis avec la distribution. | 0,00 €          |
| Sendmail ou Postfix (serveur Mail)             | Fournis avec la distribution. | 0,00 €          |
| Kdevelop (Environnement de développement)      | Fournis avec la distribution. | 0,00 €          |
| The Gimp (Graphisme)                           | Fournis avec la distribution. | 0,00 €          |
| Open Office (Suite bureautique)                | Fournis avec la distribution. | 0,00 €          |
| The Exchange Project<br>(Système d'e-commerce) | Téléchargement gratuit.       | 0,00 €          |
|  |                               |                 |
| <b>TOTAL</b>                                   |                               | <b>222,73 €</b> |

Solution Windows:

|   | Nombre de copies | Prix               |
|---|------------------|--------------------|
| Norton Anti-virus 2002                              | 51 copies.       | 2.007,87 €         |
| Microsoft Internet Information Server (serveur Web) | 2 copies.        | 0,00 €             |
| Microsoft Windows 2000 Advanced Serveur             | 5 copies.        | 23.325,00 €        |
| Microsoft Commerce Server                           | 1 copie.         | 12.407,91 €        |
| Microsoft ISA 2000 Server                           | 1 copie.         | 1.550,15 €         |
| Microsoft SQL Server                                | 1 copie.         | 5.028,32 €         |
| Microsoft Exchange 2000                             | 1 copie.         | 1.345,74 €         |
| Microsoft Windows XP Professionnel                  | 50 copies.       | 16.684,50 €        |
| Microsoft Visual Studio 6.0                         | 3 copies.        | 2.943,87 €         |
| Microsoft Office Standard                           | 50 copies.       | 27.025,50 €        |
| Adobe Photoshop                                     | 2 copies.        | 1.846,02 €         |
| Licences complémentaires                            | 45 licences.     | 4.878,00 €         |
|   |                  |                    |
| <b>TOTAL</b>  |                  | <b>99.042.88 €</b> |

### 3.5.2 Scénario 2: 100 utilisateurs

- 95 stations de travail standards;
- 3 stations de travail pour développeurs;
- 2 stations de travail pour graphistes;
- 1 serveur mail;
- 2 serveurs de fichiers et d'impression;
- 1 serveur Firewall/Proxy;
- 1 serveur Intranet & serveur SQL;
- 1 serveur E-business (SQL et serveur Web);

#### Coût de la solution Linux

|  | Nombre de copies              | Prix            |
|--|-------------------------------|-----------------|
| Distribution Linux (ex. Red Hat)               | 1 copie nécessaire.           | 222,73 €        |
| Apache (Serveur Web)                           | Fournie avec la distribution. | 0,00 €          |
| Squid (Serveur Proxy)                          | Fournie avec la distribution. | 0,00 €          |
| PostgreSQL                                     | Fournie avec la distribution. | 0,00 €          |
| Iptables (Firewall)                            | Fournie avec la distribution. | 0,00 €          |
| Sendmail ou Postfix (serveur Mail)             | Fournie avec la distribution. | 0,00 €          |
| Kdevelop (Environnement de développement)      | Fournie avec la distribution. | 0,00 €          |
| The Gimp (Graphisme)                           | Fournie avec la distribution. | 0,00 €          |
| Open Office (Suite bureautique)                | Fournie avec la distribution. | 0,00 €          |
| The Exchange Project<br>(Systeme d'e-commerce) | Téléchargement gratuit.       | 0,00 €          |
|  |                               |                 |
| <b>TOTAL</b>                                   |                               | <b>222,73 €</b> |



Coût de la solution Windows:

|   | Nombre de copies      | Prix                |
|---|-----------------------|---------------------|
| Norton Anti-virus 2002                              | 101 copies.           | 3.976,37 €          |
| Microsoft Internet Information Server (serveur Web) | 2 copies.             | 0,00 €              |
| Microsoft Windows 2000 Serveur                      | 6 copies.             | 27.990,00 €         |
| Microsoft Commerce Server                           | 1 copie.              | 12.407,91 €         |
| Microsoft ISA 2000 Server                           | 1 copie.              | 1.550,15 €          |
| Microsoft SQL Server                                | 1 copie.              | 5.028,32 €          |
| Microsoft Exchange 2000                             | 1 copie.              | 1.345,74 €          |
| Microsoft Windows XP Professionnel                  | 100 copies.           | 33.369,00 €         |
| Microsoft Visual Studio 6.0                         | 3 copies.             | 2.943,87 €          |
| Microsoft Office Standard                           | 100 copies.           | 54.051,00 €         |
| Adobe Photoshop                                     | 2 copies.             | 1.846,02 €          |
| Licences complémentaires                            | 95 licences Exchange. | 10.298,00 €         |
|   |                       |                     |
| <b>TOTAL</b>  |                       | <b>154.806,38 €</b> |

### 3.5.3 Scénario 3: 250 utilisateurs

- 245 stations de travail standards;
- 3 stations de travail pour développeurs;
- 2 stations de travail pour graphistes;
- 1 serveur mail;
- 5 serveurs de fichiers et d'impression;
- 1 serveur Firewall/Proxy;
- 1 serveur Intranet & serveur SQL;
- 1 serveur E-business (SQL et serveur Web);

#### Coût de la solution Linux

|  | Nombre de copies              | Prix            |
|--|-------------------------------|-----------------|
| Distribution Linux (ex. Red Hat)               | 1 copie nécessaire.           | 222,73 €        |
| Apache (Serveur Web)                           | Fournie avec la distribution. | 0,00 €          |
| Squid (Serveur Proxy)                          | Fournie avec la distribution. | 0,00 €          |
| PostgreSQL                                     | Fournie avec la distribution. | 0,00 €          |
| Iptables (Firewall)                            | Fournie avec la distribution. | 0,00 €          |
| Sendmail ou Postfix (serveur Mail)             | Fournie avec la distribution. | 0,00 €          |
| Kdevelop (Environnement de développement)      | Fournie avec la distribution. | 0,00 €          |
| The Gimp (Graphisme)                           | Fournie avec la distribution. | 0,00 €          |
| Open Office (Suite bureautique)                | Fournie avec la distribution. | 0,00 €          |
| The Exchange Project<br>(Système d'e-commerce) | Téléchargement gratuit.       | 0,00 €          |
|  |                               |                 |
| <b>TOTAL</b>                                   |                               | <b>222,73 €</b> |

Coût de la solution Windows:

|   | Nombre de copies       | Prix                |
|---|------------------------|---------------------|
| Norton Anti-virus 2002                              | 251 copies.            | 9.881,87 €          |
| Microsoft Internet Information Server (serveur Web) | 2 copies.              | 0,00 €              |
| Microsoft Windows 2000 Serveur                      | 9 copies.              | 41.985,00 €         |
| Microsoft Commerce Server                           | 1 copie.               | 12.407,91 €         |
| Microsoft ISA 2000 Server                           | 1 copie.               | 1.550,15 €          |
| Microsoft SQL Server                                | 1 copie.               | 5.028,32 €          |
| Microsoft Exchange 2000                             | 1 copie.               | 1.345,74 €          |
| Microsoft Windows XP Professionnel                  | 250 copies.            | 83.422,50 €         |
| Microsoft Visual Studio 6.0                         | 3 copies.              | 2.943,87 €          |
| Microsoft Office Standard                           | 250 copies.            | 135.127,50 €        |
| Adobe Photoshop                                     | 2 copies.              | 1.846,02 €          |
| Licences complémentaires                            | 249 licences Exchange. | 26.558,00 €         |
|   |                        |                     |
| <b>TOTAL</b>  |                        | <b>322.096,88 €</b> |

### 3.5.4 Conclusions

Pour tenter de résumer ces chiffres, voici donc un tableau récapitulatif:

|                              | Solution Microsoft | Solution Linux | Gain avec la solution Linux | Gain/utilisateur |
|------------------------------|--------------------|----------------|-----------------------------|------------------|
| Société A (50 utilisateurs). | 99.042.88 €        | 222,73 €       | 98.820,15 €                 | 1.976,40 €       |
| Société B (100 utilisateurs) | 154.806,38 €       | 222,73 €       | 154.583,65 €                | 1.545,84 €       |
| Société C (250 utilisateurs) | 322.096,88 €       | 222,73 €       | 321.874,15 €                | 1.287,50 €       |

Comme nous pouvons le constater, ces investissements en solutions propriétaires ne sont pas proportionnels au nombre d'utilisateurs. De fait, plus l'entreprise est importante, moins ses dépenses sont importantes par utilisateur. Si on pose l'hypothèse que le chiffre d'affaire est d'autant plus important que le nombre d'utilisateurs croît, l'impact de cette dépense sur le budget de l'entreprise est moindre et l'économie en termes relatifs réalisée en choisissant les logiciels libres plutôt que les logiciels propriétaires est de ce fait diminuée.

|                                 | Solution Microsoft | Moyenne par utilisateur |
|---------------------------------|--------------------|-------------------------|
| Société A avec 50 utilisateurs  | 99.042.88 €        | 1.980,86 €              |
| Société B avec 100 utilisateurs | 154.806,38 €       | 1.548,06 €              |
| Société C avec 250 utilisateurs | 322.096,88 €       | 1.288,39 €              |

Nous pouvons donc conclure qu'il est moins gênant pour les grosses sociétés de s'équiper avec des solutions de type propriétaire si nous n'observons que les coûts et leur importance par rapport au chiffre d'affaire.

## **Chapitre 4: le cas de l'Administration péruvienne**

Ce chapitre fait référence à une proposition de loi péruvienne qui a conduit le responsable de Microsoft Pérou et le député parlementaire à s'échanger de la correspondance à propos de cette idée de faire fonctionner les bureaux d'Etat avec des logiciels libres. Il est intéressant de lire ici de part et d'autre une argumentation très représentative des questions que beaucoup de monde pourrait se poser, ainsi que réponses qui sont celles de pas mal de responsables de logiciels propriétaires par rapport aux logiciels libres. Ces réponses sont assez intéressantes car elles montrent souvent une mauvaise connaissance de ce que sont réellement les logiciels libres, et ce chef d'une grande partie de la population. La correspondance et le projet de loi est disponible à la lecture en annexe.

### **4.1 Contexte**

Une proposition de loi a été déposée par deux membres du parlement péruvien concernant l'usage des logiciels libres dans les agences gouvernementales.

L'utilisation des technologies de l'information, comprenant les logiciels libres, est devenue avec le temps un instrument idéal pour la préservation des données au niveau de l'Etat.

Selon la constitution péruvienne (comme selon la constitution belge), toute personne a le droit de solliciter (section 5 de l'article 2) l'information de n'importe quelle entité publique au cours de la période indiquée par la loi, à un coût raisonnable. L'information qui affecte l'intimité personnelle ou la sécurité nationale et qui est expressément exclue par loi n'est pas sujette à révélation. Après avoir établi cela, il y a un souci évident d'établir les bases institutionnelles qui protégeront la liberté des citoyens quant à l'accès à l'information d'une part et d'autre part à la non-révélation d'informations affectant l'intimité personnelle et familiale ou de sécurité nationale.

La garantie de ces droits n'est pas seulement basée sur la bonne volonté des agents de l'état pour respecter les normes de la constitution, mais également sur l'utilisation des technologies qui contribuent à une protection efficace des droits des citoyens.

L'utilisation du logiciel libre dans toutes les agences de l'état va dans cette direction.

Au départ, on peut dire que les principes fondamentaux qui conduisent la proposition sont étroitement liés aux garanties de base d'un état démocratique que l'on peut résumer ainsi:

- libre accès des citoyens à l'information publique;
- pérennité des données publiques;
- sécurité de l'état et des citoyens.

Quoiqu'il soit correct de dire que l'adoption du logiciel libre n'est pas nécessaire pour respecter la loi, son utilisation réduira considérablement les installations irrégulières de logiciels propriétaires.

Beaucoup de pays prônent l'usage de logiciels libres. La NASA (U.S.A) et la marine des Etats-Unis d'Amérique, entre autres organismes, ont adopté le logiciel libre pour certains de leurs besoins, comme d'autres gouvernements et entités privées.

### Analyse coûts-avantages

Même s'il est vrai que le logiciel libre en comparaison avec le logiciel propriétaire représente une épargne substantielle pour l'Etat, ce n'est pas le point principal pour lequel son utilisation est discutée dans cette proposition. En effet, ici, une meilleure protection des droits des citoyens constitue un avantage non-mesurable qui doit être pris en considération d'un point de vue analyse coûts-avantages.

Nous pouvons résumer les avantages du projet dans les matières suivantes:

- Sécurité Nationale

L'Etat, pour continuer ses activités, doit stocker et traiter l'information liée aux citoyens. Le rapport entre l'individu et l'Etat dépend de la confidentialité et de l'intégrité de ces données, qui doivent être préservées de trois risques spécifiques:

- le risque de révélation: les données confidentielles doivent être manipulées de telle manière que leur accès ne soit rendu possible qu'aux personnes et aux établissements autorisés.

- le risque d'impossibilité de l'accès: les données doivent être stockées de manière telle que l'accès à celles-ci par les personnes et les établissements autorisés soit garanti pour toute leur période d'utilité.
- le risque de manipulation: le changement des données doit être restreint, et seulement possible à les personnes autorisées à le faire.

Avec le logiciel libre, tous ces risques sont considérablement atténués. Il permet à l'utilisateur de faire une inspection complète et approfondie des mécanismes qui sont employés pour traiter des données. Le fait que le logiciel libre permet l'inspection de ses sources est une excellente mesure de sécurité parce qu'avoir ses mécanismes exposés aux yeux des professionnels qualifiés rend plus difficile la dissimulation de fonctions malveillantes, même si l'utilisateur ne prend pas le temps de les rechercher.

- L'indépendance technologique

Avec le logiciel propriétaire, il n'y a aucune liberté de contrat pour les extensions et corrections du système en service; une dépendance technologique est forgée, dans laquelle le fournisseur est en position de dicter ses conditions unilatéralement.

Le logiciel libre donne le droit à l'utilisateur de contrôler et modifier le programme pour l'améliorer selon ses besoins. Cette liberté n'est pas uniquement applicable aux programmeurs, car bien qu'ils soient incapables de modifier le programme par eux-mêmes, les utilisateurs en bénéficient considérablement aussi, parce qu'ils peuvent employer n'importe quel programmeur (pas nécessairement l'auteur original) pour corriger des erreurs ou pour ajouter une fonctionnalité.

- Développement local

Le logiciel propriétaire donne à l'utilisateur la liberté d'exécuter le programme mais pas celle de l'inspecter ou de le modifier, de sorte que l'utilisateur soit dépendant d'une technologie qu'il ne comprend pas et qu'il n'a pas la possibilité d'apprendre à connaître. Les professionnels qui pourraient aider l'utilisateur à rencontrer ses buts sont même limités. Puisque le fonctionnement du programme est secret et son inspection interdite, il n'est pas possible d'aider efficacement l'utilisateur. Les professionnels locaux voient donc leur potentiel à offrir de la valeur ajoutée considérablement limité, et leurs perspectives d'emplois à long terme diminuées. Le logiciel libre neutralise considérablement ces inconvénients spécifiques du logiciel propriétaire.

- Coût du logiciel

Il est considérablement réduit, parce qu'il n'y a plus aucun besoin de demander des licences additionnelles pour continuer à utiliser le programme, au contraire de ce qui se produit avec le logiciel propriétaire. Il est important que l'utilisateur puisse garder ses coûts sous contrôle, sinon il pourrait être empêché d'atteindre ses buts à cause de dépenses non-planifiées.

- Plus de possibilités d'emplois

L'utilisation de logiciels libres permettra de personnaliser ces logiciels en interne ou de faire appel à des programmeurs locaux, ce qui sera générateur d'emplois. Ceci n'est évidemment pas possible avec les logiciels propriétaires. Mettre des personnes au chômage par suite de la dépendance technologique de l'Etat au logiciel propriétaire n'est plus vrai avec les logiciels libres. Au contraire, des professionnels locaux seront assignés à la maintenance du logiciel libre.

- Coûts

Le coût de changement pour le logiciel libre est limité aux coûts liés au procédé de migration. Même s'il est vrai que le procédé de migration implique des coûts d'étude, d'implémentation, de conversion de données, de formation du personnel (et par la suite, des dépenses dans le développement et des dépenses en temps), il est néanmoins certain que ce ne sont que des coûts fixes et non récurrents. Le logiciel propriétaire a ses coûts aussi, qui ont été payés et qui ne peuvent pas être récupérés. Hormis ces coûts, il y a d'autres implications du logiciel propriétaire: la nécessité constante de mises à jour. Ces coûts sont permanents, et au fil du temps, excèdent les coûts fixes initiaux de la migration.



## 4.2 Réaction de Microsoft

Face à cette proposition de loi, la branche péruvienne du géant de l'informatique Microsoft a adressé une lettre au parlementaire péruvien Edgar Virranueva Nuñez pour lui signifier certaines erreurs d'appréciation sur l'étendue des conséquences qu'une telle proposition de loi pourrait avoir.

Voici résumées les principales remarques du responsable de Microsoft Pérou, Juan Alberto González.

- 1. Ainsi, en obligeant l'État à favoriser un modèle de commerce qui s'appuie exclusivement sur le logiciel à code source ouvert, le projet ne fera que décourager les sociétés de fabrication locales et internationales qui sont celles qui réalisent les véritables investissements, créent un nombre significatif d'emplois directs et indirects et contribuent au PIB contrairement à un modèle de logiciel à code source ouvert qui tend à avoir un impact économique toujours moindre du fait qu'il crée principalement des emplois de service.
- 2. Le projet de loi impose l'utilisation de logiciel à code source ouvert sans considérer les dangers que ceci peut entraîner d'un point de vue de la sécurité, de la garantie et des possibles violations des droits de propriété intellectuelle de tiers.
- 3. Le projet utilise de manière erronée les concepts du logiciel à code source ouvert, qui n'est pas nécessairement du logiciel libre ou de coût nul, aboutissant à des conclusions équivoques sur les économies pour l'État, sans une analyse des coûts et bénéfices pour étayer votre position.
- 4. L'un des arguments derrière le projet de loi est la prétendue gratuité du logiciel à code source ouvert, comparée au coût du logiciel commercial, sans tenir compte du fait qu'il existe des modalités de licence en volume qui peuvent être très avantageuses pour l'État, comme cela se fait dans d'autres pays.
- 5. De plus, l'alternative adoptée pour le projet est clairement plus coûteuse du fait des coûts élevés de la migration logicielle, et met en péril la compatibilité et la possibilité d'interopérabilité des plates-formes informatiques au sein de l'État, et entre l'État et le secteur privé, compte tenu des centaines de versions de logiciel à code source ouvert sur le marché.

### 4.3 Réponse du député

Suite à la lettre de Mr Alberto González, le député Villanueva Nuñez répondit à chaque accusation, parvenant à démontrer tous les arguments de son détracteur.<sup>74</sup>

Je vous livre ici les principales réponses aux accusations.

1. [...] Le modèle des services, adopté par un grand nombre d'entreprises de l'industrie informatique, est bien plus significatif, en termes économiques, et de façon croissante, que le commerce de licences sur les programmes.

D'un autre côté, le secteur privé dispose de la plus grande liberté pour choisir le modèle économique qui convient le mieux à ses intérêts, même si cette liberté de choix est souvent obscurcie de manière subliminale par les investissements disproportionnés dans le marketing des producteurs de logiciel propriétaire. [...]

[...]Le logiciel libre crée des emplois techniquement plus qualifiés et on génère un cadre pour la libre concurrence où le succès n'est limité que par la capacité d'offrir du bon support technique et de la qualité de service, on stimule le marché et on enrichit le patrimoine commun de la connaissance, en ouvrant des alternatives pour générer des services de grande valeur ajoutée et de meilleur profil de qualité profitant à tous les acteurs : producteurs, prestataires de services et consommateurs. [...]

2. [...]il est bien connu que le logiciel (propriétaire ou libre) contient des erreurs de programmation ou « bugs » (en jargon informatique) dans ses lignes de code. De même, il est de notoriété publique que les bugs dans le logiciel libre sont moins nombreux, et qu'ils sont réparés bien plus rapidement que dans le logiciel propriétaire. Ce n'est pas en vain que de nombreux organismes publics responsables de la sécurité informatique des systèmes d'institutions de l'État dans les pays développés recommandent l'utilisation de logiciel libre dans des conditions égales de sécurité et d'efficacité.

Il est impossible de prouver que le logiciel propriétaire est plus sûr que le libre, sauf par un examen détaillé, public et ouvert, par la communauté scientifique et les utilisateurs en général. Or, cette démonstration est impossible parce que le modèle même du logiciel propriétaire interdit cette analyse, si bien que la garantie de sécurité se base sur la parole ambiguë (mais

---

74 [http://www.pimientolinux.com/peru2ms/villanueva\\_to\\_ms\\_fr.html](http://www.pimientolinux.com/peru2ms/villanueva_to_ms_fr.html)

vraisemblablement partielle) du producteur du logiciel ou de ses contractants.

Il faut se souvenir que, dans de nombreux cas, les conditions de la licence incluent des clauses de « Non-Disclosure » [NdT : Non Dévoilement] qui interdisent aux utilisateurs de révéler ouvertement les failles de sécurité découvertes dans le produit propriétaire sous licence.

Respect de la garantie :

Comme vous le savez parfaitement, ou pourrez le découvrir en lisant le « Contrat de Licence pour l'Utilisateur Final » [NdT : EULA] des produits dont vous commercialisez la licence, dans la très large majorité des cas, les garanties sont limitées au remplacement du support de distribution s'il est défectueux, mais en aucun cas elles ne prévoient de compensations pour les dommages directs ou indirects, manque à gagner, etc. si suite à un bug de sécurité dans un quelconque (sic) de vos produits, non réparé par vous, un attaquant parvenait à compromettre des systèmes cruciaux pour les services de l'État : quelle garantie, quelles réparations ou quelles compensations donneraient votre société en accord avec les conditions de votre licence? Les garanties du logiciel propriétaire, comme les programmes sont livrés « AS IS » [NdT: tel quel], ce qui veut dire dans l'état dans lequel ils se trouvent, sans aucune responsabilité additionnelle pour le fournisseur concernant sa fonctionnalité, ne diffèrent aucunement de celles habituelles dans le logiciel libre.

Sur la propriété intellectuelle :[...]

[...]Le modèle du logiciel libre n'implique en aucune façon l'ignorance de ces lois et en fait, la grande majorité du logiciel libre est couverte par le copyright. [...]

1. [...]Cette remarque est fautive, en principe la gratuité et la liberté sont des concepts orthogonaux : il y a du logiciel propriétaire et onéreux (par exemple, MS Office), du logiciel propriétaire et gratuit (MS Internet Explorer), du logiciel libre et onéreux (distributions RedHat, SuSE, etc. du système GNU/Linux), du logiciel libre et gratuit (Apache, OpenOffice, Mozilla), et du logiciel sous différentes modalités de licence (MySQL).

Il est certain que le logiciel libre n'est pas nécessairement gratuit. Et le texte du projet ne dit pas qu'il doit l'être comme vous l'aurez bien noté après l'avoir lu. Les définitions incluses dans le projet déterminent clairement « quoi » considérer comme logiciel libre, sans jamais faire référence à la gratuité. Bien qu'il soit fait mention des économies réalisées en terme de non

paiement des licences de logiciel propriétaire, les fondements du projet mentionnent clairement les garanties fondamentales qui doivent être préservées et la stimulation du développement technologique local. [...]

2. [...]J'ai déjà indiqué que ce qui est en question n'est pas le coût du logiciel, mais les principes de liberté d'information, d'accessibilité et de sécurité. Ces arguments ont été largement traités dans les paragraphes précédents, auxquels je vous prie de vous référer.[...]
3. [...]Le premier argument, celui de la migration qui implique des coûts élevés, est en fait un argument en faveur du projet. En effet, plus le temps passe et plus la migration vers une autre technologie sera onéreuse ; et dans le même temps, les risques de sécurité associés au logiciel propriétaire augmenteront aussi. De cette manière, l'utilisation de systèmes et de formats propriétaires rendra l'État encore plus dépendant des fournisseurs. Au contraire une fois implantée la politique d'utilisation du logiciel libre (implantation qui, certes, a un coût), la migration d'un système vers un autre se fait facilement, puisque toutes les données sont stockées dans des formats ouverts. D'autre part, la migration vers un environnement de logiciel ouvert n'implique pas plus de coûts que celle entre deux environnements distincts de logiciel propriétaire, ce qui invalide complètement votre argument.

Le second argument se réfère à l'interopérabilité des plates-formes informatiques au sein de l'État, et entre l'État et le secteur privé. Cette affirmation démontre une ignorance des mécanismes de fabrication du logiciel libre, qui ne maximise pas la dépendance de l'utilisateur par rapport à une plate-forme donnée, comme c'est habituellement le cas dans le domaine du logiciel propriétaire. Même lorsqu'il existe plusieurs distributions d'un logiciel libre et plusieurs programmes susceptibles d'être employés pour une même fonction, l'interopérabilité reste garantie autant par l'emploi de formats standards, exigé dans le projet, que par la possibilité de créer un logiciel interopérable à partir du code source disponible.[...]

Le 16 juillet 2002, la presse via Internet<sup>75</sup> a publié l'information selon laquelle Bill Gates aurait rencontré le président péruvien Alejandro Toledo et aurait donné gracieusement au alentours de 550.000 dollars en argent, logiciels et en consulting à utiliser par le Pérou pour l'éducation des enfants péruviens et pour la mise en place d'e-government.

---

75 <http://www.theregister.co.uk/content/4/26207.html>

## Conclusions

### Quelle compatibilité entre le monde du « libre » et celui de l'entreprise?

A première vue, les mondes du logiciel libre et de l'entreprise s'opposent. Les entreprises possèdent généralement un point de vue négatif sur le logiciel libre (GNU/Linux en particulier puisque c'est un des logiciels libres les plus connus), même si une évolution sensible est observable depuis peu. Ce changement est lié à plusieurs raisons: la liberté en elle-même (ce qui n'est pas toujours bien acceptée dans le monde de l'entreprise) ainsi que le terme « logiciel libre » (pouvant faire penser à un certain côté « anarchiste »). De plus, l'absence de structure régissant le monde du logiciel libre (d'où son succès comme constaté précédemment et certaines initiatives comme le projet GNU mises à part), peut freiner son adoption à cause de l'apparente absence de garantie. Aucune société de taille respectable ne propose actuellement de solutions à base de logiciels libres sur une grande échelle, la société RedHat n'étant qu'un acteur mineur par rapport à des géants comme Microsoft, Sun ou Oracle. Le côté gratuit, couplé à une absence totale de marketing gêne les décideurs. Ceux-ci placent les logiciels libres au même niveau que les logiciels gratuits (également appelés « freeware ») ne justifiant leur existence que par leur gratuité. Pourtant comme le rappelle la Free Software Foundation, la gratuité d'un logiciel libre n'est qu'un détail et absolument pas sa raison d'être.

Afin de définir une politique logicielle, les entreprises se posent généralement les questions suivantes avant de savoir si elle est compatible ou utile à la communauté économique: quelle maintenance pourrions-nous attendre pour un produit gratuit? Quelle serait la pérennité de ce produit? Quelle qualité technique pourrions-nous en attendre (fonctionnalité, performance, fiabilité)? Quelle serait sa compatibilité avec les standards du marché?

### - la maintenance

Contrairement à une idée reçue, mais fausse, la maintenance des logiciels libres est généralement mieux assurée que celle des logiciels commerciaux. La plupart (surtout les plus importants) possèdent une structure de maintenance organisée pouvant évoluer au fil du temps. Grâce au réseau et à la multiplicité des intervenants, la réactivité aux problèmes qui surviennent est extrêmement grande. Après identification du problème, le système libre GNU/Linux fut le premier à être corrigé (bien avant l'ensemble des systèmes commerciaux). En outre, pour certaines exigences particulières,

il reste toujours possible de recourir à une maintenance personnalisée, adaptée aux besoins de l'entreprise et payante (comme tout service) grâce à la disponibilité des codes sources. En fait, le développement de l'usage des logiciels libres remplace une activité commerciale centralisée (monopole) d'édition dont la protection étouffe à terme le développement économique et technologique, et qui est fort peu créatrice d'emplois, par une activité commerciale de services plus créatrice d'emplois décentralisés et plus concurrentielle.

### - la pérennité

Là aussi, la pérennité sur le long terme peut être assurée par les entreprises elles-mêmes ou par un prestataire de leur choix grâce à la disponibilité des sources. Il est néanmoins possible de penser qu'une large communauté d'utilisateurs tendra d'avantage à pérenniser ses produits. On pourra toujours trouver des éléments capables d'assurer le suivi technique au sein de cette communauté, comme le montre la pérennisation d'anciens systèmes, comme CP/M (système d'exploitation 8 bits antérieur à Dos) ou Amiga, ayant perdu tout intérêt économique. CP/M n'était pas un logiciel libre mais il demeurait assez simple afin de pouvoir l'analyser et en faire des simulateurs. Cette pérennité se fonde sur une masse d'utilisateurs à l'expérience globalement plus importante que celle d'une société commerciale pouvant en outre être amenée à abandonner des produits pour des raisons de stratégie industrielle et à ne plus assurer qu'une maintenance dégradée. La disponibilité des sources est donc un gage de pérennité quel que soit le logiciel.

### - les qualités techniques

En ce qui concerne les qualités immédiates telles que fonctionnalité, performance ou encore fiabilité, les expériences, les tests, les évaluations et les comparaisons publiés montrent que de célèbres logiciels libres font concurrence à leurs adversaires commerciaux et les dépassent souvent de manière très nette. Une récente comparaison de serveurs de bases de données réalisée par IBM a montré un très net avantage en performances des solutions sous Linux RedHat par rapport à celles sous Windows 2000.

### - le respect des standards

Pour ce qui est de la compatibilité avec les standards du marché, les logiciels libres, fondés sur une coopération volontaire, intègrent naturellement le respect des normes les plus ouvertes et les plus répandues. En fait, certaines organisations basées sur le modèle libre, comme l'Internet Society

([www.isoc.org](http://www.isoc.org)), sont créatrices de standards (appelés RFC) maintenant universellement utilisés. À l'inverse, le développement économique du logiciel commercial se fonde très largement sur des guerres de standards de fait (parfois jusque devant les tribunaux comme pour le langage Java) se traduisant par une rétention d'informations et par une instabilité permanente des produits. Nous pouvons craindre un changement constant des formats de représentation non documentés publiquement entraînant, à terme, la perte d'une partie du patrimoine documentaire numérisé. Ainsi le logiciel libre s'avère tout à fait compatible avec le monde de l'entreprise d'un point de vue stratégique.

Pour parler ensuite d'intérêt réel pour une entreprise, il faut inclure l'intérêt économique à cela. Mais suite à une comparaison chiffrée des éléments d'un réseau « typique » d'une entreprise, la preuve a été faite que les solutions libres apportent un gain d'argent réel par rapport aux solutions de type propriétaire.

Pour ces évaluations, nous avons considéré un salaire égal pour les administrateurs systèmes qui vont devoir être engagés pour répondre au travail de gestion de ces infrastructures. Un travail de réflexion s'impose aussi. Ne vaut-il pas mieux chercher un peu plus et avoir un administrateur système GNU/Linux qui ,même si il prend peut être plus de temps pour configurer les scripts d'automatisation de tâches la première fois qu'il automatise les routines, fera gagner plus de temps quotidiennement au moment des sauvegardes de sécurité? De même, si nous mesurons l'économie réalisée en migrant de Microsoft Exchange (si il faut posséder 50 licences-clients) vers une solution libre, cela équivaut au prix d'un serveur. Autant de preuves que les logiciels libres sont d'un intérêt économique réel et non négligeable lorsqu'ils sont en adéquation avec les besoins et les nécessités de l'entreprise.

Cet argent peut donc être réinvesti à l'intérieur même de l'entreprise ou profiter en augmentation du bénéfice d'exploitation.

Le logiciel libre fournit une réelle alternative aux entreprises. Cela lui offre les moyens de se prendre en main grâce à des logiciels qu'elle peut gérer elle-même et qui sont contrôlés de par le monde par des milliers d'yeux. En cas de problème, des personnes peuvent réagir immédiatement et fournir une solution très rapide. Les éditeurs de logiciels propriétaires sont généralement plus lents ou attendent qu'un certain nombre de problèmes ou plaintes arrivent avant de bouger et trouver une réponse.

De plus, les administrateurs systèmes gérant les systèmes Open Source sont des spécialistes. On ne peut pas prétendre connaître GNU/Linux ou un autre OSS/FS et pouvoir le gérer en cliquant sur quelques boutons « oui/non ». Cela passe par une compréhension globale des mécanismes et des fichiers de configuration généralement en mode texte. Tout cela donne aux entreprises une garantie quant à la personne et aux logiciels utilisés.

L'ensemble de des recherches et analyses que j'ai effectuées me font conclure que les logiciels libres sont à considérer avec le plus grand sérieux pour toute entreprise désirant posséder un véritable réseau qui tient la route et non pas une solution toute-boite et peu personnalisable. Une politique informatique à long-terme ne peut s'envisager sans continuité des logiciels utilisés, des personnes utilisées et sans évolution interne en continuité avec ce qui se fait actuellement. Tout cela est possible avec les logiciels libres.

Certains logiciels sont gardés secrètement sous peine de perdre un avantage compétitif vis à vis des concurrents. Mais un contrôle sur le code source doit être possible pour éviter qu'au cas où le fournisseur du logiciel tombe en faillite, l'entreprise se retrouve incapable de pouvoir faire évoluer son logiciel par rapport à de nouvelles nécessités. De ce fait, le code source devrait toujours pouvoir être disponible et modifiable pour tout investissement en logiciels informatiques.

Les logiciels libres n'ont pas pour fonction de tout remplacer. Leur fonction est de donner une autre ampleur à des logiciels qui pourraient profiter à tous et dont les modifications enrichiront tout le monde. C'est un principe de partage. Et tout le monde a à y gagner.



---

## Lexique

- **Benchmark** Logiciel destiné à tester les performances d'un ordinateur ou d'un composant. Les plus célèbres sont conçus par les laboratoires de la société Ziff Davis.
- **Binaire** Les fichiers binaires sont des fichiers exécutables, issus de la phase de compilation d'un programme. Ils permettent, lorsqu'ils sont exécutés, de démarrer une application, comme un traitement de texte, un logiciel de messagerie, etc...
- **Bug** Erreur dans la programmation d'une application (ou d'un OS) qui empêche le bon déroulement dans certains types d'utilisation, parfois jusqu'à provoquer un plantage de la machine. L'expression viendrait d'un insecte tombé dans un des premiers ordinateurs et qui l'aurait déréglé ...
- **Cluster** Ensemble logique de serveurs qui garantissent une haute disponibilité des ressources et une répartition des charges de traitement.
- **Compiler** Pour réaliser une application, on utilise un éditeur de texte pour la saisie du code source, puis un compilateur pour le transformer en instructions binaires compréhensibles par la machine.
- **Cracker** Un cracker désigne un programmeur spécialisé dans le cassage de code, de mots de passes ou de protection de logiciels. Ce pirate informatique cherche aussi à pénétrer les réseaux à des fins malveillantes en accédant aux serveurs d'applications ou aux systèmes d'informations d'une entreprise.
- **CTO** Chief Technology Officer ou responsable technologique en français.
- **Démon** Programme qui tourne sans intervention humaine. Par exemple lpd est un démon qui contrôle le flux des travaux d'impression vers l'imprimante sous GNU/Linux.

- 
- **Distribution** Linux n'est qu'un noyau. Pour en faire un système complet, il est nécessaire de collecter un ensemble de programmes à travers Internet, et de les organiser pour obtenir un système fonctionnel. Des distributions (Red Hat, Mandrake, ...) contiennent le noyau, ainsi que tous les programmes nécessaires au fonctionnement de l'environnement GNU/Linux.
  - **Droits d'accès** Dans un système d'exploitation multi-utilisateurs, il est impératif que les fichiers des utilisateurs soient protégés les uns par rapport aux autres. Les droits d'accès déterminent qui sont les utilisateurs autorisés à lire, à modifier et à exécuter des fichiers.
  - **Editeur de texte** Il s'agit d'un programme qui permet d'éditer des fichiers au format texte, sans formatage ni gestion des polices de caractères. Il permet, par exemple, de modifier des fichiers de configuration ou des programmes sources.
  - **Freeware** Logiciel totalement gratuit.
  - **Gcc** Compilateur C associé au projet GNU.
  - **GNU** Association de programmeurs pour l'écriture et la diffusion de logiciels libres (pour Linux par exemple). Ce sont eux qui distribuent l'éditeur de texte Emacs. Leur projet GNU est en avant le langage Scheme comme langage embarqué universel pour les applications.
  - **Hacker** C'est avant tout un programmeur informatique. Bidouilleur et curieux, le hacker n'a qu'un seul but, faire évoluer ses connaissances et celles des autres. A ne pas confondre avec Cracker! Le hacker n'a pas d'intention malveillante: c'est un spécialiste, un expert en systèmes informatique et réseau, et en principe à tout ce qui touche de près ou de loin aux couches IP. Ce surdoué en informatique est un enfant terrible des systèmes, technologies et langages de l'info et du net!
  - **Hardware** Le hardware qualifie le matériel informatique en général.

- 
- Hits                      Nombre de fois qu'un visiteur a accédé au site Internet ou qu'un visiteur a cliqué sur un lien.
  - How-to                    Il s'agit d'une collection de documents ayant trait à de nombreux sujets très particuliers. Ceux-ci sont généralement rédigés par des bénévoles à travers le monde et font office de mode d'emploi sur l'ensemble des problématiques concernant Linux.
  - ICT                        Information and Communication Technologies ou Technologies de l'information et de la communication.
  - Log                        Les fichiers log sont des comptes rendus de l'activité d'une application, générés automatiquement et à intervalles réguliers. Ils peuvent être consultés à tout moment, notamment pour déceler l'origine d'un problème survenu lors de l'exécution d'un programme.
  - Mainframe                Environnement informatique composé d'un système central et de stations clientes.
  - Middleware                Logiciel intermédiaire traduisant les données échangées entre plusieurs applicatifs afin de garantir l'interopérabilité des applications. Le Middleware fournit une interface d'applications commune de programmation d'applications.
  - NetBIOS                    Protocole pour réseaux locaux d'IBM, repris par Microsoft sous le nom de NetBeui.
  - Noyau                     Le noyau représente le système d'exploitation. Tous les accès au matériel, que ce soit la mémoire ou les périphériques, se font via le noyau. Ce dernier a également la charge de l'ordonnancement des processus, c'est-à-dire la gestion des différentes commandes effectuées.
  - OSS/FS                    Open Source Software/Free Software

- 
- **Paquetage-Package** Ce sont des fichiers d'archive permettant la distribution de logiciels, sous forme de fichiers source et/ou de binaires (ou exécutables). Ils se chargent de la gestion de dépendances logicielles par le biais de bases de données locales sur les packages préalablement installés (RPM, Debian, ...)
  - **Portage** Un portage informatique consiste à réaliser le transfert d'applications ou de systèmes d'information depuis le système d'exploitation sur lequel ils ont été conçus jusqu'à un autre système d'exploitation. Cette procédure nécessite la réécriture de toutes les parties de codes concourant à l'interaction du programme avec l'OS: entrées/sorties, interface graphique, opérations sur les fichiers et communication avec les périphériques, connexions réseaux ...
  - **Portabilité** Possibilité de portage.
  - **POSIX** \* (Portable Operating System for Computer Environment) Norme Unix de l'IEEE, numérotée « 1003.1 », et qui spécifie le noyau du système. La norme POSIX 1-b spécifie les noyaux temps réel. La « 1-c » spécifie les threads. C'était une norme excellente, mais qui fut remplacée par les Spec1170.
  - **Processus** Terme provenant de l'univers UNIX caractérisant une tâche qui est en cours d'exécution (donc chargée en mémoire vive) sur le système informatique.
  - **Root** Aussi connu sous le nom de « super utilisateur », ou administrateur du système, c'est le compte qui a tous les droits sur la machine (à l'opposé des autres utilisateurs dont l'accès est limité aux comptes, ainsi qu'aux fichiers et périphériques qui leur sont autorisés).
  - **RPM** Système de paquetages très puissant, développé par la société Red Hat. C'est actuellement le système de package le plus répandu, mais il existe une alternative chez Debian.

- 
- **Serveur** Ordinateur dédié à l'administration d'un réseau informatique. Il gère l'accès aux ressources et aux périphériques et les connexions des différents utilisateurs. Il est équipé d'un logiciel de gestion de réseau.
  - **Serveur de fichiers** Un serveur de fichiers prépare la place mémoire pour des fichiers.
  - **Serveur d'impression** Un serveur d'impression gère et exécute les sorties sur imprimantes du réseau
  - **Serveur proxy** Ordinateur qui garde en mémoire les sites demandés sur le web par les utilisateurs d'un réseau. Le serveur proxy permet d'accélérer les connections.
  - **Serveur mail** Ordinateur qui gère les messages entrants et sortants des utilisateurs d'un réseau.
  - **Serveur Intranet** Ordinateur qui gère le réseau local et privé (entreprise) qui utilise les technologies d'Internet : Web, ... mais ne s'ouvre pas aux connexions publiques extérieures.
  - **Serveur web** Ordinateur qui gère les technologies d'Internet pour permettre l'accès de personnes extérieures à un certain nombre d'informations que l'entreprise accepte de montrer (généralement par l'intermédiaire d'un site).
  - **Shareware** Logiciel diffusé massivement dont on ne paiera le prix que si on l'utilise régulièrement. Formule d'essai avant achat. Malheureusement, ou heureusement, ne repose que sur l'honnêteté et la bonne foi...
  - **Shell** Interpréteur de commandes des plate-formes UNIX, le shell est la partie la plus externe du système d'exploitation, d'où son nom, c'est l'interface utilisateur du système d'exploitation.
  - **Software** Le software désigne les programmes, les logiciels ou les CD.
  - **Standard** Norme de fait le plus souvent d'origine industrielle.

- **Spyware**                      Logiciel capable de collecter des informations diverses concernant un internaute (adresse électronique, sites visités, logiciels possédés, etc), puis de les diffuser par le biais du réseau jusqu'à son concepteur.
- **TCO**                              Total Cost of Ownership se définit par le coût total de possession englobe les différents facteurs de coût intervenant tout au long du cycle de vie d'un matériel informatique.
- **Uptime**                          Durée entre deux relancement d'un serveur.

## **Table des annexes**

Annexe 1: La définition de l'Open Source du site <http://www.opensource.org>

Annexe 2: La licence GPL

Annexe 3: La licence BSD

Annexe 4: La licence Sun (SCSL)

Annexe 5: Liste non-exhaustive de sociétés utilisant GNU/Linux de façon officielle (<http://www.linux-france.org/article/lbiz-fr/>)

Annexe 6: Liste non-exhaustive de sociétés utilisant des logiciels libres de façon officielle (<http://www.aful.org/solutions/entreprises/users.html>)

Annexe 7: La correspondance et le projet de loi du cas de l'administration péruvienne

**Annexe 1: La définition de l'Open Source du site**

**<http://www.opensource.org>**



# La définition du logiciel libre

Version 1.9

## Introduction

Le logiciel libre ("Open source" en anglais) ne signifie pas simplement le libre accès au code source. Les modalités de distribution des logiciels libres doivent répondre aux critères suivants:

### 1. Redistribution libre et gratuite

La licence ne doit pas restreindre la vente ou la distribution du logiciel libre intégré dans une autre logiciel contenant des programmes de différentes origines. La licence ne doit pas exiger de compensation d'aucune sorte en échange de cette intégration.

### 2. Code source

Le programme doit inclure le code source, et doit autoriser la distribution du code source comme de l'exécutable compilé. Quand une forme quelconque du produit est distribuée sans le code source, il doit être clairement indiqué par quel moyen il est possible d'obtenir le code source, pour une somme qui ne doit pas excéder un coût raisonnable de reproduction, ou en le chargeant gratuitement via Internet. Le code source doit être la forme privilégiée par laquelle un programmeur modifie le programme. Un code source délibérément confus est interdit. Les formes intermédiaires de code source, telles que celles résultant d'un pré-processeur ou d'un traducteur, sont interdites.

### 3. Travaux dérivés

La licence doit autoriser les modifications et les travaux dérivés, et doit permettre leur distribution dans les mêmes termes que la licence du logiciel d'origine.

### 4. Intégrité du code source de l'auteur

La licence peut restreindre la distribution du code source modifié seulement si elle autorise la distribution de patches avec le code source, dans le but de modifier le programme à la compilation. La licence doit explicitement permettre la distribution de logiciels obtenus à partir du code source modifié. La licence peut exiger que les travaux dérivés portent un nom ou un numéro de version différents du logiciel d'origine.

### 5. Absence de discrimination envers des personnes ou des groupes

La licence ne doit pas être discriminative à l'encontre de personnes ou de groupes de personnes.

### 6. Absence de discrimination envers des domaines d'activité

La licence ne doit pas restreindre ni interdire l'usage du logiciel à un quelconque domaine d'activité. Par exemple, il ne peut interdire l'usage du logiciel dans le cadre d'une activité professionnelle, ou en exclure l'usage pour la recherche génétique.

### 7. Distribution de licence

Les droits attachés au programme doivent s'appliquer à tous ceux à qui il est distribué sans qu'il leur soit besoin de se conformer à des termes de licence complémentaires.

### 8. La licence ne doit pas être spécifique à un produit

Les droits attachés au programme ne doivent pas dépendre du fait que le programme fait partie d'un logiciel en particulier. Si le programme est séparé du logiciel dans lequel il est intégré, et utilisé ou distribué selon les termes de la licence, toutes les parties à qui le programme est redistribué doivent avoir les mêmes droits que ceux accordés avec le logiciel dans lequel il est intégré à l'origine.

### 9. La licence ne doit pas imposer de restrictions sur d'autres logiciels

La licence ne doit pas imposer de restrictions sur d'autres logiciels distribués avec le programme sous licence. Par exemple, la licence ne doit pas exiger que les autres programmes distribués sur le même support physique soient aussi des logiciels libres.

## **Annexe 2: La licence GPL**

# The GNU General Public License (GPL)

Version 2, June 1991

Copyright (C) 1989, 1991 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.

## **Preamble**

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## **TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.



This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public 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.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## **NO WARRANTY**

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE

DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

### **How to Apply These Terms to Your New Programs**

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR

A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest  
in the program `Gnomovision' (which makes passes at compilers)  
written by James Hacker.

signature of Ty Coon, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## **Annexe 3: La licence BSD**

## The BSD License

The following is a BSD license template. To generate your own license, change the values of OWNER, ORGANIZATION and YEAR from their original values as given here, and substitute your own.

Note: The advertising clause in the license appearing on BSD Unix files was officially rescinded by the Director of the Office of Technology Licensing of the University of California on July 22 1999. He states that clause 3 is "hereby deleted in its entirety."

Note the new BSD license is thus equivalent to the MIT License, except for the no-endorsement final clause.

<OWNER> = Regents of the University of California

<ORGANIZATION> = University of California, Berkeley

<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used

to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **Annexe 4: La licence Sun (SCSL)**

# SUN COMMUNITY SOURCE LICENSE

Version 2.12

(Rev. Date March 5, 2001)

## RECITALS

Original Contributor has developed Specifications and Source Code implementations of certain Technology; and

Original Contributor desires to license the Technology to a large community to facilitate research, innovation and product development while maintaining compatibility of such products with the Technology as delivered by Original Contributor; and

Original Contributor desires to license certain Sun Trademarks for the purpose of branding products that are compatible with the relevant Technology delivered by Original Contributor; and

You desire to license the Technology and possibly certain Sun Trademarks from Original Contributor on the terms and conditions specified in this License.

In consideration for the mutual covenants contained herein, You and Original Contributor agree as follows:

## AGREEMENT

1. Introduction. The Sun Community Source License and effective attachments (License) may include five distinct licenses: Research Use, TCK, Internal Deployment Use, Commercial Use and Trademark License. The Research Use license is effective when You execute this License. The TCK and Internal Deployment Use licenses are effective when You execute this License, unless otherwise specified in the TCK and Internal Deployment Use attachments. The Commercial Use and Trademark licenses must be signed by You and Original Contributor in order to become effective. Once effective, these licenses and the associated requirements and responsibilities are cumulative. Capitalized terms used in this License are defined in the Glossary.



## 2. License Grants.

2.1. Original Contributor Grant. Subject to Your compliance with Sections 3, 8.10 and Attachment A of this License, Original Contributor grants to You a worldwide, royalty-free, non-exclusive license, to the extent of Original Contributor's Intellectual Property Rights covering the Original Code, Upgraded Code and Specifications, to do the following:

a) Research Use License:

(i) use, reproduce and modify the Original Code, Upgraded Code and Specifications to create Modifications and Reformatted Specifications for Research Use by You,

(ii) publish and display Original Code, Upgraded Code and Specifications with, or as part of Modifications, as permitted under Section 3.1 b) below,

(iii) reproduce and distribute copies of Original Code and Upgraded Code to Licensees and students for Research Use by You,

(iv) compile, reproduce and distribute Original Code and Upgraded Code in Executable form, and Reformatted Specifications to anyone for Research Use by You.

b) Other than the licenses expressly granted in this License, Original Contributor retains all right, title, and interest in Original Code and Upgraded Code and Specifications.

## 2.2. Your Grants.

a) To Other Licensees. You hereby grant to each Licensee a license to Your Error Corrections and Shared Modifications, of the same scope and extent as Original Contributor's licenses under Section 2.1 a) above relative to Research Use, Attachment C relative to Internal Deployment Use, and Attachment D relative to Commercial Use.

b) To Original Contributor. You hereby grant to Original Contributor a worldwide, royalty-free, non-exclusive, perpetual and irrevocable license, to the extent of Your Intellectual Property Rights covering Your Error Corrections, Shared Modifications and Reformatted Specifications, to use, reproduce, modify, display and distribute Your Error Corrections, Shared Modifications and Reformatted Specifications, in any form, including the right to sublicense such rights through

multiple tiers of distribution.

c) Other than the licenses expressly granted in Sections 2.2 a) and b) above, and the restriction set forth in Section 3.1 d)(iv) below, You retain all right, title, and interest in Your Error Corrections, Shared Modifications and Reformatted Specifications.

2.3. Contributor Modifications. You may use, reproduce, modify, display and distribute Contributor Error Corrections, Shared Modifications and Reformatted Specifications, obtained by You under this License, to the same scope and extent as with Original Code, Upgraded Code and Specifications.

2.4. Subcontracting. You may deliver the Source Code of Covered Code to other Licensees having at least a Research Use license, for the sole purpose of furnishing development services to You in connection with Your rights granted in this License. All such Licensees must execute appropriate documents with respect to such work consistent with the terms of this License, and acknowledging their work-made-for-hire status or assigning exclusive right to the work product and associated Intellectual Property Rights to You.

### 3. Requirements and Responsibilities.

3.1. Research Use License. As a condition of exercising the rights granted under Section 2.1 a) above, You agree to comply with the following:

a) Your Contribution to the Community. All Error Corrections and Shared Modifications which You create or contribute to are automatically subject to the licenses granted under Section 2.2 above. You are encouraged to license all of Your other Modifications under Section 2.2 as Shared Modifications, but are not required to do so. You agree to notify Original Contributor of any errors in the Specification.

b) Source Code Availability. You agree to provide all Your Error Corrections to Original Contributor as soon as reasonably practicable and, in any event, prior to Internal Deployment Use or Commercial Use, if applicable. Original Contributor may, at its discretion, post Source Code for Your Error Corrections and Shared Modifications on the Community Webserver. You may also post Error Corrections and Shared Modifications on a web-server of Your choice; provided, that You must take reasonable precautions to ensure that only Licensees have access to such Error Corrections and Shared Modifications. Such precautions shall include, without limitation, a

password protection scheme limited to Licensees and a click-on, download certification of Licensee status required of those attempting to download from the server. An example of an acceptable certification is attached as Attachment A-2.

c) Notices. All Error Corrections and Shared Modifications You create or contribute to must include a file documenting the additions and changes You made and the date of such additions and changes. You must also include the notice set forth in Attachment A-1 in the file header. If it is not possible to put the notice in a particular Source Code file due to its structure, then You must include the notice in a location (such as a relevant directory file), where a recipient would be most likely to look for such a notice.

d) Redistribution.

(i) Source. Covered Code may be distributed in Source Code form only to another Licensee (except for students as provided below). You may not offer or impose any terms on any Covered Code that alter the rights, requirements, or responsibilities of such Licensee. You may distribute Covered Code to students for use in connection with their course work and research projects undertaken at accredited educational institutions. Such students need not be Licensees, but must be given a copy of the notice set forth in Attachment A-3 and such notice must also be included in a file header or prominent location in the Source Code made available to such students.

(ii) Executable. You may distribute Executable version(s) of Covered Code to Licensees and other third parties only for the purpose of evaluation and comment in connection with Research Use by You and under a license of Your choice, but which limits use of such Executable version(s) of Covered Code only to that purpose.

(iii) Modified Class, Interface and Package Naming. In connection with Research Use by You only, You may use Original Contributor's class, interface and package names only to

accurately reference or invoke the Source Code files You modify. Original Contributor grants to You a limited license to the extent necessary for such purposes.

(iv) You expressly agree that any distribution, in whole or in part, of Modifications developed by You shall only be done pursuant to the term and conditions of this License.

e) Extensions.

(i) Covered Code. You may not include any Source Code of Community Code in any Extensions;

(ii) Publication. No later than the date on which You first distribute such Extension for Commercial Use, You must publish to the industry, on a non-confidential basis and free of all copyright restrictions with respect to reproduction and use, an accurate and current specification for any Extension. In addition, You must make available an appropriate test suite, pursuant to the same rights as the specification, sufficiently detailed to allow any third party reasonably skilled in the technology to produce implementations of the Extension compatible with the specification. Such test suites must be made available as soon as reasonably practicable but, in no event, later than ninety (90) days after Your first Commercial Use of the Extension. You must use reasonable efforts to promptly clarify and correct the specification and the test suite upon written request by Original Contributor.

(iii) Open. You agree to refrain from enforcing any Intellectual Property Rights You may have covering any interface(s) of Your Extension, which would prevent the implementation of such interface(s) by Original Contributor or any Licensee. This obligation does not prevent You from enforcing any Intellectual Property Right You have that would otherwise be infringed by an implementation of Your Extension.

(iv) Class, Interface and Package Naming. You may not add any packages, or any public or protected classes or interfaces with names that originate or might appear to originate from Original Contributor including, without limitation, package or class names which begin with sun, java, javax, jini, net.jini, com.sun or their equivalents in any subsequent class, interface and/ or package naming convention adopted by Original Contributor. It is specifically suggested that You name any new packages using the Unique Package Naming Convention as described in The Java Language Specification by James Gosling, Bill Joy, and Guy Steele, ISBN 0-201-63451-1, August 1996. Section 7.7 Unique Package Names, on page 125 of this specification which states, in part:

You form a unique package name by first having (or belonging to an organization that has) an Internet domain name, such as sun.com. You then reverse the name, component by component, to obtain, in this example, Com.sun, and use this as a prefix for Your package names, using a convention developed within Your organization to further administer package names.?

3.2. Additional Requirements and Responsibilities. Any additional requirements and responsibilities relating to the Technology are listed in Attachment F (Additional Requirements and Responsibilities), if applicable, and are hereby incorporated into this Section 3.

#### 4. Versions of the License.

4.1. License Versions. Original Contributor may publish revised versions of the License from time to time. Each version will be given a distinguishing version number.

4.2. Effect. Once a particular version of Covered Code has been provided under a version of the License, You may always continue to use such Covered Code under the terms of that version of the License. You may also choose to use such Covered Code under the terms of any subsequent version of the License. No one other than Original Contributor has the right to promulgate License versions.

#### 5. Disclaimer of Warranty.

5.1. COVERED CODE IS PROVIDED UNDER THIS LICENSE AS IS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. YOU AGREE TO BEAR THE ENTIRE RISK IN CONNECTION WITH YOUR USE AND DISTRIBUTION OF COVERED CODE UNDER THIS LICENSE. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT SUBJECT TO THIS DISCLAIMER.

5.2. You acknowledge that Original Code, Upgraded Code and Specifications are not designed or intended for use in (i) on-line control of aircraft, air traffic, aircraft navigation or aircraft communications; or (ii) in the design, construction, operation or maintenance of any nuclear facility. Original Contributor disclaims any express or implied warranty of fitness for such uses.

#### 6. Termination.

6.1. By You. You may terminate this Research Use license at anytime by providing written notice to Original Contributor.

6.2. By Original Contributor. This License and the rights granted hereunder will terminate:

(i) automatically if You fail to comply with the terms of this License and fail to cure such breach within 30 days of receipt of written notice of the breach;

(ii) immediately in the event of circumstances specified in Sections 7.1 and 8.4; or

(iii) at Original Contributor's discretion upon any action initiated in the first instance by You alleging that use or distribution by Original Contributor or any Licensee, of Original Code, Upgraded Code, Error Corrections or Shared Modifications contributed by You, or Specifications, infringe a patent owned or controlled by You.

6.3. Effect of Termination. Upon termination, You agree to discontinue use and return or destroy all copies of Covered Code in your possession. All sublicenses to the Covered Code which you have properly granted shall survive any termination of this License. Provisions which, by their nature, should remain in effect beyond the termination of this License shall survive including, without limitation, Sections 2.2, 3, 5, 7 and 8.

6.4. Each party waives and releases the other from any claim to compensation or indemnity for permitted or lawful termination of the business relationship established by this License.

## 7. Liability.

7.1. Infringement. Should any of the Original Code, Upgraded Code, TCK or Specifications (Materials) become the subject of a claim of infringement, Original Contributor may, at its sole option, (i) attempt to procure the rights necessary for You to continue using the Materials, (ii) modify the Materials so that they are no longer infringing, or (iii) terminate Your right to use the Materials, immediately upon written notice, and refund to You the amount, if any, having then actually been paid by You to Original Contributor for the Original Code, Upgraded Code and TCK, depreciated on a straight line, five year basis.

7.2. LIMITATION OF LIABILITY. TO THE FULL EXTENT ALLOWED BY APPLICABLE LAW, ORIGINAL CONTRIBUTOR'S LIABILITY TO YOU FOR CLAIMS RELATING TO THIS LICENSE, WHETHER FOR BREACH OR IN TORT, SHALL BE LIMITED TO ONE HUNDRED PERCENT (100%) OF THE AMOUNT HAVING THEN ACTUALLY BEEN PAID BY YOU TO ORIGINAL CONTRIBUTOR FOR ALL COPIES LICENSED HEREUNDER OF THE PARTICULAR ITEMS GIVING RISE TO SUCH CLAIM, IF ANY. IN NO EVENT WILL

YOU (RELATIVE TO YOUR SHARED MODIFICATIONS OR ERROR CORRECTIONS) OR SUN BE LIABLE FOR ANY INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THIS LICENSE (INCLUDING, WITHOUT LIMITATION, LOSS OF PROFITS, USE, DATA, OR OTHER ECONOMIC ADVANTAGE), HOWEVER IT ARISES AND ON ANY THEORY OF LIABILITY, WHETHER IN AN ACTION FOR CONTRACT, STRICT LIABILITY OR TORT (INCLUDING NEGLIGENCE) OR OTHERWISE, WHETHER OR NOT YOU OR ORIGINAL CONTRIBUTOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY REMEDY.

## 8. Miscellaneous.

8.1. Trademark. You agree to comply with the then current Sun Trademark & Logo Usage Requirements accessible through the SCSL Webpage. Except as expressly provided in the License, You are granted no right, title or license to, or interest in, any Sun Trademarks. You agree not to (i) challenge Original Contributor's ownership or use of Sun Trademarks; (ii) attempt to register any Sun Trademarks, or any mark or logo substantially similar thereto; or (iii) incorporate any Sun Trademarks into your own trademarks, product names, service marks, company names, or domain names.

8.2. Integration. This License represents the complete agreement concerning the subject matter hereof.

8.3 Assignment. Original Contributor may assign this License, and its rights and obligations hereunder, in its sole discretion. You may assign the Research Use portions of this License to a third party upon prior written notice to Original Contributor (which may be provided via the Community Web-Server). You may not assign the Commercial Use license or TCK license, including by way of merger (regardless of whether You are the surviving entity) or acquisition, without Original Contributor's prior written consent.

8.4. Severability. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Notwithstanding the foregoing, if You are prohibited by law from fully and specifically complying with Sections 2.2 or 3, this License will immediately terminate and You must immediately discontinue any use of Covered Code.

8.5. Governing Law. This License shall be governed by the laws of the United States and the State

of California, as applied to contracts entered into and to be performed in California between California residents. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded.

#### 8.6. Dispute Resolution.

a) Any dispute arising out of or relating to this License shall be finally settled by arbitration as set out herein, except that either party may bring any action, in a court of competent jurisdiction (which jurisdiction shall be exclusive), with respect to any dispute relating to such party's Intellectual Property Rights or with respect to Your compliance with the TCK license. Arbitration shall be administered: (i) by the American Arbitration Association (AAA), (ii) in accordance with the rules of the United Nations Commission on International Trade Law (UNCITRAL) (the Rules) in effect at the time of arbitration as modified herein; and (iii) the arbitrator will apply the substantive laws of California and United States. Judgment upon the award rendered by the arbitrator may be entered in any court having jurisdiction to enforce such award.

b) All arbitration proceedings shall be conducted in English by a single arbitrator selected in accordance with the Rules, who must be fluent in English and be either a retired judge or practicing attorney having at least ten (10) years litigation experience and be reasonably familiar with the technology matters relative to the dispute. Unless otherwise agreed, arbitration venue shall be in London, Tokyo, or San Francisco, whichever is closest to defendant's principal business office. The arbitrator may award monetary damages only and nothing shall preclude either party from seeking provisional or emergency relief from a court of competent jurisdiction. The arbitrator shall have no authority to award damages in excess of those permitted in this License and any such award in excess is void. All awards will be payable in U.S. dollars and may include, for the prevailing party (i) pre-judgment award interest, (ii) reasonable attorneys' fees incurred in connection with the arbitration, and (iii) reasonable costs and expenses incurred in enforcing the award. The arbitrator will order each party to produce identified documents and respond to no more than twenty-five single question interrogatories.

8.7. Construction. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

8.8. U.S. Government End Users. The Covered Code is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software and commercial computer software documentation, as such terms are used in 48 C.F.R. 12.212 (Sept. 1995).



Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein. You agree to pass this notice to Your licensees.

8.9. Press Announcements. All press announcements relative to the execution of this License must be reviewed and approved by Original Contributor and You prior to release.

8.10. International Use.

a) Export/Import Laws. Covered Code is subject to U.S. export control laws and may be subject to export or import regulations in other countries. Each party agrees to comply strictly with all such laws and regulations and acknowledges their responsibility to obtain such licenses to export, re-export, or import as may be required. You agree to pass these obligations to Your licensees.

b) Intellectual Property Protection. Due to limited intellectual property protection and enforcement in certain countries, You agree not to redistribute the Original Code, Upgraded Code, TCK and Specifications to any country other than the list of restricted countries on the SCSL Webpage.

8.11. Language. This License is in the English language only, which language shall be controlling in all respects, and all versions of this License in any other language shall be for accommodation only and shall not be binding on the parties to this License. All communications and notices made or given pursuant to this License, and all documentation and support to be provided, unless otherwise noted, shall be in the English language.

AGREED TO AND ACCEPTED BY:

You: Original Contributor:

\_\_\_\_\_ Sun Microsystems, Inc.

By: \_\_\_\_\_ By: \_\_\_\_\_

(Your Name)

Title \_\_\_\_\_ Title: \_\_\_\_\_

GLOSSARY

1. Commercial Use means any use (excluding Internal Deployment Use) or distribution, directly or indirectly of Compliant Covered Code by You to any third party, alone or bundled with any other software or hardware, for direct or indirect commercial or strategic gain or advantage, subject to execution of Attachment D by You and Original Contributor.
2. Community Code means the Original Code, Upgraded Code, Error Corrections, Shared Modifications, or any combination thereof.
3. Community Webserver(s) means the webserver(s) designated by Original Contributor for posting Error Corrections and Shared Modifications.
4. Compliant Covered Code means Covered Code that complies with the requirements of the TCK.
5. Contributor means each Licensee that creates or contributes to the creation of any Error Correction or Shared Modification.
6. Covered Code means the Original Code, Upgraded Code, Modifications, or any combination thereof.
7. Error Correction means any change made to Community Code which conforms to the Specification and corrects the adverse effect of a failure of Community Code to perform any function set forth in or required by the Specifications.
8. Executable means Covered Code that has been converted to a form other than Source Code.
9. Extension(s) means any additional classes or other programming code and/or interfaces developed by or for You which: (i) are designed for use with the Technology; (ii) constitute an API for a library of computing functions or services; and (iii) are disclosed to third party software developers for the purpose of developing software which invokes such additional classes or other programming code and/or interfaces. The foregoing shall not apply to software development by Your subcontractors to be exclusively used by You.
10. Intellectual Property Rights means worldwide statutory and common law rights associated solely with (i) patents and patent applications; (ii) works of authorship including copyrights, copyright applications, copyright registrations and moral rights; (iii) the protection of trade and

industrial secrets and confidential information; and (iv) divisions, continuations, renewals, and re-issuances of the foregoing now existing or acquired in the future.

11. Internal Deployment Use means use of Compliant Covered Code (excluding Research Use) within Your business or organization only by Your employees and/or agents, subject to execution of Attachment C by You and Original Contributor, if required.

12. Licensee means any party that has entered into and has in effect a version of this License with Original Contributor.

13. Modification(s) means (i) any change to Covered Code; (ii) any new file or other representation of computer program statements that contains any portion of Covered Code; and/or (iii) any new Source Code implementing any portion of the Specifications.

14. Original Code means the initial Source Code for the Technology as described on the Technology Download Site.

15. Original Contributor means Sun Microsystems, Inc., its affiliates and its successors and assigns.

16. Reformatted Specifications means any revision to the Specifications which translates or reformats the Specifications (as for example in connection with Your documentation) but which does not alter, subset or superset the functional or operational aspects of the Specifications.

17. Research Use means use and distribution of Covered Code only for Your research, development, educational or personal and individual use, and expressly excludes Internal Deployment Use and Commercial Use.

18. SCSL Webpage means the Sun Community Source license webpage located at <http://sun.com/software/communitysource>, or such other url that Sun may designate from time to time.

19. Shared Modifications means Modifications provided by You, at Your option, pursuant to Section 2.2, or received by You from a Contributor pursuant to Section 2.3.

20. Source Code means computer program statements written in any high-level, readable form suitable for modification and development.

21. Specifications means the specifications for the Technology and other documentation, as designated on the Technology Download Site, as may be revised by Original Contributor from time to time.

22. Sun Trademarks means Original Contributor's SUN, JAVA, and JINI trademarks and logos, whether now used or adopted in the future.

23. Technology means the technology described in Attachment B, and Upgrades.

24. Technology Compatibility Kit or TCK means the test programs, procedures and/or other requirements, designated by Original Contributor for use in verifying compliance of Covered Code with the Specifications, in conjunction with the Original Code and Upgraded Code. Original Contributor may, in its sole discretion and from time to time, revise a TCK to correct errors and/or omissions and in connection with Upgrades.

25. Technology Download Site means the site(s) designated by Original Contributor for access to the Original Code, Upgraded Code, TCK and Specifications.

26. Upgrade(s) means new versions of Technology designated exclusively by Original Contributor as an Upgrade and released by Original Contributor from time to time.

27. Upgraded Code means the Source Code for Upgrades, possibly including Modifications made by Contributors.

28. You(r) means an individual, or a legal entity acting by and through an individual or individuals, exercising rights either under this License or under a future version of this License issued pursuant to Section 4.1. For legal entities, You(r) includes any entity that by majority voting interest controls, is controlled by, or is under common control with You.

## ATTACHMENT A

### REQUIRED NOTICES

#### ATTACHMENT A-1

#### REQUIRED IN ALL CASES

The contents of this file, or the files included with this file, are subject to the current version of Sun Community Source License for [fill in name of applicable Technology] (theLicense); You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://sun.com/software/communitysource>. See the License for the rights, obligations and limitations governing use of the contents of the file.

The Original and Upgraded Code is [fill in name and version of applicable Technology]. The developer of the Original and Upgraded Code is Sun Microsystems, Inc. Sun Microsystems, Inc. owns the copyrights in the portions it created. All Rights Reserved.

Contributor(s): \_\_\_\_\_

Associated Test Suite(s) Location: \_\_\_\_\_

#### ATTACHMENT A-2

#### SAMPLE LICENSEE CERTIFICATION

By clicking the `Agree' button below, You certify that You are a Licensee in good standing under the Sun Community Source License, [fill in applicable Technology and Version] (License) and that Your access, use and distribution of code and information You may obtain at this site is subject to the License.

#### ATTACHMENT A-3

#### REQUIRED STUDENT NOTIFICATION

This software and related documentation has been obtained by your educational institution subject to the Sun Community Source License, [fill in applicable Technology]. You have been provided access to the software and related documentation for use only in connection with your course work and research activities as a matriculated student of your educational institution. Any other use is expressly prohibited.

THIS SOFTWARE AND RELATED DOCUMENTATION CONTAINS PROPRIETARY MATERIAL OF SUN MICROSYSTEMS, INC, WHICH ARE PROTECTED BY VARIOUS

## INTELLECTUAL PROPERTY RIGHTS.

You may not use this file except in compliance with the License. You may obtain a copy of the License on the web at [http://sun.com/software/ communitysource](http://sun.com/software/communitysource).

## ATTACHMENT B

Java (tm) Media Framework 2.1.1 Source Technology

Description of Technology:

Java (tm) Media Framework 2.1.1 Source Technology as described on the Technology Download Site.

## ATTACHMENT C

### INTERNAL DEPLOYMENT USE

This Attachment C is only effective for the Technology specified in Attachment B, upon execution of Attachment D (Commercial Use License) including the requirement to pay royalties. In the event of a conflict between the terms of this Attachment C and Attachment D, the terms of Attachment D shall govern.

1. Internal Deployment License Grant. Subject to Your compliance with Section 2 below, and Section 8.10 of the Research Use license; in addition to the Research Use license and the TCK license, Original Contributor grants to You a worldwide, non-exclusive license, to the extent of Original Contributor's Intellectual Property Rights covering the Original Code, Upgraded Code and Specifications, to do the following:

- a) reproduce and distribute internally, Original Code and Upgraded Code as part of Compliant Covered Code, and Specifications, for Internal Deployment Use,
- b) compile such Original Code and Upgraded Code, as part of Compliant Covered Code, and reproduce and distribute internally the same in Executable form for Internal Deployment Use, and
- c) reproduce and distribute internally, Reformatted Specifications for use in connection with

Internal Deployment Use.

2. Additional Requirements and Responsibilities. In addition to the requirements and responsibilities described under Section 3.1 of the Research Use license, and as a condition to exercising the rights granted under Section 3 above, You agree to the following additional requirements and responsibilities:

2.1. Compatibility. All Covered Code must be Compliant Covered Code prior to any Internal Deployment Use or Commercial Use, whether originating with You or acquired from a third party. Successful compatibility testing must be completed in accordance with the TCK License. If You make any further Modifications to any Covered Code previously determined to be Compliant Covered Code, you must ensure that it continues to be Compliant Covered Code.

## ATTACHMENT D

### COMMERCIAL USE LICENSE

1. Effect. This Attachment D is to the Sun Community Source License version 2.12 for Java Media Framework (SCSL). You have agreed to the terms of the SCSL by selecting the Accept button at the end of the SCSL or executing a hardcopy SCSL with Original Contributor. You acknowledge that the SCSL is binding on You. This Attachment D is effective only if signed below by You and Original Contributor, and applies to Your Commercial Use of Original Code and Upgraded Code. All capitalized terms used herein shall have the same meaning set forth in the SCSL, unless otherwise stated.

2. Term. Upon execution of this Attachment D by You and Original Contributor, this Commercial Use license shall have an initial term of three (3) years and shall automatically renew for additional one year terms unless either party provides notice to the other no less than sixty (60) days prior to an anniversary date.

3. Commercial Use License Grant. Subject to Your compliance with Section 4 below, Section 8.10 of the Research Use license, and the TCK license; in addition to the Research Use license, the TCK license, and the Internal Deployment Use license, Original Contributor grants to You a worldwide, non-exclusive, non-transferable license, to the extent of Original Contributors Intellectual Property Rights covering the Original Code, Upgraded Code and Specifications, to do the following within the specified field of use:

a) reproduce and distribute Compliant Covered Code;

b) compile Compliant Covered Code and reproduce and distribute the same in Executable form through multiple tiers of distribution; and

c) reproduce and distribute Reformatted Specifications in association with Compliant Covered Code.

4. Additional Requirements and Responsibilities. In addition to the requirements and responsibilities specified in the Research Use license, the TCK license (Attachment E) and the Internal Deployment license (Attachment C), and as a condition to exercising the rights granted in Section 3 above, You agree to the following additional requirements and responsibilities:

a) Field of Use. This License is only for desktop use of the Technology, and the Technology may not be used in conjunction with Java 2 Platform Micro Edition ("J2ME"), PersonalJava Application Environment ("PJava"), Embedded Java Application Environment ("EJava"), or Java TV Technology ("JavaTV") under this License. If You wish to use the Technology in conjunction with J2ME, PJava, EJava or Java TV, You should contact Original Contributor for the appropriate license agreement.

b) Distribution of Source Code. Source Code of Compliant Covered Code may be distributed only to another Licensee of the same Technology.

c) Distribution of Executable Code.

(i) Executable versions of Compliant Covered Code distributed for Commercial Use must include a description of the specific configuration(s) You used to achieve compliance with the TCK. Such description must include the system configuration, applicable operating system and version, applicable database and version, and all other required software services and installations, and which must be installed for end users to operate such Executable versions. This information must be publicly available and freely and readily accessible to potential end users prior to the licensing or purchase of such Executable versions.

(ii) You may distribute the Executable version(s) of Compliant Covered Code under a license of Your choice, which may contain terms different from this License, provided (i) that You are in



compliance with the terms of this License, and (ii) You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by Original Contributor or any other Contributor.

5. Indemnity/Limitation of Liability. The provisions of Section 7.1 of the Research Use license are superseded by the following:

a) Your Indemnity Obligation. You hereby agree to defend, at Your expense, any legal proceeding brought against Original Contributor or any Licensee to the extent it is based on a claim: (i) that the use, reproduction or distribution of any of Your Error Corrections or Shared Modifications is an infringement of a third party trade secret or a copyright in a country that is a signatory to the Berne Convention; (ii) arising in connection with any representation, warranty, support, indemnity, liability or other license terms You may offer in connection with any Covered Code; or (iii) arising from Your Commercial Use of Covered Code, other than a claim covered by Section 5.b) below, or a patent claim based solely on Covered Code not provided by You. You will pay all damages, costs and fees awarded by a court of competent jurisdiction, or such settlement amount negotiated by You, attributable to such claim.

b) Original Contributors Indemnity Obligation. Original Contributor will defend, at its expense, any legal proceeding brought against You, to the extent it is based on a claim that Your authorized Commercial Use of Original Code and Upgraded Code is an infringement of a third party trade secret or a copyright in a country that is a signatory to the Berne Convention, and will pay all damages, costs and fees awarded by a court of competent jurisdiction, or such settlement amount negotiated by Original Contributor, attributable to such claim. The foregoing shall not apply to any claims of intellectual property infringement based upon the combination of code or documentation supplied by Original Contributor with code, technology or documentation from other sources.

c) Right of Intervention. Original Contributor will have the right, but not the obligation, to defend You, at Original Contributors expense, in connection with a claim that Your Commercial Use of Original Code and Upgraded Code is an infringement of a third party patent and will, if Original Contributor chooses to defend You, pay all damages, costs and fees awarded by a court of competent jurisdiction, or such settlement amount negotiated by Original Contributor, attributable to such claim.

d) Prerequisites. Under Sections 5.b) and c) above, You must, and under Section 5.a) above, Original Contributor or any Licensee must: (i) provide notice of the claim promptly to the party

providing an indemnity; (ii) give the indemnifying party sole control of the defense and settlement of the claim; (iii) provide the indemnifying party, at indemnifying partys expense, all available information, assistance and authority to defend; and (iv) not have compromised or settled such claim or proceeding without the indemnifying partys prior written consent.

e) Additional Remedies. Should any Original Code, Upgraded Code, TCK, Specifications, or Modifications become, or in the indemnifying partys opinion be likely to become, the subject of a claim of infringement for which indemnity is provided above, the indemnifying party may, at its sole option, attempt to procure on reasonable terms the rights necessary for the indemnified party to exercise its license rights under this License with respect to the infringing items, or to modify the infringing items so that they are no longer infringing without substantially impairing their function or performance. If the indemnifying party is unable to do the foregoing after reasonable efforts, then the indemnifying party may send a notice of such inability to the indemnified party together with a refund of any license fees received by the indemnifying party from the indemnified party for the infringing items applicable to the indemnified partys future use or distribution of such infringing items, in which case the indemnifying party will not be liable for any damages resulting from infringing activity with respect to the infringing items occurring after such notice and refund.

## 6. Support Programs.

Support to You. Technical support is not provided to You by Original Contributor under this License. You may contract for one or more support programs from Original Contributor relating to the Technology which are described on the SCSL Webpage.

Customer Support. You are responsible for providing technical and maintenance support services to Your customers for Your products and services.

## 7. Royalties and Payments.

Technology specified in Attachment B.

Field of Use\*: \_\_\_\_\_

\*See Section 4.a) above.

Royalty per Unit or Per Concurrent User

(based on product family): \$0.00

8. Notice of Breach or Infringement. Each party shall notify the other immediately in writing when it becomes aware of any breach or violation of the terms of this License, or when You become aware of any potential or actual infringement by a third party of the Technology or Original Contributors Intellectual Property Rights therein.

9. Proprietary Rights Notices. You may not remove any copyright notices, trademark notices or other proprietary legends of Original Contributor or its suppliers contained on or in the Original Code, Upgraded Code and Specifications.

10. Notices. All written notices required by this License must be delivered in person or by means evidenced by a delivery receipt and will be effective upon receipt by the persons at the addresses specified below.

Original Contributor: You:

Sun Microsystems, Inc. \_\_\_\_\_

901 San Antonio Road \_\_\_\_\_

Palo Alto, California 94303 \_\_\_\_\_

Attn.: VP, Sun Software \_\_\_\_\_

and Technology Sales

cc: Sun Software and Attn.: \_\_\_\_\_

Technology, General

Counsel

11. Disclaimer of Agency. The relationship created hereby is that of licensor and licensee and the

parties hereby acknowledge and agree that nothing herein shall be deemed to constitute You as a franchisee of Original Contributor. You hereby waive the benefit of any state or federal statutes dealing with the establishment and regulation of franchises.

12. Confidentiality. You shall keep and maintain in confidence the terms and conditions of this Attachment D.

Agreed:

You: Original Contributor:

\_\_\_\_\_ Sun Microsystems, Inc.

By: \_\_\_\_\_ By: \_\_\_\_\_

Title \_\_\_\_\_ Title: \_\_\_\_\_

Date: \_\_\_\_\_ Date: \_\_\_\_\_

**Annexe 5: Liste non-exhaustive de sociétés utilisant  
GNU/Linux de façon officielle**

(<http://www.linux-france.org/article/lbiz-fr/>)

## Services Internet

Un nombre élevé d'entreprises proposent des services autour de l'Internet (Web, news, messagerie ...) grâce à Linux. Voici quelques exemples:

- Belgacom
- Zymurgy Systems Inc.
- Cybernet Pty Ltd
- Dynamic Solutions
- HogiaNet
- Comfo Access Information Network Canada Corp.
- NIKOMA MediaWorks GmbH
- Electropolis
- Softcraft Impresa
- Hex.Net Superhighway
- Web Point Communications
- Internet Discovery Ltd
- Vest Internett a/s
- CyberSites, Inc.
- Sonoma.Net
- InfiNet
- WaW
- Internet Gateway Inc.
- Datasync, Inc.
- Kralizec Pty Ltd
- Genesis Internet Services Limited
- Svenska Internet Centralen
- Helicon On Line
- West Philadelphia Network
- AGEDNA [d.o.o.](#)
- Net Nation
- Data Control
- Net Central Cybercafe Ltd.
- Deja News, Inc.
- iConnect Corporation

- Rogers Cable
- BokNet
- Sandhill Solutions
- Preferred Internet, Inc. (kits pour FAI)
- Blue Marble Live
- Telnet Canada Enterprises, Ltd.
- The Reference
- Internation
- Seafare Access Ltd.
- The Net Result System Services, Inc.
- Vigilant Internet services Ltd.
- WebJack.com
- Websense.Net
- Creanet
- Dataplus
- IN\*KA e.V.
- CymruNet Ltd
- 'NET - AT - WORK GmbH
- The Web Site Inc.

#### Informatique: conseil et développement

- GRMBL
- SISA
- Akumiitti Ltd
- Farrow Norris Pty Ltd
- Medonic
- Vertek Corporation
- MostlyLinux
- Pacific Digital Interactive
- Obsidian Systems CC
- Datapat GmbH
- Knox Software
- BeFree SA/NV
- TIS Software Limited
- The Clan Partnership Ltd.

- Logica PLC
- Syntel
- Intelligent Alternatives
- Fly-By-Day Consulting, Inc.
- Emit Sp z o.o.
- SysNet
- Software Builders
- C&B Consulting
- I/S LiquidCom
- Better Access NV
- Linux Canada Inc.
- Golden Retriever Corp.
- Wyoming Financial Information Systems
- Dynamis EDV-Consulting
- Real Software
- No Limits
- S&F Consultores
- Cymitar Technology Group Inc.
- Internet Business Development S.L.
- Larson Software Technology
- Rapid Data Inc.
- JumpStart Systems, Inc.
- Integrated Technologies Corp.
- Hugh Bragg (conseil réseau)
- Aquasoft
- iNetLab
- M-Tech Mercury Information Technology, Inc.
- Seattle Software Labs, Inc.
- ARDI -- makers of Executor Mac emulator software
- Wolfram Research
- Progressive Computer Concepts, Inc.
- Voxar Ltd
- Realtime Software Solutions, Inc.
- Byte Designs Ltd.
- Proven Software Inc.
- Bent Media Inc.



- tummy.com, ltd.
- Elektrondata AB
- Image Integration Inc
- Executive Consultants
- IBM

## Telecommunications

- LoopExpert Technologies
- Century Computers
- Digital Video Systems, Inc.
- Cisco Systems Inc.
- SpellCaster Telecommunications Inc.
- Quality Service Management, Inc.
- Telechamada, Chamada de Pessoas, SA
- RTV Regional-TV Services GmbH
- Sony WorldWide Networks

## Instrumentation et systèmes de contrôle/asservissement

- Fluke Corporation
- Erik Thiele (who built an autonomous robot)

## Ingénierie

- UNITED RAILWAY SIGNAL GROUP, INC.
- Virginia Power
- Atrax Engineering
- Sony Electronics Inc
- Kärnkraft Säkerhet och Utbildning AB
- RHR Software
- Townsend Engineering Services, Inc.
- RMD Engineering Inc.
- The Silver Hammer Group Ltd.

## Construction

- Triton ETD
- Morrison Industries

## Commerce

- Universal Computer Services
- Design Dimensions

## Préetrochimie

- ACCEL
- Decollement Consulting, Ltd.
- Canadian Association of Petroleum Producers

## Édition

- Byte Magazine (McGraw Hill)
- The Linux Journal (Specialized Systems Consultants, Inc.)
- Digisoft Software Development
- The Auto Channel, Inc.
- Linux-Magazin Verlag
- Digital Domain (film Titanic)
- Éditions O'Reilly
- Where.com

## Finance

- INTECH
- Sallie Mae Inc.
- Netherlands Foreign Investment Agency

## Automobile

- Debis Systemhaus CCS GmbH
- Mercedes-Benz AG

## Santé

- The Alberta Provincial Mental Health Advisory Board

## Armée

- Science Applications International Corporation
- U.S. Army Publications and Printing Command
- U.S. Navy: Personnel Support Activity, San Diego

## Agriculture

- Agdia, Inc.
- Animal and Plant Health Inspection Service

## Sports et loisirs

- The Calgary Winter Club
- ROSCO/Entertainment Technology

## Immobilier et BTP

- Wellsford Residential Property Trust
- Keyline Building Materials
- Re/Max Team Ideal Realty Inc.
- Com-Steel, [L.L.C.](#)

## Tourisme

- Allied Tours
- Intransco
- Travel Concepts of Atlanta

## Transports

- Yellow Cab Service Corporation

## Linux (matériel, logiciel, livres ...)

- Apache Digital Corporation
- Linux Systems Labs
- Revolutionary Software, Inc.
- Craftwork Solutions, Inc.
- Sangoma Technologies Inc.
- Numerical Algorithms Group, Inc.
- All-Linux Shopping Mall
- WorkGroup Solutions, Inc
- Real Magic LINUX
- VirtuFlex Software Corp.
- Quality Software Solutions, Inc.
- DCG Computers, Inc.
- S.u.S.E. GmbH
- LiSA Consulting
- Red Hat Software, Inc.
- Cyrix utilise apparemment Linux afin de tester ses processeurs.
- AMD utilise apparemment Linux afin de tester ses processeurs.
- Centaur tient compte de Linux pour améliorer ses processeurs
- Technology Associates, Inc.
- Dataforge
- VA Research
- Linux Central

## Enseignement et recherche

- Calif. Polytechnic State Univ.
- Chair for Technical Thermodynamics
- New Zealand Forest Research Institute
- GLACIER
- Institut für Informations- und Datenverarbeitung
- Millbury Public Schools
- University of Rio Grande, CS Dept.
- Janus Pannonius University, Faculté de Droit
- Complex of Schools of Environmental Engineering
- Texas Tech University Department of Physics
- Mankato State University
- Studio of Arts And Sciences

## Agence d'état

- National Disaster Communication Response Team
- United States Postal Service

## Associations à but non lucratif et caritatives

- Community Information & Referral
- California Veterinary Diagnostic Laboratory System
- Beauregard Parish Public Library

**Annexe 6 : Liste non-exhaustive de sociétés utilisant des logiciels libres de façon officielle**

(<http://www.iful.org/solutions/entreprises/users.html>)

## **1. Liste d'entreprises ou d'organisations qui utilisent Linux en France**

La Poste, Lectra Systèmes, Pick Systems, Schlumberger, la BNP, l'Observatoire Européen de l'Audiovisuel (conseil de l'Europe), l'Oréal, l'armée, Canon, Médiavision, Radio-France, Publicis, la Sagem, les Éditions Lavoisier, la DGA, l'Institut Européen, Cetim, Yahoo!, Alcatel, Siemens, Bouygues Telecom, Dim, l'ART (autorité de régulation des télécom), la Cité des Sciences de la Villette, la SOGERES, Ubi-Soft, les éditions Gauthier-Villars, Tefal, France Telecom, la DGAC, le Ministère de l'Équipement, la CPAM de Strasbourg...

## **2. Liste d'entreprises ou d'organisations qui utilisent Linux dans le monde**

US Postal, Yellow Cab, la NASA, Ikea, Fujitec, Sony, Corel Computers, l'armée et la marine américaines, Digital Domain (effets spéciaux du Titanic), Mercedes-Benz, Cisco, Netscape, IBM, Microsoft, Apple, Digital Equipment, Dreamworks, ILM, le magazine Byte...

Et aussi plusieurs centaines de fournisseurs d'accès Internet et d'organisations éducatives.

## **3. Liste d'entreprises ou d'organisations qui utilisent des logiciels libres dans le monde**

Adobe, Agfa, AMD, Amdahl, AT&T, Baan, Bellcore, Boeing, Bosch, British Telecom, Caltech, Canon, le CERN, Compaq, Cygnus, Daimler-Benz, Dasa, DEC, DLR, Ericsson, FAA, Ford, GMD, Honeywell, Hewlett-Packard, IBM, Intel, J.P. Morgan, Kodak, Lockheed, Lucent, MCI, Motorola, NASA, Netscape, Nokia, Oracle, Philips Raytheon, SAIC, SAP, Silicon Graphics, Shell, Siemens, Star Division, Sun, Thomson, Texas Instruments, US Air Force, US Navy, VocalTec, Xerox, the Bank of America, le FBI, la famille royale britannique, le Financial Times...

**Annexe 7 : La correspondance et le projet de loi du cas  
de l'administration péruvienne**



## 7.1 Le projet de loi

Los Congresistas de la República que suscriben, **EDGAR VILLANUEVA NÚÑEZ** y **JACQUES RODRICH ACKERMAN**, ejerciendo el derecho de iniciativa legislativa que les confiere el artículo 107° de la Constitución Política del Perú, presentan la siguiente iniciativa legislativa:

### **LEY DE USO DE SOFTWARE LIBRE EN LA ADMINISTRACIÓN PÚBLICA**

#### EXPOSICION DE MOTIVOS

La complejidad del mundo en que vivimos exige permanentemente de los países una constante revisión y adaptación de sus marcos institucionales logrando de esta forma estar al compás de los nuevos retos que nos impone el sorprendente desarrollo tecnológico.

El descubrimiento de nuevas tecnologías informáticas, entre ellas la del Software libre, se ha configurado con el devenir del tiempo en un instrumento idóneo para asegurar de una manera más idónea la protección de la información con la que el Estado cuenta.

De esta forma la tecnología cumple su función facilitadora de las diferentes y múltiples actividades humanas, siendo una de ellas, el manejo de información reservada en las canteras del Estado.

Según la Constitución Política del Perú en el inciso 5 del artículo 2°, *“toda persona tiene derecho a solicitar sin expresión de causa la información que requiera y a recibirla de cualquier entidad pública, en el plazo legal, con el costo que suponga el pedido. Se exceptúan las informaciones que afecten la intimidad personal y las que expresamente se excluyan por ley o por razones de seguridad nacional”*.

De la misma forma el inciso 6 del mismo artículo, enfatiza el derecho de toda persona a ***“que los servicios informáticos, computarizados o no, públicos o privados, no suministren informaciones que afecten la intimidad personal y familiar”***.

En este orden de ideas, es evidente la preocupación de nuestra carta fundamental de establecer reaseguros institucionales que por un lado protejan la libertad de los ciudadanos para acceder a información pública y por el otro, la de guardar la debida reserva de información que afecten la intimidad personal y familiar, así como razones de seguridad nacional.

La garantía de estos derechos consagrados en nuestra Constitución, no pasa únicamente por la buena voluntad de los agentes del Estado para el cumplimiento normativo de la Constitución, sino también por el empleo de tecnologías que en unos casos coadyuvan y en otros no a una efectiva protección de dichos derechos ciudadanos.

Es en este contexto imperioso para el Estado incorporar aquellas tecnologías que ayudan a reforzar el ejercicio del derecho de la información de los ciudadanos y su debida reserva en los casos que lo ameriten.

La utilización del Software Libre en todas las instituciones del Estado, apunta en este sentido. Básicamente podemos decir que los principios elementales que animan al presente Proyecto de Ley se vinculan a garantías básicas del Estado democrático de derecho y lo podemos resumir en los siguientes:

1. Libre acceso del ciudadano a la información pública
2. Perennidad de los datos públicos
3. Seguridad del estado y de los ciudadanos

Para garantizar el *libre acceso de los ciudadanos a la información pública*, resulta indispensable que la codificación de los datos no esté ligada a un único proveedor. El uso de formatos estándar y abiertos permite garantizar este libre acceso, logrando si fuera necesario la creación de software compatible.

Para garantizar la *perennidad de los datos públicos*, es indispensable que la utilización y el mantenimiento del software no dependan de la buena voluntad de los proveedores, ni de las condiciones monopólicas, impuestas por éstos. Se precisan sistemas cuya evolución pueda ser garantizada gracias a la disponibilidad del código fuente.

Para *garantizar la seguridad nacional*, resulta indispensable contar con sistemas desprovistos de elementos que permitan el control a distancia o la transmisión no deseada de información a terceros. Por lo tanto, se requieren sistemas cuyo código fuente sea libremente accesible al público para permitir su examen por el propio Estado, los ciudadanos y un gran número de expertos independientes en el mundo.

Esta propuesta aporta mayor seguridad, pues el conocimiento del código fuente eliminará el creciente número de programas con código espía.

De igual forma, la iniciativa de ley potencia la seguridad de los ciudadanos, tanto en su condición de titulares legítimos de la información manejada por el Estado, cuanto en su condición de consumidores. En este último caso, permite el surgimiento de una oferta extensa de software libre desprovisto de potencial código espía susceptible de poner en riesgo la vida privada y las libertades individuales.

El Estado en aras de mejorar la calidad de la gestión pública en tanto custodio y administrador de información privada, establecerá las condiciones en que los organismos estatales adquirirán software en el futuro, es decir de un modo compatible con las garantías constitucionales y principios básicos antes desarrollados.

El proyecto expresa claramente que para ser aceptable para el Estado un programa o software cualquiera, no basta con que el programa sea técnicamente suficiente para llevar a cabo una tarea, sino que además las condiciones de contratación deben satisfacer una serie de requisitos en materia de licencia, sin los cuales el Estado no puede garantizar al ciudadano el procesamiento adecuado de sus datos, velando por su integridad, confidencialidad y accesibilidad a lo largo del tiempo, aspectos críticos para su desempeño.

El Estado establece condiciones para el empleo del software por parte de las instituciones estatales, sin inmiscuirse de modo alguno en las transacciones del sector privado. Constituye un principio jurídicamente reconocido que el Estado no tiene el amplio espectro de libertad contractual del sector privado, pues precisamente está limitado en su accionar por el deber de transparencia de los actos públicos, y en este sentido la tutela del mejor interés común debe tener preeminencia cuando se legisla sobre la materia.

El proyecto así mismo garantiza el principio de igualdad ante la ley, pues ninguna persona natural y jurídica está excluida del derecho de proveer estos bienes, en las condiciones fijadas en la presente iniciativa y sin más limitaciones que las establecidas en la Ley de Contrataciones y Adquisiciones del Estado (T.U.O. Decreto Supremo N° 012-2001-PCM)

Adicionalmente a estas ventajas podemos resaltar una serie de beneficios que como consecuencia de esta medida se empezaría a manifestar inmediatamente después de ser ejecutadas.

En primer lugar están las oportunidades de trabajo para programadores locales. Del universo de software para servidores que se comercializó en los EE.UU. el año pasado, el 27% correspondió a programas “libres”, proporción verdaderamente significativa para ese enorme y exigente mercado. La cifra es elocuente y constituye una respuesta contundente a quienes creen que el software libre implicará una fuerte limitación a la ocupación de los programadores del país. Al contrario, la iniciativa permitirá liberar una gran cantidad de recursos, y un incentivo para potenciar la creatividad humana.

Al emplear el software libre los profesionales pueden analizar a fondo los problemas y mejorar los desarrollos en todos los casos que sea necesario, nutriéndose del software libre disponible globalmente bajo distintas licencias. Constituye un campo ideal para aplicar creatividad, aspecto en el que los jóvenes peruanos alcanzarían buenos desempeños.

Por otro lado, mediante el software libre se elimina el uso de software ilegal que campea en algunas instituciones del Estado. El uso no permitido de software dentro del Estado o la mera sospecha de ello constituye un poderoso incentivo para cualquier funcionario público para modificar esa situación que atentatoria contra los derechos de propiedad intelectual.

Si bien es correcto decir que no es necesaria la adopción de software libre para cumplir con la ley, pero su empleo generalizado reducirá drásticamente las situaciones irregulares y obrará como “vector de contagio legal”, tanto dentro del estado como en el ámbito privado.

Son muchos los países que están reconociendo formalmente el uso exclusivo del Software libre en el sector estatal.

Entre ellos tenemos a Francia donde está en discusión una norma legal sobre el tema. El gobierno de la ciudad de México (DF) ya ha iniciado una importante migración para la adopción de software libre en forma generalizada, siendo este país líder en occidente. También Brasil, el estado de Recife ha decidido su adopción. La República Popular China ha adoptado desde hace varios años el software libre como una política de estado. Al igual que en los países escandinavos. En los EE.UU., la NASA y la US NAVY, entre muchas otras organizaciones, han adoptado software libre para alguna de sus necesidades, entre otras iniciativas gubernamentales y del sector privado.

Finalmente, el Proyecto de Ley en cuestión otorga a la Presidencia del Consejo de Ministros la ejecución de la presente ley por ser este organismo que concentra la dirección de todas las instituciones gubernamentales. En este sentido tiene una ventaja estratégica para realizar la reforma pertinente y el proceso migratorio de software propietario a software libre.

Es en este orden de ideas que se ha precisado estos aspectos en la presente proposición legislativa.

### ANALISIS COSTO BENEFICIO

La presente iniciativa no genera gasto alguno al erario nacional. Eso sí, para el cumplimiento de sus fines será necesario producir una reasignación del gasto gubernamental cuya incidencia se circunscribe a lo efectivamente gastado por cada organismo gubernamental en los procesos de contrataciones y licitaciones del Estado para la adquisición de programas informáticos.

Si bien es cierto, el software libre con relación al software propietario representa un ahorro sustancial a la economía del Estado, no es el punto principal de apoyo. Como señalamos antes, su

ventaja comparativa se focaliza en los reaseguros tecnológicos que el programa otorga a la información con la que cuenta el Estado, información que en muchos casos es de carácter reservada.

En este sentido una mejor protección de los derechos ciudadanos constituye un beneficio inconmensurable que debe ser reconocido desde el punto de vista del análisis costo beneficio.

Podemos resumir los beneficios del proyecto en los siguientes tópicos:

- Seguridad Nacional
  - El Estado para cumplir sus funciones debe almacenar y procesar información relativa a los ciudadanos. La relación entre el individuo y el Estado depende de la privacidad e integridad de estos datos, que deben ser adecuadamente resguardados contra tres riesgos específicos:
  - Riesgo de filtración, los datos confidenciales deben ser tratados de tal manera que el acceso a ellos sea posible exclusivamente para las personas e instituciones autorizadas.
  - Riesgo de imposibilidad de acceso, los datos deben ser almacenados de tal forma que el acceso a ellos por parte de las personas e instituciones autorizadas esté garantizado durante toda la vida útil de la información.
  - Riesgo de manipulación, la modificación de los datos debe estar restringida, nuevamente a las personas e instituciones autorizadas.
    - Con el software libre estos riesgos se atenúan considerablemente.
    - Permite al usuario la inspección completa y exhaustiva del mecanismo mediante el cual procesa los datos. El hecho de que el programa de software libre permite la inspección del programa es una excelente medida de seguridad, ya que al estar expuestos los mecanismos, estos están constantemente a la vista de profesionales capacitados, con lo que se vuelve inmensamente más difícil ocultar funciones maliciosas, aun si el usuario final no se toma el trabajo de buscarlas él mismo.
- Independencia tecnológica
  - Con el software propietario no hay libertad de contratación en lo que se refiere a

ampliaciones y correcciones del sistema que utiliza, se produce una dependencia tecnológica en la que el proveedor está en condiciones de dictar unilateralmente términos, plazos y precios.

- Con el Software libre se permite al usuario el control, corrección y modificación del programa para adecuarlo a sus necesidades. Esta libertad no está destinada solamente a los programadores Si bien son éstos los que pueden capitalizarla en primera mano, los usuarios también se benefician enormemente, porque de esta manera pueden contratar a cualquier programador (no necesariamente al autor original) para que corrija errores o añada funcionalidad.
  
- El desarrollo local
  - En el caso del software propietario el usuario esta habilitado par ejecutar un programa, pero no para inspeccionarlo ni modificarlo, entonces no pude aprender de él, se vuelve dependiente de una tecnología que no sólo no comprende sino que le está expresamente vedada. Los profesionales de su entorno, que podrían ayudarlo a alcanzar sus metas, están igualmente limitados: como e funcionamiento del programa es secreto, y su inspección está prohibida, no es posible arreglarlo. De esa manera los profesionales locales ven sus posibilidades de ofrecer valor agregado cada vez más limitadas, y sus horizontes laborales se estrechan junto con sus chances de aprender más. Con el software libre se neutraliza enormemente estás desventajas del software propietario.
  
- El costo del software
  - Se reduce considerablemente al ser libre pues no hay necesidad de estar solicitando sistemáticamente las licencias del caso para continuar con la utilización del programa. Esto sucede con el software propietario. Es importante para el usuario poder mantener estos costos bajo control, pues de lo contrario puede llegar a verse impedido de llevar a cabo sus metas, a fuerza de erogaciones no planificadas. He aquí una vez más la dependencia tecnológica que ayuda a enfrentar el software libre.
  
- Mayores fuentes de trabajo
  - Con el software libre se libera mano de obra existente en el país que estaba enfrascada

como consecuencia de la dependencia tecnológica del software propietario. Ahora se asignarían recursos de los usuarios (en esta caso las instituciones del Estado) para mantenimiento y soporte informático del software libre.

- Fomento de la creatividad e iniciativa empresarial.

#### Costos

El gran costo que supone el cambio de software propietario a software libre se circunscribe al proceso migratorio. Si bien es cierto que el proceso migratorio involucra costos en relevamientos, toma de decisiones para implementar los nuevos sistemas, mano de obra para implementar el cambio, conversión de datos, reentrenamiento del personal, y eventualmente gastos en licencias y/o desarrollo y tiempo; no es menos cierto que todos estos costos son fijos y se pagan por única vez en cambio, el software propietario en funcionamiento ahora, también tiene sus costos fijos que fueron pagados y no pueden ser recuperados. Pero además de estos costos hay otros costos involucrados en el software propietario: actualizaciones permanentes (aveces acentuada por un monopolio auto sostenido) y sobre todo el inmenso precio que tiene para el estado la pérdida de las libertades que le garantizan el control de su propia información. Estos costos son permanentes y crecientes a lo largo del tiempo y tarde o temprano superan a los costos fijos de realizar una migración.

En fin, son mayores los beneficios a los costos que el proceso de migración supone.

### **INCIDENCIA DE LA NORMA SOBRE LA LEGISLACION NACIONAL**

El presente proyecto no afecta ninguna ley toda vez que sobre la materia no se ha legislado aún sobre la materia.

Por lo anteriormente expuesto y

#### **CONSIDERANDO:**

Que, los incisos 5 y 6 del artículo 2° de la Constitución Política protegen tanto el derecho información de los ciudadanos, así como las garantías suficientes a los efectos de tutelar eficientemente la reserva de la información en los casos que la ley los exige.



Que, al ser el software libre el medio tecnológico más idóneo para resguardar los derechos antes mencionados es imperativo que el Estado utilice en todas sus instituciones dicho sistema.

En consecuencia, habiendo cumplido con lo establecido en el artículo 75° del Reglamento del Congreso de la República, se propone el siguiente proyecto de Ley:

## **FORMULA LEGAL**

### **EL CONGRESO DE LA REPUBLICA:**

Ha dado la Ley siguiente:

### **LEY DE USO DE SOFTWARE LIBRE EN LA ADMINISTRACIÓN PÚBLICA**

#### **Artículo 1°.- Objeto de la ley**

Empléase en todas las instituciones del Estado el uso exclusivo de programas o software libres en sus sistemas y equipamientos de informática.

#### **Artículo 2°.- Ámbito de aplicación**

Los Poderes Ejecutivo, Legislativo y Judicial, los Organismos autónomos y descentralizados sean regionales o locales y las empresas donde el Estado posea mayoría accionaria, harán uso de programas o software libres en sus sistemas y equipamientos de informática.

### **Artículo 3°.- Autoridad de aplicación**

La autoridad encargada de poner en ejecución la presente ley será la Presidencia del Consejo de Ministros.

### **Artículo 4°.- Definición de Software libre**

Para los efectos de la presente ley se entenderá por programa o software libre, aquel cuya licencia de uso garantice al usuario, sin costo adicional, las siguientes facultades:

1. Uso irrestricto del programa para cualquier propósito.
2. Acceso irrestricto al código fuente o de origen respectivo.
3. Inspección exhaustiva de los mecanismos de funcionamiento del programa.
4. Uso de los mecanismos internos y de porciones arbitrarias del programa para adaptarlos a las necesidades del usuario.
5. Confección y distribución libre de copias del programa.
6. Modificación del programa y distribución libre tanto de las alteraciones como del nuevo programa resultante, bajo las mismas condiciones del programa original.

### **Artículo 5°.- Excepciones**

En caso de no existir una solución que utilice software libre y permita satisfacer una necesidad determinada, las instituciones del Estado podrán adoptar las siguientes alternativas, atendiendo al orden de prelación siguiente:

Si mediaran exigencias de tiempo verificables para atender un problema técnico y se halle disponible en el mercado software propietarios, el organismo que lo demande podrá gestionar ante la Autoridad de Aplicación un permiso de excepción de utilización de software propietario que reúna las siguientes características:

1. Se seleccionará en primer término a los programas que cumplan con todos los criterios enunciados en el artículo 4° de la presente ley, excepto por la facultad de distribución del programa modificado. En este caso, el permiso de excepción podrá ser definitivo.
2. Si no se pudiera disponer de programas de la categoría precedente, se deberán escoger aquellos para los que exista un proyecto de desarrollo avanzado de tipo libre. En este caso, el permiso de excepción será transitorio y caducará automáticamente cuando el software libre pase a

estar disponible con la funcionalidad que sea necesaria.

3. Si no se encontraren productos de estas condiciones, se podrá optar por programas propietarios, pero el permiso de excepción emanado de la autoridad de aplicación caducará automáticamente a los dos años de emitido, debiendo ser renovado previa constatación de que no exista disponible en el mercado una solución de software libre satisfactoria.

La autoridad de aplicación otorgará un permiso de excepción únicamente si el organismo estatal solicitante garantice el almacenamiento de los datos en formatos abiertos, sin perjuicio del pago de las licencias propietarias respectivas.

#### **Artículo 6°.- Permisos educativos**

Toda entidad educativa dependiente del Estado, está habilitada para gestionar un permiso de software propietario para su uso en investigación, previo pago de los derechos de autor correspondientes y las licencias del caso, siempre que el objeto de investigación esté directamente asociado al uso del programa en cuestión.

#### **Artículo 7°.- Transparencia de las excepciones**

Las excepciones emanadas de la autoridad de aplicación deberán ser sustentadas y publicadas en la página web del Portal del Estado.

La resolución que autoriza la excepción deberá enumerar los requisitos funcionales concretos que el programa debe satisfacer.

#### **Artículo 8°.- Autorización excepcional**

En caso que alguna entidad del Estado comprendido en el artículo 2° de la presente ley, es autorizado excepcionalmente para adquirir software propietario para almacenar o procesar datos cuya reserva sea necesario preservar, la autoridad de aplicación deberá publicar en el Portal del estado un informe donde se expliquen los riesgos asociados con el uso de software de dichas características para esa aplicación en particular.

Los permisos de excepción otorgados a los organismos del Estado relacionados con la seguridad y la defensa nacional están exceptuados de la obligación anteriormente expuesta.

## **Artículo 9°.- Responsabilidades**

La máxima autoridad administrativa y autoridad técnica e informática de cada institución del Estado asumen la responsabilidad por el cumplimiento de esta ley.

## **Artículo 10°.- Norma Reglamentaria**

El poder ejecutivo reglamentará en un plazo de ciento ochenta días, las condiciones, tiempos y formas en que se efectuará la transición de la situación actual a una que satisfaga las condiciones de la presente Ley y orientará, en tal sentido, las licitaciones y contrataciones futuras de software realizadas a cualquier título.

Así mismo, se encargará de dirigir el proceso migratorio del sistema de software propietario a libre, en todos los casos que las circunstancias lo exija.

## **Artículo 11°.- Glosario de Términos**

1. **Programa o “Software”**, a cualquier secuencia de instrucciones usada por un dispositivo de procesamiento digital de datos para llevar a cabo una tarea específica o resolver un problema determinado.
2. **Ejecución o empleo de un programa**, al acto de utilizarlo sobre cualquier dispositivo de procesamiento digital de datos para realizar una función.
3. **Usuario**, a aquella persona física o jurídica que emplea el software.
- d. **Código fuente o de origen, o programa fuente o de origen**, al conjunto completo de instrucciones y archivos digitales originales creados o modificados por quien los programara, más todos los archivos digitales de soporte, como tablas de datos, imágenes, especificaciones, documentación, y todo otro elemento que sea necesario para producir el programa ejecutable a partir de ellos. Como excepción, podrán excluirse de este conjunto aquellas herramientas y programas que sean habitualmente distribuidos como software libre por otros medios como, entre otros, compiladores, sistemas operativos y librerías.
5. **Programa o software libre**, a aquel cuyo empleo garantice al usuario, sin costo adicional, las siguientes facultades:
  - e.1. Ejecución irrestricta del programa para cualquier propósito.
  - e.2. Acceso irrestricto al código fuente o de origen respectivo.

- e.3. inspección exhaustiva de los mecanismos de funcionamiento del programa.
  - e.4. Uso de los mecanismos internos y de cualquier porción arbitraria del programa para adaptarlo a las necesidades del usuario.
  - e.5. Confección y distribución pública de copias del programa.
  - e.6. Modificación del programa y distribución libre, tanto de las alteraciones como del nuevo programa resultante, bajo las mismas condiciones del programa original.
5. **Software propietario (programa no libre)**, a aquel que no reúna todos los requisitos señalados en el término precedente.
6. **Formato abierto**, a cualquier modo de codificación de información digital que satisfaga tanto los estándares existentes así como las siguientes condiciones tales que:
- g.1. Su documentación técnica completa esté disponible públicamente.
  - g.2. El código fuente de al menos una implementación de referencia completa esté disponible públicamente.
  - g.3. No existan restricciones para la confección de programas que almacenen, transmitan, reciban o accedan a datos codificados de esta manera.

Comuníquese al Señor Presidente de la República para su promulgación.

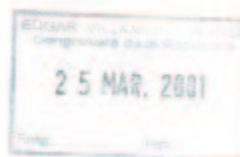
En Lima a los

Lima, 09 de abril de 2002

## 7.2 La correspondance de Microsoft

Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052-6399  
USA

Microsoft Perú S.R.L.  
Av. Víctor Andrés Belaunde 147  
Vía Principal N° 123, Torre Real 1, Piso 3  
Centro Empresarial  
Lima 27, Perú. Telf.: 211-5600 Fax: 421-7807  
<http://www.microsoft.com>



**Microsoft**

San Isidro, 21 de Marzo de 2002

Señor  
Edgar Villanueva Nuñez  
Congresista de la República  
Presente.-

Estimados señores:

Primeramente, queremos agradecerle la oportunidad que nos brindó de informarle cómo venimos trabajando en el País en beneficio del sector público, siempre buscando las mejores alternativas para lograr la implementación de programas que permitan consolidar las iniciativas de modernización y transparencia del Estado. Precisamente, fruto de nuestra reunión hoy Usted conoce de nuestros avances a nivel internacional en el diseño de nuevos servicios para el ciudadano, dentro del marco de un Estado modelo que respeta y protege los derechos de autor.

Este accionar, tal como conversamos, es parte de una iniciativa mundial y hoy en día existen diversas experiencias que han permitido colaborar con programas de apoyo al Estado y a la comunidad en la adopción de la tecnología como un elemento estratégico para impactar en la calidad de vida de los ciudadanos.

De otro lado, como quedamos en esta reunión, nosotros asistimos al Foro realizado en el Congreso de la República el 6 de marzo, a propósito del proyecto de ley que Usted lidera, en donde pudimos escuchar las diferentes presentaciones que hoy nos llevan a exponer nuestra posición a fin de que Usted tenga un panorama más amplio de la real situación.

1. El proyecto establece la obligatoriedad de que todo organismo público debe emplear exclusivamente software libre, es decir de código abierto, lo cual *traspasa los principios de la igualdad ante la ley, el de no discriminación y los derechos a la libre iniciativa privada, libertad de industria y contratación protegidos en la Constitución.*
2. El proyecto, al hacer obligatorio el uso de software de código abierto, *establecería un tratamiento discriminatorio y no competitivo en la contratación y adquisición de los organismos públicos, contraviniendo los principios de base de la Ley 26850 de Contrataciones y Adquisiciones del Estado.*

Microsoft Corporation  
One Microsoft Way  
Redmond, Wa 98052-6399  
USA

Microsoft Perú S.R.L.  
Av. Victor Andrés Bolognini 147  
Vta Principal N° 123, Torre Real 1, Piso 3  
Centro Empresarial  
Lima 27, Perú - Telf: 211-5600 Fax: 421-7307  
<http://www.microsoft.com>

**Microsoft**

3. Así, al obligar al Estado a favorecer un modelo de negocios que apoyaría exclusivamente el software de código abierto, el proyecto sólo estaría desalentando a las compañías fabricantes locales e internacionales que son las que verdaderamente realizan importantes inversiones, crean un significativo número de puestos de empleos directos e indirectos, además de contribuir al PBI vs. un modelo de software de código abierto que tiende a tener un impacto económico cada vez menor debido a que crea principalmente empleos en servicio.
4. El proyecto de ley impone el uso de software de código abierto sin considerar los peligros que esto pueda conllevar desde el punto de vista de seguridad, garantía y posible violación de los derechos de propiedad intelectual de terceros.
5. El proyecto maneja de manera errónea los conceptos de software de código abierto, que no necesariamente implica que sea software libre o de costo cero, llegando a realizar *conclusiones equivocadas sobre ahorros para el Estado, sin ningún sustento costo beneficio* que valide la posición.
6. *Es equivocado pensar que el Software de Código Abierto es gratuito.* Investigaciones realizadas por Gartner Group (importante investigadora del mercado tecnológico reconocida a nivel mundial) han señalado que el costo de adquisición del software (sistema operativo y aplicaciones) se reduce a sólo 8% del total de costos que las empresas e instituciones deben asumir como consecuencia del uso racional y realmente provechoso de la tecnología. El otro 92% lo constituyen: costos de implantación, capacitación, soporte, mantenimiento, administración e inoperatividad.
7. Uno de los argumentos que sustentan el proyecto de ley es la supuesta gratuidad del software de código abierto, comparado con los costos del software comercial, sin tener en cuenta que *existen modalidades de licenciamiento por volumen* que pueden ser sumamente ventajosas para el Estado, tal como se ha logrado en otros países.
8. Adicionalmente, la alternativa adoptada por el proyecto (i) es claramente más costosa por los *altos costos que supone una migración* y (ii) pone en *riesgo la compatibilidad y posibilidad de interoperabilidad* de las plataformas informáticas dentro del Estado, y entre el Estado y el sector privado, dada la centena de versiones que existen de software de código abierto en el mercado.
9. El software de código abierto en su mayoría no ofrece los niveles de servicio adecuados ni la garantía de fabricantes reconocidos para lograr mayor productividad por parte de los usuarios, lo cual ha motivado que diferentes entidades públicas hayan retrocedido en su decisión de ir por una solución de software de código abierto y se encuentren utilizando software comercial en su lugar.

Microsoft Corporation  
One Microsoft Way  
Redmond, Wa 98052-6399  
USA

Microsoft Perú S.R.L.  
Av. Víctor Andrés Belaúnde 147  
Vía Principal N° 123, Torre Real 1, Piso 3  
Centro Empresarial  
Lima 27, Perú - Telf: 211-5600 Fax: 421-7307  
<http://www.microsoft.com>

**Microsoft**

10. *El proyecto desincentiva la creatividad de la industria peruana de software, que factura US\$ 40 millones/año, exporta US\$ 4 millones (10mo. en ranking de productos de exportación no tradicional, más que artesanías) y es una fuente de empleo altamente calificado. Con una Ley que incentive el uso de software de código abierto, los programadores de software pierden sus derechos de propiedad intelectual y su principal fuente de retribución.*
11. *El software de código abierto, al poder ser distribuido gratuitamente, tampoco permite generar ingresos para sus desarrolladores por medio de la exportación. De esta forma, se debilita el efecto multiplicador de la venta de software a otros países y por lo tanto el crecimiento de esta industria, cuando contrariamente las normas de un Gobierno deben estimular la industria local.*
12. *En el Foro se discutió sobre la importancia del uso de software de código abierto en la educación, sin comentar el rotundo fracaso de esta iniciativa en un país como México, en donde precisamente los funcionarios del Estado que fundamentaron el proyecto, hoy expresan que el software de código abierto no permitió brindar una experiencia de aprendizaje a alumnos en la escuela, no se contó con los niveles de capacitación a nivel nacional para dar soporte adecuado a la plataforma, y el software no contó y no cuenta con los niveles de integración para la plataforma que existe en las escuelas.*
13. *Si el software de código abierto satisface todos los requerimientos de las entidades del Estado ¿porqué se requiere de una Ley para adoptarlo? ¿No debería ser el mercado el que decida libremente cuáles son los productos que le dan más beneficios o valor?*

Agradezco de sobremanera la atención prestada a la presente, queremos reiterarle nuestro interés de reunirnos con usted para poder exponer con más detalle nuestros puntos de vista al proyecto presentado por usted, y ponernos a su plena disposición para compartir experiencias e información que estamos seguros podrán aportar para un mejor análisis e implementación de una iniciativa que tiene por objetivo la modernización y transparencia del Estado, en beneficio del ciudadano.

Atentamente,

Juan Alberto González  
Gerente General  
Microsoft Perú



## **Annexe 8: GNU Free Documentation Licence**

# GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 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 written 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 that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

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. (For example, 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.

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 "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. 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 designed for human modification. Opaque formats include PostScript, PDF, 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 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.

## **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 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. 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 less than five).
- 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", and 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. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

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.

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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the

other self-contained works thus compiled with the Document, on account of their being thus compiled, if they 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 quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around 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 provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## **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.

# Bibliographie

## Ouvrages et documentation

- BEHLENDORF Brian, BRADNER Scott, HALERLY Jim, McKUSICK Kirk, O'REILLY Tim, PAQUIN Tom, PERENS Bruce, RAYMOND Eric, STALLMAN Richard, TIEMAN Michael, TORVALDS Linus, VIXIE Paul, WALL Larry, YOUNG Bob, *Tribune Libre (Ténors de l'Informatique libre)*, Paris, O'Reilly, 1999, 295 pages
- DI COSMO Roberto, *Piège dans le Cyberspace*, 2000, Alexandrie Online, 45 pages
- PICARDE Gabriel, *Linux pour le décideur*, Paris, OEM, 1999, 328 pages
- SMETS-SOLARES Jean-Paul et FAUCON Benoît, *Logiciels Libres (Liberté-égalité-Business)*, 1999, Paris, Edispher, 250 pages

## Périodique et rapport

- BRISSET Antoine, *Mémoire en marketing sur le marché du système d'exploitation GNU/Linux*, [http://www.linux-france.org/article/these/memoire-brisset/brisset\\_monoblock.html](http://www.linux-france.org/article/these/memoire-brisset/brisset_monoblock.html)
- CARCENAC Thierry, *Pour une Administration Electronique Citoyenne (Méthode et moyens)*, 2001, <http://www.internet.gouv.fr/francais/textesref/rapcarcenac/sommaire.htm>, 93 pages
- CERT Coordination Center, *Trends in denial of Service Attack Technology*, 2001, Carnegie Mellon University, 20 pages
- HERITIER Jean-Claude, *Les logiciels libres, un modèle économique viable?*, 2001, IECS de Strasbourg, 71 pages
- Working group on Libre Software of the European Council, *Free Software/Open Source: Information Society Opportunities for Europe?*, 2000, <http://eu.conecta.it/paper/paper.html>, 42 pages

### Articles de revues

- CALDERS Alfons, « L'E-business préfère-t-il Linux à Windows XP? », in *Techniques et Management*, février 2002, pages 12-17
- KINDERMANS Guy, « Liberté, égalité, rentabilité », in *Data News*, 03/05/2002, page 6
- LANG Bernard, « Des logiciels libres à la disposition de tous », in *Le Monde Diplomatique*, de janvier 1998, page 26
- s.a., « HP s'engage dans l'Open Source », in *Linux Loader*, n°8 Juiller/Août 2001, pages 56-57
- s.a., « IBM et Linux: comme une envie de marché », in *Linux Loader*, n°6 Mars Avril 2001, pages 44-46

### Sites Internet

- Alcove, <http://www.alcove.com/fr/index>
- Arkane Media, <http://www.arkane-media.net>
- ARTNER Bob, *Research shows Linux support*,  
<http://www.zdnet.com.au/newstech/os/story/0,2000024997,20261699,00.htm>
- Asynux, <http://www.asynux.fr/>
- Atrid, <http://www.atrid.fr/accueil/accueil.html>
- Attrition.org, *Defacements by Operating System OS Graphs, August, 1999 - April, 2001*,  
<http://attrition.org/mirror/attrition/os-graphs.html>
- BERGER Olivier, COUCHET Frédéric, DRIEU Benjamin, RIDEAU François-René, *Linux est-il vraiment libre et gratuit?*, <http://www.april.org/articles/divers/libre-gratuit.html>
- Berkman center for Internet ans Society, *The power of Openness*, 2001,  
<http://www.eon.law.harvard.edu/opencode/h2o>
- BERSTEIN D.J., *Internet host SMTP server survey*, <http://cr.yip.to/surveys/smtpsoftware6.txt>
- BRADFORD Ed, *Pipes in Linux, Windows 2000, and Windows XP*,  
<http://www.ibm.com/link/developerworks/?open&t=grl,l=252,p=pipes>
- BRADFORS Ed, *Managing processes and threads*,  
<http://www.ibm.com/link/developerworks/?Open&t=grl,l=252,p=mgth>

- Bugtraq, <http://www.bugtraq.org>
- Bull, <http://www.bull.com/fr>
- Business Wire, *Study by Evans Data Corp. Reveals That International Developers Twice as Likely as North Americans to Primarily Target Linux*, [http://www.businesswire.com/cgi-bin/f\\_headline.cgi?bw.111301/213170209](http://www.businesswire.com/cgi-bin/f_headline.cgi?bw.111301/213170209)
- Business Wire, *IBM And Business Partners Celebrate Milestone in Growth of Linux Applications; Linux Applications Increase More Than 30 Percent*, [http://www.businesswire.com/cgi-bin/f\\_headline.cgi?bw.062701/211782585&ticker=IBM](http://www.businesswire.com/cgi-bin/f_headline.cgi?bw.062701/211782585&ticker=IBM)
- CNET.COM, <http://www.cnet.com>
- COONOR Deni, *Linux slips slowly into the enterprise realm*, <http://www.nwfusion.com/news/2001/0319specialfocus.html>
- COSTELLO Sam, *RSA: Security in 2002 worse than 2001, exec says*, <http://www.cnn.com/2002/TECH/internet/02/25/2002.security.idg/index.html>
- Cybersource, *Linux vs Windows (The bottom Line)*, [http://www.stampede.org/~matt/linux\\_vs\\_windows\\_pricing\\_comparison.pdf](http://www.stampede.org/~matt/linux_vs_windows_pricing_comparison.pdf)
- Cybersource, *Linux vs Windows (Total Cost of Ownership Comparison)*, [http://www.cyber.com.au/cyber/about/linux\\_vs\\_windows\\_tco\\_comparison.pdf](http://www.cyber.com.au/cyber/about/linux_vs_windows_tco_comparison.pdf)
- DRIEU Benjamin, *Licences, du copyright au copyleft*, <http://www.april.org/articles/divers/licences.html>
- Dr PEELIN Nic et Dr SATCHELL Julian, *Analysis of Impact of Open Source Software*, [http://www.govtalk.gov.uk/documents/QinetiQ\\_OSS\\_rep.pdf](http://www.govtalk.gov.uk/documents/QinetiQ_OSS_rep.pdf)
- Fast Center, <http://www.fastcenter.com/>
- FOUCART Stéphane, *Les géants du secteur s'adaptent*, 25/12/2001, <http://www.lemonde.fr>
- FOUCART Stéphane, *Linux, l'électrons libre qui inquiète Microsoft*, 25/12/2001, <http://www.lemonde.fr>
- FSF, *Qu'est ce qu'un logiciel libre?*, <http://www.fsf.org/philosophy/free-sw.fr.html>
- GABRIEL Richard P. et JOY William N., *Sun Community Source Licence Principles*, <http://www.sun.com/981208/scsl/principles.html>
- Gartner, *Lack of Security Processes Keeps Sending Enterprises to Code Red*, <http://www.gartner.com/DisplayDocument?id=336339>

- GHOSH Rishab, *Les marchés « marmites » : un modèle économique pour le commerce de biens et de services gratuits sur l'Internet*, [http://www.linux-france.org/article/these/marmite/fr-cooking\\_pot\\_monoblock.html](http://www.linux-france.org/article/these/marmite/fr-cooking_pot_monoblock.html)
- GHOSH Rishab Aiyer, *Qu'est ce qui motive les développeurs de logiciels libre?*, [http://www.linux-france.org/article/these/interview/torvalds/fr-lt\\_first\\_monday.199803.html](http://www.linux-france.org/article/these/interview/torvalds/fr-lt_first_monday.199803.html)
- GODDEN Frans, *How do Linux and Windows NT measure up in real life?*, <http://gnet.dhs.org/stories/bloor.php3>
- GONZALES-BARAHOMA Jesus, *Development models*, [http://eu.conecta.it/paper/Development\\_models.html](http://eu.conecta.it/paper/Development_models.html)
- GONZALES-BARAHOMA Jesus, *General idea of open source software*, [http://eu.conecta.it/paper/General\\_idea\\_open\\_source.html](http://eu.conecta.it/paper/General_idea_open_source.html)
- HARGROVE William, HOFFMAN Forrest et STERLING Thomas, *The Do-It-Yourself Supercomputer*, <http://www.sciam.com/2001/0801issue/0801hargrove.html>
- HECKER Franck, *Setting up Shop: The Business of Open-Source Software*, <http://www.hecker.org/writings/setting-up-shop.html>
- IBM, <http://www.ibm.be>
- IBM, *It's a matter of trust*, <http://www.ibm.be>
- Idaya, *ISPs believe large IT company adoption of Linux to be benefit not threat reveals Idaya/freeVSD research*, [http://www.idaya.co.uk/news/newsdesk/idaya\\_ispsurvey.phtml?member=ea1df79a5918559b4a8ce19ebbf22b0](http://www.idaya.co.uk/news/newsdesk/idaya_ispsurvey.phtml?member=ea1df79a5918559b4a8ce19ebbf22b0)
- IDC, <http://www.idc.com/itforecaster/itf20000808.stm>
- IEC , *Prefixes for binary multiples*, <http://physics.nist.gov/cuu/Units/binary.htm>
- Infoworld, <http://www.infoworld.com>
- IT-Optics, <http://www.it-optics.com/index2.htm>
- KAVEN Oliver, *Performance Tests: File Server Throughput and Response Time*, <http://www.pcmag.com/article/0,2997,s%253D25068%2526a%253D16554,00.asp>
- KINGDON Jim, *Free Software Models*, <http://www.stromian.com/bizmod.html>
- KIRCH John, *Comparaison Microsoft Windows NT serveur 4.0 - Unix*, <http://www.linux-france.org/article/these/unix-vs-nt/unix-vs-nt.html>

- LANG Bernard, *Logiciels libres et entreprises*,  
[http://www.terminal.sgdg.org/no\\_speciaux/80\\_81/Lang.html](http://www.terminal.sgdg.org/no_speciaux/80_81/Lang.html)
- LEAVITT Neal, *Linux at a turning point?*,  
[http://www.computer.org/computer/homepage/june/ind\\_trends/index.htm](http://www.computer.org/computer/homepage/june/ind_trends/index.htm)
- Les Trophées de la nouvelle économie, <http://www.jdnet.fr/lille/010131arkane.shtm>
- LETTICE John, *Peru mulls Free Software, Gates gives \$550k to Peru Prez*,  
<http://www.theregister.co.uk/content/4/26207.html>
- Linux@Business, <http://www.linux-at-business.com/>
- Mandrake Business Cases, <http://www.mandrakebizcases.com/>
- MANNING Bill, *in-addr version distribution*, <http://www.isi.edu/~bmanning/in-addr-versions.html>
- Microsoft, *System Requirements*,  
<http://www.microsoft.com/windows2000/server/evaluation/sysreqs/default.asp>
- MIDDLETON James, *NT remains hackers' favourite*, <http://www.vnunet.com/News/1116081>
- MILLER Robin, *90% Windows, 5% Mac, 5% Linux? Not true!*,  
<http://www.theregister.co.uk/content/4/19661.html>
- Mind, <http://mind.be/>
- NADEAU Tom, *Microsoft Turns the Screws*, <http://osopinion.com/perl/story/9849.html>
- Netcraft Web Server Survey, <http://www.netcraft.net/survey>
- Netcraft Web Site Finder, <http://www.netcraft.net>
- NF Consulting, <http://www.linux-en-entreprise.com/index.htm>
- NOVAK Kevin et MUELLER Patrick, *Are we there yet*,  
<http://www.networkcomputing.com/1224/1224f1.html>
- Open Cascade, <http://www.opencascade.org/>
- Open Office Project, <http://www.openoffice.org>
- Open Source Initiative, *La définition du logiciel Libre*, <http://www.opensource.org/docs/osd-francais.php>
- Open Source Research Community, <http://opensource.mit.edu/>

- ORLOWSKI Andrew, *No one's using Linux, claims Microsoft*, <http://www.theregister.co.uk/content/4/19662.html>
- PERENS Bruce, *La définition de l'Open Source*, [http://www.linux-france.org/article/these/the\\_osd/fr-the\\_open\\_source\\_definition\\_monoblock.html](http://www.linux-france.org/article/these/the_osd/fr-the_open_source_definition_monoblock.html)
- PERERA Rick, *German Parliament consider Linux switch*, [http://www.idg.net/ic\\_712902\\_1794\\_9-10000.html](http://www.idg.net/ic_712902_1794_9-10000.html)
- Peruvian Activism, <http://www.pimientolinux.com/peru2ms/>
- PESCAROTE John, *Commentary: Another worm, more patches*, <http://news.com.com/2009-1001-273288.html?legacy=cnet&tag=nbs>
- PFAU Thomas, *An Analysis of Microsoft's TCO Comparison - Part 1*, <http://fud-counter.nl.linux.org/tech/TCO.html>
- PFAU Thomas, *An Analysis of Microsoft's TCO Comparison - Part 2*, <http://fud-counter.nl.linux.org/tech/TCO2.html>
- PRASAD G., *Le guide pratique de Linux destiné aux décideurs (Pouvez-vous utiliser Linux dans votre entreprise de façon profitable ?)*, [http://www.linux-france.org/article/these/guide\\_linux/index.html](http://www.linux-france.org/article/these/guide_linux/index.html)
- RAYMOND Eric, *A la conquête de la noosphère*, [http://www.linux-france.org/article/these/noosphere/homesteading-fr\\_monoblock.html](http://www.linux-france.org/article/these/noosphere/homesteading-fr_monoblock.html)
- RAYMOND Eric, *La cathédrale et le bazar*, [http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar\\_monoblock.html](http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar_monoblock.html)
- RAYMOND Eric, *Le chaudron magique*, [http://www.linux-france.org/article/these/magic-cauldron/magic-cauldron-fr\\_monoblock.html](http://www.linux-france.org/article/these/magic-cauldron/magic-cauldron-fr_monoblock.html)
- Redcorp.be, <http://www.redcorp.be>
- Red Hat, *Red Hat Linux 7.3 Technical Detail*, <http://www.redhat.com/software/linux/technical>
- Rothman Jeffrey et BUCKMAN John, *Which OS is Fastest for High-Performance Network Applications?*, <http://www.samag.com/articles/2001/0107/>
- Rothman Jeffrey et BUCKMAN John, *Which OS is Fastest -- FreeBSD Follow-Up*, <http://www.samag.com/articles/2001/0108/>

- s.a., *Les coûts de la bureautique d'entreprise*, [http://webbo.enst-bretagne.fr/tig/logicielLibre/R/I29/cout\\_total\\_invest.html](http://webbo.enst-bretagne.fr/tig/logicielLibre/R/I29/cout_total_invest.html)
- s.a., *Le Logiciel Libre est plus fiable*, <http://www.gnu.org/software/reliability.fr.html>
- s.a., *Les licences commentées*, <http://www.fsf.org/philosophy/license-list.fr.html>
- s.a., *Logiciels libre, une introduction*, [http://www.april.org/articles/divers/intro\\_ll.html](http://www.april.org/articles/divers/intro_ll.html)
- s.a., *Qu'est ce qu'un logiciel?*, <http://www.ccip.fr/irpi/faq/logiciel/definition.htm>
- Security Poral, <http://www.securityportal.com/>
- Security Space, [http://www.securityspace.com/s\\_survey](http://www.securityspace.com/s_survey)
- SGI France, <http://www.sgi.fr/>
- SHANKLAND Stephen, KANE Margaret et LEMOS Robert, *How Linux saved Amazon millions*, <http://news.com.com/2100-1001-275155.html?legacy=cnet&tag=owv>
- SHANKLAND Stephen, *Linux sales surge past competitors*, <http://news.cnet.com/news/0-1003-200-1546430.html>
- Silicon.com, *Ford looks to open source - Microsoft gets worried*, <http://www.silicon.com/bin/bladerunner?30REQEVENT=&REQAUTH=21046&14001REQSUB=REQINT1=45449>
- Standard Performance Evaluation Corporation, <http://www.spec.org/>
- Sun, *ERP Platform-Related Analysis Total Cost of Ownership Study*, <http://www.sun.com/servers/workgroup/tco/metastudy.html>
- Sys Admin, <http://www.sysadminmag.com/>
- Sysformance, *Sites utilizing Microsoft Webserver-Software turn in inferior reliability results in study of Swiss websites*, <http://www.syscontrol.ch/e/news/Serversoftware.html>
- Sysformance, *SWePIX - Swiss Web Performance Index*, <http://www.syscontrol.ch/e/SWePIX/SWePIXe.html>
- TCP, *L'étude IBM*, <http://www.tpc.org/tpch/results/h-ttperf.idc>
- Techrepublic, <http://www.techrepublic.com>
- The GNU Project and the Free Software Foundation, <http://www.gnu.org>
- The Internet Operating System Counter, <http://www.leb.net/hzo/ioscount>



- The Snark, *Linux: The penguin marches on*, <http://www.it-director.com/article.php?id=2332>
- The Wurzler Group, <http://www.jszum.com/>
- Top500 Supercomputers Site, <http://www.top500.org/list/2001/06>
- Trusecure, *Open Source Security: a look at the security benefits of source code access*, [http://www.trusecure.com/html/tspub/whitepapers/open\\_source\\_security5.pdf](http://www.trusecure.com/html/tspub/whitepapers/open_source_security5.pdf)
- University of Wisconsin, *Fuzz Testing of Application Reliability*, <http://www.cs.wisc.edu/~bart/fuzz/fuzz.html>
- VA Linux, <http://www.vasoftware.com/>
- VALLIERE Rob, *Linux as a replacement for Windows 2000*, <http://www.robval.com/linux/desktop/index.asp>
- WHEELER David, *Home Page*, <http://www.dwheeler.com>
- ZDNET, <http://www.zdnet.com/eweek/stories/general/0,11011,2792860,00.html>
- ZDNET, <http://www.zdnet.com/filters/printerfriendly/0,6061,2760874-2,00.html>
- ZDNET, <http://www.zdnet.com/zdnn/stories/news/0,4586,2772060,00.html>
- ZDNET.FR, <http://www.zdnet.fr>
- Zona Research, *The New Religion: Linux and Open Source*, <http://www.zdnet.com/eweek/stories/general/0,11011,2651826,00.html>

# Table des matières

Remerciements

Avant-propos

Plan du mémoire

Introduction

|   |    |
|---|----|
| Chapitre 1: Les logiciels libres, Open Source et Linux.....       | 1  |
| 1.1 Définition.....   | 1  |
| 1.2 Historique GNU/Linux et *BSD.....                             | 1  |
| 1.3 GNU/Linux, ce concurrent de Microsoft.....                    | 3  |
| 1.4 Le concept de logiciel libre.....                             | 5  |
| 1.5 Quelques caractéristiques.....                                | 7  |
| 1.6 Les licences d'utilisation.....                               | 7  |
| 1.6.1 Les licences libres.....                                    | 8  |
| 1.6.2 Etude d'une licence non-libre: la licence Sun .....         | 10 |
| 1.7 L'importance de l'ouverture des sources.....                  | 12 |
| 1.8 Le cas du logiciel libre dans l'Administration française..... | 13 |
| 1.9 Quel futur pour de tels logiciels?.....                       | 16 |
| 1.9.1 De l'enrichissement des logiciels libres.....               | 16 |
| 1.9.2 Les limites de l'approche propriétaire .....                | 19 |

|   |    |
|---|----|
| Chapitre 2 : Business modèles.....  | 22 |
| 2.1 Quelques modèles.....   | 22 |
| 2.1.1 Les modèles économiques financés par la valeur d'utilisation.....       | 22 |
| 2.1.2 Les modèles exploitant une valeur d'acquisition indirecte.....          | 23 |
| 1.Vendre à perte pour se positionner sur un marché .....                      | 23 |
| 2.Gel des gadgets .....   | 23 |
| 3.Donner la recette et ouvrir un restaurant .....                             | 24 |
| 4.Accessoires .....   | 25 |
| 5.Libérer l'avenir, vendre le présent .....                                   | 25 |
| 6.Libérer le logiciel, vendre la marque .....                                 | 25 |
| 7.Libérer le logiciel, vendre le contenu .....                                | 26 |
| 2.2 Quelques exemples .....   | 26 |
| 2.2.1 Le cas Apache: partage des coûts.....                                   | 26 |
| 2.2.2 Le cas Cisco: étaler les risques.....                                   | 27 |
| 2.2.3 Le cas Open-Cascade .....   | 28 |
| 2.2.4.Le cas du Bundestag .....   | 30 |
| Chapitre 3: Les logiciels libres contre les logiciels propriétaires.....      | 31 |
| 3.1 Situation actuelle du logiciel libre et Open Source en entreprises.....   | 31 |
| 3.2 Quels avantages pour les sociétés utilisatrices de logiciels libres?..... | 42 |
| 3.2.1 Compatibilité (matérielle, protocoles, ...)......                       | 42 |
| 3.2.2 Portabilité.....  | 44 |
| 3.2.3. Performance.....   | 45 |
| 3.2.4 Fiabilité.....  | 50 |
| 3.2.5 Sécurité.....   | 54 |
| 3.2.6 Pérennité.....  | 62 |
| 3.2.7 Indépendance des vendeurs.....  | 62 |
| 3.2.8 Support et formation.....   | 62 |
| 3.2.9 TCO.....  | 66 |

|  |     |
|--|-----|
| 3.3 Avantages apparents des logiciels propriétaires.....                                   | 72  |
| 3.4 Quels sont les apports des logiciels Open Source et propriétaires à l'entreprise?..... | 76  |
| 3.4.1 De multiples avantages .....   | 76  |
| 3.4.2 et quelques limites.....   | 79  |
| 3.5 Comparaison chiffrée de solutions de type Microsoft et de type Linux.....              | 81  |
| 3.5.1 Scénario 1: 50 utilisateurs.....   | 86  |
| 3.5.2 Scénario 2: 100 utilisateurs .....   | 88  |
| 3.5.3 Scénario 3: 250 utilisateurs .....   | 90  |
| 3.5.4 Conclusions.....   | 92  |
| Chapitre 4: le cas de l'Administration péruvienne.....                                     | 93  |
| 4.1 Contexte.....  | 93  |
| 4.2 Réaction de Microsoft.....   | 97  |
| 4.3 Réponse du député.....   | 98  |
| Conclusions.....   | 101 |
| Lexique  |     |
| Annexes  |     |
| Bibliographie  |     |