



Beste de savoir

La GPL, la licence des cornus

12 août 2019

Table des matières

1.	La GPL et le logiciel libre : les deux font la paire	1
1.1.	Logiciel libre : la définition de Richard	1
1.2.	Pas de GPL sans code source	2
1.3.	Quelques nuances	2
2.	La GPL : l'étendard du logiciel libre	3
2.1.	La GPL et l'argent	3
2.2.	La GPL, c'est comme l'entropie...	3
2.3.	Quelques déclinaisons de la licence	4
3.	Fonctionnement de la GPL	4
3.1.	Copyright et copyleft	4
4.	Conclusion	5

La *GNU General Public License* (GPL) a été écrite par Richard Stallmann. Elle fait sa première apparition en février 1989 pour protéger le code des programmes qu'il écrit pour le projet GNU ([1 ↗](#)).

Elle en est actuellement à sa version 3 (GPLv3) et est déclinée en 3 saveurs (la GPL, la LGPL et la AGPL). Elle s'applique à tout ce qui dispose d'un code source et donc en particulier aux programmes informatiques.

1. La GPL et le logiciel libre : les deux font la paire

Vouloir parler de la GPL sans présenter ce qu'est le logiciel libre, ce serait un peu comme expliquer pourquoi les pommes tombent par terre en oubliant de mentionner la gravitation. La GPL et le logiciel libre sont étroitement reliés. Peut-être parce que l'inventeur de la première est le premier à écrire une définition de la seconde ?

1.1. Logiciel libre : la définition de Richard

Richard Stallman définit un *logiciel libre* par quatre grandes libertés qu'il doit donner à son utilisateur :

- **Liberté 0** — liberté d'utiliser le programme comme bon lui semble, pour n'importe quel usage ;
- **Liberté 1** — liberté d'étudier le programme sous toutes ses coutures et de le modifier ;
- **Liberté 2** — liberté de partager le programme avec qui il veut et comme il veut ;
- **Liberté 3** — liberté de partager ses modifications du programme avec qui il veut et comme il veut.

1. La GPL et le logiciel libre : les deux font la paire

Tout programme qui ne respecte pas l'une ou plusieurs de ces règles n'est pas libre et est appelé *logiciel propriétaire*. Bref, n'est pas logiciel libre qui veut !

J'ai repris cette définition de celle donnée par la Free Software Foundation. On ne peut pas comprendre la GPL sans avoir compris cette idée de logiciel libre, alors prenez peut être le temps de relire cette définition, ou d'aller consulter les explications supplémentaires du site de la Free Software Foundation. (2 ↗)

1.2. Pas de GPL sans code source

Cette définition a des conséquences assez immédiates.

Par exemple, pour que la liberté 1 puisse être exercée, un utilisateur doit pouvoir accéder aux sources du programme. Sans ces sources, il ne lui est pas possible d'étudier le fonctionnement du programme ou de l'adapter à ses besoins, etc. En choisissant la GPL, le développeur a l'obligation de donner accès au code source du programme qu'il donne ou vend.

Prenez toutefois garde, logiciel libre logiciel open source ! La différence entre les deux est à deux doigts d'être capilotractée mais mérite toutefois une longue parenthèse à la fin de cet article. [1]

1.3. Quelques nuances

À la première lecture de ces libertés, on peut penser qu'elles nous permettent de faire un peu n'importe quoi. Oui et non. Je ne souhaite pas entrer dans une étude de cas complète de ce qu'on peut faire ou ne pas faire avec la GPL (la FAQ ↗ de la Free Software Foundation remplit déjà parfaitement cette mission (3 ↗)), cependant j'aimerais mettre la lumière sur quelques situations qui m'ont permis de mieux appréhender cette licence.

1.3.1. Question de diffusion

La GPL n'entre vraiment en vigueur que lorsqu'un programme est diffusé. Par exemple, la GPL impose de donner accès aux sources du programme non pas à tout le monde, mais à tous ceux ayant reçu le programme. Autrement dit, ce n'est pas parce qu'un programme est sous la GPL que tout le monde peut en réclamer les sources. Uniquement les personnes qui en possèdent une copie.

1.3.2. Pas de rétrodroits

En quelque sorte, de par l'application des quatre libertés, l'auteur d'un programme perd tous les droits sur les copies qu'il en diffuse. Ainsi, si l'utilisateur décide de modifier le programme, il n'est pas tenu d'en informer le premier auteur. Pas même qu'il n'est tenu de lui fournir le code source de ses modifications s'il en fait la demande.

On voit ainsi des entreprises qui modifient pour leur besoin un programme libre et qui commercialisent ce programme modifié mais sans faire profiter de ces modifications la communauté libre.

2. La GPL : l'étendard du logiciel libre

Si cela peut rendre ronchons les libristes les plus passionnés, cela est en parfait respect avec la GPL.

Remarque : GPL oblige, rien n'empêche les clients de ladite entreprise de reverser à la communauté le programme modifié qu'ils ont acheté.

2. La GPL : l'étendard du logiciel libre

La GPL permet d'incarner la définition du logiciel libre dans un formalisme juridique. N'y allant pas avec le dos de la cuillère, elle a deux implications qui font grincer des dents. En premier, la conséquence des libertés sur la vente de programmes libres et en second l'obligation d'adopter la GPL pour tout programme intégrant un programme protégé par la GPL.

2.1. La GPL et l'argent

Peut-on vendre un logiciel libre ?

Oui. Si vous lisez bien la GPL, vous remarquerez qu'il n'est jamais écrit qu'un logiciel libre ne peut être vendu. Au contraire même, les libertés 2 et 3 vous permettent de diffuser votre copie du programme comme bon vous semble, avec ou sans modifications. Dès lors, rien ne vous empêche de le rendre payant.

Mais vous devez être conscient d'une chose : si vous disposez des quatre libertés, la personne à qui vous l'aurez vendu aussi. Elle pourra alors revendre, ou donner la copie du programme que vous lui aurez vendue, aussi librement que vous. Et vous ne pourrez rien y faire sans rendre votre programme privé.

Je pense que vous voyez assez rapidement le problème que cela peut poser... [2]

2.2. La GPL, c'est comme l'entropie...

L'entropie d'un système fermé ne peut qu'augmenter. La GPL, elle, ne peut que s'étendre. Pour être respectée, la GPL demande qu'un programme qu'elle protège doit rester libre, dans toutes les circonstances et quelles soient les conditions dans lesquelles il est utilisé. Ainsi, si un programme libre est intégré dans un programme plus grand, ce programme doit lui aussi être libre. Pour plagier Wikipedia, c'est ce que j'appellerai le caractère héréditaire de la GPL et c'est sur ce point que les esprits s'échauffent.

Une partie importante des programmes libres sont ce que l'on appelle des bibliothèques. Ce ne sont pas des programmes indépendants qui peuvent s'exécuter seuls, mais plutôt des boîtes à outils spécialisées que tous les autres programmes peuvent venir utiliser. Or, dès qu'un programme fait appel à une bibliothèque sous licence GPL, il doit lui aussi adopter cette licence.

Les choses sont en réalité un peu plus subtiles que cela car l'obligation dépend de la nature de la liaison entre le programme et la bibliothèque.

3. Fonctionnement de la GPL

D'un côté, les libristes utilisent ce principe comme une arme pour propager le logiciel libre : si quelqu'un souhaite utiliser leurs bibliothèques libres, il doit rendre son programme libre, ce qui augmente le volume total de programmes libres. De l'autre, les développeurs de programme privés s'efforcent d'utiliser toutes les entourloupes possibles pour contourner cette règle et utiliser des bibliothèques libres dans leurs propres programmes.

Stallmann le reconnaît lui-même, cette décision est stratégique. (4 [↗](#)) Cette licence permet aux logiciels libres de se battre sur une sorte de pied d'égalité avec le logiciel privé. Les logiciels privés profitent de financements conséquents, les logiciels libres d'un large choix de bibliothèques de qualité!

2.3. Quelques déclinaisons de la licence

L'obligation de protéger avec la GPL tout programme lié à une bibliothèque sous la GPL peut toutefois être un frein pour l'adoption de logiciels libres. Dans ce cas, une licence un tantinet plus faible, la LGPL peut prendre la relève.

La *Lesser General Public Licence* est en tout point identique à la GPL sauf en ce qui concerne l'hérité.

Lorsqu'un programme protégé par la LGPL est intégré dans un plus grand programme, celui-ci n'est pas obligé d'adopter la GPL. L'obligation de donner accès à l'utilisateur au code source et tout le *tutti quanti* des libertés restent et demeurent, mais restent limités au code sous la LGPL.

Grâce à cette licence, certaines bibliothèques libres peuvent donc être adjointe à un programme privé.

Il existe enfin une dernière déclinaison de la GPL. L'*Affero General Public Licence* permet de protéger aussi bien que la GPL les programmes utilisables à travers un réseau, et en particulier les applications web.

3. Fonctionnement de la GPL

3.1. Copyright et copyleft

Pour fonctionner, la GPL utilise le cadre légal du copyright et le retourne à ses propres fins.

Copyright.

Le copyright est un droit décrit dans la loi d'un pays qui garanti au créateur d'un travail intellectuel les droits exclusifs sur son utilisation et sa distribution. Ce droit s'applique souvent pour un temps limité (par exemple jusqu'à un certain nombre d'années après la mort du créateur) avec l'intention de permettre au créateur de recevoir un bénéfice sur le travail qu'il a fourni.

(d'après [Wikipedia](#) [↗](#))

Par l'intermédiaire d'une licence, l'auteur décrit de quelle façon il souhaite disposer des droits qu'il possède sur son travail. La GPL lui permet de donner les 4 libertés fondamentales aux

4. Conclusion

personnes ayant reçu son programme. C'est assez extraordinaire. Alors que le copyright interdit par défaut la diffusion, la modification et le détournement d'un travail sans l'autorisation explicite de l'auteur, les licences libres l'utilisent pour faire entièrement l'inverse ! L'auteur permet à tout le monde de diffuser, modifier et détourner son travail sans qu'un accord supplémentaire soit nécessaire ; la licence fait office d'autorisation.

Lorsqu'une licence impose que les travaux dérivés de ce qu'elle protège soient publiés sous la même licence (comme la GPL), on parle de *copyleft*.

Remarquez que la notion de copyright n'existe pas dans tous les pays. C'est en fait plutôt une notion de droit anglo-saxon. En France, il existe un droit similaire : le droit d'auteur. Par exemple, le copyright américain permet à un auteur de verser son travail dans le domaine public ce que le droit d'auteur français ne permet pas.

4. Conclusion

La GPL est une licence peu ordinaire. Elle défend mordicus les valeurs du logiciel libre tout en assurant une totale liberté aux utilisateurs. Il ne faut toutefois pas oublier qu'il n'y a pas de consensus sur la définition du logiciel libre telle qu'elle est dictée par le Free Software Foundation et sur laquelle la GPL s'appuie. Pour découvrir les licences alternatives à la GPL qui définissent autrement *le libre*, c'est vers les licences open source qu'il faut se tourner.

Bref, la GPL est bien une licence pour gnou¹ qui libre comme l'air, sait aussi se défendre.

~2ohm

Article original sur mon [site perso](#)

Quelques liens et sources

- (1) [The History of the GPL](#) : histoire de la GPL.
- (2) [Qu'est-ce que le logiciel libre ?](#) : définition du logiciel libre.
- (3) [FAQ sur les licences GNU](#) : FAQ de la Free Software Foundation.
- (4) [Copyleft : idéalisme pragmatique](#) : sur le rôle stratégique de la GPL pour la promotion du logiciel libre.
- [Pourquoi vous ne devriez pas utiliser la LGPL pour votre prochaine bibliothèque](#) : essai sur l'utilisation de la LGPL.

Notes

[1] Logiciel Open Source

Un logiciel open source n'est pas nécessairement libre. La façon dont je vois cette différence est que l'open source, c'est comme le logiciel libre mais sans son idéologie. Ses critères sont notamment plus faibles sur la liberté 0. Ainsi, d'après cette liberté, lorsqu'un logiciel libre est utilisé dans un appareil, l'utilisateur doit pouvoir modifier ce logiciel, l'adapter, le mettre à jour, le remplacer librement. Pour respecter la licence de ce programme, le fabricant de l'appareil est même tenu de fournir à l'utilisateur les moyens nécessaires d'exercer cette liberté. Un programme open source n'impose pas ça.

4. Conclusion

L'exemple le plus connu concerne les téléphones sous Android. Android est un logiciel open source, vous pouvez télécharger et consulter son code sur internet. Pourtant vous ne pouvez pas modifier librement la version qui est installée sur votre appareil ; le constructeur vous en empêche.

Pour grattouiller plus sur les détails, consultez là aussi la page de la Free Software Foundation dont je me tire mes informations ainsi que la définition de l'open source sur le site de l'Open Source Initiative.

Voir

- [Pourquoi l'« open source » passe à coté du problème que soulève le logiciel libre](#) ↗
- [The Open Source Definition \(Annotated\)](#) ↗

[2] GPL et profits

En fait, une approche classique pour tirer des profits d'un logiciel protégé par la GPL est d'utiliser une licence double. Le programme sous la GPL est distribué gratuitement ou vendu à une somme symbolique. Ce même programme sous une licence qui accepte qu'il soit intégré dans un logiciel privateur est vendu cher.

Je n'entre pas dans les détails, mais vous renvoie vers la page [wikipedia](#) ↗ .

Un grand merci à [Arius](#) ↗ pour sa relecture!

1. Parce que le gnou s'appelle GNU.