

Contraintes juridiques

Logiciels libres et Open source

J.-P. CARUANA – P. PUJALTE – F. THEROND

16 décembre 2002

Table des matières

1	De l'origine des logiciels libres	2
2	Définition	2
2.1	Du logiciel libre	2
2.2	Des logiciels open source	3
3	Des licences	4
3.1	Du logiciel libre copylefté	4
3.2	Du logiciel libre non-copylefté	4
3.3	Du logiciel du domaine public	5
3.4	Du logiciel couvert par la GPL	5
4	Des enjeux du logiciel libre	5
4.1	Point de vue de l'utilisateur	5
4.2	Point de vue du développeur	6
4.3	Une philosophie	6
5	De la brevetabilité des logiciels largement contestée	7
5.1	Historique et contexte des brevets logiciels en Europe	7
5.2	Les arguments des partisans des brevets logiciels largement contestables	8
5.3	Prise en compte des particularités des logiciels	9
5.4	Brevets logiciels et démocratie	9
5.5	Conclusion	10
6	Exemples de logiciels libres	10
6.1	OpenOffice.org et StarOffice	10
6.2	Emacs, le fleuron du logiciel libre	11
6.3	Linux, le système d'exploitation alternatif	12
6.4	Autres exemples de logiciels libres célèbres pour Linux	12

1 De l'origine des logiciels libres

Dans les années 70, les programmeurs qui travaillaient essentiellement dans les laboratoires étaient payés pour programmer et non pas pour créer des logiciels à but commerciaux. Au début des années 80, avec la démocratisation de l'informatique, les programmeurs ont commencé à travailler pour des logiciels commerciaux. Ces logiciels propriétaires ne laissaient plus les utilisateurs libres de faire tout ce qu'ils désiraient de leur logiciel. "Qui partage avec son voisin est un pirate. Qui souhaite la moindre modification doit nous supplier de la lui faire". C'est contre cette position que Richard Stallman a décidé de réagir en créant le projet GNU (Gnu is Not Unix), qui se basait sur une licence d'un nouveau genre : GPL (General Public Licence). Les logiciels sous cette licence sont appelés "logiciels libres" (où la notion de liberté est le cheval de bataille de la communauté). En 1997, Eric Raymond a décidé de modifier l'appellation de logiciel libre par "logiciel Open Source". Son souci était de trouver une façon de promouvoir les idées entourant le logiciel libre vers les gens qui avaient autrefois fui le concept. Il s'inquiétait des effets du message apparemment hostile à toute perspective mercantile de la Free Software Foundation de Stallman, qui pourrait interdire aux utilisateurs de bénéficier de la puissance du logiciel libre.

2 Définition

Il existe des divergences de point de vue entre les partisans des logiciels libres et ceux des logiciels Open Source. Richard Stallman, le fondateur de la licence GPL (voir plus bas) pense que la définition Open Source a été détournée à des fins commerciales. Pour cela regardons les définitions propres à chacune de ces expressions.

2.1 Du logiciel libre

L'expression *Logiciel libre* fait référence à la liberté pour les utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Plus précisément, elle fait référence à quatre types de libertés pour l'utilisateur du logiciel :

- La liberté d'exécuter le programme, pour tous les usages. C'est la logique même. Pour utiliser un programme il faut en avoir le droit. C'est le droit de base de toutes les licences, même les plus restrictives.
- 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. Ici, nous entrons dans des considérations plus rares dans le monde des logiciels propriétaires. Cette liberté permet à l'utilisateur d'améliorer le programme. Ainsi, il fonctionnera mieux et l'utilisateur en sera plus satisfait. Cette liberté permet également de satisfaire la

curiosité de l'utilisateur en connaissant parfaitement le produit qu'il utilise et s'assurer qu'aucun vice caché ne le gênera à l'avenir.

- La liberté de redistribuer des copies. Liberté est synonyme de globalité. Pour être libre il faut pouvoir partager avec les autres, échanger des informations, offrir de l'aide ...
- La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté. Pour ceci, l'accès au code source est une condition requise.

Un programme est un logiciel libre si les utilisateurs ont toutes ces libertés. Ainsi, vous êtes libre de redistribuer des copies, avec ou sans modification, gratuitement ou non, à tout le monde, partout. Être libre de faire ceci signifie (entre autres) que vous n'avez pas à demander ou à payer pour en avoir la permission.

2.2 Des logiciels open source

La signification évidente de *Open source* est "Vous pouvez avoir le code source". Cette catégorie de logiciels n'est clairement pas la même que celle de logiciels libres. Elle inclut les logiciels libres, mais aussi les logiciels semi-libres tels que Xv, et même les logiciels non-libres tels que Qt sous sa licence d'origine. L'Open Source Definition permet donc plus de libertés que la GPL. Elle permet surtout une plus grande promiscuité lors d'un mélange de code propriétaire avec du code open source. Il existe plusieurs versions de la définition d'un logiciel open source. Voici les éléments principaux de cette définition (version 1.9) :

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

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

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.

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.

C'est sur ce dernier point que la vision du logiciel libre de Stallmann diffère de celle des partisans de l'open source. Cela a partiellement divisé la communauté Linux. Concrètement, cela a débouché sur le développement de deux environnements graphiques distincts pour Linux : *KDE* pour les partisans open source et *Gnome* pour la GNU / GPL.

3 Des licences

Il n'existe pas de licences libres, mais uniquement des licences qu'il est possible de qualifier du mot libre. Il est courant de faire automatiquement la liaison entre Logiciels Libres et GPL. Les licences "Open Source" peuvent être de même nature que celles du logiciel libres mais elles incluent également des licences propriétaires. Vu le contexte de l'étude "Logiciel libre / Logiciel open source" nous allons nous limiter à l'énumération de quelques licences dites "libres". Nous ne verrons ni les licences semi libres ni les licences propriétaires (qui pourraient définir certains logiciels open source).

3.1 Du logiciel libre copylefté

Le logiciel sous copyleft 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 toujours être un logiciel libre.

Dans le projet GNU, presque tous les logiciels sont soumis au copyleft, car le but est de donner à chaque utilisateur les libertés garanties par le terme *logiciel libre*. GNU a donc mis en place la licence GPL dont on reparlera plus tard qui permet ainsi d'être sûr que le logiciel restera dans le monde libre.

3.2 Du logiciel libre non-copylefté

Le logiciel libre non-copylefté est diffusé par son auteur avec la permission de le redistribuer et de le modifier, mais aussi d'y ajouter d'autres restrictions. On nomme souvent ses logiciels comme semi-libres.

Si un programme est libre, mais non-copylefté, alors certaines copies ou versions modifiées peuvent ne plus être libres. Une société informatique peut compiler ce programme, avec ou sans modifications, et distribuer le fichier exécutable sous forme de produit logiciel propriétaire.

3.3 Du logiciel du domaine public

Logiciel du domaine public veut dire logiciel non soumis aux droits d'auteurs. C'est un cas particulier de logiciel libre *non-copylefté*, ce qui veut dire que certaines copies, ou certaines versions modifiées, peuvent ne pas être libres. Un logiciel appartenant au domaine public ne possède aucun droit de copie et tous ses éléments peuvent être vus, copiés et utilisés quelque soit le but.

3.4 Du logiciel couvert par la GPL

La GNU GPL (Licence Publique Générale GNU8) est un ensemble spécifique de conditions de distribution pour copylefter un programme. Le projet GNU l'utilise comme conditions de distribution de la plupart de ses logiciels.

La licence GPL est aujourd'hui la plus utilisée dans le monde du logiciel libre et favorise son bon développement.

Il existe beaucoup d'autres licences autour des logiciels libres : on peut citer rapidement la BSD Licence, La Licence Artistique Clarifiée. . .

4 Des enjeux du logiciel libre

Le Logiciel libre a un enjeu capital dans dans une industrie du logiciel essentiellement dominée par des sociétés Nord Américaines.

Sur plusieurs point cet enjeu est considérable.

4.1 Point de vue de l'utilisateur

Dans l'industrie du logiciel propriétaire, le client n'est pas toujours roi. Les dérives des modèles propriétaires sont nombreuses. Elles visent systématiquement à rendre captif les utilisateurs et tendent vers un système où quelques sociétés contrôlent l'information.

Pour l'utilisateur, le logiciel libre a de multiples avantages :

1. La sécurité : Par l'accès au source et la possibilité de pouvoir réagir rapidement en cas de faille constatée sans avoir à repasser par l'éditeur ou devoir attendre que celle-ci soit corrigée dans la prochaine version.
2. La qualité : Par la mise en place de procédures de déclaration des "bogues" et leur correction par la communauté de développeurs.
3. La pérennité : Par l'adoption de formats ouverts garantissant l'accès aux données sur le long terme, et l'indépendance de l'utilisateur.
4. L'interopérabilité : La possibilité d'assurer la compatibilité des formats de stockages et le développement d'outils de conversion.
5. Le prix : lorsque le coût total d'acquisition est un élément important (Pour un étudiant à titre individuel ou un centre de ressource informatique à vocation pédagogique par exemple)

4.2 Point de vue du développeur

Alors que les logiciels libres n'ont quasiment que des avantages pour les utilisateurs, la question est plus délicate à appréhender du point de vue des développeurs car elle suppose l'adhésion à un modèle économique. D'un point de vue individuel d'un développeur-auteur, la participation à une communauté élaborant un logiciel libre peut avoir plusieurs origines :

1. La notoriété et la visibilité : la diffusion importante des logiciels libres permet aux développeurs les plus actifs de faire reconnaître leurs compétences.
2. Le "débugage" : il s'agit du retour d'expérience entre les communautés d'utilisateurs et de développeurs. Le signalement d'erreurs par les utilisateurs, limite le temps passé par le développeur à cette tâche fastidieuse.
3. La mutualisation de l'effort de développement : dans le cas du développement d'un logiciel dont il a besoin, Il bénéficie du travail fourni par les autres développeurs et peut compter sur la complémentarité de leurs compétences. Par exemple, la traduction des documents s'appuie systématiquement sur la communauté des utilisateurs/développeurs.
4. L'acceptation de standards : La gratuité du logiciel et les garanties apportées par une diffusion sous licence libre permet la mise en place d'une communauté importante d'utilisateurs. Cette reconnaissance des utilisateurs permet l'acceptation de formats de données utilisés en tant que standard.

4.3 Une philosophie

Pour autant, ne vanter que les mérites techniques des logiciels libres en négligeant leur philosophie conduit à une impasse. L'enjeu réel du logiciel libre est avant tout social et politique. Si les logiciels libres suscitent aujourd'hui un intérêt technique à court terme, leur avantage technique n'est que la retombée, après quinze ans de combat, d'un modèle qui vaut surtout par ses effets à long terme. Ne voir que le court terme, c'est s'exposer continuellement à retomber dans les pièges du logiciel propriétaire, c'est ne pas apprendre. Le vrai moteur du Libre Logiciel est bien la Liberté, terme devant être pris dans le sens civique, politique : liberté d'expression, liberté d'association, liberté d'entreprise, liberté d'user à sa guise de l'information disponible et de la partager au bénéfice de chacun, donc de tous.

De plus en plus de personnes peuvent utiliser les logiciels libres pour leur côté pratique, mais cela n'accroît pas pour autant la communauté du logiciel libre et ne la pérennise pas. Le combat du logiciel libre doit perdurer tant que la notion de logiciel existera. Or, qui peut garantir que les logiciels libres que nous utilisons actuellement seront toujours pertinents dans trente ans ou tout

simplement l'année prochaine ? Nous ne braderons pas notre liberté pour de simples questions de commodité. Nous devons soutenir le logiciel libre pour ce qu'il est, même si le logiciel propriétaire devait s'avérer momentanément plus puissant ou plus efficace. Les questions de liberté et d'intérêt social sont au centre des préoccupations du monde du libre.

Le mouvement du logiciel libre, se référant à l'utilité sociale, s'oppose à l'appropriation individuelle de la production intellectuelle dans le logiciel. Profitant actuellement du succès des nouvelles technologies de l'information et de la communication, des groupes d'intérêt se mobilisent pour renforcer la dérive vers l'appropriation intellectuelle, au détriment de l'intérêt général qui veut que les connaissances soient un bien public universel, et également au détriment des droits fondamentaux que sont la liberté d'accès à l'information et la liberté d'expression. Profitant des idées reçues et des fantasmes collectifs qui circulent sur les nouvelles technologies et sur les nouvelles interactions sociales, stigmatisant le partage et matérialisant l'immatériel, les lobbies de l'appropriation intellectuelle n'hésitent pas à détourner le droit d'auteur, principalement pour le plus grand profit de ceux qui deviennent justement inutiles à la production numérique. Et tant pis si le citoyen voit disparaître des droits qu'il lui semblerait naturel d'exercer dans la rue ou dans une société analogique ! C'est pourtant oublier un peu vite que la motivation originelle et officielle de la propriété intellectuelle était, et est toujours, de préserver l'intérêt de l'Humanité en reversant dans le domaine public une oeuvre qui survit ainsi à son créateur.

Finalement, le mouvement du logiciel libre prend racine dans un idéal qui postule la liberté absolue et le caractère universel du savoir et de l'information ; idéal qui n'est pas exclusif au logiciel.

5 De la brevetabilité des logiciels largement contestée

Le but affiché des brevets est de favoriser l'innovation technologique, favoriser la publication des inventions et leur mise en oeuvre. Mais qu'en est-il des brevets de logiciel ?

5.1 Historique et contexte des brevets logiciels en Europe

Le texte qui régit les brevets en Europe, la première directive d'examen sur la brevetabilité, a été signé lors de la convention sur la délivrance des brevets européens du 5 octobre 1973 à Munich, et mis en application le premier juin 1978. Ce texte explique ce que sont les inventions. "Les programmes d'ordinateurs" à l'instar des mathématiques en sont exclus.

Or, le poids économique des logiciels et plus largement de l'informatique a pris de plus en plus d'importance. Cela a conduit à la naissance de groupes

de pression, au sein desquels on retrouve de grands groupes européens (Alcatel, Siemens) mais également américains (IBM), afin que l'Europe aligne sa politique de brevets logiciels sur celle des États-Unis et du Japon. C'est ainsi que, contrairement à ce que stipule le texte d'examen de brevetabilité qui exclut les logiciels des objets brevetables, l'OEB a délivré plus de 20000 brevets logiciels.

Bien entendu, d'un point de vue juridique, ces brevets n'ont pas de valeur réelle en Europe. L'OEB tente donc depuis quelques années d'officialiser ces pratiques par une modification de la directive d'examen de brevetabilité de 1973. C'est ainsi qu'a eu lieu, en novembre 2000, la conférence pour la révision de la convention sur les brevets qui avait pour but de moderniser le texte de 1973. À l'issue de cette convention, il a été décidé de ne pas supprimer les programmes d'ordinateurs de la liste des inventions non brevetables en attendant qu'une consultation soit effectuée au niveau des états européens.

Malheureusement, alors que les consultations sont toujours en cours au niveau des états européens, l'OEB n'a pas attendu leur avis et a modifié le texte d'examen de brevetabilité le 05 octobre 2001, afin de prendre en compte la brevetabilité des logiciels qui produisent un "effet technique". Pire, une proposition de directive de la commission des communautés européennes, concernant la brevetabilité des inventions mises en oeuvre par ordinateur, a vu le jour le 20 février 2002.

5.2 Les arguments des partisans des brevets logiciels largement contestables

Les partisans des brevets logiciels sont essentiellement de grands groupes industriels, les multinationales, les spécialistes en propriété industrielle et les juristes. Ils mettent en avant un certain nombre de points pour appuyer leurs thèses.

De leur point de vue, le manque de clarté de la brevetabilité des logiciels pénalise les entreprises européennes et la régularisation des textes est nécessaire. Pourtant, force est de constater que sur les 20000 brevets européens accordés par l'OEB, 75% sont américaine. Du coup, on peut facilement en conclure que cela pénalise surtout les américains. . .

Aucune étude ne prouve que les brevets logiciels favorisent effectivement l'innovation. La réussite des grandes sociétés américaines de logiciels n'est absolument pas à imputer aux brevets, mais surtout à une politique commerciale bien plus aiguisée.

De plus, à part quelques grands groupes industriels, personne ne semble spécialement demandeur de brevets logiciels et surtout pas les PME. Cela est souvent imputé à une méconnaissance des brevets. En réalité, les gains qu'apportent les brevets n'apparaissent pas clairement à la plupart des acteurs économiques. Par contre, les inconvénients, eux, sont évidents : ils coûtent chers, ils accroissent les risques de contentieux et, par la même, les

frais juridiques.

5.3 Prise en compte des particularités des logiciels

Les particularités des logiciels vont à l'encontre de la brevetabilité :

- un logiciel est un bien immatériel, c'est une création de l'esprit. De par cette spécificité, il est nettement plus adapté aux droits d'auteurs ;
- un logiciel, ce sont des milliers de lignes de code, de multiples fonctionnalités imbriquées. Appliquer les brevets aux logiciels est quasiment impossible ;
- les investissements pour réaliser un logiciel innovant ne sont en général absolument pas comparables à des investissements industriels ;
- les cycles de changements en informatique sont extrêmement courts (3 ans), un logiciel innovant prend donc un avantage décisif en sortant le premier ;
- la durée d'un brevet (20 ans), n'est absolument pas justifiée et pourrait être extrêmement préjudiciable à l'innovation. Possédiez-vous un micro-ordinateur il y a 20 ans ?

5.4 Brevets logiciels et démocratie

À la lecture de la proposition de directive concernant la brevetabilité des inventions mises en oeuvre par ordinateur, on est édifié par les commentaires concernant la consultation. Sur les 1450 réponses reçues, il est indiqué qu'elles émanent principalement (à 90% !) des développeurs issus du logiciel libre, mais qu'au final, le poids économique des partisans des brevets logiciels est nettement plus important et fait donc pencher la balance en leur faveur. La loi du plus riche est donc la meilleure !

C'est pourtant la démocratie que tente de faire prévaloir l'alliance Eurolinux en recueillant les signatures d'une pétition contre les brevets logiciels qui a déjà été signée à ce jour par 300 PME et plus de 100000 personnes dans toute l'Europe. Eurolinux regroupe des associations (AFUL, APRIL, FFII, SSLUG...) et des sociétés commerciales européennes (SuSE, Mandrakesoft, MySQL...) afin de promouvoir une informatique européenne fondée sur les droits d'auteur, les standards ouverts, la libre concurrence et les libertés.

Le critère économique n'est donc pas le seul à devoir être pris en compte. Il faut remarquer malgré tout que sur ce critère, l'importance des PME en France et en Europe semble être extrêmement sous-estimé. Le fait que les partisans des brevets logiciels contribuent plus au PIB et à l'emploi (critères qui semblent être pris en compte par la commission) que l'ensemble des PME du secteur logiciel reste à démontrer.

Prendre en compte les seuls critères économiques est donc très dangereux. Il est risqué de passer à côté d'un grand changement : celui des logiciels libres. Ils représentent une chance pour l'Europe de se défaire de l'emprise

des sociétés américaines, et d'obtenir son indépendance technologique. Les logiciels libres sont sûrement la seule chance d'éviter que des pans entiers des infrastructures informatiques soient aux mains de quelques-uns : ils doivent être utilisables par tous avec un minimum de freins économiques. Les logiciels libres, ce sont des milliers de développeurs, sources d'innovation, dont l'idéal de liberté est menacé par les brevets logiciels.

5.5 Conclusion

Le système des brevets logiciels aux États-Unis a très largement démontré qu'il était inapplicable. Suivre ce modèle, agrémenté de quelques précautions supplémentaires, et croire ainsi que l'Europe s'en sortirait mieux que ces concurrents américains est un leurre.

Les logiciels, comme toutes les créations immatérielles, sont déjà protégés par les droits d'auteur. Vouloir ajouter une protection juridique supplémentaire aux logiciels en leur octroyant la possibilité d'être brevetés, c'est faire passer les logiciels du système de propriété intellectuelle au système de propriété industrielle. Or le processus de création d'un logiciel relève bien plus d'un processus intellectuel que d'un processus industriel.

Aller vers les brevets logiciels, c'est s'aligner sur les règles du jeu qu'ont fixé les grandes sociétés américaines en vue de renforcer leurs positions historiquement dominantes, en profitant du système de rentes qu'induit le brevet logiciel. Ceci équivaut à un renoncement total d'une indépendance technologique de l'Europe.

Aller vers les brevets logiciels c'est décourager, voir tuer les PME du logiciel européen pour qui l'innovation est un facteur de survie. Mettre en place le système de brevets logiciels, c'est ajouter des contraintes lourdes pour les PME, et donc, par là même, freiner leur capacité d'innovation. C'est enfin mépriser de nombreux développeurs indépendants, jeunes pour la plupart, qui n'ont pas besoin des brevets logiciels et qui pourraient souffrir des décisions peu éclairées de leurs aînés.

6 Exemples de logiciels libres

6.1 OpenOffice.org et StarOffice

Star Division, une société allemande fondée dans les années 80, a développé une suite bureautique de grande qualité pour les systèmes Unix, Linux et Windows. Cette suite était gratuite si on s'enregistrait auprès de la maison mère et qu'on s'engageait à ne s'en servir qu'à des fins personnelles et non commerciales.

Cette suite avait de nombreux avantages : en plus d'être gratuite, elle proposait une très grande compatibilité avec les documents générés par la célèbre suite de Microsoft : Microsoft Office.

Puis Sun Microsystems racheta la firme allemande en 1999 et continua de développer la suite jusqu'à la version 5.2. Sun prit alors une grande décision à ce moment : ils allaient continuer le développement de la suite vers une version 6 tout en ouvrant et en distribuant les sources de la future version 6 (aujourd'hui commercialisée). C'est ainsi que naquit le projet OpenOffice.org, toujours épaulé par Sun.

Sun Microsystems, en plaçant sa suite bureautique StarOffice dans le circuit de l'OpenSource, est ainsi un précurseur : c'est effectivement la première fois qu'un logiciel commercial accessible au grand public aussi bien qu'aux professionnels est ainsi proposé à la communauté.

Pour OpenOffice.org, deux licences *OpenSource* sont employées. Il s'agit respectivement de la licence GPL (General Public License) et d'une variante de Sun Microsystems, la SISSL (Sun Industry Standards Source License). Cette double-licence, procédé communément employé dans les projets OpenSource professionnels, garantit aussi bien la liberté de travail sur les sources que la compatibilité avec un environnement de société commerciale.

La licence SISSL donne plus de liberté au programmeur qu'une licence purement GPL.

Les différences entre les version StarOffice et OpenOffice.org : Le code source utilisé pour OpenOffice.org ne correspond pas à l'intégralité du code de StarOffice. La raison en est que Sun a payé certaines licences pour des parties tierces du code incluses dans StarOffice et pour lesquelles elle n'a pas la permission de les utiliser pour OpenOffice.org. Ces parties qui sont ou seront présentes dans StarOffice mais ne le seront pas dans OpenOffice.org sont :

- certaines polices (incluant surtout des polices asiatiques) ;
- le serveur de base de données (Adabas D) ;
- les modèles ;
- la Gallery de clipart étendue ;
- certaines fonctions de tris (versions asiatiques) ;
- certains filtres de fichiers.

La communauté OpenOffice.org travaille au développement des ces parties. Certaines sont déjà présentes comme les modèles et les images pour la Gallery.

6.2 Emacs, le fleuron du logiciel libre

Emacs, le fameux éditeur de texte désormais disponible sur toutes les plateformes, est l'enfant du fondateur du projet GNU et de la FSF (Free Software Fondation) : Richard Stallmann. Il est disponible avec presque toutes les distributions GNU/Linux. Sa puissance (Emacs est entièrement configurable) et sa modularité (chacun installe les composants qui lui sont nécessaires, XEmacs peut même servir à lire son courrier électronique ou à naviguer sur Internet) en font quasiment un environnement de développe-

ment intégré (il est possible de compiler ses programmes directement depuis une invite Emacs).

Emacs fut le premier logiciel sous licence GNU. En 1984, Richard Stallman dut abandonner le code de Emacs parce que une entreprise réclama des droits de propriété sur les développements qui ont eu lieu entre 1980 et 1984. Ne pouvant abandonner 10 années de travail, Richard Stallman créa le projet GNU et écrivit GNU-Emacs, plus communément appelé Emacs.

À noter : l'existence d'un autre produit, XEmacs, qui ne possède que peu de différences avec Emacs. Il y a cependant une lutte plus religieuse qu'autre chose pour savoir lequel est le meilleur. Ces deux produits sont des produits sous licence GPL.

6.3 Linux, le système d'exploitation alternatif

Linux, ou plutôt GNU/Linux est un système d'exploitation issu du projet GNU de créer un système d'exploitation entièrement libre. Richard Stallman préfère l'appellation GNU/Linux car elle montre mieux que Linux n'est en fait que le noyau du système d'exploitation libre, les autres composants étant des logiciels sous licence GNU/GPL.

Le noyau de Linux, écrit par Linus Torvald au début des années 1990 a permis aux applications libres déjà existantes depuis la fondation de GNU en 1984 de fonctionner sur un composant libre (avant l'écriture du noyau les logiciels libres fonctionnaient sous Unix). Aujourd'hui, il existe beaucoup de distributions différentes de l'OS libre, les plus connues étant Red Hat, Mandrake ou encore Debian. Ces distributions sont en fait des collections de paquetages (libres pour la plupart d'entre eux) livrés avec un programme d'installation du système. Les distributions comme Red Hat ou Mandrake sont issues de sociétés de services qui proposent une assistance et une documentation complète de leur distribution. Elles ne laissent en libre accès qu'une partie de leur distribution, les versions complètes pour les entreprises étant payantes avec des paquetages propriétaires. Cependant il reste des distributions de Linux fidèles à la philosophie du logiciel libre comme Debian ou encore Gentoo composées exclusivement de paquetages libres.

6.4 Autres exemples de logiciels libres célèbres pour Linux

Linux est livré dans la plupart de ses distributions, avec une flopée de logiciels libres de grande qualité. Il y en a de plusieurs types :

- Des navigateurs Web comme Mozilla (le moteur de Netscape) ou Opera ;
- De nombreux autres outils de communications : courrier électronique, discussions (news et IRC), logiciels de travail en groupes, serveurs de faxes ;
- The GIMP, logiciel de retouche d'images et de création graphique comparable, en mieux, à Adobe Photoshop et d'autres logiciels de création

- graphique (2D et 3D);
- \LaTeX , traitement de texte utilisé dans les milieux scientifiques.