

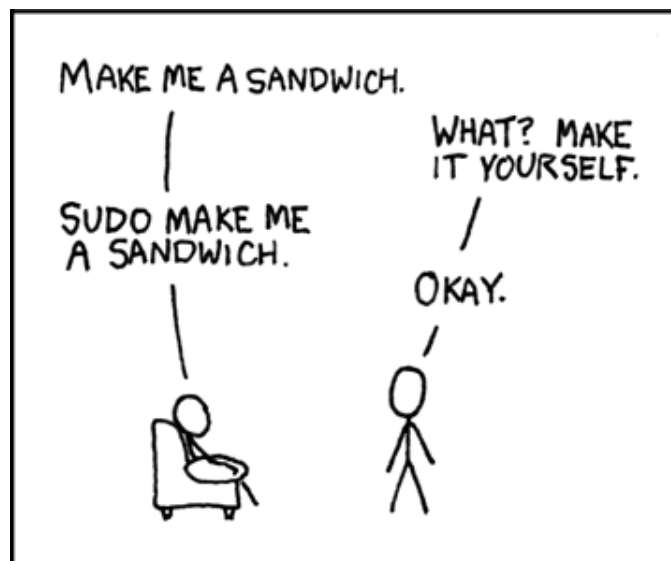
---

## GE28 – Fiche de lecture

# RICHARD STALLMAN ET LA RÉVOLUTION DU LOGICIEL LIBRE

---

CHABOT Simon



8 juin 2010

*La route est longue, mais la voie est sûre...*

# Table des matières

<b>1</b>	<b>Une soif de liberté</b>	<b>4</b>
1.1	Richard Matthew Stallman <i>alias</i> RMS . . . . .	4
1.1.1	Une frustration . . . . .	4
1.1.2	La naissance de GNU . . . . .	5
1.1.3	RMS aujourd'hui . . . . .	6
1.2	Le mouvement libre . . . . .	6
1.2.1	Définition . . . . .	6
1.2.2	Les avantages du mouvement libre . . . . .	7
1.2.3	Les inconvénients du mouvement libre . . . . .	8
<b>2</b>	<b>1984, la révolution</b>	<b>12</b>
2.1	Une révolution technologique . . . . .	12
2.1.1	L'internet . . . . .	12
2.1.2	La révolution technologique par l'innovation . . . . .	13
2.2	Une révolution sociale . . . . .	15
2.2.1	« Freedom or Power »[2] . . . . .	15
2.2.2	Égalité . . . . .	15
2.2.3	Fraternité . . . . .	16

# Introduction

« Chaque génération a son philosophe, un écrivain ou un artiste qui saisit et incarne l'imaginaire du moment. Il arrive que ces philosophes soient reconnus de leur vivant, mais le plus souvent il faut que la patine du temps fasse son effet. Que cette reconnaissance soit immédiate ou différée, une époque est marquée par les gens qui expriment leurs idéaux, que ce soit dans les murmures d'un poème, ou dans le grondement d'un mouvement politique. Notre génération a un philosophe. Ce n'est ni un artiste ni un écrivain. C'est un informaticien. » — Lawrence Lessig<sup>1</sup>

Cet informaticien dont parle Lawrence Lessig n'est autre que Richard Matthew Stallman. Homme de sciences et d'esprit, Richard Stallman, est à l'origine d'un nouveau mouvement culturel. Le mouvement *libre*. On parle même d'une *révolution*. Nous allons essayer de montrer dans ce document en quoi consiste ce mouvement et pourquoi est-il qualifiable de *révolution*. Ce document est basé sur la biographie autorisée de Richard Stallman, intitulée « RICHARD STALLMAN ET LA RÉVOLUTION DU LOGICIEL LIBRE »[1], espérant ainsi vous donner envie de lire ce livre, placé sous licence libre. Ce livre écrit par le journaliste Sam Williams, "a été *hacké*" par Richard Stallman lui-même, proposant ainsi de nombreux commentaires et critiques de la part du personnage principal. De ce fait, ce livre est vivant et plein de petites anecdotes sur ce personnage si particulier.

Dans un premier temps, nous nous intéresserons aux motivations de Richard Stallman vis-à-vis de ce mouvement. Nous expliciterons ses caractéristiques, essaierons de démontrer ses avantages et ses inconvénients. Nous regarderons donc les caractéristiques sur un plan économique, mais aussi sur un plan pratique. Pourquoi devrait-on utiliser des logiciels libres ? Pourquoi peu de personnes s'en servent réellement, etc. Puis, dans un second, nous montrons pourquoi ce mouvement est révolutionnaire. Révolutionnaire aussi bien sur le plan technologique que sur le plan social.

---

1. Professeur de droit de renommée internationale, spécialisé dans le droit constitutionnel et dans le droit de la propriété intellectuelle. Il est un défenseur de la liberté sur Internet. Il reçoit en 2002 le prix pour le développement du logiciel libre de la Free Software Foundation.

# Chapitre 1

## Une soif de liberté

### 1.1 Richard Matthew Stallman *alias* RMS

Richard Matthew Stallman, aussi connu sous les initiales RMS, est un américain né à Manhattan le 16 mars 1953. Stallman a toujours été un élève surdoué, ce qui lui causa quelques frustrations dans son enfance. Il a toujours refusé l'autorité, notamment celle de sa mère ou de ses professeurs. « *La majeure partie de ce que j'avais lu à propos de la démocratie et des libertés individuelles, je l'appliquais à moi-même. Car je ne voyais aucune raison d'en exclure les enfants.* »[1] dit-il.

Doué en sciences, il fait ses études à l'université de Harvard où il obtient d'excellents résultats. Il obtient son premier emploi au laboratoire d'intelligence artificielle du MIT<sup>1</sup>. Cette offre d'emploi se fit par hasard. En effet, Stallman avait vaguement entendu parlé de ce laboratoire et venait simplement demandé s'il y avait quelques manuels supplémentaires pour les étudiants curieux. Mais à défaut de repartir avec un manuel, il repartit avec un emploi.

Stallman alternait donc entre ses études à Harvard et cet emploi. Il consacrait sa semaine à Harvard et passait son week-end — vendredi soir compris — à coder au AI LAB<sup>2</sup>. Il prit très vite goût à cette vie de *hacker*<sup>3</sup>. Le vendredi soir, il sortait avec ses collègues dans *China town* pour manger, et puis ils rentraient ensuite pour coder. C'est là-bas qu'il apprit l'éthique *hacker*, en autre le partage des connaissances, le perfectionnisme et le refus de l'autorité.

#### 1.1.1 Une frustration

Les *hackers* du AI LAB prenaient plaisir à optimiser leurs systèmes et à y ajouter de nouvelles fonctionnalités. Le laboratoire disposait d'une imprimante un peu capricieuse. Celle-ci tombait souvent en panne. Stallman a donc développé un petit programme qui indiquait aux membres du laboratoire quand l'imprimante était en panne. La solution n'était pas parfaite, car elle n'empêchait pas l'imprimante de tomber en panne, mais elle évitait les déplacements inutiles. Un jour, ils reçurent, enthousiastes, une imprimante offerte par Xerox Corporation. Celle-ci fonctionnait

---

1. Massachusetts Institute of Technology

2. Artificial Intelligence Laboratory

3. Dans le document présent, *hacker* fait référence à un programmeur astucieux et débrouillard, à une personne doué d'une « *ingéniosité espiègle* » — Playful cleverness, en anglais — dirait Stallman et non à une personne qui aime casser des systèmes informatique; image véhiculée par les médias.

bien, elle était rapide et l'impression était de qualité. Mais à l'usage, il est apparu que le papier bourrait fréquemment. Désireux de mettre en place la même astuce qu'avec l'imprimante précédente, Stallman avait besoin d'un accès au code source du programme pilotant l'imprimante. Le code source est la *recette* du programme. La *recette* est compréhensible par l'homme puis une fois traduite en langage machine, elle est compréhensible par l'ordinateur. Xerox avait fourni la forme compilée — traduite — du programme mais pas la recette. Stallman était donc dans l'impossibilité de mettre en place son correctif. Il connaissait quelqu'un qui disposait de ce code source. Naturellement, comme les *hackers* l'ont toujours fait quand ils leur manquaient un programme, il alla le lui demandé. Mais celui-ci refusa. Il avait passé un accord de non divulgation. Stallman se senti trahi. Comprenant ce qui allait se passer, et que ce phénomène allait se répandre il décida de créer le projet GNU<sup>4</sup>.

### 1.1.2 La naissance de GNU

*« Nous, hackers, cherchons toujours un nom drôle ou coquin pour un logiciel, parce que cela fait partie intégrante du plaisir de l'écrire. Nous avons cette tradition des acronymes récursifs pour signaler que tel programme que vous écrivez est similaire à un autre déjà existant. Je cherchais un acronyme récursif pour Something is not Unix. J'ai essayé les 26 lettres avant de découvrir qu'aucune combinaison ne composait un mot. J'ai décidé de recourir à la contraction du is qui me permettrait d'obtenir un acronyme de trois lettres Something's not Unix. J'ai réessayé des lettres et je suis tombé sur GNU. C'est tout. » — RMS*

GNU est un projet lancé à l'initiative de Richard Stallman. Ce projet avait pour but de faire un système d'exploitation libre à 100%. Ce projet regroupe aujourd'hui plusieurs centaines de logiciels libres disponibles pour tout un chacun. Au début, Stallman semblait être un fou, un idéaliste. Son projet semblait si infaisable que peu de monde l'on suivit, au début. Ainsi, il refusa d'utiliser tout logiciel qui n'était pas libre, tout logiciel qui allait à l'encontre de ses convictions. Il s'engagea donc à bâtir un système d'exploitation libre *« ou de mourir à la tâche ... de vieillesse bien entendu »* plaisante-t-il. Il quitta, en 1985, le MIT pour se consacrer au projet. De temps en temps, l'école prenait les logiciels développés et les publiait sous licence propriétaire. Si l'école avait pris son projet pour le rendre propriétaire, l'effort aurait été vain. En démissionnant, personne ne pouvait le lui prendre. Le directeur du laboratoire l'autorisa cependant de continuer à utiliser les ordinateurs du laboratoire.

Aujourd'hui, le projet GNU est achevé. Il est possible d'utiliser un système d'exploitation libre à 100%. Il a été possible d'aller jusqu'au bout car des personnes comme Stallman voulaient un système qui soit libre à 100% et pas seulement à 99%. Sans quoi, jamais un tel projet aurait pu être achevé. Le projet GNU est toujours en évolution. Il insiste sur le fait qu'il y a encore de nombreuses choses à améliorer ou à créer. Ainsi, il met l'accent sur les domaines où les logiciels privateurs<sup>5</sup> sont

---

4. GNU's Not Unix

5. Dans tout ce document, l'expression "logiciel privateur" sera privilégiée face à "logiciel propriétaire", car cette expression met plus en valeur l'aspect privatif du logiciel *non-libre* que l'expression "propriétaire". Privatif, dans le sens où les utilisateurs sont privés de leurs libertés. De plus, l'expression "logiciel propriétaire" laisse à croire qu'il faudrait renoncer à ses droits d'auteurs pour créer un logiciel libre. Ce qui est faux ; la GPL est basée sur le copyright.

supérieurs d'un point de vu technique<sup>6</sup>. Ces domaines sont notamment la gestion de l'Adobe flash<sup>7</sup>, l'édition vidéo ou encore la gestion du format PDF<sup>8</sup>.

### 1.1.3 RMS aujourd'hui

Aujourd'hui, alors que le projet GNU est achevé, Stallman ne reste pas inactif. Il lui est quasiment impossible de programmer aujourd'hui — à cause de ses mains qui lui font mal — il parcourt donc le monde où il donne des conférences sur le logiciel libre et ses vertus. Stallman est vu comme comme un évangéliste du logiciel libre. Il joue notamment là dessus lorsque pendant ses conférences, il se déguise en *Saint IGNUcius*, bénissant les ordinateurs utilisant des logiciels libres. Son but aujourd'hui est de libérer les utilisateurs utilisant des systèmes injustes *sic*.

## 1.2 Le mouvement libre

### 1.2.1 Définition

Lorsque Stallman a entrepris la création de GNU, il a rédigé la GENERAL PUBLIC LICENCE (GPL). Cette licence a été écrite pour être utilisée sur les programmes du projet GNU. Elle apparait comme l'un des meilleurs *hacks* de Stallman. En effet, la GPL favorise le partage de connaissances tout en trouvant sa force et sa résistance dans le copyright. Lorsque l'auteur d'une œuvre soumet son travail à la GPL, il ne se détache pas des ses droits d'auteur ; il dit simplement « *je vous autorise à travailler sur mon œuvre, à l'étudier, la partager et vous l'approprier tant que vous autorisez les autres personnes à jouir de ces mêmes droits.* » On retrouve la notion de *Copyleft*. Le Copyleft est un jeu de mot avec Copyright. Le Copyleft est la possibilité offerte par l'auteur d'une œuvre, de l'étudier, de la modifier et de la partager, tant que ces possibilités sont conservées.

Ainsi, la GPL définit quatre libertés fondamentales :

Liberté 0 : La liberté d'exécuter le logiciel, pour n'importe quel usage ;

Liberté 1 : La liberté d'étudier le fonctionnement d'un programme et de l'adapter à ses besoins, ce qui passe par l'accès aux codes sources ;

Liberté 2 : La liberté de redistribuer des copies ;

Liberté 3 : La liberté de rendre publiques des versions modifiées pour en faire bénéficier la communauté.

Un logiciel qui ne respecte pas l'une de ces quatre libertés n'est pas libre. On voit donc que la *liberté* d'un logiciel n'est pas une question de licence ou bien de prix. Un logiciel libre peut être vendu, mais le vendeur ne peut pas interdire à l'acheteur de le distribuer gratuitement. De ce fait, les logiciels libres sont très souvent gratuits.

---

6. D'un point de vu technique, les logiciels libres ne sont pas toujours supérieurs. « *Parfois oui, parfois non. Ce qui importe, c'est que le logiciel libre soit éthiquement supérieur.* » — RMS

7. Gestion de contenus animés sur internet.

8. <http://www.fsf.org/campaigns/priority-projects/> : liste des projets prioritaires

## 1.2.2 Les avantages du mouvement libre

Par essence, le logiciel libre favorise le partage de connaissances. Il permet à tout le monde de comprendre s'il le souhaite le fonctionnement dans le moindre détail de tel ou tel logiciel. Personne ne devrait être autorisé à restreindre l'accès à la connaissance. Pourtant, certaines personnes produisant du logiciel propriétaires se le permettent. C'est en ce sens que les logiciels propriétaires sont injustes. Ils prennent leurs utilisateurs en otages, les obligeant à se plier aux volontés de l'auteur — mises à jour obligatoires, bugs pas toujours corrigés.

### Un avantage économique

Les logiciels libres, étant souvent gratuits, permettent une meilleure gestion du budget. Les entreprises peuvent, si elles le souhaitent faire des dons aux auteurs afin de les soutenir ou bien les payer pour qu'ils développent les fonctionnalités dont elles ont besoin, ce qui n'est pas possible avec du logiciel propriétaire, et ainsi avoir un logiciel qui répond au mieux à leurs besoins. Pour installer un logiciel propriétaire, il faut très souvent acheter une licence par ordinateur pour l'installer en toute légalité. Pour installer un logiciel libre, il suffit de récupérer le logiciel voulu et de l'installer tout naturellement. Le logiciel propriétaire, du fait de son prix souvent élevé est sujet à de nombreux téléchargements illégaux<sup>9</sup>. Les entreprises produisant ce logiciel chiffrent alors leur *perte* financière sur le nombre fois où le logiciel a été téléchargé. Ces chiffres sont erronés car les entreprises ne sont perdantes que dans la mesure où les utilisateurs auraient été prêts à payer pour utiliser le logiciel ; ce qui n'est pas assuré. Le logiciel propriétaire favorise donc en un sens le téléchargement illégal.

### Une source d'indépendance

Le logiciel libre est aussi source d'indépendance. Les utilisateurs ne sont plus contraints à un logiciel. Un logiciel libre, du fait qu'il puisse être modifié et distribué, dispose souvent de nombreux équivalents réalisant une tâche identique mais de façon différente. Il s'agit de temps en temps d'un même logiciel ayant été divisé en deux logiciels différents et suivant un développement différent. On parle de *fork*. Ainsi, on se retrouve avec deux logiciels qui font des tâches similaires avec des fonctionnalités différentes ou bien une façon différente de fonctionner. L'utilisateur est alors libre d'essayer différents logiciels et de garder celui qui lui convient le mieux. Le logiciel propriétaire, du fait qu'il ne partage pas son code source empêche ce système de *fork* et offre donc un panel de choix plus restreint. On peut prendre par exemple Audacious, qui est un lecteur audio, c'est un *fork* de Beep Media Player qui est lui même un *fork* de XMMS. On se retrouve ainsi un choix assez conséquent de logiciels. De plus, les logiciels libres utilisent le plus souvent des formats standards. Il est donc possible de changer de logiciel alors que des données ont déjà été générées. L'utilisateur est donc indépendant du logiciel.

---

9. *Piratage* est un mot inapproprié. En effet, le piratage est un acte qui consiste à piller un navire, détruire une richesse. Ce qui n'a absolument rien à voir avec le contexte. Il s'agit d'un terme abusif utilisé par les entreprises produisant du logiciel propriétaire (ou des la musique ou des films) pour inculquer au grand public que le partage est nocif et mauvais.

## Une sécurité accrue

De plus, les logiciels libres sont très sécurisés. La plupart des logiciels privés basent leur sécurité sur le secret. En empêchant les personnes de comprendre comment fonctionne leur logiciel, ils espèrent obtenir de la sécurité. Le logiciel libre raisonne différemment. Étant donné que le code est ouvert, il est impossible de baser la sécurité sur du secret. La sécurité est donc dans le code source lui-même. D'une part, parce qu'un logiciel libre est en constante évolution, dès qu'une faille est découverte, les développeurs la corrigent immédiatement — combien de failles de sécurité sous Internet Explorer ont vu des mois avant d'être corrigées ?

D'une autre part, le logiciel libre n'est pas soumis aux aléas de l'économie. Il n'y a donc pas de pression économique obligeant un logiciel à sortir à une date  $t$  précise. Tout logiciel a un cycle de développement. On distingue couramment trois versions. La version *alpha* qui est le prototype, comportant de nombreux bugs qui sont corrigés au maximum. S'en suit la version *béta*, qui est proposée à l'essai. Les *béta-testeurs*, comme on les appelle, font ensuite des retours aux développeurs sur leurs impressions concernant le logiciel et/ou les bugs à corriger. Le logiciel est donc adapté selon les remarques. Et pour finir, sort la version *gold*, la version stable. Afin de gagner en rapidité, par exemple, Microsoft n'hésite pas à commercialiser directement les versions *béta* de ses logiciels, économisant ainsi le travail de débogage[7]. Nous pouvons aussi évoquer la loi « *Linus* » qui est la suivante :

*« Étant donné un ensemble de bêta-testeurs et de co-développeurs suffisamment grand, chaque problème sera rapidement isolé, et sa solution semblera évidente à quelqu'un. »*

Ainsi, en accord avec cette loi, les logiciels libres sont plus aptes à être sécurisés que les logiciels privés car les développeurs de logiciels libres sont des milliers dans le monde entier et n'importe quelle entreprise ne peut rivaliser sur ce point là.

De part ses nombreux avantages — sécurité, indépendance, budget, choix, etc. — de nombreux services publics ont décidé d'utiliser des technologies libres. On peut notamment citer l'Assemblée Nationale, la gendarmerie, des écoles et universités (comme l'INSA de Lyon). Le point le plus important, c'est que les logiciels libres libèrent les utilisateurs. C'est l'utilisateur qui contrôle le logiciel et non l'inverse.

### 1.2.3 Les inconvénients du mouvement libre

#### Le syndrome du “Aïe, *Linux* c'est pas comme Windows”

Nous venons de voir que les logiciels libres offraient des avantages non négligeables. Cependant, le passage d'une solution privée à une solution libre peut être difficile pour un particulier ou une entreprise. Difficile non pas parce que les logiciels libres sont plus durs à utiliser que leurs équivalents privés mais parce qu'ils sont différents. Comme ils sont différents, il est nécessaire d'avoir un temps d'adaptation, plus ou moins long selon le logiciel. Cela peut repousser plus d'un utilisateur. C'est là que réside le véritable problème. De plus, bien souvent, une image complètement erronée du logiciel libre est véhiculée par les médias. On peut par exemple citer M<sup>me</sup> Albanel, à l'époque ministre de la Culture et de la Communication, qui fait croire qu'il est possible de sécuriser son ordinateur, avec le logiciel libre OpenOffice.org qui fait office de pare-feu :



« Par exemple, nous au ministère, nous avons un logiciel libre, qui s'appelle OpenOffice et il y a effectivement un logiciel de sécurisation qui empêche en effet le ministère à la Culture d'avoir accès, bien sûr, et les éditeurs de logiciels libres fournissent des pare-feux, et fournissent même des pare-feux gratuits. »

Soit l'auditoire de M<sup>me</sup> Albanel sait qu'OpenOffice.org est une suite bureautique et comprend qu'elle ne sait pas tellement se qu'elle dit au sujet de ce logiciel libre, soit il ne le sait pas, et va retenir des idées fausses sur le logiciel libre. C'est un véritable problème. Supposons qu'un l'utilisateur *lambda* écoute un discours, à peu près correct, sur le logiciel libre. Il va retenir du discours qu'un logiciel libre équivalent à son logiciel privateur, est un logiciel identique en tout point à celui qu'il utilise mais qui est en plus gratuit et sans faille (ou sans virus, dans le cas du système d'exploitation). L'utilisateur *lambda* va vouloir essayer un logiciel libre — le système d'exploitation GNU/Linux, par exemple — en pensant qu'il sera meilleur que son logiciel actuel mais avec un fonctionnement et une interface identique. Le fourvoisement est ici, un logiciel libre ne peut pas être meilleur qu'un logiciel privateur et identique à la fois. Pour être meilleur, il faut nécessairement un fonctionnement différent et l'utilisateur se trouve alors perdu parce qu'il n'a pas ce qu'il attendait. Il est habitué à effectuer l'action *x* pour accomplir la tâche *y*. Lorsqu'on lui dit, maintenant pour faire *y*, tu devras faire *z*, il est perdu et considère chaque différence comme un échec. Il retrouve alors son logiciel privateur ; faute d'investissement. Les entreprises souhaitant “faire le saut” devront proposer à leurs employés des formations afin de les guider<sup>10</sup>. Cela peut donc se traduire par une baisse de rendement pendant quelques temps. On pourrait aussi souligner le fait que ces formations ont un coup. Mais, le budget nécessaire à ces formations peut être trouver dans les économies faites sur les licences qui ne sont plus à acheter. De plus, une fois ces formations payées, ces économies se traduiront en bénéfices car les formations ne seront plus à refaire, et les licences ne seront plus à racheter, alors qu'elles auraient dû l'être avec des logiciels privateurs.

## Le prix

« C'est gratuit, alors c'est nécessairement moins bien. »

Un logiciel libre est différent d'un logiciel non-libre. Il faut donc, comme nous l'avons dit, un temps d'adaptation. C'est cette période d'adaptation qui est critique. Critique dans le sens où c'est à ce moment que l'utilisateur va être *converti* ou non. On remarque souvent que les personnes qui critiquent GNU/LINUX sont des personnes qui ne connaissent pas du tout le système ou bien alors des personnes qui l'ont *essayé* très rapidement. Je mets “essayé” en italique, car bien souvent les personnes qui critiquent GNU/LINUX se sont arrêtés aux premières différences rencontrées. Mais alors, pourquoi ne rencontrons nous que très peu de personnes qui ont essayé Mac, et qui sont retournés sur Windows — parce que Mac est très différent de Windows — ? Très peu en vérité. Pourquoi ? Parce que Mac est payant. Les acheteurs en veulent pour leur argent, ils vont s'investir, s'acheter un livre sur l'utilisation d'un Mac au quotidien. Ils vont même abandonner Word pour Pages... C'est à dire qu'ils

---

10. Il avait été proposé aux députés de l'Assemblée Nationale des formations pour apprendre à utiliser Ubuntu et OpenOffice.org. Plusieurs interviews à ce sujet ont montré que peu d'entre eux en ont eu besoin.

vont accepter de repartir à zéro et se livrer à l'expérience. La transition se fera, parce qu'ils se seront investis. Ils ont payé pour cela.

Avec GNU/LINUX, il est hors de question d'abandonner Word pour OpenOffice.org, parce que c'est différent. (La question ne se posait pas pour Pages). De plus, *avoir un Mac, ça fait bien*. Apple est une entreprise à la mode, ses produits sont vendus à un prix élevés. L'effet Veblen met en évidence le fait que les consommateurs seront plus intéressés par un produit cher et laisseront en marge un produit dont le prix est plus bas. En conséquence, les logiciels libres, étant communément gratuits, suscitent moins d'intérêts de la part des utilisateurs, que les logiciels propriétaires, payant. Le prix faible des logiciels libres, qui semble à première vue un avantage indéniable peut être transformé en inconvénient, à cause de notre société de consommation.

La figure 1.1 page 10 nous permet d'illustrer nos propos quand à l'investissement personnel nécessaire. Pour réaliser une tâche simple, il va être plus économique — au sens large, c'est à dire le temps nécessaire à la mise en place d'une solution, le prix, etc. — d'utiliser Windows, mais dès que la tâche se complexifie, GNU/Linux sera plus apte à résoudre le problème ; gain de temps et/ou gain économique. C'est en ce sens que GNU/Linux nécessite un temps d'adaptation et d'investissement<sup>11</sup>. Il faut réussir à traverser ce fossé pour trouver plaisir à utiliser ce système d'exploitation. Les utilisateurs qui ne se sont pas assez investis, ou bien qui sont partis de préjugés et se sont bloqués à la première différences sont les utilisateurs qui n'ont pas su *sortir de zone la grise du graphique*. Il en va de même pour Mac, à la différence que les utilisateurs, pour des raisons financières, se sont investis.

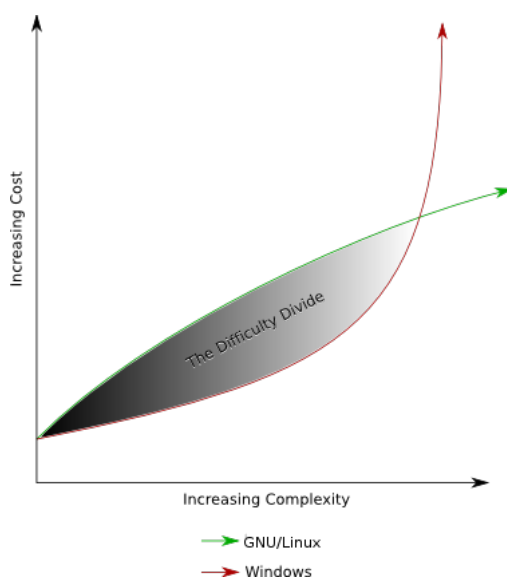


FIGURE 1.1 – Fossé de difficulté

## Un besoin pas toujours assouvi

« *Seul un fou échange sa liberté contre des fonctionnalités* » — RMS

Les logiciels libres sont souvent développés, au départ, pour répondre aux besoins de l'auteur lui-même. Par la suite, si le logiciel se révèle utile à d'autres personnes,

11. Tout changement d'habitude nécessite un temps d'adaptation et d'investissement

alors d'autres programmeurs rejoindront le projet et ce dernier sera de plus en plus connu. Le problème qui peut se poser est que les programmeurs qui rejoignent le projet, et qui sont des bénévoles, développent le logiciel pour leurs propres besoins — « *Par des programmeurs, pour des programmeurs* », entend on des fois — et non pas pour les besoins de l'entreprise *X* ou *Y*. Tant mieux si *X* ou *Y* peuvent trouver un intérêt à utiliser le logiciel, mais pour les développeurs, il ne s'agit pas d'un but en soit. De ce fait, les entreprises ne trouvent pas toujours les logiciels libres répondant exactement à leurs besoins. Alors qu'il y a fort à parier qu'un logiciel privateur sera disponible pour répondre à leurs besoins techniques, car un logiciel privateur est développé pour répondre exactement aux besoins de l'entreprise *X* ou *Y*. Encore une fois, c'est un système différent. Le libre est développé en espérant être utile<sup>12</sup>, alors que le privateur est développé pour être utilisé massivement. Il se trouve aussi que l'informatique libre a un petit peu de retard dans quelques domaines (comme précisé dans la partie 1.1.2 p. 6).

### “Trop de choix tue le choix”

Je finirai par ce dernier point. Nous avons vu dans la partie 1.2.2, qu'un des avantages du libre était le large panel de choix en matière de logiciel qui pouvait s'offrir aux utilisateurs. L'utilisateur *lambda* qui a toujours utilisé du logiciel privateur n'est pas habitué à tant de choix. Pour lui, pour faire une tâche il y a un logiciel. Pour l'utilisateur *gamma* qui a toujours utilisé du logiciel libre est habitué à avoir du choix. Pour lui, pour une tâche, il y a plein de logiciels ; il suffit de trouver celui qui correspond le mieux à ses besoins. L'utilisateur *lambda* qui va vouloir “faire le saut” va devoir se heurter à ce choix. Il va devoir passer du temps à chercher quel logiciel lui correspond le mieux. Souvent, il n'aura pas envie de faire cette recherche et préférera le logiciel qu'on lui tend sous le nez, le logiciel privateur. Cependant, admettons que l'utilisateur *lambda* ait choisi le logiciel qui lui convienne. Une autre difficulté va surgir. Enfin, une autre différence, devrais-je dire car il ne s'agit pas d'une difficulté. Lorsqu'il rencontrait un problème avec son logiciel privateur, l'utilisateur *lambda* appelait une *hot-line* et était en droit d'exiger une réponse à son problème ; il a payé pour avoir un service après vente. De l'autre côté, l'utilisateur *gamma*, lui va aller sur internet, voir sur des forums et souvent lire la documentation du logiciel<sup>13</sup> en question. Mais il n'est pas en droit d'exiger une réponse — une réponse est néanmoins apportée dans la majorité des cas. Ainsi, l'utilisateur *lambda* sera quelque peu déstabilisé au début, mais monnayant un petit effort parviendra à faire le saut.

---

12. Dans les en-têtes des codes sources des logiciels soumis à la GPL on retrouve ceci : « *This program is distributed in the hope that it will be useful, but without any warranty [...]* »

13. « *RTFM* » : “Read The Fuc...ing Manual”, peut on voir sur les forums, en réponse aux utilisateurs n'ayant pas pris la peine de lire les wikis ou bien les pages de manuels.

# Chapitre 2

## 1984, la révolution

On parle de révolution technologique lorsqu'une évolution sans précédent voit le jour. On parle de révolution sociale lorsqu'il y a une volonté de changement. Le libre entre précisément dans ces deux cas. Ainsi, nous montrerons dans un premier temps pourquoi le libre est une révolution technologique, en se basant notamment sur le fait qu'il s'agisse d'une technologie innovante. Dans un second, nous montrerons en quoi s'agit il d'une révolution sociale en montrant la volonté de changement, qui est intrinsèque au libre.

### 2.1 Une révolution technologique

#### 2.1.1 L'internet

Lorsque Stallman débute le projet GNU, il ne sait pas encore dans quel combat il s'engage. Il ne sait même pas s'il va réussir. Il n'est pas possible à l'époque d'imaginer ce qu'allait devenir les logiciels libres et le projet. Aujourd'hui, on ne peut plus se passer de logiciels libres. On en utilise tous, et tous les jours. En effet, internet vit grâce au logiciel libre — et inversement. Ainsi, en faisant la moindre recherche sur Google, par exemple, vous faites fonctionner sur les ordinateurs de Google des logiciels libres. Quand vous allez sur un site internet, il y a fort à parier que celui-ci utilise, pour vous envoyer les informations nécessaires à afficher la page web, un logiciel nommé Apache. Il s'agit encore d'un logiciel libre. Vous utilisez une Livebox ou une Neufbox, vous utilisez du logiciel libre. L'un ne va pas s'en l'autre. Si internet a grandi et s'il est devenu ce qu'il est aujourd'hui c'est parce que les technologies utilisées étaient et sont libres. Je ne dis pas celui gratuitement ; ce sont les faits. Internet existe parce que les technologies utilisées sont connues de tous et que tout le monde utilise les mêmes. Si un groupe de personnes développe sa petite technologie dans son coin, et s'il ne souhaite pas publier son fonctionnement alors peu de gens vont s'en servir et elle restera limitée aux membres du groupe. Internet a besoin d'être interopérable pour fonctionner. Tous les usagés doivent être égaux devant Internet.

#### Les technologies privatrices et l'internet

Tous les usagés doivent être égaux devant Internet, venons nous de dire. Mais aujourd'hui, est-ce vraiment le cas ? Si l'on regarde le web tel qu'il était il y a quelques

années, on constate que c'était le cas. Tout le monde avait accès au même contenu, était en mesure d'aller sur les mêmes sites web. Même les personnes atteintes de déficience visuelle pouvaient accéder à l'information<sup>1</sup>.

Aujourd'hui ce n'est plus possible. Certaines entreprises, Adobe par exemple, ont mis au point une technologie, privative, de gestion de contenus animés sur le web<sup>2</sup>. Cette technologie est appréciée car elle permet de faire des petites animations avec un joli rendu. Adobe Flash est utilisée sur Youtube, par exemple. Il y a deux problèmes majeurs. Le premier, est que si l'entreprise — Adobe, dans notre exemple — refuse de libérer le code permettant de lire l'animation ou refuse de créer un lecteur pour chaque plateforme informatique, alors les utilisateurs de ces plateformes seront dans l'impossibilité d'accéder à l'information. Les développeurs de logiciels libres vont alors devoir essayer de deviner le fonctionnement de l'animation pour créer un lecteur<sup>3</sup>. On parle de *rétro ingénierie*. Adobe propose aux utilisateurs de GNU/Linux, un lecteur. Mais celui-ci n'est pas libre ; il en faut un libre. Tous les utilisateurs n'ont donc pas accès à la même information. Internet n'est plus une zone d'équité. Le second problème est pour les personnes atteintes de déficience visuelle. Autrefois, elles pouvaient lire les sites web, avec l'Adobe Flash, elles ne peuvent plus. Comment peuvent-elles alors lire l'information donnée sur de tels sites<sup>4</sup> ? Un web libre permet une accessibilité importante. Ne l'oublions pas.

## 2.1.2 La révolution technologique par l'innovation

« *Les logiciels libres tuent la concurrence. Ils donnent gratuitement leur travail !* » ou bien « *Les logiciels libres sans les logiciels propriétaires ne sont plus rien. Ils en ont besoin pour trouver les idées et les copier* » entend-on dire. Que nenni. Il a vrai, comme nous l'avons montré, les logiciels libres sont communément gratuits. Mais ils sont aussi source de concurrence et d'innovation.

### La coopération

Si une entreprise *A* produit du logiciel libre, elle va naturellement publier ses sources. Il est vrai, la concurrence, l'entreprise *B*, va pouvoir en profiter. Bâtir son produit sur le logiciel de *A* pour en faire un meilleur — sans devoir réinventer la roue. Elle va ensuite devoir, elle aussi, le publier (notion de Copyleft<sup>5</sup>). L'entreprise initiale, *A*, va à son tour pouvoir s'en servir, l'améliorer etc. Le cercle ne se brise pas et est, de plus, vertueux (voir table 2.1.2 p. 14). Les entreprises sont concurrentes et voulant chacune faire mieux que l'autres, sont sources d'innovations. Si maintenant, l'entreprise *A* n'avait pas voulu libérer son logiciel, l'entreprise *B* n'aurait

- 
1. Grâce à des logiciels permettant une lecture audio des contenus textuels.
  2. *Adobe Flash*
  3. C'est ce qui actuellement fait pour le flash, afin d'avoir un lecteur libre de flash
  4. Quelques exemples :
    - <http://parcs.disneylandparis.fr>
    - <http://www.arthuretlesminimoys.com/>
    - <http://www.gala-utc.net/>
    - ...

5. Le *Copyleft* est la possibilité donnée par l'auteur d'un travail soumis au droit d'auteur de copier, d'utiliser, d'étudier de modifier et/ou distribuer son œuvre dans la mesure où ces possibilités restent préservées.

pu l'améliorer et à son tour  $A$  n'aurait pas pu profiter de cette amélioration. On se serait alors retrouvé dans une configuration de monopole, où la concurrence entre  $A$  et  $B$  n'aurait pu être possible. Et ainsi l'innovation aurait été moindre. On peut par exemple citer le cas d'Orange qui a ouvert le code source des logiciels utilisés dans sa livebox. Les logiciels utilisés étaient majoritairement libres à l'origine. Orange devait donc les laisser libre ; même après modification. Tous les logiciels utilisés par les livebox sont libres<sup>6</sup>.

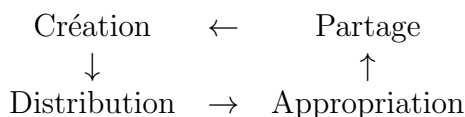


TABLE 2.1 – Cercle vertueux du libre

## Qui copie ?

Le second point qui dit que les logiciels libres font simplement une copie des logiciels propriétaires est aussi erroné. « *Si, ils copient ! Regardez, ils ont des fenêtres, des menus, des boutons “fermer”, “réduire” et “agrandir” !* ». Afin de montrer que les logiciels libres ne *copient* pas et sont source d'innovation, faisons une analogie avec le monde automobile. Toutes entreprises automobiles, aussi concurrentes qu'elles puissent l'être font des voitures avec les mêmes caractéristiques. Toutes les voitures ont un volant, un accélérateur, un frein, un moteur, des clignotants, ou encore des phares. Aucun constructeur automobile ne va aller mettre les boutons permettant de gérer les phares au milieu des boutons de la radio. Pourquoi ? Tout simplement, parce que ce ne serait pas pratique. Ils vont tous classer les boutons. Les boutons pour les phares et les clignotant à gauche du volant, les boutons pour les essuie-glaces à droite et ceux de la radio sur la radio. Ils respectent cela et se “copient tous dessus”, car c'est la norme communément admise. Libre à chaque constructeur de rajouter sur le volant un bouton pour le klaxon ou bien encore pour la radio. Mais, en général, chaque voiture à la même configuration.

Il en va de même pour les logiciels. Les menus, les fenêtres sont utilisés sur tous les systèmes car ils sont pratiques, non pas parce qu'ils ont été copiés<sup>7</sup>. Les entreprises produisant du logiciel propriétaires et les logiciels libres sont sources d'innovations. Les entreprises produisant du logiciels propriétaires, typiquement Apple et Microsoft, font de la *R&D*. Les logiciels libres eux, se basent sur les besoins et les retours des utilisateurs pour améliorer l'existant et combler les manques existants, produisant ainsi de l'innovation. On peut par exemple citer Compiz Fusion qui est le premier gestionnaire de bureau *3D* fonctionnel. Il est de plus libre. L'effet *3D* de Windows Vista est bien timide face à Compiz Fusion. Alors que Windows Vista permet de parcourir les fenêtres avec un effet d'empilement en trois dimensions, Compiz Fusion permet de faire tourner son bureau sur un cube, de faire tourner les fenêtres sur elles-mêmes offrant un effet de recto-verso, de faire tomber de la neige sur le bureau, faire du feu, faire disparaître les fenêtres en fumée, ou bien en morceaux ou en spirale, de faire des fenêtres molles... Et j'en passe. Lorsque Windows Vista est sortie, son effet

6. <http://www.livebox-floss.com/>

7. Dans le cas des fenêtres, boutons etc, on parle du paradigme *WIMP* pour “Windows, Icons, Menus and Pointing device”.

3D était nouveau pour le grand public et est apparu comme innovant ; alors qu'il n'en était rien. L'innovation venait du monde libre. Il en va de même dans d'autres domaines. On peut citer *VNC*<sup>8</sup>, qui permet de prendre le contrôle d'un ordinateur à distance, ou encore les *LiveCD* qui permettent de lancer un système d'exploitation (typiquement une distribution GNU/Linux), depuis un CD-rom directement dans la mémoire vive de l'ordinateur, sans altérer le disque dur.

## 2.2 Une révolution sociale

« *Je peux définir le logiciel libre en trois mots : Liberté, Égalité, Fraternité.* »

Stallman commence très souvent ses conférences en France par cette expression, intriguant ainsi son auditoire. Le logiciel libre est une révolution, car il s'agit d'un combat pour la liberté, l'égalité et la fraternité.

### 2.2.1 « Freedom or Power »[2]

Un petit groupe de personnes, principalement les personnes en tête des entreprises produisant du logiciel propriétaire, choisissent les règles que doivent adopter les utilisateurs. C'est à cette élite de personnes qu'il revient le droit de décider si oui ou non les utilisateurs pourront partager ou étudier ce logiciel. L'utilisateur n'a aucun contrôle. Il doit se plier aux règles imposées. Il faut certes des règles pour vivre en société. Mais celles-ci ne doivent pas être injustes et imposées par une élite (sans quoi on peut parler de dictature...). À l'heure d'aujourd'hui, la loi autorise les entreprises à priver leurs clients de libertés. L'État détient le monopole de la violence, et lui seul doit pouvoir priver, au nom de la loi, une personne de ses libertés (et encore faut-il que ces lois soient justes<sup>9</sup>). Les États qui utilisent des logiciels propriétaires sont soumis aux mêmes règles que l'utilisateur *lambda*. De ce fait, il ne peut être indépendant. « *Le Code Informatique fait loi* » a dit Lawrence Lessig. En effet, aujourd'hui la majeure partie de ce que nous faisons est limitée par les systèmes informatiques. Si le Code Informatique fait loi, qui doit décider de ce Code Informatique ? Un petit groupe d'élite ? ou la communauté toute entière ?

Aujourd'hui, les utilisateurs ont le droit de choisir l'entreprise qui va leur retirer leurs libertés (ou bien, opter pour une solution libre). On ne peut pas appeler cela de la *liberté*.

### 2.2.2 Égalité

Égalité, parce que toutes les personnes disposent des mêmes droits. Personne ne peut décider pour tout le monde. C'est à la communauté de choisir. Bien sûr, il y a des personnes pour prendre des décisions, mais celles-ci sont élues et changent fréquemment. Par exemple, pour le projet Debian GNU/Linux, on dénombre douze chefs de projet qui se sont succédés depuis 1993. Et combien même les décisions prises ne conviendraient pas, il est toujours possible à l'utilisateur mécontent de

---

8. Virtual Network Computing

9. Dans les années 1950, aux États Unis, il était illégal pour une personne Noire de s'asseoir à l'avant d'un bus. Personne ne dira que cette loi était juste.

modifier (ou faire modifier par un tiers) le logiciel en question ou encore de choisir un des nombreux logiciels équivalents. Tout le monde est sur le même pied d'égalité.

Égalité, parce qu'il ne peut y avoir de monopoles. La libre concurrence est possible, il ne peut pas y avoir d'abus de position, comme on l'a constaté avec Microsoft. Microsoft fournissait par défaut, au grand damne des développeurs web<sup>10</sup>, son navigateur web Internet Explorer. La Commission Européenne a donc conduit Microsoft en justice pour abus de position dominante. Internet Explorer était — et est encore — utilisé sur la plupart des ordinateurs. Google, dont la plupart de ses services sont basés sur l'Internet a besoin que ses clients utilisent un navigateur web fonctionnel et respectueux des standards établis. De ce fait, il développa Google Chrome, un navigateur basé sur le projet open source Chromium, pour ses utilisateurs. De plus, Google a ajouté sa voix au procès opposant Microsoft à la Commission Européenne. Microsoft propose donc, depuis le mois de mars 2010, le choix du navigateur web.

### 2.2.3 Fraternité

Les logiciels libres promeuvent le partage, la fraternité. Quoi de plus normal que de partager avec son voisin, dans la mesure où ce partage ne nous affecte pas ? Dans le monde matériel, le partage est plus difficile que dans le monde immatériel de l'information. Si je partage mon sandwich, alors j'en perds nécessairement une partie ; je suis directement affecté. En revanche, si je partage mon logiciel libre ou mon algorithme, je ne perds rien ; je ne suis pas directement affecté. Si la personne avec qui je le partage choisit de le modifier puis de le partager, je ne suis pas non plus directement affecté. Les bénéfices de la personne sont bien plus importants que les pertes que j'aurai pu subir. On ne peut pas appliquer les mêmes règles que nous utilisons toujours les jours dans le monde matériel au monde immatériel, les moyens de copie et de partage ne sont pas les mêmes. Cette fraternité permet de baser de nouvelles créations sur des créations plus anciennes. Le modèle privé ne permet pas cela, et oblige les développeurs à partir de zéro, ce qui est une perte de temps. Le logiciel libre favorise le partage de connaissances tel qu'il est enseigné tous les jours à l'école. Pourquoi devrait on arrêter de partager les connaissances une fois le cursus scolaire terminé ? Nous apprenons à partager nos connaissances, à travailler ensemble. La personne qui refuserait d'expliquer un théorème mathématique qu'il a compris à une autre personne serait bien mal vue. Comment doit on alors voir une personne qui refuse nous expliquer le fonctionnement d'un algorithme ?

---

10. Internet Explorer est un navigateur très peu respectueux des standards établis.



# Conclusion

Le monde du libre, né il y a maintenant plus de vingt ans, est en pleine expansion. De plus en plus d'utilisateurs font le saut. On compte 12 millions d'utilisateurs de la distribution GNU/Linux Ubuntu. Le problème c'est qu'une majeure partie des utilisateurs des distributions GNU/Linux ignore les enjeux en cours. Nombreuses seront les personnes qui vont installer une distribution GNU/Linux sans se soucier de leur liberté. Ils choisissent GNU/Linux (ou une autre solution libre) parce que c'est pratique, efficace et gratuit, non pas parce que c'est libre. Il est vrai que c'est pratique et efficace ; mais ça ne doit pas être le but premier. Il est satisfaisant de savoir que plus en plus d'utilisateurs choisissent le libre, mais décevant de savoir que peu d'entre eux s'intéressent à leur liberté. La liberté n'est pas quelque chose d'acquis ; on peut très vite la perdre. C'est pourquoi il faut préférer une solution libre à une solution privatrice et, pour pousser, open source. L'open source passe à côté du problème du libre, car elle n'assure pas toutes les libertés fondamentales aux utilisateurs.

Le libre traduit donc une volonté de changement. Une volonté de liberté et de partage. Le libre est en train d'être démocratisé. Les utilisateurs doivent être avertis des enjeux en cours.

*« Vous ne changez pas les choses en vous battant contre la réalité.  
Pour changer quelque chose, construisez un nouveau modèle qui rend  
l'ancien modèle obsolète » — R. Buckminster Fuller*

# Bibliographie

- [1] Richard Stallman, Sam Williams et Christophe Masutt *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*. Eyrolles / Framasoft, 2010
- [2] Richard Stallman *Freedom or Power*, essai publié sur [www.gnu.org](http://www.gnu.org)
- [3] Raphaël Metz *Logiciel libre : les pingouins se déchaînent*, Framablog, 2002
- [4] Sebsauvage *Linux n'est pas Windows !*, article publié sur [www.commentcamarche.net](http://www.commentcamarche.net), 2009
- [5] aKa *Comparaison graphique entre Windows et Linux*, article publié sur [www.framablog.org](http://www.framablog.org), 2008
- [6] Saunders *Open Source innovations*, article publié sur [www.linuxformat.co.uk](http://www.linuxformat.co.uk), 2009
- [7] Olivier Blondeau *Genèse et subversion du capitalisme informationnel*, article publié dans le journal LA PENSÉE, 1999