

Le Cycle de Vie du Logiciel Libre

Étudiants: Badredine Ladjemi, Bruno Miralles, Vincent Berthier
Tuteur: M. Christian Métairie

22 mai 2005

Table des matières

1	Définition d'un logiciel libre	4
2	Comparaison des différents types de logiciels et licences qui les définissent	4
2.1	Logiciel du domaine public	4
2.2	Logiciel Libre non copy-lefté	5
2.3	Logiciel Libre sous copyleft	5
2.4	Logiciel Open Source	5
2.5	Logiciel Propriétaire	5
2.6	Freeware (ou graticiel)	6
2.7	Shareware (ou partagiciel)	6
2.8	Logiciel privé	6
3	Le développeur de logiciels libres	6
3.1	Portrait robot	6
3.2	Pourquoi développer un logiciel ?	7
3.3	Pourquoi développer en libre ?	7
3.3.1	absence d'intérêt commercial	7
3.3.2	intérêt pédagogique	8
3.3.3	reconnaissance des pairs	8
3.3.4	amélioration de la technique	8
3.3.5	choix éthique	9
4	Le modèle de développement du logiciel libre	9
4.1	Bâtie sur Internet	9
4.1.1	les listes de diffusion	9
4.1.2	la page web	9
4.1.3	les Gestionnaires de Code Source	9
4.1.4	autres...	10
4.2	Organisation entre développeurs	10
4.2.1	le pouvoir entre les mains des développeurs	10
4.2.2	remise en cause de la méthodologie traditionnelle	11
4.2.3	théorie du « Bazaar »	11
5	Les Succès...	12
5.1	Apache	12
5.2	PHP, MySQL	13
5.3	Mozilla Firefox	14
5.4	OpenOffice.Org	14

5.5	Autres...	15
-----	-----------	----

Introduction

Le logiciel libre est médiatisé depuis quelque temps mais peu de gens comprennent ce qu'il signifie. Il est ainsi nécessaire de comprendre ce qu'est un logiciel libre, de bien savoir faire la différence entre les différents types de logiciels libres ou non-libres et comprendre les nuances entre logiciel libre. Comprendre quelle est la motivation de la création d'un logiciel libre et l'originalité de ce modèle est également vital à la compréhension et à la dissipation des mythes qui entourent le monde.

1 Définition d'un logiciel libre

Le logiciel libre est un logiciel qui est fourni avec l'autorisation pour quiconque de l'utiliser, de le copier, et de le distribuer, soit sous une forme conforme à l'original, soit avec des modifications (gratuitement ou contre un certain montant). Ce qui signifie en particulier que son code source doit être disponible. En anglais le terme « Free Software » est ambigu, il peut aussi bien signifier un logiciel libre qu'un logiciel gratuit (freeware). C'est pourquoi insister sur la liberté est à ce point important. Un logiciel libre n'appartient pas forcément à tout le monde ni n'est forcément gratuit comme on pourrait le croire.

2 Comparaison des différents types de logiciels et licences qui les définissent

Un logiciel peut être rendu libre de nombreuses manières différentes, un grand nombre de détails sont variables en fonction des licences, mais tout en gardant au logiciel son caractère libre. Description rapide des différents types de logiciels :

2.1 Logiciel du domaine public

C'est un logiciel non soumis aux droits d'auteurs. C'est un logiciel libre qui n'est pas sous «gauche d'auteur » Si le code source est dans le domaine public, certaines copies ou versions modifiées peuvent ne pas être libres. Il est possible que le logiciel soit dans le domaine public, mais que son code source ne soit pas disponible, il ne s'agit alors pas d'un logiciel libre, car le fondement du logiciel libre est la disponibilité du code source. Le terme « domaine public » est parfois utilisé de façon peu précise pour dire « libre »

ou « disponible gratuitement ». toutefois, « domaine public » est un terme qui signifie que le logiciel n'est pas « soumis aux droits d'auteur ».

2.2 Logiciel Libre non copy-lefté

La licence APACHE est une licence pour logiciels libres non «copy-leftés». Ce type de logiciel est diffusé par son auteur avec la permission de le redistribuer et de le modifier, mais aussi d'ajouter d'autres restrictions. Si un programme est libre et non « copy-lefté », alors certaines copies ou versions modifiées peuvent ne plus être libres du tout. Une société peut compiler ce programme, avec ou sans modifications, et distribuer le fichier exécutable sous forme de produit logiciel propriétaire.

2.3 Logiciel Libre sous copyleft

La licence GNU GPL (Licence Publique Générale GNU) est une licence pour logiciels libres sous « copyleft ». Un logiciel sous « copyleft » (littéralement sous gauche d'auteur) est un logiciel libre, dont les conditions de distribution interdisent aux nouveaux distributeurs d'ajouter des restrictions supplémentaires lorsqu'ils redistribuent ou modifient le logiciel. Ceci veut dire que chaque copie du logiciel, même si elle a été modifiée, doit être un logiciel libre. Le « copyleft » est un concept général ; pour l'appliquer à un programme, il faut définir les conditions de distribution. Il y a de nombreuses façons d'écrire ces conditions de distribution, donc il y a de nombreuses licences de logiciels libres sous « copyleft ». Mais dans la pratique actuelle, la plupart des logiciels libres sous « copyleft » utilisent la licence GNU GPL.

2.4 Logiciel Open Source

Le terme logiciel « open source » (littéralement à source ouvert) est utilisé par certaines personnes pour signifier plus ou moins la même chose que logiciel libre. Mais ce type de logiciel accepte des restrictions sur les licences qui l'éloignent du concept de logiciel libre.

2.5 Logiciel Propriétaire

Un logiciel propriétaire est un logiciel qui n'est pas libre. Son utilisation, sa redistribution ou sa modification sont interdites, ou exigent une autorisation spécifique, ou sont tellement restreintes qu'il n'est pas possible de le faire librement.

2.6 Freeware (ou graticiel)

Le terme « freeware » n'a pas de définition claire communément acceptée, mais elle est utilisée couramment pour des paquetages qui autorisent la redistribution mais pas la modification (et dont le code source n'est pas disponible). Ces paquetages ne sont pas des logiciels libres, il n'est donc pas conseillé de dire « freeware » pour des logiciels libres.

2.7 Shareware (ou partagiciel)

Le partagiciel est un logiciel qui s'accompagne de la permission de redistribuer des copies, mais qui mentionne que toute personne qui continue à en utiliser une copie est obligé de payer des royalties. Les sharewares ne sont pas des logiciels libres pour deux raisons :

- Le code source n'est pratiquement jamais fourni, il est donc impossible de modifier le programme.
- Il n'est pas permis d'effectuer une copie et de l'installer sans acquitter un droit de licence, même pour des individus impliqués dans des activités à but non lucratif.

2.8 Logiciel privé

C'est un logiciel qui n'est accessible uniquement pour un utilisateur (seul, groupe, ou entreprise). Aucun fichier ou code source n'est disponible aux autres utilisateurs, ce n'est donc pas un logiciel libre.

3 Le développeur de logiciels libres

3.1 Portrait robot

Selon [l'étude de l'Université de Maachstricht](#), le profil type d'un développeur de logiciel libre serait le suivant :

sexe masculin à 98%

age 26 ans, plus de 60% des interrogés ont entre 20 et 30 ans ; passé 45 ans le pourcentage de développeurs de logiciel libre est même inférieur à celui des moins de 18 ans

age de début de 16 à 28 ans, ce qui indique que la plupart des développeurs de logiciel libre sont encore des débutants dans le domaine (moins de 3 ans d'expérience)

diplôme plus de 70% possèdent un diplôme d'enseignement supérieur

profession plus de 50% de développeurs à divers échelons, 1/3 d'étudiants et seulement 10% d'universitaires

système d'exploitation préféré GNU/Linux avec plus de 90% des réponses, MS Windows se situant même en 3ème position avec 2,2%, des pourcentages totalement opposés à ceux du grand public

nombre de projets de 1 à 5 pour 70% d'entre eux

motivation apprendre et développer de nouvelles compétences et les partager pour la grande majorité, améliorer le logiciel libre et participer à cette nouvelle forme de coopération et parce que le logiciel ne devrait pas être propriétaire (par ordre d'importance et à des proportions équivalentes 1/3 chacune)

ses stars Linus Torvalds (auteur de Linux), Richard M. Stallman (dirigeant de la fondation pour le logiciel libre et auteur d'Emacs)

nationalité citoyen de l'UE à plus de 75%; l'enquête étant européenne, les 2 premiers pays sont la France (16%) suivie de l'Allemagne (12%), les USA n'arrivent que 3ème avec 10%

mobilité les USA sont le pays le plus attractif pour un développeur de logiciel libre, suivi de peu par l'Allemagne. La France arrive en dernière position.

3.2 Pourquoi développer un logiciel ?

La plupart des projets de logiciels propriétaires sont des entreprises commerciales en vue de réaliser des bénéfices. Ce n'est pas du tout le même objectif que dans le cadre du logiciel libre. En effet les motivations premières qui poussent à développer un logiciel libre sont

- Comblent un besoin des utilisateurs et ou le créent
- Comblent son propre besoin et aller plus loin.

cela confirme le dicton « Nécessité est mère d'invention ».

3.3 Pourquoi développer en libre ?

3.3.1 absence d'intérêt commercial

La plupart des logiciels répondent à des besoins certes, mais ce besoin n'est pas souvent solvable. Souvent ce genre de projets est réalisé sur le temps libre du développeur, il n'est pas souvent une manière pour le développeur de gagner sa vie. De plus pour avoir un projet rentable il faut disposer d'une importante force commerciale, et cibler une clientèle qui pourra rendre le projet

vable financièrement. Cela implique des coûts et des contraintes supplémentaires pour le développeur. Le développeur dans ce cas n'a pas d'intérêt à produire un logiciel propriétaire commercial...

3.3.2 intérêt pédagogique

Développer en libre c'est un moyen d'apprendre des autres, et de voir son travail évalué par d'autres.

3.3.3 reconnaissance des pairs

Selon le développeur fait du logiciel libre avant tout pour la reconnaissance par ses pairs, en effet ces derniers le jugent avant tout sur la qualité de son développement et l'ouverture du développeur à la communauté. Le développeur subit une sorte de contrainte sociale par l'utilisation de logiciels libres, qui le poussent à lui aussi contribuer à l'effort communautaire. Ce n'est qu'en contribuant qu'il peut se voir reconnaître en tant que membre positif de la communauté. Ce phénomène non restreint à l'informatique se nomme «egoboo», en français boost de l'égo, on comprends bien pourquoi... Selon certains il revêt une si grande importance en informatique à cause du déficit de reconnaissance sociale chez le développeur.

3.3.4 amélioration de la technique

Développer un logiciel libre, c'est avoir un logiciel potentiellement meilleur. C'est la philosophie OpenSource. Elle se base sur deux principes :

- la « loi linus » : plus le nombre d'utilisateurs est grand, plus le nombre de bugs sera réduit.
- la contrainte d'ouvrir son code entraîne un réflexe chez le développeur de présenter quelque chose de «propre», «rigoureux».

En effet, lorsqu'un logiciel libre est distribué il l'est avec ses sources, à partir de là, il est simple pour l'utilisateur, si tant est qu'il soit également développeur de corriger le bug par lui même et de proposer cette modification. Lorsque l'on parle de bug, il est fréquent qu'il s'agisse également d'ajout de fonctionnalités, de traductions etc. Les utilisateurs peuvent d'eux-mêmes répondre à leur besoins spécifiques ce qui allège la tâche du développeur principal à la résolution des problèmes majeurs. De plus l'ouverture du code source nécessite une organisation et une documentation compréhensible pour le développeur lambda. Cela impacte la qualité du code source. On a pu en voir des exemples lors de libération de code source par certaines entreprises (Sun, IBM) qui ont du réaliser un nettoyage et une restructuration de leur code source avant distribution.

3.3.5 choix éthique

Issu du monde universitaire, le monde des développeurs de logiciel libre en hérite une grande partie de ses principes. Le fait de rendre son logiciel libre reprends le paradigme humaniste du savoir disponible pour tous. La puissance morale de cet argument en font une puissante justification dans cette démarche du développeur.

4 Le modèle de développement du logiciel libre

4.1 Bâtie sur Internet

Au début fut l'Internet, lieu de prédilection des échanges inter-universités.

4.1.1 les listes de diffusion

Les listes de diffusion ou mailing-lists sont des mails envoyés à l'ensemble des membres de la liste. Chaque mail envoyé à la liste par un de ses membres est reçu par tous les membres. C'est le moyen de communication communautaire premier dans l'histoire d'Internet et de base chez les développeurs. Il permet de converser de façon différé sur un sujet bien précis. C'est un moyen très apprécié car il permet de tempérer une discussion en direct par son aspect différé qui pousse l'auteur d'un mail à mieux structurer sa pensée avant de communiquer.

4.1.2 la page web

La page web est un moyen simple de publier ses travaux de façon plaisante. Elle contient la plupart du temps un lien pour le téléchargement, l'avancement du projet, la documentation, un lien vers les autres moyens de communications, et le moyen de contacter son auteur. Elle permet aussi de garder une trace, un historique du projet, et avoir un point d'attache auxquels les utilisateurs pourront se référer.

4.1.3 les Gestionnaires de Code Source

ou SCM en anglais. Ils servent à travailler à plusieurs de façon simultanée ou différée avec sauvegarde des historiques des modifications et d'annotation des modifications. Ils sont donc un outil central dans le travail collaboratif. Surtout lorsque les personnes n'ont pas forcément la possibilité de faire des réunions de façon très fréquente pour se répartir le travail, ou lorsque le travail n'est pas vraiment réparti de façon rigoureuse. Le plus célèbre se

nomme CVS. Et ils sont un élément tellement crucial du développement de logiciel libre que lorsque *Linus Torvalds*¹ a décidé d'utiliser un logiciel non-libre (Bitkeeper) pour Linux s'en est suivit une polémique qui n'a pas cessé d'enfler et qui a entraîné la création de nombreux concurrents libres. Récemment Linus a décidé de se séparer de Bitkeeper de de développer son propre outil. Cet évènement a aussi fait la une de toute une partie du Web.

4.1.4 autres...

IRC chat orienté salle de discussion, permettant à un groupe de personnes de communiquer de façon simultanée et instantanée

devBlog site web composé d'entrées présentées de façon anti-chronologique (du plus récent au plus ancien) traitant des nouveautés et des orientations prises lors du développement d'un projet. C'est outil de plus en plus prisé pour permettre de promouvoir son projet et permettre de l'améliorer par la communication avec ses utilisateurs.

Messagerie Instantanée similaire à IRC mais centrée sur la communication à 2.

Groupe de discussion (ou USENET) forum de discussion thématique, ce média a été rendu célèbre parce qu'il a été utilisé par Linus Torvalds pour faire sa célèbre annonce de la création du projet Linux

4.2 Organisation entre développeurs

4.2.1 le pouvoir entre les mains des développeurs

Contrairement aux développements de logiciels propriétaires commerciales l'initiative ne vient pas d'en haut, et les programmeurs ne sont pas de simples exécutants. Ce sont eux qui prennent les initiatives et prennent en main la conception du logiciel. Eux qui choisissent leurs outils, et n'ont pas à se soucier d'études marketing défavorable, de marché hostile ou de concurrents monopolistiques. Cela entraîne un recentrement sur la technique, et plus notablement sur les fonctionnalités. Cela peut avoir des effets pervers conduisant à produire des logiciels obscurs d'utilisation aux néophytes. Mais souvent l'effet est positif et stimule l'innovation. Les innovations ne suscitent forcément l'admiration. Ainsi des innovations dites sous-terraines ou transparentes non forcément visibles à l'utilisateur au premier regard peuvent lui apparaître par la suite.

¹auteur initial et directeur du développement de Linux

4.2.2 remise en cause de la méthodologie traditionnelle

L'organisation spécifique du développement de logiciel libre est souvent vue comme étant une cassure avec les méthodologies traditionnelles. En effet les phases de conception et de réalisation ainsi que l'interaction avec l'utilisateur sont fortement chamboulés. Dans la méthodologie traditionnelle l'utilisateur exprime un besoin, le développeur conçoit le logiciel qui y réponds, le réalise ou le fait réaliser suivant la conception déterminée, et livre le produit final à l'utilisateur. Ce modèle est dit en V². Dans le logiciel libre par contre, le programmeur recherche le besoin à combler, conçoit de façon assez succincte, et réalise en modifiant sans cesse sa conception originelle. Il arrive donc souvent que le projet soit réécrit totalement ou en partie. Cependant l'avantage est qu'un prototype est rapidement disponible et permet de tester le concept même du logiciel. De plus une structure claire et efficace est rendue nécessaire par la mise à disposition du code source. Le programmeur se sent obligé de réaliser un code suffisamment propre pour qu'un tiers puisse le comprendre et ou l'apprécier.

4.2.3 théorie du « Bazaar »

Selon les théories de Eric S. Raymond, inspiré en cela par son expérience propre ou par le développement de logiciels libres tel Linux, il existe des lois quant aux développements de logiciel libre, ce sont celles-ci :

1. Tout bon logiciel commence par gratter un développeur là ou ça le démange.

cela signifie que c'est le développeur qui est l'initiateur d'un logiciel et non le client ou un quelconque supérieur hiérarchique

2. Les bons programmeurs savent quoi écrire. Les grands programmeurs savent quoi réécrire (et réutiliser).

la tentation de réécrire à chaque fois ce qui a déjà été réalisé est une triste réalité pour les logiciels propriétaires. Cependant avec le logiciel libre il devient aisé de partir de code source déjà existant pour bâtir un nouveau logiciel ou tout simplement l'étendre ou l'améliorer.

3. « Prévoyez d'en jeter un, car de toute manière, vous le ferez. »(Fred Brooks, "The Mythical Man-Month", chapitre 11)

signifie qu'il faut se remettre en cause, et éventuellement revoir la conception et abandonner certaines parties du code source

²http://fr.wikipedia.org/wiki/Cycle_en_V

« Bien souvent, les solutions les plus innovantes, les plus frappantes, apparaissent lorsque vous réalisez que votre approche du problème était mauvaise. »

ainsi de nombreux logiciels libres ont connu des réécritures complète

4. De l'importance d'avoir des utilisateurs

les utilisateurs sont très importants pour l'amélioration du logiciel et la plus grande corrélation entre la demande effective et les fonctionnalités offertes par le logiciel.

les écouter est donc primordial

5. Quand un programme ne vous intéresse le plus, votre dernier devoir à son égard est de le confier à un successeur compétent.

un grand avantage du logiciel libre est qu'un logiciel abandonné par son auteur est toujours sauvable, il n'est pas voué à une mort certaine. Des passations de projets sont ainsi monnaie courantes.

6. Distribuez tôt, mettez à jour souvent.

LE PARADIGME du développement de logiciel libre.

ne pas attendre de disposer d'une solution répondant à tous les besoins. Répondre aux besoins primordiaux suffit

et surtout évoluer rapidement en répondant à la demande et non pas en suivant un carnet de route tracé.

7. Étant donné un ensemble de bêta-testeurs et de co-développeurs suffisamment grand, chaque problème sera rapidement isolé, et sa solution semblera évidente à quelqu'un. Ou, moins formellement, « Étant donné suffisamment d'observateurs, tous les bogues sautent aux yeux. » C'est ce que j'appelle : « La Loi de Linus ».

8. Il est presque aussi important de savoir reconnaître les bonnes idées de vos utilisateurs que d'avoir de bonnes idées vous-même. C'est même préférable, parfois.

les utilisateurs et les autres développeurs sont partie prenante du processus du développement

les écouter et adopter leurs idées est indispensable

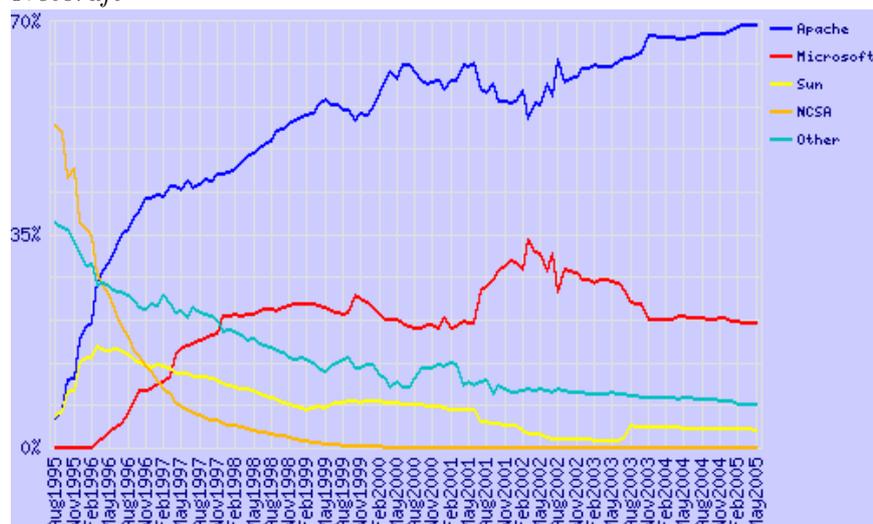
5 Les Succès...

5.1 Apache

Apache le serveur HTTP (Web) de l'[Apache Software Foundation](#) a été créé dans l'optique de rester libre et de fonctionner sur de nombreux

systèmes d'exploitations. L'objectif du projet est de fournir un serveur HTTP sécurisé, efficace et extensible en accord avec les standards actuel. Il se hisse d'ailleurs en première place des serveurs web depuis avril 1996 avec près de 70% des parts de marché à ce jour.

Évolution des parts de marché de serveur Web jusqu'à ce jour d'après Netcraft³



Aujourd'hui

le consortium Apache gère des dizaines de projets dont les plus célèbre le logiciel anti-spam **SpamAssassin** et **Ant** un logiciel en Java⁴ très utilisé des développeurs.

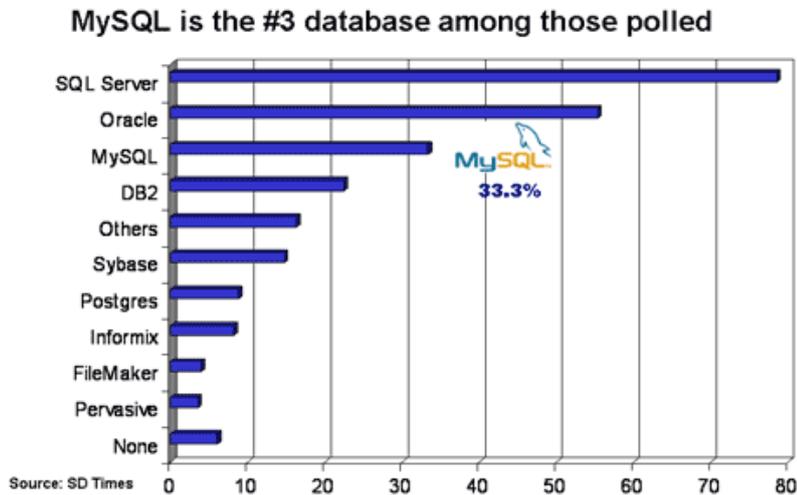
5.2 PHP, MySQL

Sur la lancée d'Apache deux logiciels se sont imposés. **PHP**, le langage de script pour le web et **MySQL** une base de donnée bien intégrée à PHP. Selon une *étude de PHPIndex*, PHP est un outil de premier plan. En effet au 1/10/2004, PHP est utilisé par 87% des sociétés cotées au CAC40 (contre 78% en 2003) et 95% des plus importantes sociétés en terme de bénéfice (contre 80% en 2003). **MySQL** est le troisième logiciel de base de données après ORACLE et SQL SERVER de MICROSOFT 2 gigantesques ténors du marché avec 33,3% des parts de marché. Il arrive ainsi devant toutes les autres base de données commerciales comme DB2 d'IBM pourtant réputé.

Part de marché de MySQL

³http://news.netcraft.com/archives/web_server_survey.html

⁴langage de programmation



5.3 Mozilla Firefox

Mozilla est un projet de création de suite Internet complète incluant un navigateur web et un client mail notamment. **Mozilla Firefox** a été d'abord créé comme un projet indépendant de Mozilla basé sur son code sous le nom de Phœnix. Cependant il a été ensuite intégré comme navigateur officiel de la suite Mozilla. Le projet Mozilla Browser a donc été abandonné en sa faveur ! Il est réputé pour son excellent technique et les avancées en terme d'utilisabilité tel le blocage des pop-ups (publicité non sollicités sur le web) et la gestion des onglets (possibilité de visiter plusieurs pages webs depuis la même fenêtre). Il détient aujourd'hui selon une *étude de XiTi datant du 24/04/2005* une part de marché moyenne en Europe de l'ordre 13,31% avec pour la France 13.3%, l'Allemagne 22.8% et la Finlande 30.9% ce qui le place en deuxième position derrière *Microsoft Internet Explorer* installé par défaut sur toute nouvelle machine...

5.4 OpenOffice.Org

OpenOffice.org est une suite bureautique concurrente de Microsoft Office. Au départ initié par SUN qui l'a rendu libre il a depuis décollé littéralement aussi bien en utilisation qu'en richesse de fonctionnalités. Il dispose aujourd'hui de fonctionnalités dont Microsoft Office ne possède pas, bien que l'inverse soit également vrai. OpenOffice.org a ainsi été déployé par la gendarmerie française et à travers de nombreuses entreprises et administrations dans le monde.

5.5 Autres...

BIND serveur de DNS (permet d'identifier les machines et de faire la correspondance entre l'IP et les noms de domaines), si vous vous connectez sur un quelconque réseau avec une gestion des noms vous passez quasi-obligatoirement par lui...

SendMail serveur de mail leader dans le domaine, la plupart de vos emails passent probablement par SendMail!

Conclusion

Nous avons du vu que le logiciel libre dispose d'une définition claire mais qu'il existe une grande variété de logiciels libres. Et également qu'il n'y a pas de doute à avoir quant à la différence entre les logiciels libres et non-libres. Mais le logiciel libre ce n'est pas uniquement du logiciel et une licence, c'est également et on l'a vu un mode de développement et une attitude original. Et c'est peut-être là que réside la grande force et la nouveauté dans le logiciel libre...

Références

- [À la conquête de la noosphère] Eric S. Raymond, Traducteurs : Emmanuel Fleury & Sébastien Blondeel http://www.linux-france.org/article/these/noosphere/homesteading-fr_monoblock.html, version 1.14, du 21 novembre 1998 ; traduit au printemps 1999
- [Qu'est-ce qu'un Logiciel Libre ?] GNU <http://www.gnu.org/philosophy/free-sw.fr.html>
- [Wikipedia : Logiciel Libre] Wikipedia http://fr.wikipedia.org/wiki/Logiciel_libre
- [Pourquoi les logiciels ne doivent pas avoir de propriétaire] Richard M. Stallman <http://www.gnu.org/philosophy/why-free.fr.html>
- [Catégories de logiciels libres et non libres] GNU <http://www.gnu.org/philosophy/categories.fr.html>
- [Free/Libre and Open Source Software : Survey and Study] <http://www.infonomics.nl/FLOSS/report/>
- [Linux et le développement décentralisé] Auteur : Christopher B. Browne, Traducteurs : Sébastien Blondeel et Gaël Duval http://www.linux-france.org/article/these/lsf-fr/lsf-fr_monoblock.html
- [La cathédrale, le bazar et le conseil municipal] Auteur : Alan Cox, Traducteurs : Géraud Canet & Emmanuel Fleury, Édition : Sébastien Blondeel http://www.linux-france.org/article/these/conseil-municipal/bazaar_fr_monoblock.html
- [La cathédrale et le bazar] Eric S. Raymond [http://www.linux-france.org/article/these/cathedrale-bazar_monoblock.html](http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar_monoblock.html) (site originel en anglais :)
- [Tribune Libre] TRIBUNE LIBRE *Ténors de l'Informatique Libre* 1999 par éditions O'Reilly, <http://www.oreilly.fr/divers/tribune-libre/>
- [The Emerging Economics of Open Source Software] Bruce Perens <http://perens.com/Articles/Economic.html>