



Développement de logiciels libres

Chapitre 4 Histoire du logiciel libre



Pablo Rauzy <pr@up8.edu>
pablo.rauzy.name/teaching/dll

Histoire du logiciel libre

- ▶ Jusque dans les années 1970, les logiciels étaient utilisables, étudiables, et modifiables par leurs utilisateurs, du moins dans le cadre universitaire.
- ▶ Cela n'était pas le résultat de règles ou de lois mais simplement dû au fonctionnement de la recherche.

- ▶ Jusque dans les années 1970, les logiciels étaient utilisables, étudiables, et modifiables par leurs utilisateurs, du moins dans le cadre universitaire.
- ▶ Cela n'était pas le résultat de règles ou de lois mais simplement dû au fonctionnement de la recherche.
- ▶ En dehors du milieu académique, seules de grosses compagnies pouvaient acheter des ordinateurs.
- ▶ Les logiciels (systèmes et applications) étaient fournis avec les ordinateurs, et pas vu comme un produit commercial.
- ▶ Autant les vendeurs que les acheteurs avaient intérêts à permettre la création de communautés de développeurs pour favoriser l'amélioration des logiciels.

- ▶ C'est dans ce contexte que dans les années 1980, Richard Stallman, alors au MIT, commence à constater des restrictions dans les possibilités d'utilisation de certains logiciels.
- ▶ Il se trouve face au problème éthique de devoir développer des logiciels dont l'utilisation sera restreinte, qui ne pourront pas être partagés en raison des droits du propriétaire du logiciel (souvent distinct du créateur).
- ▶ L'anecdote raconte que l'élément déclencheur dans l'esprit de RMS est un pilote d'imprimante cassé qu'il voulait réparer.
- ▶ Robert Sproull, l'auteur du pilote, 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.

- ▶ Bien qu'anecdotique, cette petite histoire est souvent prise comme étant le point de départ de l'informatique libre.
- ▶ C'est à partir de là que RMS consacre son énergie à résoudre ce problème de conscience.
- ▶ Cela fait de lui le premier et le plus emblématique des ambassadeurs du logiciel libre.

Problèmes

- ▶ Analogie abusive avec les objets matériels :
 - création artificielle d'une économie de la rareté.

Problèmes

- ▶ Analogie abusive avec les objets matériels :
 - création artificielle d'une économie de la rareté.

- ▶ Analogie avec les recettes de cuisines :
 - le problème n'est pas seulement de permettre la réutilisation du code source, mais de faire en sorte que l'information qu'il contient reste en libre circulation, et que les avantages qu'il procure restent librement utilisables.

Problèmes

- ▶ Analogie abusive avec les objets matériels :
 - création artificielle d'une économie de la rareté.
- ▶ Analogie avec les recettes de cuisines :
 - le problème n'est pas seulement de permettre la réutilisation du code source, mais de faire en sorte que l'information qu'il contient reste en libre circulation, et que les avantages qu'il procure restent librement utilisables.
- ▶ Les logiciels sont des moyens de productions :
 - la propriété lucrative des logiciels, à opposer à la propriété d'usage, pose le problème de la dépossession de ses outils de travail et donc une forme d'aliénation.

Solution

- ▶ RMS lance alors le projet GNU en 1984.
- ▶ GNU, pour *GNU's Not Unix*, se veut être un clone libre et compatible d'UNIX, afin de faciliter la transition.
- ▶ Dans un premier temps, il écrit les core-utils, un éditeur de texte (Emacs), et un compilateur (GCC).
- ▶ Le projet GNU prévoit aussi un noyau pour son système : Mach.
- ▶ C'est finalement le noyau Linux (publié en 1991 par Linus Torvalds) qui sera utilisé, d'où l'appellation GNU/Linux.
- ▶ La première version fonctionnelle entièrement libre est construite en 1992.

- ▶ Une première ébauche de définition du logiciel libre est proposée en 1986 par RMS :
« *Premièrement, la liberté de copier un programme et de le redistribuer à vos voisins, qu'ils puissent ainsi l'utiliser aussi bien que vous. Deuxièmement, la liberté de modifier un programme, que vous puissiez le contrôler plutôt qu'il vous contrôle ; pour cela, le code doit vous être accessible. »*
- ▶ Aujourd'hui plusieurs définitions co-existent.

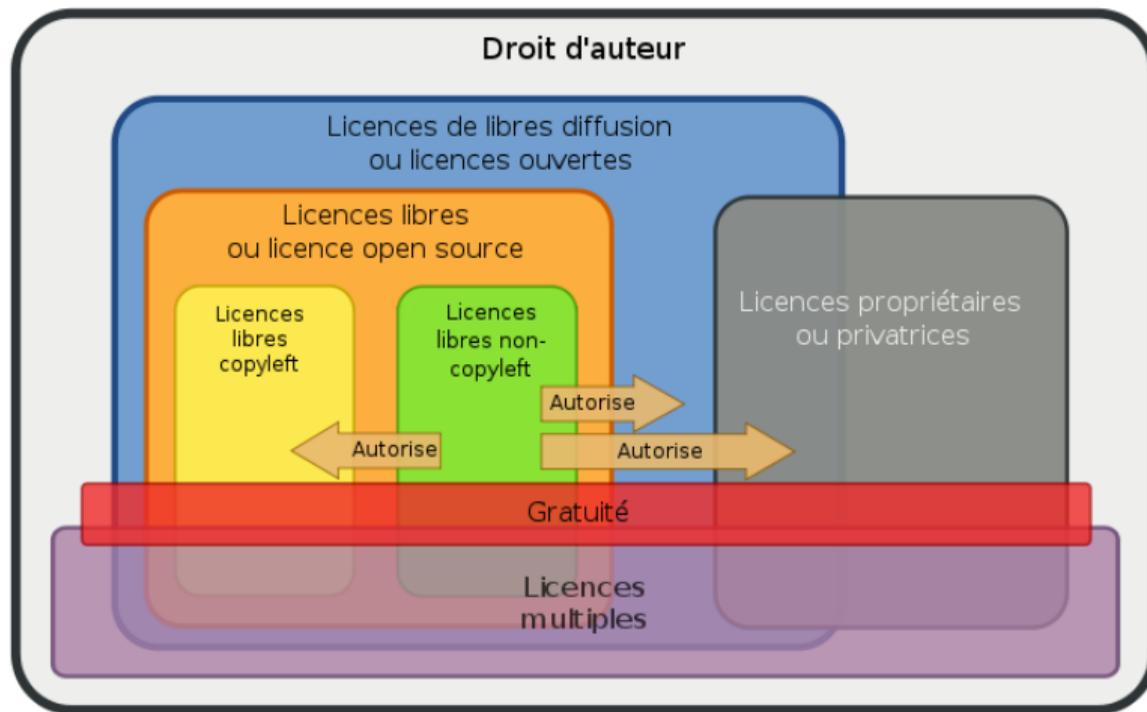
Définition de la Free Software Foundation

- ▶ Pour la FSF, fondée par RMS pour porter le projet GNU, un logiciel est *libre* si il confère à ses utilisateurs ces quatre libertés :
 0. la liberté d'exécuter le programme, pour tous les usages ;
 1. la liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
 2. la liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies) ;
 3. la liberté d'améliorer le programme et de distribuer ces améliorations au public, pour en faire profiter toute la communauté.
- ▶ Ces libertés doivent être irrévocables.

Définition de l'Open Source Initiative

- ▶ Pour être accepté comme *open source* par l'OSI, un logiciel doit être distribué dans le respect des critères suivants :
 - permettre la redistribution librement,
 - le code source doit être disponible,
 - il doit être possible de créer des oeuvres dérivées,
 - le respect de l'intégrité du code de l'autre original,
 - la non-discrimination contre les personnes ou les groupes,
 - le non-discrimination contre les cas d'utilisation,
 - la licence doit s'appliquer à tous,
 - la licence ne doit pas être spécifique au produit,
 - la licence ne doit pas imposer de restriction sur d'autres logiciels,
 - la licence doit être technologiquement neutre.

- ▶ Ces critères sont repris (avec modifications) des Debian Free Software Guidelines.



- ▶ *« 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. »*
- ▶ Distinction entre le copyleft fort (contaminant) et le copyleft faible.

- ▶ *SPDX* est un format utilisé pour documenter des informations sur les licences de logiciels.
- ▶ Il est aussi utilisé pour avoir des méta-données automatiquement exploitables.
- ▶ Vous pouvez voir en commentaire, en entête de certains fichiers de code source, un identifiant de licence au format SPDX.
 - Exemple : `SPDX-License-Identifier: GPL-3.0-or-later`.

- ▶ Le projet GNU a créé une licence libre pour sa documentation, la FDL.
 - La *Free Documentation Licence* était à l'origine la licence de Wikipédia.
- ▶ Depuis, le principe des licences libre a pu être étendu à bien d'autres domaines.
 - Exemples : licence Art libre, Artistic licence, etc.

- ▶ Les licences libres les plus connues et utilisées pour les œuvres culturelles sont les licences *Creative Commons*.
- ▶ Elles sont composées à partir de quatre conditions d'usage :
 -  Attribution (BY),
 -  Non Commercial (NC),
 -  No Derivative (ND),
 -  Share Alike (SA).
- ▶ Six compositions sont valides :
 - CC-BY-NC-ND, (considérée non-libre)
 - CC-BY-NC-SA, (considérée non-libre)
 - CC-BY-NC, (considérée non-libre)
 - CC-BY-ND, (considérée non-libre)
 - CC-BY-SA,
 - CC-BY.
- ▶ À cela s'ajoute la licence spéciale CC0, qui simule le domaine public.

- ▶ Les licences Creative Commons ont l'avantage d'exister sous trois formes :
 - Juridique, légalement valide dans de nombreuses juridictions (notamment adaptée aussi bien au copyright qu'aux droits d'auteurs).
 - Compréhensible, avec un résumé clair des droits et devoirs donnés par la licence.
 - Informatique, sous forme de métadonnées dans les documents qui permettent de détecter la licence automatiquement.