

Petit guide sur les logiciels libres à l'intention des professionnels et des juristes

Stephen J. Davidson*, Leonard, Street and Deinard, Minneapolis, Minnesota (États-Unis d'Amérique) Copyright 2004, Stephen J. Davidson, Tous droits réservés

Table des matières

<u>Sujet</u>	<u>Page</u>
Une composante de l'écosystème	1
Quelques définitions utiles	2
Réalités économiques et pratiques	3
La conception	4
Les systèmes de licence	4
À propos des standards	6
Le coût total de possession	7
La licence publique générale et la licence publique générale limitée	8
Conclusion	13

* Stephen Davidson est responsable du Service du droit de la propriété intellectuelle et des technologies de l'information du cabinet d'avocats Leonard, Street and Deinard, basé à Minneapolis (État du Minnesota), au sein duquel il représente des clients dans l'ensemble des États-Unis dans le cadre de projets de développement et dans des transactions et des litiges en rapport avec l'informatique et les technologies de l'information. Il a été président de l'Association du droit de l'informatique et professeur adjoint de droit des technologies de l'information à la faculté de droit de l'Université du Minnesota. M. Davidson a été intervenant dans de nombreuses conférences sur le droit de l'informatique aux États-Unis d'Amérique, au Canada, en Europe, en Asie et en Amérique latine et a fait partie du comité de rédaction de plusieurs revues de droit de l'informatique importantes. On peut prendre contact avec lui à l'adresse suivante : steved@leonard.com.

Une composante de l'écosystème

Le logiciel libre[†] est l'une des composantes d'un écosystème informatique qui offre aux concepteurs et aux utilisateurs de logiciels un mode non conventionnel de conception et de distribution des logiciels. Il coexiste, dans cet environnement, avec un large éventail d'autres méthodes de conception et de distribution de logiciels, notamment les logiciels du domaine public, les logiciels gratuits, les partagiciels, les logiciels protégés commerciaux, et même les logiciels fantômes. Les logiciels libres peuvent être des outils logiciels, des systèmes d'exploitation ou d'autres applications[‡].

Le principal avocat de la cause du logiciel libre est la Free Software Foundation, créée vers le milieu des années 1980 par l'informaticien Richard Stallman. Cette association se donne pour mission : 1) de défendre le droit des utilisateurs d'ordinateurs d'utiliser, d'étudier, de copier, de modifier et de diffuser à nouveau des programmes d'ordinateurs; 2) de promouvoir la conception et l'utilisation de logiciels et de documentation gratuits; 3) de sensibiliser le public aux aspects éthiques et politiques de la question de la liberté d'utilisation des logiciels; 4) de concevoir de nouveaux logiciels gratuits; 5) de "faire de ces logiciels un système cohérent permettant d'éliminer le besoin d'avoir recours à des logiciels protégés[§].

Le présent exposé a pour but d'aider le lecteur à comprendre les différences fondamentales qui existent entre les logiciels libres et les logiciels commerciaux du point de vue des incidences, sur le plan pratique, de l'utilisation de l'un ou l'autre type de logiciel. Il présente également quelques-unes des questions plus générales qui devraient être l'objet d'une réflexion de la part des concepteurs de logiciels, des pouvoirs publics et des entreprises afin, s'agissant de leurs propres politiques générales et de leurs activités d'achat, qu'ils prennent des décisions qui tiennent compte non seulement des coûts et des avantages immédiats qui découleront de leurs choix, mais aussi des conséquences à long terme qu'auront ces décisions qu'ils prennent aujourd'hui sur leur économie et leur place dans le marché mondial de la technologie de demain.

Quelques définitions utiles

[†] En anglais, le mot "source", contenu dans le terme qui désigne les logiciels libres, "open source software" – littéralement, "logiciel à code ouvert" –, fait référence au code source des programmes d'ordinateurs. Les programmes d'ordinateurs ou les systèmes d'exploitation sont, à l'origine, écrits en langage de programmation par un être humain. C'est ce que l'on appelle le code source du logiciel. Pour qu'un ordinateur puisse, dans la pratique, utiliser ce programme, il doit traduire ce dernier, à partir du code source, dans le langage machine qu'il comprend et qu'il peut exécuter. On appelle ce processus de traduction la "compilation" (<http://iet.ucdavis.edu/glossary.cfm#s>). Le code source est le texte que les programmeurs saisissent puis modifient de manière à ce que les programmes ainsi conçus puissent être exécutés. Pour modifier un programme, le programmeur modifie son code source pour ensuite créer un nouveau programme à partir de ce code source modifié. Sans code source, il n'est, en général, pas possible de modifier ou de réparer un programme au-delà de certaines limites étroites prévues par son auteur original. Voir (<http://www.google.com/search?hl=en&lr=&oi=defmore&q=define:source+code>).

[‡] Un répertoire de logiciels libres peut être consulté à l'adresse <http://directory.fsf.org/>.

[§] Voir <http://www.fsf.org/fsf/fsf.html>.

Les logiciels commerciaux ou protégés ne sont, traditionnellement, vendus qu'en version binaire et exécutable, les concepteurs se réservant la prérogative d'en connaître le code source, de les modifier, de les vendre et d'autoriser des tiers à accomplir ces actes. Il n'est pas rare pour des concepteurs de logiciels protégés de parler du code source de leurs logiciels comme des "plus beaux fleurons" de leur entreprise et de les protéger jalousement contre toute divulgation à des tiers. La distinction fondamentale entre les logiciels libres et les logiciels protégés réside dans le fait que le code source des premiers est disponible pour quiconque reçoit une licence permettant de les utiliser et que, bien souvent, ils font l'objet d'une autorisation de portée générale de les modifier et de les rediffuser, tant en version binaire qu'en version code source.

Le jeu des forces du marché a eu pour effet, ces dernières années, de faire apparaître de nombreuses variantes de logiciels commerciaux et de logiciels libres ayant chacune leurs caractéristiques propres. Cet éventail de logiciels disponibles comprend :

1. des logiciels commerciaux protégés dont le code source n'est accessible qu'au concepteur initial;
2. des logiciels commerciaux protégés dont le code source est l'objet d'une licence d'utilisation accordée à des utilisateurs autorisés à condition qu'ils le tiennent strictement confidentiel et ne s'en servent que pour entretien ou modification du logiciel à leur usage personnel;
3. des partagiciels, dont le code source est accessible aux titulaires de licences à des fins bien précises tout en étant l'objet de restrictions en matière d'utilisation et de divulgation; des logiciels "communautaires", dont le code source est accessible à une communauté bien déterminée d'utilisateurs qui peuvent l'utiliser à des fins générales; il est néanmoins, dans ce cas, l'objet de restrictions en matière d'utilisation, de modification et de diffusion;
4. des logiciels que l'on peut véritablement qualifier de libres, dont le code source est accessible et peut être librement utilisé, modifié et diffusé, mais dont la licence d'utilisation peut être assortie de conditions et d'obligations qui font que ces logiciels ne se prêtent pas à un usage commercial (nous y reviendrons). Ces catégories, sous l'action des forces du marché, ont tendance à se chevaucher, et les distinctions entre les divers types de logiciels à se brouiller, mais elles n'en restent pas moins suffisamment marquées pour être très débattues. Ce débat a été en large partie suscité par l'émergence du mouvement en faveur du logiciel libre, dont les adeptes défendent la cause du logiciel gratuit avec une ferveur quasi religieuse.

Malgré les différences évidentes que présente le logiciel libre avec les logiciels protégés, les partagiciels et les logiciels à code source communautaire, il n'est pas aisé d'en donner une définition précise. Une recherche Google® récente a livré pas moins de dix définitions provenant de diverses sources, chacune d'entre elles reflétant un point de vue légèrement différent des autres sur la question **. La définition suivante a le mérite de la clarté et de la concision :

** Voir <http://www.google.com/search?hl=en&lr=&oi=defmore&q=define:Open+source+software>.

Un logiciel libre est un logiciel dont le code de programmation sur lequel il repose est accessible à tous les utilisateurs, ces derniers étant libres de le lire, de le modifier et de créer de nouvelles versions de ce logiciel intégrant ces modifications. Il existe de nombreux types de logiciels libres, ces derniers se distinguant essentiellement par les conditions de la licence d'utilisation ayant trait à la possibilité (ou à l'obligation) de diffuser à nouveau des copies (modifiées) du code source^{††}.

Réalités économiques et pratiques

Parmi les différences les plus importantes, sur le plan pratique, entre les logiciels commerciaux et les logiciels libres, citons la manière dont ils sont conçus, le système de licence régissant leur diffusion, leur degré de conformité avec les normes et ce qu'il est convenu d'appeler leur coût total de possession. Bien qu'une étude détaillée de ces questions sorte du cadre de présent exposé et ait fait l'objet de nombreuses conférences de plusieurs jours au cours des dernières années, il convient de formuler ici quelques remarques générales.

La conception

Les logiciels libres sont, le plus souvent, conçus par des particuliers très doués ou par des groupes ou "communautés" informels de programmeurs souhaitant résoudre une difficulté technique et partager le résultat de leurs travaux avec le reste du monde. Certains tenants du modèle du logiciel libre soutiennent qu'il engendre un plus haut niveau d'innovation, d'autres affirment qu'il produit des logiciels techniquement égaux voire supérieurs à leurs concurrents commerciaux. Le mode de diffusion fait qu'il peut exister de multiples variantes d'un même produit logiciel libre, y compris de nombreux dérivés sur lesquels on a relativement peu de documentation et qui peuvent présenter de légères – et parfois même de grandes – différences sur le plan du fonctionnement.

Les logiciels commerciaux ou protégés sont généralement conçus et mis au point pour répondre à un besoin du marché, ou à tout le moins à ce qui est perçu comme étant un besoin du marché. Leurs caractéristiques ont tendance à être dictées par le marché et par les utilisateurs. La conception de tels logiciels se fait selon un processus relativement bien défini et méthodique et, généralement, les produits qui en résultent sont accompagnés d'une documentation adéquate, ont été soumis à des tests de qualité et bénéficient d'un soutien technique. Les caractéristiques des logiciels commerciaux sont généralement le fruit d'une longue évolution qui a été dictée par le marché. L'utilisateur, bien entendu, paie pour l'ensemble de ces prestations, qu'il en veuille ou non.

^{††} Voir www.cross-web.com/information/glossary.htm.

Les systèmes de licence

Les concepteurs de logiciels commerciaux ont recours à des systèmes de licence ayant les caractéristiques suivantes : 1) ils leur permettent d'exploiter leur propriété intellectuelle de manière à générer des recettes suffisantes pour financer leurs activités de recherche-développement et de commercialisation et pour absorber leurs autres coûts tout en dégagant un bénéfice, modeste dans certains cas, colossal dans d'autres; 2) ils limitent le champ d'utilisation du produit et sa transmissibilité, interdisent l'ingénierie inverse et limitent les garanties et la responsabilité. Les utilisateurs, quand ils utilisent des logiciels protégés, paient pour les avantages qu'ils en retirent, mais aussi parfois pour des avantages dont ils n'ont pas besoin. Il n'en reste pas moins que le principe économique fondamental sur lequel repose l'octroi de licences d'utilisation de logiciels est que chaque partie en retire l'avantage escompté au prix escompté.

Cela est également vrai des logiciels libres, mais leur système de licence est très différent. Le terme logiciel libre ne signifie pas nécessairement que l'on soit libre de faire ce que l'on veut avec ce type de logiciel, même si c'est parfois le cas. Deux modèles prévalent en la matière. Le premier, appelé modèle BSD (BSD étant l'acronyme de Berkeley Software Distribution), est très souple et permet à l'utilisateur de faire à peu près ce qu'il veut avec les logiciels qui en relèvent, notamment de les modifier et d'en diffuser des dérivés gratuits ou protégés, à condition que chaque copie soit accompagnée d'une licence dont la nature est précisée et qui comprenne une mention de réserve du droit d'auteur et des clauses limitatives de garantie et de responsabilité. Le système d'exploitation UNIX est un exemple de logiciel vendu sous une licence de type BSD (entre autres modes d'octroi), tandis que le système d'exploitation Solaris®, fabriqué par Sun Microsystems, est un exemple de dérivé protégé du système UNIX BSD.

Le deuxième modèle dominant, s'il permet de librement utiliser, modifier et diffuser à nouveau les logiciels qui en relèvent, est très restrictif. Les licences de ce type les plus courantes sont la Licence publique générale (General Public License) (GPL) de la Free Software Foundation et la Licence publique générale limitée (Lesser General Public License) (LGPL), qui autorisent la modification et la diffusion de dérivés gratuits des logiciels mais interdisent d'en créer des versions protégées. Le système d'exploitation LINUX est un exemple de logiciel libre dérivé du système UNIX. De nombreux outils logiciels tels que le compilateur GNU C sont également vendus sous une licence de type GPL.

Un autre type de licence libre est la licence communautaire, dont les conditions peuvent être très différentes d'une "communauté" à l'autre. Ainsi, une licence émanant d'une communauté universitaire peut être très souple, tandis que, par exemple, la licence communautaire de Sun Microsystems pour la plateforme Java® comporte des restrictions très importantes. Les lecteurs qui souhaiteraient approfondir la question pourront consulter le site Internet d'Open Source Initiative, qui donne plus de 50 exemples de licences libres actuellement utilisées^{‡‡}.

^{‡‡} <http://www.opensource.org/licenses/bsd-license.php>.

À propos des standards

Deux des particularités des logiciels protégés qui contribuent à leur protection sont la faculté qu'ont leurs propriétaires de conserver des droits de propriété intellectuelle attachés à leurs éléments distinctifs et de rester maître de leurs caractéristiques. Cette faculté peut s'étendre aux interfaces de programmes d'application (API) ou à d'autres aspects des logiciels ayant une incidence sur l'interopérabilité. C'est ce qui explique la tendance de certains défenseurs des logiciels libres à assimiler "code source ouvert" à "norme ouverte". On pourrait, cependant, avancer que les normes se constituent essentiellement de deux manières, l'une étant à travers l'adoption généralisée d'une caractéristique par le marché des utilisateurs, l'autre étant par le recours à des organismes de normalisation.

En réalité, presque toutes les normes ouvertes importantes ont été mises au point par des groupements d'entreprises du secteur privé. Il est aussi un fait que la nature même des logiciels ouverts, la relative facilité avec laquelle ils peuvent, en toute liberté, être modifiés, fait qu'ils ont tendance à s'écarter du standard. Du reste, les distributeurs commerciaux de logiciels libres les modifient souvent à dessein afin de se différencier les uns des autres. Ainsi, bien que les sociétés IBM, SCO Group, Red Hat, et Sun Microsystems commercialisent toutes des dérivés du système d'exploitation UNIX, ces derniers ne se ressemblent pas du tout et n'ont rien de "standard".

Le coût total de possession

Le terme "code source ouvert" est souvent considéré comme étant synonyme de "logiciel libre" ("free software" en anglais), et de nombreux tenants du système du code source ouvert font valoir que l'un des grands avantages de ce dernier est qu'il permet d'acquérir des logiciels sans avoir à payer de droits de licence. Cependant, comme le fait remarquer la Free Software Foundation, "le logiciel libre est une question de liberté, et non de prix. Pour comprendre ce concept, il faut entendre le mot 'libre' au sens de 'liberté de parole', et non pas à celui d'entrée libre"^{§§}. Les logiciels que nous considérons généralement comme étant 'libres' sont, il est vrai, plus ou moins gratuits à l'acquisition – c'est-à-dire qu'ils peuvent être acquis, copiés et utilisés gratuitement. Cependant, lorsque l'on étudie le coût d'un logiciel, il est important de tenir compte du *coût total* de son acquisition et de son utilisation – ce que l'on désigne par le terme de "coût total de possession".

Arrêtons-nous, par exemple, sur la question des coûts d'adaptation des logiciels "libres" aux besoins particuliers de l'utilisateur. Les caractéristiques des logiciels commerciaux sont généralement le fruit d'études de marché et d'investissements dans des activités de développement visant à répondre aux besoins d'une communauté d'utilisateurs ou du marché. Même s'ils ne sont pas toujours en adéquation parfaite avec ces besoins, les logiciels commerciaux sont, en règle générale, conçus de manière à y répondre dans une

^{§§} Voir : <http://www.gnu.org/philosophy/free-sw.html>.

large mesure. Les logiciels libres, en revanche, sont souvent conçus sans grand souci de répondre aux besoins des utilisateurs finaux et sont diffusés sous une forme qui incite à les modifier. Pour répondre aux besoins d'un utilisateur donné, il peut être nécessaire de les modifier, à un coût parfois considérable. En outre, les concepteurs de logiciels commerciaux fournissent généralement un service d'entretien et de support technique pour leurs produits, ce qui, bien souvent, n'est pas le cas des concepteurs de logiciels libres.

Il est important, lorsque l'on étudie le coût des logiciels "libres", de tenir compte des coûts de modification et d'entretien de ces logiciels, du coût du support technique, du coût de la documentation de l'utilisateur et de la documentation technique indispensables (toute documentation dont les logiciels libres sont généralement dépourvus), du coût des contrôles de qualité (les logiciels libres ne sont généralement pas couverts par une garantie et sont le plus souvent fabriqués et diffusés sans être soumis à des essais aussi rigoureux que ceux auxquels procèdent les concepteurs de produits commerciaux), ainsi que des coûts d'adaptation, d'implémentation, de correction des défauts, de développement continu et, enfin, des coûts liés aux questions de sécurité. Des coûts de formation doivent également être pris en compte, et ce à deux niveaux – formation du personnel technique, formation des utilisateurs finaux –, car la main d'œuvre disponible aura probablement des compétences acquises à travers l'utilisation de produits commerciaux et pourra manquer des connaissances requises pour prendre en charge et utiliser des substituts libres, rendant une formation spécialisée complémentaire nécessaire.

L'analyse coûts-avantages doit, en définitive, être réalisée en se posant la question fondamentale suivante : une solution informatique fondée sur un logiciel protégé oblige-t-elle l'utilisateur à déboursier une somme exagérée pour des prestations dont il n'a pas besoin, ou offre-t-elle au contraire des caractéristiques et des avantages qui permettront à l'utilisateur d'être, à bien des égards, plus productif qu'il ne le serait avec des solutions concurrentes libres, sauf à consentir des investissements supplémentaires dont le coût dépasserait celui d'une solution fondée sur un logiciel commercial?

La licence publique générale et la licence publique générale limitée

Le type de logiciel libre qui a suscité le plus d'intérêt – et qui a causé le plus de nuits blanches aux concepteurs de logiciels commerciaux, aux investisseurs et aux personnes jouant un rôle dans les fusions-acquisitions d'entreprises de logiciels – est celui diffusé sous la licence publique générale (GPL) et sous la licence publique générale limitée (LGPL), qui émanent toutes deux de la Free Software Foundation. L'origine de la Free Software Foundation et de son initiative pour le logiciel gratuit (projet GNU) sont retracés sur son site Internet. Ce texte est reproduit ci-dessous :

Les participants au projet GNU ont conçu un système complet de logiciels libres portant le nom de "GNU" ("GNU's not UNIX", littéralement, GNU n'est pas UNIX) et ayant une compatibilité ascendante avec UNIX. Le

document initial de Richard Stallman sur le projet GNU, intitulé Manifeste GNU, a été traduit en plusieurs langues. Le nom GNU a été retenu parce qu'il répondait à plusieurs conditions : d'une part, il constituait un acronyme récursif; d'autre part, il s'agissait d'un vrai mot; enfin, il était amusant à prononcer (ou à Chanter). L'annonce initiale du lancement du projet GNU, rédigée en 1983, est également disponible sur ce site.

Le mot "libre", tel qu'il figure au paragraphe précédent, fait référence à la notion de liberté, et non à celle de prix. Pour acquérir un logiciel GNU, parfois il sera nécessaire de le payer, parfois il ne le sera pas. Dans les deux cas, une fois acquis le programme, l'acquéreur jouit de trois libertés bien précises. La première est celle de copier le programme et de le donner à ses amis ou à ses collègues; la deuxième est celle de modifier le programme comme il l'entend, et ce, grâce à l'accès sans restriction au code source; la troisième est celle de diffuser une version améliorée de ce logiciel et d'ainsi apporter une contribution à la communauté (les logiciels GNU peuvent être diffusés à nouveau, gratuitement ou moyennant rétribution de l'acte physique d'en transmettre une copie).

Le projet GNU a été conçu en 1983 dans le but de faire revivre l'état d'esprit qui prévalait auparavant au sein de la communauté de l'informatique et de rendre la coopération à nouveau possible en supprimant les entraves à cette dernière imposées par les propriétaires de logiciels protégés.

En 1971, lorsque Richard Stallman entamait sa carrière au sein du Massachusetts Institute of Technology (MIT), il travaillait dans un groupe qui avait exclusivement recours à des logiciels libres. Les entreprises d'informatique elles-mêmes diffusaient régulièrement des logiciels libres. Les programmeurs étaient libres de coopérer entre eux, et ne s'en privaient pas.

Dès les années 1980, presque tous les logiciels avaient été protégés, ce qui signifiait que leurs propriétaires interdisaient et empêchaient la coopération entre les utilisateurs, rendant ainsi nécessaire l'élaboration du projet GNU^{***} :

Dans un exposé intitulé "The GNU Operating System and the Free Software Movement (le système d'exploitation GNU et le Free Software Movement), publié à l'origine dans le livre "Open Sources" (codes source ouverts), Richard Stallman, fondateur du Free Software Movement, écrivait :

Peut de temps avant de mettre en route le projet GNU, j'ai entendu parler du compilateur prêt-à-monter de la Free University

^{***} <http://www.gnu.org/gnu/gnu-history.html>.

(“Free University Compiler Kit”), également connu sous le nom de VUCK (le mot signifiant “libre” en néerlandais s’écrivant avec un “V”). Ce compilateur pouvait prendre en charge divers langages de programmation, notamment le langage C et le langage Pascal, ainsi que diverses machines cibles. J’ai écrit à l’auteur de ce compilateur afin de lui demander si ce dernier pouvait être utilisé dans le cadre du projet GNU.

Il m’a répondu de manière sarcastique, me disant que si l’université était gratuite (“Free”), le compilateur, lui, ne l’était pas.

Le but du projet GNU était de rendre les utilisateurs libres, pas de se contenter d’être populaire. Nous avons besoin de conditions de distribution qui empêcheraient de transformer du logiciel GNU en logiciel protégé. Nous avons utilisé la méthode du “copyleft” (“gauche d’auteur”, ou “à laisser copier”).

Le “copyleft” utilise les lois du droit d’auteur, en les retournant pour leur faire servir le but opposé de ce pour quoi elles ont été conçues : ce n’est pas une manière de privatiser du logiciel, mais une manière de le laisser “libre”.

L’idée centrale du “copyleft” est de donner à quiconque la permission d’exécuter le programme, de le copier, de le modifier, et d’en distribuer des versions modifiées – mais pas la permission d’ajouter des restrictions de son cru. C’est ainsi que les libertés cruciales qui définissent le “logiciel libre” sont garanties pour quiconque en possède une copie; elles deviennent des droits inaliénables.

Pour que le “copyleft” soit efficace, il faut que les versions modifiées demeurent libres, afin de s’assurer que toute œuvre dérivée de notre travail reste disponible à la communauté en cas de publication. Quand un programmeur professionnel se porte volontaire pour améliorer le logiciel GNU, c’est le “copyleft” qui empêche son employeur de dire : “Vous ne pouvez pas partager ces modifications, car nous allons les utiliser dans le cadre de notre version propriétaire du programme.”

Il est essentiel d’imposer que les modifications restent libres si on souhaite garantir la liberté de tout utilisateur du programme. Les sociétés qui ont privatisé le système de fenêtre X ont en général fait quelques modifications pour le porter sur leurs systèmes et sur leur matériel. Il s’agissait de modifications ténues comparées à X dans son ensemble, mais elles n’en étaient pas pour autant négligeables. Si le fait de procéder à des modifications pouvait servir de prétexte à ôter leur liberté aux utilisateurs, il serait facile pour quiconque de s’en servir à son avantage.

Le problème de la réunion d'un programme libre avec du code non libre est similaire. Une telle combinaison sera indubitablement non libre; les libertés absentes de la partie non libre du programme ne se trouveraient pas non plus dans l'ensemble résultant de cette compilation. Autoriser de telles pratiques ouvrirait une voie d'eau suffisante pour couler le navire. C'est pourquoi il est crucial pour le "copyleft" d'exiger qu'un programme sous "copyleft" ne puisse pas être inclus dans une version plus grande sans que cette dernière ne soit également couverte par le "copyleft".

La mise en œuvre spécifique du "copyleft" que nous avons utilisée pour la plupart des logiciels GNU a été la licence publique générale de GNU, en abrégé GNU GPL. Nous disposons d'autres types de "copyleft" pour des circonstances particulières. Les manuels du projet GNU sont eux aussi couverts par le "copyleft", mais en utilisent une version très simplifiée, car il n'est pas nécessaire de faire appel à toute la complexité de la GNU GPL dans le cadre de manuels^{†††}.

Ce qui donne à beaucoup des insomnies, c'est l'idée qu'un logiciel censément protégé puisse devenir soumis aux conditions de la licence GNU si l'on considère qu'il est "fondé sur", "ajouté à", "combiné avec", "dérivé de" ou une "version modifiée d'un logiciel distribué sous licence GNU. Le problème est que sauf dans les cas extrêmes, il est difficile de savoir avec certitude ce que recouvrent exactement les termes employés dans les licences GNU, et il est très facile pour un programmeur indiscipliné de s'aventurer dans une zone grise où le logiciel protégé de son employeur pourrait être soumis au régime du "copyleft". La conséquence la plus redoutée serait que ce logiciel censément protégé pourrait ainsi être soumis à des conditions ou obligations imposant sa libre diffusion, avec large permission de modification et de rediffusion et obligation de mettre le code source à la disposition de tous les titulaires de licence.

Bien des logiciels diffusés sous GPL sont utiles pour apporter des fonctionnalités dont on a besoin dans de plus grands systèmes, et les programmeurs qui travaillent sur de la matière logicielle protégée sont parfois tentés de simplement les y incorporer, plutôt que de créer du tout au tout la fonctionnalité équivalente. En outre, de nombreux outils utiles et appréciés de programmation et de développement sont diffusés sous GPL ou LGPL, et certains de ces outils injectent des parties d'eux-mêmes dans le logiciel qu'ils servent à produire. De l'avis de certains, d'ailleurs, le simple fait d'écrire un logiciel qui puisse fonctionner avec telle ou telle composante sous GPL suffit à rendre les conditions de la GPL applicables au nouveau logiciel^{†††}. Or, si l'on voit clairement dans certains cas que l'utilisation d'un code source sous GPL pour élaborer un nouveau produit aura pour conséquence que le produit résultant sera soumis aux conditions de la GPL, dans de nombreux autres cas les choses ne sont pas aussi tranchées dans un sens ou dans l'autre. Le fait qu'un logiciel protégé peut tomber sous les conditions de la licence libre si l'on y

^{†††} <http://www.gnu.org/gnu/thegnuproject.html> (note omise).

^{†††} Voir <http://kerneltrap.org/node/view/1735>.

inclut par inadvertance un petit élément de code source libre a pu inciter à qualifier le code sous GPL/LGPL de “viral”.

De nombreuses questions que l’on se pose au sujet des incidences subtiles des licences GNU et de leur impact sur le monde réel de la mise au point et de la diffusion de logiciels resteront sans réponse sauf construction judiciaire, et un examen complet de ces questions demanderait bien plus que le présent document. Toutefois, pour alimenter la réflexion, voici une liste de questions qui donnent à certains des insomnies :

1. Les licences GPL et LGPL sont-elles des contrats qui imposent des obligations positives au preneur de licence, ce qui l’expose à se voir ordonner l’exécution en nature en cas de manquement, ou sont-elles simplement des autorisations conditionnelles, de sorte que le fait, par exemple, de ne pas mettre le code source à la disposition des preneurs de licence en aval serait une inobservation de condition exposant le preneur de licence à des poursuites pour atteinte au droit d’auteur?
2. Considère-t-on d’une manière générale que les employés de sociétés propriétaires de logiciels protégés sont autorisés à soumettre leur employeur aux conditions des licences GPL et LGPL du simple fait qu’ils téléchargent et utilisent les outils logiciels libres?
3. Même si la licence GPL ou LGPL est considérée comme acceptée et opposable dans une circonstance donnée, la licence comporte-t-elle des conditions particulières qui pourraient n’être pas applicables parce que léonines, interdites par la législation fédérale ou contraires à l’ordre public, ou pour quelque autre raison?
4. La notion d’un produit “dérivé” telle qu’employée dans les licences GPL et LGPL est-elle à prendre au sens technique juridique que lui donne la législation traditionnelle du droit d’auteur, ou faut-il y voir quelque chose de différent dans le contexte d’une licence “copyleft”?
5. Que signifie “combiner” un logiciel protégé avec un élément logiciel sous GPL ou LGPL? S’agit-il seulement, littéralement, d’une greffe ou d’un “lien statique” établi lors de la création du programme, ou peut-on y inclure aussi le “lien dynamique” qui s’établit lors de l’exploitation du programme sur l’ordinateur de l’utilisateur final? Et peut-on aussi y inclure des programmes qui opèrent en interconnexion avec des logiciels sous GPL ou LGPL s’ils sont couplés de manière suffisamment intime?
6. Est-il possible de “guérir” un produit protégé dans lequel on aurait par inadvertance ou sans autorisation utilisé un code sous GPL ou LGPL en supprimant celui-ci ou en le remplaçant par un nouveau composant protégé, même après que la version initiale “infectée” a été distribuée?

7. L'acte consistant à fournir des copies d'un logiciel à des sociétés affiliées constitue-t-il "diffusion" au sens des licences GPL et LGPL?
8. L'acte consistant à mettre un logiciel passivement à disposition pour téléchargement sur l'Internet constitue-t-il "diffusion" au sens des licences GPL et LGPL?
9. L'acte consistant à mettre une fonctionnalité logicielle à disposition sur l'Internet (*p. ex.* dans le cas d'un fournisseur de services logiciels) constitue-t-il "diffusion" de ce logiciel au sens des licences GPL et LGPL?
10. Quelles sont les incidences des licences GPL et LGPL sur le plan des brevets?
11. En cas de différend quant au sens d'une condition particulière de la licence GPL ou LGPL, quelle intention l'emporte : l'intention des parties à la transaction particulière pour laquelle la licence a été employée, ou l'intention de la personne ou de l'organisation (la Free Software Foundation) qui a créé le document de licence?^{§§§}
12. Quelles politiques et pratiques les concepteurs de logiciels protégés peuvent-ils adopter pour faire en sorte que leurs produits ne soient pas affectés par la GPL ou la LGPL?
13. En quoi consiste la diligence raisonnable attendue de qui prévoit d'acquérir ou d'investir dans une entreprise dont la valeur dépend de la propriété intellectuelle attachée à ses logiciels?
14. Quelles sont les obligations en matière d'audit et de divulgation d'une entreprise publique dont la valeur dépend de la propriété intellectuelle attachée à ses logiciels?

Conclusion

Les membres de l'industrie du logiciel indépendant nous ont ouvert une époque de bouleversements telle que le monde n'en avait pas connu depuis la révolution industrielle. Les milliards de dollars qu'ils ont investis dans le développement de leur technologie n'auraient jamais rien donné sans la protection des fruits de leur travail intellectuel. Le logiciel libre a cependant trouvé sa place dans l'écosystème informatique et il donne aux concepteurs et aux utilisateurs de logiciels la possibilité importante d'un style différent de développement et de distribution. Les deux sont nécessaires et ils servent chacun leur but.

^{§§§} On trouvera les réponses de la Free Software Foundation's aux questions concernant les licences GPL et LGPL à l'adresse <http://www.fsf.org/licenses/gpl-faq.html>.

Pour prendre la décision soit d'adopter des solutions offertes par les logiciels libres, soit de payer des logiciels commerciaux, il est important de considérer comment la solution du logiciel libre se place par rapport à celle du logiciel commercial non seulement en termes de caractéristiques techniques, mais aussi en termes d'investissement global et de coût total de possession. Il importe également de considérer si l'utilisation d'une solution libre répondra aux besoins de l'utilisateur en ce qui concerne la compatibilité opérationnelle avec d'autres programmes et la nécessité de pouvoir faire des transactions et communiquer avec des tiers dans des industries et sur des marchés où des produits commerciaux constituent le standard de fait.

Enfin, il est important d'avoir conscience que le mouvement du logiciel libre a pour motivations non seulement de bénéficier aux concepteurs et aux utilisateurs de logiciels par la diffusion de logiciels "libres", mais aussi de supprimer la nécessité de recourir à des logiciels protégés. Par des actions de sensibilisation et d'autres efforts, le mouvement du logiciel libre a persuadé des sociétés et même des pouvoirs publics de privilégier les produits libres, et les licences GPL et LGPL du projet GNU ont soulevé nombre de questions sérieuses sur l'intégrité des droits de propriété intellectuelle attachés aux logiciels développés dans l'entreprise privée. Nous espérons que ce document aura fourni des éléments d'information utiles et matière à réflexion.