

notamment *compilée*, éventuellement à la condition de rendre disponible le code source correspondant. L'utilisateur doit pouvoir fusionner des logiciels libres dont il n'est pas lui-même l'auteur. La FSF accepte toutefois des restrictions mineures quant à la façon dont un logiciel modifié doit être présenté lorsqu'il est redistribué.

Définition du projet Debian

Le projet *Debian*, commencé en 1993 avec le soutien de la *Free Software Foundation*, a développé *Les principes du logiciel libre selon Debian*^[8] également connus sous l'acronyme *DFSG* (pour *Debian Free Software Guidelines*). Proposé en 1997 par Bruce Perens^[9], le côté pratique y est mis en évidence en 10 points, en précisant par exemple : la non-discrimination des utilisateurs et des usages ; les restrictions acceptables en matière de préservation du code source de l'auteur original ; ou encore l'acceptation explicite de la modification du nom d'un logiciel dont on redistribue une modification (on peut noter à ce propos que les noms de plusieurs logiciels libres sont des *marques déposées* : par exemple *Linux*^[10], *Mozilla*^[11] ou *Apache*^[12]).

Cette définition établie pour *Debian* a été reprise en 1998 dans le cadre de l'*Open Source Initiative* (OSI) pour établir l'*Open Source Definition*. La terminologie *FLOSS* est ensuite apparue pour tenter de réunir les notions de logiciel libre et logiciel open source.

« Logiciel libre » et « open source »



Logo de la *Free Software Foundation*

Littéralement, *open source* signifie « code source ouvert ». Toutefois, on utilise généralement *open source* en référence à l'*Open Source Definition* introduite par l'*Open Source Initiative* (OSI) en 1998, qui souhaitait une autre terminologie pour les logiciels libres, qui se voulait en anglais moins ambiguë et plus adaptée au monde des affaires que *Free Software*.

Le mouvement pour le logiciel libre a défini des règles sur des principes éthiques, celui pour l'open source (qui en découle) a proposé une traduction fonctionnelle. Cela a déclenché un temps quelques différends relatifs au respect de ces principes. Les défenseurs du logiciel libre considèrent que le logiciel libre est une affaire de *philosophie*, tandis que les partisans de l'open source rejettent toute philosophie^[13].

Cette initiative a causé une controverse avec Richard Stallman et la *Free Software Foundation*^[14] qui regrettaient la mise en avant des principes techniques aux dépens de l'*éthique*. Richard Stallman explique aussi pourquoi le logiciel libre est meilleur que l'open source^[15] et pourquoi l'« open source » passe à côté du problème que soulève le logiciel libre^[16].

S'il persiste des désaccords entre ces mouvements, ils restent très proches et un travail conjoint d'harmonisation fait que les définitions officielles du logiciel libre par la *Free Software Foundation* et de l'open source par l'*Open Source Initiative* renvoient dans la pratique aux mêmes

licences, à quelques rares exceptions près (versions 1.x de l'*Apple Public Source License* par exemple^[17]).

Histoire du logiciel libre

Article détaillé : *Histoire du logiciel libre*.

À l'époque des premiers ordinateurs, c'était le *matériel informatique* qui était censé constituer la source de revenus, le *logiciel* n'étant qu'un moyen d'en faciliter la vente. L'accès au *code source* était normal, car nul n'achetait un ordinateur sans disposer d'une équipe de *programmeurs*. Bien avant même la création d'*Unix*, les milieux professionnels et universitaires s'échangeaient volontiers logiciels et codes sources, et les constructeurs cédaient le leur pour rien jusqu'à ce que les lois antitrust le leur interdisent afin de permettre l'exercice d'une concurrence dans ce domaine^{[18],[19]}. En outre, jusqu'aux *années 1970*, il n'était pas encore tout à fait clair que le *droit d'auteur* s'applique aux logiciels.^[contexte nécessaire]

Par décisions de justice^[contexte nécessaire], les constructeurs sont contraints de facturer séparément leurs logiciels au début des *années 1970* ; en quinze ans, l'avènement de la *micro-informatique* va généraliser ce modèle et donner un essor aux éditeurs de logiciels qui s'orientent vers la vente de licences d'utilisation. Un exemple souvent cité pour illustrer ce tournant est une *lettre ouverte de Bill Gates aux hobbyistes*, leur enjoignant de cesser de copier illicitement les logiciels.

Les constructeurs ont parallèlement restreint l'accès au *code source* des programmes, comme en atteste la stratégie OCO, *object-code only* d'IBM à partir du début des *années 1980*. Il devient impossible, et dans certains cas interdit, d'étudier, de corriger ou d'améliorer les logiciels acquis. Non seulement l'utilisateur ne peut plus adapter le logiciel à ses souhaits, mais en cas de *bug*, il se retrouve dépendant du bon vouloir de l'éditeur du logiciel. Enfin, la copie, une opération naturelle pour un ordinateur, devient en règle générale interdite (par défaut, le droit d'auteur interdit la copie non explicitement autorisée). Les logiciels disponibles uniquement sous ces conditions restrictives deviennent alors la règle, et les logiciels jusqu'alors librement échangés se retrouvent souvent intégrés dans des produits commerciaux figés et non partageables.



Richard Stallman, fondateur du Mouvement du logiciel libre

Richard Stallman, alors chercheur au laboratoire d'intelligence artificielle du *Massachusetts Institute of Technology*, ressent profondément ce changement lorsque

les collègues, avec qui il travaillait et échangeait des logiciels jusqu'ici, sont engagés à leur tour pour utiliser et produire des logiciels qu'ils ne pourront plus étudier et partager. Stallman commença à constater ces restrictions en présence de programmes sur lesquels il ne pouvait intervenir, un pilote d'imprimante notamment^[20]. Robert Sproull aurait refusé de lui fournir le code source en raison d'un contrat de non divulgation que Xerox avait passé avec lui, pratique encore peu courante à l'époque^[21].

En 1984 il crée donc le *projet GNU*, qui a pour objectif de construire un système d'exploitation compatible avec *Unix*, et dont la totalité des logiciels est libre. Parallèlement aux travaux de développement engagés, Richard Stallman fonde la *Free Software Foundation*. Afin de donner une assise solide à son projet, Richard Stallman définit précisément la notion de *logiciel libre* et il rédige la *licence publique générale GNU (GPL)* qui utilise le droit d'auteur pour garantir la pérennité des libertés accordées aux utilisateurs (et donc interdire la possibilité qu'une évolution ne soit plus libre). Il publie également le *manifeste GNU* afin de marquer le début du projet GNU, puis les bulletins GNU périodiquement. Le projet débute par le développement de *GNU emacs* un éditeur de texte.

Au début des *années 1990* la majorité du système GNU était fonctionnelle, seul le *noyau* manquait. Le Noyau est la partie d'un système d'exploitation qui communique avec le matériel et alloue les ressources aux autres programmes. Le projet *GNU Hurd* dont les spécifications avaient été définies par Stallman lui-même avait pour but de développer un noyau libre. Cependant le projet prenait du retard, et la conception choisie pour celui-ci paraissait difficile à mettre en œuvre. En 1991, *Linus Torvalds*, un étudiant finlandais, écrit son propre noyau, puis le publie un peu plus tard sous licence GPL. Son noyau, nommé *Linux*, associé aux logiciels GNU précédemment développés, permettait alors de former un système d'exploitation complètement libre et pleinement utilisable^[22].

Les années suivantes ont vu un nombre considérable de projets de logiciel libre se lancer. La *fsf* référence actuellement plus de 30 000 paquets de logiciel libre. Depuis la fin des *années 1990*, le succès des logiciels libres, notamment de *GNU/Linux*, suscite un vif intérêt dans l'industrie informatique et les médias^[Note 4].

Caractéristiques des logiciels libres

Accès aux sources

Les libertés d'étudier et modifier un logiciel supposent un accès au *code source* du logiciel. L'accès au code source est important car les logiciels sont généralement distribués sous une forme *compilée en langage machine*, prêts à être exécutés par un *ordinateur*. Mais le langage machine est très peu lisible par l'homme et rend l'étude du logiciel excessivement pénible voire impossible. L'accès au code source est donc une condition nécessaire pour permettre à l'utilisateur de savoir ce que le programme fait réellement^[23].

Contrôle par l'utilisateur



Capture d'écran de *Trisquel Gnu/Linux*, un système d'exploitation constitué exclusivement de logiciels libres.

Un des objectifs principaux du logiciel libre est de permettre à l'utilisateur d'avoir le contrôle sur son ordinateur et sur les logiciels qu'il utilise^{[24],[25]}. Ce contrôle est donné individuellement : chacun peut étudier en détail ce que fait le logiciel, et le modifier s'il le souhaite. Mais les utilisateurs ont aussi le contrôle de manière collective sur leur ordinateur^[26] : on ne peut étudier l'ensemble des logiciels que l'on utilise, mais on peut être certain que si un logiciel contient une fonctionnalité cachée ou malveillante, elle sera découverte un jour et un correctif sera proposé. Ceci a de plus, pour effet de dissuader, la plupart du temps, les développeurs d'ajouter de telles fonctionnalités^[27].

Richard Stallman pense qu'en utilisant des logiciels propriétaires, les utilisateurs perdent la possibilité de contrôler une partie de leur propre vie^[28]. Les propriétaires de logiciels propriétaires (les développeurs, ou l'entreprise pour laquelle ils travaillent) ont le pouvoir d'espionner ou de restreindre les utilisateurs^[29].

La définition du logiciel libre par la *fsf* précise^[1]:

« *Quand les utilisateurs ne contrôlent pas le programme, c'est le programme qui contrôle les utilisateurs. Le développeur contrôle le programme, et par ce biais, contrôle les utilisateurs. Ce programme non libre, ou « privé », devient donc l'instrument d'un pouvoir injuste.* »

Le 25 octobre 2013, dans l'émission *Ce soir (ou jamais !)*, cette notion de contrôle fait débat entre *Jacques Attali* et *Jérémie Zimmermann*. Jacques Attali affirme "Le mot libre est très mal adapté car les logiciels dit libres sont souvent attachés à une seule entreprise". Selon Jérémie Zimmermann "la liberté pour les logiciels c'est un vrai concept [...] ce que ça veut dire c'est contrôler la machine plutôt que de se laisser contrôler par elle", ce au-quoi réponds Attali que "Si vous laissez chacun avoir la possibilité de contrôler sa confidentialité alors vous rendez impossible l'état de droit".^[30]

Communautés

La qualité du logiciel est souvent proportionnelle au nombre de développeurs. Plus la communauté de développement s'étend, plus elle devient un gage de qualité et de réactivité. De la même manière, la communauté des utilisateurs, ayant comme rôle principal

de faire remonter des dysfonctionnements et des suggestions, a une influence proportionnelle à sa taille.

Les développeurs et utilisateurs de logiciels libres sont généralement indépendants, sans tendance politique particulière, et leurs motivations sont diverses. Ils réalisent ou utilisent des outils fondés sur un mode de production reposant sur la collaboration, l'*entraide*, le *partage*, la *mutualisation*. À ce titre, Bill Gates a comparé les communautés du libre à des « *communistes* d'un nouveau genre »^[31], bien que *Richard Stallman* démente être ou avoir été communiste^[32].

En France, les communautés *Ubuntu* et *Mozilla* comptent parmi les plus actives. Il existe également des associations de défense et promotion du logiciel libre telles que l'*April*, l'*AFUL* ou le réseau *Framasoft*.

Formats ouverts

Les logiciels libres ont, dans leur grande majorité, tendance à respecter les *formats standards ouverts*, ce qui favorise l'*interopérabilité*. Par exemple, le développement du *navigateur web Mozilla Firefox* s'applique à respecter autant que possible les recommandations émises par le *World Wide Web Consortium*. De même, *LibreOffice* et *OpenOffice.org* utilisent le format *OpenDocument*, aux spécifications publiques définies à l'origine par le comité technique d'OASIS et certifié comme norme *ISO*^[33], incarnant ainsi une valeur pérenne pour les applications bureautiques. L'utilisation de formats ouverts dont l'ensemble des spécifications techniques sont connues, garantit l'accès aux documents numériques archivés.

Toutefois, la préférence pour les formats ouverts, si elle est effectivement constatée, n'est pas un élément de la définition du logiciel libre.

L'adhésion à une politique de formats ouverts touche à présent largement le monde de l'industrie informatique. À titre d'exemple, les spécifications du langage *C#*, une création de *Microsoft* fortement inspirée par le langage *Java* de *Sun Microsystems*, ont été publiées et rendues exploitables par d'autres acteurs de l'informatique avant même celles de *Java*.

Commercialisation

La commercialisation des logiciels libres est possible mais la liberté n° 2 en interdit l'exclusivité. Cette caractéristique rend délicat le retour sur investissement financier pour les développeurs du logiciel par la vente du logiciel. Autrement dit: les logiciels libres ne permettent pas de rétribution directe des auteurs. C'est pour cette raison que ces derniers se tournent souvent vers la vente de services associés à l'utilisation du logiciel (voir l'article consacré aux *sociétés de services en logiciels libres*).

Les licences libres les plus « contraignantes » imposent que tout projet qui réutilise le code source du logiciel libre devienne lui-même un logiciel libre. Les éditeurs traditionnels s'y refusant catégoriquement, cette disposition les enferme dans le monde du « non-libre » (l'univers des logiciels propriétaires). Il se crée alors un fossé entre les deux mondes, qui est dommageable du point de vue de l'interopérabilité et de l'harmonie entre des systèmes libres et propriétaires qui sont amenés à « cohabiter ».

Il existe aussi une forme de distribution mixte: il s'agit d'un système de *double licence* selon lequel l'éditeur propose d'une part une distribution libre (et généralement gratuite) et, en parallèle, une distribution fermée et payante, assortie des conditions commerciales classiques des logiciels non-libres (et parfois de fonctionnalités supplémentaires).

La politique controversée des grands comptes

Certains partisans du logiciel libre voient une incohérence dans la politique de certaines entreprises ayant affiché une adhésion aux principes de l'*open source* mais étant favorables à l'extension du logiciel au champ du brevetable. Pour eux, le logiciel libre (via la *Free Software Foundation*) lutte ouvertement contre la brevetabilité des logiciels et, conséquemment, contre l'exclusivité d'exploitation d'une idée par une seule personne ou entreprise. Ils craignent qu'à terme, certains membres de l'*open source* détiennent des portefeuilles de brevets sur *des idées*, leur permettant de réclamer des redevances sur chaque portion de code source.

IBM, par exemple, déposa en *Europe* et aux *États-Unis* certains algorithmes cruciaux dans le fonctionnement général du noyau ou du système d'exploitation *GNU/Linux* et porta devant des tribunaux des affaires afin que les *brevets logiciels* soient institués et reconnus en *Europe*. Pourtant, cette même entreprise autorise l'usage de 500 de ces brevets^[34] aux développeurs de logiciels libres et open source.

Dans la pratique, tout procès en *propriété intellectuelle* coûte en temps et en argent. Il est donc difficile pour la communauté *open source* de faire valoir ses droits face aux *multinationales*. Certains de ses détracteurs l'accusent de recevoir parfois d'une multinationale des *coups de pouce* l'aidant à lutter contre l'influence d'une autre (ce fut un moment le cas entre *Sun* et *Microsoft* dans l'affaire *OpenOffice*)^[réf. nécessaire]. *IBM* finance également la communauté des logiciels libres contre les actions de *Microsoft*, etc.^[réf. nécessaire]. Le système *no win, no pay* autorisé aux États-Unis (et qui fut utilisé pour défendre les victimes pauvres de *Bhopal*) peut éventuellement permettre à la communauté du libre de défendre ses droits, bien que les dossiers soient par nature plus complexe, demandant parfois d'examiner des centaines de milliers de lignes de code.

Sécurité

La sécurité relative des logiciels libres et propriétaires est sujette à débat.

- le libre accès au *code source* permet l'examen du logiciel par des experts indépendants ;
- le libre accès au *code source* rend impossible le recours à la *sécurité par l'obscurité*, ce qui est considéré comme un avantage ou un défaut, selon le point de vue ;
- la découverte de *failles de sécurité* est facilitée par la publication du code source. En effet, l'ouverture du code permet statistiquement à un plus grand nombre de personnes d'avoir la possibilité de repérer et de corriger des vulnérabilités.

Innovation

L'effet du logiciel libre sur l'innovation est sujet à débat.

Les partisans du logiciel libre considèrent qu'il permet de garantir une activité innovante. Parce qu'un large panel de contributeurs peuvent y apporter de nouvelles fonctionnalités. Ce dynamisme accélérerait les échanges et favoriserait l'innovation alors que le principe de "verrouillage" nuirait à ce dynamisme. Enfin l'accès à l'intégralité du code source ne permet pas de garder longtemps un avantage concurrentiel ce qui pousserait à l'innovation. ^[réf. souhaitée]

Certains innovateurs comme *Pranav Mistry* du *Media lab* (MIT, États-Unis) proposent^[35] de mettre à la fois les innovations matérielles et les innovations logicielles en licence libre pour tous.

De l'autre coté certains acteurs du monde du *logiciel propriétaire* considèrent au contraire que le logiciel libre peut avoir un effet négatif sur l'innovation en décourageant l'investissement d'acteurs privés^[36].

Financement

Il existe une grande disparité à l'intérieur des logiciels sous "licence libre" en ce qui concerne leur financement. D'une part une grande majorité de petits logiciels non financés (bénévolat), et d'autre part des logiciels majeurs du marché, financés principalement par les grands acteurs de l'industrie informatique, ainsi *Linux* est financé entre autres par *Red Hat*, *IBM* et *Intel*, *Firefox* par *Google*.

En France les entreprises consacrent « 16 % de part de budget informatique alloués aux logiciels libres^[37] »

Licences

Base légale

Licences d'exploitation des œuvres de l'esprit

Lois et principes sous-tendant les licences

Copyleft

Domaine public

Droit d'auteur

Open source

Open Source Definition

Classification des licences

Licences de libre diffusion

- CC-**-*NC
- CC-**-*ND

Licences libres

- *Licences libres copyleft*
 - *Licence Art Libre*
 - *Licence CeCILL*
 - *CC-BY-SA*
 - *GFDL*
 - *GPL*
 - *LGPL*
 - *ODbL*
- *Licences libres non copyleft*
 - *Licence BSD*
 - *CC0*
 - *CC-BY*
 - *Licence X11 ou MIT*

Licences propriétaires

- Beaucoup de licences spécifiques de logiciels commerciaux
- *Licences de partagiciels*

Mouvements utilisant des licences libres ou de libres diffusion

Culture

- *Art Libre (libre)*
- *Culture Libre (libre)*
- *Musique Libre (libre diffusion)*

Logiciel

- *Graticiel (libre diffusion)*