



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Les logiciels libres: une solution alternative aux environnements propriétaires: cas particulier d'une administration publique

Parent, Didier

Award date:
2003

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année académique 2002-2003

Les logiciels libres :
une solution alternative aux
environnements propriétaires
Cas particulier d'une administration publique

Didier Parent

Mémoire présenté en vue de l'obtention du grade de Licencié en Informatique

Résumé

Le but de ce mémoire est de présenter une solution alternative aux environnements basés sur les logiciels propriétaires.

Dans ce travail, je tenterai de décrire les implications d'une migration d'un environnement informatique vers un système utilisant les logiciels libres.

J'apporterai au responsable du projet toutes les informations utiles et nécessaires à la compréhension de la philosophie et du cadre juridique sur lequel reposent les logiciels libres.

Un plan de gestion de la migration est proposé et un cas concret illustre la démarche.

Afin de mieux comprendre l'importance de ce sujet, le travail présente également l'évolution des mentalités qui pousse à favoriser les logiciels libres dans les administrations et décrit quelques initiatives dans le domaine.

Mots clés :

Logiciels propriétaires

Logiciels libres

Migration

Alternative

Philosophie

Abstract

The goal of this memory is to present an alternative to the environments based on the closed source software.

In this work, I will try to describe the implications of a migration of a data-processing environment towards a system using the free or open source software.

I will bring to the person in charge for the project all useful and necessary information to the comprehension of philosophy and the legal framework on which the free software are based.

A management plan of the migration is proposed and a concrete case illustrates the steps.

In order to better understand the importance of this subject, the project also presents the evolution of mentalities that pushes to support the free software in the administrations and described some initiatives in this case.

Keywords:

Closed source software

Free Software

Migration

Alternative

Philosophy

AVANT-PROPOS

Je tiens particulièrement à remercier :

Le professeur Jean Ramaekers, promoteur de ce mémoire, pour sa disponibilité et les conseils judicieux qu'il m'a prodigués tout au long de ce travail.

Les personnes, qui par leurs connaissances approfondies du domaine, m'ont fournis des informations :

Jean-luc Voisin – Mobistar

Eric Dupuis – Facultés Universitaires de Louvain la Neuve

Oliver Schneider – Arafox

Jean-Dominique Veuve – Service Public Fédéral – Personnel et Organisation

Le professeur Lefèvre - Consultant

Les services de l'Union des Villes et Communes de Wallonie.

Les membres du Jury pour l'intérêt qu'ils voudront bien porter à la lecture de ce document.

Toutes les personnes qui, par leur aide et leur soutien, m'ont permis de mener à terme mes études et de réaliser ce travail, en particulier ma famille.

TABLE DES MATIERES

BIBLIOGRAPHIE	7
GLOSSAIRE	8
1. Introduction	11
1.1. Contexte.....	11
1.2. Objectifs	11
1.3. La méthode.....	11
1.4. Le plan.....	12
LE MONDE DES LOGICIELS LIBRES	13
2. Le « monde libre ».....	15
2.1. L'Histoire	15
2.2. Qu'est-ce qu'un <i>logiciel libre</i> ?.....	16
2.3. La licence	17
2.4. La communauté du libre.....	20
2.5. La cathédrale et le bazar	21
2.6. Conclusion.....	22
3. Les qualités et défauts des logiciels libres.....	23
3.1. La Fiabilité	23
3.2. L'Innovation.....	24
3.3. La Pérennité.....	25
3.4. La Sécurité.....	26
3.5. Les Finances	29
3.6. Interopérabilité	30
LE MONDE DES ADMINISTRATIONS	31
4. Le domaine public	33
4.1. Les besoins	36
ANALYSE D'UNE MIGRATION.....	39
5. La gestion du changement.....	41
5.1. Objectifs et contraintes.....	41
5.2. Inventaire du matériel et des logiciels.....	44
5.3. La démarche	44
5.4. Le système d'exploitation	45
5.5. Les distributions	46
1 Le contenu d'une distribution GNU/Linux.....	46
2 Les fournisseurs des distributions	46
3 Le choix d'une distribution.....	47
5.6. Les alternatives logicielles	47
1 Les services sous linux.....	47
2 La bureautique sous Linux	47
3 Atelier logiciel.....	48
5.7. Quelques conseils pour bien démarrer une migration	48

ANALYSE D'UN CAS SPÉCIFIQUE.....	51
6. Etude de migration	53
6.1. Etat des lieux (partie serveurs)	53
6.2. Etat des lieux (partie clients).....	54
6.3. Migration vers Linux.....	54
6.4. Inventaire matériel.....	59
7. Analyse des coûts comparés et plan de migration.....	61
7.1. Les coûts.....	61
7.2. Le planning.....	63
8. CONCLUSIONS	67
Annexes.....	69
1. La Solution que propose Microsoft et Intel.....	71
2. Réponse du Pérou aux remarques de Microsoft.....	75
3. The table of equivalents / replacements / analogs of Windows software in Linux.....	87
4. Tableau financier des pages 53 et 54.....	112

BIBLIOGRAPHIE

<http://gnuwin.epfl.ch/articles/fr/index.html>

GnuWin II : Site qui contient de nombreux articles en français sur le monde des logiciels libres.

<http://LinuxFr.org>

Contient une foule d'articles sur Linux, Palladium, TCPA, DRM

http://ccomb.free.fr/TCPA_Stallman_fr.html

Pouvez-vous faire confiance à votre ordinateur ? Par [Richard Stallman](#) (octobre 2002)

Traduction du document source original en anglais :

<http://newsforge.com/newsforge/02/10/21/1449250.shtml?tid=19>

<http://www.groupeps.be/fichiers/propositions/A287.pdf>

Proposition d'ordonnance relative à l'utilisation de logiciels libres dans les administrations régionales de Bruxelles-Capitale

déposée par Alain Bultot et Michel Mook

<http://www.infonomics.nl/FLOSS/report/index.htm>

International Institute of Infonomics

University of Maastricht, The Netherlands

- Free/Libre and Open Source Software : Survey and Study -

<http://www.opensource.org/docs/definition.php>

Définitions sur les licences de type Open Source

<http://www.sei.cmu.edu/opensystems/>

The Open System Approach at the SEI (Software Engineering Institute)

<http://www.microsoft.com/presspass/features/2002/jul02/0724palladiumwp.asp>

La définition de Palladium par Microsoft

<http://www.droit-technologie.org/>

Droit et Nouvelles Technologies" - Avenue Vanden Thoren 80 - 1160 Bruxelles – Belgique

<http://linuxshop.ru/linuxbegin/win-lin-soft-en/>

The table of equivalents / replacements / analogs of Windows software in Linux

<http://www.aful.org>

Association Francophone des Utilisateurs de Linux et des Logiciels Libres

<http://www.freepatents.org/liberty/>

Logiciels libre, liberté, égalité, business – Information de base sur le monde du libre

GLOSSAIRE des termes et acronymes

ACL	<i>Access Control List</i> - Liste des personnes ou autres entités autorisées, ainsi que de leurs droits d'accès à chaque ressource (liste des droits d'accès).
A.S.W.A.D	<i>Agent-Supported Workflow In Public Administration</i> Consortium composé de partenaires du domaine public et privé utilisant les technologies basées sur les logiciels libres (Administrations, Ministères, Universités, ...).
Backdoor	Porte d'entrée dérobée et non-documentée dans un système informatique. Ce moyen d'accès peut avoir été volontairement ajouté à un logiciel ou peut résulter d'un bogue. Cette fonctionnalité cachée est potentiellement très dangereuse puisque des pirates finiront par la découvrir.
B.D.C	<i>Backup Domain Controller</i> - Contrôleur de domaine de secours qui seconde le PDC en cas de problème.
Bogue	Traduction française de « bug ». Un bogue est une erreur dans la programmation d'un logiciel qui entraîne des dysfonctionnements.
Copyright	Avis de l'existence du droit d'auteur inscrite au début d'un ouvrage, ou d'un logiciel, en vue de le protéger contre toute reproduction ou exploitation illégale
Copyleft	Comme dit la F.S.F : « <i>Les développeurs de logiciels propriétaires utilisent le copyright pour restreindre la liberté des utilisateurs ; nous utilisons le copyleft pour la garantir</i> » http://www.fsf.org/home.fr.html Licence réglementant la mise à disposition des sources, la modification et la redistribution de celles-ci, en garantissant l'impossibilité de les commercialiser.
D.M.C.A	<i>Digital Millenium Act</i> – La transposition européenne est E.U.C.D
D.R.M	<i>Digital Right Management</i> – Gestion des droits numériques Contrôle de contenus protégés par copyright
Extranet	Réseau informatique à caractère commercial, constitué des Intranets de plusieurs entreprises qui communiquent entre elles, à travers le réseau Internet, au moyen d'un serveur Web sécurisé.
Firewall	Dispositif informatique qui permet le passage sélectif des flux d'information entre un réseau interne et un réseau public, ainsi que la neutralisation des tentatives de pénétration en provenance du réseau public.
Firewalling	Action produite par un firewall
Fritz Chip	Nom donné à la puce qui gère la protection des systèmes TCPA. En l'honneur du sénateur Fritz Hollings qui fait pression sur le congrès pour mettre du DRM partout.
F.S.F	<i>Free Software Fondation</i> - organisation fondée par Richard Stallman, à l'origine du projet GNU et des licences GPL et LGPL.
Hacker	Criminel informatique qui exploite les failles dans une procédure d'accès pour casser un système informatique, qui viole l'intégrité de ce système en dérobant, altérant ou détruisant de l'information, ou qui copie frauduleusement des logiciels.
IIS	<i>Internet Information Server</i> - Service Web, FTP et Gopher conçu par Microsoft
Intranet	Réseau informatique privé qui utilise les protocoles de communication et les technologies du réseau Internet.
LDAP	<i>Lightweight Directory Access Protocol</i> - Protocole de gestion d'annuaires

Licence	La licence d'un logiciel est un texte qui conditionne l'utilisation du programme (droit ou non de le copier, etc.). Dans le cas d'un logiciel propriétaire, il faut acheter une licence pour chaque ordinateur qui utilise le programme. Dans le cas du libre, cette licence sert juste à rappeler que le programme est libre.
Lobbying	Action menée par un groupe de pression auprès de décideurs ou de médias dans le but de défendre une cause ou des intérêts particuliers, ou d'influer sur l'opinion publique.
Logiciels Libres	Logiciel livré avec son code source de manière qu'il puisse être copié, modifié et redistribué, évoluant ainsi de façon continue vers une version plus perfectionnée, dans un contexte de développement coopératif et communautaire.
Logiciels propriétaires	Logiciel qui est spécifique à un développeur donné, ce qui veut dire qu'il n'est pas nécessairement conforme à une norme ou un standard, qu'il n'est pas toujours compatible avec d'autres logiciels, qu'il est protégé par le droit d'auteur et qu'il faut l'acheter ou acquérir une licence pour pouvoir l'utiliser.
Middleware	Dans un système informatique en réseau structuré par couches, ensemble que constitue la partie logicielle qui est destinée à assurer l'interfaçage entre les divers modules. Le terme français est <i>logiciel des couches intermédiaires</i>
Palladium	Technologie de Microsoft basée sur des matériels proches de TCPA pour empêcher l'exécution de contenus non certifiés.
Patch	Morceau de code source devant être ajouté à un logiciel pour corriger un bogue ou ajouter une fonctionnalité. On utilise parfois la traduction française «rustine».
P.D.C	<i>Primary Domain Controller</i> - Contrôleur de domaine principal. Secondé par le BDC
Proxy	Logiciel de sécurité servant d'intermédiaire entre le navigateur d'un internaute utilisant un réseau local protégé par un coupe-feu, et le serveur Web qu'il veut consulter, permettant ainsi à des données de sortir du réseau local et d'y entrer, sans mettre en danger la sécurité du réseau.
Proxy Cache	Serveur mandataire doté d'une mémoire de masse pouvant stocker et mettre à jour régulièrement les pages Web les plus consultées par un groupe d'internautes, raccourcissant ainsi le temps d'accès à ces pages consultables en différé, et réduisant le trafic sur le réseau.
Reverse Proxy	Action de cacher en sortie, des informations internes au système
S.A.M	<i>System Availability Monitor</i> - Fonction utilisée pour la reconnaissance des incidents dans le cadre de la gestion des systèmes.
Système d'exploitation	« Premier programme » de l'ordinateur : c'est lui qui sert d'interface entre l'utilisateur et les logiciels employés. Lorsqu'on allume un ordinateur, le système d'exploitation se lance, et c'est au sein de cet environnement que l'utilisateur fait fonctionner ses programmes. Ces programmes sont conçus en fonction de ce système d'exploitation. Windows (et son ancêtre MS-DOS) est le système d'exploitation de Microsoft, fonctionnant sur les PC.
T.C.P.A	<i>Trusted Computing Platform Alliance</i> – Technologie matériel
WINS	<i>Windows Internet Name Service</i> - Service d'attribution de nom de domaine sous Windows, qui remplace le numéro Internet d'un ordinateur par une adresse Interne, constituée d'abréviations alphabétiques plus faciles à interpréter et à mémoriser que des chiffres.

1. Introduction

1.1. Contexte

L'évolution des systèmes informatiques et la dépendance de ceux-ci par rapport aux grands éditeurs de logiciels, demandent une réflexion approfondie sur l'utilisation d'autres types de logiciels répondant aux objectifs d'une administration ou d'une entreprise.

La qualité actuelle des *logiciels libres* offre une solution alternative aux *logiciels propriétaires*. Le périmètre fonctionnel couvert par les logiciels s'accroît régulièrement et s'étend maintenant des serveurs aux applications clientes. L'offre de services s'est multipliée et diversifiée.

De nombreux gouvernements introduisent aujourd'hui des projets de loi favorisant l'utilisation des logiciels libres dans le cadre de leur mission. D'autres proposent simplement une réflexion sur ce type de projet.

1.2. Objectifs

Ce document poursuit un double objectif :

1. Il est présenté comme mémoire de fin d'études de la « *Licence en informatique* » par Didier PARENT dans le cadre des ses études effectuées à l'Institut d'informatique des Facultés Universitaires Notre Dame de NAMUR – Belgique.
2. Il s'agit également d'un travail dont le but est de permettre l'approche des logiciels dits « *libres* » et de connaître les implications d'une migration d'un environnement informatique vers les logiciels libres.

Il ne s'agit donc pas d'écrire un livre technique, il n'existe pas de quantité disponible, et ce travail n'apporterait pas grand chose. Ce travail tentera de proposer la solution alternative aux logiciels propriétaires.

Il donnera les informations nécessaires au responsable du projet concernant la compréhension de la philosophie, le cadre législatif, et les alternatives techniques.

1.3. La méthode

N'ayant au départ aucune connaissance des logiciels libres, ma démarche a d'abord été de me familiariser avec la philosophie, la technique de base, les avantages et les inconvénients des logiciels *open sources*. L'Internet est une source d'information inépuisable dans ce domaine. En effet, le monde des logiciels libres n'existerait pas sans ce moyen de communication qui n'a aucune frontière. Le salon *Solutions Linux* du CNIT Paris la Défense de 2003, fût également l'occasion pour moi, d'approcher des acteurs du monde libre, et de lier connaissance avec des belges impliqués dans la politique informatique des administrations.

Par la suite, j'ai analysé l'environnement du domaine public et recherché en quoi les logiciels libres pourraient apporter des solutions d'avenir. A cette fin, j'ai recherché les

projets européens, belges voire mondiaux qui tendent à promouvoir les logiciels libres dans les administrations.

Nous avons ainsi déterminé les objectifs et les contraintes d'une migration d'un environnement propriétaire vers une système basé sur les logiciels libres.

Pour mettre en évidence le cheminement que doit emprunter une administration qui souhaite migrer son informatique vers les logiciels libres, nous avons pris le cas concret d'une administration belge et étudié l'environnement existant en comparant les coûts d'une migration Windows 2000 ou Linux et en établissant un plan de migration sur trois ans.

Cette partie du travail n'est pas une étude approfondie de l'administration choisie mais permet d'avoir une approche complète de la démarche que devrait suivre le gestionnaire informatique d'une administration publique quelconque. Elle reprend les grandes étapes de la migration :

- L'étude de l'existant
- L'évaluation des coûts
- Le plan de migration

Elle décrit également les contraintes, les objectifs, les questions qu'il faut se poser, de même que la mobilisation nécessaire.

1.4. Le plan

Après une brève description du monde des logiciels *Open Source* (chapitre 2), je présente les qualités et défauts des logiciels libres au chapitre 3

Une présentation des avancées actuelles dans le domaine public avec une idée des objectifs qui poussent à la migration des systèmes informatiques des administrations, est proposé au chapitre 4.

Le chapitre 5 décrit la manière de gérer la migration, les questions qui en découlent, les contraintes techniques et humaines. Enfin, je termine en donnant quelques alternatives concernant les logiciels libres.

Sur base d'un cas concret le chapitre 6 propose l'analyse d'une administration et présente la démarche exposée au chapitre précédent.

Une analyse des coûts et un plan sur 3 ans présente un outil utilisable par d'autres administrations.

Les chapitre 8 reprend la conclusion et précède une annexe présentant une information sur des aspects induits par ce travail, tels que la solution Palladium proposée par Microsoft.

Le monde des logiciels libres

2. Le « monde libre »

2.1. L'Histoire

De nos jours, le logiciel libre le plus connu, est sans aucun doute le logiciel « *Linux* ». En effet, ce système d'exploitation⁽¹⁾ est devenu, avec les années, la première alternative aux systèmes d'exploitation propriétaires tels que Windows, Unix, etc...

Unix était un système d'exploitation né en 1969 et développé par la société de télécommunications AT & T. Pour conserver son monopole dans les télécommunications, AT & T avait renoncé à avoir une activité commerciale en informatique. Dès lors, la société n'avait pas le droit d'empêcher les informaticiens (notamment dans les universités) d'avoir accès au code source du programme.

En 1984, la Cour suprême américaine brise le monopole de AT & T (c'est la naissance des « baby bells », multiples petites compagnies de téléphone). La société recouvre dans le même temps son droit de commercialiser des produits informatiques. Unix ne peut donc plus être librement copié et modifié.

C'est alors que Richard Stallman⁽²⁾ (pour l'anecdote, on dit que c'est le fait de ne pas avoir accès au code source du programme pilotant sa nouvelle imprimante qui a été pour lui le déclic), décide de lancer le projet G.N.U⁽³⁾ en fondant la F.S.F (*Free Software Fondation*, Fondation du Logiciel Libre). Il développe alors les concepts juridiques de *CopyLeft* et de licence G.P.L (General Public Licence) et met sur pied un certain nombre de logiciels compatibles avec Unix. [Richard Stallman, « Le manifeste GNU », 1984]

Alors même que le principe fait rapidement des émules et que de nombreux logiciels libres voient le jour, le système d'exploitation Hurd, que Richard Stallman tente de mettre au point, n'avance que très lentement. Parallèlement, un étudiant finlandais, Linus Torvalds⁽⁴⁾, passionné d'informatique et passablement agacé par « *cette architecture minable avec ce système d'exploitation MS-DOS minable [l'ancêtre de Windows] des PC* », décide de mettre sur pied un système d'exploitation qu'il baptise Linux, contraction de son prénom et d'Unix. [Jean-Paul Smets-Solanes et Benoît Faucon, *Logiciels libres : Liberté, égalité, business*, Edispher, 1999]

Le 25 août 1991, il poste un message sur Internet annonçant la création de son système d'exploitation. Aussitôt, des centaines de développeurs du monde entier se joignent à lui, utilisant notamment les outils déjà mis au point par la *Free Software Foundation*. En 1994 sort une première version aboutie de Linux ; en 1999, la version 2.2 est lancée.

En 2003, le nombre d'utilisateurs de Linux s'élèverait à plus de 10 millions. On estime que 30% des serveurs vendus dans le monde fonctionnaient sous Linux.

⁽¹⁾ Système qui gère les ressources matérielles et qui permet la communication utilisateur/matériel

⁽²⁾ Fondateur de la Free Software Fondation et précurseur dans la philosophie Open Source

⁽³⁾ G.N.U pour « *Gnu is Not Unix* » – « *Gnu n'est pas Unix* » le G n'ayant pas de signification propre puisqu'il relance le mot GNU

⁽⁴⁾ Etudiant finlandais qui mettra en route le projet Linux

Tous les secteurs sont concernés : les banques, les opérateurs Internet et télécoms, les compagnies d'assurances, l'industrie. L'absence de licence payante serait un facteur d'économie. Ainsi, selon un responsable du Ministère de la Culture en France, la migration de petits serveurs vers Linux aurait permis une diminution du coût total d'utilisation de 30 à 50%. Mais la gratuité n'est pas tout !

2.2. Qu'est-ce qu'un logiciel libre ?

Limiter le phénomène Open Source à la gratuité des logiciels serait une erreur d'analyse, car cela reviendrait à nier une profonde révolution culturelle et sociologique du monde des informaticiens.

Le *logiciel libre* est caractérisé par deux éléments :

- L'*Open source* : signifie que les lignes de code qui composent sa source doivent être accessibles à tous, **sans restrictions de modifications ou d'utilisations** ;
- Une licence dite *libre* : une licence qui ne permet à personne d'en interdire l'utilisation.

A l'inverse, les logiciels *propriétaires*, sont soumis à une licence de copyright qui en interdit l'usage gratuit et la reproduction. Même si certains paramètres du logiciel permettent d'en modifier l'aspect ou le fonctionnement, l'adaptabilité de celui-ci aux besoins d'une entreprise n'est pas possible.

« Pour protéger vos droits, nous devons apporter des restrictions, qui vont interdire à quiconque de vous dénier ces droits ou de vous demander de vous en désister », explique le préambule de la GPL (*General Public Licence*⁽¹⁾), la licence mise au point par Stallman au sein de la *Free Software Foundation*. Il s'agit de détourner la protection habituelle du copyright pour créer un "copyleft" : jeu de mots subtil qui repose sur l'idée que l'auteur laisse faire des copies. Un logiciel sous licence GPL a un auteur dûment mentionné, mais celui-ci énonce sous le copyright que « ce programme est un logiciel libre ; vous pouvez le redistribuer et/ou le modifier conformément aux dispositions de la Licence Publique Générale GNU ». [Traduction française « non-officielle » (la licence ne peut être utilisée qu'en anglais) de la GPL, disponible sur www.april.org/gnu/gpl_french.html]

Sans cette protection, un utilisateur qui voudrait utiliser le travail d'un autre, y effectuer des modifications et le commercialiser sous forme propriétaire, pourrait parfaitement le faire.

Le credo des tenants du logiciel libre réside dans le fait que le copyright (l'auteur conserve ses droits et les concède aux usagers) et *a fortiori* le brevet (qui offre une protection encore plus forte : les inventeurs d'un procédé en conservent le contrôle pendant 20 ans), n'ont pas de sens pour un programme informatique.

Comparé à un théorème mathématique, « la plupart des brevets (aux États-Unis et au Japon) sur le logiciel résultent plus d'une découverte d'une propriété mathématique, d'un algorithme ou d'une méthode d'organisation, que de la mise au point d'un procédé industriel original et complexe ». Or, comme le note Bernard Lang, « On imagine la

⁽¹⁾ Beaucoup d'autres licences existent. Ex : la licence BSD de l'université de Berkeley en Californie, la licence MIT le Massachusetts Institute of Technology, etc...

frustration des mathématiciens si on leur disait tout à coup que les théorèmes sont une propriété privée, qu'il faut payer pour avoir le droit de les utiliser et que, en outre, les preuves étant secrètes, ils doivent donc faire confiance à la société qui les vend, tout en spécifiant qu'elle ne saurait être tenue pour responsable des erreurs éventuelles. La confiance que l'on peut accorder à une théorie mathématique est le résultat d'un processus social fondé sur la libre circulation de l'information dans la communauté scientifique : elle se développe par la coopération, la critique, la concurrence, les ajouts et transformations, en étant à chaque étape soumise à l'appréciation et au contrôle des pairs. La preuve mathématique est en théorie un objet rigoureux, mais en pratique si complexe que seul ce processus en garantit la crédibilité. Il en va d'ailleurs de même pour bien d'autres disciplines scientifiques, et c'est aussi comme cela que se développent efficacement les programmes informatiques. ». [Bernard Lang, « Internet libère les logiciels », *La Recherche*, n°328, février 2000]

2.3. La licence

Des définitions des logiciels libres ont été produites par deux organismes : la Free Software Foundation et l'Open Source Initiative. Ces définitions sont, en pratique, équivalentes.

Pour la Free Software Foundation (FSF), un logiciel est considéré comme libre si sa licence accorde les quatre libertés suivantes à l'utilisateur :

- La liberté d'exécuter le programme, pour tous les usages.
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à ses besoins. Pour ceci l'accès au code source est une condition requise.
- La liberté de redistribuer des copies.
- La liberté d'améliorer le programme et de publier ses améliorations, pour en faire profiter toute la communauté d'utilisateurs et de développeurs. Pour ceci l'accès au code source est une condition requise.

La FSF a créé le concept de « gauche d'auteur » (*copyleft*). Celui-ci définit une licence qui reprend les quatre libertés décrites ci-dessus et dont les termes doivent être repris à l'identique en cas de nouvelle distribution. Ceci permet d'éviter qu'une distribution du logiciel modifié restreigne les droits initiaux.

⇒ C'est donc le contenu de la licence qui permet de qualifier le logiciel de libre ou de propriétaire, et non la simple disponibilité des sources.

Alors que la définition de la FSF place l'accent sur les grands principes à respecter pour qu'un logiciel puisse être qualifié de logiciel libre, les critères de l'Open Source Initiative (OSI) sont plus pratiques et plus techniques. Ils permettent de déterminer la nature libre ou propriétaire d'une licence de logiciel. Ils se décomposent en neuf points :

1. Redistribution libre

La licence ne doit pas restreindre la vente ou la distribution du logiciel libre intégré dans un 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 de l'obtenir, 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-compileur 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 fichiers patches avec le code source, dans le but de modifier le programme à la compilation. L'auteur peut ainsi garantir l'intégrité de son code source dans le processus de diffusion successive du logiciel. 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 différent du logiciel d'origine.

5. Absence de discrimination envers des personnes ou des groupes

La licence ne doit pas être discriminatoire 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.

[Réutilisation des logiciels et Bouquet du libre - <http://www.atica.pm.gouv.fr/>]

Ces différences de terminologie n'impliquent aucune différence pratique. Les logiciels libres et les logiciels à code source ouvert (*Open Source*) désignent la même réalité.

La différence se situe à un niveau théorique. Les tenants du logiciel libre emploient ce terme pour insister sur les quatre libertés attachées à un logiciel. Ces quatre libertés sont des critères mais également des justifications éthiques à l'existence des logiciels libres. Les tenants de l'emploi du terme « code source ouvert » (*Open Source*) insistent sur les caractéristiques opérationnelles, et en particulier la disponibilité du code source sans tenter de fournir de justification éthique.

Certaines licences, comme la licence GPL, interdisent de construire une solution propriétaire à partir de logiciels libres. D'autres licences, comme les licences de type MIT ou BSD, permettent les travaux dérivés propriétaires. Cette possibilité offerte à l'exploitation propriétaire d'une base de code constitue la principale distinction entre les différentes licences de logiciels libres.

L'ambiguïté contenue en anglais dans le terme *free* pourrait laisser supposer qu'un logiciel libre est forcément gratuit. Or il n'en est rien. Le support physique de distribution, l'assistance fonctionnelle et technique, la formation ou encore l'adaptation du produit peuvent être soumis à rétribution, et c'est généralement le cas. Le terme *free* est associé à la notion de liberté et non de gratuité.

En parallèle avec les logiciels libres, il faut distinguer les *freeware* et les *shareware*.

- Les *freeware* sont des petits logiciels gratuits fréquemment développés par un développeur unique et dont la vocation est, en général, d'offrir à tout le monde un accessoire bureautique sans grande prétention. Le code source n'est généralement pas disponible.
- Le *shareware* est très similaire au *freeware*. La différence est principalement au niveau de la contribution financière demandée par l'auteur du logiciel. Le prix demandé reste toutefois assez faible. De plus, la licence contient en général des restrictions dans la durée d'utilisation, le nombre de copie, les fonctionnalités accessibles. Pour ce type de licence, le code source n'est, en général, pas proposé.

	Gratuit	Distribuable	Usage Illimité	Sources disponibles	Sources modifiables	Tous les dérivés aux mêmes conditions
Propriétaires			✓			
FreeWare	✓	✓	✓			
Shareware		✓				
Libres	✓	✓	✓	✓	✓	✓

2.4. La communauté du libre

Le modèle du logiciel libre se base sur une communauté de développeurs qui ont fondé leur philosophie sur une *utopie*. Cette utopie implique que tout doit être accessible à tous, que l'argent n'est pas le maître mot de toute chose, mais que le plus intéressant est la reconnaissance des autres dans la communauté.

La raison pour laquelle tous ces développeurs prennent le temps pour créer, modifier, améliorer des logiciels, réside principalement dans leur égoïsme. Comme le disait John Hall, président de Linux International : « *Pourquoi les peintres amateurs peignent-ils ? Parce qu'ils cherchent la reconnaissance de leurs pairs. Pour les développeurs de logiciels libres, c'est pareil* » [le journal *Libération*, 23.VIII.1999]

Eric S. Raymond a tenté de théoriser les rapports existant entre les linuxiens : « *Il est très clair que la société des développeurs Open Source est en fait une société du don. A l'intérieur de celle-ci, il n'existe pas de pénurie de produits de première nécessité — espace disque, bande passante, puissance informatique. Le logiciel est librement partagé. Cette abondance crée une situation où la seule mesure valable de réussite face aux autres est la réputation qu'on se fait auprès de ses pairs.* » [Eric S. Raymond, « À la conquête de la noosphère » [la noosphère est le "territoire des idées"]]

Malgré tout, il serait naïf de considérer que le développeur travaille uniquement pour la communauté. Il travaille d'abord pour lui. Le logiciel est donc utile par lui-même à son créateur. La communauté du libre est d'abord une communauté d'intérêt. Donner à tous la possibilité de l'utiliser n'est pas un acte gratuit. Ce faisant, les possibilités d'amélioration deviennent énormes, proportionnelles à la communauté. En effet, même si les utilisateurs ne sont pas capables de corriger les *bugs*, ils peuvent les identifier.

L'apport de la communauté peut également s'orienter vers des domaines plus artistiques : comme l'apport d'un logo ou la modification des interfaces. Il peut également s'orienter vers des besoins plus classiques tels que la création d'une documentation ou la traduction dans une autre langue.

Plus il y a d'utilisateurs, meilleur sera le programme.

Plus un projet est bon et meilleur par rapport à ses concurrents⁽¹⁾, plus vite il se développera. « *Les ressources financières pour payer la bande passante ou pour pouvoir*

⁽¹⁾ Concurrent est ici considéré comme un autre projet libre qui intellectuellement serait en concurrence. La concurrence vers le projet qui regroupe le plus d'adhérents

se nourrir sont des commodités. Les programmeurs représentent la ressource que l'on cherche à acquérir. Dans le monde des logiciels propriétaires, c'est le contraire ! Le but est de faire de l'argent, et les développeurs sont des commodités ». [Unix -- Pas de panique ! - Cyrille DUNANT - Emmanuel ECKARD]

Enfin, on remarquera également que si une très grande société peut rassembler des milliers d'ingénieurs, la communauté du libre réunit des millions de personnes qui contribuent plus ou moins à l'édifice. Même Microsoft a dû créer une équipe qui ne sert uniquement qu'à tester les développements. Chaque maillon de la communauté du libre a donc son utilité.

2.5. La cathédrale et le bazar

Comme l'explique Eric S. Raymond, un important contributeur à la communauté du libre, il existe deux méthodes imaginées susceptibles de produire des logiciels :

- la méthode des cathédrales : avec toute la préparation, la structure de gestion et le suivi nécessaire à la création d'une cathédrale ;
- la méthode du bazar : un monde plein d'idées, *de rituels et d'approches différentes à partir duquel un système stable et cohérent ne pourrait apparemment émerger que par une succession de miracles.* [Eric S. Raymond, *La Cathédrale et le Bazar* (traduction de *The Cathedral and the Bazaar*, traduit par Sébastien Blondeel)]

Du point de vue de la programmation, une différence fondamentale entre les deux méthodes. Ainsi, dans le modèle des cathédrales, les bogues et les problèmes de développement représentent des phénomènes difficiles, ennuyeux, insidieux et profonds. Il faut à une poignée de développeurs, testeurs et ingénieurs, des mois de recherche minutieuse avant de vouloir se laisser convaincre que tous les bogues ont été éliminés. D'où des intervalles de temps assez conséquents entre les mises à jour, et l'inévitable déception quand on se rend compte que la mise à jour tant attendue n'est pas parfaite.

Par contre dans le modèle du bazar, les bogues apparaîtront comme un phénomène de surface – ou, en tous cas, sauteront rapidement aux yeux lorsqu'un millier de co-développeurs avides se précipiteront sur toute nouvelle mise à jour.

Il semblerait que la loi de Darwin soit d'application dans cette théorie. *Seuls les plus forts survivent.*

Plusieurs groupes peuvent développer une solution « concurrente ». La plus performante finira par l'emporter car choisie par le plus grand nombre, et deviendra un standard de fait.

Dès lors, il y aura non seulement beaucoup plus de personnes pour déboguer mais également, voire principalement, beaucoup plus de solutions envisagées à la source.

Même si Microsoft se trompe sur un choix technologique, celui-ci aura déjà coûté tellement cher à l'entreprise qu'elle préférera défendre, voire imposer cette mauvaise solution.

2.6. Conclusion

Que représentent les logiciels libres pour notre société ?

Pour des utilisateurs lambda qui ont pris l'habitude de copier illégalement les programmes informatiques (ce que nous sommes tous peu ou prou), la licence libre semble parfois n'être pas grand-chose. Mais l'on oublie ce simple fait.

En théorie, il faut payer des droits pour avoir le loisir de taper son courrier sur son ordinateur. Les logiciels libres ont cette vertu première qui est celle de remettre les programmes informatiques à leur vraie place, celle de simples outils qui appartiennent à tout le monde (peut-on breveter des tournures de langage ?).

Les logiciels libres sont aussi la réalisation concrète d'une utopie où la propriété et l'individualisme sont abolis.

3. Les qualités et défauts des logiciels libres.

Les qualités des logiciels libres tendent vers une amélioration constante qui atteint maintenant des niveaux remarquables. J'essayerai ici d'expliquer les raisons de ces qualités et mettrai également l'accent sur les défauts liés aux logiciels libres.

3.1. La Fiabilité

La qualité des logiciels, c'est-à-dire entre autre la fiabilité, est l'une des questions majeures à soulever si l'on désire comparer le logiciel libre au logiciel propriétaire. Il est admis que cette notion de qualité technique est, d'une manière générale corrélée au faible nombre de bogues et à la capacité de les corriger efficacement.

Plusieurs facteurs concourent à l'établissement de cette fiabilité

- Le projet naît d'un besoin d'un ou plusieurs développeurs qui sont réunis par une même démarche volontaire. Ils sont motivés par le besoin qui a suscité le projet. Seuls les plus performants seront reconnus, par le gestionnaire du projet pour faire évoluer celui-ci. Cette démarche motivante incite à la vigilance dans la qualité du développement et l'élimination des bogues.
- Le but d'un projet est la génération d'une application, d'un code source, sans interférence avec d'autres objectifs qui pourraient aller à l'encontre de la qualité de ce code. Par exemple, une opportunité de sortie (« release ») calquée sur une stratégie commerciale ou marketing comme c'est fréquent dans le monde du logiciel propriétaire. Pour diverses raisons, on assiste souvent à des sorties prématurées de logiciels parfois quasi inutilisables. Plus grave, certains éditeurs de logiciels propriétaires sont soupçonnés régulièrement d'éviter volontairement la correction de petits bogues pour rythmer et justifier des mises à jour payantes du logiciel.
- N'oublions pas que la seule mesure valable de réussite face aux autres est la réputation qu'on se fait auprès de ses pairs. Il existe donc une forte concurrence entre les développeurs du projet pour gagner en prestige ou en influence. Cette compétition s'exerce d'autant plus que la transparence sur le code source est totale. Quoi de plus stimulant que de savoir que la qualité du code source sera surveillée et jugée par vos pairs, et que de ce jugement dépendra votre influence et votre rôle dans le projet. De plus la version définitive du logiciel n'intégrera qu'une portion des développements proposés. Seuls les meilleurs seront récompensés de leurs efforts. Des efforts que l'on ne pourrait demander à des développeurs non consentant.
- Finalement la mobilisation des ressources pour écrire un logiciel libre est incompatible avec les finances d'un éditeur de logiciels propriétaires, aussi puissant soit-il. Il est en effet fréquent que plusieurs dizaines de développeurs aient réfléchi, codé, amélioré la même portion de code source : un procédé de contrôle qualité certes trop luxueux pour un éditeur propriétaire mais évidemment très efficace!

Les avantages : le logiciel libre présente l'avantage de reposer sur une équipe motivée dont les ressources humaines et matérielles sont quasi sans limites. Le mode même de fonctionnement de la communauté du libre fait qu'elle produit des logiciels de qualité beaucoup plus vite que n'importe quelle entreprise privée. La correction des éventuels bogues est immédiate, accessible à tous et vérifiable par tous.

Les inconvénients : la fiabilité repose sur la bonne volonté des testeurs et développeurs de signaler ou de corriger les erreurs. Leur survie financière n'en dépend pas.

3.2. L'Innovation

Durant des décennies, la tentative d'obtenir des standards par consensus a rarement abouti et a laissé place aux standards « de facto ». Entre l'époque où chaque constructeur d'ordinateur avait son propre système d'exploitation, incompatible avec celui de ses concurrents, et l'ère Microsoft imposant un monopole de fait via ses logiciels DOS et Windows, la marge de manœuvre des entreprises et des administrations, pour faire évoluer leur système d'information, n'a jamais été très importante. Cette prévalence faite au standard « de facto » aboutit à instaurer des cadres techniques plus ou moins rigides, même si l'on ne peut nier que, parallèlement, de réels progrès technologiques ont été réalisés au bénéfice des clients.

Par opposition aux logiciels commerciaux distribués sous leur forme exécutable et ne pouvant être modifiés, les logiciels libres sont diffusés sous forme de code source, donc adaptables aux besoins de l'utilisateur. La licence GPL (General Public Licence), qui régit la plupart des logiciels libres, spécifie que le code source peut être remanié à condition que ces modifications soient publiées et que les auteurs soient mentionnés. Elle préserve donc les droits de l'auteur en lui donnant le moyen de contrôler le devenir de son logiciel. Il existerait aujourd'hui plus de 11 000 logiciels libres dont 3 900 dédiés aux serveurs et au "middleware".

Les facteurs qui influencent la qualité d'un logiciel entraînent les mêmes résultats d'un point de vue de l'innovation. Les ressources disponibles en homme, en matériel et surtout en force de production entraîne inmanquablement l'innovation vers des possibilités quasi illimitées.

Cette évolution rapide suscite naturellement des critiques : l'absence de garantie, une mauvaise visibilité sur les évolutions futures et surtout le risque de fragmentation de l'offre. Sur ce point, la communauté libre s'est organisée. Par exemple, la publication de Linux Standard Base (LSB), spécification technique décrivant le code logiciel commun à toutes les distributions, permet une interopérabilité accrue entre les différentes versions de Linux. Cette initiative du Free Standards Group évite ce qui s'était passé en son temps en matière d'incompatibilité des versions Unix.

Malgré tout pour qu'un projet arrive à maturité, il faut que le coaching de celui-ci soit bien mené. A cette fin, il faut que la communauté accepte : que celui que l'on considère comme le dépositaire du projet, de par son charisme et ses compétences, soit aussi celui qui fasse les choix et qui oriente l'évolution du logiciel. Dans le cas contraire, il faut que la communauté s'auto régularise par consensus afin de déterminer quelle est l'innovation qui touche le plus d'utilisateurs du logiciel.

Les avantages : l'innovation dans le monde libre est poussée par :

- le besoin de nouvelles fonctionnalités demandées par le co-développeur lui-même ou par la communauté
- le nombre de co-développeurs qui interviennent dans le projet
- Le désir d'être reconnu pour son talent de développeur par d'autres développeurs de talent (plus que par le salaire d'un employeur)

Les inconvénients : les dissensions entre les développeurs ou gestionnaires du projet peuvent entraîner l'abandon de celui-ci ou provoquer une démultiplication d'un projet.

3.3. La Pérennité

La définition du Larousse 2003 est la suivante : *Caractère de ce qui dure toujours ou très longtemps* ».

Mon expérience dans le domaine de la programmation ma fréquemment amené à remarquer que la première inquiétude du *prospect*⁽¹⁾ était la solidité de l'entreprise afin de pérenniser son futur investissement. Pour avoir changé de société plusieurs fois, j'ai pu remarquer que cette inquiétude avait sa raison d'être. Les sociétés qui nous proposent, aujourd'hui leurs logiciels seront-elles toujours là demain pour les maintenir et les faire évoluer.

L'abandon unilatéral de maintenance de la part d'un éditeur propriétaire est de fait, de plus en plus fréquent, et souvent prévu contractuellement!

Le piège se referme encore un peu plus de nos jours lorsque l'on constate la nécessité croissante d'une interopérabilité globale, dans un contexte d'entreprise étendue où les clients, les fournisseurs et les partenaires doivent communiquer avec les mêmes protocoles, les mêmes formats de données, etc. Ceci pousse chacun, insidieusement, vers des monopoles de fait. Dans ces logiciels, le client se trouve souvent en situation de payer des mises à jour qui corrigent en fait des vices cachés ou pire avec un logiciel que l'éditeur ne maintient plus.

Dans le cas des logiciels libres, la pérennité est inaliénable puisque qu'elle ne dépend plus de la survie ou des revirements d'un éditeur propriétaire, seul autorisé à lire et à modifier le code source du logiciel.

L'entreprise utilisatrice aura toujours la liberté de mettre en compétition plusieurs fournisseurs sur la maintenance, la garantie ou l'évolution d'un logiciel libre qu'elle utilise.

A condition de prendre un logiciel libre ayant déjà fait ses preuves et accepté comme tel par son milieu d'activité.

Le logiciel n'est pas non plus lié à un développeur mais à une communauté suffisamment nombreuse pour continuer le suivi d'un projet, même si les co-développeurs changent fréquemment. Le logiciel libre mérite bien son nom : il libère l'entreprise de la relation déséquilibrée qu'entretiennent les éditeurs de logiciels propriétaires avec leurs clients.

Attention donc au choix d'un projet libre, il faut regarder le nombre et la qualité des contributeurs !!!

⁽¹⁾ Futur client ou client potentiel

Les avantages : Les logiciels libres ont une durée de vie qui suit les besoins des utilisateurs et non les plans de marketing.

Le fait de travailler dans un monde open source permet également de faire évoluer le logiciel en interne ou de trouver un prestataire d'accord de faire la maintenance de ce logiciel.

Les inconvénients : Le gage de la pérennité pour les logiciels libres repose notamment sur la taille de la communauté de développeurs et de clients. Nous devons prendre en compte ce critère en sélectionnant un logiciel libre.

De nombreux projets intéressants ont été abandonnés par manque de contributeurs.

3.4. La Sécurité

Les entreprises et les administrations ont un impératif de protection du patrimoine d'information qui se double aujourd'hui d'obligations en matière de responsabilité légale. Ainsi, en France, le nouveau Code Pénal par son article 226-17 instaure un délit de manquement à la sécurité des informations nominatives. Une entreprise ou une administration peut ainsi se voir reprocher un manque de vigilance et de protection de données confidentielles à l'occasion d'un accès malveillant (par exemple dans le domaine des paiements électroniques). Cette responsabilité n'est pas à prendre à la légère lorsque l'on sait que 60 % environ de sociétés dans le monde ont subi au moins une fraude de leur système d'information.

Pour garantir un haut niveau de sécurité, plusieurs conditions doivent être réunies. La première est certainement d'avoir une vision précise des failles de son système. Ce qui implique une connaissance approfondie des éléments de sécurité des systèmes d'exploitation, des pare-feu et des divers applicatifs. Trop souvent, faute de transparence, les Directions des Systèmes d'Information ne peuvent s'en remettre qu'au sérieux de leurs fournisseurs et à leur éthique. Mais l'expérience montre qu'en poussant les investigations, certains produits ont révélé des pièges : des mouchards par exemple ou des « outils de flicage » qui renseignent l'éditeur sur le comportement de son client. C'est ce qui explique que de plus en plus d'utilisateurs sont réticents à utiliser certains produits.

Les sociétés de logiciels dits propriétaires rétorquent fréquemment que les logiciels libres ont des problèmes en terme de sécurité informatique.

Cette affirmation mérite d'être étudiée précisément puisque la sécurité représente un enjeu majeur à l'heure où tous les systèmes ou presque sont interconnectés à Internet.

De nos jours la sécurité d'un logiciel n'est plus gérée par un mot de passe inscrit *en clair* dans un logiciel. La sécurité dépend plus d'un protocole de communication ou d'un algorithme de chiffrement. La sécurité est donc d'autant mieux assurée que les protocoles et les algorithmes utilisés sont solides et, bien sûr, qu'ils ne comportent pas de failles, c'est-à-dire d'erreurs de programmation mettant en jeu la sécurité.

S'il faut admettre que la publication du code source peut profiter à une personne mal intentionnée, elle permet aussi à de nombreux experts en sécurité indépendants du projet, d'auditer le code source et donc de trouver et de corriger des failles de sécurité éventuelles. Le jeu perpétuel du glaive et du bouclier se retrouverait de nouveau à égalité si l'on ne

prenait pas en compte une grande part des audits qui ont lieu avant même la sortie officielle du logiciel, et donc avant que les failles deviennent une menace pour les utilisateurs.

La visibilité du code source permet cette vérification, mais il faut malgré tout comprendre que c'est pas du tout évident.

Néanmoins, cette publication du code source est un avantage indéniable, car l'entreprise est ainsi à même de définir son propre dispositif en fonction de ses besoins spécifiques, d'où une meilleure intégration entre applications et outils de sécurité. C'est pourquoi la communauté de l'Open Source a plus fait pour la sécurité que beaucoup d'éditeurs réunis.

L'adaptation du dispositif de sécurité devant être permanente compte tenu de l'évolution des technologies, l'entreprise a la capacité de gérer cette évolution constante **si elle se dote des moyens nécessaires.**

À l'opposé, si la non-publication des codes sources ne permet pas à des experts indépendants un audit sur la sécurité préalable à la sortie commerciale d'un logiciel propriétaire, elle n'empêche absolument pas un pirate d'exploiter une faille quelconque. À titre d'illustration, on citera le cas du logiciel Internet Explorer dont le code source n'est pas visible puisqu'il n'est pas un logiciel libre, mais qui défraie très régulièrement la chronique.

Tout le monde sait que baser la sécurité sur un programme propriétaire n'est pas plus sécurisant. Cela s'appelle la sécurité par obscurité, cela n'a jamais correctement fonctionné car il existe des désassembleurs, de la communication du code via par exemple un employé mécontent, ... C'est un fait. C'est pourquoi l'open source est généralement considéré comme plus sécurisant qu'un code propriétaire. Malgré tout rien ne prouve qu'il en soit ainsi. Le seul fait qui aille dans le sens de l'open source, c'est qu'un bug de sécurité est en général plus rapidement découvert, et donc plus rapidement corrigé.

On ne dira jamais assez que la sécurité absolue en matière informatique n'existe pas, dès lors qu'un ordinateur relié à Internet est sous tension (environ 70 % des attaques proviennent du réseau interne). Mais pour réduire les risques au maximum, il faut jouer sur tous les tableaux : prévention, information et technologie. Au-delà de l'examen systématique et préventif du code source des logiciels, une information fiable et efficace constitue donc un complément indispensable. Il s'agit de réduire au maximum la fenêtre pendant laquelle des failles de sécurité ont été repérées, alors que les correctifs ne sont pas encore installés.

Telnet, un logiciel de connexion distante des débuts d'Internet mais toujours utilisé, faisait circuler les mots de passe des utilisateurs en clair. Un de ses successeurs, SSH, utilise au contraire des algorithmes de chiffrement sophistiqués. Un des principes qui améliorent la sécurité consiste donc à introduire la cryptographie à tous les niveaux des systèmes d'information : du transport au stockage en passant par l'authentification des utilisateurs. Cela ne garantit pas l'invulnérabilité, c'est impossible, mais cela reste très efficace. Il existe une version libre de SSH (openSSH) et plusieurs versions commerciales voir : <http://www.ssh.com>)

Les efforts de la communauté open source contribuent, entre autres, à résoudre les problèmes de filtrage des données, à améliorer la détection d'intrusion ou encore à mettre en oeuvre une infrastructure à clés publiques et privées.

Cette mise à disposition par la communauté, d'applications libres en matière de sécurité, conforte l'idée qu'il est possible aujourd'hui de bâtir une infrastructure ouverte, sécurisée et administrée à partir de composants en Open Source. En effet, que ce soit au niveau des fonctions de connectivité, de l'infrastructure d'application, de l'intégration entre les applications de front et de back office, de « Groupware » et d'échanges ou de l'administration des ressources, il existe un vaste catalogue d'outils permettant à une entreprise ou à une administration de gérer l'évolution de son système d'information, avec un niveau de maîtrise qui lui assure les meilleures conditions de sécurité.

La sécurité d'un système quelconque n'est pas seulement la responsabilité d'une technologie mais bien plus de l'usage et de l'attention que les gestionnaires du système démontrent. Un Firewall n'est rien sans une configuration adaptée aux besoins et aux risques d'un environnement. Disons que les problèmes principaux en sécurité sont souvent des problèmes humains, c'est-à-dire le manque de procédures (exemples : que faire en cas de la propagation d'un virus, quelle est la politique d'application des patches, existe-t il un DRP (Disaster Recovery Plan),...)

La sécurité d'un environnement doit être étudiée :

- *en termes de risques de pertes d'informations* : la perte d'informations représentera-t-elle réellement un problème majeur pour l'entreprise ? Quel en serait le coût ?
- *en terme de divulgation de l'information* : quelle est l'importance de la divulgation ou non de mon information ? (ex : image de marque)
- *en terme d'utilisation d'un nom* : l'utilisation de mon nom dans des documents, mails, courriers, ... dont je ne suis pas propriétaire est-elle préjudiciable ?
- *en terme de reconstruction de l'environnement* : quelle serait l'incidence d'une catastrophe externe sur mon environnement informatique (incendie, inondation, ...) et combien de temps faudrait-il pour récupérer l'état d'avant catastrophe ?
- *en terme de droits liés aux utilisateurs* : les droits de l'utilisateur final sont-ils limités et doivent-ils l'être en regard des besoins repris dans les 3 premiers points ?
- *en termes technologiques* : mon système peut-il faire face aux différents types d'intrusion ?

La question principale est donc : Quel serait l'impacts d'une intrusion en terme de confidentialité, intégrité des données et disponibilité du SI ?

Ensuite il faut évaluer l'importance financière que l'on veut donner à la sécurité de son système et définir les droits et devoirs des utilisateurs.

Les avantages : un plus grand nombre d'experts peuvent résoudre les problèmes de sécurité potentiels. En cas d'incident, les parades sont souvent délivrées plus rapidement que pour les logiciels propriétaires.

Dans le modèle libre, si l'entreprise se dote des moyens nécessaires, elle est maîtresse de sa sécurité et peut ainsi l'adapter aux besoins spécifiques de son environnement.

Les inconvénients : le logiciel libre n'est réellement plus sécurisé que si tous les acteurs jouent le jeu. Une personne peut découvrir un trou de sécurité, ne pas le déclarer et en tirer un profit quelconque. Les attaques sont maintenant aussi nombreuses que pour les logiciels propriétaires.

3.5. Les Finances

La première liberté offerte par le modèle des logiciels libres, est celle d'utiliser le logiciel sans coût de licence proportionnel au nombre de postes installés, d'utilisateurs, de puissance processeur, etc. On imagine bien sûr sans peine les possibilités d'économies d'échelle pour les entreprises en terme d'achats de licence dont l'impact sur les budgets informatiques est incontrôlable. S'affranchir des politiques tarifaires disproportionnées ou erratiques des éditeurs est déjà très appréciable, mais n'oublions pas la baisse du coût d'utilisation résultant de l'absence totale de dénombrement et de gestion des licences elles-mêmes.

Le cabinet de consultants Cybersource a fait une étude comparant le TCO (Total cost of Ownership, coût total de possession) d'un parc informatique moyen sous Windows, et un autre sous GNU/Linux. Le résultat est limpide : utiliser des logiciels libres ne modifie pas les dépenses matérielles, mais supprime quasiment les dépenses en logiciels, pour une masse salariale légèrement plus coûteuse (un administrateur GNU/Linux est globalement plus compétent et diplômé qu'un administrateur Windows, d'où le surcoût salarial), tout ça pour une économie moyenne d'environ 34.26% (on parle en centaines de milliers d'euros) .

La gratuité ou les prix peu élevés, font de ces outils une solution avantageuse pour les PME, les établissements d'enseignement, les instituts de recherche ou les collectivités locales, notamment pour l'informatique en réseau, les serveurs Intranet ou les postes bureautiques individuels.

Malheureusement, certains outils nécessaires aux entreprises comme les progiciels de gestion ou les jeux éducatifs pour le grand public sont encore absents de cette offre.

D'un point de vue de la maintenance, la fiabilité des produits et surtout la possibilité, par le biais d'Internet, d'obtenir toutes les aides, mises à jours et correctifs nécessaires, permet de substantielles économies.

Les besoins en puissance du matériel sont bien moins grands avec les logiciels libres. Les équipements peuvent donc hériter d'une seconde vie.

En revanche :

- les coûts liés au changement culturel ne sont pas à négliger. Certaines équipes de logistique informatique doivent accepter de prendre en compte un modèle informatique nouveau. Les utilisateurs ne doivent pas être perturbés par une interface graphique non familière ou un passage au libre insuffisamment maîtrisé et perturbant leur activité ;

- les coûts de formation ne sont pas nuls. Ceci vaut pour tous les systèmes. Les logiciels libres offrent des fonctionnalités connues depuis longtemps dans le milieu de l'Internet, mais ces fonctionnalités nouvelles peuvent être perçues comme étant des nouveautés lorsqu'on les utilise pour la première fois.

Les avantages : les logiciels libres sont moins chers puisque gratuits tant au niveau de la licence que de la reproductivité au sein d'un réseau, de la maintenance et du coût matériel.

Les inconvénients : la mise en œuvre est considérée comme plus onéreuse car elle s'appuie sur un service proposé par des spécialistes. Par ailleurs, la compétence interne nécessaire est plus importante et donc plus chère en terme de coût salarial et de coût de formation.

3.6. Interopérabilité

L'interopérabilité c'est la possibilité de travailler avec plusieurs applications qui utilisent le même format de données. Les logiciels libres sont également appelés *les logiciels ouverts* dans le sens où ils utilisent en général un format standard reconnu par toutes les applications.

Le code source est en train de subir le même sort que les protocoles et autres normes qui sont d'ores et déjà créés, modifiés et partagés par tous les acteurs du marché. Pour le bien de tous, les mêmes algorithmes ne seront pas sans cesse réinventés. C'est donc la promesse pour les utilisateurs qu'ils pourront bénéficier d'une véritable interopérabilité dans le cadre d'une concurrence préservée.

Les avantages : les logiciels libres utilisent des standards ouverts

Les inconvénients : pourront-ils encore lire longtemps les fichiers des applications propriétaires qui actuellement sont considérés comme des standards de fait

Le monde des administrations

4. Le domaine public

A l'heure actuelle, plusieurs projets existent, tant en Belgique que dans des pays européens comme la France, l'Allemagne, l'Italie. Ceux-ci visent à favoriser l'utilisation des logiciels libres dans l'administration.

En France, l'agence ATICA⁽¹⁾ propose aux administrations une panoplie complète d'informations sur les logiciels libres. Elle tente de favoriser l'implantation de ce nouvel environnement en proposant des guides présentant les technologies, les démarches, les risques, les équivalents logiciels, etc.

Parallèlement, deux propositions sont actuellement en débat à l'assemblée, afin de donner un cadre réglementaire à ce genre d'initiatives d'utilisation des logiciels libres.

La Communauté Européenne, avec son projet ASWAD vise à créer une solution Intranet basée sur les logiciels libres (et principalement Zope). Cette solution proposera trois axes :

- Le groupeWare : Mail, Agendas partagés, gestion documentaire, ...
- Le WorkFlow : la gestion de suivi des documents
- L'assistance : aide à l'utilisation par le biais de la communauté grâce à des forums et autres systèmes de discussion.

Le but de ce projet est d'optimiser la coopération interne au sein des administrations.

En Belgique, les députés Michel Mook et Alain Bultot, ont déposé une proposition d'ordonnance visant à imposer l'usage des logiciels libres dans les administrations régionales belges et plus particulièrement wallonnes.

Voici, en quelques mots, la présentation du projet.

« A l'heure de l'e-gouvernement et de la multiplication des échanges électroniques entre l'administration et le citoyen, il s'agit de faire respecter dans les faits des libertés fondamentales : accessibilité de l'information, protection de la vie privée, libre choix dans sa consommation personnelle.

Pour les pouvoirs publics, l'objectif est d'assurer la pérennité des archives et une réelle concurrence dans les marchés publics. Un débat parfois technique mais qui contient des enjeux essentiels pour la démocratie. » <http://www.groupeps.be/fichiers/propositions>

Le projet, qui dans un premier temps avait été applaudi par l'ensemble de la classe politique et même repris dans d'autres instances gouvernementales, a été ralenti par le lobbying de certaines grosses sociétés éditrices de logiciels. Actuellement le projet reste en suspend et attend l'avis du conseil d'état.

Il ne faut pas pour autant croire que l'évolution des logiciels libres dans le domaine public est bloqué. Un cadre législatif n'est pas nécessaire à une administration pour s'orienter vers

⁽¹⁾ ATICA – Agence pour les Technologies de l'Information et de la Communication dans l'Administration

les logiciels libres. A l'inverse une décision législative imposerait à toutes les administrations concernées d'utiliser ce type de logiciel.

C'est ce qu'a tenté de faire le Pérou par l'intermédiaire du Congrès de la République. Celui-ci a proposé un projet de loi (Avril 2002) visant à imposer l'utilisation de logiciels libres répondant, comme le dit le DR. EDGAR DAVID VILLANUEVA NUNEZ Membre du Congrès de la République du Pérou, à une logique de gestion administrative.

« Les principes élémentaires qui inspirent le Projet sont liés aux garanties fondamentales d'un État démocratique de droit, telles que :

- le libre accès du citoyen à l'information publique ;
- la pérennité des données publiques ;
- la sécurité de l'État et des citoyens.

Pour être acceptable par l'État, il ne suffit pas que le logiciel soit techniquement suffisant pour mener à bien une tâche, mais il faut en plus que ses conditions contractuelles satisfassent une série de pré-requis en matière de licence, sans lesquelles l'État ne peut pas garantir au citoyen le traitement adéquat de ses données, veiller à leur intégrité, leur confidentialité et leur accessibilité au cours du temps, car ce sont des aspects critiques de son usage normal ». [Voir annexe 2]

La réponse, proposée en annexe 2, du DR. Edgar David⁽²⁾ aux remarques acerbes du représentant de Microsoft est, à cet égard, un exemple de concision, de clarté et de justesse dans les arguments.

De son côté, le Royaume-Uni n'est en reste, L'Office gouvernemental du commerce (OGC) a publié un document sur la politique du Royaume-Uni en matière de logiciels libres, sous le titre :

- Open source software : use within UK government (<http://www.ogc.gov.uk/index.asp?id=2190&>)

Dans ce document, l'office témoigne de l'intérêt gouvernemental en matière de logiciels libres. Le point de départ était le « plan d'action e-Europe » de la Commission européenne du 14 juin 2000, qui incitait les États à promouvoir notamment l'utilisation des logiciels libres dans le secteur public. Une des réponses britanniques fut de constituer le programme e-GIF, à savoir un cadre d'interopérabilité aux standards prédéfinis, imposé à l'ensemble du secteur public.

Ce document d'orientation est une nouvelle étape, qui définit la politique d'utilisation des logiciels libres du Royaume-Uni, au sein de l'administration centrale, des agences, des gouvernements locaux, des partenaires volontaires, du système national de santé.

Ce choix est justifié par différents éléments :

- le contexte économique ;
- l'exigence d'interopérabilité (à toujours prendre en compte) ;
- la volonté de réduire à la fois les coûts et les risques pour les systèmes gouvernementaux, en procurant notamment plus de flexibilité dans leurs développements, leurs perfectionnements et leurs intégrations ;
- supprimer la dépendance vis-à-vis de fournisseurs spécifiques ;
- les impératifs de sécurité.

⁽²⁾ Accessible à l'adresse <http://gnuwin.epfl.ch/articles/fr/reponseperou/rescon-fr.html> et disponible dans les annexes

L'O.G.C. mettra à jour ses directives sur les logiciels. Il démontrera les forces et les faiblesses du logiciel libre en fonction des secteurs. Il conseillera les différents acteurs dans la manière d'évaluer les mérites du logiciel libre par rapport aux solutions propriétaires.

Le plan d'action : « **eEurope 2005: une société de l'information pour tous** » propose plusieurs actions dont une en rapport avec les logiciels libres.

« Dans le cadre du plan d'action eEurope 2002, les États membres ont accepté de mettre tous les services de base en ligne pour la fin de 2002. Beaucoup a déjà été fait dans ce domaine, mais l'interactivité de nombreux services reste limitée. La Commission et la présidence belge ont organisé en novembre 2001 une conférence sur le «gouvernement électronique» afin de recenser et de diffuser des exemples de bonnes pratiques. Les ministres participants ont adopté une déclaration dans laquelle ils insistent sur la nécessité de développer l'échange des bonnes pratiques et de garantir l'inclusion et la sécurité dans la fourniture des services publics en ligne. Le développement d'un accès sûr et transparent aux services de gouvernement électronique dépend du déploiement et de l'utilisation réelle de moyens d'authentification électronique. L'action lancée en faveur de l'adoption de la carte à puce se poursuivra et pourrait valablement servir cet objectif en fournissant aussi un dispositif sécurisé à l'appui de la signature électronique.

Interopérabilité. Pour la fin 2003, la Commission adoptera, après concertation, un cadre en matière d'interopérabilité, afin de faciliter la fourniture de services paneuropéens de gouvernement électronique à la population et aux entreprises. Il abordera les questions relatives aux contenus informatifs et recommandera des mesures et des spécifications techniques afin d'unifier les systèmes d'information des administrations de l'ensemble de l'UE. Il sera fondé sur des normes ouvertes et encouragera l'utilisation de logiciels ouverts.

La recherche communautaire devrait aussi soutenir la normalisation afin de susciter une plus large utilisation des normes ouvertes et des logiciels ouverts ».

[Bruxelles, le 28.5.2002 COM(2002) 263 final, COMMUNICATION DE LA COMMISSION AU CONSEIL, AU PARLEMENT EUROPÉEN, AU COMITÉ ÉCONOMIQUE ET SOCIAL ET AU COMITÉ DES RÉGIONS, **eEurope 2005: une société de l'information pour tous**, P11-12, site http://europa.eu.int/information_society/eeurope/news_library/eeurope2005/index_en.htm]

Comme on peut le remarquer, l'Europe vise à favoriser et inciter l'utilisation des logiciels libres. Il est donc temps pour les acteurs public de progresser dans la connaissance de ce type de systèmes. La migration des systèmes actuels n'est pas encore obligatoire mais risque d'être imposée à terme.

L'utilisation d'outils gratuits est également envisagée pour les pays en voie de développement comme l'Afrique. A ce sujet un article parut dans la libre Belgique présentait des réflexions sur le sujet des réseaux wireless :

« L'objectif initial est de développer des réseaux locaux de communication libre et sans fil dans les pays d'Afrique, en partant de la récupération de matériel informatique et de l'utilisation de logiciels libres. Une solution concrète pour réduire la fracture numérique du Continent noir, régulièrement laissé pour compte tous points de vue confondus. »

[La libre Belgique du samedi 19 avril 2003 – page VI et V]

4.1. Les besoins

L'objectif premier, de la majorité des autorités qui optent pour la migration de leurs environnements vers un système à architecture ouverte, est de faire des économies. Même si cette vue de l'esprit est défendable, les arguments du projet sont plus orientés vers les nécessités du domaine public :

- *La pérennité de l'information* vient du fait que l'administration doit conserver les actes qu'elle produit pendant une longue période. Ces documents doivent toujours pouvoir être convertis. Leur accessibilité ne doit pas être liée à la survie économique ou au bon vouloir d'une entreprise particulière. Il est assez fréquent qu'un logiciel propriétaire propose une évolution de version de son produit qui impose à l'utilisateur de faire l'investissement de cette mise à jour.
- *L'accès à l'information par le citoyen* ne doit pas imposer l'usage d'un logiciel. L'utilisation d'un format libre permettrait de proposer en même temps que le document, le logiciel de lecture permettant d'utiliser celui-ci
- *L'utilisation adéquate des deniers de l'administration.* Les licences coûtent cher et surtout sont dépendantes du parc informatique de l'entité publique (ex : le ministère des finances avec ses 30.000 fonctionnaires). L'intérêt des logiciels libres tient au fait qu'ils ne sont pas soumis à une licence et que l'on peut ainsi utiliser les budgets alloués pour l'adaptation des logiciels aux besoins spécifiques de l'administration, la maintenance, l'accompagnement, la formation, etc... Ceux-ci sont réellement ouverts aux petites et moyennes entreprises de la région
- *La sécurité* du logiciel par l'utilisation de codes sources ouverts est un argument important qui favorise les initiatives actuelles. L'expertise de la communauté des développeurs et utilisateurs est également un gage de sécurité.

Ces nouvelles techniques induisent des droits mais également des devoirs. Il est donc important, lors d'un tel déploiement, d'étudier les risques liés à l'utilisation des logiciels libres.

- La diffusion publique de logiciels réalisés ou modifiés par l'administration nécessite de s'assurer que les droits des auteurs sont respectés, et de préciser les garanties qui sont fournies avec ces logiciels (qu'il s'agisse de logiciels libres ou propriétaires). La disponibilité du code source rend plus aisée la vérification d'éventuelles violations du droit d'auteur et demande donc une plus grande attention quant à la provenance du code source qui est incorporé dans le logiciel.

Ces nouvelles techniques impliquent un haut degré de capacité que l'administration située en fin de chaîne ne peut pas toujours mettre en place. Il lui est difficile de trouver pour elle des techniciens de haut niveau acceptant les salaires proposés par les administrations. Les gouvernements s'orientent donc d'avantage vers des entités centrales (europe, fédéral, ...) qui proposent des projets bien ciblés afin de les répercuter sur les administrations publiques comme les villes et communes.

C'est dans ce domaine que l'administration a beaucoup à attendre de l'utilisation de logiciels libres. La capitalisation des nombreux développements réalisés par et pour l'administration, permet de faire bénéficier rapidement l'ensemble des administrations d'un logiciel d'intérêt général développé pour une entité administrative donnée. Les administrations effectuent en général le même type de travail et suivent les mêmes directives législatives. Cette mutualisation des ressources, des connaissances et des coûts apporte une évolution souvent recherchée mais rarement atteinte au sein du domaine public.

L'Union des Villes et Communes réfléchi intensément à la possibilité de mutualiser les solutions à apporter aux problèmes communs des villes et communes. Le choix pris par cette agence s'oriente vers les logiciels libres, soit qui existent sur le marché soit développés en commun avec des partenaires privés ou publics.

Connaissant les lenteurs administratives il faudra certainement encore deux ou trois ans avant la réalisation concrète de ces volontés. Il apparaît toutefois que nous allons vers une amélioration certaine de l'utilisation standardisée des outils informatiques.

Analyse d'une migration

5. La gestion du changement

Plusieurs alternatives s'offrent à l'entreprise qui se pose la question de la migration.

- faire évoluer son système actuel vers une version plus récente des logiciels déjà utilisés ;
- migrer l'environnement actuel vers un système *Open Source* ;
- continuer avec le système actuel.

5.1. Objectifs et contraintes

Avant d'envisager la migration d'un environnement, une réflexion approfondie doit avoir lieu. Il faut définir les objectifs de la migration, poser les contraintes mais également effectuer un inventaire minutieux du matériel et des logiciels.

a. Objectifs

L'objectif de la migration est de fournir des fonctionnalités et des performances au moins égales à celles fournies par la configuration initiale, tout en apportant des services améliorés en terme de fiabilité, de sécurité et de disponibilité.

Un objectif plus pragmatique sera la réduction globale des coûts et une plus grande indépendance envers les éditeurs de logiciels.

b. Conditions

- Migrer tout le « BackOffice » ?

Pour répondre à cette question il faut analyser chacun des serveurs, définir les services qui tournent sur chacun d'eux et vérifier la possibilité de trouver des équivalents logiciels qui répondent aux fonctionnalités proposées.

Il peut s'avérer que le serveur soit tellement spécifique qu'il doive rester dans son état de départ. Pour cette raison, il faut vérifier la possibilité d'un environnement mixte qui intégrerait des serveurs ayant évolué et des serveurs conservant l'environnement de départ.

L'analyse des fonctions peut également remettre en question l'organisation des serveurs et des services installés.

Certains systèmes d'exploitation permettent de faire fonctionner un plus grand nombre de services sur un même serveur. Linux, étant un système multitâches préemptif, c'est-à-dire : capable de faire tourner simultanément plusieurs logiciels de manière totalement indépendante (*le "plantage" d'une application n'empêche pas l'ordinateur de poursuivre l'exécution des autres*) répond à ce critère.

- Migrer tout le parc ?

Lorsqu'on s'attaque aux stations clientes, l'analyse des fonctions ou logiciels utilisés est indispensable. Pour chaque logiciel, un remplaçant doit être comparé en

terme de fonctions proposées. Propose-t-il l'entièreté des fonctions du logiciel actuel, et dans le cas contraire, peut-il malgré tout être considéré comme une alternative acceptable ?

L'utilisation d'applications développées en interne peut également influencer la possibilité de migration. Sera-t-il possible, dans un laps de temps suffisamment court, de redévelopper l'application afin qu'elle fonctionne dans le nouvel environnement ?

Il faudra également poser la problématique des drivers nécessaires au fonctionnement hardware du matériel. Que ce soit sous Linux ou Windows, il est fréquemment très difficile de trouver le gestionnaire de périphérique adapté au nouveau système.

Par exemple, certains nouveaux matériels ne proposent plus les drivers pour NT4.

Une solution serait déjà, lors des nouvelles acquisitions, de choisir un matériel certifié Linux et/ou Windows 2000.

c. Contraintes

- Techniques

- Une migration "stoppable" à tout moment pour conserver un réseau hétérogène.

Il doit toujours être possible de revenir à la situation de départ si la migration ne se passe pas convenablement. Pour cette raison il est important de travailler en étapes.

- Migrer sur un environnement de test ;
- Intégrer des nouvelles solutions dans l'environnement de production.

- Pas de dégradation des performances

La migration doit apporter une amélioration et non une dégradation. Il faut donc évaluer l'amélioration des performances de chaque alternative.

- Pas de dégradation des fonctionnalités

Tant les administrateurs du réseau que les utilisateurs attendent un apport de fonctionnalités supplémentaires sans perdre les fonctions dont ils disposent.

Les administrateurs, souvent les premiers demandeurs d'une migration, souhaitent avoir des outils plus performants pour la surveillance du réseau.

- Humaines

Les contraintes humaines ne sont pas à négliger dans une migration. La raison qui fait qu'un projet n'aboutit pas, est plus souvent due à l'absence de prise en compte de la résistance au changement des utilisateurs, qu'à des impossibilités techniques.

La contrainte humaine est, bien évidemment, différente entre celui qui demande la migration et celui qui l'a subit.

Néanmoins, des contraintes générales existent.

- o La transparence pour les utilisateurs.

L'objectif de la migration est d'offrir une ergonomie au moins comparable à celle à laquelle les utilisateurs sont habitués, tout en fournissant des performances plus importantes. Pour assurer une migration en toute transparence dans l'efficacité du service il sera certainement nécessaire de prévoir des formations.

- o La continuité de service

Celle-ci ne peut jamais être interrompue, que ce soit par rapport à l'utilisateur lui-même, au patron de l'entreprise ou au client.

- o La motivation des utilisateurs

La résistance au changement des utilisateurs, surtout s'ils ne sont pas demandeurs, doit être étudiée de manière approfondie.

Il semble dès lors important de catégoriser le public cible selon :

- le niveau de connaissances informatiques ;
- le taux de résistance à l'évolution ;
- le niveau de représentativité ;
- le niveau hiérarchique au sein de l'organisation (et donc son pouvoir décisionnel).

La classification des utilisateurs permettra ainsi de définir un public cible sur lequel il sera possible de tester le nouvel environnement, et de définir la meilleure méthode pour les motiver.

A cette fin, l'utilisation de la *carotte* peut être employée. Il faut proposer à l'utilisateur une récompense à l'utilisation du nouveau système. Par exemple :

- l'utilisation du nouvel environnement lui permettra d'avoir un ordinateur plus performant.
- L'utilisateur qui décide d'éliminer définitivement l'ancienne configuration pour travailler avec la nouvelle bénéficiera d'une nouvelle imprimante laser.
- Etc

Cette façon de mettre à disposition un matériel plus performant permet d'aller plus vite dans la migration, et par conséquent de récupérer le coût des licences pour financer l'amélioration du matériel, qui était nécessaire.

Une autre catégorie d'utilisateurs à prendre en considération sera l'équipe informatique qui est chargée du support ou du développement d'applications bureautiques propriétaires ou de logiciels élaborés en interne.

Il faudra évaluer cette équipe et se poser les questions suivantes :

- L'équipe est elle prête à changer d'environnement, d'outils de développement ou de management ?
- Est-elle demandeuse de ce changement ?
- Quelles seront les incidences en terme d'emploi (nombre, qualifications, ...)?

5.2. Inventaire du matériel et des logiciels

Pour rester pragmatique et efficace il est essentiel, avant toute migration, que l'entreprise analyse ses services et matériels. Dans tous les cas il ne faudra pas hésiter à être sélectif dans les services qu'on souhaite migrer, voire à définir un plan progressif de migration des différentes applications. Il n'est pas nécessaire de créer un big bang irréaliste alors que de nombreuses applications peuvent être facilement migrées et rationalisées.

L'inventaire du matériel doit présenter les serveurs qui sont à disposition, leurs capacités (CPU, disques, mémoire, niveau de sécurité, ...) et les services qui fonctionnent sur ceux-ci. Le travail est important et doit mettre en exergue les fonctions de chacun.

Ce travail doit être effectué au niveau des serveurs mais également au niveau des stations de travail. Un bilan de l'utilisation des applications : formats des données, utilisation de toutes les fonctionnalités, possibilités de changement d'applications (méthodes de travail, appropriation par les utilisateurs) doit être réalisé. Une évaluation technique des performances du matériel sera également produite.

Ce travail est fastidieux et sera le facteur déterminant dans l'évaluation du temps nécessaire à la migration.

5.3. La démarche

La démarche devra s'articuler autour des différentes étapes clés de la migration

- L'analyse de l'existant, l'expression des besoins et des contraintes
Comme décrit dans ce chapitre
- La réalisation d'une maquette de distribution logicielle
Définition des logiciels qui peuvent remplacer ceux qui sont utilisés
- L'accompagnement du déploiement
Choix d'une société de service spécialisée dans le domaine afin d'aider à cette migration

➤ L'exploitation sur un site pilote

Test de la migration sur un site pilote afin d'être directement efficace dans l'environnement de production

➤ L'intégration

Intégration des nouvelles solutions dans l'environnement final

➤ La formation et le transfert de compétences

Formation des administrateurs du réseau. Première approche sur l'environnement de test. Formation de l'équipe de développement et du help desk.

➤ Le support et la maintenance de la solution.

Partenaire pour le support et la maintenance du nouvel environnement

➤ La motivation des utilisateurs

Définir un plan d'action pour faciliter les changements d'outils au niveau des utilisateurs finaux.

5.4. Le système d'exploitation « Linux »

Linux est d'abord un *système d'exploitation multi-plateformes* : PC, PowerPC, Sparc, Mac, matériels nomades (ordinateurs portables) et systèmes embarqués (pilotes d'ascenseurs, bornes multimedia).

Quelques caractéristiques de Linux :

- comme tous les Unix, c'est un *système multi-utilisateurs*, qui permet à plusieurs personnes de travailler en même temps sur la même machine. Chacun bénéficie de droits d'écriture et de lecture des fichiers qui lui sont propres. C'est également un *système multi-tâches*, qui offre la possibilité à plusieurs programmes de se partager les ressources du ou des processeurs d'une même machine ;
- c'est un *système léger* : par exemple, il peut fonctionner sur un PC 486, avec 8 Mo de mémoire vive, sans disque dur et avec comme applications, un éditeur de textes et un navigateur web ;
- il supporte plusieurs systèmes de fichiers (FAT 16, FAT 32, NTFS). Ce qui lui permet de gérer des partitions d'autres systèmes d'exploitation. Dès lors, sur le même disque dur, on peut accéder à des fichiers créés sous Windows à partir d'un poste de travail configuré en Linux ;
- il rend possible le *multi-amorçage* (choix, au démarrage de l'ordinateur, de fonctionner sous Linux, Windows ou ...) ;
- des émulateurs permettent d'exécuter les applications d'autres systèmes d'exploitation sans quitter GNU/Linux (autrement dit, ces trois dernières caractéristiques signifient que l'on peut installer GNU/Linux sur une machine tout en profitant des possibilités des systèmes d'exploitation antérieurs que l'on souhaite conserver) ;

- il assure des performances réseau importantes et le support de nombreux protocoles de communication [NFS, AFS, SMB (Windows), IPX, Appletalk (pour les Macs)] ;
- il offre une sécurité accrue ;
- des procédures d'installation simplifiées existent maintenant en grand nombre.

Linux gère également les écrans dans les résolutions acceptées par le matériel, grâce à la couche graphique XWindow. Il est possible d'avoir une machine Linux présentant un bureau et des fenêtres au " look " MacOS ou Windows.

Des applications libres tournent sous systèmes d'exploitation propriétaires. Inversement, des logiciels propriétaires fonctionnent sous GNU/Linux. Les deux mondes ne sont donc pas complètement cloisonnés.

5.5. Les distributions

Par "distribution", on entend un ensemble de logiciels disponibles soit sur CD-Rom (l'approche la plus répandue pour ceux qui ont décidé de faire le pas du libre) soit sur Internet via un serveur ftp. Sous cette forme, les logiciels s'installent et se désinstallent facilement. Ils ont été testés, sont compatibles entre eux et surtout, ils ne désorganisent pas le disque dur.

1 Le contenu d'une distribution GNU/Linux

- le système d'exploitation GNU/Linux lui-même,
- de nombreux logiciels libres,
- une organisation standard du disque dur,
- un système de mise à jour, via CD-Rom , ou plus couramment via ftp,
- une bonne gestion des dépendances entre logiciels (et bibliothèques) permettant des installations/désinstallations sans surprise,
- une procédure d'installation simplifiée,
- des distributions adaptées aux besoins spécifiés : serveur, machine bureautique, ordinateur pour les écoles ou pour la recherche scientifique, machine orientée multimédia.

Certaines de ces distributions s'installent au-dessus de Windows, c'est-à-dire qu'elles ne partent pas d'un disque dur vierge (ce qui est toujours possible) et ne demandent pas de partager le disque en deux ou plusieurs partitions (Windows d'un côté et Linux de l'autre).

Toutes les versions de Linux ont une base commune définie et répertoriée : conventions de déploiement des arborescences (mêmes fichiers aux mêmes endroits) et généralisation des logiciels de configuration.

2 Les fournisseurs des distributions

Des organisations (certaines étant des entreprises maintenant cotées en bourse), RedHat, Debian, S.u.S.E., Caldera, Mandrake, évaluent l'ensemble des logiciels libres disponibles et des logiciels commerciaux tournant sur plateforme libre. Elles les testent, les réunissent par paquetages (les composants logiciels pour la gestion du réseau, pour la messagerie, pour les gestionnaires de fenêtres) et les assemblent dans des distributions disponibles par CD-Rom ou ftp. Il existe également des universités, des centres de recherche, des établissements

scolaires qui créent et diffusent des distributions adaptées à leurs besoins, souvent à partir de distributions existantes.

La plupart des sociétés éditrices de distributions se rémunèrent sur les documentations fournies (principalement quand la distribution est achetée en librairie) et sur les services associés (télémaintenance).

3 Le choix d'une distribution

Des spécialistes, capables de lire le code source d'un logiciel et de comprendre les caractéristiques mouvantes d'une distribution, mettent leur savoir à la disposition de tous. On trouve ainsi des sites web qui comparent les différentes distributions. Si l'on n'est pas un professionnel de l'informatique, rien ne remplace les conseils d'un prestataire de services.

5.6. Les alternatives logicielles

1 Les services sous linux

Distribution :	Debian GNU/Linux, Linux-Mandrake, SuSE, OpenLinux (Caldera), RedHat, Corel Linux, Slackware
Serveur de fichiers :	Samba, NFS
Serveur Web :	Apache, Zope
Serveur DNS :	Bind
Serveur de messagerie :	Exim, Qmail, Sendmail
Gestionnaire de mailing List :	GNU-mailman, sympath, majordomo
Langage de scripts CGI :	Perl, PHP, Python
SGBD :	PostgreSQL, MySQL, SAPDB
Authentification distribuée :	OpenLDAP, NIS
Serveur proxy-cache :	Squid
Firewall :	Ipchains/iptables (attention ceci n'est pas un firewall complet), IPCOP, SMOOTHWALL
Réseau Privé Virtuel (VPN) :	FreeS/WAN
Clustering :	Beowulf, Mosix
Liaison sécurisée :	SSL/SSH

2 La bureautique sous Linux

Les problèmes

- Récupérer ses documents dans un format standard
- Garder la mise en page

Solution bureautique libre

Suites bureautiques :	(StarOffice/OpenOffice, Koffice)
Editeur de texte :	Lyx, Siag Xedplus, Cooledit, AbiWord, Applix Words

Tableur :	Gnumeric, KSpread
Gestion des graphiques :	GnuPlot, Screenshots
Animation des documents :	Siag Egon Animator

Utilitaires annexes

Visualiseur de documents :	Ghostview, Siag gvu, xpdf
Browser :	KDE file Manager, Siag Xfiler
mail :	"Gnome-Pim", Kmail, Mutt, Pine
Calendrier :	webcal, ical, calendar
Graphisme 2D :	Gimp, Sketch, Eeyes, Eye of Gnome, Kview, KimageShop, Killustrator, Kchart, xfig, xpaint

3 Atelier logiciel

les ressources

Langages :	C, C++, Perl, Python, PHP3, Smalltalk, Java, Ada, Pascal, Delphi/kylix, Lisp, Prolog, WildLife, Eiffel, CAML
Compilateurs :	gcc, débbugger gdb/DDD...
Contrôle de version :	CVS (Bonsai)
Middleware :	Corba (Gnome-ORBit)

Les éditeurs

Editeurs de code :	XEmacs, Vim, Kredit, GEdit, JED, Jext
Suite de développement IDE :	KDevelop
Diagrammeur UML :	Dia
Interface graphique :	Gtk, Tcl/Tk
Interface base de donnée :	Gnome-DB, KdataBase

5.7. Quelques conseils pour bien démarrer une migration

- *Bien comprendre la philosophie des logiciels libres.* Esprit de partage, incitation à copier les logiciels libres et à les redistribuer, culture du don, autant de concepts qui ne sont pas familiers dans l'univers de l'informatique.
- *Consacrer un peu de temps à la lecture de quelques textes fondateurs.* Une bonne entrée en matière : "Logiciels libres, Liberté, Égalité, Business" (<http://www.freepatents.org/liberty/>).
- *Ne pas y aller seul.* Profiter de l'expérience accumulée par les autres. De nombreuses associations de passionnés du logiciel libre, dont certains membres sont aussi des

professionnels de l'informatique, existent en Belgique et ailleurs et vous aide à faire les premiers pas (voir forum des ville et communes : <http://www.uvcw.be>).

- *Se tenir au courant constamment.* Dans le monde très mouvant de l'informatique en général, et des logiciels libres en particulier, il est nécessaire de prendre en charge sa veille technologique avec sérieux.
- *Avoir une connexion web.* Indispensable, pour se tenir au courant et pour télécharger les logiciels, et éventuellement participer à leur développement, ne serait-ce que pour signaler des bugs ou demander des fonctionnalités supplémentaires.
- *S'informer.* Le logiciel libre est parfois la cible des critiques. Il faut donc consulter un maximum de sources, prendre un certain recul et ne pas hésiter le cas échéant à aller à l'encontre d'idées reçues.
- *Ménager une transition en douceur.* Il n'est jamais bon de débiter par un projet d'envergure, même si l'on dispose de moyens importants. Il est préférable de commencer par une expérience ne touchant pas à des services vitaux. Il est donc indispensable d'y associer des utilisateurs représentatifs. On pourra s'inspirer d'une expérience ou faire appel à des ressources extérieures.
- *Former les utilisateurs et les équipes informatiques.*

Analyse d'un cas spécifique

6. Etude de migration

L'informatique de l'administration (clients et serveurs) est axée majoritairement sur le système d'exploitation Windows NT4. D'un point de vue technique, hormis certaines incompatibilités matérielles, rien n'oblige l'administration à procéder à une migration vers une solution plus récente (Windows 200x ou Linux).

Par contre, Microsoft a annoncé récemment la fin de tout support au système d'exploitation Windows NT4 Workstation et Server (un bug de sécurité a été découvert récemment dans NT4 pour lequel Microsoft ne livrera pas de correctif – la mesure préconisée est la mise à jour vers une version plus récente).

L'informatique étant un outil primordial au bon fonctionnement de l'administration, un processus d'analyse a été initié afin d'évaluer la faisabilité, les risques et les coûts inhérents à une migration vers une solution Linux.

Je vais donc au travers de ce document évaluer une telle migration et proposer un planning de migration.

6.1. **Etat des lieux (partie serveurs)**

Afin d'offrir des services à haute valeur ajoutée, la cellule informatique a mis en place au fil des années un nombre important d'outils informatiques :

Accès au réseau et aux ressources Windows

Une partie importante de cette infrastructure est dédiée à l'accès au réseau IP (dhcp), à l'authentification (PDC et BDC), aux partages des ressources (Serveur de fichiers), à la résolution de noms (WINS) et aux mises à jour des anti-virus sur le domaine « Windows » de l'administration utilisé par environ 700 personnes.

Accès à Internet et au mail

Une partie de l'infrastructure est dédiée à l'accès à Internet (routeurs), à l'utilisation du mail (Linux et PostOffice), Intranet et Extranet (IIS) et à la sécurité induite de par leur utilisation (Firwall-1 et MS Proxy).

Autres (Workflow, Gestion documentaire, GIS)

D'autres outils existent au sein de la ville. Ils sont, soit utilisés par l'ensemble de l'administration (workflow), soit de manière plus limitée par un service particulier (cartographie, bibliothèque et certains développements effectués en interne, applications légales).

Matériel utilisé

Majoritairement, des serveurs Compaq Proliant sont utilisés au niveau de l'infrastructure système. De plus un SAN a été mis en place afin de mettre à disposition une capacité disque (suffisante, centralisée et sécurisée) au niveau de certains serveurs.

6.2. Etat des lieux (partie clients)

A quelques rares exceptions près, les postes clients sont installés sous Windows NT4. Les applications utilisées au sein de l'administration sont principalement Excel et Word, ainsi qu'un certain nombre d'applications Web (ASP-SQL).

Actuellement, un nombre important de documents Word et Excel peuvent être consultés et modifiés à partir d'OpenOffice ou StarOffice (Exception faite des documents avec macros).

Concernant les applications Web, un outil comme Mozilla (moyennant quelques tests) peut remplacer avantageusement Internet Explorer.

En tout état de cause, une partie non négligeable de postes clients pourrait être migrée vers une solution Linux/Mozilla/OpenOffice. Pour étant, l'ensemble des ressources réseaux (mail, partage de fichier;...) resteraient disponibles.

Concernant les postes clients qui utilisent les applications légales ou des logiciels purement Windows, tels que Access et PowerPoint ou des développements internes en VB et WinDev, il n'y a pas de solutions purement Linux. Des émulations existent mais, soit elles ne sont pas satisfaisantes en termes de fonctionnalités (Wine), soit elles n'apportent pas de solution au problème de licence (VmWare).

En définitive, la Ville pourrait s'engager rapidement dans une phase d'évaluation et de tests en vue d'une migration des postes clients qui pourrait se faire dans le cadre d'une collaboration active entre les informaticiens et des utilisateurs volontaires (users group).

D'autre part, il est important de souligner que les coûts de licence de Microsoft Office constituent un poste non-négligeable au niveau informatique. Dans une optique de réduction des coûts (l'argent économisé pourrait servir à pré-financer une partie de la migration vers Linux) il est tout à fait envisageable de s'orienter vers une solution Office moins coûteuse voire gratuite (StarOffice et OpenOffice).

6.3. Migration vers Linux

1 Les agents du service informatique

Introduction

Pour permettre à une organisation de garder son indépendance informatique vis-à-vis du monde extérieur, elle doit disposer d'un maximum de ressources internes formées à suffisance.

Ingénieurs système

Les deux informaticiens qui ont la responsabilité des différents serveurs ont une connaissance approfondie de l'infrastructure et des services mis en place. Ce niveau de connaissance a été rendu possible par une pratique quotidienne, éventuellement par des formations et de l'auto-formation. L'orientation de cette connaissance s'est faite en fonction des choix technologiques opérés à la Ville.

Dans le cadre d'une migration vers Linux et afin de rendre le même niveau de service, il est fondamental que ces deux personnes puissent d'une part suivre une formation de qualité, et d'autre part disposer d'un aménagement de leur temps de travail qui leur permettra d'initier un cycle d'auto-formation.

Helpdesk

Le Helpdesk étant la première ligne de support, il est important que les informaticiens qui le constituent soient suffisamment formés pour répondre ou intervenir sur des sujets moins pointus mais qui demandent un niveau de réponse sans faille face aux attentes du client.

Développeurs

Les outils et langages de développements sont actuellement orientés Windows. Si la Ville entre dans un processus de migration progressive de son parc Windows vers Linux, la problématique du développement doit être posée assez rapidement. En effet, il y aura lieu d'envisager une ré-orientation des équipes vers des outils et des langages qui sont en corrélation avec la plate-forme choisie.

2 Les outils informatiques

Serveurs Web (Intranet, Extranet)

Les serveurs Web installés sont basés sur l'outil Microsoft IIS (Version 3 et 4). Au niveau du protocole (HTTP) rien ne s'oppose à une migration vers Linux/Apache. En effet, le protocole HTTP est un standard de fait édicté par la « W3C ». Toutefois, différents aspects constituent des différences, mais ne sont pas bloquants. Notamment :

- Au niveau de l'interface d'administration : elle fait partie intégrante du logiciel sous IIS. Sous Apache il y a lieu d'utiliser des outils annexes (par exemple : Webmin) qui ne sont pas aussi intégrés.
- Au niveau de l'authentification : l'intégration est plus simple avec IIS vis-à-vis d'un contrôleur de domaine.

Les technologies utilisées au niveau du développement de l'Extranet et de l'Intranet au sein de la Ville sont basées sur le couple ASP-SQLServer tous deux propriétés de Microsoft.

Au niveau du langage ASP, plusieurs possibilités existent :

- Ré-écriture complète du code dans un langage « Libre » : PHP.
- Si le langage utilisé pour développer les ASP est le VBScript : un portage propriétaire et payant existe chez SUN (utilisé sur les plates-formes Cobalt). Ce produit permet d'après SUN de développer en ASP/VBScript et de migrer l'existant. Info : <http://www.sun.com/software/chilisoft/index.html>
- Si le langage utilisé pour développer les ASP est le PerlScrip, un portage libre existe : <http://www.apache-asp.org>

Force est de constater que la solution parfaite n'existe pas. Soit une ré-orientation de l'équipe de développement et le re-développement de toutes les applications sera nécessaire, soit on s'appuyera sur une autre technologie propriétaire. La meilleure solution serait

éventuellement de geler l'évolution de l'existant et d'effectuer tout nouveau développement sur la base de logiciels libres.

Serveur de base de données (SQLServer)

La migration vers une technologie libre (telle que PostgreSQL ou MySQL) peut être envisagée moyennant une analyse approfondie de ses conséquences au niveau de la compatibilité des formats de données, des extensions SQL, de l'interopérabilité avec d'autres applications et des codes sources des applications développées en interne.

<http://www.mysql.com>

<http://www.postgresql.org>

Serveur de bases de données (Oracle)

Oracle est depuis plusieurs années entré dans le monde Linux. L'ensemble de ses produits est disponible sous Linux (RedHat et UnitedLinux). A ce niveau, on peut donc envisager deux solutions :

1. Installation d'Oracle sous Linux et transfert des données. Dans cette optique, il faudra éventuellement s'assurer le concours d'un expert Oracle pour la migration et interroger Oracle quant aux passerelles possibles au niveau des licences
2. Migration des données vers une solution libre.

Dans les deux cas de figures, il faudra garantir la pérennité des applications dépendantes d'Oracle.

Firewall & Anti-Virus Gateway

Firewall-1 de Checkpoint et Viruswall Interscan de Trendmicro sont actuellement configurés sur deux machines Windows NT4, dont le support de la part de Microsoft sera bientôt inexistant. Il y a donc lieu de s'interroger sur l'attitude que prendront d'une part les constructeurs de ces deux produits, et d'autre part, les intégrateurs. En tout état de cause, une version pour Linux existe pour ces deux produits et dispose des mêmes fonctionnalités.

En outre, Linux intègre par défaut (quelle que soit la distribution) un logiciel de Firewalling (iptables). Cette application dispose des mêmes fonctionnalités en terme de sécurité (Statique Source et Destination NAT, dynamique NAT, Statefull inspection) qu'un Firewall-1. Concernant le module de sécurité (VPN) plusieurs solutions existent sous Linux :

Logiciels VPN :

FreeS/WAN : <http://www.freeswan.org>

OpenVPN : <http://openvpn.sourceforge.net>

PoPToP : <http://www.poptop.org>

Distributions orientées Firewall/VPN :

Astaro Security Linux : <http://www.astaro.com>

Wolverine Firewall and VPN Server : <http://www.coyotelinux.com>

Proxy

La solution de Proxy/Cache actuellement utilisée à la ville est basée sur « MS Proxy ». L'accès à Internet est conditionné par une authentification (username/password) réussie via la base de données utilisateurs du PDC (SAM).

Plusieurs Logiciels Libres implémentent les fonctions de Proxy Cache et Reverse Proxy dont le plus complet est « Squid ».

Ce dernier intègre les fonctionnalités suivantes :

- Proxy Cache
- Reverse Proxy
- ACL's
- Authentification (un module spécifique a été développé pour l'authentification avec un PDC)

Infos:

<http://www.squid-cache.org>

PDC, BDC et Serveurs de Fichiers

Les fonctionnalités générales d'une infrastructure Windows sont tout à fait reproductibles sous Linux via le Logiciel Libre « SAMBA ». Une quantité croissante d'entreprises se penchent sur une solution Linux/Samba afin de remplacer leurs serveurs Windows ou d'implémenter une nouvelle infrastructure.

Malgré tout, on mettra en évidence certaines lacunes de Samba dans sa version stable :

- Réplication impossible de la base WINS entre plusieurs serveurs (Samba<->Samba ou Samba<-> Windows).
- Réplication impossible de la SAM entre plusieurs serveurs, le concept de BDC n'est donc pas possible sans l'ajout d'un protocole externe (LDAP ou NIS).
- La fonctionnalité de changement de password obligatoire après une certaine période n'est pas implémentée.

Néanmoins Samba dispose d'atouts et d'avantages certains :

- Licence gratuite.
- Le système hôte peut être Linux, réputé pour sa stabilité et sa fiabilité.
- Intégration de Samba avec LDAP (centralisation des données utilisateurs et groupes).
- Indépendance vis-à-vis de sociétés commerciales (correction des bugs rapide).

En ce qui concerne une éventuelle migration des serveurs NT4 de la Ville vers Linux/Samba plusieurs points bloquants ont été identifiés :

- Migration impossible des ACL (fichiers et répertoires).
- Politique de changement et de rétention de password impossible.

- Pilotes bêta pour les cartes SAN Qlogic limités à certaines distributions (RedHat).
- Outil de gestion des utilisateurs et des groupes inutilisable dans le cadre d'une solution SAMBA/LDAP.

Infos :

<http://www.samba.org>
<http://samba.idealx.org>
<http://www-1.ibm.com/servers/esdd/tutorials/samba>

Serveur Anti-virus (Innoculate)

Computer Associate a porté certains de ces outils sous Linux (ArcServe), mais aucune annonce n'a été faite quant au portage d' « Innoculate » sous Linux.

Suite à l'analyse du marché des antivirus, aucune solution similaire à « Innoculate » et fonctionnant sous Linux n'a pu être identifiée. Par contre des solutions antivirus sous Linux existent pour effectuer du Realtime Scanning (ServerProtect For Linux de TrendMicro).

Serveur de Backup (ArcServer 6.61)

Les parties serveur et agent d'ArcServe existent sous Linux. Toutefois, une analyse détaillée des fonctionnalités de la version Linux devra être effectuée afin d'identifier le plus précisément possible ce qui est supporté et ce qui ne l'est pas (systèmes de fichiers, ACL, permissions, distributions). La version actuelle d'ArcServe sous Linux ne supporte qu'un nombre limité de fonctionnalités « Linux ». La nouvelle version devrait sortir rapidement.

Serveurs de mail

Dans le cadre du projet de mise à disposition d'une adresse mail aux citoyens de la Ville, cette dernière a pu évaluer positivement l'utilisation de Linux dans un environnement de production.

L'utilisation interne du mail de plus en plus importante requiert, à court terme, l'achat de nouvelles licences.

Afin de solutionner cette problématique, une solution Linux peut être rapidement mise en place et offrir aux utilisateurs un même niveau de services.

Plusieurs outils mail existent sous Linux (Exim, Sendmail, Qmail, Postfix). D'après l'analyse de l'existant, la solution préconisée est la suivante :

- Linux Redhat (certifié compaq) ou Debian
- Qmail-LDAP (smtp), F-Prot (antivirus)
- Courier-IMAP (imap4 et pop3) phpQLAdmin et GoSa (outils d'administration)

Cette approche n'engendre aucune limitation au niveau du nombre de comptes mail.

Infos :

<http://www.qmail.org>
<http://www.redhat.com>
<http://www.debian.org>
<http://www.inter7.com/courierimap>

DHCP

DHCP est un protocole standard dont une implémentation existe sous Linux. La différence majeure se situe au niveau de l'interface d'administration. En effet, l'interface très complète de Windows n'a pas d'équivalent sous Linux.

Infos :

<http://www.isc.org>

<http://www.dhcp.org>

<http://www.webmin.com>

6.4. Inventaire matériel

Pour des raisons évidentes de clarté et de confidentialité, la description ci-dessous reprend une synthèse des matériels qui composent le réseau de l'administration. Un document complet généré à partir d'une application spécifique analysant le réseau (Inventory) et décrivant de manière précise le parc informatique existe, mais n'est pas joint à ce dossier.

1 Stations

Le parc informatique de l'administration est composé de plus ou moins quatre types de stations.

1. *Les Pentium de première génération (PI)*

- Processeur entre 90 et 200 mhz
- Mémoire de 32 Mb Edo pour les anciens ou 64 Mb de SDRam pour les plus récents
- Des disques durs de 1 à 3,5 Giga à 5400 tours

2. *Les Pentium de seconde génération (PII)*

- Processeur entre 233 et 450 mhz
- Mémoire de 64 Mb de SDRam
- Des disques durs de 5 à 9 Giga à 5400 tours

3. *Les Pentium de troisième génération (PIII)*

- Processeur entre 300 et 1000 mhz (peut être de type celeron)
- Mémoire de 64 Mb de SDRam jusqu'à 256 Mb pour les plus puissants
- Des disques durs de 10 Giga à 5400 tours à 15 Giga à 7200 tours

4. *Les Pentium de quatrième génération (PIV)*

- Processeur entre 1,5 et 2 Ghz
- Mémoire de 128 Mb à 256 Mb de SDRam
- Des disques durs de 20 à 30 Giga à 7200 tours

Le nombre de stations de ces 4 types se répartit comme suit :

- type 1 : ±150
- type 2 : ±170
- type 3 : ±240
- type 4 : ± 20

2 Imprimantes

Le parc d'imprimantes est essentiellement de marque HP et comprend des imprimantes :

Noir & Blanc de type HP de la série 500	50 pièces
Couleurs de type HP de la série 600, 700, 800 et 900	150 pièces
Laser de type HP de la série 4, 5, 6, 1000, 2000, 4000 et 8150	80 pièces

3 Serveurs

Par ordre d'ancienneté :

N°	Type	Ram	HDD	Sécurité
371	Pentium Pro 180 – Mono processor	192 Mb	2x4,5Gb	Mirroring
376	Pentium Pro 200 – Bi processor	256 Mb	5x9Gb	Raid 5
377	Pentium Pro 200 – Bi processor	256 Mb	5x4,5Gb	Raid 5
496	Pentium Pro 200 – Mono processor	480 Mb	3x4,5Gb	Raid 5
497	Pentium Pro 200– Mono processor	256 Mb	3x4,5Gb	Raid 5
532	Pentium 233 – Mono Processor	256 Mb	1x15Gb	Backup
578	Pentium II 450 – Mono processor	256 Mb	2x9Gb	Mirroring
579	Pentium II 450 – Mono processor	256 Mb	2x9Gb	Mirroring
1033	Pentium II 450 – Mono processor	256 Mb	1x20 Gb	Backup
1044	Pentium II 450 – Mono processor	256 Mb	3x9 Gb	Raid 5
1045	Pentium II 450 – Bi processor	1 Gb	4x9 Gb	Raid 5
1046	Pentium II 450 – Bi processor	2 Gb	4x9 Gb 2x18Gb	Raid 5 Mirroring
1047	Pentium II 450 – Mono processor	256 Mb	2x9 Gb	Mirroring
1144	Pentium III 750 – Mono processor	128 Mb	1x20 Gb	Backup
1208	Pentium III 800 – Mono processor	256 Mb	2x36Gb	Mirroring
1236	Pentium III 800 – Mono processor	512 Mb	4x18Gb	Raid 5
1301	Pentium III 1400 – Bi processor	1 Gb	2x36Gb	Mirroring
Admipro	Pentium III 750 – Mono processor	256 Mb	2x9Gb	Mirroring

7. Analyse des coûts comparés et plan de migration

Le tableau repris ci-dessous reprend la comparaison financière des deux migrations possibles : migration vers MS Windows 2000 et migration vers Linux et les logiciels *Open Source*. L'analyse se base sur la méthode du Total Cost of Ownership (TCO)⁽¹⁾.

7.1. Les coûts

POSTE PRINCIPAL	LOGICIELS	PRIX UNITAIRE	ANNEE 1			ANNEE 2			ANNEE 3		
			Qté	Linux	Windows	Qté	Linux	Windows	Qté	Linux	Windows
SYTEME D'EXPLOITATION											
	WINDOWS 2000 ADV. SERVER + 25 Clients	1.999 ?	2		3.998 ?	0		- ?	0		- ?
	WINDOWS 2000 SERVER (Licence + 20 Clients)	449 ?	1		449 ?	2		898 ?	0		- ?
	CLIENT WINDOWS 2000 SERVER	25 ?	500		12.500 ?	0		- ?	0		- ?
	LINUX RED HAT Distribution	150 ?	1	150 ?		0	- ?		0	- ?	
	WINDOWS 2000 WORKSTATION	163 ?	0		- ?	0		- ?	300		48.900 ?
	SYSTEME D'EXPLOITATION TOTAL			150 ?	16.947 ?		- ?	898 ?		- ?	48.900 ?
BASE DE DONNEES											
	MS SQL SERVER 2000 ENTR + 10 LIC Client	2.329 ?	0		- ?	2		4.658 ?	0		- ?
	MYSQL DATABASE SERVER	220 ?	1	220 ?		0	- ?		0	- ?	
	BASE DE DONNEES TOTAL			220 ?	- ?		- ?	4.658 ?		- ?	- ?
SECURITE											
	ANTIVIRUS POUR WINDOWS	- ?	0		- ?	0		- ?	0		- ?
	ANTIVIRUS POUR LINUX	572 ?	1	572 ?		0	- ?		0	- ?	
	FIREWALL - Maintenance Annuelle	7.700 ?	1		7.700 ?	1		7.700 ?	0		- ?
	FIREWALL - IPTABLE	- ?	0	- ?		0	- ?		0	- ?	
	ARCSERVE 9.0 (Backup)	590 ?	0		- ?	0		- ?	0		- ?
	DISK ARCHIVE FOR LINUX	- ?	0	- ?		1	- ?		0	- ?	
	ARCSERVE FOR LINUX	600 ?	0	- ?		1	600 ?		0	- ?	
	SECURITE TOTAL			572 ?	7.700 ?		600 ?	7.700 ?		- ?	- ?
MAIL / INTERNET											
	MS EXCHANGE 2000 SERVER ENTR	7.250 ?			- ?	1		7.250 ?	0		- ?
	SENDMAIL / POSTFIX	- ?		- ?		1	- ?		0	- ?	
	MAIL / INTERNET TOTAL			- ?	- ?		- ?	7.250 ?		- ?	- ?

⁽¹⁾ TCO = Coûts du matériel et des licences + charges d'exploitations directes (personnel, sous-traitance, services divers, ...)

	MAIL / INTERNET TOTAL			- ?	- ?		- ?	7.250 ?		- ?	- ?
LOGICIELS METIERS											
	MS VISUAL STUDIO.NET + MSDN	1.227 ?	0		- ?	3		3.681 ?	0		- ?
	BORLAND KYLIX / DELPHI	2.229 ?	0	- ?		3	6.687 ?		0	- ?	
	PHOTOSHOP 7.0	229 ?	0		- ?	2		458 ?	0		- ?
	GIMP	- ?	0	- ?		2	- ?		0	- ?	
	MS OFFICE PRO 2000	309 ?	0		- ?	0		- ?	500		154.500 ?
	OPEN OFFICE	- ?	0	- ?		0	- ?		500	- ?	
	LOGICIELS METIERS TOTAL			- ?	- ?		6.687 ?	4.139 ?		- ?	154.500 ?
CHARGES D'EXPLOITATION											
	SALAIRES	- ?	0			0			0		
	PRESTATION W2000 - Migration	800 ?	3		2.400 ?	3		2.400 ?	0		- ?
	PRESTATION LINUX - Migration	735 ?	3	2.205 ?		3	2.205 ?		0	- ?	
	PRESTATION W2000 - Support	800 ?	3		2.400 ?	0		- ?	0		- ?
	PRESTATION LINUX - Support	735 ?	3	2.205 ?		0	- ?		0	- ?	
	FORMATIONS WINDOWS 2000 (Jours Homme)	300 ?	5		1.500 ?	5		1.500 ?	0		- ?
	FORMATIONS LINUX (Jours Homme)	375 ?	15	5.625 ?		10	3.750 ?		0	- ?	
	FORMATIONS OFFICE 2000 (Jours Homme)	300 ?	0		- ?	0		- ?	100		30.000 ?
	FORMATIONS OPEN OFFICE (Jours Homme)	375 ?	0	- ?		0	- ?		400	150.000 ?	
	CHARGES D'EXPLOITATION TOTAL			10.035 ?	6.300 ?		5.955 ?	3.900 ?		150.000 ?	30.000 ?
	TOTAL GENERAL			10.977 ?	30.947 ?		13.242 ?	28.545 ?		150.000 ?	233.400 ?

Les salaires des agents intervenant dans les charges d'exploitation ne sont pas significatifs pour les raisons suivantes:

- Les agents sont considérés suffisamment compétents pour s'adapter à un nouvel environnement ;
- Le salaire des agents suit un barème qui n'est en rien influencé par le niveau de compétences.

Pour ces raisons, le montant des salaires n'est pas pris en compte dans le tableau (égale à zéro). Il est néanmoins repris dans l'analyse de coûts pour marquer son importance. Il faut toutefois faire remarquer que les exigences, en terme de compétences, d'un gestionnaire de réseau linux sont plus importantes et donc doivent avoir une influence sur la charge salariale.

Un aspect important du coût des licences Windows 2000 Server ou SQL Server réside dans la nécessité d'acquérir également des licences *client* en nombre égal au nombre de postes clients qui communiquent avec le serveur. Ceci grève fortement les montants liés à l'environnement propriétaire.

Le choix d'un environnement *Open Source* apporte, au sein de la distribution, des outils complémentaires qui ne sont pas dénués d'intérêt.

Par exemple :

- serveur Web,
- serveur Proxy,
- serveur DNS,
- serveur DHCP
- Firewall,
- serveur Mail,
- outils de gestion,
- ...

Il est toutefois hasardeux d'établir un plan de migration sur trois ans. En effet, les éléments bloquants et les outils disponibles risquent d'évoluer. Néanmoins, pour des raisons de planification budgétaire, nous garderons cette vision sur trois années, tout en considérant qu'une réévaluation du projet devra être faite tous les six mois.

7.2. Le planning

La 1^{er} Année

La première phase consistera en la formation et l'auto-formation du personnel qui sera amené à intervenir sur les plates-formes Linux.

Il conviendra de mettre en place un environnement de tests sur lequel les administrateurs pourront s'essayer aux nouvelles technologies.

L'étape suivante implémentera un serveur LDAP souvent utilisé dans la gestion des environnements Linux. Ce serveur sera la première brique de la migration du serveur Mail et DNS.

Le choix du serveur mail comme premier serveur migré est influencé par la nécessité de créer des comptes mail supplémentaires. Par ailleurs ce dernier est certainement le serveur le plus facile à faire migrer.

Si l'on se réfère au tableau des coûts, une migration Linux nécessite l'acquisition de la distribution RedHat (certifié Compaq) alors qu'une évolution du système vers Windows 2000 exigerait l'achat de deux nouvelles licences du système d'exploitation et des 500 licences clients qui y sont liées.

La technologie de Windows 2000, basée sur les actives directory, nécessitera trois jours de support à la migration, comme pour le passage à Linux.

Du côté humain, la formation des agents responsables du réseau semble plutôt bien perçue. L'apport d'une nouvelle technologie où chacun démarre au même niveau de compétences propose un challenge que les agents souhaitent relever.

En conclusion, le choix de Linux apportera non seulement un gain financier mais également une performance équivalente, tout en proposant des outils d'administration plus nombreux.

Pour appuyer ces propos, je reprendrais également les écrits de Windows & .Net e-news, revue spécialisée concernant les logiciels Microsoft et qui montrent que l'alternative Linux (basée sur Unix) n'est pas un leurre puisqu'il faudrait, d'après cet article, une nouvelle version de Windows pour arriver aux fonctionnalités proposées par Linux.

« Windows Server 2003, la nouvelle version du système d'exploitation serveur développée par Microsoft assume les rôles de serveurs d'applications et d'infrastructure réseaux. Déclinée en version 64 bits (Enterprise Edition et Datacenter Edition), elle constitue une alternative aux multiples environnements Unix en matière de serveurs d'applications stratégiques, fournissant aux entreprises internationales ou multi filiales, flexibilité, hautes performances et réponse aux problématiques de consolidation .. » [Windows & .Net e-news, la newsletter électronique hebdomadaire des Professionnels de Windows NT / 2000 et .Net - N°46 - 09 Avril 2003].

La 2ème Année

Au cours de la seconde année, il sera important d'analyser la politique future de l'équipe de développement. Il conviendra de préciser les nouveaux langages qu'utiliseront les programmeurs pour développer les nouvelles applications ou redévelopper les anciennes.

Les langages ASP et Visual-Basic, ne pourront plus être utilisés et le choix d'un langage multi-plateformes sera assurément indiqué (PHP, Java, ...).

D'autres langages propriétaires sont envisageables, tel que Kylix de chez Borland, qui propose une interface plus proche de celles que les développeurs utilisent habituellement.

Dans tous les cas, le changement d'outil engendrera un besoin considérable de formations.

Cette évolution ira de pair avec le remplacement du gestionnaire de base de données, qui sous linux pourra être MySQL ou Postgres SQL. A l'inverse, sous Windows 2000, il est également nécessaire de mettre à jour la base de données pour répondre aux exigences de sécurité et d'accessibilité. L'investissement financier sera alors bien plus important.

Dans cette solution, le coût du remplacement du système d'exploitation des serveurs est également à prendre en compte.

Cette seconde année est importante pour stabiliser l'environnement et définir la politique des années suivantes. Nous évaluerons l'impact sur le fonctionnement général après migration, finaliserons les besoins en formation pour rendre définitivement autonome l'équipe réseau, et formerons l'équipe support et développement aux nouvelles technologies et outils disponibles.

La 3ème Année

La migration ainsi stabilisée permettra de migrer le parc des stations clientes afin de réduire de manière drastique les coûts de licences.

La reproductivité des logiciels *Open Source* et libres de coûts de licences, entre fortement en compte dans l'analyse des investissements nécessaires à la migration. L'achat d'une licence (s'il est nécessaire de l'acheter) permet de reproduire l'installation de manière illimitée mais également de télécharger gratuitement sur le net la dernière version.

De plus, la forte offre de logiciels libres proposée sur le net permettra, à très court terme, de répondre aux exigences les plus diverses des utilisateurs.

Les besoins en formation sont clairement plus important pour une migration vers les logiciels libres que pour une simple évolution de versions des logiciels Microsoft.

Des formations sont toutefois demandées par les utilisateurs afin de maîtriser correctement les outils actuels. Il semble donc séduisant d'utiliser le besoin de formation, plutôt pour un nouveau logiciel dont le coût d'acquisition est nul que pour un logiciel où les dépenses en licences sont exagérées.

Néanmoins, la résistance au changement ne manquera pas d'apparaître chez les utilisateurs finaux. Il faudra donc faire une évolution en douceur en laissant, par exemple pendant une période pas trop longue, cohabiter l'ancien et le nouveau système (Windows, Office <-> Linux, Open office).

Conclusions

Le tableau des coûts met en évidence deux possibilités de choix :

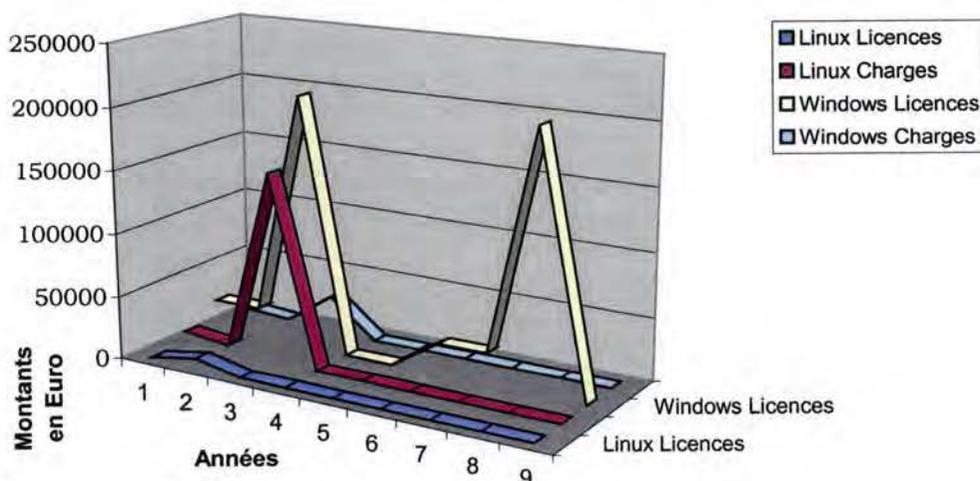
- Un investissement visant à payer les licences ;
- Un investissement visant à former et motiver les utilisateurs.

Même si le choix de la migration de doit pas être seulement influencé par l'aspect financier, il faut bien admettre que la différence est significative.

Il faudra donc également considérer l'aspect technique de ce type de migration. Le nouveau système devra proposer des avantages pour les utilisateurs (nouveaux outils, rapidité, nouvelles fonctionnalités, etc...).

Il faudra évaluer la perte de productivité due aux absences liées au temps de formation, de même que la difficultés d'adaptation aux nouveaux outils. Il faudra ensuite comparer cette perte aux améliorations que le nouveau système propose. Enfin, on mesurera l'impact de la baisse de productivité sur le fonctionnement de l'administration.

Evaluation des dépenses



L'évaluation des dépenses sur une période plus longue montre que les systèmes de licences mis en place par Microsoft, font que le droit d'usage d'un logiciel n'est plus autorisé à vie, mais sur une période de cinq ans. Cette contrainte engendre une redondance cyclique des dépenses.

A l'opposé, la charge d'exploitation d'une migration Linux n'est pas reproductible dans les années suivantes.

La migration de l'environnement sera vraisemblablement mixte, et ne comprendra que la migration des clients n'utilisant que les logiciels de la suite Office. Les autres postes migreront progressivement avec le développement en interne ou en externe, de nouvelles applications adaptées à Linux.

8. Conclusions

L'informatique est un perpétuel apprentissage. Il est impossible de connaître toutes les matières et toutes les nouveautés.

Malgré tout, le monde des logiciels libres devient incontestablement un domaine que tout informaticien doit appréhender un jour. L'importance des logiciels libres est autant technique que philosophique. Il me paraît important de remettre dans nos esprits les vrais valeurs du travail de création, du dépassement de soi et d'une saine compétition.

Ce travail m'a permis de mettre en exergue l'importance d'une réflexion sur la migration des environnements dans le domaine public.

A cet égard, plusieurs organismes centralisateurs au sein de ce domaine, pensent qu'il est temps de réfléchir à l'avenir des systèmes informatiques des administrations.

Sans être précurseur, l'étude proposée par ce travail a permis de réveiller les mentalités de l'administration dans laquelle j'exerce mon métier. La question d'une migration de l'environnement vers le monde Linux est à l'ordre du jour. Maintenant que le premier pas est fait, est-il logique de continuer ?

La mise en pratique de la méthode proposée dans ce travail va être vérifiée sur le terrain au cours des trois ou quatre prochaines années. Il sera alors intéressant de relever les imperfections du plan, et de proposer une méthode améliorée profitant des remarques soulevées par cette migration. La philosophie du monde libre m'incite également à mettre à disposition de la communauté l'expérience acquise.

J'espère que les idées présentées et développées dans ce travail profiteront aux futures administrations désireuses d'avoir la même démarche envers les logiciels libres.

Malgré tout, la survie des logiciels libres dépend encore des grands éditeurs et de leur puissance financière. Les lobby informatiques ont déjà fait plier le monde politique belge, peut-il en être de même avec le reste du monde.

Je ne suis pas loin de le penser ! Dans l'annexe 1, je reprends des informations sur la proposition de Microsoft pour notre avenir informatique. Il faut prendre ces informations comme un signal d'alarme face à ce que les grandes puissances financières veulent faire de notre planète.

Annexes

Annexe 1

La Solution que propose Microsoft et Intel

TCPA, qui signifie « Alliance pour une Informatique de Confiance » (Trusted Computing Platform Alliance en anglais), est un projet développé par Intel.

« Une nouvelle plate-forme informatique pour le 21^{ème} siècle qui améliorera la confiance dans le monde PC », tel est l'objectif d'Intel.

Palladium est un logiciel que Microsoft déclare vouloir incorporer dans les futures versions de Windows. Il s'installera sur des machines qui seront certifiées TCPA.

Palladium s'intéresse à **l'authentification de la machine** et/ou d'une pile logicielle s'exécutant sur la machine.

1. Techniquement

Palladium offre un moyen d'isoler les applications afin d'éviter l'espionnage et la modification par d'autres logiciels. Il permet également de stocker les éléments secrets tout en garantissant l'accessibilité aux seules personnes autorisées.

Comme il est dit dans la documentation de Microsoft : « *Palladium permet l'application d'une stratégie qui peut être profitable à de nombreuses parties, la mise en oeuvre restant sous le contrôle des utilisateurs. Les utilisateurs peuvent se servir de tels systèmes pour maintenir la confidentialité ou le partage contrôlé de leurs documents ou même la collaboration en ligne avec leurs amis, collaborateurs ou collègues, ou pour contrôler des opérations sensibles effectuées sur leurs machines* ».

De quoi s'agit-il réellement ? Il s'agit de renforcer la sécurité des ordinateurs en permettant au système d'exploitation et aux applications de s'appuyer sur les propriétés des composants physiques. Ainsi, on parle d'une zone réservée dans le PC (disque dur, mémoire, processeur ?) pour faire tourner les logiciels de manière sécurisée. D'autre part, une clé unique, stockée de façon matérielle (numéro de série du processeur ?) permettrait à des programmes ou des données de ne pouvoir être exécutés ou lus que sur la machine de l'utilisateur *client* et pas sur une autre. Même décrypté, un fichier son ou vidéo soumis à des droits ne pourrait être exploité que sur l'ordinateur destinataire à l'origine. Enfin, cette clé pourrait servir au décodage de données cryptées, généralement protégées par un simple mot de passe, voire une clé privée stockée sur le disque dur et, donc, *facilement* piratable.

L'objectif de Palladium est bien sûr de renforcer le cryptage des données (il serait d'ailleurs livré avec un outil de cryptage universel) et, surtout, d'appliquer la gestion des droits (*Digital Right Management* ou DRM) à tout type de documents. Autorisation de lecture, copie, destruction automatique du fichier... Une fonction certainement plus proche des besoins d'Hollywood que de celle des utilisateurs. Palladium renforcerait également la gestion anti-spam.

Du côté de Palladium, si on promet aussi l'utilisation des techniques de chiffrement et de signature électronique, on promet d'abord le concept d'une autorité centrale certificatrice. Il s'agit d'instaurer une sorte « d'appellation d'origine contrôlée » des logiciels et d'autoriser, sur

des ordinateurs spécialement conçus, uniquement l'exécution des composants dûment certifiés. Palladium est présenté comme la solution ultime en matière de sécurité, mais c'est un trompe-l'œil assez grossier. D'une part Palladium n'est pas nécessaire pour venir à bout des quelques classes de problèmes qu'il prétend éradiquer, par exemple les "buffer overflows". Mais surtout, Palladium n'est pas suffisant pour garantir l'invulnérabilité, puisque de nombreux types d'attaques sont toujours possibles. En particulier les fameux « vers applicatifs » que subissent sans cesse les utilisateurs du logiciel Outlook. Palladium semble vous proposer d'échanger votre liberté contre une tranquille sécurité. On voit mal pourtant le peu d'assurance que gagneront les utilisateurs du point de vue de la sécurité informatique, mais on perçoit parfaitement les libertés sacrifiées.

Le problème peut aller plus loin que cela puisque le développeur risque de ne pouvoir tester ses propres modifications de code. En effet, la recompilation du source va entraîner une modification de la signature et le système risque alors de refuser l'exécution de ce logiciel tant qu'il n'est pas à nouveau certifié.

Les développeurs d'outils freeware et de logiciels libres n'auront pas toujours les moyens de payer une certification pour chaque nouvelle version. Quant aux logiciels du type encodeur MP3 ou logiciel de peer-to-peer, il y a de fortes chances qu'ils se voient interdire les portes de Palladium.

Microsoft répondra que le système peut travailler soit en mode sécurisé soit dans le mode qui régit les systèmes actuels. Par cette astuce vous pourrez toujours exécuter des anciennes applications sur la machine Palladium.

Quand est-il alors des serveurs Web qui tourneront sous Palladium sécurisé ? Pourra-t-on toujours y accéder ?

Oui, si votre machine fonctionne sous Palladium. Dans le cas contraire votre accès sera refusé comme étant à risque.

Cette méthode risque de pousser, petit à petit, la communauté Internet à utiliser des ordinateurs de type TCPA – Palladium, et par là même, d'imposer un nouveau standard de fait et donc un monopole.

2. Déontologiquement

Le projet ne précise pas, le niveau de compatibilité entre les applications actuelles et le futur système sécurisé. Sans parler des autres systèmes d'exploitation, Linux notamment.

Quant à la question de la protection de la vie privée, elle est purement et simplement ignorée. En effet, un tel système permettrait, par exemple, à un loueur de films en ligne de savoir précisément qui regarde quoi et quand.

Une fois le système largement adopté, et une fois le point de non retour passé du côté des utilisateurs, le ton du géant pourrait changer du tout au tout et ses belles promesses s'évanouir dans la nature. D'ailleurs, ne peut-on pas lire à la fin du document de synthèse de Microsoft que : « *Les informations contenues dans ce document représentent la vision actuelle de Microsoft Corp sur le sujet évoqué et à cette date de publication. Parce que Microsoft doit répondre aux évolutions du marché, elles ne doivent pas être interprétées comme un*

engagement de la part de Microsoft, et Microsoft ne peut pas garantir l'exactitude de l'une de ces informations après la date de publication »

Pourquoi Microsoft ne dit-il pas clairement qu'il espère marginaliser avec ce projet ses principaux concurrents, en l'occurrence les logiciels libres ? Mais encore, Palladium pourrait accentuer les dérives actuelles qui tendent à transformer votre ordinateur en mouchard. Les libertés individuelles, évidemment plus importantes que la sécurité informatique, sont également menacées.

Depuis des années, Bill Gates rêvait de trouver un moyen pour que les Chinois payent leurs logiciels : Palladium pourrait être la réponse à sa prière.

Et les logiciels libres dans tout cela ?

TCPA va mettre fin aux libertés défendues par la GPL (Licence Publique Générale) qui sont le fondement principal des logiciels libres.

Les entreprises qui voudront proposer une version GNU/Linux compatible avec TCPA devront obtenir un certificat du consortium TCPA. Dès lors, il faudra nettoyer le code des fonctionnalités qui ne correspondent pas aux exigences de ce certificat et démontrer à un laboratoire d'évaluation (certifié) que les attaques connues ne fonctionneront pas avec la version du logiciel proposé. Cette évaluation sera suffisamment coûteuse pour laisser à l'écart la communauté du logiciel libre, mais assez permissive pour que la plupart des éditeurs de logiciels aient une chance de faire valider leurs codes sources vérolés.

Bien que le logiciel modifié soit protégé par la GPL, et que le code source soit libre, il ne pourra utiliser toutes les fonctionnalités TCPA qu'à moins d'avoir un certificat spécifique à la puce Fritz de la machine. Ce qui risque d'être onéreux (sinon au début, du moins à terme).

La communauté du libre, en créant la licence GPL, souhaitait rendre impossible l'appropriation d'un code à des fins commerciales. Ce qui pour un développeur de logiciel Open Source correspond au pilier principal de sa philosophie de l'informatique.

L'obligation de certification, que sous temps la TCPA, va décourager tout développeur de créer pour une communauté qui ne sera plus la seule à tirer avantage de son travail.

Le résultat sera un choix de logiciels libres plus faible, une évolution plus lente et surtout des correctifs moins fréquents.

A terme, la mort des logiciels libres.

Annexe 2

Lima, le 8 avril 2002.

Monsieur JUAN ALBERTO GONZÁLEZ
Directeur Général de Microsoft Pérou

Cher Monsieur.

Avant toute chose, je vous remercie de votre lettre du 25 mars 2002 dans laquelle vous exprimez la position officielle de Microsoft concernant le **Projet de Loi N° 1609, Logiciel Libre dans l'Administration Publique**, qui est inspirée sans aucun doute par le désir d'aider le Pérou à réussir à trouver sa place dans le contexte technologique global. Animé du même esprit et convaincu que nous trouverons les meilleures solutions par l'échange d'idées claires et ouvertes, je me permets de répondre, par la présente, aux commentaires contenus dans votre lettre.

Je reconnais que des opinions comme les vôtres constituent un apport significatif, mais elles m'eussent été plus utiles si, en plus de formuler des objections à caractère général (que nous analyserons en détail plus loin) vous aviez rassemblé des arguments solides sur les avantages que le logiciel propriétaire peut apporter à l'État péruvien et à ses citoyens en général, car cela aurait pu permettre un échange plus clair dans le respect des positions de chacun.

Dans le but de clarifier le débat, nous conviendrons que ce que vous appelez "logiciel à code source ouvert" est ce que le Projet définit comme "logiciel libre", sachant qu'il existe du logiciel dont le code source est distribué avec les programmes, mais qui n'est pas couvert par la définition établie dans le Projet ; et que ce que vous appelez "logiciel commercial" est ce que le Projet définit comme "propriétaire" ou "non libre", sachant qu'il existe du logiciel libre commercialisé sur le marché avec un prix comme tout autre bien ou service.

De même il est important de préciser que la proposition contenue dans le Projet auquel nous nous référons n'est pas directement en relation avec l'économie directe qui peut être réalisée par l'emploi de logiciel libre dans les institutions de l'État. Ceci est dans tous les cas, une valeur ajoutée marginale, mais en aucune manière l'objectif final du Projet. Les principes élémentaires qui inspirent le Projet sont liés aux garanties fondamentales d'un État démocratique de droit, telles que :

- Libre accès du citoyen à l'information publique ;
- Pérennité des données publiques ;
- Sécurité de l'État et des citoyens.

Pour garantir le libre accès des citoyens à l'information publique, il est indispensable que l'encodage des données ne soit pas lié à un fournisseur unique. L'utilisation de formats standards et ouverts permet de garantir ce libre accès, et d'obtenir, si nécessaire, la création de logiciel libre compatible.

Pour garantir la pérennité des données publiques, il est indispensable que l'utilisation et le maintien du logiciel ne dépendent pas de la bonne volonté des fournisseurs, ni

des conditions de monopole imposées par ceux-ci. Pour cela l'État a besoin de systèmes dont l'évolution puisse être garantie par la disponibilité du code source.

Pour garantir la sécurité de l'État ou sécurité nationale, il est indispensable de se baser sur des systèmes dépourvus d'éléments qui en permettent le contrôle à distance ou la transmission non désirée d'information à des tiers. Par conséquent, il faut des systèmes dont le code source est librement accessible au public pour permettre son examen par l'État lui-même, les citoyens, et un grand nombre d'experts indépendants dans le monde. Notre proposition apporte un plus de sécurité, puisque la connaissance du code source élimine le nombre croissant de programmes contenant potentiellement du *code espion*.

De cette façon, notre proposition renforce la sécurité de nos citoyens, à la fois en tant que détenteurs légitimes de l'information gérée par l'État, et en tant que consommateurs. Dans ce dernier cas, c'est en permettant l'apparition d'une offre étendue de logiciel libre dépourvu de potentiel *code espion* susceptible de mettre en péril la vie privée et les libertés individuelles.

En ce sens, le projet de loi se limite à établir les conditions dans lesquelles les organismes de l'État acquerront du logiciel dans le futur, à savoir, de façon compatible avec la garantie de ces principes fondamentaux.

A la lecture du projet il apparaîtra clairement qu'une fois approuvée :

- la loi n'interdit pas la production de logiciel propriétaire ;
- la loi n'interdit pas le commerce de logiciel propriétaire ;
- la loi ne dicte pas quel logiciel utiliser concrètement ;
- la loi ne dicte pas chez quel fournisseur acheter le logiciel ;
- la loi ne limite pas les termes de la licence qui couvre un produit logiciel.

Ce que le projet exprime clairement c'est que, pour être acceptable par l'État, il ne suffit pas que le logiciel soit techniquement suffisant pour mener à bien une tâche, mais il faut en plus que ses conditions contractuelles satisfassent une série de pré-requis en matière de licence, sans lesquelles l'État ne peut pas garantir au citoyen le traitement adéquat de ses données, veiller à leur intégrité, leur confidentialité et leur accessibilité au cours du temps, car ce sont des aspects critiques de son usage normal.

Nous sommes d'accord, Mr. González, sur le fait que la technologie de l'information et des communications a un impact significatif sur la qualité de vie des citoyens (sans que pour eux, l'impact soit toujours positif ou neutre d'effet). De même nous serons certainement d'accord pour dire que les valeurs de base que j'ai signalées plus haut sont fondamentales dans une nation démocratique comme le Pérou. Depuis longtemps nous cherchons une alternative permettant de garantir ces principes, qui ne consiste pas à recourir à l'emploi de logiciel libre dans les termes définis dans le projet de Loi.

Quant aux observations que vous formulez, nous allons maintenant les examiner dans le détail :

En premier lieu, vous signalez que : "1. Le projet établit l'obligation pour tout organisme public d'employer exclusivement du logiciel libre, c'est-à-dire à code source ouvert, ce qui transgresse les principes de l'égalité devant la loi, de non-discrimination et les droits à la libre initiative privée, liberté d'entreprendre et de contrat, protégés par la constitution."

Cette appréciation est une erreur. En aucune façon le projet n'affecte les droits que vous énumérez : il se limite à établir les conditions pour l'emploi de logiciel au sein des institutions de l'État, sans s'immiscer d'aucune manière dans les transactions du secteur privé. C'est un principe bien établi que l'État n'a pas la grande liberté de contrat du secteur privé, précisément parce qu'il est limité dans ses actions par le devoir de transparence des actes publics ; et en ce sens, la préservation de l'intérêt commun doit prévaloir lorsqu'il légifère en la matière.

Le projet protège l'égalité devant la Loi, et aucune personne physique ou morale n'est exclue du droit d'offrir ces biens à l'État dans les conditions fixées dans le projet et sans plus de limitations que celles établies dans la loi des Contrats et Acquisitions de l'État (T.U.O. par Décret Suprême No. 012-2001-PCM).

Le projet n'introduit aucune discrimination, puisqu'il établit uniquement **comment** ces biens doivent être fournis (ce qui est une prérogative d'État) et non **qui** doit les fournir (ce qui serait effectivement discriminatoire si les restrictions étaient fondées sur l'origine nationale, raciale, religieuse, idéologique, la préférence sexuelle, etc.) Au contraire, le projet est résolument anti-discriminatoire. Il en est ainsi parce qu'en déterminant, sans l'ombre d'un doute possible, les conditions de sélection d'un logiciel, il évite aux organismes de l'État d'utiliser des programmes dont la licence inclurait des conditions discriminatoires.

Il résulte de ce qui a été exposé dans les paragraphes précédents, que le projet n'attente pas à la libre initiative privée, puisque celle-ci peut choisir sous quelles conditions elle produit un logiciel ; certaines d'entre elles seront acceptables pour l'État, et d'autres ne le seront pas parce qu'elles contrediront la garantie des principes fondamentaux énumérés plus haut. Cette libre initiative est compatible avec la liberté d'entreprendre et la liberté de contrat (dans les limites où l'État peut exercer cette dernière). Tout sujet privé peut produire du logiciel selon les conditions requises par l'État, ou peut s'abstenir de le faire. Personne n'est forcé d'adopter un modèle de production, mais si quelqu'un désire fournir du logiciel à l'État, il lui faudra mettre en oeuvre des mécanismes garantissant les principes qui sont décrits dans le projet.

En guise d'exemple : rien dans le texte du projet n'interdit à votre société d'offrir aux organismes de l'État sa "suite" bureautique, dans les conditions définies dans le projet et à un prix que vous jugerez convenable. Si vous ne le faites pas, cela ne sera pas dû à des restrictions imposées par la loi, mais à des décisions de votre société tenant compte du mode de commercialisation de ses produits, décisions auxquelles l'État ne participe pas.

En poursuivant, vous signalez que : "2. Le projet, en rendant obligatoire l'emploi de logiciel à code source ouvert, établira un traitement discriminatoire et non compétitif pour les contrats et les acquisitions des organismes publics..."

Cette affirmation est une réitération de la précédente, la réponse se trouve quelques lignes plus haut. Cependant, arrêtons nous un instant sur votre appréciation concernant le "traitement ... non compétitif."

A l'évidence, au moment de définir un quelconque type d'acquisition, l'acheteur se fixe des conditions liées à l'usage prévu pour le bien ou le service. A partir de là, il exclut certains fabricants qui n'auront pas la possibilité de rivaliser, sans pour autant les avoir exclus "a priori", mais sur la base d'une série de principes décidés par la volonté autonome de l'acheteur, si bien que le processus s'avère finalement conforme à la loi. Et dans le projet il est établi que **personne**, n'est exclu de la compétition pour autant que la garantie des principes fondamentaux est satisfaite.

De plus le projet *stimule* la compétition, du moins il pousse à générer une offre de logiciel présentant de meilleures conditions d'utilisation, et à optimiser les travaux déjà accomplis, dans un modèle de progrès continu.

D'un autre côté, l'aspect central de la compétitivité est l'opportunité de proposer de meilleures options au consommateur. Il est impossible d'ignorer que le marketing ne joue pas un rôle neutre au moment de la présentation d'une offre au marché (du moins admettre le contraire reviendrait à dire que les investissements réalisés par les entreprises en matière de marketing sont dépourvus de sens), et par conséquent une dépense significative dans ce domaine peut influencer les décisions de l'acheteur. Cette influence du marketing est dans une large mesure réduite par le projet que nous soutenons, puisque le choix proposé dans le marché se base sur le *mérite technique* du produit et sur l'effort de commercialisation du producteur ; en ce sens, la compétitivité est accentuée, et même le plus petit producteur de logiciel peut rivaliser sur un pied d'égalité avec la plus puissante des entreprises.

Il est nécessaire de souligner qu'il n'y a pas de position plus anti-compétitive que celle des grands producteurs de logiciel propriétaire, qui fréquemment, abusent de leur position dominante, parce que dans d'innombrables cas ils proposent comme unique solution aux problèmes soulevés par les utilisateurs : "mettez à jour vos logiciels vers la nouvelle version" (à la charge de l'utilisateur évidemment) ; de plus, les interruptions arbitraires d'assistance technique sur des produits, jugés "anciens" par le fournisseur, sont communes ; ensuite pour obtenir une quelconque assistance technique, l'utilisateur est contraint de migrer (avec un coût non trivial, en particulier lorsque la migration implique des changements de plate-forme matérielle) vers de nouvelles versions. Et comme toute l'infrastructure est consolidée par des formats de données propriétaires, l'utilisateur reste "captif" de la nécessité de continuer à employer les produits du même fournisseur, à moins de consentir un énorme effort pour passer à un autre environnement (probablement tout aussi propriétaire).

Vous ajoutez : "3. 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."

Je ne suis pas d'accord avec ce que vous affirmez. En partie à cause de ce que vous-même signalez dans le paragraphe 6 de votre lettre, concernant le poids relatif des services dans le contexte de l'utilisation du logiciel. Cette contradiction, par elle-même, invalide votre position. 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.

De plus, à la lecture de votre opinion il ressort que le marché de l'État est crucial et indispensable pour l'industrie du logiciel propriétaire, à tel point que si l'État adopte ce projet, il éliminerait complètement ces sociétés du marché. En supposant, ce qui n'est pas le cas, que ce soit vrai, nous en déduisons que l'État subventionne

l'industrie du logiciel propriétaire. Dans cette hypothèse peu probable, l'État aurait alors le droit d'attribuer ses subventions au domaine qu'il considère comme ayant la plus grande valeur sociale il en résulterait que si l'État décide de subventionner le logiciel il devra le faire en préférant le libre par rapport au propriétaire, compte tenu de son effet social et de son utilisation rationnelle de l'argent des contribuables.

Concernant les emplois générés par le logiciel propriétaire dans des pays comme le nôtre, ceux-ci concernent majoritairement des tâches techniques de faible valeur ajoutée ; au niveau local, les techniciens qui offrent du support au logiciel propriétaire produit par des entreprises transnationales ne sont pas en mesure de corriger un bug, pas nécessairement faute de capacité technique ou de talent, mais parce qu'ils ne disposent pas du code source. 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.

C'est un phénomène courant dans les pays en voie de développement que les industries locales de logiciel tirent la majeure partie de leurs revenus des services ou de la fabrication de logiciel "ad hoc". Par conséquent, l'éventuel impact négatif que l'application du projet pourrait avoir dans ce secteur sera compensé par la croissance de la demande de services (à condition que ceux-ci soient conformes aux exigences de qualité). Évidemment, il est probable que les entreprises transnationales de logiciel décidant de ne pas concourir conformément à ces règles du jeu, souffrent d'une perte de revenus en termes de facturation de licences ; néanmoins, considérant que ces entreprises soutiennent que beaucoup de logiciels utilisés par l'État ont été copiés illégalement, on peut penser que l'impact ne sera pas très sérieux. Certainement, en tout cas, leur succès sera déterminé par les lois du marché dont les changements ne peuvent être évités ; de nombreuses entreprises traditionnellement associées au logiciel propriétaire ont déjà franchi le pas (au prix d'investissements importants) pour offrir des services associés au logiciel libre, ce qui démontre que les modèles ne sont pas mutuellement exclusifs.

Avec ce projet l'État décide de préserver certaines valeurs fondamentales. Et il le décide sur la base de ses pouvoirs souverains, sans affecter par là aucune des garanties constitutionnelles. Si ces valeurs peuvent être garanties sans avoir à choisir un modèle économique donné, les effets de la loi seront plus bénéfiques encore. En tout cas, il doit rester clair que l'État n'opte pas pour un modèle économique ; s'il s'avérait qu'il n'existe qu'un seul modèle économique capable de fournir du logiciel qui satisfasse la garantie de base de ces principes, cela relèverait de circonstances historiques et non d'une décision arbitraire en faveur d'un modèle donné.

Poursuivant votre lettre : "4. 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."

Faire allusion de façon abstraite aux "dangers que ceci peut entraîner", sans spécifier un seul exemple de ces supposés dangers, dénote une méconnaissance du sujet. Aussi, permettez-moi d'illustrer quelques-uns de ces points.

Concernant la sécurité :

La sécurité nationale a déjà été évoquée dans les principes fondamentaux du projet. En termes plus précis concernant la sécurité du logiciel lui-même, 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 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 confidentialité [NdT : « NDA » ou *non disclosure agreement*] 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 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 :

Les questions de propriété intellectuelle dépassent le cadre de ce projet, et elles sont couvertes par d'autres lois spécifiques. 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. En réalité, la seule présence de cette question dans vos observations démontre votre confusion quant au cadre légal où vit le logiciel libre. L'incorporation de la propriété intellectuelle d'autrui dans des travaux que l'on s'attribue par la suite n'est pas une pratique courante de la communauté du logiciel libre ; en revanche, c'est malheureusement le cas sur le terrain du logiciel propriétaire. Prenez comme exemple la condamnation par le Tribunal de Commerce de Nanterre, France, le 27 septembre 2001, de Microsoft Corp., à 3 millions de francs en dommages et intérêts, pour violation de la propriété intellectuelle (piratage, pour utiliser le terme malheureux que votre société utilise couramment dans ses publicités).

Vous poursuivez en disant que : "5. 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 bénéfiques pour étayer votre position."

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. Sachant qu'un État démocratique doit respecter ces principes, il ne lui reste aucune autre solution que d'employer du logiciel dont le code source est publiquement disponible et d'échanger de l'information uniquement dans des formats standards.

Si l'État n'employait pas de logiciel présentant ces caractéristiques, il violerait les principes républicains fondamentaux. Par chance, le logiciel libre implique en plus un coût global moindre ; néanmoins, dans l'hypothèse (aisément réfutée) où il coûterait plus cher que le logiciel propriétaire, la seule existence d'un outil logiciel libre efficace pour une fonction informatique déterminée obligerait l'État à l'utiliser ; non par force de ce projet de Loi, mais pour les principes élémentaires que nous avons énumérés au début et qui émanent de l'essence même de l'État de droit démocratique.

Vous poursuivez : "6. Il est faux de penser que le logiciel à code source ouvert est gratuit. Des études du Gartner Group (organisme étudiant le marché technologique reconnu au niveau mondial) ont révélé que le coût d'acquisition du logiciel (système d'exploitation et applications) ne représente que 8% du coût total que les entreprises et les institutions doivent assumer pour une utilisation rationnelle et réellement bénéfique de la technologie. Les autres 92% sont constitués des coûts d'installation, de déploiement, de support, de maintenance, d'administration et d'indisponibilité."

Cet argument répète celui déjà donné au paragraphe 5 et contredit en partie le paragraphe 3. Aussi nous nous en remettons aux précédents commentaires à des fins de brièveté. Nonobstant, permettez-moi de signaler que votre conclusion est fautive d'un point de vue logique : que le coût du logiciel selon le Gartner Group ne soit que de 8% du coût total d'utilisation, n'invalide d'aucune manière l'existence de logiciel gratuit, c'est-à-dire, dont le coût de la licence est zéro.

De plus dans ce paragraphe vous indiquez fort justement que les composants de service et les pertes pour indisponibilité forment une partie substantielle du coût total d'utilisation du logiciel ; ce qui, vous le noterez, entre en contradiction avec votre affirmation de la valeur mineure des services suggérée dans le paragraphe 3. En réalité, l'utilisation de logiciel libre contribue significativement à la diminution des coûts restants du cycle de vie du logiciel. Cette réduction de l'impact économique de l'installation, du support, etc. se note dans de nombreux domaines ; d'un côté, le modèle compétitif de services autour du logiciel libre, dont il est possible d'acheter le support et la maintenance auprès d'une offre variée qui rivalise sur le rapport qualité/prix. Ceci est valable pour l'installation, le déploiement, et le support, et en grande partie pour la maintenance. En second lieu, la caractéristique de

reproductibilité du modèle fait que la maintenance effectuée pour une application est facilement réutilisable, sans impliquer des coûts importants (c'est-à-dire, sans payer plus d'une fois pour la même chose) car les modifications, si on le souhaite, peuvent être incorporées au patrimoine commun de la connaissance. Troisièmement, l'énorme coût d'indisponibilité ("écrans bleus de la mort", code mal-intentionné tel que les virus, les vers et les chevaux de Troie, exceptions, fautes générales de protection et nombre d'autres maux connus) est considérablement réduit par l'emploi de logiciel plus stable ; et il est bien connu qu'une des vertus les plus remarquables du logiciel libre est sa stabilité.

Vous affirmez plus loin que : "7. 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 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."

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.

D'autre part, il existe certainement des modalités de licence en volume (malheureusement, le logiciel propriétaire ne satisfait pas les principes de base). Mais, comme vous l'avez noté dans le paragraphe immédiatement antérieur de votre lettre, cela ne permet que de réduire l'impact d'un composant qui ne pèse pas plus de 8% du coût total.

Vous poursuivez : "8. De plus, l'alternative adoptée pour le projet (i) est clairement plus coûteuse du fait des coûts élevés de la migration logicielle, et (ii) 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."

Analysons votre affirmation en deux parties. 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.

Vous dites plus loin que : "9. La majeure partie du logiciel à code source ouvert n'offre pas de niveaux de service adéquats, pas plus que de garantie de fabricants reconnus pour favoriser une grande productivité de la part des utilisateurs, ce qui a conduit différentes organisations publiques à revenir sur leur décision d'utiliser du logiciel à code source ouvert et à utiliser du logiciel commercial en lieu et place."

Cette observation n'est pas fondée. Compte tenu de la garantie, votre argument est réfuté par la réponse au paragraphe 4. Concernant les services de support, il est possible d'utiliser du logiciel libre sans eux (de la même manière qu'on le fait avec du logiciel propriétaire), mais quiconque le souhaite peut obtenir du support séparément, soit de la part d'une entreprise locale, soit de sociétés internationales, de la même manière que pour le logiciel propriétaire.

D'autre part, vous contribueriez beaucoup à notre analyse si vous pouviez nous donner des informations concernant les projets de logiciel libre *implantés* dans des entités publiques et qui ont été abandonnés en faveur de logiciel propriétaire. Nous connaissons un bon nombre de cas où l'inverse s'est produit, mais n'avons pas d'information au sujet des cas auxquels vous faites référence.

Vous continuez en observant que : "10. Le projet décourage la créativité de l'industrie péruvienne du logiciel, qui a un chiffre d'affaires de 40 millions de dollars US par an, exporte pour 4 millions de dollars US (10e au rang des produits d'exportation non traditionnels, plus que l'artisanat) et est une source d'emplois hautement qualifiés. Avec une loi qui incite à l'utilisation du logiciel à code source ouvert, les programmeurs de logiciel perdent leurs droits de propriété intellectuelle et leur principale source de revenus."

Il est assez clair que personne n'est obligé de commercialiser son code sous forme de logiciel libre. La seule chose à prendre en compte est que, si cela n'est pas fait, on ne pourra pas le vendre au secteur public. Ce n'est en aucun cas le principal marché pour l'industrie nationale du logiciel. Plus haut nous avons abordé quelques-unes des questions relatives à l'influence du projet sur la génération d'emplois techniques hautement qualifiés et dans de meilleures conditions de compétitivité, il n'est donc pas nécessaire d'insister sur ce point.

Ce qui suit dans votre affirmation est erroné. D'un côté, aucun auteur de logiciel libre ne perd ses droits de propriété intellectuelle, à moins qu'il n'ait exprimé sa volonté de placer son oeuvre dans le domaine public. Le mouvement du logiciel libre a toujours été extrêmement respectueux de la propriété intellectuelle, et a donné une reconnaissance publique très large à ses auteurs. Des noms tels que ceux de Richard Stallman, Linus Torvalds, Guido van Rossum, Larry Wall, Miguel de Icaza, Andrew Tridgell, Theo de Raadt, Andrea Arcangeli, Bruce Perens, Darren Reed, Alan Cox, Eric Raymond, et bien d'autres, sont mondialement reconnus pour leurs contributions au développement de logiciel aujourd'hui utilisé par des millions de personnes partout dans le monde, alors que les noms des auteurs d'excellents composants logiciels propriétaires, demeurent dans l'anonymat. D'un autre côté, affirmer que les revenus de droits d'auteur constituent la source principale de revenus des programmeurs péruviens est pour le moins risqué, en particulier quand on n'a apporté aucune preuve à cet effet, ni aucune démonstration de comment l'emploi de logiciel libre par l'État influencerait ces revenus.

Vous poursuivez en disant que : "11. Le logiciel à code source ouvert, puisqu'il peut être distribué gratuitement, ne permet pas de générer des revenus pour ses

développeurs par le biais de l'exportation. De cette manière, on affaiblit la synergie de la vente de logiciel à d'autres pays et par conséquent la croissance de cette industrie, alors qu'au contraire les normes d'un gouvernement doivent stimuler l'industrie locale."

Cette affirmation démontre une fois de plus une méconnaissance totale des mécanismes et du marché du logiciel libre. Elle tente d'affirmer que le marché de cession des droits non exclusifs d'utilisation à titre onéreux (vente de licence) est le seul possible pour l'industrie informatique alors que, comme vous l'avez signalé quelques paragraphes plus haut, il n'est en aucun cas le plus important. Les incitations, émanant de ce projet, à une meilleure offre de personnels qualifiés et à une expérience du logiciel libre à grande échelle permettront aux techniciens nationaux de se placer à un niveau très compétitif sur le marché du travail international.

Vous signalez plus loin que : "12. Dans le Forum on a discuté de l'importance de l'emploi de logiciel à code source ouvert dans l'éducation, sans commentaire sur le retentissant fracas de cette initiative dans un pays comme le Mexique, où précisément les fonctionnaires de l'État qui fondèrent le projet, déclarent aujourd'hui que le logiciel à code source ouvert ne permet pas d'offrir une expérience d'apprentissage aux écoliers, qu'il n'a pas eu la capacité au niveau national de fournir du support pour cette plate-forme, et qu'il n'a pas pris en compte l'intégration de la plate-forme existante dans les écoles."

Effectivement, le Mexique a fait marche arrière avec le projet Red Escolar. Cela est dû, précisément au fait que les initiateurs du projet mexicain utilisèrent le coût des licences comme principal argument, au lieu des autres raisons stipulées dans notre projet et qui sont plus fondamentales. Compte tenu de cette erreur conceptuelle, aggravée par l'absence d'appui effectif de la part du SEP (Secrétariat à l'Education Publique), ils décidèrent que l'implantation de logiciel libre dans les écoles consistait à suspendre le budget logiciel et en échange à leur envoyer un CD ROM contenant GNU/Linux. Bien sûr, ceci échoua et il ne pouvait en être autrement, de même qu'échouent les laboratoires scolaires qui utilisent des logiciels propriétaires sans disposer d'un budget pour l'installation et la maintenance. C'est précisément pour cela que notre projet de loi ne se limite pas à recommander l'emploi de logiciel libre, mais reconnaît la nécessité et ordonne la création d'un plan de migration viable, dans lequel l'État encadre précisément la transition technique pour bénéficier des avantages du logiciel libre.

Vous terminez par une question rhétorique : "13. Si le logiciel à code source ouvert satisfait tous les pré-requis des entités de l'État pourquoi une loi pour l'adopter ? Ne devrait-ce pas être le marché qui décide librement quels sont les produits qui donnent le plus de bénéfices ou de valeur ?"

Nous sommes d'accord sur le fait que pour le secteur privé, c'est le marché qui doit décider quel produit utiliser et il ne serait pas admissible que l'État interfère. Mais dans le secteur public, le raisonnement n'est pas le même : comme nous l'avons déjà dit, l'État collecte, manipule et transforme de l'information qui ne lui appartient pas, mais qui lui a été confiée par les citoyens qui, par force de loi, n'ont pas d'autre choix que de le faire. En contrepartie de cette obligation légale, l'État doit mettre en oeuvre des mesures extrêmes pour sauvegarder l'intégrité, la confidentialité et l'accessibilité de ces informations. L'emploi de logiciel propriétaire soulève de sérieux doutes quant

à l'accomplissement de ces missions, faute d'évidence concluante à ce propos, et par conséquent il n'est pas apte à être utilisé dans le secteur public.

La nécessité d'une loi se fonde d'un côté sur la matérialisation des principes fondamentaux énoncés plus haut dans le domaine spécifique du logiciel ; d'un autre côté, il est un fait que l'État n'est pas une entité idéale homogène mais qu'il est composé de multiples organismes avec différents degrés d'autonomie de décision. Étant donné que l'emploi de logiciel propriétaire est inapproprié, le fait d'établir ces règles dans la loi évitera que la décision discrétionnaire d'un quelconque fonctionnaire mette en péril l'information qui appartient aux citoyens. Et, par-dessus tout, elle constitue une réaffirmation actualisée par rapport aux moyens de traitement et de communication de l'information employés aujourd'hui, du principe républicain du service public.

Conformément à ce principe universellement accepté, le citoyen a le droit de connaître toute l'information en possession de l'État qui ne soit pas couverte par une déclaration de secret conforme à la loi. Le logiciel traite de l'information et il est lui-même de l'information. Information dans un format spécial, susceptible d'être interprété par une machine pour exécuter des actions, mais sans l'ombre d'un doute information cruciale parce que le citoyen dispose d'un droit légitime de savoir, par exemple, comment se comptabilise son vote ou se calculent ses impôts. Et pour cela, il faut pouvoir accéder librement au code source et éprouver les programmes utilisés pour le comptage électoral ou le calcul des impôts.

Je vous salue avec l'expression de ma considération la meilleure, soyez assuré que mon bureau sera toujours ouvert à l'exposé de vos points de vue, à quelque niveau de détail que vous jugeriez convenable.

Veuillez agréer mes salutations distinguées,

DR. EDGAR DAVID VILLANUEVA NUÑEZ

Membre du Congrès de la République du Pérou.

Version originale disponible sur le site de [Peruvian Activism](#).

Traduction et adaptation : **Guy Brand** <guybrand @ chimie.u-strasbg.fr>

Relectures et corrections : **Cyril Chaboisseau, Georges Khaznadar,**

Yves Ouvrard, Alain Riffart, Stéphane Casset.

\$Id: rescon-fr.html,v 1.20 2002/05/15 20:53:19 bug Exp bug \$

Annexe 3.

The table of equivalents / replacements / analogs of Windows software in Linux.

Last update: 26.02.03.

URL: <http://linuxshop.ru/linuxbegin/win-lin-soft-en/>.

The russian version is [here](#).

25.02.03: This table is mentioned at Linuxfr.org! Unfortunately, I can't read French, but anyway I have received a huge amount of comments and additions to my email - thanks!!! I do not have time even to read them all :). Soon this page will be moved to new database engine with web-forms for additions, commentaries, etc - so watch out.

27.02.03: This table is mentioned at LWN.net!!! Please, do not post link to this page on Slashdot and Newsforge :). Thanks.

One of the largest difficulties in migrating away from Windows to Linux is the lack of comparable software. Newbies usually search for analogs of Windows software for Linux, and advanced Linux-users cannot answer their questions since they often don't know too much about Windows :). This list of equivalents / replacements / analogs of Windows software in Linux is based on our own experience and on the information from the visitors of this page (thanks!).

This table is not a static - new programs can be added both to left and to right sides, and it is not fact that right column will be filled immediately. In the future we plan to migrate this table to the PHP/MySQL engine, so visitors could add the program themselves, vote for analogs, add comments, etc.

If you want to add the program in table, send mail to xoid26@linux-online.ru with the name of the program, the OS, the description (the purpose of the program, etc), and the link to official site of this program (if you know it). **All commentaries, remarks, corrections, offers and bugreports** are always welcome - send it all to xoid26@linux-online.ru.

Notes:

- 1) By default all Linux-programs in this table are free. Proprietary software for Linux is marked with the sign [Proprietary].
- 2) If there is nothing in the field of the table, except "???" - the authors of the table do not know what to place there.
- 3) If the sign (???) stands after the name of the program - the authors of the table are not sure about this program.

4) If the sign "?" stands after the description of the program - this description is wrong (may be).

The important ideological difference between Windows and Linux:

The majority of the Windows programs are made on principle "all-in-one" (each developer adds everything to his or her product). In the same way, this principle is called the "Windows-way".

The ideology of UNIX/Linux - one component or one program must execute only one task, but execute it well. ("UNIX-way"). The programs under Linux can be thought of as being like the LEGO building blocks. (for instance, if there is a program for spell checking, it can be used with the text editor or email client; or if there is the powerful command-line program for files downloading, it is simple to write the graphic interface (Front-end) for it, etc).

This principle is very important and it is necessary to know it while searching for analogs of Windows-programs in Linux :).

Attention! There may be many mistakes and bugs in the table!! ([report about the bugs](#)).

Places where you can get Linux software:

[FreshMeat.net](#) | [SourceForge.net](#) - Two biggest portals. You can find everything you want!!

[LinuxApps.com](#) | [IceWalkers.com](#) | [Linux.Tucows.com](#) - I like these catalogs.
[Download.com / Linux](#) | [LinuxSoftware.org](#) | [DaveCentral.com / Linux](#) | [FileWatcher.org](#) | [LinuxArchives.com](#) - other catalogs.

Programs / packages / libraries search systems: [RPMseek.org](#) | [Tuxfinder.com](#) | [RPMfind.net](#)

Programs and games for Windows, that can be run under Wine/WineX:

1) **Games for Windows, that can be run under WineX** (the information from the official Transgaming site). This is a [search form](#), and this is a [full list of games](#) (very big!).

2) [The biggest catalog of Windows apps, running under Wine.](#) The database of Codeweavers contains information about ~1000 applications, and the catalog is structured and has a navigation system.

The Sections:

- 1) [Networking.](#)
- 2) [Work with files.](#)
- 3) [Desktop / System software.](#)
- 4) Multimedia:

- 4.1) [Audio / CD.](#)
- 4.2) [Graphics.](#)
- 4.3) [Video and other.](#)

- 5) [Office/business.](#)
- 6) [Games.](#)
- 7) [Programming and development.](#)
- 8) [Server software.](#)
- 9) [Scientific and special programs.](#)
- 10) [Other / Humour :\).](#)

Description of the program, executed task	Windows	Linux
1) Networking.		
Web browser	Internet Explorer, Mozilla for Windows, Opera, ...	<ul style="list-style-type: none"> 1) Mozilla. 2) Galeon. 3) Konqueror. 4) Opera. 5) Phoenix. 6) Nautilus. 7) Links. (with "-g" key). 8) Dillo. (Russian language patches - here).
Console web browser	Lynx for Windows	<ul style="list-style-type: none"> 1) Links. 2) Lynx. 3) w3m.
Email client	Outlook Express, Mozilla for Windows, Eudora, Becky	<ul style="list-style-type: none"> 1) Evolution. 2) Mozilla messenger. 3) Sylpheed, Sylpheed-claws. 4) Kmail. 5) Gnus. 6) Balsa. 7) Bynari Insight GroupWare Suite. [Proprietary]
Email client in MS Outlook style	Outlook	<ul style="list-style-type: none"> 1) Evolution. 2) Bynari Insight GroupWare Suite. [Proprietary]
Email client in The Bat style	The Bat	<ul style="list-style-type: none"> 1) Sylpheed. 2) Sylpheed-claws. 3) Kmail. 4) Gnus. 5) Balsa.
Console email client	Mutt for Windows (???)	<ul style="list-style-type: none"> 1) Pine. 2) Mutt. 3) Gnus.

News reader	Outlook, Mozilla for Windows	<ol style="list-style-type: none"> 1) Knode. 2) Pan. 3) NewsReader. 4) Mozilla. 5) Sylpheed / Sylpheed-claws. Console: <ol style="list-style-type: none"> 6) Pine. 7) Mutt. 8) tin. 9) slrn.
Quick and free (of charge) browser in Opera style	Opera + crack	Phoenix .
Files downloading	Flashget, Go!zilla, Reget, Getright, ...	<ol style="list-style-type: none"> 1) Downloader for X. 2) Caitoo (former Kget). 3) Wget (console, standard). 4) Aria. 5) Axel.
Sites downloading	Teleport Pro, ...	<ol style="list-style-type: none"> 1) Htrack. 2) WWW Offline Explorer. 3) Wget (console, standard). 4) Downloader for X. 5) Pavuk.
FTP-clients	Bullet Proof FTP, CuteFTP, ...	<ol style="list-style-type: none"> 1) Gftp. 2) Dpsftp. 3) KBear. 4) IglooFTP. 5) Nftp.
Console FTP-clients	FTP in Far, ftp.exe	<ol style="list-style-type: none"> 1) Ncftp. 2) Lftp.
IRC-clients	Mirc, Klient, VIRC, Xircon, Pirch, XChat for Windows	<ol style="list-style-type: none"> 1) Xchat. 2) KVirc. 3) Irssi. 4) BitchX. 5) Ksirc. 6) Epic. 7) Sirc.
Local chat clients without server	QuickChat	Echat .
Local messaging with Windows machines	WinPopUp	<ol style="list-style-type: none"> 1) LinPopUp. 2) Kpopup.
Instant Messaging clients	ICQ, MSN, AIM, ... Trillian ICQ (freeware, can replace all common IM clients), Miranda	<ol style="list-style-type: none"> 1) Licq (ICQ). 2) Centericq (ICQ, console). 3) Alicq (ICQ). 4) Mica (ICO).

		<ul style="list-style-type: none"> 5) GnomeICU (ICQ). 6) Gaim. (AIM) 7) Kopete. 8) Everybuddy. 9) Simple Instant Messenger. 10) Imici Messenger. 11) Ickle (ICQ). 12) aMSN (MSN). 13) Kmerlin (MSN).
Jabber IM clients	Exodus, JAJC, myJabber, Psi, RhymBox, Rival, Skabber, TipicIM, Vista, Yabber	<ul style="list-style-type: none"> 1) Tkabber. 2) Psi. 3) Jabber for Linux. 4) JAJC.
Monitoring sites and mailboxes, displaying news headlines or full articles	WatzNew	Mozilla (???)
Video/audio conference	NetMeeting	GnomeMeeting .
Voice communication	Speak Freely	<ul style="list-style-type: none"> 1) Speak Freely for Unix. 2) TeamSpeak.
Firewall (packet filtering)	BlackICE, ATGuard, ZoneAlarm, Agnitum Outpost Firewall, ...	<p>iptables or more outdated ipchains (console, standard). Front ends:</p> <ul style="list-style-type: none"> 1) Kmyfirewall. 2) Easy Firewall Generator. 3) Firewall Builder. 4) Shorewall. 5) Guarddog.
IDS (Intrusion Detection System)	BlackICE, Agnitum Outpost Firewall	<ul style="list-style-type: none"> 1) Snort. 2) Portsentry / Hostsentry / Logsentry.
Visual route	VisualRoute	<ul style="list-style-type: none"> 1) Xtraceroute. 2) VisualRoute for Linux. 3) Mtr.
Content filtering	Proxomitron, ATGuard, Agnitum Outpost Firewall, ...	<ul style="list-style-type: none"> 1) Squid. 2) Squidguard. 3) Privoxy. 4) Fork. 5) Mozilla (has a possibility to filter the content).
Traffic restriction	???	<ul style="list-style-type: none"> 1) IP Relay.
Traffic accounting	???	<ul style="list-style-type: none"> 1) Tcp4me. 2) Getstatd. 3) Ipacct. 4) Ipac-ng. 5) Ibaudit.

		<ul style="list-style-type: none"> 6) Lanbilling. 7) Orso (full Squid traffic). 8) Talinux.
Peer-to-peer clients / servers, file sharing	Morpheus (Gnutella), WinMX, Napster, KaZaA (Fasttrack), eDonkey / eMule	<ul style="list-style-type: none"> 1) LimeWire for Linux. (Gnutella) 2) Lopster. (OpenNAP) 3) Gnapster. (OpenNAP) 4) Mldonkey. (eDonkey) 5) eDonkey for Linux. (eDonkey) 6) cDonkey. (eDonkey) 7) Gift client / server (Fasttrack) 8) eMule. 9) ed2k_gui.
Program for working with sound modem with many functions - from answerback and AON to fax device	PrimaFax, Venta Fax	<ul style="list-style-type: none"> 1) Gfax. (???) 2) PrimaFax for Linux. [Proprietary, 99\$] 3) mgetty + voice + sendfax.
Work with faxes	WinFax	<ul style="list-style-type: none"> 1) HylaFax. 2) Fax2Send. 3) Efax.
Dialup	Vdialer, etЯ	<ul style="list-style-type: none"> 1) Kppp. 2) X-isp. 3) wvdial. (Front ends: X-wvdial, kvdial, gtkdial). 4) Gppp. 5) Kinternet. 6) Rp3.
FTN editor	Golded	<ul style="list-style-type: none"> 1) Golded. 2) Golded+. 3) Msged.
FTN tosser	FastEcho, hpt	<ul style="list-style-type: none"> 1) hpt. 2) CrashEcho.
FTN mailer	SF-Mail, T-Mail	<ul style="list-style-type: none"> 1) ifcico. 2) qico. 3) Bforce.
Remote management	VNC, Remote Administrator, Remote Assistance, Symantec pcAnywhere, Windows Terminal Server, Rdesktop	<ul style="list-style-type: none"> 1) VNC. 2) ssh. 3) Remote management is built-in in XFree86. 4) freeS/WAN.
Transmission of the files on modem	HyperTerminal, Terminate, etc	<ul style="list-style-type: none"> 1) minicom. 2) Mstern.
Work with ssh	Putty, Irlex	<ul style="list-style-type: none"> 1) Kssh. 2) ssh. 3) GTelnet. (Telnet, SSH, Rlogin)

Network monitoring tool	Dumeter, Netmedic	1) Gkrellm . 2) Big Brother . 3) Etherape . 4) Nagios. 5) Tkined.
Network maintenance tool	HP OpenView	1) HP OpenView agents for Linux. [Proprietary] 2) Big Brother . 3) Cheops . 4) Tkined.
Protocols analysing	Windump , Ethereal	1) Ethereal . 2) Tcpdump . 3) Etherape . 4) ipxdump . (+ ipxparse for parsing)
Security scanner	???	Nessus.
Routing	???	1) Iproute2 . 2) iptables or more outdated ipchains (console, standard).
Utilites and libraries for Ethernet/IP testing	lcrzo	lcrzo
2) Work with files.		
File manager in FAR and NC style	FAR, Norton Commander, etc	1) Midnight Commander . 2) X Northern Captain . 3) Deco. 4) Portos Commander .
File manager in Windows Commander style	Total Commander (former Windows Commander)	1) Midnight Commander . 2) Krusader . ~1.3 Mb. Suitable GUI file manager for KDE. Support archiver, FTP with suitable notebook, Samba, built-in editor and graphics viewer. 3) Kcommander . 4) FileRunner (written on TCL/TK). 5) Linux Commander . 6) LinCommander . 7) Rox . 8) Emelfm .
File manager in Windows style	Windows Explorer	1) Konqueror . 2) Gnome-Commander . 3) Nautilus .
Quick viewing of local HTML documents	Internet Explorer	1) Dillo . (Russian language patches - here). 2) Konqueror . 3) Nautilus . 4) Lynx / Links .

Viewing all operations with files	Filemon	Filemon for Linux.
Work with UDF	Roxio (former Adaptec) UDF Reader, Roxio Direct CD	Linux-UDF.
Work with multi session CD + recovering information from faulty multi session disks	-	CDFS.
3) Desktop / System software.		
Simple text editor	Notepad, ...	<ol style="list-style-type: none"> 1) Kedit (KDE). 2) Gedit (Gnome). 3) Gnotepad. 4) Kate (KDE). 5) KWrite (KDE). 6) Nedit. 7) Vim, Emacs. 8) pico, joe (console). 9) Fte (console).
Simple text editor for code editing	Editor in DOS Navigator, MultiEdit	<ol style="list-style-type: none"> 1) Kate (KDE). 2) Nedit. 3) CodeCommander. 4) Vim, Emacs. 5) Mcedit (console, goes with mc).
Simple text editor with cyrillic encodings support	Bred, Rpad32, Aditor	<ol style="list-style-type: none"> 1) Kate (KDE). 2) Nedit. 3) Fte (console). 4) Patched version of Midnight Commander. 5) Vim, Emacs.
Work with compressed files	WinZip, WinRar	<ol style="list-style-type: none"> 1) FileRoller. 2) Gnozip. 3) LinZip. 4) Ark (kdeutils). 5) KArchiveur. 6) Gnohive. 7) RAR for Linux.
Viewing PDF	Adobe Acrobat Reader	<ol style="list-style-type: none"> 1) Acrobat Reader for Linux. 2) Xpdf. 3) GV.
Creating PDF	Adobe Acrobat Distiller	<ol style="list-style-type: none"> 1) Adobe Acrobat Distiller for Linux. 2) PStill. 3) PDFLatex.

Crypto	PGP	1) PGP. 2) GnuPG .
Crypto of the disc volumes	EFS (standard), PGP-Disk, BestCrypt, Private Disk Light	1) CFS. 2) TCFS . 3) BestCrypt .
Task scheduler	mstask, nCron	cron, at (standard, console).
Text recognition (OCR)	FineReader (cyrillic)	1) ClaraOcr . 2) Gocr .
Translators (cyrillic)	Prompt, Socrat	???
Eng-rus dictionaries (cyrillic)	ABBYY Lingvo, Socrat	1) Mueller . 2) Ksocrat . 3) dict+Kdict. 4) DictX . 5) Groan . 6) Mova. 7) Slowo.
Work with scanner	Programs on CD with scanner	1) Xsane . 2) Kooka .
Antivirus	Dr. Web, Kaspersky, NAV, ...	1) Dr. Web for Linux . 2) Kaspersky for Linux . 3) Ravantivirus .
System configuration	Mconfig	1) setup in Red Hat Linux. 2) redhat-config-<feature>. The graphic configurators in Redhat 8.0. 3) Linuxconf . 4) Drakeconf. 5) Webmin. 6) yast and yast2 in SuSE. 7) sysinstall in FreeBSD. 8) /dev/hands :).
Boot managers	System Commander, PowerQuest Boot Magic, ...	1) Grub . 2) Lilo. 3) ASPLoader. 4) Acronis OS Selector [Proprietary]. 5) Ranish Partition Manager . 6) osbs . 7) Symon . 8) Smart Boot Manager .
Hard disk partitions manager	PowerQuest Partition Magic	1) fips. 2) GNU Parted . (GUI - QTParted). 3) Diskdrake (Mandrake). 4) Paragon Partition Manager [Proprietary].

LVM + soft-RAID + parted + ...	-	EVMS .
Backup software	ntbackup (standard), Legato Networker	1) Legato Networker . 2) Lonetar . 3) Disk Archive .
Making images of disk partitions	DriveImage, Ghostpe	1) PartitionImage .
Task manager	TaskMan (standard), TaskInfo, ProcessExplorer NT.	1) top (the console, standard). 2) Gtop, Ktop, ...
Mouse gestures	Sensiva	Kgesture.
Text to speech	MS text to speech	1) KDE Voice Plugins . 2) Festival .
Voice recognition	ViaVoice, DragonNaturally Speaking	1) ViaVoice. 2) Sphinx .
Program for files comparison	Araxis Merge	1) WinMerge . 2) Mgdiff . 3) diff, patch.
Stream text editor?	???	sed, awk.
PIM / DB / notebook with tree view	TreePad Lite, Leo	1) TreePad Lite for Linux . 2) Yank . 3) TreeLine . 4) Gjots . 5) Leo .
Program for quick switching between resolutions and frequencies	Integrated with system, comes on CD with video card	Multires for Linux .
Program for meta search	Integrated with system, Indexing service	find (console, standard). GUI: 1) Gsearchtool . 2) Kfind .
System monitoring	System monitor (integrated with system)	1) Gkrellm .
Program for logs viewing	???	1) Xlogmaster . 2) Fwlogview . (firewall)
Data Recovery Tools	R-Studio (supports Linux partitions)	1) e2undel . 2) myrescue . 3) TestDisk . 4) unrm . 5) Channel 16 .
Text files encoder with automatic detection of character set	Total Recorder, ...	1) Enca .

4.1) Multimedia (audio / CD).		
Music / mp3 / ogg players	Winamp	<ol style="list-style-type: none"> 1) XMMS (X multimedia system). 2) Noatun. 3) Freeamp. 4) Winamp for Linux. 5) Xamp. 6) GQmpeg. 7) SnackAmp.
Console music / mp3 / ogg players	mpg123 for Windows, dosamp	<ol style="list-style-type: none"> 1) mpg123 / ogg123. 2) mpg321. 3) Orpheus.
Programs for CD burning with GUI	Nero, Roxio Easy CD Creator, ...	<ol style="list-style-type: none"> 1) K3b. (KDE) 2) XCDRoast. 3) KOnCd. 4) Eclipt Roaster. 5) Gnome Toaster. 6) CD Bake Oven. 7) KreateCD. 8) SimpleCDR-X.
CD player	CD player	<ol style="list-style-type: none"> 1) KsCD. 2) Orpheus. (console)
Slowing the rotation of CD	Slowcd, Cdslow	<ol style="list-style-type: none"> 1) <code>hdparm -E <speed></code> 2) <code>eject -x <speed></code> 3) <code>cdspeed</code>.
CD ripping / grabbing	Windows Media Player, AudioGrabber, Nero, VirtualDrive, VirtualCD, ...	<ol style="list-style-type: none"> 1) Grip. 2) Audacity. 3) RipperX. 4) tkcOggRipper. 5) <code>cdda2wav</code>. 6) Gnome poaster. 7) Cdparanoia. 8) Cd2mp3. 9) Dagrab. 10) SimpleCDR-X. 11) RatRip.
Mp3 encoders	Lame	<ol style="list-style-type: none"> 1) Lame. 2) Bladeenc. 3) NotLame. 4) L3enc. [proprietary]
Work with "Real" protocol	RealPlayer	RealPlayer for Linux .
Programs for receiving radio station	VC Radio, FMRadio, Digband Radio	<ol style="list-style-type: none"> 1) xradio. 2) cRadio. 3) Xmradio. 4) RDJ.

		5) RadioActive . 6) XMMS-FMRadio .
Audio editors	SoundForge, Cooledit, ...	1) Sweep. 2) WaveForge . 3) Sox . 4) Audacity . 5) GNUSound . 6) Ecasound .
Multitrack audio processor	???	1) Ecasound .
Sound tracker	Fasttracker, ImpulseTracker	1) Soundtracker . 2) Insotracker . 3) CheeseTracker .
Sound mixer	sndvol32	1) Opmixer . 2) aumix. 3) mix2000. 4) Mixer_app (for WindowMaker)
Notes editors	Finale , Sibelius , SmartScore	1) LilyPond . 2) Notedit . 3) MuX2d .
Midi-sequencer	Cakewalk	1) RoseGarden . 2) Brahms .
Music creation	Cakewalk, FruityLoops	1) RoseGarden .
The Synthesizer?	Virtual waves	Arts Builder (???)
4.2) Multimedia (graphics).		
Graphic files viewer	ACDSee, IrfanView	1) Xnview . 2) GQview . 3) CompuPic . 4) Kuickshow . 5) GTKSee . 6) pornview . 7) xv . 8) imgv . 9) Gwenview . 10) Gliv .
Viewing the graphic files in console	QPEG	zgv .
Simple graphic editor	Paint	1) Kpaint . 2) Tuxpaint . 3) Xpaint .
Powerful graphic editor in PhotoShop style	Adobe Photoshop, Gimp for Windows , Paint Shop Pro	1) Gimp , FilmGimp . 2) ImageMagick .

Programs for work with vector graphics	Adobe Illustrator, Corel Draw 14, Freehand	1) Sodipodi . 2) xfig . 3) Sketch . 4) Karbon14 and Kontour . 5) OpenOffice Draw . 6) Dia . 7) Tgif . 8) Gestalter .
-	Corel Draw 9	Corel Draw 9 for Linux.
-	Corel PhotoPaint 9	Corel PhotoPaint 9 for Linux .
Program for text decoration	Wordart	OpenOffice Draw .
Program for ASCII-drawing	ANSI DRAW	CanvASCII .
Converting the graphic files in pseudographics	???	aalib.
Flash playing	Flash Player	1) Flash Player for Linux (???) . 2) Plugin for Netscape/Mozilla (download here or here).
Flash creation	Macromedia Flash	???
Web-graphics creation	Macromedia Fireworks	Gimp .
3D-graphics	3D Studio MAX	1) Blender . 2) Maya . 3) KPovModeler . 4) K3Studio . 5) Moonlight . 6) GIG3DGO .
The instrument for making the photographic quality scene based of ArchiCAD 3D-models	ArtLantis Render	???
Icon editor	Microangelo	1) Gnome-iconedit . 2) Kiconedit .
Small program for making screenshots	Integrated with system (PrintScreen), Snag it, ...	1) Ksnapshot . 2) Xwpick .
Drawing structure diagrams of the database	Access	1) Dia . 2) Toolkit for Conceptual Modelling .
4.3) Multimedia (video and other)		
Video / mpeg4 players	BSplayer, Zoomplayer, Windows Media Player	1) Mplayer . 2) Xine .

Console video / mpeg4 players	do not exist	Mplayer.
DVD players	PowerDVD, WinDVD, MicroDVD, Windows Media Player	1) Ogle. 2) Mplayer. 3) Xine.
On input - DVD disc. On output - several CDs with film in DivX.	Gordian Knot	Dvdrip.
Simple video creation and editing	Windows Movie Maker	1) iMira Editing. 2) MainActor. 3) Broadcast 2000.
Professional video creation and editing	Adobe Premiere, Media Studio Pro	1) iMira Editing. 2) Cinelerra. 3) MainActor. 4) Broadcast 2000. 5) Lives.
Cutting video	Virtual Dub	1) Avidemux. 2) Kino.
Converting video	Virtual Dub	Mencoder (from Mplayer).
Work with TV-tuner	AVerTV, PowerVCR 3.0, CinePlayer DVR, ...	1) Kwintv. 2) Xawtv. 3) Zapping. 4) GnomeTV.
Work with TV-tuner in console	-	1) fbtv (console) 2) aatv (console, tv in pseudographics with aalib)
Work with QuickTime format	QuickTime Player	1) QuickTime Player for Linux. 2) Mplayer + Sorenson codec.
Program for creation of 2D and 3D effects, animation and composition	Adobe After Effects	???
Landscape generation	Bryce	???
World construction	World Construction Set, Animatek World Builder	???
5) Office/business.		
Office suite	MS Office, Openoffice for Windows	1) Openoffice. 2) Staroffice. [Proprietary] 3) Koffice. 4) HancomOffice. 5) Gnome Office. 6) Annlixware Office.

		7) Siag Office . 8) TeX, LaTeX , ...
Office suite	WordPerfect Office 2000	WordPerfect Office 2000 for Linux. (No longer available at Corel website. It was Windows version, running under Wine :).
Word processor	Word	1) Abiword . 2) WordPerfect . 3) Ted . 4) OpenOffice Writer . 5) Kword .
Spreadsheets	Excel	1) Gnumeric . 2) Abacus . 3) OpenOffice Calc . 4) Kspread .
Drawing graphs	Excel, ...	1) Kivio (Koffice) . 2) Dia . 3) KChart . 4) xfig . 5) Gnuplot . 6) GtkGraph .
Creating presentations	MS PowerPoint	OpenOffice Impress .
Local database	Access	1) KNoda . 2) Gnome DB Manager . 3) OpenOffice + MySQL . 4) InterBase . 5) BDB .
Financial accounting package (global)	???	1) Hansa Business Solutions . [Proprietary]
Financial accounting package (Russia)	"1C: Accounting"	1) Hansa Business Solutions . [Proprietary] 2) IceB . 3) "Finances without problems" . 4) Ananas . 5) E/AS .
Financial accounting package (India & Asia)	???	Kalculate . [Proprietary]
Personal finances manager	MSMoney, Quicken	1) GNUcash . 2) GnoFin . 3) Kmymoney .
Scheduling the business plan, management enterprise	MS Project, Project Expert 7	MrProject .

Automation of the enterprise (Russia)	"1C: Enterprise"	1) Keeper . [Proprietary] 2) Oblik . [Proprietary] 3) IceB . 4) Compiere .
Receiving the quotations, news, building graphs and analysing of the financial market.	Omega Research Trade Station 2000	???
Software for e-commerce and web business	Weblogic/Websphere/iPlanet	1) Weblogic for Linux . [Proprietary] 2) JBoss .
6) Games.		
Places where you can find, buy and download games	Anywhere you want :).	The Linux Game Tome (happypenguin.org) LinuxGames.com Kde Games Linux 3D Linux Game Publishing
The Linux Game List	-	http://www.icculus.org/lgfaq/gamelist.php
Games for Windows, that can be run under WineX	-	This is a search form , and this is a full list of games (very big!).
-	Tetris	1) XWelltris .
-	Standard Windows games	Kdegames .
-	Mines	1) KMines . 2) Perlmines .
-	Civilization	FreeCiv .
-	Civilization: Call to Power	Civilization: Call to Power for Linux.
-	Sid Meyer Alpha Centauri	Sid Meyer Alpha Centauri for Linux.
-	Sim City 3000	Sim City 3000 for Linux.
-	Command&Conquer	FreeCNC .
-	Warcraft 2, Starcraft (?)	FreeCraft .
-	(Win)Digger	1) Digger . 2) XDigger .
-	Arcanoid	Lbreakout .
-	Quake 1, 2, 3	Quake 1, 2, 3 for Linux .
-	CounterStrike	CounterStrike for Linux (???)
-	Urban Terror	Urban Terror for Linux.
-	DOOM	1) Jdoom . 2) Zdoom . 3) DOOM Legacy . 4) LxDoom .

		5) PrBoom .
-	Return to Castle Wolfenstein	Return to Castle Wolfenstein for Linux .
-	Descent	Descent for Linux.
-	Never Winter Nights	Never Winter Nights for Linux.
-	Unreal Tournament / Unreal Tournament 2003	Unreal Tournament / Unreal Tournament 2003 for Linux.
-	Soldier Of Fortune	Soldier Of Fortune for Linux.
-	Tribes 2	Tribes 2 for Linux.
-	Worms	Nil.
-	Lines	GtkBalls .
Sony PlayStation emulator	ePSXe for Windows	ePSXe .
Racing	Need For Speed	Tux Racer :).
Arkanoid	Zball, ...	Lbreakout2.
7) Programming and development.		
IDE	Microsoft VisualStudio .net	1) CodeForge . 2) Kdevelop + Qt3 Designer . 3) Glade + Motor or + xwpe or + any text editor. 4) Emacs, XEmacs.
Visual C++ IDE	Borland C++ Builder, MS Visual C	1) Anjuta + Glade + Devhelp. 2) KDE Studio. 3) Dev-C++. 4) Kylix . [Proprietary] (Kylix Personal Edition is free). 5) vtkBuilder . 6) foxBuilder . 7) wxDesigner .
C++ IDE	Borland Turbo C++ 3.0 for DOS	GCC + Motor .
Object Pascal IDE	Delphi	1) Kylix . [Proprietary] (Kylix Personal Edition is free). 2) Lazarus + FPC .
Pascal	Pascal, BP	1) Freepascal . 2) GNU Pascal. 3) RShell (in style of Borland Pascal 7.0)
Basic	Basic	1) Hbasic . 2) X-basic. 3) Yabasic . 4) SmallBASIC .

Prolog	VisualProlog	GNU Prolog.
Assembler	TASM, MASM	1) NASM. 2) FLAT Assembler.
Reverse Engineering	SoftIce	The source code is open :)
Debugger	-	1) gdb. 2) ddd (frontend for gdb).
WYSIWYG html editor	Macromedia Dreamweaver, Frontpage	1) Mozilla Composer. 2) Openoffice HTML editor. 3) Amaya. 4) GINF (Ginf is not Frontpage) 5) IBM's WebSphere Builder (???)
Powerful editor for site creating, contains set of samples and can be complemented with every sort and kind of plug-ins	Dreamweaver Ultradev	???
HTML editor	HomeSite	1) Quanta Plus. 2) Bluefish. 3) WebMaker. 4) Screem. 5) Toppage. 6) WebDesigner. 7) ScriptEditor.
HTML editor	Arachnofilia	Arachnofilia.
Visual Java development	JBuilder	1) Jbuilder for Linux. 2) NetBeans. 3) Eclipse. 4) Forte.
IDE for Oracle Database development	T.O.A.D., SQL Navigator, PL/SQL Developer	Tora.
CASE-facility for UML	???	1) Umbrello UML Modeller. 2) Dia+Dia2Code. 3) PoceidonCE (community edition). 4) ArgoUML.
Top-level CASE system	Rational Rose.	Rational Rose for Linux.
HEX-editor	Hiew	1) Biew. 2) KHexEdit.
Clipper compiler and preprocessor	CA-Clipper	Clip.
Platform in dot-net style	M\$.Net	Mono.

Work with CVS	WinCVS, TortoiseCVS, cvs for Windows	1) cvs (console). 2) Cervisia (KDE).
IDE for Interbase/Firebird development	IBExpert	???
8) Server software.		
Web-server	Internet Information Server, Apache for Windows	1) Apache . 2) Xitami. 3) thttp. 4) tux.
FTP-server	Internet Information Server, ServU	1) wu-ftp.d. 2) proftpd . 3) vsftpd. 4) ftp.
Language for Web-development	PHP for Windows	PHP.
Language for Web-development	Perl for Windows	Perl.
Language for Web-development	ASP	ASP module for Apache.
Database	MS SQL, MySQL for Windows	1) Sybase Adaptive Server Enterprise . [Proprietary] 2) PostgreSQL . The most advanced open source database. 3) MySQL . The most popular open source database. 4) mSQL. 5) SAP DB .
Database	IBM DB2	IBM DB2 for Linux . [Proprietary]
Database	Oracle for Windows	1) Oracle . [Proprietary] 2) Linter . (cyrillic)
Database	Informix	Informix
Database	Borland Interbase	FireBird
Email server	MDaemon , Hamster	1) Sendmail. 2) Qmail . 3) Postfix.
Email + Groupware	Microsoft Exchange	1) CommuniGate Pro . [Proprietary] 2) Bynari's Insight GroupWare Suite . [Proprietary] 3) Samsung Contact . [Proprietary] 4) Teamware Office for Linux . [Proprietary] 5) Novell Netmail . [Proprietary]

		<p>6) Amphora. Based on zope and qmail.</p> <p>7) Tutos. Based on Linux / Apache / PHP / Mysql / Sendmail.</p> <p>8) Kroupware. The project from the KDE PIM developers, which is being financed by the government of Germany.</p> <p>9) SuSe Linux Openexchange Server. [Proprietary]</p> <p>10) phpGroupWare. [Proprietary]</p>
Mail filter / spam killer	???	<p>1) Procmail.</p> <p>2) Mailfilter.</p>
Mail downloader	???	1) Fetchmail.
???	Lotus Domino for Windows	Lotus Domino for Linux . [Proprietary]
Server / router on one diskette.	Impossible	<p>As much as you want :).</p> <p>1) muLinux.</p> <p>2) Dachstein (firewall / dhcp).</p> <p>3) Serverdisk (http / ftp).</p>
Proxy server	MS Proxy Server, WinGate	<p>1) Squid.</p> <p>2) Paco.</p> <p>3) Privoxy.</p>
Server for supporting Java Servlets X JSP, can work with Apache	Tomcat	Tomcat .
9) Scientific and special programs.		
Math system in MathCad style	Mathcad	Gap.
Math system in Matlab style	Matlab	<p>1) Matlab for Linux. [FTP]</p> <p>2) Octave + Gnuplot.</p> <p>3) Scilab.</p> <p>4) R.</p> <p>5) Yorick.</p> <p>6) rlab.</p> <p>7) Yacas - small and flexible general-purpose algebraic computer system.</p>
Math system in Mathematica style	Mathematica	<p>1) Mathematica for Linux.</p> <p>2) Maxima.</p> <p>3) MuPad.</p>
Math system in Maple style	Maple	<p>1) Maple for Linux.</p> <p>2) Maxima.</p> <p>3) MuPad.</p>
Extended equation editor	Mathtype, MS Equation Editor	OpenOffice Math .
Programs for three-	SolidWorks, ...	ProEngineer Linux .

dimensional modeling		
ditto	CATIA for Windows	CATIA. It was designed under Unix, and from version 4 (2000) it was ported under Windows (not too successfully).
ditto	SolidEdge for Windows	SolidEdge (part of more powerful package Unigraphics).
Engineering	ANSYS for Windows	ANSYS.
CAD/CAM/CAE	Autocad, Microstation	1) Linuxcad . [Proprietary, ~100\$] 2) Varicad . [Proprietary]
CAD/CAM/CAE, simplified	ArchiCAD	Qcad .
Desktop Publishing Systems	Adobe PageMaker, QuarkXPress	???
Small desktop publishing systems	MS Publisher	1) Scribus - Desktop Publishing for Linux . 2) KWord .
Diagram and chart designer	Microsoft Visio	1) Kivio (Koffice) . 2) Dia . 3) KChart. 4) xfig . 5) Tgif + dotty. 6) Tulip.
Geographic image processing software	Erdas Imagine, ER Mapper , ENVI	ENVI for Linux.
GIS (Geographical information system)	ArcView	1) Grass . 2) Quantum GIS .
Vektorizator of raster maps	MapEdit, Easy Trace	???
Advanced text processing system in TeX style?	MikTeX, emTeX (DOS)	1) TeX. 2) TeTeX / LaTeX 3) Lyx (WYSIWYM). 4) Kile.
Convenient, functional and user-friendly TeX-files / dvi-files editor.	WinEdt	1) Ktexmaker2. 2) Tk LaTeX Editor . 3) Texmacs.
Statistical analysis	Statistica	"Probability and Statistics Utilities for Linux users"
Emulation of the circuit	Electronic Workbench	1) Geda. 2) Oregano. [Proprietary]
Downloader and player for Olympus dictophone	Olympus DSS Player	???

Market analysis	MetaStock	???
Electronics scheme design	PCAD	1) Eagle . 2) Geda .
The oscilloscope emulation	Winoscillo	Xoscope.
Measurement of the temperature and voltages on motherboard	MBMonitor, PCAlert	KHealthCare (KDE).
S.M.A.R.T-attributes and temperature of the hard disk	Come on CD with mainboard, Active SMART	1) smartctl . 2) Hddtemp-0.3 . 3) IDEload-0.2 . 4) Smartsuite-2.1 . 5) Smartmontools . 6) Ide-smart . 7) Smartsuite .
Memory testing	SiSoft SANDRA	Memtest86 .
Program for watching temperatures, fanspeeds, etc	SiSoft SANDRA, SiSoft SAMANTHA	1) Ksensors . 2) Imsensors .
HDD testing / benchmarking	SiSoft SANDRA, SiSoft SAMANTHA	1) hdparm . 2) Bonnie++ . 3) Iozone . 4) Dbench .
Video testing / benchmarking	Final Reality	1) X11perf . 2) Viewperf .
10) Other / Humour :) 		
System, running from CD without installing (Live CD)	Impossible	1) Knoppix . 2) Cool Linux . 3) Blin . 4) DemoLinux . 5) DyneBolic . 6) Gentoo (live CD with UT2003) . 7) Lonix . 8) Freedom Linux . 9) Virtual Linux .
Boot rescue/tools diskette	Windows system diskette	1) Linux system diskette. 2) Tomsrftbt . 3) BanShee Linux .
File systems mount	ext2fs (driver), explore2fs (program) - ext2/3 under Windows	Linux-NTFS . (driver for NTFS partitions mounting)
Sharing data/files	Windows shares	Samba.

???	Windows Domain, Active Directory	1) Samba. 2) Ldap. 3) yp.
Viewing Windows-network	Network neighborhood	1) Samba. 2) KDE Lan Browser, lisa 3) LinNeighborhood. 4) xSMBrowser .
Installing software and uninstalling	InstallShield, WISE, GhostInstaller, Microsoft Installer - the analog of rpm	1) Rpm & frontends. 2) Urpmi. 3) Apt & frontends (synaptic, ...).
Installing software from source and uninstalling	-	1) make install, make uninstall 2) CheckInstall . 3) Sinstall .
System update	Windows Update	1) Ximian Red Carpet . 2) Red Hat Network . 3) MandrakeOnline . 4) SuSE YaST Online Update . 5) Caldera Volution Online . 6) Apt.
Certification	MCSD, MCT, MCSE	1) Red Hat Certification . 2) Sair Linux and GNU Certification . 3) Linux Professional Institute Certification (LPIC) . 4) Linux+ . 5) Prometric . 6) VUE .
ZX Spectrum emulator	UnrealSpeccy	1) Xzx . 2) Glukalka . 3) Fuse .
Virtual machines	VMWare for Windows	1) VMWare for Linux . [Proprietary] 2) Win4Lin . 2) FreeMWare . 3) Bochs .
Work with Palm	Palm Desktop	1) Kpilot . 2) Jpilot .
Allows to carry on Palm .html files	iSilo	1) iSilo for Linux . 2) Plucker .
Optimization	WinBoost, TweakXP, Customizer XP, X-Setup, PowerToys, etc	Powertweak-Linux .
IP-telephony	???	Openh323 .
Bible	On-Line Bible, The SWORD	1) BibleTime (KDE) 2) Gnomesword (Gnome)
Icons on desktop	Explorer	1) Desktop File Manager.

		2) Idesk .
Screensaver	Desktop properties	xset.
Place for keeping "removed" files	Trash	Trash Can .
Checking the hard disk	Scandisk	fsck -check or reiserfsck -check.
GUI of the system	Windows Explorer	Kde, Gnome, IceWM, Windowmaker, Blackbox, Fluxbox, ...
Windows 9x GUI	Windows 9x	Macrosoft Desktop Environment .
Windows XP GUI	Windows XP	XPde .
Flavors of the sytem	9x, NT, XP	RedHat, Mandrake, ALT, Knoppix, Debian, SuSE, Gentoo, Slackware, ...
Tactics	FUD (fear, uncertainty, doubt)	Open Source!
Source code of the kernel freely available	No	Of course :)
Command interpreter	command.com :), cmd.exe, Windows Scripting Host	1) Bash. 2) Csh. 3) Zsh. 4) Ash. 5) Tcsh.
Linux console emulator	1) CygWin . 2) MKS Toolkit. 3) Bash for Windows.	-
X Window System (XFree) emulator	XFree OHJ CygWin	-
Windows emulator	-	1) Wine . 2) Transgaming WineX . 3) Crossover Office .
Free of charge operating system	Microsoft Windows. (Imagine yourself that in Russia there are 95% of users having a pirate copy of Windows :).	Linux - the Free operating system!!
-	Nimda	Slapper .
-	Wincih, klez, etc	No analogs
Backdoors and hidden keys	Decide it yourself :).	No analogs
Easter eggs , undocumented possibilities	Logo with Windows developers, Doom in Excel 95, 3D-racing in Excel 2000, etc, etc...	No analogs

The magazines	Windows Magazine	1) Linux Journal . 2) Linux Gazette . 3) Linux magazine, Linux pratico (Italy)
-	Blue Screen Of Death (BSOD)	1) Kernel panic. 2) Screensaver "bsod" :).
Whom it is necessary to curse for bugs and defects of the system	M\$, Bill Gates personally	1) Developers of the distribution. 2) All the Linux people and Linus Torvalds personally :). 3) Yourself and your own /dev/hands :)).
-	M\$.com	GNU.org , FSF.org
-	Windows.com	Linux.org
-	Bill Gates, "Road ahead"	Linus Torvalds, "Just for fun" :).
-	Bill Gates, "Business @ the speed of thought"	Richard M. Stallman, "The right to read".

History:

25.01.03: Big cleaning of the HTML code! The table is now ~10% smaller :). Many links added:

- 1) [The Linux Game List](#). (at the moment of adding this link there was about 250 games, and about 175 of them - under free license).
- 2) **Games for Windows, that can be run under WineX** (the information from the official Transgaming site). This is a [search form](#), and this is a [full list of games](#) (very big!).
- 3) [The biggest catalog of Windows apps, running under Wine](#). The database of Codeweavers contains information about ~1000 applications, and the catalog have a structure and a navigation system.

3.01.03: New engine for the table is under construction. Thanks to Anne & Theo Julienne.

29.12.02: I've got many positive responses, corrections and additions (I even do not have time to reply :). Thanks for your help!

28.12.02: The release of the English version. Announce was posted to [Newchix mail list](#). (The only English mailing list I'm subscribed at :).

(c) [Valery V. Kachurov](#), [Nesov Artem](#), 2002-2003

POSTE PRINCIPAL	LOGICIELS	PRIX UNITAIRE	ANNEE 1			ANNEE 2			ANNEE 3		
			Qté	Linux	Windows	Qté	Linux	Windows	Qté	Linux	Windows
SYTEME D'EXPLOITATION											
	WINDOWS 2000 ADV. SERVER + 25 Clients	1.999 €	2		3.998 €	0		- €	0		- €
	WINDOWS 2000 SERVER (Licence + 20 Clients)	449 €	1		449 €	2		898 €	0		- €
	CLIENT WINDOWS 2000 SERVER	25 €	500		12.500 €	0		- €	0		- €
	LINUX RED HAT Distribution	150 €	1	150 €		0	- €		0	- €	
	WINDOWS 2000 WORKSTATION	163 €	0		- €	0		- €	300		48.900 €
	SYSTEME D'EXPLOITATION TOTAL			150 €	16.947 €		- €	898 €		- €	48.900 €
BASE DE DONNEES											
	MS SQL SERVER 2000 ENTR + 10 LIC Client	2.329 €	0		- €	2		4.658 €	0		- €
	MYSQL DATABASE SERVER	220 €	1	220 €		0	- €		0	- €	
	BASE DE DONNEES TOTAL			220 €	- €		- €	4.658 €		- €	- €
SECURITE											
	ANTIVIRUS POUR WINDOWS	- €	0		- €	0		- €	0		- €
	ANTIVIRUS POUR LINUX	572 €	1	572 €		0	- €		0	- €	
	FIREWALL - Maintenance Annuelle	7.700 €	1		7.700 €	1		7.700 €	0		- €
	FIREWALL - IPTABLE	- €	0	- €		0	- €		0	- €	
	ARCserve 9.0 (Backup)	590 €	0		- €	0		- €	0		- €
	DISK ARCHIVE FOR LINUX	- €	0	- €		1	- €		0	- €	
	ARCserve FOR LINUX	600 €	0	- €		1	600 €		0	- €	
	SECURITE TOTAL			572 €	7.700 €		600 €	7.700 €		- €	- €
MAIL / INTERNET											
	MS EXCHANGE 2000 SERVER ENTR	7.250 €			- €	1		7.250 €	0		- €
	SENDMAIL / POSTFIX	- €		- €		1	- €		0	- €	
	MAIL / INTERNET TOTAL			- €	- €		- €	7.250 €		- €	- €
LOGICIELS METIERS											
	MS VISUAL STUDIO.NET + MSDN	1.227 €	0		- €	3		3.681 €	0		- €
	BORLAND KYLIX / DELPHI	2.229 €	0	- €		3	6.687 €		0	- €	
	PHOTOSHOP 7.0	229 €	0		- €	2		458 €	0		- €
	GIMP	- €	0	- €		2	- €		0	- €	
	MS OFFICE PRO 2000	309 €	0		- €	0		- €	500		154.500 €
	OPEN OFFICE	- €	0	- €		0	- €		500	- €	
	LOGICIELS METIERS TOTAL			- €	- €		6.687 €	4.139 €		- €	154.500 €
CHARGES D'EXPLOITATION											
	SALAIRES	- €	0			0			0		
	PRESTATION W2000 - Migration	800 €	3		2.400 €	3		2.400 €	0		- €
	PRESTATION LINUX - Migration	735 €	3	2.205 €		3	2.205 €		0	- €	
	PRESTATION W2000 - Support	800 €	3		2.400 €	0		- €	0		- €
	PRESTATION LINUX - Support	735 €	3	2.205 €		0	- €		0	- €	
	FORMATIONS WINDOWS 2000 (Jours Homme)	300 €	5		1.500 €	5		1.500 €	0		- €
	FORMATIONS LINUX (Jours Homme)	375 €	15	5.625 €		10	3.750 €		0	- €	
	FORMATIONS OFFICE 2000 (Jours Homme)	300 €	0		- €	0		- €	100		30.000 €
	FORMATIONS OPEN OFFICE (Jours Homme)	375 €	0	- €		0	- €		400	150.000 €	
	CHARGES D'EXPLOITATION TOTAL			10.035 €	6.300 €		5.955 €	3.900 €		150.000 €	30.000 €
TOTAL GENERAL				10.977 €	30.947 €		13.242 €	28.545 €		150.000 €	233.400 €