

Contents

1	Logiciels libres, une introduction	1
1.1	Logiciels libres, une introduction	1
1.2	Qu'est-ce qu'un logiciel libre ?	1
1.3	La FSF et le projet GNU	2
1.4	GNU/Linux, une ralit	2
1.5	Pourquoi le logiciel libre est t'il meilleur ?	3
1.5.1	Pour le technicien	3
1.5.2	Pour l'utilisateur	3
1.5.3	Pour les entreprises	4
1.6	Brevets sur les logiciels	4
1.6.1	Effets dangereux des brevets sur les logiciels	4
1.7	Pour en savoir plus	5

1 Logiciels libres, une introduction

1.1 Logiciels libres, une introduction

Qu'est ce qu'un logiciel ? Pour comprendre le concept de logiciel libre, vous devez d'abord comprendre ce qu'est un logiciel (ou encore programme ou application). Du point de vue de l'utilisateur, un logiciel est une application qui rpond l'un de ses besoins (traitement de textes, programme de dessin, jeu, ...). C'est une suite de petites instructions invisibles pour l'utilisateur, qui forment un tout cohrent. Ces logiciels ont besoin d'un systme d'exploitation pour fonctionner. Le systme d'exploitation permet d'accder aux ressources de la machine (lecteur de disquette, cran, clavier, ...). C'est aussi le systme d'exploitation qui se charge d'excuter le logiciel. Les instructions d'un logiciel sont crites dans un langage que l'ordinateur peut comprendre, le langage machine (ou langage binaire). En revanche, il est trs difficile (voir impossible) un humain de comprendre ce langage (c'est de plus interdit dans de nombreux pays). La manire la plus courante de crer un logiciel est de l'crire dans un langage informatique comprhensible par des humains, et ensuite de le traduire vers le langage binaire. Cette traduction est effectuee par un logiciel appel compilateur. Le logiciel dans sa forme comprhensible est appel code source, et dans sa version en langage machine binaire.

1.2 Qu'est-ce qu'un logiciel libre ?

Imaginez que vous vous trouviez dans un restaurant et que vous mangiez un excellent plat. Peut-tre aurez vous l'envie de le cuisiner chez vous pour vos amis ? C'est impossible, car vous n'avez pas la recette du plat. Vous pouvez toujours le manger dans le restaurant, mais mme si vous connaissez le got, vous ne savez comment le reproduire. En informatique, c'est la mme chose avec un logiciel. Le code source est la recette, le binaire est le plat cuisin. La plupart des logiciels sont distribus sans leur code source, et il est interdit d'essayer de

comprendre leur fonctionnement. Il est interdit de les partager avec vos amis, et il est interdit d'essayer de les modifier pour les adapter à vos besoins. Un logiciel libre vous garantit plusieurs libertés : la liberté de copier le logiciel pour vous ou vos amis la liberté de comprendre son fonctionnement si vous le désirez la liberté de le modifier et de distribuer vos modifications

1.3 La FSF et le projet GNU

Richard Stallman, considéré par tous comme le père des logiciels libres a commencé à penser au logiciel libre lorsqu'il travaillait au laboratoire d'intelligence artificielle au MIT. Le laboratoire possédait une imprimante qui tombait souvent en panne, mais comme les chercheurs avaient le code source du pilote le pilote est le logiciel qui sert à accéder et utiliser l'imprimante. De l'imprimante, ils avaient modifié le programme pour que l'imprimante leur envoie un signal chaque panne. Un beau jour, le laboratoire achète une nouvelle imprimante de marque Xerox, plus fiable. En revanche, le pilote de l'imprimante n'est pas fourni à la livraison. Richard Stallman entend parler plus tard d'un laboratoire qui possède les sources de ce pilote. S'y rendant, on lui répond que le laboratoire s'était engagé à ne pas diffuser les sources du pilote. Ressentant ce refus comme une agression, Richard Stallman prend alors conscience du danger de la logique propriétaire. Richard Stallman décide alors de fonder la Free Software Foundation. Conscient qu'il est impossible d'utiliser un ordinateur sans utiliser de logiciels propriétaires, il initialise le premier projet de la fondation, le projet GNU (GNU est un jeu de mots récursif signifiant GNU's Not Unix). Ce projet vise à concevoir un système d'exploitation complet et entièrement libre. Ce système sera compatible avec UNIX, mais sera différent. Aujourd'hui ce système existe, et s'appelle GNU/Linux. Pour valider ce système, une base légale est nécessaire. Cette base légale est la GNU GPL, pour GNU General Public Licence. La GNU GPL est la licence des logiciels libres par excellence. Elle détermine des conditions de distribution qui garantissent les libertés de l'utilisateur. Un programme protégé par la GPL est libre, mais la GPL impose aussi que tout travail dérivé de ce logiciel reste libre.

1.4 GNU/Linux, une réalité

Le projet GNU a commencé de zéro, et ses premiers développements avaient pour but de créer un environnement de développement utilisable. Les premiers efforts se sont donc très logiquement portés sur l'élaboration d'un compilateur (gcc), d'un éditeur de textes (Emacs), ainsi que d'un débogueur (gdb). Un noyau (Hurd) était aussi prévu, mais son élaboration prend plus de temps que prévu. En 1991, Linus Torvalds commence à créer son noyau, qui connaîtra plus tard le succès que l'on sait. Le projet GNU l'adopte en attendant que Hurd soit opérationnel, et le noyau Linux devient alors composant du système GNU/Linux. Le système GNU/Linux est réputé pour sa fiabilité et sa robustesse. Ceci est dû en partie à la liberté des logiciels qui le composent, l'accès aux sources permettant de corriger très facilement et rapidement une erreur de programmation. Le système GNU/Linux est fortement POSIX (POSIX est un ensemble de normes définissant un système UNIX idéal), ce

qui le rend très similaire à la plupart des UNIX propriétaires existants. Le noyau Linux est multi-tâches, multi-utilisateurs, et intègre la plupart des technologies les plus récentes (SMP, clustering, RAID, ...). GNU/Linux est aujourd'hui utilisé aussi bien par les entreprises que par les utilisateurs finaux. De nombreuses applications sont disponibles pour l'utilisateur non informaticien. Par exemple, l'environnement graphique Gnome permet d'utiliser son ordinateur sans utiliser la ligne de commande. Gimp est un logiciel de traitement d'images très puissant. Gnumeric est un tableur intégré au projet Gnome. Ce ne sont que des exemples, de nombreuses applications peuvent rendre d'immenses services à l'utilisateur. La sécurité et la fiabilité de ce système sont un argument supplémentaire pour l'utiliser (les virus n'existent pas). Plusieurs distributions existent, et l'installation du système GNU/Linux est désormais facile. Pour l'entreprise, les champs d'application sont vastes. La robustesse du système couple l'excellence des applications orientées réseau (Apache par exemple) en fait un système de choix pour les serveurs (le couple GNU/Linux et Apache est le plus utilisé au monde en matière de serveurs web). Les outils de développement (en particuliers les outils GNU : Emacs, gcc, cvs, ...) font de GNU/Linux une plate-forme de développement puissante et agréable à utiliser. En dehors de ces considérations techniques, l'utilisation de logiciels libres affranchit de plus l'entreprise de la main-mise d'un éditeur.

1.5 Pourquoi le logiciel libre est-il meilleur ?

1.5.1 Pour le technicien

Par définition, les sources d'un logiciel libre sont publiques et librement utilisables. Pour le technicien, c'est d'un intérêt incontestable. D'abord, il peut se permettre d'utiliser un code source codé par un autre programmeur sans crainte de poursuites. Il n'a pas besoin de "réinventer la roue". De plus, les erreurs d'un logiciel libre seront plus rapidement corrigées. L'ouverture des sources permet aux informaticiens (ou tout simplement aux utilisateurs ayant quelques connaissances) de corriger eux-mêmes les erreurs qu'ils rencontrent dans les programmes qu'ils utilisent. Les corrections seront intégrées dans la version suivante du logiciel. Pour les étudiants en informatique ou même les professionnels, l'accès aux sources permet un apprentissage des plus profonds. Pour beaucoup de professionnels, "la meilleure documentation qui puisse exister, ce sont les sources elles-mêmes".

1.5.2 Pour l'utilisateur

L'utilisateur ne bénéficie pas directement de l'accès aux sources. La plupart des automobilistes ne connaissent pas le fonctionnement d'un moteur, c'est la même chose en informatique. Maintenant, imaginons un monde où 100 En informatique, on a plus ou moins le même concept, mais là vous avez une voiture neuve qui profite de toutes les technologies développées dans des petits garages indépendants par des artisans habiles, et dont les meilleures sont rassemblées dans votre véhicule. Vous êtes sûr de la qualité tout comme de la sécurité. Dans le cas d'un vice de conception, vous êtes certain que quelqu'un aura détecté l'erreur avant vous et l'aura corrigée.

1.5.3 Pour les entreprises

Comme pour les utilisateurs, les entreprises bnficient de la qualit du logiciel. Mais en plus, le modle de dveloppement des logiciels libres est un avantage lui tout seul. Il permet de corriger trs rapidement une erreur de programmation dont les effets sont critiques pour l'entreprise, et aucune logique marchande ne conditionne les sorties des diffrentes versions des logiciels libres. Si une entreprise dcide de jouer le jeu et de dvelopper des logiciels libres, elle bnficie du support de toute la communaut du logiciel libre, tout comme elle garantit la prnit de son programme.

1.6 Brevets sur les logiciels

D'aprs la directive europenne du 14 mai 1991 concernant la protection juridique des programmes d'ordinateurs, ceux-ci sont protgs par le droit d'auteur, en tant qu'oeuvre littraire au sens de la convention de Berne pour la protection des oeuvres littraires et artistiques. La directive prcise que seule l'expression d'un programme d'ordinateur est protge et que les ides et les principes qui sont la base de la logique, des algorithmes et des langages de programmation ne sont pas protgs par le droit d'auteur. De mme, les ides et principes qui sont la base de ses interfaces ne sont pas protgs par le droit d'auteur. L'unique critre pour dterminer si un programme peut bnficier d'une protection est son originalit, en ce sens que ce programme est la cration intellectuelle propre son auteur. Un projet de directive europenne propose de remettre en cause la non brevetabilit des programmes d'ordinateurs garantie par la directive europenne de 1991 et la Convention de Munich.

1.6.1 Effets dangereux des brevets sur les logiciels

Quelques effets pervers des brevets sur les logiciels :

l'innovation Le brevet permet de restreindre l'utilisation d'une innovation pendant vingt ans. Or, dans le cas des programmes d'ordinateurs, les brevets rsultent le plus souvent de la dcouverte ou de la simple application d'une propriit mathmatique ou d'un algorithme, qui fait partie du patrimoine de l'humanit. La plupart du temps, les dcouvertes effectues par un programmeur ne sont que l'adaptation d'ides dj existantes un problme bien dfini. Aux Etats-Unis, plus de 50 000 brevets concernant l'informatique sont dposs. Dans un tel contexte, il est impossible un programmeur d'accomplir son mtier sans transgresser un brevet son insu. Les brevets informatiques couvrent des domaines divers : les lments les plus simples de l'interface d'un programme, des algorithmes qu'un tudiant de premire anne dans une cole informatique pourrait inventer nouveau, et mme l'utilisation des couleurs. Il faudrait mettre un ou deux avocats derrire chaque programmeur, et, dans la plupart des cas, il serait impossible d'crire un programme fonctionnel, car les algorithmes les plus efficaces ou vidents sont protgs par des brevets. De grands groupes dtiennent la majorit des brevets et

s'en servent pour bloquer leurs concurrents, le brevet tant ici une arme offensive et non plus défensive. Seules les très grandes compagnies de logiciels auraient le droit à la cration. Ce serait la fin du droit de chacun d'innover et de créer.

Pour l'informatique libre L'informatique libre est forcément la plus touchée par la brevetabilité des programmes d'ordinateurs. Un logiciel libre utilise par nature des techniques et des inventions découvertes par autrui, tout comme il autorise l'utilisation future des techniques développées par son programmeur. Un programmeur de logiciel libre ne peut pas non plus se permettre d'embaucher des avocats pour se protéger dans le cas probable où il enfreindrait un brevet par inadvertance. De plus, l'informatique libre posant comme principe le partage et l'utilisation mutuelle des connaissances, pour le bien de la communauté, ne peut accepter la brevetabilité des programmes d'ordinateurs. Les idées appartiennent à l'humanité.

1.7 Pour en savoir plus

Si vous voulez en savoir plus, n'hésitez pas à faire un tour sur les sites suivants : <http://www.gnu.org>, le site web du projet GNU et de la FSF. C'est le site web de référence lorsqu'on parle de logiciel libre. Il contient la philosophie GNU qui explique pourquoi il faut du logiciel libre et de nombreuses autres choses comme la GPL. <http://www.april.org>, le site de l'association APRIL (Association pour la Promotion et la Recherche en Informatique Libre). Le web contient des informations sur l'association, sur leurs travaux en cours, sur les manifestations organisées par l'association ainsi que sur la philosophie de l'informatique libre. <http://www.debian.org>, le site de la distribution de GNU/Linux la plus libre. <http://www.linux.org>, le site de départ pour toute information ou recherche de logiciels concernant le système GNU/Linux. <http://www.linux-france.org>, site d'informations (avec de nombreuses documentations) en français autour de GNU/Linux. <http://www.linux-center.fr>, site d'actualités en français autour de GNU/Linux. <http://lpf.ai.mit.edu>, le site de la League For Programming Freedom, qui s'intéresse au problème des brevets sur les logiciels. Généré le 29/08/1999 par Benjamin Drieu. Sauf mention contraire indiquée plus haut, le présent document est soumis aux conditions d'exploitation suivantes : Copyright 1999 APRIL. Ce document peut être reproduit par n'importe quel moyen que ce soit, pourvu qu'aucune modification ne soit effectuée et que cette notice soit préservée.

Article LEE00 : Leete, Gurdy; Finkelstein, Ellen; Leete, Mary, OpenOffice.org for Dummies, 2003