

# La définition de l'*Open Source*

Auteur : Bruce Perens

Traducteur : [Sébastien Blondeel](#)

1998 ; traduit en mars-avril 1999

[http://www.linux-france.org/article/these/the\\_osd/fr-the\\_open\\_source\\_definition\\_monoblock.html](http://www.linux-france.org/article/these/the_osd/fr-the_open_source_definition_monoblock.html)

---

Essai publié dans le livre *Open Sources – Voices from the Open Source Revolution*, ISBN 1-56592-582-3, janvier 1999, édité par Chris DiBona, Sam Ockman, et Mark Stone chez O'Reilly & Associates. Une version en français est [en cours d'adaptation](#) par et pour les [Éditions O'Reilly](#). Bruce Perens détient le copyright sur ce texte et l'a publié sous les termes de la [licence publique générale de GNU](#), version 2 ou ultérieure.

---

## 1. Introduction

L'utilisateur typique d'ordinateurs possède des quantités de logiciels qu'il a acquis de par le passé et qu'il n'utilise plus désormais. Il a peut-être acheté un ordinateur plus performant ou changé de constructeur, ce qui a empêché ses programmes de continuer à fonctionner. Les logiciels sont peut-être devenus obsolètes, ou ont cessé de répondre à ses besoins. Il a peut-être acheté deux ordinateurs ou plus, et ne souhaite pas déboursier quoi que ce soit pour obtenir une deuxième copie du même logiciel. Quelle qu'en soit la raison, le logiciel qui lui a coûté de l'argent voici quelques années ne remplit plus son rôle. Cette situation est-elle inévitable ?

Que penseriez-vous d'une situation dans laquelle vous auriez le droit d'obtenir une mise à jour dès que vos logiciels en éprouveraient le besoin ? Que penser d'une situation dans laquelle, en passant l'architecture *Mac* à l'architecture compatible *IBM PC*, on pourrait changer gratuitement les versions des logiciels ? Et si, quand par malheur le logiciel ne fonctionne pas ou ne remplit pas tous vos besoins, vous pouviez le faire améliorer ou le réparer par vous-même ? Peut-on imaginer que les logiciels soient encore corrigés et proposés après la banqueroute de leur éditeur ? qu'on puisse les utiliser sur la station de travail au bureau, sur l'ordinateur de bureau à la maison, et sur son ordinateur portable, plutôt que d'être limité à une seule machine ? Si tel était le cas, vous continueriez sans doute à utiliser les logiciels que vous avez achetés il y a des années. Ce sont là quelques-uns des droits que l'*Open Source* vous garantit.

La définition de l'*Open Source* est une déclaration des droits de l'utilisateur d'ordinateur. Elle définit certains droits que la licence d'un logiciel doit vous garantir pour pouvoir être qualifiée d'*Open Source*. Ceux qui ne proposent pas des programmes *Open Source* trouvent que la concurrence est de plus en plus rude avec ceux qui ont compris que les utilisateurs commencent à apprécier et à exiger les droits qu'ils auraient dû avoir depuis toujours. Des programmes comme ceux qui constituent le système d'exploitation *Gnu/Linux* et le navigateur web de *Netscape* ont acquis une grande popularité, en prenant la place de logiciels couverts par des licences plus restrictives. Les sociétés qui utilisent du logiciel *Os* profitent des avantages liés à un modèle de développement très réactif, souvent mis en place par plusieurs sociétés qui coopèrent, et dont une grande partie du travail est fournie par des individuels qui ont tout simplement besoin d'une amélioration qui réponde mieux à leurs besoins.

Les volontaires qui ont créé des produits comme *Linux* n'existent, et les sociétés ne peuvent coopérer, que grâce aux droits fournis par l'*Open Source*. Le programmeur moyen se sentirait lésé s'il investissait énormément de travail dans un programme, pour constater que le propriétaire de ce dernier se contente de

vendre la version améliorée sans rien lui proposer en retour. Mais ces mêmes programmeurs n'ont pas peur d'apporter des contributions à l'*Open Source*, car ils savent que les droits suivants leur sont alors garantis :

- le droit de faire des copies du programme, et de distribuer ces copies.
- le droit d'accéder au code source du logiciel, un préliminaire nécessaire pour pouvoir y apporter des modifications.
- le droit d'améliorer le programme.

Ces droits comptent pour celui qui contribue à améliorer des logiciels car ils maintiennent tous les développeurs au même niveau. Quiconque le souhaite a le droit de vendre un programme *Open Source*, aussi les prix seront-ils bas et le développement destiné à conquérir de nouveaux marchés sera-t-il rapide. Quiconque investit de son temps à construire des connaissances sous la forme d'un programme *Open Source* peut fournir de l'aide sur ce dernier, et cela fournit aux utilisateurs la possibilité de mettre en place leur propre assistance technique, ou de choisir parmi des assistances techniques en concurrence les unes avec les autres. Tout programmeur peut spécialiser un programme *Open Source* pour le rendre utilisable dans des marchés spécifiques afin d'atteindre de nouveaux consommateurs. Ceux qui se lancent dans de telles entreprises ne sont pas tenus d'acquiescer des *royalties* ou la moindre licence.

La raison de la réussite de cette stratégie qui semble s'inspirer du communisme, alors que le communisme est lui-même un échec patent dans le monde entier, vient du fait que l'économie de l'information est fondamentalement différente de l'économie qui règle la consommation des autres produits. On peut copier une information telle qu'un programme d'ordinateur pour un prix dérisoire. L'électricité qu'il en coûte ne dépasse par l'eurocent, pas plus que l'utilisation de l'équipement nécessaire à cet acte. En comparaison, on ne peut pas recopier une miche de pain sans dépenser une livre de farine.

## 2. L'histoire

Le concept de *logiciel libre* est ancien. Quand les ordinateurs sont entrés dans les universités, c'était en tant qu'outils de recherche. Les logiciels étaient librement passés de laboratoire en laboratoire, et les programmeurs étaient payés pour le fait de programmer, et non pas pour les programmes en eux-mêmes. Ce n'est que plus tard, quand les ordinateurs sont entrés dans le monde des affaires, que les programmeurs ont commencé à subvenir à leurs besoins en limitant les droits associés à leurs logiciels et en taxant chaque copie. Le *Logiciel Libre*, en tant qu'idée politique, a été popularisé par Richard Stallman depuis 1984, année où il a créé la *Free Software Foundation* et son projet *GNU*. La prémisse de M. Stallman est que tout le monde devrait jouir de plus de liberté, et savoir apprécier cette dernière. Il a mis sur pied un ensemble de droits dont il estimait que tout utilisateur devait pouvoir jouir, et les a codifiés au sein de la licence publique générale de *GNU*, ou *GPL*. M. Stallman a intitulé, non sans malice, cette licence « gauche d'auteur » (*copyleft*), car au lieu d'interdire, elle donne le droit de copier. M. Stallman a lui-même développé de féconds travaux en matière de logiciels libres, tels que le compilateur de langage *C* du projet *GNU*, et *GNU Emacs*, un éditeur disposant d'attraits tels que certains en parlent comme d'une religion. Ses travaux ont inspiré de nombreux autres développeurs à proposer du logiciel libre selon les conditions de la *GPL*. Même si elle n'est pas promue avec la même ferveur libertaire, la définition de l'*Open Source* reprend de nombreuses idées de M. Stallman, et on peut la considérer comme une dérivation de ses propres travaux.

La définition de l'*Open Source* a vu le jour en tant que document présentant les grandes lignes du logiciel libre au sens de la distribution *Debian* de *Gnu/Linux*. *Debian*, un système Linux de la première heure, toujours populaire de nos jours, était entièrement constitué de logiciels libres. Cependant, comme la « gauche d'auteur » n'était pas la seule licence qui prétendît couvrir du logiciel « libre », *Debian* avait des difficultés à tracer la limite entre ce qui est libre et ce qui ne l'est pas, et n'avait jamais clairement indiqué les règles qu'elle appliquait pour déterminer si un logiciel était libre ou non. Je dirigeais alors

Le projet *Debian*, et j'ai résolu ce problème en proposant le *Contrat social de Debian* et les *Lignes de conduite en matière de logiciel libre de Debian* en juillet 1997. De nombreux développeurs *Debian* ont proposé des critiques et des améliorations, que j'incorporais peu à peu dans les documents. Le *Contrat social* documentait l'intention de la distribution *Debian* de ne composer leur système que de logiciels libres, et les *Lignes de conduite en matière de logiciel libre* permettaient de classer facilement le logiciel en deux catégories, « libre » ou non, en comparant la licence du logiciel aux lignes de conduite.

Les lignes de conduite de *Debian* reçurent les louanges de la communauté du logiciel libre, et en particulier des développeurs de Linux, qui menaient alors leur propre révolution du logiciel libre en développant le premier système d'exploitation libre utilisable. Quand la société *Netscape* a décidé de faire de son navigateur pour le web un logiciel libre, elle a contacté Eric Raymond. M. Raymond est la Margaret Meade

NdT : personnage de l'Histoire des États-Unis d'Amérique qui...

du logiciel libre : il a écrit plusieurs articles anthropologiques expliquant le phénomène du logiciel libre et la culture peu à peu l'a entouré, qui sont les premiers travaux sur le sujet et qui ont fait la lumière sur ce phénomène auparavant peu connu. La direction de la société *Netscape* fut impressionnée par l'essai de M. Raymond intitulé « [La cathédrale et le bazar](#) », chronique de la réussite du développement de logiciels libres, qui n'utilise que des contributeurs volontaires non rémunérés, et l'a engagé en tant que consultant, sous un accord de non divulgation, pendant qu'elle développait une licence pour leur propre logiciel libre. M. Raymond leur a clairement indiqué qu'il fallait que cette licence suive les lignes de conduite de *Debian* pour être prise au sérieux en tant que licence de logiciel libre.

J'ai rencontré M. Raymond de façon fortuite à la conférence des *Hackers*, réunion de programmeurs créatifs et peu conventionnels qui était réservée aux invités. Nous avons traité de divers sujet par courrier électronique. Il m'a contacté en février 1997 en me présentant l'idée de l'*Open Source*. Il regrettait alors que les hommes d'affaires, conservateurs, ne soient effrayés par le mouvement de liberté initié par M. Stallman, qui rencontrait par contraste un écho très favorable auprès des programmeurs les plus libertaires. Il pensait que cela étouffait le développement de Linux dans le monde des affaires tout en encourageant sa croissance dans le milieu de la recherche. Il a rencontré les hommes d'affaires de l'industrie balbutiante qui se formait autour de Linux, et ils ont conçu ensemble un programme pour faire la campagne du concept de logiciel libre auprès de ceux qui portent des cravates. Larry Augustin, de *VA Research*, et Sam Ockman (qui a quitté *VA* un peu plus tard pour créer la société *Penguin Computing*) ont participé à l'affaire, ainsi que d'autres personnes, que je n'ai pas la chance de connaître.

Quelques mois avant de lancer le projet de l'*Open Source*, j'avais conçu l'idée de l'*Open Hardware* (NdT : matériel dont les spécifications sont ouvertes), concept similaire, mais traitant des périphériques matériels et de leurs interfaces plutôt que de programmes et de logiciels. L'*Open Hardware* n'a pas rencontré à ce jour le même succès que l'*Open Source*, mais il suit son bonhomme de chemin et vous trouverez toutes informations le concernant à l'adresse <http://www.openhardware.org/>.

M. Raymond estimait que les *Lignes de conduite* du logiciel libre selon *Debian* était le document indiqué pour définir l'*Open Source*, mais qu'il leur fallait un nom plus général et en extirper toute référence spécifique à la distribution *Debian*. J'ai édité les *Lignes de conduite* pour mettre en place la définition de l'*Open Source*. J'avais formé une corporation pour *Debian*, intitulée *SPI (logiciel d'intérêt public)*, et j'ai proposé d'enregistrer une marque déposée pour l'*Open Source* de telle sorte qu'on puisse réserver son utilisation aux logiciels qui répondent à la définition. M. Raymond étant d'accord avec cette idée, j'enregistrai la marque de certification, ce qui est une forme particulière de marque déposée, à appliquer aux produits des autres. Un mois après que j'aie

enregistré cette marque, il était devenu clair que *SPI* n'était pas le lieu d'hébergement idéal pour la marque *Open Source*, et j'en ai transféré la propriété à M. Raymond. Depuis, nous avons créé l'*Open Source Initiative* (*initiative de l'Open Source*), organisation dont le seul but est de gérer la campagne de *l'Open Source* et sa marque de certification. Au moment de cet écrit, son conseil d'administration compte six membres choisis parmi des contributeurs renommés au monde du logiciel libre, et cherche à atteindre la taille d'environ dix membres.

Lors de son lancement, la campagne de *l'Open Source* a suscité de nombreuses critiques, même au sein du contingent de Linux, qui avait déjà accepté le concept du logiciel libre. Nombreux sont ceux qui signalèrent que le terme « *Open Source* » était déjà employé dans l'industrie de XXX . D'autres pensaient que le mot « *Open* » était déjà utilisé à toutes les sauces. D'autres, simplement, préféraient le terme *Free Software*, établi. J'ai répondu que le galvaudage du terme « *Open* » ne pourrait jamais causer autant de tort que l'ambiguïté, dans la langue anglaise, du terme « *Free* » – la *liberté* ou la *gratuité*

NdT : en anglais, le mot « *free* » signifie à la fois « libre » et « gratuit », alors que le mot « *open* » signifie « ouvert ». Consulter les essais de Richard Stallman et d'Eric Raymond pour obtenir plus d'informations à ce sujet.

, la notion de « gratuité » étant plus souvent mise en avant dans le monde commercial des ordinateurs et du logiciel. Richard Stallman a plus tard pris ombrage du fait que la campagne ne mettait pas suffisamment l'accent sur la liberté, et que le mouvement de *l'Open Source* gagnant en popularité, son rôle dans la genèse du logiciel libre, ainsi que celui de la *Free Software Foundation* qu'il a créée, étaient passés sous silence – il s'est plaint d'être « effacé des livres d'histoire ». La situation s'est aggravée par la tendance qu'ont les industriels de comparer MM. Raymond et Stallman comme s'ils défendaient des philosophies contradictoires, sans comprendre qu'en réalité ils ne font qu'utiliser des méthodes différentes pour promouvoir le même concept. J'ai probablement contribué à jeter de l'huile sur le feu en opposant MM. Stallman et Raymond dans des débats tenus à l'occasion des manifestations *Linux Expo* et *Open Source Expo*. Leurs confrontations étaient si appréciées que le journal en ligne *Salon* est allé jusqu'à publier un débat qu'ils ont tenu par courrier électronique, sans jamais avoir eu l'intention de le publier. Quand nous en fûmes rendus là, j'ai demandé à M. Raymond de calmer un jeu auquel il n'avait jamais eu l'intention de participer.

Lors de l'écriture de la définition de *l'Open Source*, nombreux déjà étaient les programmes qui la satisfaisaient. Mais un problème se posait dans le cas des programmes qui ne la satisfaisaient pas, tout en plaisant aux utilisateurs.

### **3. KDE, Qt, et Troll Tech**

Il faut traiter de *KDE*, *Qt*, et *Troll Tech* dans cet essai car le groupe *KDE* et la société *Troll Tech* ont tenté de placer un produit non conforme à *l'Open Source* dans l'infrastructure de Linux, et se sont heurtés à une résistance qu'ils n'avaient pas su prévoir. Le tollé général et la menace d'un remplacement de leur produit par une solution *Open Source* ont finalement convaincu la société *Troll Tech* de changer de licence pour choisir une licence pleinement conforme à *l'Open Source*. C'est un exemple intéressant de l'acceptation enthousiaste par la communauté de la définition de *l'Open Source*, si forte qu'il fallait que la société *Troll Tech* s'y plie s'ils souhaitaient que leur produit soit bien accueilli.

*KDE* représente la première tentative d'un bureau graphique libre pour Linux. Les applications de *KDE* étaient par elles-mêmes couvertes par la *GPL*, mais elles dépendaient d'une bibliothèque graphique propriétaire, appelée *Qt*, et développée par la société *Troll Tech*. Les conditions de la licence de *Qt* interdisaient la modification ou l'utilisation en relation avec tout logiciel d'affichage graphique autre que ne soit pas *X Window System*, qui commence à se faire vieux. Pour ces autres utilisations, il fallait acheter une licence de développeur qui

s'élevait à 1 500 USD. La société *Troll Tech* fournissait des versions de *Qt* pour les systèmes *MS-Windows* et *MacOpen Source*, et c'était là sa première source de revenus. La prétendue « licence libre » pour les systèmes fonctionnant sous « X » avait pour objectif de susciter des contributions de la part des développeurs de Linux en matière de démonstrations, exemples, et autres accessoires, qu'ils revendraient ensuite dans le cadre de leurs produits pour *MS-Windows* et *MacOpen Source*, qui n'étaient pas donnés.

Malgré la clarté des problèmes posés par la licence de *Qt*, la perspective de disposer d'un bureau graphique sous Linux était si séduisante que de nombreux utilisateurs étaient prêts à fermer les yeux sur le fait que ce n'était pas vraiment une licence de type *Open Source*. Les défenseurs de l'*Open Source* remettaient *KDE* en question car ils estimaient que les développeurs de *KDE* tentaient d'estomper la définition du logiciel libre pour y inclure des éléments qui ne sont que partiellement, comme *Qt*. Ce à quoi ces derniers rétorquaient que leurs programmes étaient *Open Source*, même s'il n'en existait aucune version exécutable qui ne nécessitât pas de bibliothèque non *Open Source*. D'autres et moi pensions que les applications de *KDE* n'étaient que des fragments *Open Source* de programmes non *Open Source*, et qu'on ne pourrait pas qualifier *KDE* d'*Open Source* avant de disposer d'une version *Open Source* de *Qt*.

Les développeurs de *KDE* ont tenté de résoudre en partie le problème posé par la licence de *KDE* en négociant un accord de *Qt* libre pour *KDE* avec *Troll Tech*, selon lequel *Troll Tech* et le groupe *KDE* contrôlèrent ensemble les publications de la version libre de *Qt*, et *Troll Tech* publierait *Qt* selon les conditions d'une licence de type *Open Source* si la société devait être rachetée ou faire banqueroute.

C'est alors qu'un autre groupe de programmeurs a démarré le projet *GNOME*, un concurrent de *KDE* pleinement *Open Source*, qui visait à être plus sophistiqué et à proposer plus de fonctionnalités, pendant qu'un troisième groupe démarrait un projet intitulé *Harmony* dans le but de produire un clone de *Qt* pleinement *Open Source*, qui permettrait de faire fonctionner *KDE*. Alors que le projet *GNOME* était applaudi de tous et que le projet *Harmony* menaçait d'être utilisable, la société *Troll Tech* a compris que *Qt* ne rendrait aucun service au marché de Linux tant que sa licence demeurerait telle. *Troll Tech* a publié une licence pleinement *Open Source* pour *Qt*, désamorçant ainsi le conflit, et laissant retomber le projet *Harmony* comme un soufflé. Mais le projet *GNOME* suit son cours, et cherche maintenant à surclasser *KDE* en termes de fonctionnalités et de sophistication, plutôt qu'en matière de licence.

Avant de publier sa licence *Open Source*, la société *Troll Tech* m'en a fourni une copie pour que je l'examine, en me demandant de n'en rien laisser filtrer jusqu'au moment où ils l'annonceraient. Trop enthousiaste d'enterrer la hache de guerre avec le groupe *KDE* et victime d'un aveuglement passager que je me reprocherais, j'ai fait une pré-annonce de leur future licence avec huit heures d'avance sur une liste de diffusion consacrée à *KDE*. Ce courrier fut presque immédiatement repris, à mon grand regret, par les principaux sites d'information en ligne, dont *Slashdot*.

La nouvelle licence de *Troll Tech* a ceci de remarquable qu'elle tire avantage d'un point faible de la définition de l'*Open Source* qui accorde aux fichiers de modifications (*patches*) un statut particulier. Je souhaiterais le corriger dans une prochaine version de la définition de l'*Open Source*, mais ce nouveau texte exclurait alors de nouveau la bibliothèque *Qt*.

Au moment où j'écris ces lignes, le nombre de défenseurs de l'*Open Source* augmente exponentiellement. Les récents apports des sociétés *IBM* et *Ericsson* au mouvement de l'*Open Source* ont eu la première page. On compte deux distributions de Linux, *Yggdrasil* et *Debian*, qui distribuent des systèmes Linux conformes en tous points à l'*Open Source*, sans pour autant se priver d'une grande richesse de logiciels, tandis que plusieurs autres distributeurs, comme *Red Hat*, n'en sont pas loin. Quand le projet *GNOME* sera disponible

NdT : la première version stable en a été annoncée en février 1999.

, on disposera enfin d'un système d'exploitation avec bureau à interface utilisateur graphique, digne de concurrencer *MS-Windows NT*, entièrement *Open Source*.

#### 4. Analyse de la définition de l'*Open Source*

Dans cette section, je présenterai le texte complet de la définition de l'*Open Source*, en y entrelaçant des commentaires (*en emphase*). Vous en trouverez la version canonique à l'adresse <http://www.opensource.org/osd.html>

NdT : une version en français est disponible à l'adresse <http://www.linux-france.org/article/these/osd/fr-osd.html>.

Des coupeurs de cheveux en quatre m'ont fait remarquer que la définition de l'*Open Source* renfermait quelques ambiguïtés. J'ai reculé l'instant d'une révision car ce texte date d'à peine un an et j'aimerais qu'il soit considéré comme un texte stable. Dans l'avenir, j'y incorporerai des modifications mineures sur la forme, qui n'auront que très peu d'incidence sur le fond.

##### 4.1 Définition de l'« *Open Source* », version 1.0

« *Open Source* » implique plus que la simple diffusion du code source. La licence d'un programme « open-source » doit correspondre aux critères suivants :

*Vous remarquerez que la définition de l'Open Source n'est pas une licence de logiciel en soi. C'est une spécification de ce qu'on autorise aux licences de logiciels pour qu'elles méritent le nom d'Open Source. La définition de l'Open Source n'a pas pour vocation d'être un document juridique. Le fait qu'on la retrouve dans des licences de logiciels, comme la licence qui est proposée pour le Linux Documentation Project (projet de documentation pour Linux), m'a fait envisager d'en écrire une version plus rigoureuse, qui serait appropriée pour une telle utilisation.*

*Pour qu'un logiciel puisse être qualifié d'Open Source, il faut que toutes les conditions suivantes soient remplies, en même temps, et dans tous les cas. Il faut par exemple qu'elles s'appliquent aux versions dérivées d'un programme aussi bien qu'au programme original. Il ne suffit pas de n'en appliquer que quelques unes et pas d'autres, et il ne suffit pas de ne les appliquer que dans certaines périodes. Comme j'ai dû ferrailer pour contrer des interprétations particulièrement naïves de la définition de l'Open Source, je serai tenté d'ajouter : cela vous concerne*

##### 1. Libre redistribution.

La licence ne doit pas empêcher de vendre ou de donner le logiciel en tant que composant d'une distribution d'un ensemble contenant des programmes de diverses origines. La licence ne doit pas exiger que cette vente soit soumise à l'acquittement de droits d'auteur ou de royalties.

*Cela signifie qu'on peut faire autant de copies du logiciel qu'on le souhaite, et les vendre ou les donner, sans devoir donner d'argent à qui que ce soit pour bénéficier de ce privilège.*

*La formule « composant d'une distribution d'un ensemble contenant des programmes de diverses origines » avait pour but de combler une lacune de la licence Artistic (artistique), dont personnellement je trouve qu'elle manque de rigueur, qui a été mise au point pour Perl, à l'origine. De nos jours, presque tous les programmes qui en font usage sont aussi proposés selon des conditions de la GPL. C'est pourquoi cette clause n'est plus nécessaire, et disparaîtra peut-être d'une prochaine version de la définition de l'Open Source.*

## 2. Code source.

Le programme doit inclure le code source, et la distribution sous forme de code source comme sous forme compilée doit être autorisée. Quand une forme d'un produit n'est pas distribuée avec le code source correspondant, il doit exister un moyen clairement indiqué de télécharger le code source, depuis l'*Internet*, sans frais supplémentaires. Le code source est la forme la plus adéquate pour qu'un programmeur modifie le programme. Il n'est pas autorisé de proposer un code source rendu difficile à comprendre. Il n'est pas autorisé de proposer des formes intermédiaires, comme ce qu'engendre un pré-processeur ou un traducteur automatique.

*Le code source est un préliminaire nécessaire à la correction ou la modification d'un programme. L'intention est ici de faire en sorte que le code source soit distribué aux côtés de la version initiale et de tous les travaux qui en dériveront.*

## 3. Travaux dérivés.

La licence doit autoriser les modifications et les travaux dérivés, et leur distribution sous les mêmes conditions que celles qu'autorise la licence du programme original.

*Le logiciel est de peu d'utilité à qui ne peut pas assurer son évolution (correction des bogues, ports vers des nouveaux systèmes, apport d'améliorations), et il est nécessaire pour cela de le modifier. L'intention est ici d'autoriser tous types de modifications. Il faut autoriser qu'un travail dérivé soit distribué sous les mêmes conditions de licence que le travail original. Cependant, on n'exige pas que le producteur d'un travail dérivé utilise les mêmes conditions de licence, on n'impose que de leur laisser la possibilité de ce faire. Les différentes licences traitent ce problème de manières diverses : la licence BSD vous autorise de privatise vos modifications, alors que GPL vous l'interdit.*

*Certains auteurs de logiciels craignent que cette clause n'autorise des gens peu scrupuleux à modifier leur logiciel de sorte à mettre dans l'embarras l'auteur original du logiciel. Ils ont peur qu'un individu mal intentionné ne fasse réagir le logiciel de manière incorrecte en laissant croire que l'auteur original était un programmeur de piètre qualité. D'autres craignent que le logiciel ne soit modifié pour des utilisations criminelles, par l'addition de fonction jouant le rôle de cheval de Troie ou de techniques interdites dans certains pays ou régions, comme la cryptographie. Mais de telles actions tombent sous le coup des lois. On pense souvent à tort que les licences de logiciels devraient tout spécifier, y compris des détails comme « n'utilisez pas ce logiciel pour commettre un crime. » Mais aucune licence n'a d'existence en dehors d'un corpus de lois civiles et pénales. Considérer qu'une licence peut s'affranchir du corpus des lois en vigueur est aussi idiot que considérer qu'un document rédigé en français puisse s'affranchir du dictionnaire, auquel cas aucun des mots utilisés n'aurait la moindre signification arrêtée.*

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

La licence ne peut restreindre la redistribution du code source sous forme modifiée que si elle autorise la distribution de fichiers « patch » aux côtés du code source dans le but de modifier le programme au moment de la construction.

*Certains auteurs craignaient que d'autres ne distribuent le code source enrichi de modifications qui pourraient être perçues comme relevant du*

*travail de l'auteur original, en donnant une mauvaise image de lui. Cette clause leur donne la possibilité d'imposer que les modifications soient bien distinctes de leur propre travail, sans pour autant interdire toute modification. Certaines personnes trouvent inesthétique le fait que les modifications encourent le risque de devoir être distribuées sous la forme d'un fichier de modifications (patch) distinct du code source, alors même que des distributions de Linux comme Debian ou Red Hat font usage d'une telle procédure pour mettre en place les modifications qu'elles apportent aux programmes qu'elles distribuent. Il existe des programmes qui fondent directement les modifications au sein du code source principal, et on peut faire en sorte qu'ils soient exécutés automatiquement lors de l'extraction d'un paquetage de code source. C'est pourquoi une telle clause ne devrait causer que peu ou prou de difficultés.*

*Vous remarquerez aussi que cette clause stipule que dans le cas des fichiers de modifications, la modification n'a lieu qu'au moment de la construction. La licence publique de Qt exploite cette lacune pour imposer une licence différente, quoique plus permissive, en matière de fichiers de modifications, en contradiction avec la section 3 de la définition de l'Open Source. Il existe le projet de corriger cette lacune dans la définition sans pour autant faire perdre à la licence Qt son état de licence Open Source.*

La licence doit explicitement permettre la distribution de logiciel construit à partir du code source modifié. La licence peut exiger que les travaux dérivés portent un nom différent ou un numéro de version distinct de ceux du logiciel original.

*Cela signifie que la société Netscape, par exemple, peut insister sur le fait qu'elle seule a le droit de donner à une version du programme le nom de Netscape Navigator (tm), alors que les versions libres du programme doivent porter un nom comme Mozilla ou autre chose encore.*

## **5. Pas de discrimination entre les personnes ou les groupes.**

La licence ne doit opérer aucune discrimination à l'encontre de personnes ou de groupes de personnes.

*Une licence proposée par les Régents de l'université de Californie, à Berkeley, interdisait qu'un programme de conception de circuits électroniques soit employé par les forces de police de l'Afrique du Sud. Ce sentiment avait beau être généreux du temps de l'apartheid, cette clause n'a plus grand sens de nos jours. Certaines personnes sont toujours coincées avec le logiciel qu'elles ont acquis sous cette condition, dont les versions dérivées doivent elles aussi porter la même restriction. Les licences Open Source ne doivent rien renfermer de tel, quelle que soit la générosité qui dicte de telles intentions.*

## **6. Pas de discrimination entre les domaines d'application.**

La licence ne doit pas limiter la champ d'application du programme. Par exemple, elle ne doit pas interdire l'utilisation du programme pour faire des affaires ou dans le cadre de la recherche génétique.

*Votre logiciel doit pouvoir être utilisé aussi bien par une clinique qui pratique des avortements que par une organisation militant contre le droit à l'avortement. Ces querelles politiques relèvent de l'Assemblée Nationale, et non pas des licences de logiciels. Certaines personnes sont même extrêmement choquées par cette absence de discrimination !*

## 7. Distribution de la licence.

Les droits attachés au programme doivent s'appliquer à tous ceux à qui le programme est redistribué sans que ces parties ne doivent remplir les conditions d'une licence supplémentaire.

*La licence doit s'appliquer automatiquement, sans exiger une quelconque signature. Malheureusement, on ne dispose d'aucun précédent juridique solide en matière de validité d'une licence applicable sans signature, quand elle passe d'une seconde à une tierce personne. Cependant, cet argument considère que la licence fait partie du droit du contrat, alors que certains argumentent qu'elle relève du droit du copyright, où on trouve des cas de jurisprudence en matière de licences ne requérant pas de signature. Il y a fort à parier que ce débat aura lieu en cour de justice d'ici quelques années, si l'on en juge par l'emploi sans cesse croissant de ce type de licences et par l'essor fulgurant du mouvement de l'Open Source.*

## 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'une distribution logicielle spécifique. Si le programme est extrait de cette distribution et utilisé ou distribué selon les conditions de la licence du programme, toutes les parties auxquelles le programme est redistribué doivent bénéficier des droits accordés lorsque le programme est au sein de la distribution originale de logiciels.

*Cela signifie que vous ne pouvez pas contraindre un produit identifié en tant qu'Open Source à être utilisé en tant que partie d'une distribution particulière de Linux, ou autre. Il doit rester libre, même séparé de la distribution logicielle avec laquelle il a été fourni.*

## 9. La licence ne doit pas contaminer d'autres logiciels.

La licence ne doit pas apposer de restrictions sur d'autres logiciels distribués avec le programme qu'elle couvre. Par exemple, la licence ne doit pas exiger que tous les programmes distribués grâce au même médium soient des logiciels « open-source ».

*Une version de GhostScript (programme de rendu de PostScript) exige que le support sur lequel est distribué ce programme ne contienne que des logiciels libres. Les licences Open Source ne permettent pas cela. Heureusement, l'auteur du programme GhostScript distribue une autre version (un peu plus ancienne) de ce programme, couverte par une licence vraiment Open Source.*

*Remarquez la différence entre **dérivation** et **agglomération**. La dérivation est le fait qu'un programme renferme en son sein une portion d'un autre programme. L'agglomération est le fait de proposer deux programmes sur le même CD-ROM. Cette section de la définition de l'Open Source traite de l'agglomération, **pas** de la dérivation. C'est la section 4 qui traite de cette dernière.*

## 10. Exemples de licences.

Les licences suivantes sont des exemples de licences que nous considérons conformes à la définition de l'« Open Source » : GNU GPL, BSD, X Consortium, et Artistic. C'est aussi le cas de la MPL.

Bruce Perens a écrit le premier brouillon de ce document sous le titre « The Debian Free Software Guidelines » (lignes de conduite pour le logiciel libre

de la *Debian*), et l'a amélioré en utilisant les commentaires des développeurs *Debian* lors d'une conférence par courrier électronique qui a duré un mois, en juin 1997. Il a ôté du document les références propres à la *Debian* pour créer la « définition de l'*Open Source* ».

*Nous aurions beaucoup de problèmes si l'une de ces licences devait être modifiée et ne plus remplir les conditions de l'*Open Source* – il nous faudrait publier immédiatement une version révisée de la définition de l'*Open Source*. Ce paragraphe relève plus d'un texte d'explication que de la définition par elle-même.*

#### **4.2 Analyse des licences et de leur concordance avec la définition de l'*Open Source***

Pour comprendre la définition de l'*Open Source*, il faut examiner de quelle manière certaines licences communes en remplissent ou non les critères.

##### **domaine public**

Une idée reçue incorrecte veut que la plupart des logiciels libres relèvent du *domaine public*. La cause en est tout simplement que l'idée du logiciel libre ou de l'*Open Source* est difficile à appréhender pour de nombreuses personnes, qui expliquent alors que ces programmes sont du domaine public car c'est là le concept le plus proche qu'ils arrivent à comprendre. Ces programmes disposent clairement, cependant, d'un copyright, et sont couverts par une licence. Il se trouve simplement que c'est une licence qui accorde plus de droits aux utilisateurs qu'ils n'en ont l'habitude.

Un programme du domaine public est un programme sur lequel son auteur a délibérément choisi de ne pas faire valoir ses droits. On ne peut pas vraiment dire qu'il est assorti d'une licence ; quiconque peut l'utiliser comme bon lui semble, car quiconque peut le traiter comme s'il lui appartenait. On peut même assujettir un programme du domaine public à une nouvelle licence, en ôtant cette version modifiée du domaine public, ou en ôter le nom de l'auteur et traiter le programme comme son propre travail.

Si vous travaillez beaucoup sur un programme du domaine public, envisager le fait d'y apposer votre propre copyright et de lui appliquer une nouvelle licence. Si vous ne souhaitez pas, par exemple, qu'un tiers ne le modifie d'une manière qu'il gardera secrète ensuite, appliquez à votre version du programme la *GPL*, ou une licence similaire. La version de laquelle vous êtes parti se trouvera encore dans le domaine public, mais votre propre version sera couverte par une licence que les autres devront prendre en compte s'ils veulent l'utiliser ou en proposer des travaux dérivés.

Vous pouvez facilement rendre un programme du domaine public privé, en déclarant un copyright et en lui appliquant votre propre licence, ou simplement en déclarant « tous droits réservés. »

#### **4.3 Les licences de logiciels libres en général**

Si vous disposez d'une collection de logiciels libres telle qu'un disque renfermant une distribution de *GNU/Linux*, vous pensez peut-être posséder les programmes qui se trouvent sur ce disque. Ce n'est pas exactement vrai. Les programmes couverts par un copyright sont la propriété du détenteur du copyright, même lorsqu'ils sont couverts par une licence *Open Source* telle que la *GPL*. La licence du programme vous octroie certains droits, et d'autres droits vous incombent selon la définition d'« utilisation juste » dans le droit du copyright.

Il est important de remarquer qu'un auteur n'est pas limité à proposer un programme sous une seule licence. On peut protéger un programme par la *GPL* et en vendre en même temps une version couverte par une licence commerciale et non *Open Source*. C'est exactement cette tactique qu'emploient de nombreuses

personnes, qui veulent qu'un programme soit *Open Source* tout en souhaitant gagner de l'argent à partir de ce dernier. Ceux qui ne souhaitent pas d'une licence *Open Source* encourent le risque de devoir rémunérer un tel privilège, assurant par là même un revenu à l'auteur.

Toutes les licences que nous allons maintenant examiner ont un point en commun : aucune ne fournit la moindre garantie. L'intention est de protéger le propriétaire du logiciel de toute poursuite en justice relative à son programme. Les programmes étant souvent offerts sans contrepartie financière, c'est là une exigence raisonnable – le programme ne garantit pas un revenu suffisant pour que son auteur puisse acquitter une assurance et les frais engagés en cas de poursuite en justice.

Si les auteurs de logiciels libres perdent ce droit de ne fournir aucune garantie et sont poursuivis en justice suite aux performances des programmes qu'ils ont écrits, ils cesseront d'écrire des logiciels libres. C'est dans notre avantage d'utilisateurs que d'aider les auteurs à protéger ce droit.

### **La licence publique générale de GNU**

Reportez-vous à l'annexe B pour trouver le texte complet de la *GPL*. La *GPL* est autant un manifeste politique qu'une licence de logiciel, et une grande partie de ce texte est dévolue à justifier les motifs qui ont poussé son auteur à la rédiger. Ce dialogue politique a déplu à certains, et c'est en partie à cause de cela que d'autres licences de logiciels libres ont été écrites. Cependant, la *GPL* a été rédigée avec le concours de professeurs de droit, elle est donc bien mieux écrite que la plupart des autres licences de son genre. Je vous encourage vivement à employer la *GPL*, ou sa variante pour bibliothèques la  *LGPL*, si vous le pouvez. Si vous choisissez une autre licence, ou écrivez la vôtre propre, vérifiez que les raisons qui vous poussent à cela sont valables. Il faut bien expliquer à ceux qui écrivent leurs propres licences que ce n'est pas là une décision à prendre à la légère. Les complications inattendues qu'une licence mal écrite peut entraîner peuvent créer, pendant plusieurs décennies, un fardeau pour les utilisateurs du logiciel.

Le texte de la *GPL* n'est pas lui-même couvert par la *GPL*. Cette licence est simple : quiconque a le droit de copier et de distribuer des copies exactes du document de cette licence, mais pas de le modifier. Il faut remarquer que le texte des licences *Open Source* n'est pas en général couvert par l'*Open Source*. Il est évident qu'une licence n'offrirait qu'une protection symbolique si tout un chacun pouvait la modifier.

Les dispositions de la *GPL* satisfont la définition de l'*Open Source*. La *GPL* n'exige aucune des clauses autorisées par le paragraphe 4 de la définition de l'*Open Source*, « Intégrité du code source de l'auteur. »

La *GPL* ne vous autorise pas à *rendre des modifications secrètes*. Vos modifications doivent elles aussi être distribuées selon les termes de la *GPL*. Ainsi, l'auteur d'un programme couvert par la *GPL* recevra probablement des améliorations proposées par d'autres, y compris des sociétés commerciales qui modifient son logiciel pour leurs besoins propres.

La *GPL* n'autorise pas le fait d'incorporer un programme couvert par la *GPL* au sein d'un programme propriétaire. La définition que donne la *GPL* d'un programme propriétaire est : « tout programme dont la licence vous donne moins de droits que ne vous en donne la *GPL*. »

La *GPL* contient quelques échappatoires qui lui permettent d'être utilisée dans des programmes qui ne sont pas entièrement *Open Source*. On peut lier les bibliothèques de logiciels qui sont distribuées avec le compilateur ou le système d'exploitation avec des logiciels couverts par la *GPL* ; il en résulte un programme partiellement libre. Le détenteur du copyright (qui est en général l'auteur du programme) est la personne qui place la *GPL* sur son programme et qui dispose du droit de violer sa propre licence. C'est ce que faisaient les auteurs de *KDE* pour distribuer leurs programmes avec Qt avant que la société *Troll Tech*

ne place la bibliothèque Qt sous une licence *Open Source*. Cependant, ce droit ne s'étend pas aux tiers qui redistribuent le programme – ils doivent suivre tous les termes de la licence, même ceux que le détenteur du copyright viole, c'est pourquoi il est problématique de redistribuer un programme couvert par la *GPL* s'il contient Qt. Les développeurs de *KDE* semblent avoir résolu ce problème en appliquant la *LGPL*, et non la *GPL*, à leurs logiciels.

La rhétorique politique de la *GPL* déplaît à certaines personnes, dont une partie ont choisi pour leurs logiciels une licence moins appropriée pour la seule raison qu'ils fuient les idées de Richard Stallman et ne souhaitent pas les voir répétées dans leurs propres paquetages de logiciels.

### **La licence publique générale de GNU pour les bibliothèques**

La *LGPL* est dérivée de la *GPL* et mise au point pour les bibliothèques de logiciels. À la différence de ce qui se produit avec la *GPL*, on peut incorporer un programme couvert par la *LGPL* dans un programme propriétaire. La bibliothèque du langage *C* fournie avec les systèmes *GNU/Linux* est un exemple de logiciel couvert par la *LGPL* – on peut l'utiliser pour construire des programmes propriétaires, sans quoi Linux ne serait utile qu'aux auteurs de logiciels libres.

On peut convertir une instance d'un programme couvert par la *LGPL* en programme couvert par la *GPL* à tout moment. Quand cela s'est produit, on ne peut plus reconvertir cette instance, ou tout ce qui en a dérivé, en programme de nouveau couvert par la *LGPL*.

Les autres dispositions de la *LGPL* sont semblables à celles de la *GPL* – en fait, cette licence inclut la *GPL* en y faisant référence.

### **Les licences X, BSD, et Apache**

La licence *X* et les licences qui lui sont proches – les licences *BSD* et *Apache* – sont très différentes de la *GPL* et de la *LGPL*. Ces licences vous autorisent à tout faire avec les logiciels qu'elles couvrent. Cela, parce que les logiciels que les licences *X* et *BSD* couvraient à l'origine étaient financés par des bourses du gouvernement des États-Unis d'Amérique. Puisque les citoyens des États-Unis d'Amérique avaient déjà payé pour ces logiciels une fois, avec leurs impôts, ils ont reçu la permission d'utiliser ces logiciels comme bon leur semblait.

La permission la plus importante, qu'on ne trouve pas dans la *GPL*, et qu'on peut rendre secrètes des modifications apportées à un programme couvert par la licence *X*. En d'autres termes, vous pouvez récupérer le code source d'un programme couvert par la licence *X*, le modifier, et vendre ensuite des versions binaires de ce programme sans en distribuer le code source correspondant, et sans appliquer la licence *X* à ces modifications. Cela n'en reste pas moins de l'*Open Source*, car la définition de l'*Open Source* n'exige pas que les modifications soient couvertes par la licence originale.

Nombreux furent les autres développeurs qui ont adopté la licence *X* et ses variantes, parmi lesquels on remarquera en particulier la *BSD* (*distribution du système de Berkeley*) et le projet de serveur pour le web *Apache*. Une spécificité ennuyeuse de la licence *BSD* est qu'une de ses dispositions exige qu'on mentionne (généralement dans une note de bas de page) que le logiciel a été développé à l'université de Californie à chaque fois qu'on fait la publicité d'un aspect d'un programme couvert par la licence *BSD*. Garder la trace des logiciels couverts par la licence *BSD* dans un ensemble aussi vaste qu'une distribution *GNU/Linux*, et penser à mentionner l'université de Californie à chaque fois qu'on mentionne l'un de ces programmes dans une publicité, sont des tâches trop lourdes pour ceux qui sont dans les affaires. Au moment où j'écris ces lignes, la distribution *Debian GNU/Linux* compte plus de 2 500 paquetages logiciels, et même si seule une fraction d'entre eux était couverte par la licence *BSD*, toute publicité pour un système *GNU/Linux* comme la distribution *Debian* impliquerait des pages et des pages de notes ! Cependant, la licence du *Consortium X* ne comporte pas cette

clause relative à la publicité. Si vous envisagez d'utiliser une licence de la famille *BSD*, choisissez plutôt la licence *X*.

### **La licence *Artistic***

Même si cette licence, à l'origine, a été développée pour Perl, elle a depuis été employée pour d'autres logiciels. Je pense que c'est une licence rédigée sans soin, en ce sens qu'elle pose des exigences et qu'elle donne ensuite des échappatoires pour les contourner. C'est peut-être la raison pour laquelle presque tous les logiciels couverts par la licence *Artistic* sont désormais couverts par deux licences, offrant le choix entre la licence *Artistic* et la *GPL*.

La section 5 de la licence *Artistic* interdit la vente du logiciel, mais autorise la vente d'une distribution contenant plusieurs logiciels. Ainsi, il suffit d'emballer un programme couvert par la licence *Artistic* dans un ensemble contenant également un « *Hello world!* » de cinq lignes écrit en *C* pour avoir le droit de vendre ce paquet. C'est cette spécificité de la licence *Artistic* qui était la seule cause de l'échappatoire « composant d'une distribution d'un ensemble... » du premier paragraphe de la définition de l'*Open Source*.

L'utilisation de la licence *Artistic* déclinant, nous pensons ôter cette précision. Cela ferait de la licence *Artistic* une licence non *Open Source*. Ce n'est pas là une décision à prendre à la légère, et nous y penserons et en débattrons probablement pendant plus d'un an avant de la prendre.

La licence *Artistic* vous oblige à libérer les modifications que vous pouvez être amené à faire, tout en vous proposant une échappatoire (dans la section 7) qui vous autorise à garder secrètes des modifications, ou même à placer des portions d'un programme couvert par la licence *Artistic* dans le domaine public !

### **La licence publique de Netscape et la licence publique de Mozilla**

La *NPL* a été développée par la société *Netscape* quand elle a rendu son produit, *Netscape Navigator*, *Open Source*. En réalité, la version *Open Source* s'appelle *Mozilla* ; les gens de *Netscape* réservent la marque *Navigator* à leur propre produit. Eric Raymond et moi intervenîmes en tant que consultants non rémunérés tout au long du développement de cette licence. J'ai tenté, sans succès, de persuader la société *Netscape* d'utiliser la *GPL*, et lors de leur déclin, je les ai aidés à composer une licence qui remplirait les conditions de la définition de l'*Open Source*.

La *NPL* a ceci de particulier qu'elle renferme des privilèges particuliers, dont seul jouit la société *Netscape*. Elle lui accorde le droit de placer des modifications apportées à leur logiciel par un tiers sous une autre licence. Les gens de *Netscape* peuvent rendre ces modifications secrètes, les améliorer, et refuser de communiquer le résultat. Cette clause était nécessaire car quand la société *Netscape* a décidé de placer son produit principal sous une licence *Open Source*, elle était sous contrat avec d'autres sociétés qui ont exigé d'obtenir le logiciel *Navigator* sous une licence non *Open Source*.

La société *Netscape* a créé la *MPL*, ou *licence publique de Mozilla*, pour régler ce problème. La *MPL* ressemble beaucoup à la *NPL*, mais ne contient pas la clause qui autorise *Netscape* à placer des modifications extérieures sous une autre licence.

La *NPL* et la *MPL* vous autorisent à rendre des modifications secrètes.

De nombreuses sociétés ont adopté une variante de la *MPL* pour leurs propres programmes. C'est regrettable, car la *NPL* a été créée pour la situation professionnelle particulière dans laquelle la société *Netscape* se trouvait alors, et elle n'est pas forcément appropriée à d'autres usages. Il vaut mieux qu'elle demeure la licence de *Netscape* et de *Mozilla*, et que les autres utilisent la *GPL* ou la licence *X*.

## **4.4 Choisir une licence**

*N'écrivez pas* une nouvelle licence si vous pouvez employer l'une de celles qui

sont listées ici. La propagation de licences nombreuses et incompatibles cause du tort au logiciel *Open Source* car on ne peut pas utiliser des fragments d'un programme au sein d'un autre programme si les deux licences qui les couvrent sont incompatibles.

Abstenez-vous d'utiliser la licence *Artistic* à moins que vous ne prévoyiez de l'étudier attentivement et d'en ôter les échappatoires. Puis, faites quelques décisions :

1. Souhaitez-vous qu'on puisse *rendre des modifications secrètes* ou non ? Si vous souhaitez pouvoir obtenir, de la part de ceux qui les ont apportées, le code source des modifications, utilisez une licence qui exige cela. La *GPL* et la *LGPL* sont de bons candidats. Si cela ne vous ennuie pas qu'on puisse modifier votre programme sans vous en rendre compte, utilisez les licences *X* ou *Apache*.
2. Souhaitez-vous que quelqu'un puisse faire fusionner votre programme avec son propre logiciel, propriétaire ? Si c'est le cas, utilisez la *LGPL*, qui autorise explicitement cela sans permettre à autrui d'apporter des modifications à votre propre code sans les publier, ou utilisez les licences *Apache* ou *X*, qui autorisent le fait de rendre des modifications secrètes.
3. Souhaitez-vous qu'on puisse vendre des versions de votre programme sous licence commerciale, non *Open Source* ? Si c'est le cas, distribuez votre programme sous les termes de deux licences. Je recommande la *GPL* en tant que licence *Open Source* ; vous pourrez trouver une licence commerciale appropriée dans des livres tels que *Copyright Your Software (protégez vos logiciels grâce au copyright)*, édité par *Nolo Press*.
4. Souhaitez-vous que *tous ceux* qui utilisent votre programme acquittent une somme pour bénéficier de ce privilège ? Si c'est le cas, alors peut-être que l'*Open Source* n'est pas pour vous. Si vous trouvez satisfaction dans le fait que certains utilisateurs vous donnent de l'argent, c'est toujours possible en publiant votre programme sous une licence *Open Source*. La plupart des auteurs de logiciels *Open Source* considèrent que leurs programmes sont des contributions au bien public, et se fichent de recevoir ou non de l'argent pour ces derniers.

Le tableau ci-dessous propose une comparaison des pratiques des différentes licences :

- Licence

1. Le programme peut être mélangé avec du logiciel non libre
2. Les modifications peuvent être rendues secrètes et non communiquées à l'auteur
3. Tout un chacun peut placer le programme sous une nouvelle licence
4. La licence renferme des privilèges particuliers pour le détenteur original du copyright, qui peuvent s'appliquer à vos modifications

/GPL/ gnu public Licence	o rien			
LGPL Lesser Gnu Public License	1. oui			
BSD	1. oui	2. oui		
NPL netscape Public License	1. oui	2. oui	3. non	4. oui
MPL Mozilla Public Licence	1. oui	2. oui		
domaine public	1. oui	2.oui	3.oui	

- *GPL*

- 1. rien
- *LGPL*
  - 1. Oui
- *BSD*
  - 1. Oui
  - 2. Oui
- *NPL*
  - 1. Oui
  - 2. Oui
  - 3. non
  - 4. Oui
- *MPL*
  - 1. Oui
  - 2. Oui
- domaine public
  - 1. Oui
  - 2. Oui
  - 3. Oui

#### 4.5 L'avenir

Au moment où cet essai était sur le point de partir sous presse, la société *IBM* a rejoint le monde de l'*Open Source*, et la communauté du capital-risque découvre elle aussi l'*Open Source*. Les sociétés *Intel* et *Netscape* ont investi dans la société *Red Hat*, qui distribue *Linux*. La société *VA Research*, qui intègre des serveurs *Linux* sur des stations de travail, a annoncé qu'elle disposait d'un investisseur extérieur. La société *Sendmail Inc.*, créée pour commercialiser *sendmail*, programme de distribution du courrier électronique qu'on trouve partout, a annoncé qu'elle disposait de six millions d'USD de fonds. Le programme de courrier électronique, *Postfix*, de la société *IBM*, est proposé selon les termes d'une licence *Open Source*, et un autre programme de la société *IBM*, le compilateur *Jikes* pour le langage *Java*, est proposé selon les termes d'une licence qui tente, à l'heure où j'écris ces lignes, de remplir les clauses de la définition de l'*Open Source*, sans y parvenir tout à fait. La société *IBM* semble vouloir modifier la licence de *Jikes* pour que ce programme soit vraiment *Open Source*, et rassemble les commentaires de la communauté en ce moment même.

Deux rapports internes à la société *Microsoft*, plus connus sous le nom de *documents de Halloween*, ont fui et ont été portés à la connaissance de tous. Ces rapports documentent clairement le fait que la société *Microsoft* se sent menacée par le mouvement de l'*Open Source* et par *Linux*, et que la société *Microsoft* les attaquera tous deux pour protéger ses marchés. Il est évident que nous vivons une époque intéressante. Je pense que vous verrez la société *Microsoft* utiliser ses deux stratégies principales : placer les interfaces sous copyright, et protéger ses programmes à l'aide de brevets. Elle étendra les protocoles réseau, en y incluant des spécificités propres à *Microsoft*, qui ne seront pas disponibles pour le logiciel libre. Elle explorera, et d'autres sociétés avec elle, de nouveaux axes de recherche en informatique et brevètera tout ce qu'elle pourra avant que la communauté du logiciel libre ne puisse employer ces nouvelles techniques, et elle nous en maintiendra à distance respectueuse en exigeant le versement de sommes conséquentes pour avoir le droit d'utiliser ces brevets. J'ai écrit un essai pour le magazine sur le web *Linux World* expliquant comment combattre les ennemis de l'*Open Source* sur le front des brevets.

La bonne nouvelle, c'est que la société *Microsoft* a peur ! Dans le deuxième document *Halloween*, un employé de cette société s'extasie sur le sentiment qu'il a éprouvé en constatant qu'il pouvait facilement modifier des portions du système *Linux* pour lui faire remplir ses besoins, et que cela était bien plus facile dans le cas de *Linux* que ce ne l'était pour un employé de *Microsoft* de modifier *MS-Windows NT* !

Les coups de bélier provenant de l'intérieur sont les plus dangereux. Je pense que vous verrez également de plus en plus de tentatives de diluer la définition

de l'*Open Source* pour lui faire inclure des produits partiellement libres, comme on l'a vu dans le cas de la bibliothèque *Qt* de *KDE* avant que la société *Troll Tech* n'ait été éclairée et n'ait fourni une licence *Open Source*. La société *Microsoft*, comme d'autres, pourraient nous causer du tort en produisant énormément de logiciels, suffisamment libres pour attirer les utilisateurs, sans pour autant proposer toutes les libertés de l'*Open Source*. On peut concevoir qu'ils pourraient tuer le développement de certaines catégories de logiciels *Open Source* en fournissant des solutions « assez bonnes » et « presque assez libres ». Cependant, la forte réaction à l'encontre du projet *KDE*, avant que la bibliothèque *Qt* ne soit pleinement *Open Source*, laisse présager que de telles tentatives de la part de la société *Microsoft* et de ses semblables échoueront.

Jusqu'à présent, nous avons échappé aux chevaux de Troie. Mais supposons que quelqu'un qui ne nous aime pas propose un logiciel renfermant un cheval de Troie, une manière cachée de casser la sécurité d'un système *GNU/Linux*. Supposons ensuite que cette personne patiente jusqu'au moment où son logiciel sera bien répandu avant de rendre publique la faille et sa vulnérabilité. Le grand public aura alors vu que notre système, l'*Open Source*, peut nous laisser plus démunis face à ce type d'attaque que ne le sont les systèmes fermés de la société *Microsoft*, ce qui risque de porter un coup à sa confiance dans les logiciels *Open Source*. On pourra rétorquer que la société *Microsoft* a son compte de problèmes de sécurité, et qu'elle n'a pas besoin pour les insérer de gens extérieurs mal intentionnés, et que le modèle de l'*Open Source*, exposant le code source des programmes à la vue de tous, facilite la découverte de tels bogues. Tout bogue de ce type se produisant sur *Linux* sera corrigé le lendemain de sa publication, alors qu'un tel problème sous *MS-Windows* peut demeurer indécelé ou non corrigé pendant des années. Ils nous faut cependant améliorer notre défense face aux chevaux de Troie. Bien identifier ceux qui proposent des logiciels et des modifications représente notre meilleure défense, et cela nous permet de poursuivre en justice ceux qui proposent des chevaux de Troie. Quand je dirigeais la distribution *Debian GNU/Linux*, nous avons mis en place un système pour identifier de manière sûre tous ceux qui maintenaient des logiciels et pour qu'ils prennent part dans un réseau de cryptographie à clef publique qui nous permettrait de vérifier de qui provenait chaque logiciel. Ce type de système a pris de l'ampleur pour finalement inclure tous les développeurs de logiciels *Open Source*.

Il nous faut encore faire des améliorations gigantesques avant que *Linux* puisse être employé par l'utilisateur moyen. Nous manquons clairement d'interfaces graphiques, et ce sont les projets *KDE* et *GNOME* qui traitent ce déficit. Notre prochain but sera alors de faciliter l'administration du système : le logiciel *linuxconf* a beau régler partiellement cela, il est loin de proposer à l'utilisateur débutant un outil exhaustif d'administration système. Si le système *COAS* de la société *Caldera* tient ses promesses, il pourrait servir de base à une solution complète d'administration. Malheureusement, la société *Caldera* n'a pas pu allouer des ressources suffisantes pour que le développement du projet *COAS* soit mené à terme, et d'autres participants ont abandonné le projet, démotivés par une progression trop lente, voire inexistante.

La pléthore de distributeurs de *GNU/Linux* semble subir un écrémage, dont la société *Red Hat* sortira comme vainqueur annoncé, talonnée par la société *Caldera*. La société *Red Hat* a montré jusqu'à un solide attachement au concept de l'*Open Source*, mais l'avènement d'un nouveau président et les rumeurs d'une offre publique d'achat (*OPA*) pourraient être le signe d'un affaiblissement de cet attachement, surtout si des concurrents comme la société *Caldera*, qui ont manifesté bien moins de marques d'intérêt dans l'*Open Source*, entament les marchés de *Red Hat*. Si l'attachement des distributions commerciales de *GNU/Linux* au modèle de l'*Open Source* devait par trop s'affaiblir, on verrait vraisemblablement se mettre en place des efforts pour les remplacer par des distributions entièrement *Open Source*, telles que *Debian GNU/Linux*, plus ciblées vers les marchés commerciaux que cela n'a été le cas de la distribution *Debian*.

Malgré tous ces défis, je prédis que l'*Open Source* remportera la victoire. Le système *GNU/Linux* est devenu le système de tests des étudiants en informatique,

et ces derniers introduiront ces systèmes libres dans leur entreprise, quand ils auront obtenu leur diplôme. Les laboratoires de recherche ont adopté le modèle de l'*Open Source* car le partage de l'information est essentiel dans la méthode scientifique, et l'*Open Source* permet de partager facilement le logiciel. Les entreprises adoptent elles aussi le modèle de l'*Open Source* car il permet à des groupes de sociétés de collaborer pour résoudre un problème sans craindre une poursuite en justice pour situation de monopole, et à cause du gain apporté par les contributions libres à leurs logiciels des programmeurs du grand public. Certaines grandes sociétés ont adopté l'*Open Source* en tant que stratégie pour combattre la société *Microsoft* et s'assurer qu'aucun autre *Microsoft* ne dominera jamais plus l'industrie de l'informatique. Mais c'est dans le passé qu'il faut chercher l'indication la plus fiable de l'avenir de l'*Open Source* : en à peine quelques années, on est passé de presque rien à un solide fonds de logiciels qui résolvent de nombreux problèmes différents, et on est en passe d'atteindre le million d'utilisateurs. Nous n'avons aucune raison de lever le pied maintenant.