Chapitre I
Le logiciel

Cette partie contient :

- Logiciel, données, code
- Différents types de logiciel
- Origine des mots software et hardware
- Logiciel privateur, logiciel libre
- Contexte historique
- Logiciel libre, logiciel open source
- Une scène primitive
- Philosophie du logiciel libre
- Psychologie du développeur
- Au-delà de la psychologie
- Services se substituant au logiciel
- Les « quatre libertés essentielles » du logiciel libre
- « Libre » n'est pas « non commercial »
- Anecdote
- Ivan Illich et l'outil juste
- Evgeny Morozov et Ivan Illich
- Et Linux ?

Logiciel, données, code

Je donne du logiciel la définition naïve que je m'en suis formée : un logiciel est une suite d'instructions qui permet à un ordinateur de manipuler des données. On parle aussi de programme informatique.

Quant aux données – autre définition vague et peu technique –, c'est à peu près tout ce qui peut se convertir en information, c'est-à-dire en 0 et en 1 : des mots, des chiffres, du son, de l'image, des relations entre des données ou encore un logiciel.

Les instructions qui constituent un logiciel sont appelées son code. Celui-ci prend deux formes, code source et code binaire.

Le code source est un texte rédigé dans un langage dit de programmation. La personne qui écrit du code est appelée un développeur : elle développe un logiciel. Exemples de langages de programmation : C, Python, JavaScript ou PHP.

Pour être exécuté par un ordinateur, le code source doit être traduit en langage machine, suite de 0 et de 1 : il s'agit du code binaire. L'opération qui consiste à passer du premier au second s'appelle la compilation : on compile des sources pour obtenir un fichier binaire, aussi appelé exécutable.

Différents types de logiciel

On distingue généralement trois types de logiciel :

- le système d'exploitation ;
- les bibliothèques ;
- les applications.

Je me représente le système d'exploitation (ou OS, pour *operating system*) comme le programme qui fait tourner la machine. Il sert d'interface entre les différents logiciels et l'utilisateur d'une part, et le matériel d'autre part. Il coordonne l'exécution des différents programmes, assurant l'accès de chacun aux ressources matérielles telles que la mémoire, les processeurs – composants où sont effectués les calculs sur les données – ou divers périphériques – clavier, écran ou tout autre dispositif permettant de donner des informations à la machine et d'en recevoir.

(En somme, si l'on considère un ordinateur comme un tas de ferraille parcouru d'impulsions électriques, le système d'exploitation est le logiciel qui permet que ces impulsions ne circulent pas de façon trop désordonnée et que la machine agisse rapidement, efficacement, sans perdre d'information et sans planter.)

Les bibliothèques regroupent des portions de code, appelées fonctions ou routines, qui ne forment pas par elles-mêmes un programme complet mais auxquelles les différents logiciels peuvent faire appel ; cela évite aux développeurs de réécrire ce code et permet de réduire la taille des exécutables. Les fonctions contenues dans les bibliothèques peuvent être très générales, comme ouvrir une fenêtre ou enregistrer un fichier, ou plus spécifiques.

Les applications, enfin, sont des logiciels permettant à leurs utilisateurs d'effectuer des tâches particulières. Des exemples : jeu, navigateur Web, logiciel de messagerie ou de traitement de texte, lecteur de musique ou de vidéo.

J'ai le sentiment que la finalité du système d'exploitation se trouve dans l'ordinateur lui-même, que celle d'une bibliothèque se trouve dans les logiciels et celle d'une application, hors de la machine, dans les buts que poursuit l'utilisateur.

Un classement est toujours imparfait. Je mentionne d'autres logiciels, dont je ne sais s'ils ressortisent au système d'exploitation ou aux applications, tels que les utilitaires (ou outils système), qui permettent à l'utilisateur d'interagir avec le système d'exploitation et de paramétrer le fonctionnement de l'ordinateur, ou les logiciels de programmation.

Origine des mots software et hardware

En anglais, le matériel informatique se dit *hardware*, et le logiciel *software*. Ces mots sont parfois repris en français sans être traduits.

Je lis sur le site du *Trésor de la langue française informatisé*, que le terme *hardware*, composé de *hard*, « dur », et de *ware*, « marchandise », désignait à l'origine un « article de métal », de la « quincaillerie », puis s'est vu, dans les années 1940, appliquer au matériel des machines à calculer.

Dans les années 1950, un mot s'est imposé pour désigner le logiciel, c'est-à-dire ce qui, dans un ordinateur, n'est pas du matériel mais un programme qui peut être modifié : c'est le mot *software*, dans lequel l'adjectif *hard* est remplacé par son contraire. Le *software*, terme qui se traduit littéralement par « marchandise molle » ou « marchandise douce », ne peut se comprendre que dans son opposition au *hardware*.

Logiciel privateur, logiciel libre

La personne ou l'équipe qui a développé un logiciel peut décider d'en publier le code source, ou au contraire de le tenir caché et de n'en distribuer (en général contre rétribution) que le code binaire.

Dans le premier cas, le logiciel est dit **libre** (*free software* ou *open source*) ; son code peut être consulté, modifié, redistribué par tous. Il est dit **privateur** (ou propriétaire, de l'anglais *proprietary software*) dans le second cas.

Les utilisateurs et partisans du logiciel libre se sont donné le nom de libristes ; quant aux utilisateurs de logiciels privateurs, ce sont soit des clients, soit des pirates.

Contexte historique

Le logiciel libre est apparu quand le logiciel a cessé d'être libre. Jusqu'à la fin des années 1960, les constructeurs de machines fournissaient les sources des logiciels qui les accompagnaient. Peu à peu certains constructeurs, tout en distribuant les sources, et en y autorisant l'apport de modifications, en ont interdit la redistribution.

En 1974 aux États-Unis, le code source est soumis au copyright. Des entreprises spécialisées dans le logiciel se développent : elles vendent leur produit sous licence.

En 1976, Bill Gates adresse une *Open Letter to Hobbyists* (*Lettre ouverte aux informaticiens amateurs*) : alors que l'informatique devient accessible aux particuliers, le jeune homme déplore que la circulation de copies des logiciels empêche la juste rétribution de leurs développeurs. Sa lettre vise moins le grand public que les deux cent cinquante membres du Homebrew Computer Club, société d'informaticiens créée en Californie l'année précédente, qui a joué un grand rôle dans le développement de l'informatique ; Steve Jobs en a fait partie.

Peu à peu, les entreprises qui commercialisent les logiciels prennent l'habitude de n'en distribuer que la version binaire. Le logiciel n'est plus libre.

Logiciel libre, logiciel open source

L'idée de logiciel libre apparaît dans la première moitié des années 1980, sous l'impulsion de Richard Stallman ; celui-ci lance le Projet GNU (dont le but est de mettre au point un système d'exploitation qui soit libre) en 1984 et la *Free Software Foundation* (« Fondation pour le logiciel libre ») en 1985.

Le mouvement *open source*, lancé en 1998 par l'*Open Source Initiative*, naît d'une scission au sein de la communauté du libre.

« Libre » et « *open source* » ne sont pas synonymes. Il existe une différence idéologique entre ces deux termes. Le logiciel libre est un projet éthique : ses partisans veulent changer la société. Les tenants de l'*open source* sont plus pragmatiques : ils s'intéressent aux conséquences techniques de la publication des codes, estimant qu'elle donne les meilleurs résultats.

L'histoire a retenu une autre date, celle du lancement du noyau de système d'exploitation Linux, en 1991, par Linus Torvalds. Nous reparlerons bientôt de ce qu'est un noyau de système d'exploitation et de Linux.

Une scène primitive

La plupart des écrits traitant du logiciel libre font le récit suivant, qui met en scène Richard Stallman et une imprimante.

Nous sommes au début des années 1980. L'imprimante Xerox du laboratoire dans lequel travaille Stallman souffre de bourrages fréquents. Désireux de mettre fin au problème, notre homme entre en relation avec la société qui commercialise la machine, demande le code du logiciel qui la fait fonctionner, se le voit refuser.

Contraint de composer avec un matériel qui fonctionne mal, qu'il pourrait pourtant réparer s'il disposait des sources du programme, il comprend qu'il est victime des pratiques restrictives qui se répandent dans le monde de l'informatique, informatique qui, elle-même, se répand dans le monde.

C'est une épiphanie. Stallman se lance dans l'aventure du logiciel libre.

Philosophie du logiciel libre

Je trouve ces éléments dans un article de Richard Stallman intitulé « Le logiciel libre est encore plus essentiel maintenant ». Ce que le logiciel libre défend, c'est la liberté de ses utilisateurs. Gratuit ou payant, un logiciel privateur donne au logiciel, donc à son développeur, du pouvoir sur ses utilisateurs. Ce pouvoir peut être utilisé pour garder captif un public, qu'il empêchera d'utiliser des logiciels concurrents, mais aussi à des fins d'espionnage ou pour permettre à un tiers d'envoyer des commandes au logiciel.

Il ajoute que « toutes les formes de dommage indirect sont amplifiées lorsque l'utilisateur est une institution publique ou une école ». (On comprend que la question du libre n'est pas seulement individuelle ; elle est surtout politique.)

Le logiciel libre, en revanche, offre le contrôle à son utilisateur : ce contrôle est soit individuel (l'utilisateur est capable d'auditer le programme), soit collectif (un utilisateur donné ne saura pas forcément régler le problème qu'il rencontre ni effectuer la modification qu'il souhaiterait voir apporter, mais quelqu'un dans la communauté d'utilisateurs pourra le faire).

Psychologie du développeur

Au détour de quelques phrases du même article, Stallman esquisse une étude psychologique :

- « Les développeurs sont corrompus par le pouvoir qu'ils possèdent » ;
- Le contrôle qu'il a sur le logiciel « induit chez le développeur la tentation de faire du tort aux utilisateurs par d'autres moyens encore » ;
- « Même quand un logiciel privateur n'est pas franchement malveillant, ses développeurs sont incités à le rendre addictif, dominateur et manipulateur. »

Au-delà de la psychologie

Dire que le logiciel privateur corrompt même ceux qui le contrôlent, cela autorise à se représenter les créateurs de Google ou de Facebook comme des êtres à nous semblables, que leur création aurait changé.

Cette psychologie de comptoir s'oppose à certaines réflexions que j'ai entendues il y a quelques années, qui suivaient ce raisonnement : « Google a beaucoup de pouvoir, donc il représente un danger, mais j'ai confiance dans ses dirigeants ; c'est quand Sergey Brin et Larry Page quitteront leur poste qu'il faudra s'inquiéter. » Non. Il faut s'inquiéter parce que Google existe.

En passant du psychologique au systémique, on peut réfuter cette idée qu'une administration ne recourrait pas aux moyens technologiques dont elle dispose : quelle que soit la couleur politique revendiquée du gouvernement, un État surveillera toujours ses citoyens (et les autres) autant que la technologie dont il dispose le lui permet. Il est peu vraisemblable qu'un outil mis à la disposition d'un gouvernement dans un contexte précis ne soit pas appliqué, une fois son efficacité démontrée, et peut-être même d'ailleurs si cette efficacité n'a pas été démontrée dans le contexte de départ, à d'autres fonctions.

Services se substituant au logiciel

Toujours dans le même article, Stallman évoque la notion de « services se substituant à un logiciel » (*SaaS* pour *Software as a Service*), auxquels il adresse les mêmes reproches qu'au logiciel privateur.

Ces services, apparus plus récemment, sont tout ce qu'il est possible de faire sur ce qu'on appelle en anglais le *cloud* – et, plus rarement, en français, l'informatique nuageuse –, c'est-à-dire sur un ordinateur dont un autre que nous a le contrôle : travailler des documents texte depuis un site appartenant à Google, se servir d'Instagram pour poster

des photos, créer un Doodle ou une cagnotte, etc. (J'ai ajouté ces exemples, que Stallman ne donne pas. Beaucoup d'autres se présenteront par la suite.)

Je cite Stallman :

> Si vous utilisez un [tel service], l'opérateur du serveur contrôle votre informatique. Cela nécessite de confier toutes les données concernées à cet opérateur, qui sera à son tour obligé de les fournir à l'État. Qui ce serveur sert-il réellement en fin de compte ?

En bon idéologue, Stallman a pris l'habitude d'appeler les utilisateurs de Facebook des « utilisés » (*useds* au lieu de *users*).

Les « quatre libertés essentielles » du logiciel libre

Je recopie depuis le site du Projet GNU :

> Un programme est un logiciel libre si vous, en tant qu'utilisateur de ce programme, avez les quatre libertés essentielles :
>
> - la liberté de faire fonctionner le programme comme vous voulez, pour n'importe quel usage (liberté 0) ;
> - la liberté d'étudier le fonctionnement du programme, et de le modifier pour qu'il effectue vos tâches informatiques comme vous le souhaitez (liberté 1) ; l'accès au code source est une condition nécessaire ;
> - la liberté de redistribuer des copies, donc d'aider les autres (liberté 2) ;
> - la liberté de distribuer aux autres des copies de vos versions modifiées (liberté 3) ; en faisant cela, vous donnez à toute la communauté une possibilité de profiter de vos changements ; l'accès au code source est une condition nécessaire.
>
> Un programme est un logiciel libre s'il donne toutes ces libertés aux utilisateurs de manière adéquate. Dans le cas contraire, il est « non libre ». Bien que nous puissions faire une distinction entre différents schémas de distribution non libres, en quantifiant ce qui leur manque pour être libres, nous les considérons tous comme équivalents dans leur manque d'éthique.

« Libre » n'est pas « non commercial »

On trouve sur la même page ces considérations :

> « Logiciel libre » ne signifie pas « non commercial ». Un logiciel libre doit permettre l'usage commercial, le développement commercial et la distribution commerciale. Le développement commercial de logiciel libre n'est plus l'exception ; de tels logiciels libres commerciaux sont très importants. Vous pouvez avoir payé pour obtenir une copie d'un logiciel libre ou vous pouvez l'avoir obtenu gratuitement. Mais quelle que soit la manière dont vous vous l'êtes procuré, vous avez toujours la liberté de copier et de modifier le logiciel et même d'en vendre des copies.

Plus bas :

> Quand vous parlez de logiciel libre, il est préférable de ne pas utiliser de termes comme « donner » ou « gratuit », car ils laissent supposer que la finalité du logiciel libre est le prix et non la liberté.

Anecdote

Il y quelques années, le logiciel que j'utilisais le plus au bureau, s'est mis à fonctionner de façon étrange, me faisant perdre un temps infini. C'était un logiciel privateur.

J'ai posté des messages sur un forum spécialisé dans l'application que j'utilisais et qui était probablement le plus gros forum en langue française qui lui fût consacré. Un des modérateurs, qui a fait preuve d'une grande disponibilité, m'a écrit qu'il était « impossible de savoir ce que le logiciel faisait réellement à l'arrière-plan ». Cette impossibilité est caractéristique du privateur.

Contrairement à Stallman, je n'ai pas connu d'épiphanie ce jour-là. Il a fallu que je rédige ce texte pour que je mesure toute l'horreur de ma situation. Trois ans après cet échange, mon problème n'est d'ailleurs pas réglé.

Ivan Illich et l'outil juste

Je viens de relire *La Convivialité*, d'Ivan Illich, paru en 1973 en anglais sous le titre *Tools for Conviviality*. Je suis frappé par la convergence des réflexions de Stallman sur le logiciel libre et de celles d'Illich à propos de l'outil juste et de la société conviviale.

Illich est né en 1926, Stallman en 1953 : une génération les sépare – mais le premier a surtout publié à partir des années 1970 (*La Convivialité* date de 1973) et le second intègre le MIT, où il sera d'abord étudiant, puis employé, en 1971. Que Stallman ait lu Illich ou non importe peu : contemporains, ils assistent aux mêmes bouleversements, dressent peut-être les mêmes constats, partagent des inquiétudes. (Étant donné que peu d'outils échappent aujourd'hui à l'informatisation, on peut affirmer que la question de l'informatique a dévoré celle de l'outil.)

Je recopie quelques phrases d'Illich. Les numéros de pages placés entre crochets font référence à l'édition de *La Convivialité* parue dans la collection Points, aux éditions du Seuil, en 2014.

> J'appelle *société conviviale* une société où l'outil moderne est au service de la personne intégrée à la collectivité, et non au service d'un corps de spécialistes. Conviviale est la société où l'homme contrôle l'outil. [p. 13]

> L'outil juste répond à trois exigences : il est générateur d'efficience sans dégrader l'autonomie personnelle, il ne suscite ni esclaves ni maîtres, il élargit le rayon d'action personnel. L'homme a besoin d'un outil *avec lequel travailler*, non d'un outillage qui *travaille à sa place*. Il a besoin d'une technologie qui tire le meilleur parti de l'énergie et de l'imagination personnelles, non d'une technologie qui l'asservisse et le programme. [p. 27]

> Pour autant que je maîtrise l'outil, je charge le monde de mon sens ; pour autant que l'outil me domine, sa structure me façonne et informe la représentation que j'ai de moi-même. L'outil convivial est celui qui me laisse la plus grande latitude et le plus grand pouvoir de modifier le monde au gré de mon intention. L'outil industriel me dénie ce pouvoir ; bien plus, à travers lui, un autre que moi détermine ma demande, rétrécit ma marge de contrôle et régit mon sens. [p. 44]
>
> L'outil est convivial dans la mesure où chacun peut l'utiliser, sans difficulté, aussi souvent ou aussi rarement qu'il le désire, à des fins qu'il détermine lui-même. L'usage que chacun en fait n'empiète pas sur la liberté d'autrui d'en faire autant. [p. 45]

Ces deux phrases encore, que j'aime bien :

> L'illusion consistait à croire que la machine était un homme artificiel qui remplacerait l'esclave. [p. 41]
>
> L'outil simple, pauvre, transparent est un humble serviteur ; l'outil élaboré, complexe, secret est un maître arrogant. [p. 101]

Deux commentaires :

- Quoi que puissent laisser penser ces extraits, l'opposition que fait Illich entre outil dominant et outil convivial n'est pas binaire. Il précise, page 28 : « Ces outils peuvent se ranger en une série continue avec, aux deux extrêmes, l'outil dominant et l'outil convivial. »
- Un logiciel ou une machine qui ne sont pas conviviaux mais privateurs nous rendent passifs et nous dévalorisent – certains de mes proches, que j'entends parfois dire d'eux-mêmes, avec un ton d'impuissance et de résignation mêlées, qu'ils ne comprennent rien à l'informatique, se reconnaîtront peut-être dans la phrase d'Illich à propos de l'outil qui « nous façonne et informe la représentation que nous avons de nous-mêmes ». Un tel outil est un bloc opaque, qui se refuse à nous, qu'on finit par regarder comme doté d'un pouvoir surnaturel et peut-être d'une volonté propre. C'est aussi un objet qu'on renonce d'emblée à comprendre, donc à changer ou à faire nôtre ; il nous soumet.

Evgeny Morozov et Ivan Illich

Dans un chapitre du *Mirage numérique* (Paris, Les Prairies ordinaires, 2015) intitulé « Hackers, makers, encore un effort si vous voulez faire la révolution ! », Evgeny Morozov évoque la « naïveté d'Illich et de ses fidèles » :

> Et puis Steve Jobs est arrivé. Le projet politique de Felsenstein – construire des ordinateurs qui mineraient les institutions et permettraient aux citoyens de partager des informations et de s'organiser – s'est transformé en projet esthétique, fondé sur l'autonomie et l'émancipation individuelles. Pour Jobs, qui voyait l'ordinateur comme un « vélo pour l'esprit », il n'était pas très important que les utilisateurs puissent le programmer ou examiner son contenu.

> Jobs a certes ses torts, mais il ne faut pas sous-estimer la naïveté d'Illich et de ses fidèles. Chercher le salut dans les seuls outils n'est pas une stratégie politique plus viable que celle qui consiste à s'attaquer aux maux du capitalisme en cultivant le goût de la population pour l'art et l'artisanat. [...] Au lieu de vouloir désinstitutionnaliser la société, les radicaux auraient mieux fait de défendre sa réinstitutionnaliation : en luttant pour des réformes politiques et juridiques susceptibles de garantir la transparence et la décentralisation du pouvoir qu'ils associaient à leur technologie préférée.

Et plus bas :

> Selon [le penseur libertaire] Bookchin, en effet, la notion d'outil convivial est dénuée de sens si l'on n'examine pas de près les structures politiques et sociales dans lesquelles les outils s'inscrivent.

C'est toujours les mêmes questions : doit-on attendre d'un outil – à supposer que l'informatique soit un simple outil – qu'il résolve les problèmes qu'il a engendrés ? Quelques lignes de code suffisent-elles à changer le monde ?

Et Linux ?

Linux est un noyau de système d'exploitation – on appelle noyau la partie centrale du système d'exploitation (je reviendrai là-dessus). Sa première version a été publiée en septembre 1991 par le finlandais Linus Torvalds, d'abord sous une licence qui en interdisait la commercialisation – il n'était donc pas libre au sens où l'entendait la Free Software Foundation. Au mois de février de l'année suivante, quand il publie une nouvelle version de Linux, Torvalds adopte la licence publique générale GNU.

On se souvient que Stallman a lancé le Projet GNU avec l'ambition de mettre au point un système d'exploitation libre. Au début des années 1990, tous ses composants sont prêts, sauf le noyau. Un rapprochement entre le projet GNU et Linux s'opère de fait.

En 1992 paraît Yggdrasil, première distribution mêlant noyau Linux et outils GNU – on appelle distribution un ensemble de logiciels comprenant un système d'exploitation, des applications ainsi qu'un programme permettant d'installer tout cela.

Il faudrait en toute rigueur parler de GNU/Linux la plupart des fois où l'on parle de Linux. Il y a des cas de Linux sans GNU et de GNU sans Linux. Le choix de telle ou telle dénomination est source de débats dans la communauté du libre. Une page Wikipedia fait le point sur cette question.

Avant de poursuivre sur Linux, et d'expliquer ce qu'est un noyau de système d'exploitation – ce qui donnera l'occasion de parler du système Unix, dont Linux est un dérivé, et d'évoquer quelques héros et pionniers de l'informatique –, j'aimerais aborder la question du matériel informatique. Puisque, nous l'avons vu, le système d'exploitation sert d'interface entre logiciel et matériel, voyons en quoi consiste le matériel.

Contact : compilation@pauljimenes.fr