



# Les logiciels libres

AFF065-2008-NT455

\$Rev: 407 \$

2011-04-03 13:22:17+02:00

Sebastien Devaux

dexter@detexia.com

**Résumé :** Les logiciels libres ont été créés en réaction à des restrictions jugées excessives imposées par certains éditeurs qui souhaitent protéger leur propriété intellectuelle. Ils ont été par conséquent définis à partir des droits des utilisateurs qui doivent être préservés autant, sinon plus, que ceux des auteurs. Certaines licences libres incluent même des dispositions visant à garantir ces droits même en cas de modification ou d'intégration d'un tel logiciel. Ceci n'empêche aucunement son utilisation dans un cadre professionnel ou industriel, cela permet même de s'affranchir d'une certaine emprise des éditeurs de logiciels. Par contre lors de l'intégration d'un logiciel libre dans un produit vendu à son client, certaines précautions sont à prendre pour que la diffusion puisse avoir lieu dans le respect mutuel des droits de tous les protagonistes : auteurs du logiciel libre, auteurs du produit vendu et client acheteur du nouvel ensemble.

**Mots-clé :** logiciel libre, licence, génie logiciel, développement industriel

## Table des matières

~ \* ~ \* ~

<b>1</b>	<b>Présentation.....</b>	<b>3</b>
1.1	Historique.....	3
1.2	Définition.....	4
1.3	Les licences libres courantes.....	5
1.4	Précisions et idées reçues.....	5
<b>2</b>	<b>Usage professionnel et industriel.....</b>	<b>6</b>
2.1	Cas d'utilisation.....	6
2.2	Utilisation en tant qu'outillage.....	6
2.3	Intégration dans un logiciel.....	6
2.4	Environnements critiques et certifications.....	7
2.5	Défense et secteur publique.....	8
<b>3</b>	<b>Annexes.....</b>	<b>10</b>
3.1	Quelques logiciels libres.....	10
3.1.1	Systèmes d'exploitation.....	10
3.1.2	Bureautique et multimédia.....	10
3.1.3	Développement et ingénierie.....	10
3.1.4	Serveurs.....	11
3.2	Les licences libres.....	11
3.3	L'édition des liens.....	12
	<b>Table des URL .....</b>	<b>14</b>

# 1 Présentation

~ \* ~ \* ~

## 1.1 Historique

Les logiciels sont nés libres. A l'origine de l'informatique ils étaient gracieusement fournis avec l'achat du matériel exactement comme on reçoit encore de nos jours un livret de recettes lorsqu'on achète des ustensiles de cuisine. Ces quelques programmes rudimentaires n'étaient alors que des exemples d'utilisation du matériel. Son possesseur avait toute liberté de les utiliser, de les étudier et de les modifier à loisir.

L'industrie informatique se perfectionnant, les logiciels sont devenus un élément indispensable et les quelques recettes d'utilisation sont devenues des systèmes d'exploitation complets et des applications sophistiqués dont l'effort de développement et de mise au point allait grandissant jusqu'à devenir non négligeable. L'industrie du logiciel commença alors à émerger de l'industrie informatique encore jeune avec l'apparition de sociétés spécialisées dont l'unique activité sera désormais la rédaction de programme. Ces sociétés ont naturellement eu le souci de protéger leur savoir faire et leurs investissements. Elles ont pour cela cessé de fournir les sources de leur programmes et progressivement limité les droits de duplication, de modification et d'utilisation pour ne plus réellement vendre des logiciels mais, au lieu de cela, des licences. Une licence est un document contractuel, plus ou moins tacitement accepté par le futur utilisateur lors de son acte d'achat, qui décrit les droits d'utilisation du logiciel qui lui sont concédés.

Si cette situation n'a pas été gênante pour beaucoup d'utilisateurs, certains d'entre eux ont commencé à être plus particulièrement contraints par ces dispositions. En particulier un enseignant du fameux MIT, Richard Stallman, a considéré qu'il n'était même plus possible pour lui de mener à bien sa tâche d'enseignement puisque les matériels vieillissant du MIT étaient progressivement remplacés par des ordinateurs plus récents pour lesquels aucun logiciel n'était disponible avec ses sources. Il n'était donc plus possible d'examiner et de modifier les programmes reçus pour les étudier et enseigner leur fonctionnement. Devant cette impasse il a décidé de quitter son emploi pour créer des logiciels libres ainsi qu'une fondation pour leur promotion : la Free Software Foundation <sup>[url1]</sup> est créée en 1985 et elle possède maintenant une antenne française. <sup>[url2]</sup>

Cette fondation a lancé le développement d'un système d'exploitation complet baptisé GNU (acronyme récursif qui se développe en GNU's Not Unix). Ce système s'inspire d'Unix et a l'ambition de constituer à terme un système POSIX complet accompagné de nombreuses applications.

Le mouvement du logiciel libre souffre un peu de la confusion engendrée par le mot anglais free qui a deux significations : libre et gratuit. Le plus important pour ses créateurs est la liberté, la gratuité souvent constaté n'étant finalement qu'un effet secondaire qui s'il présente quelques inconvénients aura au moins eu l'immense avantage d'avoir considérablement aidé à la diffusion massive des logiciels libres.

En 1991, de nombreux programmes avaient déjà été écrits pour le système GNU mais il lui manquait encore une brique essentielle : le noyau. Sa mise au point était beaucoup plus longue et on ne disposait toujours pas à cette date de système GNU même si tous les outils déjà développés pour lui étaient déjà assez populaire dans la communauté des informaticiens. A cette époque, un jeune finlandais, Linus Torvalds, a entrepris de créer un noyau de système multi-utilisateur et multi-tâche à partir du système minix qui est un système minimal créé pour l'enseignement. Avec le développement d'internet il a rapidement été rejoint par de nombreux passionnés qui ont ensemble développé un noyau un peu moins ambitieux techniquement que le noyau du système GNU mais dont la mise au point d'une première version utilisable a été beaucoup plus rapide. Ce nouveau noyau baptisé Linux en hommage à son créateur et en référence à de nombreux programmes se terminant en X du monde Unix a été intégré dans le système GNU. Ce nouveau système GNU/Linux s'est répandu par l'intermédiaire d'internet car il a rendu accessible pour un coût quasi nul un système de type Unix sur les ordinateurs personnel PC les plus communs alors que ce genre de système était jusque là réservé à des stations de travail très coûteuses.

Par commodité on utilise souvent le terme Linux seul pour désigner un système d'exploitation complet alors que Linux n'est en fait qu'un noyau de système d'exploitation. On devrait désigner un tel système sous le nom de GNU/Linux, c'est à dire qu'il s'agit du système GNU construit avec le noyau Linux. Il existe des système GNU avec d'autres noyaux. On citera en particulier le système GNU/HURD qui contient le noyau HURD qui est le noyau originellement prévu pour le système GNU mais dont le développement n'est toujours pas achevé.

## 1.2 Définition

La définition précise d'un logiciel libre est pour certains détails parfois encore sujet à polémiques. Il s'est toutefois dégagé un certain consensus relatifs à quelques critères fondamentaux. Un logiciel pourra être déclaré libre s'il accorde au moins les trois libertés fondamentales suivantes à ses utilisateurs :

- Liberté d'exécuter (utiliser) le logiciel à n'importe quelle fin.
- Liberté d'étudier et de modifier le logiciel à sa convenance.
- Liberté de copier et redistribuer le logiciel et ses éventuelles modifications pour son propre usage ou à des tiers.

Pourquoi ces trois libertés ? Que reproche-t-on finalement à certains éditeurs de logiciels dont les clauses d'utilisations sont jugées abusives ?

La première liberté peut sembler aller de soi mais elle ne l'est pas forcément, certains procédés et logiciels sont protégés par des brevets et le simple fait d'utiliser un logiciel dans un cadre autre pour réaliser des tâches pour lequel il n'a pas été prévu peut constituer en soi une violation de brevet. L'éditeur peut craindre également voir émerger des services basés sur ses propres logiciels qui entrent en concurrence avec son activité. En général vous n'avez pas le droit de vendre des prestations de votre utilisation de votre logiciel à des tiers qui pourraient ainsi être dispensé de son achat. Certains éditeurs ont aussi introduit des limites sur le nombres d'utilisateurs simultanés d'un logiciel même s'il n'est pas déployé sur plus d'un ordinateur.

Il devenu impossible d'exercer la deuxième liberté avec les logiciels propriétaires. Pour protéger leur propriété intellectuelle, les éditeurs ont rapidement cessé de fournir les sources des programmes mais uniquement les binaires exécutables prêts à l'emploi. Si cela n'empêche pas l'utilisation courante telle qu'elle a été prévue par les concepteurs, cela interdit en revanche toute adaptation du logiciel à des particularités personnelles, il faut donc au contraire obligatoirement adapter sa méthode de travail à la logique des concepteurs du logiciel ce qui peut s'avérer délicat. De même en cas d'anomalie gênante constatée, on n'a aucun recours autre que d'attendre la bonne volonté de l'éditeur pour l'obtention d'un correctif. En pratique ce dernier se dégage de toute obligation via les clauses de garantie de ses contrats de licences (il n'y a en général strictement aucune garantie sur ce que fait le logiciel) et incitera fortement à l'achat d'une nouvelle édition pour régler ou contourner le problème.

La dernière liberté de duplication est essentielle, pour permettre la copie de sauvegarde d'un système et surtout sa restauration, mais aussi l'évolution d'un système informatique. On peut ainsi décider de répartir différemment ses outils informatiques pour passer par exemple d'un système centralisé où le logiciel est installé sur un seul appareil, vers un système décentralisé où le logiciel est réparti sur plusieurs sites, ce qui selon la définition juridique de chaque site peut être considéré comme une redistribution à un tiers.

R.Stallman a non seulement popularisé l'idée de logiciel libre mais aussi inventé des outils pour le défendre dont le fameux copyleft. Il s'agit du détournement du principe de copyright dont les éditeurs de logiciel ont abusé pour défendre la liberté des utilisateurs en garantissant qu'un logiciel libre le restera. Il a créé une licence à inclure dans sa déclaration de copyright qui autorise et encourage la copie du logiciel. Elle exige également que toutes ses modifications soient soumises à la même règle.

## 1.3 Les licences libres courantes

Les droits de l'utilisateur sont en général décrits en détail dans la licence qui accompagne le logiciel. Ce document détaille également les restrictions imposées car même pour les logiciels libres, la liberté de l'utilisateur a des limites. Ces limitations sont plus ou moins importantes selon les familles de licences :

- Licence GPL (GNU Public Licence ou General Public Licence) : Licence créée par R. Stallman afin de promouvoir et défendre son combat. Elle accorde évidemment les libertés fondamentale décrites dans la définition ci-avant mais y ajoute une importante restriction : toute modification d'un logiciel GPL doit rester GPL. Ceci, en pratique, interdit l'intégration d'un logiciel GPL dans un logiciel non GPL (l'intégration étant définie par la notion d'édition des liens, cf. annexes).
- Licence LGPL (Lesser GPL) : lors de la création de bibliothèques, les obligations de la licence GPL ont été jugées trop fortes et présentaient un risque important de freiner leur diffusion et leur acceptation : l'interdiction d'effectuer une édition des liens avec un logiciel non-GPL interdit totalement l'utilisation d'une telle bibliothèque par un logiciel non GPL. En réponse cette nouvelle licence autorise ce genre d'utilisation en levant les restrictions portant sur l'édition des liens.
- Licence BSD, Licence Apache : ce sont les licences les plus permissives, la seule contrainte importante est de citer la source et le copyright originel lors de la redistribution ou de l'intégration du logiciel.

En cas d'incertitude sur la compatibilité des droits d'un logiciel libre avec votre usage, le plus simple est de contacter les auteurs. La plupart du temps, ils sont assez facilement joignables et relativement ouverts à des situations non couvertes par la licence qu'ils ont choisi et pourrons vous proposer à ce moment une licence spécifique (peut être en échange d'une contrepartie).

## 1.4 Précisions et idées reçues.

On assimile souvent les logiciel libres aux logiciels "open-source" (à sources ouverts). En général les deux termes sont effectivement interchangeables et le terme "open-source" n'a aucune ambiguïté comparé à "free" qui littéralement peut signifier selon le contexte libre ou gratuit. En réalité, pour un logiciel libre, la publication des sources n'est finalement pas une fin en soi, mais la conséquence de l'obligation de permettre la modification du logiciel.

On prétend parfois qu'il est interdit de vendre un logiciel libre. En réalité cela ne l'est pas du tout. R.Stallman lui même a financé son combat pendant un certain temps par la vente de ses premiers logiciels dont le fameux Emacs. Il est par contre difficile de dégager des revenus importants sur le modèle de la vente de licence avec des logiciels libres car en vertu du droit de l'utilisateur de dupliquer lui-même le logiciel, un modèle de rétribution basé sur le volume des copies sera forcément assez limité. L'utilisateur est même en droit de vendre lui-même de nouvelles copies pour son propre compte.

Techniquement, il n'y a pas de raison qu'un logiciel libre soit meilleur ou non qu'un logiciel propriétaire. Par contre l'accès aux sources permet à quiconque le souhaite d'étudier le fonctionnement du logiciel et d'y réparer des défaillances. Dès qu'un logiciel libre rencontre un certain succès, il devient "naturellement" meilleur au fur et à mesure que ses utilisateurs éclairés corrigent les anomalies qui les gênent.

Logiciels libres et entreprise commerciale ne sont pas forcément incompatibles. Tout le monde utilise souvent quotidiennement Linux sans le savoir au travers des services d'une entreprise qui a déjà acquis est une position dominante sur le marché des moteurs de recherche, à savoir Google. Cette société malgré tous les services qu'elle offre gratuitement à de nombreux utilisateurs est loin d'être une organisation à but non lucratif et a de grande ambitions. Ses fondateurs sont bien conscient du pouvoir qu'ils sont en train de constituer et des craintes que cela peut inspirer, sinon pourquoi auraient-ils retenu comme devise et consigne à leurs employés, la phrase : "don't be evil <sup>[url3]</sup> " ? (voir aussi l'article Google en français <sup>[url4]</sup> sur wikipedia qui aborde ce sujet.

## 2 Usage professionnel et industriel

~ \* ~ \* ~

### 2.1 Cas d'utilisation

Selon les contexte d'utilisation, les logiciels libres et surtout les obligations décrites dans leurs licences peuvent avoir différentes incidences. De même dans certains domaines d'application les caractéristiques d'un logiciel libres pourront être un atout majeur ou à l'inverse insuffisants car heureusement malgré la perception assez négatives des éditeurs de logiciels qui peut transparaître de cet article, la majorité fait heureusement un travail très profitable et utile.

### 2.2 Utilisation en tant qu'outillage

L'utilisation de logiciel libre en tant qu'outillage interne en remplacement de leurs équivalent propriétaire ne pose aucun problème. Si la licence du logiciel est réellement libre, elle ne peut imposer aucune règle d'usage limitative qui imposerait quelque restriction que ce soit sur ce qui est produit à partir de ces outils. Même les licences les plus strictes comme le GPL n'imposent rien sur ce qui est produit avec le logiciel libre. Ceci est particulièrement important pour des outils comme les générateurs de code qui n'imposent pas que le code qu'ils génèrent soit libre.

Pour les outils de tous les jours (bureautique), on trouvera plus ou moins facilement des outils libres équivalents. La principale difficulté sera de trouver des produits capables de manipuler les formats de fichier issus des outils propriétaires que l'on utilise déjà ou que nos interlocuteurs (clients ou fournisseurs) utilisent pour communiquer avec eux.

Si le premier intérêt du logiciel libre est souvent l'économie sur l'achat de licences, la dernière remarque du paragraphe précédent révèle sa principal qualité : avec un logiciel libre on devient indépendant du fournisseur du logiciel. On n'utilise plus une boîte noire qui enregistre les données que l'on produit dans un format opaque et rend complètement captif vis à vis du fournisseur. L'accès au code source garantit la possibilité (technique et légale uniquement) de concevoir et utiliser des convertisseurs vers d'autres formats si besoin. Ainsi, quoi qu'il arrive, il est toujours possible d'accéder à ses propres données.

Aujourd'hui, plus rien n'empêche de se libérer de l'emprise d'un éditeur envahissant si on accepte de changer un tout petit peu ses habitudes (voir en annexe pour une liste de logiciels libres courants pour la bureautique). Cela permettra également d'améliorer sa situation légale en facilitant l'élimination de copies de logiciels propriétaires non déclarées ou de shareware indéfiniment utilisés en version théoriquement limitée dans le temps.

On relèvera sur le même thème le danger de l'utilisation d'outils informatiques à la location et du dernier concept à la mode du cloud computing : que deviennent les données produites et/ou hébergées par de tels outils lorsque les droits d'utilisation cessent ? Qui est le propriétaire des données ? Est-ce l'auteur c'est à dire l'individu qui a utilisé l'outil ou bien l'éditeur de l'outil ? Face à la quantité de ressources disponibles dans toute entreprise qui est loin d'être utilisée à son plein potentiel, cette nouvelle approche de la commercialisation d'outils informatiques semble avant tout une innovation destinée à rendre les clients encore plus captifs qu'avec un modèle déclinant de vente de licence par utilisateur.

### 2.3 Intégration dans un logiciel

L'intégration d'un logiciel libre dans un logiciel industriel même propriétaire est souvent possible, l'étendue exact des obligations imposées dépend du type de licences employées, selon les trois grandes familles de licences décrites précédemment :

- Le cas des licences type BSD/Apache est le plus simple : il suffit de joindre lors de la distribution du produit comprenant le logiciel libre intégré :
  - Sa dénomination originale.
  - La citation du copyright original.

- Le fichier de licence original.
- Il est d'usage de laisser l'utilisateur disposer des sources du logiciel libre (et uniquement celui-ci, la distribution des sources du produit qui l'inclut ne sont pas requis) même si cela n'est pas toujours explicitement demandé. En l'absence, l'utilisateur pourra toujours consulter les éléments précédents pour obtenir une copie des sources.

En pratique, le plus simple est de joindre l'archive originale téléchargée avant intégration qui en général contient tout ces éléments.

Enfin il est demandé de ne pas introduire d'ambiguïté entre le logiciel libre et le produit dans lequel il s'intègre et on choisira pour ce faire une dénomination clairement différente pour le produit.

- Pour les logiciels de type LGPL, il n'y a pas plus d'obligations que dans le cas présent, tant que le logiciel libre n'est pas modifié avant intégration.
- Le cas des logiciel de type GPL est un peu plus difficile selon le mode d'intégration. Si cette intégration se fait par édition de lien, tout le code lié doit alors être publié selon les termes de la licences GPL ce qui peut être en contradiction avec les règles de confidentialité industrielle qui s'applique au produit. Dans ce cas l'utilisation de code GPL n'est alors pas possible. Le seul moyen serait que le produit utilise le logiciel libre comme le ferait un utilisateur humain, c'est à dire le démarrer dans un processus séparé et dialoguer avec lui avec les seules interfaces de communication et de dialogue qu'il offre pendant son exécution.

Si l'intégration du logiciel libre nécessite au préalable qu'il soit modifié, le statut des modifications apportées dépend lui aussi du type de licence :

- Pour les licence de type BSD/Apache, il est possible de considérer que les modification font parti du produit et non du logiciel libre lui-même et il n'y a pas d'autre obligation qui s'ajoute à celle déjà décrite plus haut pour son intégration.
- Pour les licences de type GPL/LGPL, les modifications doivent être distribuées selon les termes de la licence, c'est à dire que les modifications sont elles-mêmes soumises à la licence originale du logiciel libre, en pratique les sources modifiés doivent être accessibles au moins jusqu'aux utilisateurs finaux du produit (et uniquement ceux-ci sans pour autant pouvoir leur interdire de le diffuser à leur tour).

Il est également d'usage de faire suivre les modifications vers les auteurs du logiciel original qui auront alors la liberté de les intégrer ou non dans leur distribution officielle. Selon l'ampleur de la modification et les modalités de gestion et de maintenance du logiciel libre modifié cela peut permettre de gagner une participation active sur ce logiciel en devenant un co-auteur reconnu et/ou un co-détenteur du copyright et ainsi influencer sur sa destiné.

La situation juridique d'un logiciel libre est loin d'être claire en France pour une raison simple : La très grande majorité des licences les plus utilisées sont rédigées exclusivement en anglais et n'ont pas été officiellement traduites, souvent volontairement par leurs promoteurs pour éviter des distorsions qu'une mauvaise traduction pourrait introduire. Cependant, en droit français, il est requis que tout texte visant à exprimer des droits contractuels (une licence est assimilable à un contrat passé entre l'auteur et l'utilisateur) soit rédigé en français. On peut donc s'interroger sur la réelle validité légale de ces licences sur le sol français surtout quand le seul lien qui lie l'auteur (souvent étranger et n'ayant jamais posé le pied sur le sol français) et l'utilisateur n'est qu'un acte de téléchargement d'un fichier. Cet état de fait a été l'une des principale motivation de la création de la famille de licences CeCILL <sup>[url5]</sup> dans laquelle on trouvera des déclinaisons (en non pas de simples traductions) françaises des trois types de licences présentés ici.

## 2.4 Environnements critiques et certifications

Comme il a été signalé précédemment, le niveau de qualité d'un logiciel libre est comparable à celui d'un logiciel propriétaire. D'un point de vu technique il n'y a aucune raison qu'il en soit autrement. Il y a bien

évidemment pour chaque logiciel considéré des écarts mais ces écarts seront parfois en faveur du logiciel libre, parfois en sa défaveur.

Il y a cependant un point particulier où le logiciel libre est en retard. Il s'agit de sa compatibilité avec un environnement critique tout particulièrement là où la vie humaine est mise en jeu comme dans le médical ou l'aéronautique. Encore une fois il ne s'agit pas d'un problème technique et la réactivité de la communauté des développeurs de logiciel libre a déjà été éprouvée mainte fois avec succès ce qui devrait jouer en sa faveur. Cependant pour ce type d'application, il ne suffit pas d'être capable de fournir des programmes avec le meilleurs taux de bugs par ligne de code. Il faut en plus respecter un protocole de qualification et certifier la conformité à des normes spécifiques des métiers visés. Les procédures de certification sont parfois très couteuses et elles ne sont que trop rarement à la portée des auteurs. En effet ces logiciels sont souvent développés par des communautés informelles de personnes individuelles volontaires. Hormis les aspects purement financiers, la qualification du logiciel est souvent bloquée par l'absence totale de structure administrative et juridique capable de la prendre en charge et surtout de l'assumer.

Le logiciel libre n'est pourtant pas dans l'impasse : n'importe quelle entité capable de certifier un logiciel peut le faire pour n'importe quel logiciel libre puisque tout y est accessible, il sera également possible de procéder aux modifications nécessaires aux cas où certaines exigences ne seraient pas remplies. Il devient même plus facile de constituer des groupements d'intérêt pour se charger de cette tâche car il n'y aura aucun ralentissement provoqué par une quelconque clause de confidentialité puisque tout est publique.

L'intérêt serait aussi de pouvoir faire bénéficier au plus grand nombre d'une bibliothèque de logiciels certifiés et de réduire plus facilement la portée des certifications futures par la possibilité de comparer effectivement, ligne par ligne les évolutions survenues entre une version certifiée et une nouvelle à certifier.

## **2.5 Défense et secteur publique**

La liberté offerte à un utilisateur reste valable quel que soit cet utilisateur. Peu importe qu'il s'agisse d'un individu, d'une organisation, d'une entreprise ou même d'un état. Pour une entreprise ou un état, cela peut s'avérer stratégique et vital. En effet un état ne sera libre que s'il est souverain et s'il a effectivement les moyens d'assurer sa souveraineté. La souveraineté absolue n'est aujourd'hui plus possible car plus personne, même un état, ne peut vivre en autarcie complète. On cherchera quand même à améliorer sa souveraineté en limitant au maximum sa dépendance des autres pour le fonctionnement de systèmes critiques.

De nos jours, le fonctionnement de toute organisation dépend plus ou moins d'un ou plusieurs systèmes d'information, donc du bon fonctionnement de plusieurs logiciels. C'est particulièrement vraie pour l'administration d'un état et l'organisation de sa défense. Que se passerait il si les systèmes informatiques du trésor public (comptes bancaire, collecte des impôts, distribution des fonds), ou les système de commandement et de communication des armées cessaient brutalement de fonctionner ? Le pays serait totalement paralysé et n'aurait plus aucun moyen de répondre à une agression.

Supposons qu'un seul logiciel particulier, par exemple (au hasard) un système d'exploitation, soit nécessaire au bon fonctionnement de tous ces systèmes ? Quelle serait alors l'emprise de fournisseur de ce logiciel sur l'état qui aurait eu l'imprudence de mettre ainsi tous ses œufs dans le même panier. En l'état actuel des choses, l'impact serait relativement faible. Le système fonctionne tant bien que mal avec ou sans l'assentiment du fournisseur. Au pire on reste sur un statu quo, le système ne peut plus évoluer et tant que les équipements ont des défaillances ponctuelles et localisées, il y a toujours moyen de s'en remettre. Par contre si on laisse le fournisseur migrer ces systèmes vers un modèle où on lui délègue certains traitements (cf. note sur le cloud computing ci-avant), où même simplement l'autorisation d'utiliser certains produits, autorisation qu'il prétend réaliser pour "notre sécurité", il acquiert de fait la capacité de paralyser totalement le système, donc le pays entier s'il décide unilatéralement de ne plus autoriser l'utilisation des services qu'il fournit. C'est un risque à prendre en considération pour les systèmes critiques d'un état à plus forte raison si ce fournisseur est étranger.

Le logiciel libre peut améliorer la souveraineté d'un état par deux moyens :

- Il est parfois la seule alternative à une offre commerciale insuffisamment diversifiée.
- Dans le cas où la diversité des systèmes est nuisible à leur fonctionnement (il est toujours plus difficile de faire fonctionner de concert des équipements et des logiciels complètement hétérogènes), le choix d'un logiciel libre permet de s'affranchir totalement de la dépendance d'un fournisseur particulier. La disponibilité des sources permettra toujours de déléguer si besoin la maintenance et l'évolution des systèmes à un tiers au cas où les relations avec le fournisseur d'origine viendraient à se dégrader.

Cette prise de conscience commence à être partagée par beaucoup d'organismes publiques qui sont alors très motivés par les apports du logiciel libre, autres que la simple économie sur l'achat de licences qui reste encore trop souvent la première motivation. C'est le cas en particulier du ministère de la défense qui a même explicitement établi une préférence pour le logiciel libre dans la directive n° 1/DEF/DGSIC <sup>[url6]</sup> qui contient entre autre un paragraphe intitulé "Privilégier les logiciels libres à coût global, risques et efficacité comparable".

## 3 Annexes

~ \* ~ \* ~

### 3.1 Quelques logiciels libres.

#### 3.1.1 Systèmes d'exploitation

- [FreeBSD](#) <sup>[url7]</sup>, [NetBSD](#) <sup>[url8]</sup>, [OpenBSD](#) <sup>[url9]</sup> : systèmes Unix descendant de la fameuse branche [BSD](#). <sup>[url10]</sup>
- [OpenSolaris](#) <sup>[url11]</sup> : déclinaison libre du système Unix de [Sun microsystems](#). <sup>[url12]</sup>
- [Darwin](#) <sup>[url13]</sup> : couche système fondamentale de type Unix qui est à la base de MacOS X.
- GNU/Linux : il n'y a pas un Linux mais des Linux, une multitude de Linux. On appelle ces variantes des distributions. Chacune est un regroupement cohérent de logiciels libres (parfois quelques composants non libres sont inclus), organisés et maintenus pour fournir des suites logicielles complètes incluant système d'exploitation, pilotes de périphériques et applications. Voici quelques distributions courantes :
  - [Debian](#) <sup>[url14]</sup> : distribution communautaire créée et maintenu par des volontaire répartis dans le monde entier.
  - [RedHat](#) <sup>[url15]</sup> : une des premières distribution soutenue par une entreprise commerciale créée spécifiquement à cette fin ayant rencontré un succès significative.
  - [Ubuntu](#) <sup>[url16]</sup> : basée sur debian, cette distribution se veut plus accessible et agréable pour l'utilisateur. Les dernière nouveauté y sont intégré plus rapidement afin de tenter de suivre au plus près la prise en charge des matériels les plus récents. Cette distribution a été créée et est toujours soutenue par [Canonical](#) <sup>[url17]</sup>, une entreprise Sud-Africaine.
  - [Mandriva](#) <sup>[url18]</sup> : distribution française assez populaire à travers le monde dont l'un des principaux objectifs est la facilité d'utilisation. Nommée à l'origine Mandrake, elle est issue d'une distribution RedHat et conserve encore aujourd'hui beaucoup de similitudes avec cette dernière. Elle est devenue Mandriva suite au rachat et la fusion de la distribution brésilienne Conectiva.
  - ... <sup>[url19]</sup>
- [ReactOS](#) <sup>[url20]</sup> : système dont l'ambition est réaliser une alternative libre compatible au niveau binaire (c'est à dire que les programmes compilés doivent fonctionner sans aucune modification) avec Windows NT et dans la mesure du possible, ses successeurs.

#### 3.1.2 Bureautique et multimédia

- [Firefox](#) <sup>[url21]</sup> : navigateur internet.
- [Thunderbird](#) <sup>[url22]</sup> : client de messagerie électronique.
- [OpenOffice](#) <sup>[url23]</sup> : suite bureautique comprenant traitement de texte, tableur, présentation, ... A noter qu'il s'agit de la première suite bureautique qui prend en charge un format de document normalisé par l'ISO, le format ODF.
- [The Gimp](#) <sup>[url24]</sup> : graphisme et retouche d'image.
- [Dia](#) <sup>[url25]</sup> : éditeur de diagrammes.
- [VLC media player](#) <sup>[url26]</sup> : Lecteur de médias audio/vidéo de tous formats.
- [7-zip](#) <sup>[url27]</sup> : Décompression, Compression et archivages de fichiers.

#### 3.1.3 Développement et ingénierie

- [GCC](#) <sup>[url28]</sup> : suite de compilateurs du projet GNU. Il prend en charge plusieurs langages dont C, C+, Fortran, Java.

- OpenJDK <sup>[url29]</sup> : déclinaison libre de la plateforme Java de SUN microsystems.
- Mono <sup>[url30]</sup> : mise en œuvre libre de la plateforme .Net de Microsoft. La compatibilité binaire avec la version 2.0 du framework Microsoft est aujourd'hui atteinte.
- Octave <sup>[url31]</sup> : langage de haut niveau pour le calcul scientifique, la simulation et la résolution de problèmes linéaires et non linéaires, ayant de plus une bonne compatibilité avec Matlab <sup>[url32]</sup>, l'un des leader du domaine.
- Scilab <sup>[url33]</sup> : suite logicielle pour le calcul scientifique et la simulation.
- KiCad <sup>[url34]</sup> : suite logicielle pour la conception et le développement de circuits électroniques.

### 3.1.4 Serveurs

- Apache <sup>[url35]</sup> : serveur HTTP.
- Sendmail <sup>[url36]</sup> : serveur SMTP, MTA (Mail Transfert Agent).
- Bind <sup>[url37]</sup> : serveur DNS.
- Samba <sup>[url38]</sup> : serveur SMB/CIFS, protocole de partage de fichiers des systèmes Windows.
- FileZilla Server <sup>[url39]</sup> : serveur FTP
- OpenSSH <sup>[url40]</sup> : shell distant sur canal de communication chiffré SSL.
- MySQL <sup>[url41]</sup> : système de gestion de base de donnée relationnelle.
- PostgreSQL <sup>[url42]</sup> : système de gestion de base de donnée relationnelle.

La plupart des logiciels libres cités dans les paragraphes précédents sont directement inclus dans les distributions Linux les plus courantes. Beaucoup d'entre eux sont également disponibles pour les systèmes Windows. Il est donc assez facile de tester et utiliser progressivement les logiciels libres tout en conservant son équipement informatique et ses logiciels actuels.

Aucun (ou presque) logiciel propriétaire n'a été utilisé pour la rédaction et la mise en forme de ce document :

- Edition du document source avec gvim <sup>[url43]</sup> (différentes versions selon les postes de travail)
- Mise en forme XHTML avec le processeur xslt de la bibliothèque XML du projet GNOME : libxml2 <sup>[url44]</sup>
- Mise en forme PDF avec le processeur XSL:FO Apache FOP <sup>[url45]</sup>
- Majoritairement éditer à l'aide d'un PC sous Linux mais aussi (d'où le "presque" en début de note) de temps en temps depuis un PC Windows XP.

## 3.2 Les licences libres

- BSD : <http://www.freebsd.org/copyright/license.html> , non-copyleft.
- Apache : <http://www.apache.org/licenses/LICENSE-2.0> , non-copyleft.
- MPL (Mozilla Public License) : <http://www.mozilla.org/MPL/MPL-1.1.html> , non-copyleft.
- X11/MIT : <http://www.xfree86.org/current/LICENSE5.html> , non-copyleft.
- GNU GPL : <http://www.fsf.org/licensing/licenses/gpl.html> , copyleft.
- GNU LGPL : <http://www.fsf.org/licensing/licenses/lgpl.html> , copyleft, édition des liens permise avec code non LGPL.
- GNU FDL : <http://www.fsf.org/licensing/licenses/fdl.html> , copyleft, variante de la GPL dédiée à la documentation. L'auteur peut verrouiller la modification de certaines portions du document lorsqu'il souhaite que ses opinions personnelles qui y sont exprimées ne soient pas altérées.
- CeCILL : <http://www.cecill.info>
  - CeCILL (Compatible GPL) : [http://www.cecill.info/licences/Licence\\_CeCILL\\_V2-fr.txt](http://www.cecill.info/licences/Licence_CeCILL_V2-fr.txt) , copyleft.

- CeCILL-B (équivalente au type BSD) : [http://www.cecill.info/licences/Licence\\_CeCILL-B\\_V1-fr.txt](http://www.cecill.info/licences/Licence_CeCILL-B_V1-fr.txt) , non-copyleft.
- CeCILL-C (équivalente au type LGPL) : [http://www.cecill.info/licences/Licence\\_CeCILL-C\\_V1-fr.txt](http://www.cecill.info/licences/Licence_CeCILL-C_V1-fr.txt) , copyleft, édition des liens permise avec code non CeCILL-C.

L'OSI <sup>[uri56]</sup> (Open Source Initiative), est un organisme public californien fondé pour promouvoir les logiciels à sources ouverts (ce que sont les logiciels libres). Sa principal contribution est d'avoir relu et analysé les licences afin de classer et confirmer ou infirmer leurs qualités et caractéristiques. L'inventaire publié sur son site complète de façon exhaustive les quelques éléments présentés dans le présent annexe.

### 3.3 L'édition des liens

Pour comprendre ce qu'est l'édition des liens, voici en quelques mots comment s'organise la fabrication d'un logiciel. Concrètement, la plupart du temps, un logiciel se présente sous la forme d'un fichier dit exécutable (les fameux fichiers suffixés .exe dans le cas de Windows). De leur côté, les informaticiens rédigent leurs programmes à l'aide d'un langage textuel à peu près humainement lisible. Un tel programme est enregistré dans un fichier texte (on parle aussi de fichier ASCII). La fabrication consiste à transformer ce fichier texte en fichier exécutable. Pour cela on utilise un compilateur. Il s'agit d'un programme spécialisé qui lit le fichier texte du programmeur, pour y traduire les directives du langage de programmation en instructions compréhensibles par l'ordinateur ciblé (chaque modèle de processeur possède son propre ensemble d'instruction) qu'il enregistre dans un fichier image exécutable.

Au fur et à mesure qu'un programme prend de l'importance et devient de plus en plus complexe, on a besoin de plusieurs fichiers textes pour le décrire dans son ensemble. La fabrication se déroule alors en deux étapes : la compilation, puis l'édition des liens. Chaque fichier texte fourni au compilateur est traité individuellement pour produire un fichier intermédiaire, dit fichier objet. Une étape supplémentaire, l'édition des liens, se charge ensuite d'agréger tous les fichiers objets en un fichier exécutable.

En réalité, l'édition des liens a lieu même lorsqu'on a qu'un seul fichier d'entrée. En effet l'une des tâches de l'édition des liens est de résoudre des calculs d'adresses qui ne sont pas traités lors de la compilation, ces deux étapes sont simplement transparente pour le programmeur lorsqu'il soumet son unique fichier. De plus les programmeurs utilisent maintenant toujours des fonctions écrites par d'autres et fournies sous forme de bibliothèques qui sont des briques logicielles pré-compilées en fichiers objets et rassemblés dans des fichiers archives que l'édition des liens se chargera d'inclure dans l'exécutable final.

Pour réduire la consommation de mémoire vive et de masse des logiciels, les systèmes d'exploitation modernes proposent un système de liaison dynamique, c'est à dire qu'une partie de l'édition des liens est retardée au démarrage du logiciel. Un tel système sera capable de ne charger en mémoire qu'une seule fois une même bibliothèque dynamique (sur windows ce sont les fichiers .dll) qui est utilisée simultanément par deux logiciels.

Si on en revient aux restrictions portant sur l'édition des liens de la licence GNU GPL, elle interdit à un programme soumis à une licence incompatible avec la GPL de se lier à une bibliothèque GPL mais pas l'inverse. C'est à dire qu'un programme GPL peut utiliser une bibliothèque non-GPL. L'édition des liens se fait selon une relation de dépendance utilisateur/utilisé et compte-tenu du caractère dynamique de l'édition des liens, il en résulte la contrainte concrète suivante : du code GPL ne peut pas être utilisé par du code propriétaire dans un même processus (pour des environnements virtuel de type Java ou .Net on assimilera le processus à une instance de machine virtuelle ce qui peut être extrêmement limitatif avec ces plateformes). Cette contrainte exige donc pour être contournée des efforts importants qui seront de plus pénalisant en terme de performance du résultat. Comme la raison d'être d'une bibliothèque est justement d'être utilisée par d'autres programmes, il a par conséquent été décidé la création d'une nouvelle licence dénommée LGPL pour assouplir cette contrainte et faciliter la diffusion de ces briques logicielles.

Il peut sembler curieux que R. Stallman ait permis qu'un logiciel libre GPL utilise des bibliothèques propriétaires. Cela a été nécessaire à une époque où rien n'existait encore et où il n'était pas non plus raisonnable de tout recommencer de zéro "par le bas", sans avoir au préalable réalisé un certain nombre d'outils libres pour travailler à la suite du projet GNU. Il a toutefois toujours prévenu explicitement que cela ne serait jamais une bonne façon de faire : un logiciel est-il réellement libre si sa mise en œuvre dépend d'un composant critique non libre ? C'est pourquoi il s'est longtemps déclaré opposé à l'utilisation du langage java pour créer des logiciels libres, ce cas particulier ayant été récemment résolu par l'ouverture par SUN de la plateforme Java.

## Table des URL

~ \* ~ \* ~

- [url1] *Free Software Foundation* : <http://www.fsf.org/>
- [url2] *antenne française.* : <http://fsffrance.org/>
- [url3] *don't be evil* : [http://en.wikipedia.org/wiki/Don't\\_be\\_evil](http://en.wikipedia.org/wiki/Don't_be_evil)
- [url4] *l'article Google en français* : <http://fr.wikipedia.org/wiki/Google>
- [url5] *CeCILL* : <http://www.cecill.info>
- [url6] *1/DEF/DGSIC* : [http://www.boc.sga.defense.gouv.fr/visu/boc\\_visu4.php?nor=M0652897X&id=8994](http://www.boc.sga.defense.gouv.fr/visu/boc_visu4.php?nor=M0652897X&id=8994)
- [url7] *FreeBSD* : <http://www.freebsd.org/>
- [url8] *NetBSD* : <http://www.netbsd.org/>
- [url9] *OpenBSD* : <http://www.openbsd.org/>
- [url10] *BSD.* : [http://fr.wikipedia.org/wiki/Berkeley\\_Software\\_Distribution](http://fr.wikipedia.org/wiki/Berkeley_Software_Distribution)
- [url11] *OpenSolaris* : <http://opensolaris.org/>
- [url12] *Sun microsystems.* : <http://www.sun.com/>
- [url13] *Darwin* : <http://www.opensource.apple.com/darwinsource/>
- [url14] *Debian* : <http://www.debian.org>
- [url15] *RedHat* : <http://www.redhat.com/>
- [url16] *Ubuntu* : <http://www.ubuntu.com>
- [url17] *Canonical* : <http://www.canonical.com>
- [url18] *Mandriva* : <http://www.mandriva.com>
- [url19] *...* : [http://fr.wikipedia.org/wiki/Distribution\\_Linux](http://fr.wikipedia.org/wiki/Distribution_Linux)
- [url20] *ReactOS* : <http://www.reactos.org/>
- [url21] *Firefox* : <http://www.mozilla.com>
- [url22] *Thunderbird* : <http://www.mozilla.com>
- [url23] *OpenOffice* : <http://www.openoffice.org>
- [url24] *The Gimp* : <http://www.gimp.org/>
- [url25] *Dia* : <http://live.gnome.org/Dia>
- [url26] *VLC media player* : <http://www.videolan.org/vlc/>
- [url27] *7-zip* : <http://www.7-zip.org/>
- [url28] *GCC* : <http://gcc.gnu.org/>
- [url29] *OpenJDK* : <http://openjdk.java.net/>
- [url30] *Mono* : <http://www.mono-project.com/>
- [url31] *Octave* : <http://www.gnu.org/software/octave/>
- [url32] *Matlab* : <http://www.mathworks.fr/>
- [url33] *Scilab* : <http://www.scilab.org>
- [url34] *KiCad* : [http://www.lis.inpg.fr/realise\\_au\\_lis/kicad/](http://www.lis.inpg.fr/realise_au_lis/kicad/)
- [url35] *Apache* : <http://www.apache.org/>
- [url36] *Sendmail* : <http://www.sendmail.org/>
- [url37] *Bind* : <https://www.isc.org/software/bind>
- [url38] *Samba* : <http://www.samba.org/>
- [url39] *FileZilla Server* : <http://filezilla-project.org/>
- [url40] *OpenSSH* : <http://www.openssh.com/>
- [url41] *MySQL* : <http://www.mysql.fr/>
- [url42] *PostgreSQL* : <http://www.postgresql.org/>
- [url43] *gvim* : <http://www.vim.org>

- [url44] *libxml2* : <http://xmlsoft.org/>
- [url45] *FOP* : <http://xmlgraphics.apache.org/fop>
- [url46] : <http://www.freebsd.org/copyright/license.html>
- [url47] : <http://www.apache.org/licenses/LICENSE-2.0>
- [url48] : <http://www.mozilla.org/MPL/MPL-1.1.html>
- [url49] : <http://www.xfree86.org/current/LICENSE5.html>
- [url50] : <http://www.fsf.org/licensing/licenses/gpl.html>
- [url51] : <http://www.fsf.org/licensing/licenses/lgpl.html>
- [url52] : <http://www.fsf.org/licensing/licenses/fdl.html>
- [url53] : [http://www.cecill.info/licences/Licence\\_CeCILL\\_V2-fr.txt](http://www.cecill.info/licences/Licence_CeCILL_V2-fr.txt)
- [url54] : [http://www.cecill.info/licences/Licence\\_CeCILL-B\\_V1-fr.txt](http://www.cecill.info/licences/Licence_CeCILL-B_V1-fr.txt)
- [url55] : [http://www.cecill.info/licences/Licence\\_CeCILL-C\\_V1-fr.txt](http://www.cecill.info/licences/Licence_CeCILL-C_V1-fr.txt)
- [url56] *OSI* : <http://www.opensource.org>