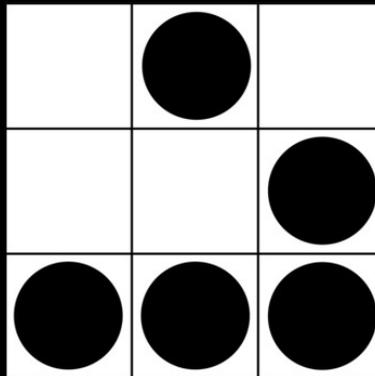


Pour la Liberté

Logiciels Libres, Société Libre

Textes de Richard M. Stallman

Version 9.3



Copyright © U.C.H Pour la Liberté

Permission vous est donnée de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU Free Documentation License, Version 1.1 ou ultérieure publiée par la Free Software Foundation.

Une copie de cette Licence est incluse dans la section « GNU Free Documentation License » de ce document.

☞ ☛ ☜ ☝ ☞ ☛ ☜ ☝ ☞ ☛ ☜ ☝

Pour la Liberté...

Logiciels Libres, Société Libre

Les principes de l'éthique ne peuvent pas changer. Ils restent les mêmes dans toutes les situations. Mais pour qu'ils s'appliquent quelle que soit la question ou situation, vous devez regarder les faits pour comparer les alternatives, et voir quelles en seront les conséquences. Un changement de technologie ne change jamais les principes de l'éthique, mais peut modifier ses conséquences sur les mêmes choix, qui peuvent avoir des résultats différents, comme cela s'est produit dans le domaine des lois sur le copyright. Nous sommes dans une situation où les changements de technologie ont affecté les facteurs éthiques qui pèsent sur les décisions à propos des lois sur le copyright et changent de politique appropriée pour la société.

Richard M. Stallman

Table des matières

Introduction.....	3
Livre I: Le projet GNU et le Logiciel Libre.....	7
Le projet GNU.....	9
Le Manifeste GNU.....	23
Définition d'un logiciel libre.....	31
Pourquoi les logiciels ne doivent pas avoir de propriétaire.....	35
Qu'y a-t-il dans un nom ?.....	39
Pourquoi « logiciel libre » est-il meilleur que « open source ».....	42
Écrire des logiciels libres si vous travaillez à l'université.....	47
Vendre des logiciels libres.....	49
Logiciels et manuels libres.....	52
The Free Software Song.....	54
Livre II: Copyright, Copyleft, et Brevets.....	56
Le droit de lire.....	58
Mauvaise interprétation du droit d'auteur.....	62
Les scientifiques doivent « mettre le droit d'auteur de côté ».....	70
Qu'est-ce que le copyleft ?.....	72
Copyleft : Idéalisme pragmatique.....	74
Livre III: À propos des licences.....	76
Pouvez-vous faire confiance à votre ordinateur ?.....	78
Pourquoi les logiciels devraient être libres.....	82
Droit d'auteur et mondialisation à l'âge des réseaux informatiques.....	94
Logiciel Libre : Liberté et coopération.....	110
Termes prêtant à confusion, que vous devriez éviter.....	134
Annexe A - Licence de documentation libre GNU.....	141
Annexe B - Texte original de la licence GNU FDL.....	147

Introduction

Chaque génération a son philosophe, un écrivain ou un artiste qui capte l'air du temps. Quelquefois, ces philosophes sont reconnus comme tels; souvent cela prend des générations avant qu'ils soient reconnus. Mais reconnus ou pas, un temps reste marqué par les gens qui parlent de leurs idéaux, dans le murmure d'un poème, ou l'explosion d'un mouvement politique.

Notre génération a un philosophe. Il n'est ni artiste, ni écrivain professionnel. Il est programmeur. Richard Stallman a commencé son travail dans les laboratoires du MIT, en tant que programmeur et architecte concevant les logiciels de système d'exploitation. Il a bâti sa carrière publique, comme programmeur et architecte en fondant un mouvement pour la liberté dans un monde de plus en plus défini par le « code ».

Le « code » est la technologie qui fait fonctionner les ordinateurs. Qu'il soit inscrit dans le logiciel ou gravé dans le matériel, c'est un ensemble d'instructions, d'abord écrit en mots, qui dirige la fonctionnalité des machines. Ces machines — ordinateurs — définissent et contrôlent de plus en plus notre vie. Elles déterminent la manière dont se connectent les téléphones, et ce qui passe à la télé. Elles décident si la vidéo peut-être diffusée à travers une liaison large bande vers un ordinateur. Elles contrôlent ce qu'un ordinateur rapporte à son constructeur. Ces machines nous gouvernent. Le code gouverne ces machines.

Quel contrôle devrions-nous avoir sur ce code ? Quel discernement? Quelle liberté devrait-il y avoir pour rivaliser avec le contrôle qu'il autorise ? Quel pouvoir ?

Ces questions ont été le défi de la vie de Stallman. Par ses travaux et ses paroles, il nous a poussé à voir l'importance de garder le code « libre ». Non pas libre au sens où les « pisseurs de code » ne doivent pas être rétribués, mais libre au sens où la construction et le contrôle du code doivent être accessibles à tous, et que tout un chacun a le droit d'en prendre le contrôle, comme il ou elle l'entend. Voici ce qu'est le « logiciel libre »; le « logiciel libre » est une réponse pour un monde fondé sur le code.

« Libre ». Stallman se plaint de l'ambiguïté dans ses propres termes. Il n'y a pas de quoi se plaindre. La perplexité force les gens à réfléchir, et ce mot « libre » fait que ça marche plutôt bien. Pour l'acception américaine moderne du terme, « logiciel libre » paraît utopique, impossible. Rien, pas même le déjeûner n'est libre (gratuit). Comment les « mots » les plus importants faisant fonctionner les machines les plus critiques qui gouvernent le monde pourraient-ils être « libres ». Comment une société saine pourrait-elle aspirer à un tel idéal ?

Cependant, la drôle de consonnance du mot « libre » vient de nous, pas du mot lui-même. « Libre » a différents sens, seuls l'un d'entre eux se réfère au « prix ». Un sens beaucoup plus fondamental de « libre » est « libre » que Stallman emploie dans l'expression « discours libre », ou peut-être mieux dans l'expression « travail libre ». Pas libre au sens de gratuit, mais libre au sens de contrôle par d'autres limité. Le logiciel libre est le contrôle transparent, et ouvert au changement, comme les lois sur la liberté, ou les lois d'une « société libre », elles sont libres quand elles rendent leur contrôle accessible et ouvert au changement. Le but du « Mouvement du logiciel libre » de Stallman est de faire le plus de code transparent possible, et sujet au changement en le rendant « libre ».

Le résultat de cette interprétation est un outil extraordinairement malin nommé « copyleft » mis en oeuvre au moyen d'une licence nommée GPL. Utilisant la puissance de la loi sur le droit d'auteur, le « logiciel libre » n'assure pas seulement l'ouverture, et la possibilité de changement, mais également que les autres logiciels qui prennent et utilisent des « logiciels libres » (et qui techniquement sont pris en compte en tant que « travail dérivé ») doivent être également libres eux-mêmes. Si vous utilisez et adaptez un programme libre, et diffusez alors cette version adaptée au public, la version diffusée doit être aussi libre que la version de laquelle elle a été adaptée. Elle le doit, ou le droit d'auteur serait violé.

Le « logiciel libre », comme les démocraties, a ses ennemis. Microsoft est entré en guerre contre la GPL, avertissant, à qui veut l'entendre, que la GPL est une licence « dangereuse ». Les dangers qu'il nomme cependant, sont grandement illusoire. D'autres s'opposent à la « coercition » dans l'insistance de la GPL pour que les versions modifiées soient également libres. Mais une condition n'est pas une coercition. S'il ne s'agit pas de coercition pour Microsoft de refuser la permission aux utilisateurs de distribuer les versions

modifiées de son produit Office sans le payer (vraisemblablement) des millions, alors, il ne s'agit pas de coercition quand la GPL insiste pour que les versions modifiées de logiciels libres le soient aussi.

Et puis, il y a ceux pour qui le message de Stallman est trop extrémiste. Mais il n'est pas extrémiste. En effet, il tombe sous le sens que le travail de Stallman est une simple traduction des libertés que notre tradition a modelé dans le monde d'avant le code. Le « logiciel libre » assurerait que le monde gouverné par le code soit aussi « libre » que notre tradition qui a bâti le monde d'avant le code.

Par exemple : une « société libre » est réglementée par la loi. Mais il y a des limites que toute société libre place dans cette réglementation par la loi : aucune société qui garderait ses lois secrètes ne pourrait être qualifiée de libre. Aucun gouvernement qui cacherait à ses citoyens ses règlements ne pourrait se maintenir dans notre tradition. La loi contrôle. Mais elle le fait avec justice, seulement quand elle est visible. Et la loi est visible quand ses termes sont connus et contrôlables par ceux auxquels elle s'applique, ou par les représentants de ceux auxquels elle s'applique (avocats, corps législatif).

Cette condition sur le droit s'étend au-delà du travail du corps législatif. Pensez à la pratique du droit dans les tribunaux américains. Les avocats sont engagés par leurs clients pour faire avancer les intérêts de leurs clients. Quelquefois, cet intérêt est réalisé au moyen d'un procès. Dans le cheminement de ce litige, les avocats rédigent des conclusions. Ces conclusions en retour affectent les avis rendus par les juges. Ces avis décident de qui gagne une affaire en particulier, ou si une certaine loi est conforme à la constitution.

Toute la matière dans ce processus est libre au sens où l'entend Stallman. Les conclusions juridiques sont ouvertes et libres d'accès pour tous. Les arguments sont transparents (ce qui est différent de dire qu'ils sont bons) et le raisonnement peut être repris sans la permission des avocats d'origine. Les avis qu'ils produisent peuvent être cités dans de futures conclusions. Ils peuvent être copiés et intégrés dans une autre conclusion ou avis. Le « code source » du droit américain est par conception, et par principe, ouvert et libre pour tout un chacun. Et les avocats le font — car c'est à cette aune que se mesure la créativité d'une grande conclusion, par la réutilisation de ce qui a été produit auparavant. Le source est libre; la créativité et une économie sont bâtis sur ce modèle.

Cette économie du code libre (et ici j'entends le code libre juridique) n'affame pas les avocats. Les firmes d'avocats ont suffisamment de motivation pour produire de bonnes conclusions même si ce qu'elles produisent peut être pris et copié par n'importe qui. L'avocat est un artisan; son produit est public. Cependant, l'artisanat n'est pas de la charité. Les avocats sont payés; le public ne demande pas un tel travail sans payer. Pourtant cette économie prospère, avec du neuf ajouté à du vieux.

Nous pourrions imaginer une pratique juridique qui soit différente — les conclusions et les argumentaires seraient gardés secrets; des décisions qui annonceraient un résultat mais pas le raisonnement. Des lois qui seraient conservées par la police mais qui ne seraient connues de personne d'autre. Une réglementation qui fonctionnerait sans expliquer les règles.

Nous pourrions nous représenter cette société, mais nous ne pourrions imaginer la qualifier de « libre ». Que les primes dans une telle société soient mieux ou plus efficacement allouées, une telle société ne pourrait être connue comme libre. Les idéaux de la liberté, de la vie dans une société libre, demande plus qu'une application efficace. Au contraire, l'ouverture et la transparence sont les contraintes à l'intérieur desquelles un système de droit se construit, et non grâce à des options ajoutées par commodité pour les dirigeants. La vie régentée par le code logiciel ne devrait pas être autrement.

Écrire du code n'est pas du litige. Il est meilleur, plus riche, plus productif. Le droit est un exemple évident qui montre que la créativité et les primes ne dépendent pas du contrôle total sur les produits créés. Tout comme le jazz, les romans, ou l'architecture, le droit se construit sur ce qui a été construit auparavant. Ces ajouts et changements sont ce qu'a toujours été la créativité. Et une société libre fait partie de celles qui assurent que ces ressources les plus importantes demeurent libres en ce sens.

Pour la première fois, ce livre recueille les écrits et les conférences de Richard Stallman dans un style qui rend leur subtilité et leur force évidentes. Les essais couvrent un large panorama, du droit d'auteur à l'histoire du mouvement du logiciel libre. Ils recèlent beaucoup d'arguments peu connus, et parmi ceux-ci, une explication particulièrement perspicace sur les changements de situation qui ont rendu suspect le droit d'auteur aux yeux du monde numérique. Ils serviront de ressource pour ceux qui cherchent à comprendre la

pensée de cet homme puissant — puissant dans ses idées, dans sa passion, et dans son intégrité, même s'il est impuissant dans toute autre voie. Ils en inspireront d'autres qui prendront ces idées, et en construiront de nouvelles.

Je ne connais pas bien Stallman. Je le connais suffisamment pour savoir qu'il est difficile à apprécier. Il est autoritaire, souvent impatient. Sa colère peut éclater à l'encontre de ses amis aussi facilement que de ses adversaires. Il est intransigeant et obstiné; patient dans ces deux travers.

Cependant, quand notre monde comprendra enfin la puissance et le danger du code — quand il comprendra finalement que le code, comme les lois ou comme les gouvernements, doivent être transparents pour être libres — alors il regardera retrospectivement ce programmeur intransigeant et obstiné et reconnaîtra la vision pour laquelle il s'est battu pour la rendre tangible : la vision d'un monde où la liberté et le savoir survivent au compilateur. Et nous nous rendrons compte qu'aucun homme, de par ses actes et ses paroles, n'en aura fait autant pour rendre possible la liberté que cette future société pourra avoir.

Nous n'avons pas encore gagné cette liberté. Nous pouvons très bien échouer à l'atteindre. Mais que nous y arrivions ou que nous y échouions, dans ces essais est brossé un tableau de ce que la liberté pourrait être. Et dans le monde qui a produit ces actes et ces paroles, il y a l'inspiration pour qui veut, comme Stallman, de se battre pour créer cette liberté.

Lawrence Lessig
Professeur de droit, École de droit de Stanford.

*Free Software Foundation, Inc.,
51 Franklin St, Fifth Floor, Boston, MA 02110, USA*

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

*Traduction : Cédric Corazza.
Révision : trad-gnu@april.org
Dernière mise à jour : 2009/02/04 21:28:36*

Traductions de cette page

[English](#) [en] [Español](#) [es] [فارسی](#) [fa] [Français](#) [fr] [Nederlands](#) [nl] [简体中文](#) [zh-cn]

Livre I: Le projet GNU et le Logiciel Libre



Cette notice est valable pour l'ensemble des articles, relativement:

Copyright © Richard Stallman,

ou Copyright © Free Software Foundation, Inc.,

La reproduction exacte et la distribution intégrale de chaque article est permise sur n'importe quel support d'archivage, pourvu que sa notice soit préservée.

Le projet GNU

par Richard Stallman

publié à l'origine dans le livre « *Open Sources* »

La première communauté qui partageait le logiciel

En 1971, quand j'ai commencé à travailler au laboratoire d'intelligence artificielle (I.A.) du M.I.T. (NdT : Institut de technologie du Massachusetts), j'ai intégré une communauté qui partageait le logiciel depuis de nombreuses années déjà. Le partage du logiciel n'était pas limité à notre communauté; c'est une notion aussi ancienne que les premiers ordinateurs, tout comme on partage des recettes depuis les débuts de la cuisine. Mais nous partagions davantage que la plupart.

Le laboratoire d'I.A. utilisait un système d'exploitation à temps partagé appelé ITS (système à temps partagé incompatible) que les hackers de l'équipe avaient écrit et mis au point en langage assembleur pour le Digital PDP-10, l'un des grands ordinateurs de l'époque. En tant que membre de cette communauté, hacker système de l'équipe du laboratoire d'IA, mon travail consistait à améliorer ce système.

Nous ne qualifions pas nos productions de « logiciels libres », car ce terme n'existait pas encore; c'est pourtant ce qu'elles étaient. Quand d'autres universitaires ou quand des ingénieurs souhaitaient utiliser et porter l'un de nos programmes, nous les laissions volontiers faire. Et quand on remarquait que quelqu'un utilisait un programme intéressant mais inconnu, on pouvait toujours en obtenir le code source, afin de le lire, le modifier, ou d'en réutiliser des parties dans le cadre d'un nouveau programme.

(1) L'utilisation du mot « hacker » dans le sens de « qui viole des systèmes de sécurité », est un amalgame instillé par les mass media. Nous autres hackers refusons de reconnaître ce sens, et continuons d'utiliser ce mot dans le sens « qui aime programmer et apprécie de le faire de manière astucieuse et intelligente ». (NdT : en français, on peut utiliser le néologisme « bitouilleur » pour désigner l'état d'esprit de celui qui « touille des bits »).

L'effondrement de la communauté

La situation s'est modifiée de manière radicale au début des années 1980 quand la société Digital a mis fin à la série des PDP-10. Cette architecture, élégante et puissante dans les années 60, ne pouvait pas s'étendre naturellement aux plus grands espaces d'adressage qu'on trouvait dans les années 80. Cela rendait obsolètes la quasi-totalité des programmes constituant ITS.

La communauté des hackers du laboratoire d'I.A. s'était effondrée peu de temps auparavant. La plupart d'entre eux avaient été engagés par une nouvelle société, Symbolics, et ceux qui étaient restés ne parvenaient pas à maintenir la communauté (le livre *Hackers*, écrit par Steve Levy, décrit ces événements et dépeint clairement l'apogée de cette communauté). Quand le laboratoire a, en 1982, choisi d'acheter un nouveau PDP-10, ses administrateurs ont décidé de remplacer ITS par le système de temps partagé de la société Digital, qui n'était pas libre.

Les ordinateurs modernes d'alors, tels que le VAX ou le 68020, disposaient de leurs propres systèmes d'exploitation, mais aucun d'entre eux n'était un logiciel libre : il fallait signer un accord de non divulgation pour en obtenir ne serait-ce que des copies exécutables.

Cela signifiait que la première étape de l'utilisation d'un ordinateur était de promettre de ne pas aider son prochain. On interdisait toute communauté coopérative. La règle qu'édictaient ceux qui détenaient le monopole d'un logiciel propriétaire était « Qui partage avec son voisin est un pirate. Qui souhaite la moindre modification doit nous supplier de la lui faire ».

L'idée que le système social du logiciel propriétaire - le système qui vous interdit de partager ou d'échanger le logiciel - est antisocial, immoral, et qu'il est tout bonnement incorrect, surprendra peut-être certains lecteurs. Mais comment qualifier autrement un système fondé sur la division et l'isolement des utilisateurs? Les lecteurs surpris par cette idée ont probablement pris le système social du logiciel propriétaire pour argent comptant, ou l'ont jugé en employant les termes suggérés par les entreprises vivant de logiciels propriétaires. Les éditeurs de logiciels travaillent durement, et depuis longtemps, pour convaincre

tout un chacun qu'il n'existe qu'un seul point de vue sur la question - le leur.

Quand les éditeurs de logiciels parlent de « faire respecter » leurs « droits » ou de « couper court au [piratage](#) », ce qu'ils *disent* est secondaire. Le véritable message se trouve entre les lignes, et il consiste en des hypothèses de travail qu'ils considèrent comme acquises; nous sommes censés les accepter les yeux fermés. Passons-les donc en revue.

La première hypothèse est que les sociétés éditrices de logiciels disposent d'un droit naturel, incontestable, à être propriétaire du logiciel et asseoir ainsi leur pouvoir sur tous ses utilisateurs (si c'était là un droit naturel, on ne pourrait formuler aucune objection, indépendamment du tort qu'il cause à tous). Il est intéressant de remarquer que la constitution et la tradition juridique des États-Unis d'Amérique rejettent toutes deux cette idée; le copyright n'est pas un droit naturel, mais un monopole artificiel, imposé par l'État, qui restreint le droit naturel qu'ont les utilisateurs de copier le logiciel.

Une autre hypothèse sous-jacente est que seules importent les fonctionnalités du logiciel, et que les utilisateurs d'ordinateurs n'ont pas leur mot à dire quant au modèle de société qu'ils souhaitent voir mettre en place.

Une troisième hypothèse est qu'on ne disposerait d'aucun logiciel utilisable (ou qu'on ne disposerait jamais d'un logiciel qui s'acquitte de telle ou telle tâche en particulier) si on ne garantissait pas à une société l'assise d'un pouvoir sur les utilisateurs du programme. Cette idée était plausible, jusqu'à ce que le mouvement du logiciel libre démontrât qu'on peut produire toutes sortes de logiciels utiles sans qu'il soit nécessaire de les barder de chaînes.

Si l'on se refuse à accepter ces hypothèses, et si on examine ces questions en se fondant sur une moralité dictée par le bon sens, qui place les utilisateurs au premier plan, on parvient à des conclusions bien différentes. Les utilisateurs des ordinateurs doivent être libres de modifier les programmes pour qu'ils répondent mieux à leurs besoins, et libres de partager le logiciel, car la société est fondée sur l'aide à autrui.

La place me manque ici pour développer le raisonnement menant à cette conclusion, aussi renverrai-je le lecteur au document [Pourquoi le logiciel ne devrait appartenir à personne](#).

Une profonde prise de décision

Avec la fin de ma communauté, il m'était impossible de continuer comme de par le passé. J'étais au lieu de cela confronté à une profonde prise de décision.

La solution de facilité était de rejoindre le monde du logiciel propriétaire, de signer des accords de non divulgation et promettre ainsi de ne pas aider mon ami hacker. J'aurais aussi été, très probablement, amené à développer du logiciel qui aurait été publié en fonction d'exigences de non divulgation, augmentant la pression qui en inciterait d'autres à trahir également leurs semblables.

J'aurais pu gagner ma vie de cette manière, et peut-être me serais-je même amusé à écrire du code. Mais je savais qu'à la fin de ma carrière, je n'aurais à contempler que des années de construction de murs pour séparer les gens, et que j'aurais l'impression d'avoir employé ma vie à rendre le monde pire.

J'avais déjà eu l'expérience douloureuse des accords de non divulgation, quand quelqu'un m'avait refusé, ainsi qu'au laboratoire d'I.A. du M.I.T., l'accès au code source du programme de contrôle de notre imprimante (l'absence de certaines fonctionnalités dans ce programme rendait l'utilisation de l'imprimante très frustrante). Aussi ne pouvais-je pas me dire que les accords de non divulgation étaient bénins. J'avais été très fâché du refus de cette personne de partager avec nous ; je ne pouvais pas, moi aussi, me comporter d'une telle manière à l'égard de mon prochain.

Une autre possibilité, radicale mais déplaisante, était d'abandonner l'informatique. De cette manière, mes capacités ne seraient pas employées à mauvais escient, mais elles n'en seraient pas moins gaspillées. Je ne me rendrais pas coupable de diviser et de restreindre les droits des utilisateurs d'ordinateurs, mais cela se produirait malgré tout.

Alors, j'ai cherché une façon pour un programmeur de se rendre utile pour la bonne cause. Je me suis demandé si je ne pouvais pas écrire un ou plusieurs programmes qui permettraient de souder à nouveau une communauté.

La réponse était limpide : le besoin le plus pressant était un système d'exploitation. C'est le logiciel le plus crucial pour commencer à utiliser un ordinateur. Un système d'exploitation ouvre de nombreuses portes ; sans système, l'ordinateur est inexploitable. Un système d'exploitation libre rendrait à nouveau possible une communauté de hackers, travaillant en mode coopératif - et tout un chacun serait invité à participer. Et tout un chacun pourrait utiliser un ordinateur sans devoir adhérer à une conspiration visant à priver ses amis des logiciels qu'il utilise.

En tant que développeur de système d'exploitation, j'avais les compétences requises. Aussi, bien que le succès ne me semblât pas garanti, j'ai pensé être le candidat de choix pour ce travail. J'ai décidé de rendre le système compatible avec Unix de manière à le rendre portable, et pour le rendre plus accessible aux utilisateurs d'Unix. J'ai opté pour le nom GNU, fidèle en cela à une tradition des hackers, car c'est un acronyme récuratif qui signifie « GNU's Not Unix » (GNU N'est pas Unix).

Un système d'exploitation ne se limite pas à un noyau, qui suffit à peine à exécuter d'autres programmes. Dans les années 1970, tout système d'exploitation digne de ce nom disposait d'interpréteurs de commandes, d'assembleurs, de compilateurs, d'interpréteurs, de débogueurs, d'éditeurs de textes, de logiciels de courrier électronique, pour ne citer que quelques exemples. C'était le cas d'ITS, c'était le cas de Multics, c'était le cas de VMS, et c'était le cas d'Unix. Ce serait aussi le cas du système d'exploitation GNU.

Plus tard, j'ai entendu ces mots, attribués à Hillel :

If I am not for myself, who will be for me?
If I am only for myself, what am I?
If not now, when?¹

C'est dans cet état d'esprit que j'ai pris la décision de lancer le projet GNU.

(1) En tant qu'athée, je ne suis pas d'aucun meneur religieux, mais j'admire parfois ce que l'un d'entre eux a dit.

Free comme liberté²

Le terme « free software » est mal compris : il n'a rien à voir avec le prix. Il parle de liberté. Voici donc la définition d'un logiciel libre : un programme est un logiciel libre pour vous, utilisateur en particulier, si :

- Vous avez la liberté de l'exécuter, pour quelque motif que ce soit.
- Vous avez la liberté de modifier le programme afin qu'il corresponde mieux à vos besoins (dans la pratique, pour que cette liberté prenne effet, il vous faut pouvoir accéder au code source, puisqu'opérer des modifications au sein d'un programme dont on ne dispose pas du code source est un exercice extrêmement difficile).
- Vous disposez de la liberté d'en redistribuer des copies, que ce soit gratuitement ou contre une somme d'argent.
- Vous avez la liberté de distribuer des versions modifiées du programme, afin que la communauté puisse bénéficier de vos améliorations.

Puisque le mot « free » se réfère ici à la liberté, et non au prix, il n'est pas contradictoire de vendre des copies de logiciels libres. En réalité, cette liberté est cruciale : les compilations de logiciels libres vendues sur CD-ROM sont importantes pour la communauté, et le produit de leur vente permet de lever des fonds pour le développement du logiciel libre. C'est pourquoi on ne peut pas qualifier de logiciel libre un logiciel qu'on n'a pas la liberté d'inclure dans de telles compilations.

Le mot « free » étant ambigu, on a longtemps cherché des solutions de remplacement, mais personne n'en a trouvé de convenable. La langue anglaise compte plus de mots et de nuances que toute autre langue, mais elle souffre de l'absence d'un mot simple, univoque, qui ait le sens de « free », comme liberté - « unfettered » (terme littéraire signifiant « sans entrave ») étant le meilleur candidat, d'un point de vue sémantique, des mots comme « liberated » (« libéré »), « freedom » (« liberté »), et « open » (« ouvert ») présentant tous un sens incorrect ou un autre inconvénient.

Les logiciels et le système du projet GNU

C'est une gageure que de développer un système complet. Pour mener ce projet à bien, j'ai décidé d'adapter et de réutiliser les logiciels libres existants, quand cela était possible. J'ai par exemple décidé dès le début d'utiliser TeX; comme formateur de texte principal ; quelques années plus tard, j'ai décidé d'utiliser le système de fenêtrage X plutôt que d'écrire un autre système de fenêtrage pour le projet GNU.

Cette décision a rendu le système GNU distinct de la réunion de tous les logiciels GNU. Le système GNU comprend des programmes qui ne sont pas des logiciels GNU, ce sont des programmes qui ont été développés par d'autres, dans le cadre d'autres projets, pour leurs buts propres, mais qu'on peut réutiliser, car ce sont des logiciels libres.

La genèse du projet

En janvier 1984, j'ai démissionné de mon poste au MIT et j'ai commencé à écrire les logiciels du projet GNU. Il était nécessaire que je quitte le M.I.T. pour empêcher ce dernier de s'immiscer dans la distribution du projet GNU en tant que logiciel libre. Si j'étais resté dans l'équipe, le M.I.T. aurait pu se déclarer le propriétaire de mon travail, et lui imposer ses propres conditions de distribution, voire en faire un paquetage de logiciels propriétaires. Je n'avais pas l'intention d'abattre autant de travail et de le voir rendu inutilisable pour ce à quoi il était destiné : créer une nouvelle communauté qui partage le logiciel.

Cependant, le professeur Winston, qui dirigeait alors le laboratoire d'I.A. du M.I.T., m'a gentiment invité à continuer à utiliser les équipements du laboratoire

Les premiers pas

Peu de temps avant de me lancer dans le projet GNU, j'avais entendu parler du Free University Compiler Kit³ plus connu sous le nom de VUCK (en néerlandais, le mot « free » commence par un V). Ce compilateur avait été mis au point dans l'intention de traiter plusieurs langages, parmi lesquels C et Pascal, et de produire des binaires pour de nombreuses machines cibles. J'ai écrit à son auteur en lui demandant la permission d'utiliser ce compilateur dans le cadre du projet GNU.

Il répondit d'un ton railleur, en déclarant (en anglais) que l'université était « free ». M. Stallman ne sait pas s'il voulait icidire « libre » ou « gratuite », mais pas le compilateur. J'ai alors décidé que le premier programme du projet GNU serait un compilateur traitant de plusieurs langages, sur plusieurs platesformes.

En espérant m'épargner la peine d'écrire tout le compilateur moi-même, j'ai obtenu le code source du compilateur Pastel, qui était un compilateur pour plusieurs plates-formes, développé au laboratoire Lawrence Livermore. Il compilait, et c'était aussi le langage dans lequel il avait été écrit, une version étendue de Pascal, mise au point pour jouer le rôle de langage de programmation système. J'y ai ajouté une interface pour le C, et j'ai entrepris le portage de ce programme sur le Motorola 68000. Mais j'ai dû abandonner quand j'ai découvert que ce compilateur ne fonctionnait qu'avec plusieurs méga-octets d'espace de pile disponibles, alors que le système Unix du 68000 ne gérait que 64 Ko d'espace de pile.

C'est alors que j'ai compris que le compilateur Pastel avait été mis au point de telle manière qu'il analysait le fichier en entrée, en faisait un arbre syntaxique, convertissait cet arbre syntaxique en chaîne d'« instructions », et engendrait ensuite le fichier de sortie, sans jamais libérer le moindre espace mémoire occupé. J'ai alors compris qu'il me faudrait réécrire un nouveau compilateur en partant de zéro. Ce compilateur est maintenant disponible, il s'appelle GCC ; il n'utilise rien du compilateur Pastel, mais j'ai réussi à adapter et à réutiliser l'analyseur syntaxique que j'avais écrit pour le C. Mais tout cela ne s'est produit que quelques années plus tard; j'ai d'abord travaillé sur GNU Emacs.

GNU Emacs

J'ai commencé à travailler sur GNU Emacs en septembre 1984, et ce programme commençait à devenir utilisable début 1985. Cela m'a permis d'utiliser des systèmes Unix pour éditer mes fichiers; vi et ed me laissant froid, j'avais jusqu'alors utilisé d'autres types de machines pour éditer mes fichiers.

C'est alors que j'ai reçu des requêtes de gens souhaitant utiliser GNU Emacs, ce qui a soulevé le problème de sa distribution. Je l'avais bien sûr proposé sur le serveur ftp de l'ordinateur du MIT que j'utilisais (cet ordinateur, prep.ai.mit.edu, a ainsi été promu au rang de site de distribution par ftp principal du projet GNU ; quelques années plus tard, à la fin de son exploitation, nous avons transféré ce nom sur notre nouveau serveur ftp). Mais à l'époque, une proportion importante des personnes intéressées n'avaient pas d'accès à l'Internet et ne pouvaient pas obtenir une copie du programme par ftp. La question se posait en ces termes : que devais-je leur dire ?

J'aurais pu leur dire : « Trouvez un ami qui dispose d'un accès au réseau et qui vous fera une copie. » J'aurais pu également procéder comme j'avais procédé avec la version originale d'Emacs, sur PDP-10, et leur dire : « Envoyez-moi une bande et une enveloppe timbrée auto-adressée, et je vous les renverrai avec Emacs. » Mais j'étais sans emploi, et je cherchais des moyens de gagner de l'argent grâce au logiciel libre. C'est pourquoi j'ai annoncé que j'enverrais une bande à quiconque en désirait une, en échange d'une contribution de 150 dollars américains. De cette manière, je mettais en place une entreprise autour du marché de la distribution du logiciel libre, entreprise précurseur des sociétés qu'on trouve aujourd'hui et qui distribuent des systèmes GNU entiers fondés sur Linux.

Un programme est-il libre pour chacun de ses utilisateurs ?

Si un programme est un logiciel libre au moment où il quitte les mains de son auteur, cela ne signifie pas nécessairement qu'il sera un logiciel libre pour quiconque en possédera une copie. Le logiciel relevant du [domaine public](#), par exemple (qui ne tombe sous le coup d'aucun copyright), est du logiciel libre ; mais tout un chacun peut en produire une version propriétaire modifiée. De façon comparable, de nombreux programmes libres sont couverts par des copyrights mais distribués sous des licences permissives qui autorisent la création de versions modifiées et propriétaires.

L'exemple le plus frappant de ce problème est le système de fenêtrage X. Développé au M.I.T., et distribué sous forme de logiciel libre sous une licence permissive, il a rapidement été adopté par divers constructeurs. Ils ont ajouté X à leurs systèmes Unix propriétaires, sous forme binaire uniquement, en le frappant du même accord de non divulgation. Ces exemplaires de X n'étaient en rien du logiciel plus libre que le reste d'Unix.

Les développeurs du système de fenêtrage X ne voyaient là nul problème (ils s'attendaient à cela et souhaitaient un tel résultat). Leur but n'était pas la liberté, mais la simple « réussite », définie comme le fait d'« avoir beaucoup d'utilisateurs. » Peu leur importait la liberté de leurs utilisateurs, seul leur nombre revêtait de l'importance à leurs yeux.

Cela a conduit à une situation paradoxale, où deux différentes façons d'évaluer la liberté répondaient de manières différentes à la question « Ce programme est-il libre ? » Qui fondait son jugement sur la liberté fournie par les conditions de distribution de la distribution du M.I.T., concluait que X était un logiciel libre. Mais qui mesurait la liberté de l'utilisateur type de X, devait conclure que X était un logiciel propriétaire. La plupart des utilisateurs de X exécutaient des versions propriétaires fournies avec des systèmes Unix, et non la version libre.

Le copyleft et la GPL de GNU

Le but du projet GNU était de rendre les utilisateurs libres, pas de se contenter d'être populaire. Nous avons besoin de conditions de distribution qui empêcheraient de transformer du logiciel GNU en logiciel propriétaire. Nous avons utilisé la méthode du copyleft (1), ou « gauche d'auteur ».

Le gauche d'auteur utilise les lois du droit d'auteur, en les retournant pour leur faire servir le but opposé de ce pour quoi elles ont été conçues : ce n'est pas une manière de privatiser du logiciel, mais une manière de le laisser « libre ».

L'idée centrale du gauche d'auteur 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.

Pour que le gauche d'auteur soit efficace, il faut que les versions modifiées demeurent libres, afin de s'assurer que toute œuvre dérivée de notre travail reste disponible à la communauté en cas de publication. Quand des programmeurs professionnels se portent volontaires pour améliorer le logiciel GNU, c'est le gauche d'auteur qui empêche leurs employeurs de dire : « Vous ne pouvez pas partager ces modifications, car nous allons les utiliser dans le cadre de notre version propriétaire du programme. »

Il est essentiel d'imposer que les modifications restent libres si on souhaite garantir la liberté de tout utilisateur du programme. Les sociétés qui ont privatisé le système de fenêtrage X faisaient en général quelques modifications pour le porter sur leurs systèmes et sur leur matériel. Ces modifications étaient ténues si on les comparait à X dans son ensemble, mais elles n'en étaient pas pour autant faciles. Si le fait de procéder à des modifications pouvait servir de prétexte à ôter leur liberté aux utilisateurs, il serait facile pour quiconque de s'en servir à son avantage.

Le problème de la réunion d'un programme libre avec du code non libre est similaire. Une telle combinaison serait indubitablement non libre; les libertés absentes de la partie non libre du programme ne se trouveraient pas non plus dans l'ensemble résultat de cette compilation. Autoriser de telles pratiques ouvrirait une voie d'eau suffisante pour couler le navire. C'est pourquoi il est crucial pour le gauche d'auteur d'exiger qu'un programme couvert par le gauche d'auteur ne puisse pas être inclus dans une version plus grande sans que cette dernière ne soit également couverte par le gauche d'auteur.

La mise en œuvre spécifique du gauche d'auteur que nous avons utilisée pour la plupart des logiciels GNU fut la GNU General Public License (licence publique générale de GNU), ou GNU GPL en abrégé. Nous disposons d'autres types de gauche d'auteur pour des circonstances particulières. Les manuels du projet GNU sont eux aussi couverts par le gauche d'auteur, mais en utilisent une version très simplifiée, car il n'est pas nécessaire de faire appel à toute la complexité de la GNU GPL dans le cadre de manuels.(2)

(1)En 1984 ou 1985, Don Hopkins (dont l'imagination était sans bornes) m'a envoyé une lettre. Il avait écrit sur l'enveloppe plusieurs phrases amusantes, et notamment celle-ci : « Copyleft - all rights reversed. »⁴. J'ai utilisé le mot « copyleft » pour donner un nom au concept de distribution que je développais alors.

(2) Nous utilisons maintenant la [GNU Free Documentation License](#) pour la documentation.

La Free Software Foundation, ou Fondation pour le Logiciel Libre

Emacs attirant de plus en plus l'attention, le projet GNU comptait un nombre croissant de participants, et nous avons décidé qu'il était temps de repartir à la chasse aux fonds. En 1985, nous avons donc créé la fondation du logiciel libre (FSF), une association à but non lucratif, exemptée d'impôts, pour le développement de logiciel libre.

La FSF a récupéré le marché de la distribution de logiciel libre sur bandes, auxquelles elle ajouta ensuite d'autres logiciels libres (GNU ou non), et par la vente de manuels libres. La FSF accepte les dons, mais la plus grande partie de ses recettes est toujours provenue des ventes - de copies de logiciel libre ou d'autres services associés. De nos jours, elle vend des CD-ROM de code source, des CD-ROM de binaires, des manuels de qualité (tout cela, en autorisant la redistribution et les modifications), et des distributions Deluxe (dans lesquelles nous construisons tous les logiciels pour la plate-forme de votre choix).

Les employés de la fondation du logiciel libre ont écrit et maintenu un grand nombre de paquetages logiciels du projet GNU, en particulier la bibliothèque du langage C et l'interpréteur de commandes. La bibliothèque du langage C est ce qu'utilise tout programme fonctionnant sur un système GNU/Linux pour communiquer avec Linux. Elle a été développée par Roland McGrath, membre de l'équipe de la fondation du logiciel libre. L'interpréteur de commandes employé sur la plupart des systèmes GNU/Linux est BASH, le Bourne-Again Shell, qui a été développé par Brian Fox, employé de la FSF.

Nous avons financé le développement de ces programmes car le projet GNU ne se limitait pas aux outils ou à un environnement de développement. Notre but était la mise en place d'un système d'exploitation complet, et de tels programmes étaient nécessaires pour l'atteindre.

(1) « Bourne-Again Shell »⁵ est un clin d'œil au nom « BourneShell », qui était l'interpréteur de commandes habituel sur Unix.

Assistance technique au logiciel libre

La philosophie du logiciel libre rejette une pratique spécifique, très répandue dans l'industrie du logiciel, mais elle ne s'oppose pas au monde des affaires. Quand des entreprises respectent la liberté des utilisateurs, nous leur souhaitons de réussir.

La vente de copies d'Emacs est une forme d'affaires fondées sur du logiciel libre. Quand la FSF a récupéré ce marché, j'ai dû chercher une autre solution pour gagner ma vie. Je l'ai trouvée sous la forme de vente de services associés au logiciel libre que j'avais développé. Cela consistait à enseigner des thèmes tels que la programmation de GNU Emacs et la personnalisation de GCC, et à développer du logiciel, principalement en portant GCC sur de nouvelles plateformes.

De nos jours, chacune de ces activités lucratives fondées sur le logiciel libre est proposée par de nombreuses sociétés. Certaines distribuent des compilations de logiciel libre sur CD-ROM ; d'autres vendent de l'assistance technique en répondant à des questions d'utilisateurs, en corrigeant des bogues, et en insérant de nouvelles fonctionnalités majeures. On commence même à observer des sociétés de logiciel libre fondées sur la mise sur le marché de nouveaux logiciels libres.

Prenez garde, toutefois : certaines des sociétés qui s'associent à la dénomination « open source »⁶. Il s'agit ici de sociétés qui font peu de cas du logiciel libre et choisissent un slogan calculé pour s'attirer les faveurs du public, fondent en réalité leur activité sur du logiciel propriétaire, qui interagit avec du logiciel libre. Ce ne sont pas des sociétés de logiciel libre, ce sont des sociétés de logiciel propriétaire dont les produits détournent les utilisateurs de leur liberté. Elles appellent cela de la « valeur ajoutée », ce qui reflète quelles valeurs elles souhaitent nous voir adopter : préférer la facilité à la liberté. Si nous faisons passer la liberté au premier plan, il nous faut leur donner le nom de produits à « liberté soustraite ».

Objectifs techniques

L'objectif principal du projet GNU était le logiciel libre. Même si GNU ne jouissait d'aucun avantage technique sur Unix, il disposerait d'un avantage social, en autorisant les utilisateurs à coopérer, et d'un avantage éthique, en respectant la liberté de l'utilisateur.

Mais il était naturel d'appliquer à ce travail les standards bien connus du développement logiciel de qualité en utilisant par exemple des structures de données allouées dynamiquement pour éviter de mettre en place des limites fixées arbitrairement, et en gérant tous les caractères possibles encodables sur 8 bits, partout où cela avait un sens.

De plus, nous rejetions l'accent mis par Unix sur les petites quantités de mémoire, en décidant de ne pas nous occuper des architectures 16 bits (il était clair que les architectures 32 bits seraient la norme au moment de la finalisation du système GNU), et en ne faisant aucun effort pour réduire la consommation mémoire en deçà d'un méga-octet. Dans les programmes pour lesquels il n'était pas crucial de manipuler des fichiers de tailles importantes, nous encourageons les programmeurs à lire le fichier en entrée, d'une traite, en mémoire, et d'analyser ensuite son contenu sans plus se préoccuper des entrées/sorties.

Ces décisions ont rendu de nombreux programmes du projet GNU supérieurs à leurs équivalents sous Unix en termes de fiabilité et de vitesse d'exécution.

Les ordinateurs offerts

La réputation du projet GNU croissant, on nous offrait des machines sous Unix pour nous aider à le mener à bien. Elles nous furent bien utiles, car le meilleur moyen de développer les composants de GNU était de travailler sur un système Unix, dont on remplaçait les composants un par un. Mais cela a posé un problème éthique : était-il correct ou non, pour nous, de posséder des copies d'Unix ?

Unix était (et demeure) du logiciel propriétaire, et la philosophie du projet GNU nous demandait de ne pas utiliser de logiciels propriétaire. Mais, en appliquant le même raisonnement que celui qui conclut qu'il est légitime de faire usage de violence en situation de légitime défense, j'ai conclu qu'il était légitime d'utiliser un paquetage propriétaire quand cela était crucial pour développer une solution de remplacement libre, qui en aiderait d'autres à se passer de ce même paquetage propriétaire.

Mais ce mal avait beau être justifiable, il n'en restait pas moins un mal. De nos jours, nous ne possédons plus aucune copie d'Unix, car nous les avons toutes remplacées par des systèmes d'exploitation libres. Quand nous ne parvenions pas à substituer au système d'exploitation d'une machine un système libre, nous remplaçons la machine.

La GNU Task List, ou liste des tâches du projet GNU

Le projet GNU suivant son cours, on trouvait ou développait un nombre croissant de composants du système, et il est finalement devenu utile de faire la liste des parties manquantes. Nous l'avons utilisée pour recruter des développeurs afin d'écrire ces dernières. Cette liste a pris le nom de GNU task list. En plus des composants manquants d'Unix, nous y avons listé plusieurs autres projets utiles, de logiciel et de documentation, que nous jugeons nécessaires au sein d'un système réellement complet.

De nos jours, on ne trouve presque plus aucun composant d'Unix dans la liste des tâches du projet GNU - ces travaux ont tous été menés à bien, si on néglige certains composants non essentiels. Mais la liste est pleine de projets qu'on pourrait qualifier d'« applications ». Tout programme qui fait envie à une classe non restreinte d'utilisateurs constituerait un ajout utile à un système d'exploitation.

On trouve même des jeux dans la liste des tâches (et c'est le cas depuis le commencement). Unix proposait des jeux, ce devait naturellement être également le cas de GNU. Mais il n'était pas nécessaire d'être compatible en matière de jeux, aussi n'avons-nous pas suivi la liste des jeux d'Unix. Nous avons plutôt listé un spectre de divers types de jeux qui plairaient vraisemblablement aux utilisateurs.

La GNU Library GPL, ou licence publique générale de GNU pour les bibliothèques

La bibliothèque du langage C du projet GNU fait appel à un gauchiste d'auteur particulier, appelé la GNU Library General Public License (1) (licence publique générale de GNU pour les bibliothèques, ou GNU LGPL), qui autorise la liaison de logiciel propriétaire avec la bibliothèque. Pourquoi une telle exception ?

Ce n'est pas une question de principe ; aucun principe ne dicte que les logiciels propriétaires ont le droit de contenir notre code (pourquoi contribuer à un projet qui affirme refuser de partager avec nous ?). L'utilisation de la LGPL dans le cadre de la bibliothèque du langage C, ou de toute autre bibliothèque, est un choix stratégique.

La bibliothèque du langage C joue un rôle générique ; tout système propriétaire, tout compilateur, dispose d'une bibliothèque du langage C. C'est pourquoi limiter l'utilisation de notre bibliothèque du langage C au logiciel libre n'aurait donné aucun avantage au logiciel libre - cela n'aurait eu pour effet que de décourager l'utilisation de notre bibliothèque.

Il existe une exception à cette règle : sur un système GNU (et GNU/Linux est l'un de ces systèmes), la bibliothèque du langage C de GNU est la seule disponible. Aussi, ses conditions de distribution déterminent s'il est possible de compiler un programme propriétaire sur le système GNU. Il n'existe aucune raison éthique d'autoriser des applications propriétaires sur le système GNU, mais d'un point de vue stratégique, il semble que les interdire découragerait plus l'utilisation d'un système GNU que cela n'encouragerait le développement d'applications libres.

C'est pourquoi l'utilisation de la GPL pour les bibliothèques (ou LGPL) est une bonne stratégie dans le cadre de la bibliothèque du langage C. En ce qui concerne les autres bibliothèques, il faut prendre la décision stratégique au cas par cas. Quand une bibliothèque remplit une tâche particulière qui peut faciliter l'écriture de certains types de programmes, la distribuer sous les conditions de la GPL, en limitant son utilisation aux programmes libres, est une manière d'aider les développeurs de logiciels libres et de leur accorder un avantage à l'encontre du logiciel propriétaire.

Considérons GNU Readline, une bibliothèque développée dans le but de fournir une édition de ligne de commande pour l'interpréteur de commandes BASH. Cette bibliothèque est distribuée sous la licence publique générale ordinaire de GNU, et non pas sous la LGPL. Cela a probablement pour effet de réduire l'utilisation de la bibliothèque Readline, mais cela n'induit aucune perte en ce qui nous concerne. Pendant ce temps, on compte au moins une application utile qui a été libérée, uniquement dans le but de pouvoir utiliser

la bibliothèque Readline, et c'est là un gain réel pour la communauté.

Les développeurs de logiciel propriétaire jouissent des avantages que leur confère l'argent; les développeurs de logiciel libre doivent compenser cela en s'épaulant les uns les autres. J'espère qu'un jour nous disposerons de toute une collection de bibliothèques couvertes par la GPL, et pour lesquelles il n'existera pas d'équivalent dans le monde du logiciel propriétaire. Nous disposerons ainsi de modules utiles, utilisables en tant que blocs de construction de nouveaux logiciels libres, et apportant un avantage considérable à la continuation du développement du logiciel libre.

(1) Cette licence s'appelle maintenant la GNU Lesser General Public License, pour éviter de laisser penser que toutes les bibliothèques doivent l'utiliser.

Gratter là où ça démange

Éric Raymond affirme que « Tout bon logiciel commence par gratter un développeur là où ça le démange. ». Cela se produit peut-être, parfois, mais de nombreux composants essentiels du logiciel GNU ont été développés dans le but de disposer d'un système d'exploitation libre complet. Ils ont été inspirés par une vision et un projet à long terme, pas par un coup de tête.

Nous avons par exemple développé la bibliothèque du langage C de GNU car un système de type Unix a besoin d'une bibliothèque du langage C, nous avons développé le Bourne-Again Shell (BASH) car un système de type Unix a besoin d'un interpréteur de commandes, et nous avons développé GNU tar car un système de type Unix a besoin d'un programme d'archivage. Il en va de même pour les programmes que j'ai développé, à savoir le compilateur C de GNU, GNU Emacs, GDB, et GNU Make.

Certains programmes du projet GNU ont été développés pour répondre aux menaces qui pesaient sur notre liberté. C'est ainsi que nous avons développé gzip en remplacement du programme Compress, que la communauté avait perdu suite aux brevets logiciels déposés sur LZW. Nous avons trouvé des gens pour développer LessTif, et plus récemment nous avons démarré les projets GNOME et Harmony, en réponse aux problèmes posés par certaines bibliothèques propriétaires (lire ci-après). Nous sommes en train de développer le GNU Privacy Guard (le gardien de l'intimité de GNU, ou GPG) pour remplacer un logiciel de chiffrement populaire mais pas libre, car les utilisateurs ne devraient pas devoir choisir entre la préservation de leur intimité et la préservation de leur liberté.

Bien sûr, les gens qui écrivent ces programmes se sont intéressés à ce travail, et de nombreux contributeurs ont ajouté de nouvelles fonctionnalités car elles comblaient leurs besoins ou les intéressaient. Mais ce n'est pas là la raison première de ces programmes.

Des développements inattendus

Au commencement du projet GNU, j'ai imaginé que nous développerions le système GNU dans sa globalité avant de le publier. Les choses se sont passées différemment.

Puisque chaque composant du système GNU était mis en œuvre sur un système Unix, chaque composant pouvait fonctionner sur des systèmes Unix, bien avant que le système GNU ne soit disponible dans sa globalité. Certains de ces programmes sont devenus populaires, et leurs utilisateurs ont commencé à travailler sur des extensions et des ports - vers les diverses versions d'Unix, incompatibles entre elles, et parfois, sur d'autres systèmes encore.

Ce processus a rendu ces programmes bien plus complets, et a drainé des fonds et des participants vers le projet GNU. Mais il a probablement eu également pour effet de retarder de plusieurs années la mise au point d'un système en état de fonctionnement, puisque les développeurs du projet GNU passaient leur temps à s'occuper de ces ports et à proposer des nouvelles fonctionnalités aux composants existants, plutôt que de continuer à développer peu à peu les composants manquants.

Le GNU Hurd

En 1990, le système GNU était presque terminé; le seul composant principal qui manquait encore à l'appel était le noyau. Nous avons décidé d'implémenter le noyau sous la forme d'une série de processus

serveurs qui fonctionneraient au-dessus de Mach. Mach est un micro-noyau développé à l'université Carnegie-Mellon puis à l'université de l'Utah; le GNU Hurd est une série de serveurs (ou une « horde de gnous ») qui fonctionnent au-dessus de Mach, et remplissent les diverses fonctions d'un noyau Unix. Le développement a été retardé car nous attendions que Mach soit publié sous forme de logiciel libre, comme cela avait été promis.

L'une des raisons qui ont dicté ce choix était d'éviter ce qui semblait être la partie la plus difficile du travail : déboguer un programme de noyau sans disposer pour cela d'un débogueur au niveau du code source. Ce travail avait déjà été fait, dans Mach, et nous pensions déboguer les serveurs du Hurd en tant que programmes utilisateur, à l'aide de GDB. Mais cela prit beaucoup de temps, et les serveurs à plusieurs processus légers, qui s'envoyaient des messages les uns aux autres, se sont révélés très difficiles à déboguer. La consolidation du Hurd s'est étalée sur plusieurs années.

Alix

À l'origine, le noyau du système GNU n'était pas censé s'appeler Hurd. Son premier nom était Alix - du nom de celle qui à l'époque était l'objet de ma flamme. Administratrice de systèmes Unix, elle avait fait remarquer que son prénom ressemblait aux noms typiques des versions de systèmes Unix; elle s'en était ouverte auprès d'amis en plaisantant : « Il faudrait baptiser un noyau de mon nom. » Je suis resté coi, mais ai décidé de lui faire la surprise d'appeler Alix le noyau du système GNU.

Mais les choses ont changé. Michael Bushnell (maintenant, il s'agit de Thomas), le développeur principal du noyau, préférait le nom Hurd, et a confiné le nom Alix à une certaine partie du noyau - la partie qui se chargeait d'intercepter les appels système et de les gérer en envoyant des messages aux serveurs du Hurd.

Finalement, Alix et moi mêmes fin à notre relation, et elle a changé de nom; de manière indépendante, le concept du Hurd avait évolué de telle sorte que ce serait la bibliothèque du langage C qui enverrait directement des messages aux serveurs, ce qui a fait disparaître le composant Alix du projet.

Mais avant que ces choses ne se produisissent, un de ses amis avait remarqué le nom Alix dans le code source du Hurd, et s'en était ouvert auprès d'elle. Finalement, ce nom avait rempli son office.

Linux et GNU/Linux

Le GNU Hurd n'est pas encore utilisable de manière intensive. Heureusement, on dispose d'un autre noyau. En 1991, Linus Torvalds a développé un noyau compatible avec Unix et lui a donné le nom de Linux. Aux alentours de 1992, la jonction de Linux et du système GNU, qui était presque complet, a fourni un système d'exploitation libre et complet (ce travail de jonction était lui-même, bien sûr, considérable). C'est grâce à Linux qu'on peut désormais employer une version du système GNU.

On appelle cette version du système « GNU/Linux » pour signaler qu'il est composé du système GNU et du noyau Linux.

Les défis à venir

Nous avons fait la preuve de notre capacité à développer un large spectre de logiciel libre. Cela ne signifie pas que nous sommes invincibles et que rien ne peut nous arrêter. Certains défis rendent incertain l'avenir du logiciel libre; et il faudra des efforts et une endurance soutenus pour les relever, pendant parfois plusieurs années. Il faudra montrer le genre de détermination dont les gens font preuve quand ils accordent de la valeur à leur liberté et qu'ils ne laisseront personne la leur voler.

Les quatre sections suivantes discutent de ces défis.

Le matériel secret

Les fabricants de matériel tendent de plus en plus à garder leurs spécifications secrètes. Cela rend plus difficile l'écriture de pilotes de périphériques libres afin de permettre à Linux et au projet XFree86 de

reconnaître de nouveaux matériels. Nous disposons aujourd'hui de systèmes entièrement libres, mais cela pourrait ne plus être le cas dans l'avenir, si nous ne pouvons plus proposer des pilotes pour les ordinateurs de demain.

On peut résoudre ce problème de deux manières. Les programmeurs peuvent analyser l'ensemble afin de deviner comment prendre en compte le matériel. Les autres peuvent choisir le matériel qui est reconnu par du logiciel libre; plus nous serons nombreux, plus la politique de garder les spécifications secrètes sera vouée à l'échec.

La rétro-ingénierie est un travail conséquent; disposerons-nous de programmeurs suffisamment déterminés pour le prendre en main ? Oui - si nous avons construit un sentiment puissant selon lequel le logiciel libre est une question de principe, et que les pilotes non libres sont inacceptables. Et serons-nous nombreux à dépenser un peu plus d'argent, ou à passer un peu de temps, pour que nous puissions utiliser des pilotes libres ? Oui - si la détermination afférente à la liberté est largement répandue.

(Note de 2008 : Ce problème s'applique au BIOS aussi. Il existe un BIOS libre, coreboot ; le problème est d'obtenir les spécifications pour les machines pour que coreboot puisse les gérer).

Les bibliothèques non libres

Une bibliothèque non libre qui fonctionne sur des systèmes d'exploitation libres se comporte comme un piège vis-à-vis des développeurs de logiciel libre. Les fonctionnalités attrayantes de cette bibliothèque sont l'appât; si vous utilisez la bibliothèque, vous tombez dans le piège, car votre programme ne peut pas être utilisé de manière utile au sein d'un système d'exploitation libre (pour être strict, on pourrait y inclure le programme, mais on ne pourrait pas **l'exécuter** en l'absence de la bibliothèque incriminée). Pire encore, si un programme qui utilise une bibliothèque propriétaire devient populaire, il peut attirer d'autres programmeurs peu soupçonneux dans le piège.

Ce problème s'est posé pour la première fois avec la boîte à outils Motif, dans les années 80. Même s'il n'existait pas encore de systèmes d'exploitation libres, il était limpide que Motif leur causerait des problèmes, plus tard. Le projet GNU a réagi de deux manières : en demandant aux projets de logiciel libre de rendre l'utilisation de Motif facultative en privilégiant les gadgets de la boîte à outils X, libre, et en recherchant un volontaire pour écrire une solution de remplacement libre à Motif. Ce travail prit de nombreuses années; LessTif, développé par les Hungry Programmers (les « Programmeurs affamés »), n'est devenu suffisamment étendu pour faire fonctionner la plupart des applications utilisant Motif qu'en 1997.

De 1996 à 1998, une compilation conséquente de logiciel libre, le bureau KDE, a fait usage d'une autre bibliothèque non libre de boîte à outils pour l'interface graphique utilisateur, appelée Qt. Les systèmes GNU/Linux libres ne pouvaient pas utiliser KDE, car nous ne pouvions pas utiliser la bibliothèque.

Les systèmes GNU/Linux libres ne pouvaient pas utiliser KDE, car nous ne pouvions pas utiliser la bibliothèque. Cependant, certains distributeurs commerciaux de systèmes GNU/Linux n'ont pas été assez stricts pour coller au logiciel libre et ont ajouté KDE dans leurs systèmes - produisant un système disposant d'un plus grand nombre de fonctionnalités, mais souffrant d'une liberté réduite. Le groupe KDE encourageait activement un plus grand nombre de programmeurs à utiliser la bibliothèque Qt, et des millions de « nouveaux utilisateurs de Linux » n'ont jamais eu connaissance du fait que tout ceci posait un problème. La situation était sinistre.

La communauté du logiciel libre a répondu à ce problème de deux manières : GNOME et Harmony.

GNOME, le GNU Network Object Model Environment (environnement de GNU de modèle d'objets pour le réseau), est le projet de bureau de GNU. Démarré en 1997 par Miguel de Icaza, et développé avec l'aide de la société Red Hat Software, GNOME avait pour but de fournir des fonctionnalités de bureau similaires, en utilisant exclusivement du logiciel libre. Il jouit aussi d'avantages techniques, comme le fait de collaborer avec toute une variété de langages, et de ne pas de se limiter au C++. Mais son objectif principal est la liberté : ne pas imposer l'utilisation du moindre logiciel non libre.

Harmony est une bibliothèque compatible de remplacement, conçue pour permettre l'utilisation des logiciels de KDE sans faire appel à Qt.

En novembre 1998, les développeurs de Qt ont annoncé une modification de leur licence qui, quand elle sera effective, fera de Qt un logiciel libre. On ne peut pas en être sûr, mais je pense que cette décision est en partie imputable à la réponse ferme qu'a faite la communauté au problème que Qt posait quand il n'était pas libre (la nouvelle licence n'est pas pratique ni équitable, aussi demeure-t-il préférable d'éviter d'utiliser Qt).

[Note ultérieure : en septembre 2000, Qt fut distribuée sous la GPL de GNU, ce qui résolvait essentiellement ce problème.]

Comment répondrons-nous à la prochaine bibliothèque non libre mais alléchante ? La communauté comprendra-t-elle dans son entier la nécessité de ne pas tomber dans le piège ? Ou serons-nous nombreux à préférer la facilité à la liberté, et à produire un autre problème majeur ? Notre avenir dépend de notre philosophie.

Les brevets sur les logiciels

La pire menace provient des brevets sur les logiciels, susceptibles de placer des algorithmes et des fonctionnalités hors de portée des logiciels libres pendant une période qui peut atteindre vingt ans. Les brevets sur l'algorithme de compression LZW ont été déposés en 1983, et nous ne pouvons toujours pas diffuser des logiciels libres qui produisent des images au format GIF correctement compressées. En 1998, la menace d'une poursuite pour cause de violation de brevets a mis fin à la distribution d'un programme libre qui produisait des données sonores compressées au format MP3.

Il existe plusieurs manières de répondre au problème des brevets : on peut rechercher des preuves qui invalident un brevet, et on peut rechercher d'autres solutions pour remplir une tâche. Mais chacune de ces méthodes ne fonctionne que dans certains cas; quand les deux échouent, il se peut qu'un brevet empêche le logiciel libre de disposer de fonctionnalités souhaitées par les utilisateurs. Que ferons-nous dans ce genre de situation ?

Ceux d'entre nous qui prètent de la valeur au logiciel libre par amour de la liberté continueront à utiliser du logiciel libre dans tous les cas. On pourra travailler sans utiliser de fonctionnalités protégées par des brevets. Mais ceux d'entre nous qui prètent de la valeur au logiciel libre car ils s'attendent à trouver là des logiciels techniquement supérieurs sont susceptibles de critiquer l'idée même du logiciel libre quand un brevet l'empêchera de progresser plus avant. Ainsi, même s'il est utile de discuter de l'efficacité, dans la pratique, du modèle de développement de type « cathédrale » [\(1\)](#), et de la fiabilité et de la puissance de certains logiciels libres, il ne faut pas s'en tenir là. Il nous faut parler de liberté et de principes.

(1) Il aurait été plus clair d'écrire « du modèle « bazaar », étant donné qu'il s'agissait de la nouvelle alternative, controversée au début.

La documentation libre

Il ne faut pas chercher les lacunes les plus graves de nos systèmes d'exploitations libres dans le logiciel - c'est l'absence de manuels libres corrects qu'on puisse inclure dans nos systèmes qui se fait le plus cruellement sentir. La documentation est essentielle dans tout paquetage logiciel; quand un paquetage logiciel important ne dispose pas d'un bon manuel libre, il s'agit d'un manque crucial. On en compte de nombreux aujourd'hui.

La documentation libre, tout comme le logiciel libre, est une question de liberté, pas de prix. La raison d'être d'un manuel libre est très proche de celle d'un logiciel libre : il s'agit d'offrir certaines libertés à tous les utilisateurs. Il faut autoriser la redistribution (y compris la vente commerciale), en ligne et sur papier, de telle sorte que le manuel puisse accompagner toute copie du programme.

Il est également crucial d'autoriser les modifications. En règle générale, je ne pense pas qu'il soit essentiel d'autoriser tout un chacun à modifier toutes sortes d'articles et de livres. Je ne pense pas, par exemple, que vous ou moi soyons tenus de donner la permission de modifier des textes comme le présent article, qui expose nos actions et nos idées.

Mais il existe une raison particulière, pour laquelle il est crucial de disposer de la liberté de modifier la documentation afférente au logiciel libre. Quand on jouit de son droit de modifier le logiciel, et d'ajouter des fonctionnalités ou de modifier les fonctionnalités présentes, le programmeur consciencieux mettra

immédiatement à jour le manuel (afin de fournir une documentation précise et utilisable aux côtés du programme modifié). Un manuel qui n'autorise pas les programmeurs à être consciencieux et à terminer leur travail, ne remplit pas les besoins de notre communauté.

Il est acceptable d'apposer certaines limites sur la manière dont les modifications sont faites. Il est par exemple envisageable d'exiger de préserver la notice de copyright de l'auteur original, les conditions de distribution, ou la liste des auteurs. D'exiger que les versions modifiées contiennent une notice qui stipule qu'elles ont été modifiées, et même d'interdire de modifier ou d'ôter des sections entières, pourvu que ces sections ne traitent pas de considérations techniques, ne pose pas non plus de problèmes, car cela n'interdit pas au programmeur consciencieux d'adapter le manuel afin qu'il corresponde au programme modifié par ses soins. En d'autres termes, cela n'empêche la communauté du logiciel libre d'utiliser pleinement le manuel.

En revanche, il faut autoriser la modification des portions *techniques* du manuel, et la distribution du résultat de ces modifications par tous les médias habituels, à travers tous les canaux habituels; sans quoi, les restrictions font obstruction à la communauté, le manuel n'est pas libre, et il nous en faut un autre.

Les développeurs de logiciels libres seront-ils déterminés, auront-ils conscience du fait qu'il est nécessaire de produire tout un spectre de manuels libres ? Une fois de plus, notre avenir dépend de notre philosophie.

Il nous faut faire l'apologie de la liberté

On estime aujourd'hui à dix millions le nombre d'utilisateurs de systèmes GNU/Linux et Red Hat Linux de par le monde. Le logiciel libre propose tant d'avantages pratiques que les utilisateurs s'y ruent pour des raisons purement pratiques.

Cet état de fait a des conséquences heureuses, qui n'échapperont à personne : on voit plus de développeurs intéressés par la production de logiciels libres, les entreprises de logiciels libres comptent plus de clients, et il est plus facile d'encourager les sociétés à développer des logiciels libres commerciaux, plutôt que des produits logiciels propriétaires.

Mais l'intérêt pour le logiciel libre croît plus vite que la prise de conscience de la philosophie sur laquelle il se fonde, et cela provoque des problèmes. Notre capacité à relever les défis et à répondre aux menaces évoqués plus haut dépend de notre volonté à défendre chèrement notre liberté. Pour nous assurer que notre communauté partage cette volonté, il nous faut répandre ces idées auprès des nouveaux utilisateurs au fur et à mesure qu'ils rejoignent notre communauté.

Mais nous négligeons ce travail; on dépense bien plus d'efforts pour attirer de nouveaux utilisateurs dans notre communauté qu'on n'en dépense pour leur enseigner l'éducation civique qui lui est attachée. Ces deux efforts sont nécessaires, et il nous faut les équilibrer.

« Open Source »

En 1998, il est devenu plus difficile de sensibiliser les nouveaux utilisateurs à la notion de liberté dans le logiciel, quand une portion de notre communauté a choisi d'arrêter d'utiliser le terme « Free Software » pour lui préférer la dénomination « Open Source software »⁷.

Certains de ceux qui ont choisi ce nouveau nom avaient en tête de mettre fin à la confusion souvent constatée entre les mots « free » et « gratuit » - ce qui est un objectif valable. D'autres, au contraire, avaient pour objectif de laisser de côté le principe qui a depuis toujours motivé le mouvement du logiciel libre et le projet GNU, afin de cibler les cadres et les utilisateurs professionnels, dont beaucoup ont une idéologie où la liberté, la communauté, et les principes, cèdent le pas aux profits. Ainsi, la rhétorique de l'« Open Source » met l'accent sur le potentiel pour faire du logiciel puissant et de grande qualité, mais occulte délibérément les idées de liberté, de communauté, et de principes.

Les magazines « Linux » illustrent clairement cet exemple (ils sont bourrés de publicités pour des logiciels propriétaires qui fonctionnent sur le système GNU/Linux). Quand le prochain Motif ou Qt poindra, ces magazines mettront-ils les programmeurs en garde en leur demandant de s'en éloigner, ou passeront-ils des publicités pour ces produits ?

La communauté a beaucoup à gagner de la participation des entreprises; toutes choses étant égales par ailleurs, cette contribution est utile. Mais sacrifier à cette aide les discours traitant de liberté et de principes peut avoir des conséquences désastreuses; cela déséquilibre encore plus la situation précédente, où on voit que l'éducation civique des nouveaux utilisateurs s'avère difficile lorsqu'ils affluent.

Les termes « Free Software » et « Open Source » décrivent tous deux plus ou moins la même catégorie de logiciels, mais correspondent à des conceptions différentes du logiciel et des valeurs qui lui sont associées. Le projet GNU continue d'utiliser le terme « Free Software » pour exprimer l'idée que la liberté est plus importante que la seule technique.

Jetez-vous à l'eau

La philosophie de Yoda (il ne faut pas essayer) est attirante, mais elle ne s'applique pas à moi. J'ai effectué la plupart de mes travaux sans savoir si j'étais capable de les mener à bien, et sans savoir si ces derniers, une fois menés à bien, suffiraient aux buts que je leur avais fixés. Mais j'ai tenté ma chance, car il n'y avait personne d'autre que moi entre l'ennemi et ma cité. À ma grande surprise, j'ai parfois réussi.

J'ai parfois échoué; certaines de mes cités sont tombées. Je trouvais alors une autre cité menacée, et je me préparais pour une nouvelle bataille. Avec le temps, j'ai appris à reconnaître les menaces et à m'interposer entre ces dernières et ma cité, en appelant mes amis hackers à la rescousse.

Maintenant, il arrive souvent que je ne sois pas seul. C'est pour moi un soulagement et une joie de constater que tout un régiment de hackers se mobilise pour faire front, et je réalise qu'il se peut que cette cité survive - pour le moment. Mais les dangers grandissent chaque année, et maintenant la société Microsoft a explicitement pris notre communauté dans son collimateur. L'avenir de la liberté n'est pas un fait acquis. Ne le considérez pas comme tel ! Si vous souhaitez conserver votre liberté, il vous faut vous préparer à la défendre.

Notes

1. On peut rendre l'esprit de ce poème comme suit :
Si je ne suis rien pour moi-même, qui sera pour moi ?
Si je suis tout pour moi-même, que suis-je ?
Si ce n'est pas aujourd'hui, alors quand ?
2. En anglais, le « libre » de « logiciel libre » se dit « free ». Malheureusement, ce mot a une autre acception, indépendante et incorrecte ici, il signifie également « gratuit ». Cette ambiguïté a causé énormément de tort au mouvement du logiciel libre.
3. En anglais, le placement des mots ne permet pas de déterminer s'il s'agit d'un « kit compilateur libre de l'université » ou d'un « kit compilateur de l'université libre ».
4. « couvert par le gauche d'auteur, tous droits renversés. ».
5. Le mot anglais bash a le sens de « coup, choc » et la signification de cet acronyme est double; c'est à la fois une nouvelle version de l'interpréteur de commandes Bourne, et une allusion aux chrétiens qui se sont sentis renaître dans cette religion, et qu'aux États-Unis d'Amérique on qualifie de born again Christians.
6. Littéralement, « [logiciel dont le] code source est ouvert ». C'est une périphrase lourde et inélégante en français, mais qui résout en anglais l'ambiguïté discutée plus haut, bien que l'auteur rejette cette solution, pour des raisons expliquées à la fin de cet article.
7. Encore et toujours cette ambiguïté de la langue anglaise. « software » signifie « logiciel ». « free » signifie à la fois « libre », sens qui est pertinent ici, et « gratuit », qualité qui n'est qu'un effet de bord des logiciels libres. « open source » signifie « dont le code source est ouvert ».

Copyright © 1998, 2001, 2002, 2005, 2006, 2007 Richard Stallman

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Sébastien Blondeel.

Révision : trad-gnu@april.org

Dernière mise à jour : 2008/09/14 08:27:41

Traductions de cette page

[Bosanski](#) [bs] [Català](#) [ca] [Česky](#) [cs] [Deutsch](#) [de] [Ελληνικά](#) [el] [English](#) [en] [Español](#) [es] [Suomi](#) [fi] [Français](#) [fr] [Bahasa Indonesia](#) [id] [Italiano](#) [it] [日本語](#) [ja] [한국어](#) [ko] [Nederlands](#) [nl] [Polski](#) [pl] [Русский](#) [ru] [简体中文](#) [zh-cn] [繁體中文](#) [zh-tw]

Le Manifeste GNU

Le manifeste GNU, reproduit ci-dessous, a été écrit par [Richard Stallman](#) au commencement du projet GNU pour encourager la participation et le soutien de tous. Au cours des premières années, il y a eu quelques petites mises à jour pour tenir compte des développements, mais il nous semble maintenant plus judicieux de le laisser en l'état.

Nous avons appris depuis qu'il y avait quelques incompréhensions; celles-ci peuvent être corrigées en changeant quelques mots. Des notes de bas de page ajoutées en 1993 aident à clarifier ces points.

Pour les dernières informations sur les logiciels GNU actuellement disponibles, veuillez vous référer à notre [site web](#), en particulier notre [liste de logiciels](#). Pour savoir comment contribuer, consultez <http://www.gnu.org/help/help.fr.html>.

Qu'est ce que GNU ? Gnu N'est pas Unix !

GNU, l'acronyme de GNU's Not Unix (GNU n'est pas Unix), est le nom du système complet de logiciels compatible Unix que j'écris pour pouvoir le donner librement à tous ceux qui en auraient besoin. ¹ De nombreux bénévoles m'aident. Les contributions en temps, en argent, en logiciels et en matériel sont les bienvenues.

Pour l'instant, nous avons un éditeur de texte, Emacs, utilisant le Lisp pour écrire des commandes d'édition, un débogueur, un générateur d'analyseurs syntaxiques compatible avec YACC, un éditeur de liens, et environ trente-cinq autres utilitaires. Un shell (un interprète de commandes) est presque terminé. Un nouveau compilateur C portable et optimisé s'est compilé lui-même et devrait être disponible cette année. Un premier noyau existe, mais nécessite plus de fonctions pour émuler Unix. Quand le noyau et le compilateur seront terminés, il sera possible de distribuer un système GNU propice au développement. Nous utiliserons TeX comme formateur de textes, mais un nroff est en cours de développement. Nous utiliserons aussi le système libre et portable X Window System. Par la suite, nous ajouterons un Common Lisp portable, le jeu Empire, un tableur et des centaines d'autres choses, plus une documentation en ligne. Nous espérons fournir, finalement, tout ce qui peut être utile et qui est normalement inclus dans un système Unix et plus encore.

GNU pourra exécuter des programmes Unix mais ne sera pas identique à Unix. Nous ferons toutes les améliorations que nous jugerons appropriées, en nous fondant sur nos expériences avec d'autres systèmes d'exploitation. En particulier, nous prévoyons d'avoir des fichiers avec des noms longs, des numéros de version de fichier, un système de fichiers à tolérance de panne, éventuellement un système de complétion des noms de fichiers, un dispositif d'affichage indépendant du terminal, et peut-être, finalement, un système de fenêtrage fondé sur Lisp, au travers duquel plusieurs programmes Lisp et autres programmes Unix pourront partager un écran. Le C et le Lisp seront tous les deux disponibles comme langages de programmation système. Nous essayerons de supporter UUCP, MIT Chaosnet, et les protocoles de l'Internet pour la communication.

Initialement, GNU vise les machines de classe 68000/16000 avec de la mémoire virtuelle, car ce sont les machines les plus simples pour le faire fonctionner. Nous laissons l'effort supplémentaire pour l'adapter sur de plus faibles machines à ceux qui voudront l'utiliser sur celles-ci.

Pour éviter d'horribles confusions, veuillez bien prononcer le « G » de « GNU » quand vous parlez de ce projet.

Pourquoi dois-je écrire GNU

J'estime que la Règle d'or est que, si j'aime un programme, je dois le partager avec d'autres qui aiment ce programme. Les éditeurs de logiciels cherchent à diviser et à conquérir les utilisateurs, en interdisant à chacun de partager avec les autres. Je refuse de rompre la solidarité avec les autres utilisateurs de cette manière. Je ne peux pas, en mon âme et conscience, signer un accord de non-divulgence ou une licence de logiciels. Pendant des années, j'ai œuvré au sein du Laboratoire d'Intelligence Artificielle du MIT pour résister à ces tendances, mais finalement, ils sont allés trop loin : je ne pouvais pas rester dans une institution où de telles choses avaient lieu contre ma volonté.

Pour pouvoir continuer à utiliser les ordinateurs en accord avec ma conscience, j'ai décidé de rassembler un ensemble suffisant de logiciels libres, pour pouvoir me débrouiller sans logiciels non libres. J'ai démissionné du laboratoire d'Intelligence Artificielle pour que le MIT ne puisse invoquer toutes les excuses légales pour m'empêcher de distribuer GNU librement.

Pourquoi GNU sera compatible avec Unix

Unix n'est pas pour moi le système parfait, mais il n'est pas trop mauvais. Les fonctions essentielles d'Unix semblent être les bonnes, et je pense pouvoir compléter ce qui manque à Unix sans les gâcher. Et un système compatible Unix serait commode à adopter par de nombreuses personnes.

Disponibilité de GNU

GNU n'est pas dans le domaine public. Tout le monde aura le droit de modifier et redistribuer GNU, mais aucun distributeur ne pourra restreindre ces futures redistributions. C'est-à-dire que des modifications propriétaires seront interdites. Je veux être sûr que toutes les versions de GNU restent libres.

Pourquoi beaucoup de programmeurs veulent contribuer

J'ai rencontré beaucoup de programmeurs enthousiasmés par GNU et qui souhaitent contribuer.

De nombreux programmeurs sont insatisfaits de la commercialisation de logiciels systèmes. Il se peut que cela leur permette de gagner plus d'argent, mais cela les amène forcément à se sentir en conflit avec les autres programmeurs en général, plutôt que d'être camarades. L'acte fondamental d'une amitié entre des programmeurs est le partage des programmes ; les arrangements commerciaux typiquement utilisés de nos jours interdisent aux programmeurs de considérer les autres comme des amis. L'acheteur de programmes doit choisir entre l'amitié et l'obéissance à la loi. Naturellement, un grand nombre décident que l'amitié est plus importante. Mais ceux qui respectent la loi se sentent souvent mal à l'aise face à ce seul choix. Ils sont désabusés et pensent que programmer n'est qu'une façon de gagner de l'argent.

En utilisant GNU plutôt que des programmes propriétaires, nous pouvons être amicaux envers tout le monde tout en respectant la loi. De plus, GNU est une source d'inspiration et une bannière sous laquelle d'autres peuvent nous rejoindre dans le partage. Ceci peut nous procurer un sentiment d'harmonie, impossible à atteindre avec des logiciels qui ne sont pas libres. Pour environ la moitié des programmeurs avec lesquels j'ai discuté, c'est une satisfaction importante que l'argent ne peut pas remplacer.

Comment vous pouvez contribuer

(Actuellement, pour la liste des tâches logicielles sur lesquelles travailler, consultez [la liste des tâches GNU](#). Pour d'autres manières de contribuer, consultez <http://www.gnu.org/help/help.fr.html>.)

Je demande aux fabricants d'ordinateurs de faire don de machines et d'argent. Je demande aux individus de faire don de programmes et de travail.

Une des conséquences à laquelle vous pouvez vous attendre si vous donnez des machines, c'est que GNU tournera dessus très rapidement. Les machines doivent être complètes, prêtes à l'utilisation, sans besoin de système particulier de climatisation ou d'alimentation.

J'ai trouvé de nombreux programmeurs impatientes de contribuer à mi-temps pour GNU. Pour la plupart des projets, un tel travail distribué à temps partiel serait très difficile à coordonner; les diverses parties codées indépendamment ne fonctionneraient pas ensemble. Mais ce problème n'existe pas dans le cas du projet de remplacement d'Unix. Un système Unix complet contient des centaines d'utilitaires, chacun étant documenté séparément. La plupart des spécifications des interfaces sont déterminées par la compatibilité avec Unix. Si chaque collaborateur peut écrire un remplacement compatible pour un seul utilitaire Unix et l'intégrer proprement à la place de l'original sur un système Unix, il s'ensuit que ces utilitaires fonctionneront ensemble sans problème. Même en faisant quelques concessions à la loi de Murphy qui créera quelques

problèmes inattendus, l'assemblage de ces composants sera une tâche réalisable. (Le noyau demandera quand même une communication plus soutenue et sera réalisé par un petit groupe).

Si je reçois des dons financiers, je pourrais embaucher quelques personnes à temps plein ou à mi-temps. Le salaire ne sera peut-être pas très élevé par rapport au marché, mais je cherche des personnes pour lesquelles l'esprit de communauté est aussi important que l'appât du gain. Je considère que c'est une façon de permettre à quelques personnes dévouées de consacrer toutes leurs ressources au projet GNU, en leur évitant d'avoir à gagner leur vie autrement.

Pourquoi tous les utilisateurs en bénéficieront

Une fois GNU achevé, tout le monde pourra obtenir de bons logiciels libres comme l'air ².

Ceci représente beaucoup plus que l'économie d'une licence Unix. Cela veut dire que l'on va éviter la duplication inutile du travail de programmation. Cet effort pourra plutôt se diriger vers l'avancement du domaine informatique.

Les sources du système complet seront disponibles pour tous. Et cela aura pour résultat qu'un utilisateur qui a besoin de changer un composant du système aura toujours la liberté d'effectuer des changements lui-même, ou d'engager une personne ou une société capable d'effectuer ces changements pour lui. Les utilisateurs ne seront plus à la merci d'une seule personne ou d'une seule société qui possède les sources du programme et qui est la seule à pouvoir effectuer des changements.

Les écoles pourront fournir un milieu beaucoup plus éducatif en encourageant chaque étudiant à étudier et à améliorer le code du système. Le laboratoire informatique d'Harvard avait comme politique de n'installer aucun programme sur le système si ses sources n'étaient pas disponibles, et ils soutenaient cette politique en refusant carrément d'installer certains programmes. Cela m'a beaucoup inspiré.

Enfin, les frais engendrés par les questions d'appartenance et de droits des logiciels ne seront plus d'actualité.

Les mesures pour faire payer les licences des programmes et de leurs copies génèrent toujours un coût important pour la société en général, à cause des mécanismes nécessaires pour calculer combien (c'est-à-dire quels programmes) chacun doit payer. Et il faudrait un État policier pour appliquer parfaitement ces mesures. Prenons une station orbitale, où l'air doit être fabriqué à un coût important: facturer chaque litre inspiré peut être justifié, mais porter un masque/compteur toute la journée et toute la nuit est intolérable même si on a de quoi payer la facture. Et les caméras de surveillance placées partout pour vérifier que vous ne retirez jamais le masque/compteur seraient inacceptables. Il vaut mieux financer la fabrication de l'air avec une taxe par personne et se débarrasser des masques.

De copier tout ou des parties d'un logiciel semble aussi naturel à un programmeur que de respirer, tout aussi productif. Cela aussi devrait être libre.

Quelques objections facilement contrées aux objectifs de GNU

« Personne ne s'en servira si c'est gratuit, car cela veut dire que l'on ne peut compter sur aucun support. »

« Il faut faire payer le logiciel pour financer le service après-vente. »

Il y a des gens qui préfèrent payer pour GNU et le service plutôt que d'obtenir GNU sans service. Une société qui propose le service uniquement à ceux qui ont obtenu GNU gratuitement, devrait être rentable ³.

Nous devons faire la distinction entre le support en termes de réel travail de programmation et le simple support d'assistance. On ne peut pas compter sur le premier de la part d'un simple revendeur. Si votre problème n'est pas suffisamment répandu, le revendeur vous enverra balader.

Si votre société a besoin d'un support fiable, la seule façon est d'avoir toutes les sources et tous les outils nécessaires. À partir de là, vous pouvez engager n'importe quelle personne qualifiée pour régler les problèmes. Vous n'êtes pas à la merci d'une seule personne. Avec Unix, le prix des sources rend cette solution

inabordable pour la plupart des sociétés. Avec GNU ce sera facile. Il serait éventuellement concevable qu'il n'y ait personne de disponible, mais les modalités de distribution ne sont pas responsables de ce problème. GNU ne propose pas de régler tous les problèmes, mais seulement quelques-uns.

Pendant ce temps, les utilisateurs qui n'y connaissent rien en informatique ont besoin d'assistance et de personnes pour les aider à faire ce qu'ils pourraient faire très bien eux-mêmes si seulement ils s'y connaissaient.

De tels services pourraient être proposés par des sociétés qui ne font que des cours d'initiation et des réparations. S'il est vrai que les utilisateurs préfèrent dépenser de l'argent pour un logiciel intégrant un service après-vente, ils seront aussi d'accord pour payer simplement le service, ayant obtenu le logiciel gratuitement. Les sociétés de service se feront concurrence sur la qualité et le prix de leurs prestations. Les utilisateurs ne seront pas limités à une société particulière. En même temps, ceux d'entre nous qui n'ont pas besoin du service pourront utiliser le logiciel sans payer le service.

« On ne peut pas être connu sans publicité et il faut payer le logiciel pour financer la publicité. »

« Ca ne sert à rien de faire de la publicité si on peut obtenir le logiciel gratuitement. »

Il existe plusieurs formes de publicité gratuite ou bon marché qui peuvent être utilisées pour informer de nombreux utilisateurs au sujet de GNU. Cependant, il est peut-être vrai que l'on peut atteindre plus d'utilisateurs avec de la publicité. Si cela est vrai, une société qui fait de la publicité sur le service payant de copie et de distribution de GNU doit être suffisamment rentable pour assurer sa propre publicité et bien davantage. Ainsi, seuls les utilisateurs qui bénéficient de la publicité la payent.

En revanche, si de nombreuses personnes obtiennent GNU par leurs relations, de telles sociétés ne seraient pas rentables, et cela démontrerait que la publicité n'était pas vraiment nécessaire pour répandre GNU. Pourquoi est-ce que les partisans du libre échange ne veulent pas laisser cette décision au marché libre [4](#) ?

« Ma société a besoin d'un système propriétaire pour être compétitive. »

GNU va retirer les systèmes d'exploitation du domaine de la concurrence. Vous ne pourrez pas être avantagé dans ce domaine, mais votre concurrent non plus. Vous pourrez rivaliser dans d'autres domaines. Si votre domaine est la vente de systèmes d'exploitation, vous n'aimerez pas GNU, et c'est tant pis pour vous. Si votre domaine est différent, GNU peut vous éviter d'être poussé dans le domaine onéreux de la vente de systèmes d'exploitation.

J'aimerais bien voir le développement de GNU financé par des dons de fabricants et utilisateurs, réduisant ainsi les coûts pour chacun [5](#).

« Les programmeurs ne méritent-ils pas d'être récompensés pour leur créativité ? »

Si quelque chose mérite une récompense, c'est bien la contribution sociale. La créativité peut être une contribution sociale, mais seulement tant que la société est libre de profiter des résultats. Si les programmeurs méritent d'être récompensés pour la création de logiciels innovants, de même, ils méritent d'être punis s'ils limitent l'utilisation de leurs programmes.

« Un programmeur ne devrait-il pas avoir le droit de demander une récompense pour sa créativité ? »

Il n'y a rien de mal à vouloir être payé pour son travail, ou à chercher à augmenter ses revenus, tant que l'on n'utilise pas de méthodes destructives. Mais les méthodes pratiquées dans le domaine du logiciel sont fondées sur la destruction.

Extraire de l'argent aux utilisateurs d'un programme en limitant son utilisation est destructeur, car ces restrictions réduisent l'utilité du programme. Ce qui à son tour réduit la richesse apportée par ce programme à l'humanité. Quand le choix de limiter est délibéré, les conséquences néfastes qui en découlent sont de la destruction délibérée.

La raison pour laquelle un bon citoyen ne doit pas utiliser de telles méthodes destructrices pour augmenter sa richesse personnelle est que si tout le monde faisait de même, il y aurait un appauvrissement général dû à la destruction mutuelle. C'est ce que l'on appelle la morale kantienne, ou la Règle d'or. Puisque

je n'apprécie pas les conséquences qui adviennent si tout le monde fait de la rétention d'informations, je ne dois pas trouver acceptable qu'un individu le fasse. Plus précisément, le désir d'être récompensé pour sa création ne justifie pas que l'on prive le monde en général de toute ou partie de cette créativité.

« Les programmeurs ne vont-ils pas mourir de faim ? »

Je peux répondre que personne n'est forcé d'être un programmeur. La plupart d'entre nous n'arriverait pas à se faire payer pour faire des grimaces dans la rue. Mais nous ne sommes pas pour autant condamnés à passer notre vie dans la rue à faire des grimaces et à mourir de faim. On fait autre chose.

Mais c'est une mauvaise réponse, car elle accepte a priori de la question : c'est-à-dire que sans possession du logiciel, les programmeurs ne pourraient pas recevoir le moindre sou. C'est, soi-disant, tout ou rien.

La vraie raison pour laquelle les programmeurs ne vont pas mourir de faim est qu'il sera quand même possible pour eux d'être payés pour programmer; seulement, peut-être pas autant qu'actuellement.

La restriction des copies n'est pas la seule base des affaires du domaine des logiciels. C'est la base la plus commune, car c'est la plus rentable. Si ces restrictions étaient interdites ou rejetées par le client, les éditeurs passeraient à d'autres formes d'organisation, qui sont actuellement moins utilisées. Il y a de nombreuses façons d'organiser une entreprise.

Il est probable qu'avec ce nouveau système, la programmation serait moins rentable qu'elle ne l'est actuellement. Mais ce n'est pas un argument valable contre le changement. Il n'est pas considéré comme injuste que les caissières gagnent ce qu'elles gagnent. Si les programmeurs gagnaient la même chose, ce ne serait pas non plus une injustice. (En pratique, ils gagneraient quand même beaucoup plus).

« Les gens n'ont-ils pas le droit de gérer l'utilisation de leur créativité ? »

« Contrôler l'utilisation que l'on fait de ses idées » revient à contrôler la vie des autres; et c'est souvent utilisé pour leur rendre la vie plus difficile.

Ceux qui ont étudié le problème de la propriété intellectuelle ⁶ à fond (les avocats, les juristes, etc.) soutiennent qu'il n'existe aucun droit intrinsèque à la propriété intellectuelle. Les différents droits de soi-disant propriété intellectuelle reconnus par le gouvernement ont été créés par des législations précises dans des buts bien précis.

Par exemple, le système de brevets a été établi pour encourager les inventeurs à divulguer les détails de leurs inventions. Sa raison d'être était d'aider la société plutôt que d'avantager les inventeurs. À l'époque, la durée de vie de 17 ans pour un brevet était court par rapport à la cadence des évolutions technologiques. Puisque les brevets ne concernent que les fabricants, pour lesquels le coût et l'effort d'établir une licence sont minimes comparés à la mise en production, les brevets ne font souvent pas trop de tort. Ils ne gênent pas la plupart des individus qui utilisent des produits brevetés.

Le concept de droit d'auteur n'existait pas dans l'Antiquité, les auteurs copiaient souvent, et beaucoup, l'œuvre des autres. Cette pratique était utile, et c'est de cette seule façon que les travaux de certains auteurs ont survécus ne serait-ce qu'en partie. Le système du droit d'auteur a été créé expressément pour encourager les auteurs. Dans le domaine pour lequel ce système a été inventé, les livres, qui pouvaient seulement être copiés en imprimerie ne causait pas beaucoup de tort, et ne gênait pas la plupart des personnes qui lisaient ces livres.

Tous les droits de propriété intellectuelle ne sont que des licences accordées par la société parce que nous pensions, à tort ou à raison, que la société en général bénéficierait de ces accords. Mais dans chaque situation précise, nous devons nous demander : bénéficierons-nous vraiment d'accorder cette licence ? Quels actes autorisons-nous avec cette licence ?

Le cas des logiciels aujourd'hui est très différent de celui des livres il y a un siècle. Le fait que la manière la plus répandue de copier un programme est entre voisins, le fait qu'un programme contient à la fois du code source et du code binaire bien distinct, et le fait qu'un programme est utilisé plutôt que lu comme divertissement, se réunissent pour créer une situation dans laquelle celui qui applique le droit d'auteur fait du tort à la société, matériellement et spirituellement; cette personne ne devrait pas appliquer le droit d'auteur,

que la loi l'y autorise ou non.

« La compétition permet de mieux faire les choses. »

Le paradigme de la compétition est une course : en récompensant le vainqueur, nous encourageons tout le monde à courir plus vite. Quand le capitalisme fonctionne réellement de cette façon, tout marche bien; mais ses partisans ont tort s'ils pensent que cela fonctionne toujours de cette façon. Si les coureurs oublient le pourquoi de la récompense, et deviennent obsédés par la victoire, quelles que soient les méthodes employées, ils risquent de trouver d'autres stratégies telles qu'agresser les autres concurrents. Si tous les coureurs s'engageaient dans un combat, ils finiraient tous en retard.

Les logiciels propriétaires et secrets sont l'équivalent moral des coureurs qui se battent. Malheureusement, le seul arbitre que l'on ait ne semble pas s'opposer aux combats ; il se contente de les réguler (« Pour dix mètres parcourus, vous avez le droit de tirer un coup de feu »). Il devrait en fait séparer les combattants, et punir les coureurs qui tentent de se battre.

« Les gens s'arrêteront-ils de programmer sans l'appât du gain ? »

En fait, beaucoup de gens programmeront même sans aucun bénéfice financier. La programmation exerce une fascination irrésistible pour quelques-uns, généralement ceux qui programment le mieux. Il n'y a aucune pénurie de musiciens professionnels qui continuent à jouer, même sans l'espoir de pouvoir en faire leur gagne-pain.

Mais en fait cette question, bien qu'elle soit souvent posée, ne convient pas à la situation. Les salaires des programmeurs ne disparaîtront pas mais diminueront peut-être. La question devient donc, trouvera-t-on des programmeurs qui travailleront pour une moindre rémunération ? D'après mon expérience, la réponse est oui.

Pendant plus de dix ans, plusieurs des meilleurs programmeurs du monde ont travaillé au laboratoire d'Intelligence Artificielle du MIT pour un salaire bien moins important que ce qu'ils auraient touchés ailleurs. Ils étaient récompensés de plusieurs autres manières: la notoriété, le respect des autres, par exemple. Et la créativité est une récompense en soi.

Et puis la plupart sont partis pour faire le même travail pour beaucoup plus d'argent.

Les faits démontrent que les gens programmeront pour d'autres raisons que l'accumulation de richesses; mais si on leur propose beaucoup plus d'argent, ils s'y attendront finalement et l'exigeront. Les organismes qui payent moins bien ont du mal face à ceux qui payent bien, mais ils devraient pouvoir s'en sortir si les gros payeurs sont bannis.

« Nos besoins en programmeurs sont tellement importants que s'ils interdisent le partage, nous ne pouvons que leur obéir. »

La situation n'est jamais aussi désespérée au point d'être amené à obéir à une telle interdiction.

« Les programmeurs doivent bien gagner leur pain. »

À court terme, cela est vrai. Cependant, il y a de nombreuses possibilités offertes à un programmeur pour vivre décemment sans pour autant vendre le droit d'utiliser un programme. Cette façon est la plus répandue actuellement, car c'est celle qui engendre le plus de profit pour les programmeurs et les hommes d'affaires, et non parce que c'est la seule manière de gagner son pain. Vous pouvez facilement trouver d'autres manières si vous le voulez. Voici quelques exemples.

Un fabricant arrivant avec un nouvel ordinateur payera pour le portage des systèmes d'exploitation sur le nouveau matériel.

L'offre de services d'enseignement, de conseil et de maintenance peut permettre la création d'emplois.

Les personnes avec des idées nouvelles peuvent distribuer des logiciels librement ⁷, en demandant des dons aux utilisateurs satisfaits ou en offrant un service de conseil. J'ai déjà rencontré des personnes travaillant ainsi.

Les utilisateurs ayant des besoins en commun, peuvent créer des groupes d'utilisateurs et payer des

cotisations. Un tel groupe pourrait faire appel à une société de développement pour écrire les programmes spécifiques pour ses membres.

Toutes sortes de développement pourraient être financés par une taxe sur les logiciels :

Supposons que chaque personne qui achète un ordinateur doit payer x pour cent du prix en tant que taxe sur les logiciels. Le gouvernement reverserait cette somme à un organisme tel que la NSF pour subvenir au développement de logiciels

Mais si l'acheteur fait lui-même un don au développement de logiciels, il pourra être crédité pour cette taxe. Il pourrait donner au projet de son choix, car il espérera profiter des résultats à l'achèvement du projet. Il pourra donc être exempté de la taxe si le montant de sa donation recouvre celle-ci.

Le taux de la taxe pourrait être déterminé par un vote de ceux qui la payent, pondéré par le montant de l'imposition.

Les conséquences :

- La communauté des utilisateurs soutient le développement des logiciels.
- Cette communauté décide du niveau du soutien nécessaire.
- Pour les utilisateurs qui se soucient de quels projets profitent de leur participation, ils pourront les choisir eux-mêmes.

À terme, rendre les programmes libres est un pas vers le monde d'après pénurie, quand personne ne devra travailler très dur juste pour survivre. Les gens seront libres de se consacrer à des activités ludiques telles que la programmation, après avoir, bien entendu, passé les dix heures par semaine nécessaires pour des œuvres telles que la législation, la thérapie de famille, la réparation de robots et l'exploration d'astéroïdes. Il n'y aura donc plus besoin de gagner sa vie en programmant.

Nous avons déjà beaucoup réduit la quantité de travail que la société entière doit fournir pour sa productivité, mais seulement une petite part se traduit en temps de loisirs pour les travailleurs, car beaucoup d'activités non productives sont nécessaires pour accompagner l'activité productive. Les raisons principales sont la bureaucratie et la lutte isométrique contre la concurrence. Le logiciel libre va réduire grandement ces fuites du domaine du développement logiciel. Nous devons faire cela, pour que les gains de productivité se traduisent en moins d'heures de travail pour nous.

Notes

1. Le choix des mots ici était irréflecté. L'idée était que personne n'aurait à payer l'autorisation d'utiliser le système GNU. Mais cela n'était pas clair, et les gens ont souvent compris que les copies de GNU devaient toujours être distribuées peu chères ou gratuitement. Cela n'a jamais été l'intention; plus tard, le manifeste mentionne pour les sociétés la possibilité de fournir un service de distribution rentable. Par la suite, j'ai appris à bien faire la distinction entre « free » dans le sens de libre, et « free » dans le sens de gratuit. Le logiciel libre est un programme que les utilisateurs ont la liberté de distribuer et de modifier. Certains utilisateurs peuvent obtenir des copies gratuitement, tandis que d'autres les paieront et si cela peut rapporter de quoi financer l'amélioration de programmes, tant mieux. L'important est que toute personne détenant une copie a le droit de l'utiliser en collaboration avec d'autres.
2. Voilà un autre endroit où je n'ai pas fait la distinction entre les deux définitions de « libre ». La phrase telle quelle n'est pas fautive, vous pouvez obtenir des copies de logiciels GNU gratuitement, par vos amis ou par l'Internet. Mais elle suggère effectivement la mauvaise idée. [NdT : En français, la distinction entre « libre » et « gratuit » est évidente.]
3. De telles sociétés existent actuellement.
4. Depuis dix ans, la Free Software Foundation a levé la plupart de ses fonds à travers un service de distribution, bien qu'elle soit une association plutôt qu'une société. Vous pouvez [commander des logiciels, des livres, ... à la FSF](#).
5. Un groupe de sociétés a réuni des fonds vers 1991 pour financer la maintenance du compilateur C de GNU.
6. Dans les années 80, je n'avais pas encore réalisé à quel point il était déroutant de parler du « problème » de la « propriété intellectuelle ». Ce terme est évidemment partial; plus subtil dans le fait qu'il mélange diverses lois disparates qui traitent de problèmes très différents. De nos jours, je presse les gens à rejeter totalement le terme « propriété intellectuelle », de peur qu'il ne conduise d'autres personnes à supposer que ces lois forment une solution cohérente. La façon d'être clair est de parler de brevets, de droits d'auteur, et de marques déposées séparément. Voir [des explications plus détaillées](#) sur la manière dont ce terme sème la confusion et le parti-pris.
7. Par la suite, nous avons appris la distinction entre « logiciel libre » et « graticiel » (« freeware »). Le terme « graticiel » signifie logiciel que vous êtes libre de redistribuer, mais généralement, vous n'êtes pas libre de l'étudier ou de changer le code source; donc la plupart d'entre eux ne sont pas des logiciels libres. Voir la page [Termes prêtant à confusion, que vous devriez éviter](#) pour plus d'explications.

Copyright © 1985, 1993, 2003, 2005, 2007, 2008 Free Software Foundation, Inc.,

51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée et que le distributeur donne au destinataire la permission pour d'autre redistribution ainsi que permis par cette notice.

Il n'est pas permis de faire de versions modifiées de ce document.

Traduction : ?.

Révision : trad-gnu@april.org

Dernière mise à jour : Date: 2008/11/27 21:26:44

Traductions de cette page

[Català](#) [ca] [Česky](#) [cs] [Deutsch](#) [de] [ελληνικά](#) [el] [English](#) [en] [Español](#) [es] [Français](#) [fr] [עברית](#) [he] [Hrvatski](#) [hr] [Italiano](#) [it]
[한국어](#) [ko] [Македонски](#) [mk] [Nederlands](#) [nl] [Polski](#) [pl] [português do Brasil](#) [pt-br] [Српски](#) [sr] [Svenska](#) [sv] [தமிழ்](#) [ta]

Définition d'un logiciel libre

Nous maintenons cette définition du logiciel libre pour décrire clairement les conditions à remplir pour qu'un logiciel soit considéré comme libre.

L'expression « Logiciel libre » fait référence à la liberté et non pas au prix. Pour comprendre le concept, vous devez penser à la « liberté d'expression », pas à « l'entrée libre ».

L'expression « Logiciel libre » fait référence à la liberté pour les utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Plus précisément, elle fait référence à quatre types de liberté pour l'utilisateur du logiciel :

- La liberté d'exécuter le programme, pour tous les usages (liberté 0).
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins (liberté 1). Pour ceci l'accès au code source est une condition requise.
- La liberté de redistribuer des copies, donc d'aider votre voisin, (liberté 2).
- La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté (liberté 3). Pour ceci l'accès au code source est une condition requise.

Un programme est un logiciel libre si les utilisateurs ont toutes ces libertés. Ainsi, vous êtes libre de redistribuer des copies, avec ou sans modification, gratuitement ou non, à tout le monde, partout. Être libre de faire ceci signifie (entre autre) que vous n'avez pas à demander ou à payer pour en avoir la permission.

Vous devez aussi avoir la liberté de faire des modifications et de les utiliser à titre privé dans votre travail ou vos loisirs, sans en mentionner l'existence. Si vous publiez vos modifications, vous n'êtes pas obligé de prévenir quelqu'un de particulier ou de le faire d'une manière particulière.

La liberté d'utiliser un programme est la liberté pour tout type de personne ou d'organisation de l'utiliser pour tout type de système informatique, pour tout type de tâche et sans être obligé de communiquer à ce sujet avec le développeur ou tout autre entité spécifique. Dans cette liberté, il est question de *l'utilisateur*, pas du développeur ; en tant qu'utilisateur, vous êtes libre d'exécuter un programme comme bon vous semble et si vous le redistribuez à quelqu'un d'autre, cette personne est libre de l'exécuter comme bon lui semble, mais vous n'êtes pas autorisé à lui imposer vos conditions.

La liberté de redistribuer des copies doit inclure les formes binaires ou exécutables du programme (tout comme le code source) à la fois pour les versions modifiées ou non modifiées du programme. (Diffuser des programmes sous une forme exécutable est nécessaire pour une installation commode des systèmes d'exploitation libres). Il y a une exception s'il n'y a pas moyen de produire une version binaire ou exécutable (puisque certains langages ne supportent pas cette caractéristique), mais le public doit avoir la liberté de distribuer de telles formes s'ils ont un moyen d'en produire.

Pour avoir la liberté d'effectuer des modifications et de publier des versions améliorées, vous devez avoir l'accès au code source du programme. Par conséquent, l'accessibilité du code source est une condition requise pour un logiciel libre.

Un moyen important de modifier un programme est de le fusionner avec des modules ou des sous-programmes disponibles libres. Si la licence du programme indique que vous ne pouvez pas le fusionner dans un module existant, telle que la nécessité que vous soyez le détenteur du copyright de tout code que vous ajoutez, alors la licence est trop restrictive pour être qualifiée de libre.

Pour que ces libertés soient réelles, elles doivent être irrévocables tant que vous n'avez rien fait de mal; si le développeur du logiciel a le droit de révoquer la licence sans que vous n'ayez fait quoi que ce soit pour le justifier, le logiciel n'est pas libre.

Cependant, certains types de règles sur la manière de distribuer le logiciel libre sont acceptables tant que ces règles ne rentrent pas en conflit avec les libertés fondamentales. Par exemple, le copyleft (pour résumer très simplement) est une règle qui établit que lorsque vous redistribuez les programmes, vous ne pouvez ajouter de restrictions pour retirer les libertés fondamentales au public. Cette règle ne rentre pas en conflit avec les libertés fondamentales ; en fait, elle les protège.

« Logiciel libre » ne signifie pas « non commercial ». Un logiciel libre doit être disponible pour un usage commercial, pour 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 indifféremment de la manière dont vous vous l'êtes procuré, vous avez toujours la liberté de copier et de modifier un logiciel et même d'en [vendre des copies](#).

Qu'un changement constitue une amélioration ou pas est subjectif. Si vos modifications sont limitées, en substance, aux changements que quelqu'un d'autre considère comme une amélioration, ce n'est pas la liberté.

Les règles sur la manière d'emballer une version modifiée sont acceptables si elles n'entravent pas votre liberté de la publier, ou votre liberté de faire et d'utiliser pour votre usage personnel des versions modifiées. Les règles disant « si vous publiez le programme par ce moyen, vous devez le faire par ce moyen aussi » sont acceptables aux mêmes conditions (notez que de telles règles doivent vous laisser le choix de publier ou non le programme). Les règles qui nécessitent que le code source soit publié pour les utilisateurs pour les versions que vous rendez publiques sont aussi acceptables. Il est également acceptable que la licence l'exige, si vous avez distribué une version modifiée et qu'un développeur précédent vous en demande une copie, vous devez lui envoyer, ou que vous indiquiez vos modifications.

Dans le projet GNU, nous utilisons le « [copyleft](#) » pour protéger ces libertés. Mais des [logiciels libres non-copyleftés](#) existent aussi. Nous croyons qu'il y a de bonnes raisons qui font [qu'il est mieux d'utiliser le copyleft](#), mais si votre programme est libre non-copylefté, nous pouvons tout de même l'utiliser.

Lisez [Les catégories de Logiciel Libre](#), où sont décrites les relations entre « logiciel libre », « logiciel copylefté » et les autres catégories de logiciel.

Parfois le contrôle gouvernemental des exportations ou des sanctions économiques peuvent vous priver de la liberté de distribuer des copies de programmes à l'étranger. Les développeurs de logiciels n'ont pas le pouvoir d'éliminer ou de passer outre ces restrictions, mais ce qu'ils peuvent et doivent faire, est de refuser d'imposer eux-mêmes des conditions à l'utilisation des programmes. De cette manière, les restrictions n'affecteront pas les activités et les personnes se trouvant hors de la juridiction de leurs gouvernements.

La plupart des licences de logiciels sont basées sur le droit d'auteur, or les types d'exigences que le droit d'auteur peut imposer ont des limites. Si une licence basée sur le droit d'auteur respecte la liberté de la manière décrite plus haut, il est improbable que nous ayons une autre sorte de problème que nous n'ayons pas anticipé (bien que cela arrive parfois). Cependant, certaines licences de logiciels sont basées sur le droit du contrat et les contrats impliquent un champ bien plus large de restrictions. Cela signifie qu'il y a bien plus de possibilités pour qu'une licence de ce type puisse restreindre de manière inacceptable la liberté des utilisateurs et ainsi devenir non libre.

Nous ne pouvons pas lister tout ce qui pourrait se passer. Si une licence basée sur le contrat restreint l'utilisateur d'une manière inhabituelle que les licences basées sur le copyright ne peuvent pas faire et qui n'est pas mentionnée ici comme légitime, nous devons y réfléchir et nous concluons probablement qu'elle n'est pas libre.

Quand vous parlez des logiciels libres, il est préférable de ne pas utiliser de termes comme « donner » ou « gratuit », car ils laissent supposer que la finalité des logiciels libres est le prix et non la liberté. Certains termes répandus comme « piratage » comportent des idées auxquelles nous espérons que vous n'adhérez pas. Lisez [Termes prêtant à confusion, que vous devriez éviter](#) pour un essai sur l'utilisation de ces termes. Nous avons aussi une liste de [traductions de « free software »](#) dans de nombreuses langues.

Enfin, notez que les critères tels que ceux développés dans cette définition du logiciel libre demandent une réflexion sérieuse quant à leur interprétation. Pour décider si une licence de logiciel particulière est définie comme libre, nous la jugeons sur ces critères pour déterminer si elle convient à leur esprit tout comme à leur formulation précise. Si une licence inclut des restrictions inacceptables, nous la rejetons même si nous n'avons pas anticipé le problème dans ces critères. Quelquefois les conditions d'une licence soulève un problème qui nécessite des réflexions intenses, y compris des discussions avec un juriste, avant que nous puissions décider si la condition est acceptable. Quand nous arrivons à une conclusion

concernant un nouveau problème, nous mettons souvent à jour ces critères pour rendre plus facile le fait de savoir si une licence est une licence logicielle libre ou non.

Si vous voulez savoir si une licence spécifique est définie comme « libre », reportez-vous à notre [liste de licences](#). Si la licence qui vous intéresse n'y est pas listée, vous pouvez nous demander des précisions en nous envoyant un mail à [<licensing@gnu.org>](mailto:licensing@gnu.org).

Si vous envisager d'écrire une nouvelle licence, veuillez contacter la FSF en écrivant à cette adresse. La prolifération de différentes licences de logiciels libres implique un travail grandissant pour les utilisateurs dans la compréhension des licences ; nous pouvons vous aider à trouver une licence existante de logiciel libre qui réponde à vos besoins.

Si ce n'est pas possible, si vous avez vraiment besoin d'une nouvelle licence, avec notre aide, vous pouvez vous assurer que la licence est vraiment une licence de logiciel libre et éviter divers problèmes pratiques.

Au-delà du logiciel

Les [manuels de logiciels doivent être libres](#), pour les mêmes raisons que les logiciels doivent être libres, et parce que les manuels font en effet partie des logiciels.

Les mêmes arguments peuvent aussi s'appliquer à d'autres types de travaux à usage pratique — c'est-à-dire, des travaux qui intègrent de la connaissance utile, tels que les ouvrages éducatifs et les ouvrages de référence. [Wikipedia](#) est le meilleur exemple connu.

Tout type d'œuvre *peut* être libre, et la définition du logiciel libre a été étendu à la définition des [œuvres culturelles libres](#) applicable à tout type d'œuvre.

Open Source ?

Un autre groupe a commencé à utiliser le terme « open source » pour exprimer quelque chose de proche (mais pas d'identique) au « logiciel libre ». Nous préférons le terme « logiciel libre » parce que, une fois que vous avez entendu que ce terme réfère à la liberté plutôt qu'au prix, il appelle à prêter attention à la liberté. Le mot « open » [ne rend pas compte de cela](#).

Histoire

De temps en temps, nous révisons la définition du logiciel libre pour la clarifier. Nous fournissons ici une liste de ces modifications, ainsi que les liens pour illustrer exactement ce qui a changé, de sorte que d'autres puissent l'examiner s'ils le désirent.

- [Version 1.77](#) : clarifie que tout changement rétro-actif de la licence est inacceptable, même si ce n'est pas décrit comme un remplacement complet.
- [Version 1.74](#) : Clarification de quatre points pas assez explicites, ou mentionnés dans divers endroits mais pas reportés partout :
 - « Améliorations » ne signifie pas que la licence peut de façon substantielle limiter le type de versions modifiées que vous pouvez publier. La liberté 3 comprend la distribution de versions modifiées, pas seulement des changements.
 - Le droit de fusionner dans des modules existants se réfère à ceux qui sont sous licence appropriée.
 - Déclare explicitement la conclusion sur le point parlant des contrôles à l'exportation.
 - Imposer un changement de licence constitue une révocation de l'ancienne licence.
- [Version 1.57](#) : Ajout de la section « Au-delà du logiciel ».
- [Version 1.46](#) : Clarification des buts significatifs dans la liberté d'exécuter un programme dans n'importe quel but.
- [Version 1.41](#) : Clarification de la formulation au sujet des licences basées sur des contrats.
- [Version 1.40](#) : Explication qu'une licence libre doit vous autoriser à utiliser d'autres logiciels libres disponibles pour créer vos modifications.

- [Version 1.39](#) : Note qu'il est acceptable pour une licence de nécessiter que vous produisiez les sources des versions de logiciels mis à disposition au public.
- [Version 1.31](#) : Note qu'il est acceptable pour une licence de nécessiter de vous identifier en tant qu'auteur de modifications. D'autres clarifications mineures dans le texte.
- [Version 1.23](#) : Adresse les problèmes potentiels relatifs aux licences basées sur des contrats.
- [Version 1.16](#) : Explique pourquoi la distribution de binaires est importante.
- [Version 1.11](#) : Note qu'une licence libre peut nécessiter que vous envoyiez une copie des versions que vous distribuez à l'auteur.

Il y a des espaces entre les numéros de versions car il y a eu beaucoup d'autres changements qui n'affectent pas du tout l'essence de la définition. Ils corrigent plutôt des liens, ajoutent des traductions, etc. Si vous voulez consulter la liste intégrale des changements, vous pouvez le faire sur l'[interface cvsweb](#).

Copyright © 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 Free Software Foundation, Inc.,

51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Karl Pradène.

Révision : trad-gnu@april.org

Dernière mise à jour : Date: 2009/01/02 21:27:31

Traductions de cette page

[Afrikaans](#) [af] [العربية](#) [ar] [Azərbaycanca](#) [az] [български](#) [bg] [বাংলা](#) [bn] [Bosanski](#) [bs] [Català](#) [ca] [Česky](#) [cs] [Dansk](#) [da] [Deutsch](#) [de] [Ελληνικά](#) [el] [English](#) [en] [Esperanto](#) [eo] [Español](#) [es] [فارسی](#) [fa] [Français](#) [fr] [Galego](#) [gl] [עברית](#) [he] [Hrvatski](#) [hr] [Magyar](#) [hu] [Bahasa Indonesia](#) [id] [Italiano](#) [it] [日本語](#) [ja] [한국어](#) [ko] [Norsk \(Bokmål\)](#) [nb] [Nederlands](#) [nl] [Polski](#) [pl] [português do Brasil](#) [pt-br] [Română](#) [ro] [Русский](#) [ru] [Slovensko](#) [sl] [Српски](#) [sr] [Svenska](#) [sv] [தமிழ்](#) [ta] [Tagalog](#) [tl] [Türkçe](#) [tr] [简体中文](#) [zh-cn] [繁體中文](#) [zh-tw]

Pourquoi les logiciels ne doivent pas avoir de propriétaire

par Richard Stallman

Les techniques numériques de l'information contribuent à l'intérêt général en rendant plus commodes la copie et la modification de l'information. Les ordinateurs apportent la promesse de faciliter ces opérations pour tous.

Tout le monde ne veut pas de cette simplification. Le système du droit de copie attribue aux programmes informatiques des « propriétaires », qui pour la plupart souhaitent en garder pour eux les bénéfices potentiels et non les ouvrir au public. Ils veulent être seuls à pouvoir copier et modifier les logiciels que nous utilisons.

Le système du droit de copie s'est développé en même temps que l'imprimerie, une technique de copie à grande échelle. Le droit de copie était adapté à cette technologie parce qu'il ne limitait que la copie à grande échelle. Il ne privait pas les lecteurs de livres de leurs libertés : le lecteur moyen ne possédait pas de presse à imprimer, et il lui arrivait de recopier des livres avec sa plume et son encrier. Les lecteurs ne se voyaient pas traînés devant les tribunaux parce qu'ils avaient ainsi recopié des livres.

Les techniques numériques sont plus souples que la presse d'imprimerie. Une fois sous forme numérique, il devient facile de recopier l'information pour en faire profiter d'autres personnes. Cette souplesse place le support numérique en porte-à-faux dans un système comme le droit de copie. C'est pour cette raison que de plus en plus souvent des mesures sévères et désagréables sont prises afin de renforcer le droit de copie pour les logiciels. Par exemple les quatre pratiques suivantes de l'Association des Éditeurs de Logiciels (Software Publishers Association, SPA) :

- Une propagande massive clamant qu'il est mal de désobéir aux propriétaires afin d'aider ses amis.
- Une incitation à la dénonciation de ses camarades ou de ses collègues.
- Des visites surprises, avec l'aide de la police, dans les bureaux et dans les écoles, au cours desquelles on exige des personnes la preuve qu'elles sont innocentes du délit de copie illégale.
- Des poursuites menées par le gouvernement américain à la demande de la SPA, à l'encontre de personnes comme David LaMacchia, du M.I.T., non pour avoir recopié un logiciel, mais simplement pour avoir laissé des moyens de copie sans surveillance et ne pas avoir réussi à en empêcher l'utilisation.

Ces pratiques rappellent toutes les quatre celles de l'ancienne Union Soviétique. Sous ce régime à présent aboli, chaque photocopieuse était gardée pour empêcher la copie interdite, et les individus étaient obligés de recopier l'information en secret et de la diffuser de la main à la main sous forme de « samizdats ». Évidemment, les motifs de cette restriction n'étaient pas les mêmes : en Union Soviétique ils étaient politiques, aux États-Unis c'est le profit. Mais ce sont les mesures qui nous affectent, et non pas les motifs. Toute tentative de blocage de la diffusion de l'information pour quelque raison que ce soit conduit aux mêmes méthodes et à la même brutalité.

Les propriétaires ont inventé divers arguments pour justifier leur prise de contrôle de la manière dont nous utilisons l'information :

- Les insultes.

Les propriétaires emploient des expressions péjoratives comme « pirate » ou « vol » en les associant à une terminologie plus technique comme « propriété intellectuelle » ou « préjudice ». Ils conduisent ainsi le public à penser comme ils le veulent, par une analogie simpliste entre les programmes d'ordinateurs et les objets du monde physique.

Nos idées et nos intuitions sur la propriété des objets matériels se rapportent à la question de savoir s'il est juste d'*emporter un objet* qui appartient à quelqu'un d'autre. Elles ne s'appliquent pas directement à la *recopie* de quelque chose. Mais les propriétaires nous demandent de les appliquer quand même.

- L'exagération.

Les propriétaires disent subir des « dommages » ou des « pertes économiques » du fait que les utilisateurs recopient eux-mêmes les programmes. Pourtant le fait de la copie n'a aucun effet direct pour le propriétaire et ne fait de mal à personne. Le propriétaire ne subit une perte que dans la mesure où la personne qui fait cette copie aurait été prête à payer au propriétaire le prix d'un autre exemplaire.

Or en y réfléchissant un petit peu, on conclut vite que la plupart de ces personnes n'auraient pas acheté le logiciel. Ce qui n'empêche nullement les propriétaires de calculer leurs « pertes » comme si toutes ces personnes avaient été des acheteurs potentiels. Le moins qu'on puisse dire c'est qu'ils exagèrent.

- Le droit.

Les propriétaires parlent souvent des dispositions légales et des pénalités dont ils peuvent nous menacer. Implicitement, ils veulent nous dire là que les lois d'aujourd'hui reflètent un point de vue moral incontestable, et en même temps nous invitent à considérer les pénalités encourues comme des faits de nature, dont personne ne porte la responsabilité.

Ce type d'argumentation n'a pas été taillé pour résister au raisonnement critique mais pour venir renforcer une pensée routinière.

En aucune façon les lois ne sont des arbitres du bien et du mal. Tout Américain devrait savoir qu'il y a quarante ans, dans de nombreux États, il était illégal pour un Noir de s'asseoir à l'avant d'un autobus. Cependant seuls les racistes diront que c'était mal de le faire.

- Les droits naturels.

Souvent les auteurs revendiquent leur attachement affectif aux programmes qu'ils ont écrits et nous en font déduire que leurs désirs et leurs intérêts au sujet de ces programmes sont plus importants que ceux de toutes les autres personnes, plus importants même que ceux du monde entier. Il faut ici remarquer que la plupart du temps ce sont les sociétés et non les auteurs qui détiennent les droits de copie sur les logiciels, mais nous sommes censés négliger cette incohérence.

À ceux qui énoncent comme un axiome moral l'idée que l'auteur est plus important que le public, je peux seulement répondre que pour ma part, bien qu'auteur de logiciels très connus, je dis que c'est du chiqué.

Si les gens ont tellement tendance à sympathiser avec les arguments du type « droits naturels », c'est généralement pour l'une des deux raisons suivantes.

La première de ces raisons repose sur une analogie abusive avec les objets matériels. Si je prépare un plat de spaghetti, cela ne me va pas me convenir que quelqu'un d'autre que moi les mange, parce qu'alors je ne pourrai plus les manger moi-même. En mangeant mes spaghetti, l'autre personne me cause un dommage dans l'exacte mesure de son bénéfice à elle. D'elle ou de moi, une seule personne peut avoir les spaghetti, la question est de savoir qui. La plus petite distinction entre nous deux suffit à faire pencher la balance morale.

Mais le cas d'un programme que j'ai écrit est très différent. Si vous le faites fonctionner ou si vous le modifiez, cela vous profite directement mais ne m'affecte que d'une manière indirecte. Le fait que vous en donniez ou non une copie à un ami vous profite beaucoup plus, à vous et à votre ami, qu'il ne me dérange, moi. Il ne faut pas que j'aie le pouvoir de vous en empêcher. Personne ne doit avoir ce pouvoir.

Deuxièmement, les gens ont entendu dire que les droits naturels des auteurs constituaient une tradition universellement acceptée et jamais remise en question par notre société.

Pourtant historiquement c'est l'inverse qui est vrai. Au moment de la rédaction de la Constitution des États-Unis, l'idée de droits naturels pour les auteurs fut proposée mais finalement rejetée. C'est pourquoi la Constitution se borne à *autoriser* les systèmes de droit de copie, sans toutefois les rendre *obligatoires*. La Constitution énonce aussi que de tels systèmes doivent être temporaires, que leur but est de favoriser le progrès et non pas de récompenser les auteurs. Le droit de copie récompense certes les auteurs, et les éditeurs encore plus, mais toujours en tant que moyen d'infléchir leur

comportement.

La véritable tradition de notre société, c'est que le droit d'auteur empiète sur les droits naturels du public et ne peut se justifier que dans l'intérêt de ce même public.

- L'Économie.

Finalement, on nous dit qu'il faut aux logiciels des propriétaires parce que cela encourage leur production.

Contrairement aux autres arguments, celui-ci repose du moins sur une approche légitime du sujet. Le but est valable en effet: satisfaire les utilisateurs de logiciels. L'expérience montre que les gens produiront plus de logiciels s'ils sont mieux payés pour le faire.

Mais cet argument économique a le défaut d'être basé sur l'hypothèse discutable que la différence se fera en fonction de l'argent que nous voulons bien donner. On tient pour acquis que ce que nous voulons c'est de la « production de logiciels », avec ou sans propriétaire.

Le public accepte facilement cette hypothèse parce qu'elle correspond à son expérience des objets matériels. Si nous prenons l'exemple du sandwich, on peut sans doute réussir à en trouver deux qui soient identiques, l'un gratuit, l'autre payant. Si c'est le cas la somme payée ou non est la seule différence entre les deux. Que vous deviez ou non le payer, le sandwich aura le même goût, la même valeur nutritive et en tous cas vous ne pourrez le manger qu'une seule fois. Le fait le sandwich vous ait été fourni, ou non, par un propriétaire n'a d'influence directe que sur la somme d'argent dont vous disposez par la suite.

Ceci est vrai pour tout type d'objet matériel. Le fait qu'ils aient ou non un propriétaire ne change pas leur *nature*, ni leur utilité au cas où vous en faites l'acquisition.

Pour un programme c'est différent. Le fait qu'il ait un propriétaire modifie nettement sa nature et ce que vous pouvez en faire si vous en achetez un exemplaire. Et pas seulement pour une question d'argent, car le système de propriétaires de logiciels encourage ces derniers à produire un bien qui n'est pas celui dont la société a besoin. Il est la cause d'une pollution morale imperceptible qui nous affecte tous.

En effet, de quoi la société a-t-elle besoin ? D'une information vraiment disponible pour ses citoyens. Par exemple, des programmes que les gens peuvent réparer, modifier, adapter, améliorer, et pas seulement faire marcher. Or ce que les propriétaires de logiciels livrent généralement est une boîte noire que personne ne peut étudier ni modifier.

La société a aussi besoin de liberté. Quand un programme a un propriétaire, ses utilisateurs perdent la liberté de contrôler un des aspects de leur vie.

Surtout, la société a besoin d'encourager l'esprit de coopération volontaire de ses citoyens. Quand les propriétaires de logiciels qualifient de « pillage » le fait d'aider notre voisin d'une façon simple et naturelle, ils polluent l'esprit civique de notre société.

C'est pourquoi nous disons que le [logiciel libre](#) est une question de liberté, pas de prix.

L'argument économique des propriétaires est un faux argument, mais le problème économique est un vrai problème. Certaines personnes écrivent des logiciels utiles pour le plaisir ou pour conquérir l'admiration et la reconnaissance, mais si nous voulons plus de logiciels que ceux que ces personnes écrivent il nous faut récolter des fonds.

Depuis maintenant dix ans, les développeurs de logiciels libres essaient, avec un certain succès, diverses méthodes pour trouver des financements. Il n'est pas indispensable pour cela d'enrichir quelqu'un. Le revenu d'une famille américaine moyenne, autour de 35 mille dollars, a fait la preuve de sa capacité suffisante comme stimulant pour beaucoup de métiers moins satisfaisants que la programmation.

Pendant des années, jusqu'à ce que la création d'une association le rende superflu, j'ai gagné ma vie avec les améliorations que je faisais ponctuellement aux logiciels que j'avais écrits. Chacune de ces améliorations était ajoutée à la version livrée en standard, devenant ipso facto disponible au public. Les clients me payaient pour travailler aux améliorations dont ils avaient besoin et qui ne coïncidaient pas

forcément avec les fonctionnalités que j'aurais autrement considérées comme prioritaires.

La [Free Software Foundation \(FSF\)](#), une association exemptée d'impôt pour le développement du logiciel libre, récolte des fonds par la [vente](#) de [CD-ROMs](#) GNU, de [T-shirts](#), de [manuels](#) et de [distributions de luxe](#) — que chaque utilisateur a la possibilité de copier ou de modifier librement, ainsi que par les [dons](#). Nous avons maintenant cinq programmeurs et l'équipe compte aussi trois employés pour traiter les commandes par courrier.

Certains développeurs de logiciels libres gagnent leur vie en vendant des services de support technique. Cygnus Support, avec 50 employés [au moment de la rédaction de cet article], estime que 15 pourcent de l'activité de ses équipes est consacrée au développement de logiciels libres—pourcentage fort respectable pour une société de logiciels.

D'autres sociétés, parmi lesquelles Intel, Motorola, Texas Instruments et Analog Devices, se sont regroupées pour financer le développement du compilateur libre GNU pour le langage C. Dans le même temps, le compilateur libre GNU pour le langage Ada est soutenu financièrement par la US Air Force, car celle-ci pense que c'est le moyen le plus économique d'obtenir un compilateur de haute qualité. [Les subventions de US Air Force ont cessé il y a quelque temps; le compilateur GNU pour Ada est maintenant en service et sa maintenance est subventionnée par des sociétés commerciales.]

Tous ces exemples sont à petite échelle. Le mouvement du logiciel libre est encore petit et encore jeune. Mais l'exemple de la radio financée par les auditeurs dans ce pays [les États-Unis] montre qu'il est possible de soutenir une activité importante sans forcer chaque utilisateur à payer.

En tant qu'utilisateur d'ordinateur aujourd'hui, vous utilisez peut-être un [programme propriétaire](#). Si votre ami vous en demande une copie, ce serait mal de refuser. La coopération est plus importante que le droit de copie. Cependant, dans le fond, la coopération en circuit fermé ne permettra pas de bâtir une bonne société. Chacun devrait aspirer à une vie honnête menée ouvertement et avec fierté, et pour cela il faut dire « non ! » au logiciel propriétaire.

Vous méritez de pouvoir coopérer ouvertement et librement avec les autres personnes qui utilisent les logiciels. Vous méritez de pouvoir apprendre comment marche un logiciel et de l'enseigner à vos élèves. Vous méritez de pouvoir engager votre programmeur favori pour réparer le logiciel quand il est cassé.

Vous méritez le logiciel libre.

Copyright © 1994 Richard Stallman

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Michel Rodriguez.

Révision : trad-gnu@april.org

Dernière mise à jour : Date: 2009/01/25 09:28:12

Traductions de cette page

[български](#) [bg] [Català](#) [ca] [Česky](#) [cs] [Dansk](#) [da] [Deutsch](#) [de] [ελληνικά](#) [el] [English](#) [en] [Español](#) [es] [فارسی](#) [fa] [Français](#) [fr]
[Hrvatski](#) [hr] [Magyar](#) [hu] [Bahasa Indonesia](#) [id] [Italiano](#) [it] [日本語](#) [ja] [한국어](#) [ko] [മലയാളം](#) [ml] [Nederlands](#) [nl] [Polski](#) [pl]
[português do Brasil](#) [pt-br] [Русский](#) [ru] [தமிழ்](#) [ta] [Türkçe](#) [tr]

Qu'y a-t-il dans un nom ?

par [Richard Stallman](#)

Si vous souhaitez en savoir plus sur ce sujet, vous pouvez aussi consulter notre [FAQ GNU/Linux](#) et notre page sur [Les utilisateurs de GNU qui n'ont jamais entendu parler de GNU](#).

Les noms véhiculent des significations ; le choix des noms détermine la signification de ce que nous disons. Un nom mal choisi donne une mauvaise idée aux gens. Une rose, quel que soit son nom, sentira toujours aussi bon, mais si vous l'appellez stylo, les gens seront plutôt surpris quand ils essaieront d'écrire avec. Et si vous appelez les stylos «roses», les gens pourraient ne pas réaliser à quoi ils servent. Si vous appelez notre système d'exploitation «Linux», cela véhicule une idée fautive de l'origine du système, de son histoire et de son but. Si vous l'appellez [GNU/Linux](#), cela transporte (mais pas en détail) une idée correcte.

Mais est-ce important pour notre communauté ? Est-ce important que les gens connaissent l'origine, l'histoire et le but du système ? Oui, parce que ceux qui oublient l'histoire sont souvent condamnés à la répétition. Le Monde Libre qui s'est développé autour de [GNU/Linux](#) n'est pas en sécurité ; les problèmes qui nous ont amené à développer GNU ne sont pas complètement éradiqués et ils menacent de revenir.

Quand j'explique pourquoi il est correct d'appeler le système d'exploitation «GNU/Linux» plutôt que «Linux», les gens répondent parfois de la manière suivante :

Bien que le projet GNU mérite le crédit pour son travail, est-ce que cela vaut la peine de s'agiter quand les personnes ne le donnent pas ? Est-ce que la chose importante n'est pas que le travail soit fait et pas celui qui l'a fait ? Vous devriez vous calmer, être fiers du travail bien fait et ne pas vous soucier de crédit.

Ce serait un sage conseil si seulement la situation était telle, si le travail était fait et s'il était temps de se calmer. Si seulement c'était vrai ! Mais les problèmes abondent et il n'est pas temps de considérer le futur comme acquis. La force de notre communauté repose sur son engagement dans la liberté et la coopération. Utiliser le nom [GNU/Linux](#) est une manière pour les personnes de se le rappeler et d'informer les autres de ces buts.

Il est possible d'écrire de bons logiciels sans penser à GNU ; beaucoup de bons travaux ont été faits au nom de Linux aussi. Mais «Linux» a été associé même s'il a d'abord été inventé avec une philosophie qui ne s'engage pas dans la liberté de coopérer. Comme le nom est de plus en plus utilisé par le commerce, nous aurons encore plus de mal à le connecter à l'esprit de la communauté.

Un grand défi pour le futur du logiciel libre vient de la tendance des compagnies de distribution de «Linux» à ajouter des logiciels non libres à [GNU/Linux](#), au nom de la commodité et de la puissance. Tous les développeurs de distributions commerciales majeures le font, aucune ne produit une distribution entièrement libre. La plupart des compagnies n'identifie pas clairement les produits non libres dans leurs distributions ; beaucoup développent même des logiciels non libres et les ajoutent au système. Quelques unes annoncent de façon scandaleuse des systèmes «Linux» dont le droit d'utilisation est concédé par poste, ce qui donne à l'utilisateur autant de liberté que Microsoft Windows.

Les personnes justifient l'ajout de logiciel non libre au nom de la «popularité de Linux», mettant en fait la popularité au-dessus de la liberté. Parfois, c'est ouvertement admis. Par exemple, Wired Magazine dit que Robert McMillan, éditeur de Linux Magazine, «pense que le mouvement vers le logiciel open source devrait être soutenu par des décisions techniques, plutôt que politiques». Et le PDG de Caldera a demandé ouvertement aux utilisateurs de laisser tomber le but de la liberté pour travailler à la place en faveur de «la popularité de Linux».

Ajouter des logiciels non libres au système [GNU/Linux](#) pourrait augmenter sa popularité, si nous signifions par popularité le nombre de personnes utilisant une partie de [GNU/Linux](#) en combinaison avec des logiciels non libres. Mais en même temps, cela encourage implicitement la communauté à accepter des logiciels non libres comme une bonne chose et à oublier le but de la liberté. Ça ne sert à rien de conduire plus vite si vous ne pouvez pas rester sur la route.

Quand un « ajout » non libre est une bibliothèque ou un outil de développement, il peut devenir un piège pour les développeurs de logiciel libre. Quand ils écrivent un logiciel libre qui dépend d'un outil non libre, leur logiciel ne peut pas faire partie d'un système complètement libre. Motif et Qt ont emprisonné de nombreux logiciels libres de cette manière dans le passé, en créant des problèmes dont la solution a pris des années. Le problème de Motif n'est toujours pas entièrement résolu, puisque LessTif a encore besoin de peaufinage (des volontaires, s'il vous plaît !). L'implémentation Java non libre de Sun a désormais un effet similaire : [Le piège Java](#). (Note historique : En décembre 2006, Sun est sur le point de republier sa plateforme Java sous licence GNU GPL).

Si notre communauté continue d'aller dans cette direction, cela peut rediriger le futur de [GNU/Linux](#) vers une mosaïque de composants libres et non libres. Dans cinq ans, nous aurons certainement toujours beaucoup de logiciels libres ; mais si nous ne faisons pas attention, ils seront difficilement utilisables sans les logiciels non libres que les utilisateurs espèrent trouver avec. Si cela arrive, notre campagne pour la liberté aura échoué.

Si la publication d'alternatives libres était simplement une question de programmation, la résolution de problèmes futurs pourrait devenir plus simple au fur et à mesure que les ressources de développement de notre communauté augmentent. Mais nous faisons face à des obstacles qui menacent de la rendre plus difficile : des lois qui interdisent le logiciel libre. Comme les brevets logiciels se dressent (lisez la pétition petition.eurolinux.org et signez la !) et comme des lois telles que la DMCA sont utilisées pour interdire le développement de logiciels libres concernant d'importants besoins comme la lecture de DVD ou l'écoute d'un flux RealAudio, nous nous trouverons sans autre manière claire de nous battre contre les formats de fichiers brevetés et secrets que de **rejeter les programmes non libres qui les utilisent**.

Répondre à ces défis demandera plusieurs sortes d'effort. Mais ce dont nous avons besoin avant tout, pour affronter tous types d'adversité, est de garder en mémoire le but de la liberté de coopérer. Nous ne pouvons pas nous reposer sur le simple désir de logiciels puissants et fiables pour motiver les gens à faire de grands efforts. Nous avons besoin de cette détermination que les gens ont quand ils se battent pour leur liberté et leur communauté, la détermination de tenir pendant des années et de ne pas abandonner.

Dans notre communauté, ce but et cette détermination émanent principalement du projet GNU. Nous sommes ceux qui parlons de liberté et de communauté comme de quelque chose qu'il faut défendre; les organisations qui parlent de « Linux » ne le disent normalement pas. Les magazines sur « Linux » sont typiquement pleins de publicités pour des logiciels non libres; les compagnies qui empaquettent « Linux » ajoutent des logiciels non libres au système; d'autres compagnies « supportent Linux » avec des applications non libres; les groupes d'utilisateurs de « Linux » invitent typiquement des revendeurs à présenter ces applications. Le point central où les personnes de notre communauté sont susceptibles de croiser l'idée de la liberté et de la détermination est le Projet GNU.

Mais quand les gens la croisent, est-ce qu'ils penseront qu'elle les concerne ?

Les personnes qui savent qu'elles utilisent un système qui vient du projet GNU peuvent voir une relation directe entre eux et GNU. Elles ne vont pas forcément être d'accord avec notre philosophie, mais au moins elles verront une raison d'y penser sérieusement. Par contre, les personnes qui se considèrent comme des « utilisateurs Linux » et qui croient que le projet GNU « a développé des outils qui ont prouvé leur efficacité pour Linux », perçoivent typiquement une relation indirecte entre GNU et eux. Elles peuvent tout simplement passer à côté de la philosophie de GNU.

Le projet GNU est idéaliste et tous ceux qui encouragent l'idéalisme aujourd'hui font face à un grand obstacle : l'idéologie dominante qui encourage les personnes à rejeter l'idéalisme comme « manquant de sens pratique ». Notre idéalisme a été on ne peut plus pratique : c'est bien la raison pour laquelle nous avons un système d'exploitation [GNU/Linux](#). Les personnes qui aiment ce système devraient savoir que c'est notre idéalisme qui l'a rendu possible, concret.

Si « le travail » avait été fait, s'il n'y avait rien d'autre en jeu à part le crédit, peut-être qu'il serait plus sage de laisser tomber le sujet. Mais nous ne sommes pas dans cette position. Pour inciter les gens à faire le travail qui doit être fait, nous avons besoin d'être reconnus pour ce que nous avons déjà fait. S'il vous plaît, aidez-nous en appelant le système d'exploitation [GNU/Linux](#).

P.S. : pour une explication de l'histoire du système GNU/Linux par rapport à la question du nom, voir <http://www.gnu.org/gnu/linux-and-gnu.fr.html>. Pour des réponses aux questions et arguments sur ce sujet, veuillez consulter la [FAQ GNU/Linux](#).

Copyright © 2000, 2006, 2007 Richard Stallman

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Pierre-Yves Enderlin.

Révision : trad-gnu@april.org

Dernière mise à jour : Date: 2008/09/14 08:27:42

Traductions de cette page

[български](#) [bg] [Català](#) [ca] [Česky](#) [cs] [Deutsch](#) [de] [Ελληνικά](#) [el] [English](#) [en] [Español](#) [es] [Français](#) [fr] [עברית](#) [he] [Italiano](#) [it]
[한국어](#) [ko] [Nederlands](#) [nl] [Polski](#) [pl] [português do Brasil](#) [pt-br] [Română](#) [ro] [српски](#) [sr] [Türkçe](#) [tr]

Pourquoi « logiciel libre » est-il meilleur que « open source »

[Pourquoi l'« open source » passe à côté du problème que soulève le logiciel libre](#) est une version mise à jour de cet article.

Bien que la liberté d'un logiciel ne soit pas dépendante du nom qu'on lui donne, il y a des différences dans les noms qu'on lui accorde : des mots différents apportent des idées différentes.

En 1998, quelques personnes de la communauté du logiciel libre ont commencé à utiliser le terme « [open source](#) » (*logiciel ouvert*) au lieu de « [logiciel libre](#) » (*free software*). Le terme « open source » devint rapidement associé à une approche, à une philosophie, à des valeurs et même à des critères différents pour lesquels les licences sont acceptables. Le Mouvement du logiciel libre et le Mouvement open source sont aujourd'hui des [mouvements séparés](#), avec des vues et des objectifs différents, bien que nous puissions travailler ensemble, et nous le faisons, sur quelques projets concrets.

La différence fondamentale entre les deux mouvements se situe dans leurs valeurs, leurs façons de voir le monde. Pour le Mouvement open source, la question de savoir si un logiciel devrait être open source est une question pratique, non éthique. Comme quelqu'un l'a dit : « le Mouvement open source est une méthodologie de développement ; le Mouvement du logiciel libre, un mouvement social ». Pour le Mouvement open source, un logiciel non libre est une solution sous-optimale. Pour le Mouvement du logiciel libre, le logiciel non libre est un problème social et le logiciel libre en est la solution.

Relations entre le Mouvement du logiciel libre et le Mouvement open source

Le Mouvement du logiciel libre et le Mouvement open source sont comme deux partis politiques à travers notre communauté.

Les groupes radicaux des années 1960 sont connus pour leur factionalisme : les organisations se séparent à cause de divergences dans des détails de stratégie, et se détestent alors. Ils sont d'accord sur les principes de base, et n'ont de divergences que sur des recommandations pratiques ; mais ils se considèrent comme des ennemis, et se battent sauvagement.

Pour le Mouvement du logiciel libre et le Mouvement open source, c'est tout simplement le contraire. Nous ne sommes pas d'accord sur les principes de base, mais nous sommes d'accord sur la plupart des recommandations pratiques. Nous travaillons ensemble sur de nombreux projets spécifiques.

Dans le Mouvement du logiciel libre, nous ne pensons pas que le Mouvement open source soit un ennemi. L'ennemi est [le logiciel propriétaire](#). Mais nous voulons que les personnes de notre communauté sachent que nous ne sommes pas pareils qu'eux !

Alors veuillez mentionner le Mouvement du logiciel libre lorsque vous parlez du travail que vous avez accompli, et du logiciel que vous avez développé, tout comme du système d'exploitation [GNU/Linux](#).

Comparaison des deux termes

Le reste de cet article compare les termes « logiciel libre » et « open source ». Il montre pourquoi le terme « open source » ne résout aucun problème, et en fait en crée même certains.

Ambiguïté

Le terme « free software » pose un problème d'ambiguïté : une signification involontaire, « logiciel gratuit », (*N.d.T. : free =gratuit ou libre en anglais*) est aussi valable que la signification voulue, « logiciel qui accorde des libertés à l'utilisateur ». Nous résolvons ce problème en publiant [une définition plus précise du logiciel libre](#), mais ce n'est qu'une solution partielle ; elle ne peut pas éliminer complètement le problème. Un terme correct et non ambigu serait meilleur, en supposant que cela n'entraîne pas d'autres problèmes.

Malheureusement, toutes les alternatives ont leurs propres défauts. Nous avons étudié de nombreuses alternatives que l'on nous a suggérées, et alors que certaines évitaient ce problème, elles en entraînaient de

nouveaux, aucune n'étant suffisamment « correcte » pour que ce soit une bonne idée de l'utiliser. Chaque alternative proposée pour « free software » posait les mêmes sortes de problèmes sémantiques, ou pire, ce qui est le cas de « open source ».

La définition officielle de « logiciel open source », telle qu'elle est publiée par la Open Source Initiative, est très proche de notre définition de logiciel libre ; cependant, elle est un peu vague sur certains aspects, et ils ont accepté quelques licences que nous considérons restrictives de façon inacceptable pour les utilisateurs. La signification évidente de « open source » est « Vous pouvez avoir le code source ». Cette catégorie de logiciels n'est clairement pas la même que celle de logiciels libres. Elle inclut les logiciels libres, mais aussi les logiciels semi-libres tels que Xv, et même les logiciels non-libres tels que Qt sous sa licence d'origine.

La signification explicite d'« open source » n'est pas la signification que ses partisans désirent (leur définition « officielle » est plus proche de « logiciel libre »). Le résultat est que le public comprend généralement mal la signification de ces termes. Voici comment l'écrivain Neal Stephenson définit « open source » :

Linux est un logiciel « open source » signifie, simplement, que n'importe qui peut obtenir des copies de ses fichiers sources.

Je ne pense pas qu'il avait l'idée de délibérément rejeter ou critiquer la définition « officielle ». Il a simplement appliqué les conventions de la langue anglaise, et tiré la conclusion naturelle. L'état du Kansas a publié une définition similaire :

De l'utilisation de logiciel open source (OSS). L'OSS est un logiciel pour lequel le code source est librement et publiquement accessible, bien que les accords de licence spécifiques varient sur ce que chacun est autorisé à faire avec ce code.

Bien sûr, il est possible de résoudre ce problème en publiant une définition précise, comme nous l'avons fait pour « free software ».

Mais l'explication pour « logiciel libre » est simple -- une personne qui a saisi l'idée de « discours libre, pas bière gratuite » (« free speech, not free beer ») ne commettra plus l'erreur. Il n'y a pas de manière aussi succincte pour expliquer la signification de « open source » et pour montrer clairement pourquoi la définition naturelle n'est pas la bonne.

Peur de la liberté

L'argument principal pour le terme « open source » est que « free software » rend certaines personnes méfiantes. C'est vrai : parler de liberté, de questions éthiques, de responsabilités aussi bien que de convenance, c'est demander aux gens de réfléchir à des choses qu'ils préféreraient ignorer; cela peut induire une certaine gêne. Mais nous ne rendrions pas la société meilleure si nous arrêtons de parler de ces choses.

Il y a quelques années, des développeurs de logiciels libres ont remarqué cette réaction, et certains ont commencé à explorer des voies pour l'éviter. Ils ont imaginé qu'en restant prudent à propos de l'éthique et de la liberté, et en ne parlant que des bénéfices pratiques et immédiats de certains logiciels libres, ils pouvaient « vendre » le logiciel libre plus efficacement à certains utilisateurs, principalement en entreprise. C'est pour cela qu'ils ont utilisé le terme « open source », pour être « plus acceptable en entreprise ».

Cette approche a prouvé son efficacité, à sa manière. Aujourd'hui, beaucoup de personnes se mettent au logiciel libre pour des raisons purement pratiques. C'est une bonne chose, mais ce n'est pas la seule chose que nous devons entreprendre ! Attirer des utilisateurs vers le logiciel libre n'est pas le but final que nous devons atteindre, mais juste le premier pas.

Tôt ou tard ces utilisateurs seront tentés de retourner au logiciel propriétaire pour d'autres avantages pratiques. Des entreprises mineures cherchent à profiter d'une telle tentation, et pourquoi les utilisateurs refuseraient-ils ? Seulement s'ils ont compris *l'importance de la liberté* dans le logiciel libre pour elle-même. C'est à nous de répandre cette idée, et pour le faire nous devons parler de liberté. Une certaine partie de l'approche « prudente » est utile pour la communauté, mais nous devons également avoir beaucoup de

discours sur la liberté.

À présent, nous avons beaucoup de partisans de l'approche « prudente », mais pas assez de discours sur la liberté. La plupart des personnes concernées par le logiciel libre parlent peu de la liberté, souvent parce qu'elle cherchent à être « plus acceptables en entreprise ». Les distributeurs de logiciels présentent essentiellement cet aspect. Quelques distributions du système d'exploitation [GNU/Linux](#) ajoutent des paquetages propriétaires au système libre de base, et ils espèrent que les utilisateurs considèrent cela comme un avantage, plutôt qu'un retour en arrière.

Nous n'arrivons pas à gérer le flux entrant des utilisateurs de logiciels libres, nous n'arrivons pas à leur parler de liberté et de notre communauté dès qu'ils y entrent. C'est pourquoi les logiciels non-libres (comme Qt, à l'époque où celui-ci devint populaire) et des distributions en partie non-libres de systèmes d'exploitation trouvent une terre si fertile. Abandonner maintenant le mot « libre » serait une erreur; nous devons parler plus (et pas moins) de la liberté.

Espérons que ceux qui utilisent le terme « open source » attireront en fait plus d'utilisateurs dans notre communauté mais s'ils y arrivent, le restant de notre communauté devra travailler encore plus durement pour porter le problème de la liberté aux oreilles de ces utilisateurs. Nous devons dire « C'est un logiciel libre et il vous donne la liberté » plus souvent et plus fort qu'auparavant.

Est-ce qu'une marque déposée nous aiderait ?

Les partisans du « logiciel open source » ont essayé d'en faire une marque déposée, en disant que ça leur permettra d'éviter les abus. L'essai échoua lorsque la démarche cessa de prendre effet en 1999 ; ainsi, le statut légal d'« open source » est le même que celui de « free software » il n'y a pas de contrainte *légal*e à son utilisation. J'ai entendu parler d'entreprises appelant des logiciels « open source » même s'ils ne correspondaient pas à la définition officielle ; j'ai personnellement observé quelques exemples.

Mais est-ce que cela ferait une grosse différence d'utiliser un terme qui est déposé ? Pas nécessairement.

Des entreprises font parfois des annonces qui donnent l'impression qu'un programme est un « logiciel open source » sans vraiment le dire explicitement. Par exemple, les annonces d'IBM à propos d'un programme qui ne correspond pas à la définition officielle dit ceci :

Comme c'est courant dans la communauté open source, les utilisateurs de la technologie... vont aussi pouvoir collaborer avec IBM...

Ceci ne disait pas vraiment que le programme *était* « open source », mais beaucoup de lecteurs n'ont pas noté ce détail (je devrais noter qu'IBM essayait sincèrement de rendre son programme libre, et adopta plus tard une nouvelle licence qui le rendit libre et « open source »; mais quand l'annonce fut faite, le programme n'était ni l'un ni l'autre).

Et voici comment Cygnus Solutions, qui a été formée pour être une entreprise de logiciels libres et qui a étendu par la suite ses activités (pour ainsi dire) dans le monde du logiciel propriétaire, fit de la publicité pour des logiciels propriétaires :

Cygnus Solutions fait partie des leaders du marché de l'open source et vient juste de lancer deux nouveaux produits sur le marché Linux.

Contrairement à IBM, Cygnus n'essayait pas de rendre ses logiciels libres, et ses logiciels étaient loin d'être des logiciels libres. Mais Cygnus ne dit pas vraiment que ce sont des « logiciels open source », ils écrivent juste une phrase vague pour essayer d'obtenir l'attitude favorable qui vient avec ce terme.

Ces observations suggèrent qu'une marque déposée n'aurait pas vraiment résolu les problèmes du terme « open source ».

Incompréhensions(?) au sujet d'« open source »

La définition de l'open source est assez claire, et il est complètement clair que le programme typiquement non libre n'y correspond pas. Alors vous pourriez penser qu'une « entreprise open source » fait des logiciels libres, n'est-ce pas ? Hélas, nombreuses entreprises essayent de lui donner une autre définition.

Au colloque « Open Source Developers Day » en août 1998, plusieurs des développeurs commerciaux invités dirent qu'ils n'avaient l'intention de rendre qu'une partie de leur travail libre (ou « open source »). La partie principale de leur activité est de développer des extensions propriétaires (logiciels ou [manuels](#)) à vendre aux utilisateurs de logiciel libre. Ils nous demandent de considérer ceci comme légitime en tant que part de notre communauté, car une partie des bénéfices est reversé au développement de logiciels libres.

En effet, ces entreprises cherchent à acquérir l'apparence favorable de l'« open source » pour leurs logiciels propriétaires, tout en sachant que ce ne sont pas des « logiciels open source », car ils ont quelques relations avec le logiciel libre ou parce que la même entreprise maintient des logiciels libres (le fondateur d'une entreprise dit explicitement qu'il consacrerait le minimum de travail que la communauté accepterait dans le logiciel libre qu'ils maintiennent).

À travers les années, de nombreuses entreprises ont contribué au développement de logiciels libres. Certaines de ces entreprises développaient d'abord des logiciels propriétaires, mais les deux activités étaient séparées; ainsi, nous pouvions ignorer leurs produits non libres, et travailler avec eux sur leurs projets libres. Alors, nous pouvions par la suite les remercier honnêtement pour leur contribution au logiciel libre, sans parler du reste de leurs activités.

Nous ne pouvons plus faire de même avec ces nouvelles entreprises, car elles n'avanceront pas ainsi. Ces entreprises essayent activement de conduire le public à considérer leurs activités en bloc; elles veulent que nous considérions leurs logiciels non libres aussi favorablement qu'une vraie contribution, même si ce n'en est pas une. Elles se présentent elles-mêmes comme des « entreprises open source », en espérant que nous allons ressentir un sentiment positif à leur propos, et que nous allons l'appliquer aveuglément.

Cette pratique manipulatrice ne serait pas moins grave si elle était faite en utilisant le terme « logiciel libre ». Mais des entreprises n'ont pas l'air d'utiliser le terme « logiciel libre » de cette manière; peut-être que l'association avec l'idéalisme le rend inapproprié. Le terme « open source » a ouvert la porte à ces dérives.

Fin 1998, à une exposition dédiée au système d'exploitation qu'on appelle souvent « [Linux](#) », l'intervenant était un cadre d'une importante entreprise de logiciels. Il a certainement été invité grâce à la décision de son entreprise de « supporter » ce système. Malheureusement, leur support consiste à éditer des logiciels non libres qui fonctionnent sur ce système, en d'autres mots, utiliser notre communauté comme marché mais sans y contribuer.

Il dit « Nous ne rendrons pas notre produit open source, mais peut-être le rendrons-nous open source en « interne ». Si nous autorisons notre support client à avoir accès au code source, ils pourraient corriger des bugs pour le client, et nous pourrions fournir un meilleur produit et un meilleur service ». (Ce n'est pas la citation exacte puisque je ne l'ai pas notée, mais ç'en est l'essentiel).

Des personnes du public me dirent plus tard « Il n'a tout simplement rien compris ». Mais est-ce le cas ? Quel point n'a-t-il pas compris ?

Il n'a pas raté tout ce qui est habituellement associé au terme « open source ». Ceci ne dit rien sur la liberté, mais dit seulement qu'autoriser plus de personnes à accéder au code source et aider à l'améliorer rendra le développement plus rapide et plus efficace. Le cadre a complètement assimilé ce point; en négligeant pour d'autres raisons de conduire cette approche jusqu'à la fin, en incluant les utilisateurs, il a juste pensé à le mettre en œuvre partiellement, à l'intérieur de l'entreprise.

Ce qu'il n'a pas compris c'est que l'« open source » n'a pas été conçu pour relever que les utilisateurs *méritent* la liberté.

Faire progresser l'idée de la liberté est un gros travail (qui a besoin de votre aide). Le projet GNU s'attachera au terme « logiciel libre ». Si vous croyez que la liberté et la communauté sont importantes en tant

que telles, pas juste pour la commodité qu'elles apportent, rejoignez-nous en utilisant le terme « logiciel libre ».

Joe Barr a écrit un article intitulé [Live and let license](#) (croissez et faites des licences) qui donne son point de vue sur le problème.

Le [document sur la motivation des développeurs de logiciels libres](#) de Lakhani et Wolf, dit qu'une part considérable d'entre eux est motivée par l'idée que le logiciel devrait être libre. Ce, en dépit du fait qu'ils ont interrogé les développeurs sur SourceForge, un site qui ne soutient pas l'idée qu'il s'agit d'un problème éthique.

Copyright © 1998, 1999, 2000, 2001, 2007 Free Software Foundation, Inc.,

51 Franklin St, Fifth Floor, Boston, MA 02110, USA

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Benjamin Drieu.

Révision : trad-gnu@april.org

Dernière mise à jour : Date: 2009/01/19 09:27:59

Traductions de cette page

[български](#) [bg] [Česky](#) [cs] [Deutsch](#) [de] [English](#) [en] [Español](#) [es] [Français](#) [fr] [עברית](#) [he] [Italiano](#) [it]
[한국어](#) [ko] [Polski](#) [pl] [Română](#) [ro] [Русский](#) [ru] [Српски](#) [sr] [简体中文](#) [zh-cn] [繁體中文](#) [zh-tw]

Écrire des logiciels libres si vous travaillez à l'université

Dans le Mouvement du logiciel libre, nous pensons que les utilisateurs d'ordinateurs devraient être libres de modifier et de redistribuer les logiciels qu'ils utilisent. Le « libre » de logiciel libre renvoie bien à la notion de liberté : il signifie que les utilisateurs ont la liberté d'exécuter, de modifier et de redistribuer le logiciel¹. Le logiciel libre contribue au savoir humain, au contraire des logiciels non libres. Les universités devraient donc encourager le logiciel libre dans l'intérêt de l'avancée de la connaissance humaine, de la même manière qu'elles devraient encourager les scientifiques et les étudiants à publier leurs travaux.

Hélas, bien des directeurs d'universités ont une attitude possessive envers le logiciel (et envers la science); ils considèrent les programmes comme autant d'opportunités de revenus, et non comme une chance de participer à la connaissance humaine. Les développeurs de logiciels libres ont dû faire face à cette tendance depuis presque 20 ans.

Quand j'ai commencé à développer le [système d'exploitation GNU](#) en 1984, mon premier pas a été de démissionner de mon poste au MIT. J'ai fait cela spécialement pour que le bureau des licences du MIT ne puisse interférer avec la diffusion de GNU en tant que logiciel libre. L'approche que j'avais élaborée pour les licences des programmes de GNU permet de garantir que toutes les versions modifiées doivent être aussi des logiciels libres, un concept qui s'est approfondi dans [la Licence publique générale GNU](#) (GPL GNU), et je ne voulais pas avoir à supplier l'administration du MIT de me laisser l'utiliser.

Pendant des années, des universitaires ont souvent contacté la Fondation pour le logiciel libre afin d'obtenir des conseils sur la manière de s'arranger avec des directions qui voient seulement dans le logiciel quelque chose qui puisse se vendre. Une bonne méthode, qui peut même être appliquée sur des projets financés de façon spécifique, est de baser votre travail sur un programme existant qui a été publié sous licence GPL GNU. Vous pouvez alors dire aux gestionnaires : « Nous n'avons pas le droit de diffuser la version modifiée autrement que sous GPL GNU (toute autre façon constituerait une violation de copyright). » Une fois évanouis sous leurs yeux les symboles de dollars, ils consentiront généralement à le diffuser en tant que logiciel libre.

Vous pouvez aussi appeler à l'aide votre sponsor financier. Quand un groupe de NYU a développé le compilateur Ada GNU, avec le financement de l'U.S. Air Force, le contrat précisait explicitement que le code résultant serait cédé à la Fondation pour le logiciel libre. Obtenez un arrangement avec votre sponsor tout d'abord, puis faites remarquer poliment à l'administration de votre université que ce point n'est pas renégociable. Elle préférera un contrat pour développer un logiciel libre que pas de contrat du tout, donc elle s'en accommodera vraisemblablement.

Quoi que vous fassiez, posez le problème au plus tôt (certainement avant que le programme ne soit à moitié terminé). À ce stade, l'université a encore besoin de vous, donc vous pouvez négocier durement : dites à l'administration que vous terminerez le programme, le rendrez utilisable, s'ils acceptent par écrit d'en faire un logiciel libre (sous la licence de logiciel libre que vous aurez choisie). Sinon vous travaillerez dessus juste assez pour pouvoir écrire un papier à son propos, et ne créerez jamais une version suffisamment correcte pour être diffusée. Quand la direction comprendra qu'elle n'a le choix qu'entre un paquetage de logiciel libre qui créditera l'université et rien du tout, elle choisira généralement la première solution.

Toutes les universités n'ont pas des politiques possessives. L'Université du Texas a une politique qui permet facilement de diffuser un logiciel développé là-bas en tant que logiciel libre sous la Licence publique générale GNU. Univas au Brésil et l'Institut indien des technologies de l'information à Hyderabad, en Inde, ont tous deux adopté une politique en faveur de la diffusion des logiciels sous GPL. En développant tout d'abord l'appui du corps professoral, vous pouvez peut-être instaurer une politique de ce genre dans votre université. Présentez le problème comme une question de principe : est-ce que la mission de l'université est de faire progresser la connaissance humaine, ou est-ce que son seul but est de s'entretenir elle-même ?

Quelle que soit l'approche que vous choisissiez, faire preuve de détermination et opter pour une perspective éthique, comme nous le faisons au sein du Mouvement du logiciel libre, aide toujours. Pour faire preuve d'éthique envers le public, le logiciel devrait être libre pour l'ensemble du public.

Beaucoup de développeurs de logiciels libres font état d'une motivation pour produire du logiciel

libre pragmatique et limitée : ils soutiennent que permettre aux autres de partager et de modifier le logiciel a pour avantage la construction de logiciels puissants et fiables. Si ces valeurs vous motivent à développer du logiciel libre, c'est bon et bien, et nous vous remercions pour votre contribution. Mais ces valeurs ne vous aideront pas à rester ferme face à la direction de l'université quand elle essaiera de faire pression ou de vous inciter à rendre le programme non libre.

Par exemple, elle pourra avancer : « Nous pourrons le rendre encore plus puissant et fiable avec tout l'argent que nous en obtiendrons ». Cet argument peut s'avérer vrai ou faux au final, mais il est difficile de le démentir à l'avance. Ils pourraient suggérer une licence permettant d'offrir des copies « gratuites, réservées à une utilisation académique », ce qui signifie au public qu'il ne mérite pas de bénéficier de la liberté, et ils ajouteraient que cela vous permettra d'obtenir une coopération universitaire, ce qui est (selon eux) tout ce dont vous avez besoin.

Si vous partez de valeurs « pragmatiques », il est difficile de donner une bonne raison pour rejeter ces propositions sans issue, alors que vous pouvez le faire facilement si vous basez votre position sur des valeurs éthiques et politiques. Qu'y a-t'il de bon à faire un programme puissant et fiable au prix de la liberté des utilisateurs ? La liberté ne devrait-elle pas s'appliquer tout autant à l'extérieur qu'à l'intérieur de l'Université ? Les réponses sont évidentes si la liberté et la communauté font partie de vos objectifs. Le logiciel libre respecte la liberté des utilisateurs, alors que les logiciels non libres la bafouent.

Rien ne peut renforcer plus votre résolution que le fait de savoir que la liberté de la communauté dépend, en cette occasion, de vous.

Notes

1. Dans l'original en anglais, le terme « free » est employé, avec l'ambiguïté propre à la langue anglaise associée à ce mot qui signifie aussi bien « gratuit », ou « libre »

Copyright © 2002 Richard Stallman

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Jean-Jacques Puig.

Révision : trad-gnu@april.org

Dernière mise à jour : Date: 2009/01/25 09:28:12

Traductions de cette page

[Català](#) [ca] [Česky](#) [cs] [English](#) [en] [Español](#) [es] [Français](#) [fr] [עברית](#) [he] [Italiano](#) [it] [Nederlands](#) [nl]

[Polski](#) [pl] [português do Brasil](#) [pt-br]

Vendre des logiciels libres

Beaucoup de personnes croient que l'esprit du projet GNU est de ne pas faire payer la distribution de copies de logiciels, ou alors le moins possible : juste assez pour couvrir les frais.

En fait, nous encourageons ceux qui distribuent des [logiciels libres](#) à les faire payer le prix qu'ils veulent ou peuvent. Si cela vous semble surprenant, continuez à lire.

Le mot anglais « free » (libre) a deux sens, il peut aussi bien faire référence au prix qu'à la liberté. Quand nous parlons de « free software » (logiciel libre) nous parlons de la liberté, pas du prix. Plus particulièrement, il signifie qu'un utilisateur est libre d'utiliser un programme, de le modifier et de le redistribuer, avec ou sans modifications.

Les logiciels libres sont parfois distribués gratuitement, et parfois contre rémunération. Un même programme est souvent disponible sous ces deux versions à partir de sources différentes. Le programme est libre en dépit de son prix, car les utilisateurs ont toute liberté dans son utilisation.

[Les logiciels propriétaires](#) sont souvent vendus à un prix élevé, mais parfois un revendeur peut vous en donner une copie gratuite. Cela n'en fait pas pour autant un logiciel libre. Qu'il soit gratuit ou payant, le programme n'est pas libre car les utilisateurs n'ont aucune liberté.

Puisque le prix n'a pas d'importance lorsque nous parlons de logiciel libre, un prix bas ne rend pas un logiciel plus « libre ». Ainsi, si vous redistribuez des copies de logiciels libres, vous pouvez aussi bien fixer un prix élevé que *rentrer dans vos frais*. La redistribution de logiciels libres est une activité honorable et totalement légale; si vous l'exercez, vous pouvez très bien en tirer du profit.

Le logiciel libre est le projet de toute une communauté, et tous ceux qui en dépendent devraient chercher à soutenir la communauté. Pour un distributeur, la manière d'y contribuer est de donner une part de ses bénéfices à la [FSF](#) ou à un autre projet de développement de logiciels libres. En soutenant des équipes de développement, vous faites avancer le logiciel libre.

La distribution de logiciels libres est une chance de rassembler des fonds pour le développement. Ne la laissez pas passer !

Pour faire un don, vous devez avoir des fonds en réserve. Si vous fixez un prix trop faible, vous n'aurez pas de réserve pour soutenir le développement.

Est-ce qu'un prix plus élevé lésera des utilisateurs ?

Certains s'inquiètent parfois qu'un prix trop élevé mette le logiciel libre hors de portée des utilisateurs n'ayant pas beaucoup de moyens financiers. En ce qui concerne les [logiciels propriétaires](#), c'est exactement ce que fait un prix élevé, mais c'est différent pour le logiciel libre.

La différence est que le logiciel libre tend naturellement à se répandre, et qu'il y a différentes façons de se le procurer.

Les rapaces du logiciel vendent leur âme au diable pour vous empêcher d'utiliser un programme propriétaire sans payer le prix fort. Si le prix est élevé, il devient vraiment difficile pour certains utilisateurs d'utiliser le programme.

Avec le logiciel libre, les utilisateurs n'ont pas à payer la distribution pour utiliser le logiciel. Ils peuvent copier le programme à partir de la copie d'un ami, ou avec son aide s'il a accès au réseau. Plusieurs utilisateurs peuvent également se cotiser pour acheter un CD-ROM et installer le logiciel chacun à son tour. Un prix élevé n'est pas un obstacle majeur si le logiciel est libre.

Est-ce qu'un prix plus élevé découragera l'utilisation du logiciel libre ?

Une autre inquiétude est souvent exprimée à propos de la popularité du logiciel libre. Certains pensent qu'un prix élevé réduira le nombre d'utilisateurs ou qu'un prix faible encouragera certainement ces derniers.

C'est vrai dans le cas d'un logiciel propriétaire — mais c'est différent dans le cas d'un logiciel libre. Il y a tellement de façons d'avoir des copies que le prix payé en échange de la distribution a beaucoup moins d'effet sur la popularité.

À longue échéance, le nombre d'utilisateurs du logiciel libre est déterminé par *les capacités du logiciel libre* et par sa simplicité d'utilisation. De nombreux utilisateurs continueront à utiliser des logiciels propriétaires si le logiciel libre ne peut pas faire tout ce qu'ils veulent. Ainsi, si nous voulons faire augmenter le nombre d'utilisateurs à longue échéance, nous devons avant tout *développer plus de logiciels libres*.

La façon la plus directe est d'écrire vous-même des [logiciels libres](#) ou des [manuels](#) qui manquent. Mais si vous assurez la distribution plutôt que la création, la meilleure façon de nous aider est de lever des fonds pour aider les autres à en écrire.

Les mots « vendre des logiciels » peut aussi induire en erreur

Stricto sensus, « vendre » signifie échanger des biens contre de l'argent. Vendre une copie d'un logiciel libre est légal, et nous encourageons cette pratique.

Cependant, quand les gens pensent à « [vendre des logiciels](#) », ils l'imaginent habituellement de la même manière que la plupart des entreprises : rendre le logiciel propriétaire plutôt que libre.

Alors à moins que vous ne soyez prêts à faire des distinctions précises, comme le fait cet article, nous vous suggérons d'éviter le terme « vendre des logiciels » et de choisir un autre vocabulaire à la place. Par exemple, vous pourriez dire « distribuer des logiciels libres contre rémunération », ce qui lève toute ambiguïté.

Prix élevés ou bas et la GNU GPL

En dehors d'une seule situation spécifique, la [Licence Publique Générale GNU](#) (GNU GPL) n'a pas d'exigences en ce qui concerne le prix que vous pouvez demander pour la distribution d'un logiciel libre. Vous pouvez ne rien demander, ou alors un centime, un franc, un million de francs. Cela ne dépend que de vous, de l'offre et de la demande, alors ne venez pas vous plaindre si personne ne veut payer un million de francs pour une copie.

La seule exception est le cas où les binaires sont distribués sans le code source complet. Ceux qui font cela sont obligés par la GNU GPL de donner le code source sur toutes demandes ultérieures. Sans limite fixée au prix du code source, ils peuvent fixer un prix trop élevé pour que quelqu'un puisse payer (un million de francs, par exemple), et ainsi prétendre distribuer le code source alors qu'ils le dissimulent. Ainsi, nous devons dans ce cas limiter le prix du code source pour assurer la liberté de l'utilisateur. Cependant, dans des situations normales il n'y a pas de telle justification pour limiter le prix de distribution, alors nous ne le faisons pas.

Parfois des entreprises, dont les activités franchissent la limite de ce que la GNU GPL permet, implorent des autorisations, en disant qu'ils «ne vont pas faire payer les logiciels GNU», ou d'autres choses du même style. Cela ne les mènera pas loin. Le logiciel libre c'est avant tout la liberté, et faire respecter la GPL c'est défendre cette liberté. Quand nous défendons la liberté des utilisateurs, nous ne nous occupons pas de problèmes annexes comme le prix de la distribution. La liberté est la question, la seule et l'unique.

Copyright © 1996, 1997, 1998, 2001, 2007 Free Software Foundation, Inc.,

51 Franklin St, Fifth Floor, Boston, MA 02110, USA

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Benjamin Drieu.

Révision : trad-gnu@april.org

Dernière mise à jour : Date: 2009/01/22 21:28:10

Traductions de cette page

[العربية](#) [ar] [български](#) [bg] [Català](#) [ca] [Česky](#) [cs] [Deutsch](#) [de] [English](#) [en] [Español](#) [es] [Français](#) [fr]
[Bahasa Indonesia](#) [id] [Italiano](#) [it] [日本語](#) [ja] [മലയാളം](#) [ml] [Nederlands](#) [nl] [Polski](#) [pl]
[português do Brasil](#) [pt-br] [Română](#) [ro] [Русский](#) [ru] [Српски](#) [sr] [简体中文](#) [zh-cn] [繁體中文](#) [zh-tw]

Logiciels et manuels libres

La [Licence pour la Documentation Libre GNU](#) (GNU Free Documentation License)

Le plus grand défaut des systèmes d'exploitation libres n'est pas dans le logiciel, mais dans le manque de bons manuels libres que nous pouvons y inclure. Beaucoup de nos programmes les plus importants ne sont pas fournis avec des manuels complets. La documentation est une partie essentielle de tout logiciel; quand un logiciel libre important n'est pas fourni avec un manuel libre, c'est un manque majeur. Aujourd'hui, nous avons de nombreux manques importants.

Il y a de nombreuses années, j'ai voulu essayer d'apprendre Perl. J'avais une copie d'un manuel libre, mais je ne l'ai pas trouvé facile d'accès. Lorsque j'ai demandé à des utilisateurs de Perl s'il existait une alternative, ils me dirent qu'il y avait de meilleurs manuels d'introduction, mais que ceux-ci n'étaient pas libres.

Mais pourquoi cela ? Les auteurs de ces bons manuels les ont écrit pour O'Reilly Associates, qui les publie sous des termes restrictifs (pas de copie, pas de modification, les sources ne sont pas disponibles). Cela les exclut de la communauté du logiciel libre.

Ce n'était pas la première fois que ce genre de choses se produisait, et (malheureusement pour notre communauté) c'en est loin d'être fini. Les éditeurs de manuels propriétaires ont encouragé un grand nombre d'auteurs à restreindre leurs manuels depuis. J'ai souvent entendu un utilisateur de GNU me parler d'un manuel qu'il était en train d'écrire et avec lequel il comptait aider le projet GNU, puis décevoir mes espoirs en expliquant qu'il avait signé un contrat avec un éditeur qui restreindrait son manuel de telle manière que nous ne pourrions pas l'utiliser.

Étant donné la rareté de bons rédacteurs en langue anglaise parmi les programmeurs, nous ne pouvons pas nous permettre de perdre des manuels de cette manière.

L'intérêt d'une documentation libre (tout comme pour un logiciel libre) est la liberté, pas le prix. Le problème avec ces manuels n'était pas que O'Reilly Associates vende les versions imprimées de ses manuels, ce qui est bon en soi. (La Free Software Foundation [vend aussi des impressions](#) des [manuels GNU](#)). Mais les manuels GNU sont disponibles sous forme de code source, alors que ces manuels-là ne sont disponibles que sous forme imprimée. Les manuels de GNU sont distribués avec la permission de les copier et de les modifier; mais pas ces manuels de Perl. Ces restrictions sont le problème.

Les conditions à remplir pour un manuel libre sont à peu près les mêmes que pour un logiciel; il s'agit de donner à tous les utilisateurs certaines libertés. La redistribution (y compris une distribution commerciale) doit être autorisée afin que le manuel accompagne chaque copie du programme, de manière électronique ou imprimée. Permettre les modifications est crucial aussi.

En règle générale, je ne crois pas qu'il soit essentiel que nous ayons la permission de modifier toutes sortes d'articles ou de livres. Les problèmes de l'écriture ne sont pas forcément les mêmes que ceux du logiciel. Par exemple, je ne crois pas que vous ou moi devrions nous sentir obligés de donner la permission de modifier des articles tels que celui-ci, qui décrivent nos actions et nos positions.

Mais il y a une raison particulière pour laquelle la liberté de modifier des documentations libres traitant de logiciels libres est cruciale. Lorsque les programmeurs exercent leur droit de modifier un logiciel et d'ajouter ou de modifier des fonctionnalités, s'ils sont consciencieux, ils changeront aussi le manuel afin de pouvoir fournir une documentation précise et utilisable avec leur propre version du programme. Un manuel qui interdirait aux programmeurs d'être consciencieux et de finir leur travail, ou qui leur imposerait d'écrire un nouveau manuel à partir de zéro s'ils modifient le programme ne répond pas aux besoins de notre communauté.

Même si un refus total des modifications est inacceptable, quelques limites sur la manière de modifier une documentation ne pose pas de problème. Par exemple, il est normal d'avoir des injonctions de préserver la notice de copyright originale, les termes de distribution ou la liste des auteurs. Il n'y a pas non plus de problème à demander que les versions modifiées incluent une notice expliquant qu'il s'agit d'une version modifiée, et même d'avoir des sections entières qui ne puissent ni être supprimées ni être modifiées,

du moment qu'il ne s'agit pas de sections ayant trait à des sujets techniques (certains manuels GNU en ont).

Ce type de restrictions n'est pas un problème, car d'une manière pratique elles n'empêchent pas le programmeur consciencieux d'adapter le manuel pour correspondre au programme modifié. En d'autres termes, elles n'empêchent pas la communauté du logiciel libre de faire son œuvre à la fois sur le programme et sur le manuel.

De toutes façons, il doit être possible de modifier toute la partie *technique* du manuel ; sinon ces restrictions bloquent la communauté, le manuel n'est pas libre, et nous avons besoin d'un autre manuel.

Malheureusement, il est souvent difficile de trouver quelqu'un pour écrire un autre manuel quand un manuel propriétaire existe déjà. L'obstacle est que la plupart des utilisateurs pensent qu'un manuel propriétaire est suffisamment bon, alors ils ne ressentent pas le besoin d'écrire un manuel libre. Il ne voient pas qu'un système d'exploitation libre a une fissure qui nécessite un colmatage.

Pourquoi ces utilisateurs pensent-ils que les manuels propriétaires sont suffisamment bons ? La plupart ne se sont pas penchés sur le problème. J'espère que cet article sera utile dans ce sens.

D'autres utilisateurs considèrent les manuels propriétaires acceptables pour la même raison qu'énormément de personnes considèrent le logiciel propriétaire acceptable : ils pensent en termes purement pratiques, sans mettre la liberté en compte. Ces personnes sont attachées à leurs opinions, mais comme ces opinions découlent de valeurs qui n'incluent pas la liberté, ils ne sont pas un modèle pour ceux d'entre nous qui s'attachent à la liberté.

Je vous encourage à parler de ce problème autour de vous. Nous continuons à perdre des manuels au profit d'éditions propriétaires. Si nous faisons savoir que les manuels propriétaires ne sont pas suffisants, peut-être que la prochaine personne qui voudra aider GNU en écrivant de la documentation réalisera, avant qu'il soit trop tard, qu'elle devra avant tout la rendre libre.

Nous pouvons de plus encourager les éditeurs à vendre des manuels libres et dénués de copyright au lieu de manuels propriétaires. Une façon de le faire est de vérifier les termes de distribution de manuels avant de les acheter, et de préférer les manuels sans copyright à des manuels copyrightés.

[Note : Nous maintenons une [page Web listant les livres publiés, par d'autres éditeurs, en tant que documentation libre](#)]

Copyright © 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 Free Software Foundation, Inc.,
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

*Traduction : Benjamin Drieu.
Révision : trad-gnu@april.org*

Dernière mise à jour : 2009/01/02 09:27:23

Traductions de cette page

[български](#) [bg] [Català](#) [ca] [Česky](#) [cs] [Deutsch](#) [de] [English](#) [en] [Español](#) [es] [Français](#) [fr] [עברית](#) [he]
[Hrvatski](#) [hr] [Bahasa Indonesia](#) [id] [Italiano](#) [it] [日本語](#) [ja] [Nederlands](#) [nl] [Polski](#) [pl]
[português do Brasil](#) [pt-br] [Română](#) [ro] [Русский](#) [ru] [српски](#) [sr] [தமிழ்](#) [ta]

The Free Software Song

To the melody of Sadi Moma.

Join us now and share the so - ftware You'll be free ha - ckers, you'll be
Hoar-ders may get piles of mo - ney, That is true, ha - ckers, that is
When we have e - nough free so - ftware At our call, ha - ckers, at our

free. Join us now and share the so - ftware You'll be - free ha - ckers, you'll be free.
true. But they can - not help their neigh-bors That's not good, ha - ckers, that's not good.
call, We'll kick out those dir - ty li - cen - ses E - ver more, ha - ckers, E - ver more.

Partition créée par David Madore avec le logiciel GNU LilyPond.

*Texte de Richard Stallman, adaptation française (ci-dessous) par HackAngel
d'après la traduction de Emily Kanzari et Richard Stallman*

Join us now and share the software;
You'll be free, hackers, you'll be free.
Join us now and share the software;
You'll be free, hackers, you'll be free.

Hoarders may get piles of money,
That is true, hackers, that is true.
But they cannot help their neighbors;
That's not good, hackers, that's not good.

When we have enough free software
At our call, hackers, at our call,
We'll kick out those dirty licenses
Ever more, hackers, ever more.

Join us now and share the software;
You'll be free, hackers, you'll be free.
Join us now and share the software;
You'll be free, hackers, you'll be free.

Rejoignez-nous maintenant et partagez le logiciel;
Vous serez libres, hackers, vous serez libres.
Rejoignez-nous maintenant et partagez le logiciel;
Vous serez libres, hackers, vous serez libres.

Les avares peuvent empiler de l'argent;
Cela est vrai, hackers, cela est vrai.
Mais ils ne peuvent pas aider leurs voisins;
Ce n'est pas bon, hackers, ce n'est pas bon.

Quand nous aurons assez de logiciels libres;
À notre appel, hackers, à notre appel.
Nous expulserons ces sales licences;
Pour toujours, hackers, pour toujours.

Rejoignez-nous maintenant et partagez le logiciel;
Vous serez libres, hackers, vous serez libres.
Rejoignez-nous maintenant et partagez le logiciel;
Vous serez libres, hackers, vous serez libres.

Vous pouvez aussi télécharger [la partition](#) de la chanson transcrite à l'aide de [GNU LilyPond](#).
La [source](#) LilyPond de la partition est aussi disponible pour téléchargement.

Avis

La chanson sur cette page a été obtenue dans les [archives de courriels](#) du projet GNU.

La Fondation pour le logiciel libre ne détient pas le copyright sur cette chanson.

Sadi Moma

Voici une description de la chanson de Sadi Moma, à l'origine une chanson de danse bulgare, sous une forme ad-hoc de notation utilisant des lettres comme notes. Une tiret signifie que la note précédente continue; il y a sept temps par mesure.

D-CB-A- B-CBAG- G--A--B C--B-BD A--A--- CDCB---
D-CB-A- B-CBAG- G--A--B C--B-BD A--A--- A-----

Voici la seconde voix (ne se trouve pas dans le fichier).

B-AG-D- G-GG-D- G--G--- G--G-G- D--D--- G-----
B-AG-D- G-GG-D- G--G--- G--G-G- D--D--- E-----

Copyright © 1993, 2001, 2002, 2003, 2004, 2005 2006 Richard Stallman,

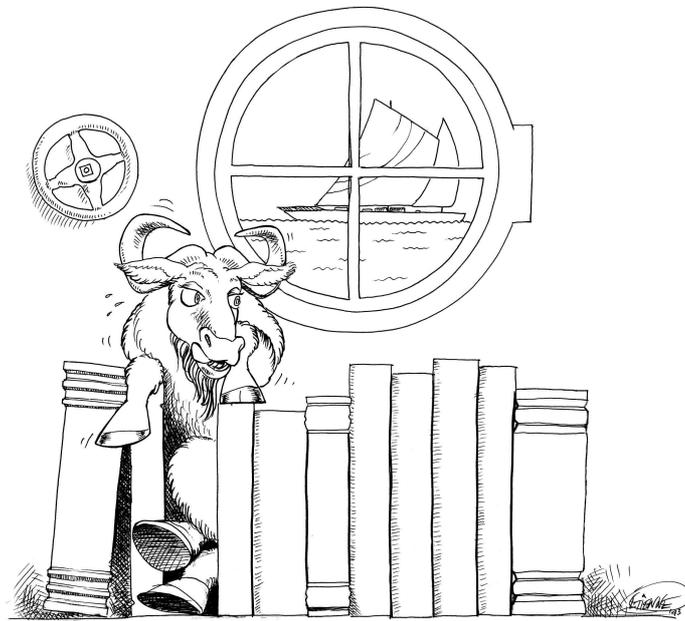
La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Dernière mise à jour : 2000/23/04

Traductions de cette page

[Català](#) [Deutsch](#) [English](#) [Español](#) [Français](#) [한국어](#) [简体中文](#) [繁體中文](#)

Livre II: Copyright, Copyleft, et Brevets



Cette notice est valable pour l'ensemble des articles, relativement:

Copyright © Richard Stallman,

ou Copyright © Free Software Foundation, Inc.,

La reproduction exacte et la distribution intégrale de chaque article est permise sur n'importe quel support d'archivage, pourvu que sa notice soit préservée.

Le droit de lire

par *Richard Stallman*

Cet article a été publié dans la parution de février 1997 de Communications of the ACM (volume 40, numéro 2).

(extrait de « The Road to Tycho », une collection d'articles sur les antécédents de la Révolution lunaire, publiée à Luna City en 2096)

Pour Dan Halbert, la route vers Tycho commença à l'université — quand Lissa Lenz lui demanda de lui prêter son ordinateur. Le sien était en panne, et à moins qu'elle puisse en emprunter un autre, elle échouerait son projet de mi-session. Il n'y avait personne d'autre à qui elle osait demander, à part Dan.

Ceci posa un dilemme à Dan. Il se devait de l'aider — mais s'il lui prêtait son ordinateur, elle pourrait lire ses livres. À part le fait que vous pouviez aller en prison pour plusieurs années pour avoir laissé quelqu'un lire vos livres, l'idée même le choqua au départ. Comme à tout le monde, on lui avait enseigné dès l'école primaire que partager des livres était malicieux et immoral — une chose que seuls les pirates font.

Et il était improbable que la SPA — la Software Protection Authority — manquerait de le pincer. Dans ses cours sur les logiciels, Dan avait appris que chaque livre avait un moniteur de copyright qui rapportait quand et où il était lu, et par qui, à la Centrale des licences. (Elle utilisait ces informations pour attraper les lecteurs pirates, mais aussi pour vendre des renseignements personnels à des détaillants.) La prochaine fois que son ordinateur serait en réseau, la Centrale des licences se rendrait compte. Dan, comme propriétaire d'ordinateur, subirait les punitions les plus sévères — pour ne pas avoir tout tenté pour éviter le délit.

Bien sûr, Lissa n'avait pas nécessairement l'intention de lire ses livres. Elle pourrait ne vouloir l'ordinateur que pour écrire son projet. Mais Dan savait qu'elle venait d'une famille de classe moyenne et qu'elle arrivait difficilement à payer ses frais de scolarité, sans compter ses frais de lecture. Lire les livres de Dan pourrait être sa seule façon d'obtenir son diplôme. Il comprenait cette situation; lui-même avait eu à emprunter pour payer pour tous les articles scientifiques qu'il avait eu à lire. (10% de ces frais allaient aux chercheurs qui écrivaient ces articles; puisque Dan visait une carrière académique, il pouvait espérer que si ses propres articles scientifiques étaient souvent lus, il gagnerait un revenu suffisant pour rembourser sa dette).

Par la suite, Dan apprendrait qu'il y eut un temps où n'importe qui pouvait aller à la bibliothèque et lire des articles de journaux, et même des livres, sans avoir à payer. Il y avait des universitaires indépendants qui lisaient des milliers de pages sans subventions des bibliothèques gouvernementales. Mais dans les années 1990, les éditeurs aussi bien commerciaux qu'à but non lucratif avaient commencé à facturer l'accès. En 2047, les bibliothèques offrant un accès public gratuit à la littérature scientifique n'étaient qu'un pâle souvenir.

Il y avait des façons, bien sûr, de contourner la SPA et la Centrale des licences. Elles étaient elles-mêmes illégales. Dan avait eu un compagnon de classe dans son cours sur les logiciels, Frank Martucci, qui avait obtenu un outil illégal de débogage, et l'avait utilisé pour outrepasser le code du moniteur de copyright quand il lisait des livres. Mais il en avait parlé à trop d'amis, et l'un d'eux l'a dénoncé auprès de la SPA pour une récompense (des étudiants criblés de dettes pouvaient facilement être tentés par la trahison). En 2047, Frank était en prison, non pas pour lecture pirate, mais pour possession d'un débogueur.

Dan apprendrait plus tard qu'il y eut un temps où n'importe qui pouvait posséder des outils de débogage. Il y avait même des outils de débogage disponibles gratuitement sur des CD ou qu'on pouvait télécharger du Net. Mais des usagers ordinaires commencèrent à s'en servir pour outrepasser les moniteurs de copyright, et finalement un juge a décidé que c'était devenu leur principale utilisation en pratique. Ceci voulait dire qu'ils étaient illégaux; les développeurs de ces débogueurs furent envoyés en prison.

Les programmeurs avaient encore besoin d'outils pour déboguer, bien sûr, mais les vendeurs de débogueurs en 2047 ne distribuaient que des copies numérotées, et seulement à des programmeurs officiellement licenciés et soumis. Le débogueur que Dan utilisait dans son cours sur les logiciels était gardé derrière un garde-barrière spécial afin qu'il ne puisse servir que pour les exercices du cours.

Il était aussi possible de contourner les moniteurs de copyright en installant un noyau système modifié. Dan apprendrait finalement l'existence de noyaux libres, et même de systèmes d'exploitation entièrement libres, qui avaient existé au tournant du siècle. Mais non seulement étaient-ils illégaux, comme les débogueurs, mais vous ne pouviez en installer un, si vous en aviez un, sans connaître le mot de passe de l'utilisateur superviseur de votre ordinateur. Or, ni le FBI ni l'Aide technique Microsoft ne vous le révélerait.

Dan conclut qu'il ne pouvait simplement prêter son ordinateur à Lissa. Mais il ne pouvait refuser de l'aider, car il l'aimait. Chaque chance de lui parler le remplissait d'aise. Et le fait qu'elle l'avait choisi pour demander de l'aide pouvait signifier qu'elle l'aimait aussi.

Dan résolut le dilemme en faisant une chose encore plus impensable — il lui prêta l'ordinateur, et lui dit son mot de passe. Ainsi, si Lissa lisait ses livres, la Centrale des licences penserait que c'était lui qui les lisait. C'était quand même un délit, mais la SPA ne s'en rendrait pas compte automatiquement. Ils ne s'en rendraient compte que si Lissa le dénonçait.

Bien sûr, si l'école devait un jour apprendre qu'il avait donné son propre mot de passe à Lissa, ce serait la fin de leurs études, peu importe ce à quoi le mot de passe aurait servi. La politique de l'école était que toute interférence avec ses mécanismes de surveillance de l'utilisation des ordinateurs par les étudiants était punissable. Il n'importait pas qu'aucun mal n'ait été fait — l'offense était de se rendre difficile à surveiller par les administrateurs. Ils supposaient que ça signifiait que vous faisiez quelque chose d'autre qui était interdit, et ils n'avaient pas besoin de savoir de quoi il s'agissait.

Les étudiants n'étaient habituellement pas expulsés pour cela — pas directement. Ils étaient plutôt bannis des systèmes informatiques de l'école, et échouaient inévitablement leurs cours.

Plus tard, Dan apprendrait que ce genre de politique n'a commencé dans les universités que dans les années 1980, quand des étudiants commencèrent à être nombreux à utiliser des ordinateurs. Avant, les universités avaient une approche différente au sujet de la discipline auprès des étudiants; elles punissaient des activités qui causaient du tort, et non pas simplement celles qui soulevaient des doutes.

Lissa ne dénonça pas Dan à la SPA. La décision de Dan de l'aider mena à leur mariage, et les amena aussi à remettre en question ce qu'on leur avait enseigné durant leur enfance au sujet du piratage. Le couple se mit à lire sur l'histoire du copyright, sur l'Union soviétique et ses restrictions sur la copie, et même sur la Constitution originale des États-Unis. Ils déménagèrent à Luna, où ils trouvèrent d'autres gens qui comme eux avaient pris leurs distances par rapport au bras long de la SPA. Quand la révolte de Tycho commença en 2062, le droit universel de lire devint bientôt un de ses buts principaux.

Note de l'auteur

Cette note a été mise à jour en 2007.

C'est aujourd'hui même qu'on se bat pour le droit de lire. Même si cela pourrait prendre 50 ans pour que notre façon de vivre actuelle s'efface dans l'obscurité, la plupart des lois et pratiques décrites précédemment ont déjà été proposées ; plusieurs ont été décrétées légalement aux USA et ailleurs. Aux USA, le DMCA (ndt : Loi sur le droit d'auteur du millénaire numérique) de 1998 a établi la base légale pour restreindre la lecture et le prêt de livres numérisés (et d'autres données également). L'Union Européenne a imposé des restrictions similaires dans une directive de 2001 concernant le copyright. En France, avec la loi DADVSI, la simple possession d'une copie de DeCSS, le programme libre permettant de déchiffrer une vidéo sur un DVD, est un délit.

En 2001, le sénateur Hollings (disposant de fonds Disney) a proposé une loi appelée SSSCA qui exigerait que chaque nouvel ordinateur ait des fonctionnalités obligatoires de restriction des copies, restrictions que l'utilisateur ne pourrait pas contourner. Après le Clipper Chip et d'autres propositions similaires de « key-escrow » émanant du gouvernement américain, ceci démontre une tendance à long terme : les systèmes informatiques sont de plus en plus configurés pour donner à des absents un puissant contrôle sur les personnes qui utilisent effectivement ces systèmes informatiques. Le SSSCA a depuis été rebaptisé CBDTPA (on pourrait penser cela comme le « Consume But Don't Try Programming Act » — la loi « Consommez mais n'essayez pas de programmer »).

Les Républicains prirent le contrôle du Sénat américain peu de temps après. Ils sont moins liés à

Hollywood que les Démocrates, aussi n'ont-ils pas pressé ces propositions. Maintenant que les Démocrates sont de retour aux commandes, le danger est encore une fois plus grand.

En 2001, les USA ont commencé à tenter d'utiliser la proposition de traité « Free Trade Area of the Americas (FTAA) » pour imposer les mêmes règles à tous les pays de l'hémisphère Ouest. Le FTAA est l'un de ces traités de soi-disant « libre commerce », mais réellement conçu pour donner au monde du business un pouvoir croissant sur les gouvernements démocratiques. L'imposition de lois du genre DMCA est typique de cet esprit. Le FTAA a été effectivement tué par Lula, président du Brésil, qui a rejeté la condition du DMCA et d'autres.

Depuis lors, les États-Unis ont imposé des conditions similaires à des pays comme l'Australie ou le Mexique, par le biais d'accords de « libre échange » bilatéraux, et à des pays tel que le Costa Rica par le biais du CAFTA (N.d.T. : Accord de libre échange d'Amérique Centrale). Le président Correa de l'Équateur a refusé de signer l'accord de « libre échange », mais l'Équateur a adopté quelque chose de similaire au DMCA en 2003. La nouvelle constitution de l'Équateur peut fournir une opportunité de s'en débarrasser.

Une des idées dans cette histoire n'avait en fait pas été proposée jusqu'en 2002. C'est l'idée que le FBI et Microsoft garderaient le mot de passe administrateur (root) de votre ordinateur et ne vous autoriseraient pas à l'obtenir.

Les instigateurs de ce dessein lui ont donné des noms tels que « informatique de confiance » et « Palladium ». Nous l'appelons « [informatique déloyale](#) », car le résultat est de faire obéir votre ordinateur à des sociétés plutôt qu'à vous. Ceci a été mis en œuvre en 2007 dans [Windows Vista](#) ; nous nous attendons à ce que Apple fasse quelque chose de semblable. Dans cette construction, c'est le fabriquant qui conserve le code secret, mais le FBI aurait pu de peine à l'obtenir.

Ce que conserve Microsoft n'est pas exactement un mot de passe au sens traditionnel du terme ; personne ne le tape sur son clavier. C'est plutôt une signature et une clé de chiffrement qui correspondent à une seconde clé stockée sur votre ordinateur dans un endroit que vous ne pouvez lire. Ceci permet à Microsoft et potentiellement tout site Web coopérant avec lui, le contrôle sur ce que l'utilisateur peut faire avec son propre ordinateur.

Vista donne aussi à Microsoft des pouvoirs supplémentaires ; par exemple, Microsoft peut forcer l'installation de mises à jour et il peut ordonner à toutes les machines fonctionnant avec Vista de refuser d'exécuter un pilote de périphérique. Le but principal de toutes ces restrictions dans Vista est de faire des DRM que les utilisateurs ne peuvent pas outrepasser.

La SPA, qui veut en fait dire Software Publisher's Association, a été remplacée dans ce rôle de police par la BSA (Business Software Alliance). Ce n'est pas aujourd'hui une force policière officielle. De façon officieuse, elle se comporte ainsi. Utilisant des méthodes rappelant celles de l'ex-Union Soviétique, elle invite les gens à faire de la délation à l'endroit de leurs collègues et amis; une campagne de terreur orchestrée par la BSA en 2001 en Argentine a dévoilé des menaces selon lesquelles les gens partageant du logiciel pourraient être jetés en prison.

À l'époque où cette histoire a été écrite, la SPA menaçait de petits fournisseurs d'accès à Internet, en exigeant qu'ils lui permettent de surveiller tous les usagers. La plupart des fournisseurs capitulèrent lorsqu'ils furent menacés, parce qu'ils n'avaient pas les moyens de contre-attaquer en cour. (Atlanta Journal-Constitution, 1er octobre 1996, page D3.) Au moins un fournisseur, Community ConneXion à Oakland en Californie, a refusé la demande et a été réellement poursuivi. Plus tard, la SPA a laissé tomber la plainte, mais elle a obtenu le DMCA, qui lui donne le pouvoir qu'ils recherchaient.

Les politiques universitaires de sécurité décrites précédemment ne sont pas imaginaires. Par exemple, un ordinateur dans une université de la région de Chicago affiche le message suivant quand on s'y branche (les guillemets sont dans l'original — ce qui suit est une traduction) :

« Ce système est réservé aux usagers autorisés. Les individus qui utilisent ce système informatique sans autorisation ou au delà de leur autorisation pourront faire l'objet d'une surveillance et d'un enregistrement par le personnel de toutes leurs activités sur ce système. Lors de la surveillance d'individus utilisant le système inadéquatement, ou lors d'activités d'entretien du système, les activités d'usagers autorisés pourraient aussi être surveillées. Quiconque utilise

ce système consent expressément à une telle surveillance et est avisé que si cette surveillance révèle des indices d'une possible activité illégale ou violation des règlements de l'Université, le personnel du système peut fournir ces indices aux autorités de l'Université et/ou aux forces de l'ordre ».

Il s'agit d'une approche intéressante face au Quatrième amendement : faire pression sur presque tout le monde pour qu'il accepte d'avance de renoncer aux droits qu'il leur accorde.

Références

- Le « livre blanc » de l'administration : Information Infrastructure Task Force, Intellectual Property and the National Information Infrastructure: The Report of the Working Group on Intellectual Property Rights (1995).
- [Une explication du livre blanc : The Copyright Grab](#), Pamela Samuelson, Wired, Jan. 1996
- [Sold Out](#), James Boyle, New York Times, 31 March 1996
- Public Data or Private Data, Washington Post, 4 Nov 1996. Nous avons un lien vers l'article mais le Washington Post a décidé de faire payer la lecture d'articles sur le web, et nous avons donc choisi de supprimer le lien.
- [Union for the Public Domain](#)— une organisation dont le but est de résister et de renverser la surextension des pouvoirs du copyright et des brevets.

Autres textes à lire

- [Philosophie du projet GNU](#)
- [Copy Protection: Just Say No](#), publié dans Computer World.

La [note de l'auteur](#) parle de la bataille pour le droit de lire, et de la surveillance électronique. La bataille a déjà commencé, voici deux liens vers des articles présentant les technologies déjà développées pour interdire le droit à la lecture :

- [Electronic Publishing](#) : Un article dont le sujet est la distribution de livres sous forme électronique, et les questions de copyright affectant le droit de lire une copie.
- [Books inside Computers](#) Les logiciels pour contrôler qui peut lire des livres et des documents sur un PC.

Copyright © 1996 Richard Stallman

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Pierre Sarrazin.

Révision : trad-gnu@april.org

Dernière mise à jour : 2009/01/21 21:28:07

Traductions de cette page

[български](#) [bg] [Česky](#) [cs] [Deutsch](#) [de] [English](#) [en] [Español](#) [es] [Suomi](#) [fi] [Français](#) [fr] [עברית](#) [he] [Magyar](#) [hu]
[Italian](#) [it] [日本語](#) [ja] [한국어](#) [ko] [Nederlands](#) [nl] [Polski](#) [pl] [português do Brasil](#) [pt-br] [Русский](#) [ru] [Slovensko](#) [sl] [Svenska](#) [sv]

Mauvaise interprétation du droit d'auteur

Quelque chose d'étrange et de dangereux est en train d'arriver dans la loi sur le droit d'auteur. Dans la Constitution américaine, le droit d'auteur existe au bénéfice des utilisateurs — ceux qui lisent les livres, écoutent de la musique, regardent des films ou utilisent des logiciels — non pour l'intérêt des éditeurs ou des auteurs. Cependant, alors que les gens tendent de plus en plus à rejeter et transgresser le droit d'auteur qui leur est imposé « pour leur propre bénéfice », le gouvernement ajoute plus de restrictions, et essaie d'effrayer le public pour qu'il rentre dans le rang, avec de nouvelles sanctions sévères.

Comment la politique de droit d'auteur en est-elle arrivée à être diamétralement opposée à son intention déclarée ? Et comment pouvons-nous la ramener en phase avec cette intention ? Pour comprendre, nous devons commencer par regarder aux sources de la loi sur le droit d'auteur des États-Unis : la Constitution américaine.

Le droit d'auteur dans la Constitution américaine

Quand la Constitution américaine fut rédigée, l'idée que les auteurs seraient autorisés à un monopole sur le droit d'auteur fut proposée — et rejetée. Les fondateurs de notre pays adoptèrent un principe différent, que le droit d'auteur n'est pas un droit naturel des auteurs, mais une concession artificielle qui leur est faite dans l'intérêt du progrès. La Constitution donne la permission d'un système de droit d'auteur avec ce paragraphe (Article I, Section 8) :

« [Le Congrès aura le pouvoir] De promouvoir le progrès de la science et des arts utiles, en assurant pour un temps limité, aux auteurs et inventeurs, un droit exclusif sur leurs écrits et découvertes respectifs ; »

La Cour Suprême a affirmé à maintes reprises que la promotion du progrès signifie que les travaux couverts par le droit d'auteur bénéficient aux utilisateurs. Par exemple, dans *Fox Film contre Doyal*, la cour a statué :

« Le seul intérêt des États-Unis et l'objet premier de conférer le monopole [du droit d'auteur] reposent sur les bénéfices généraux tirés du travail des auteurs par le public. »

Cette décision fondamentale explique pourquoi le droit d'auteur n'est pas **requis** par la Constitution, seulement **permis** comme une option — et pourquoi il est supposé durer un « temps limité ». Si le droit d'auteur était un droit naturel, quelque chose que les auteurs ont parce qu'ils le méritent, rien ne pourrait justifier que ce droit se termine après un certain temps, pas plus que la maison de quiconque deviendrait propriété publique après un certain laps de temps après sa construction.

Le « marchandage du droit d'auteur »

Le système du droit d'auteur fonctionne en fournissant des privilèges et donc bénéficie aux éditeurs et aux auteurs; mais il ne le fait pas pour leur intérêt. Il le fait plutôt pour modifier leur comportement : pour donner une motivation aux auteurs d'écrire plus et de publier plus. En fait, le gouvernement utilise les droits naturels du public, en son nom, comme contrepartie d'un marché pour apporter au public plus d'œuvres publiées. Les érudits en droit appelle ce concept « le marchandage du droit d'auteur ». C'est comme l'achat par le gouvernement d'une autoroute ou d'un avion en utilisant l'argent du contribuable, excepté que le gouvernement utilise notre liberté au lieu de notre argent.

Mais le marchandage tel qu'il existe effectivement, est-il une bonne tractation pour le public ? Beaucoup d'autres alternatives de négociation sont possibles ; laquelle est la meilleure ? Chaque problème de la politique du droit d'auteur réside dans cette question. Si nous interprétons mal la nature de la question, nous aurons tendance à décider des solutions de façon incorrecte.

La Constitution autorise l'octroi du pouvoir du droit d'auteur aux auteurs. En pratique, les auteurs cèdent, en général, leurs droits aux éditeurs ; habituellement, ce sont les éditeurs, pas les auteurs, qui exercent

ce pouvoir et obtiennent la plupart des bénéfices, bien que les auteurs en obtiennent une petite part. Par conséquent, ce sont généralement les éditeurs qui font du lobbying pour augmenter le pouvoir du droit d'auteur. Pour refléter au mieux la réalité du droit d'auteur plutôt que son mythe, cet article se réfère aux éditeurs plutôt qu'aux auteurs comme les détenteurs de droits d'auteur. Il fait aussi référence aux utilisateurs d'œuvres couvertes par droits d'auteur comme « lecteurs », même si leur utilisation ne signifie pas forcément une lecture, car « les utilisateurs » est un terme vague et abstrait.

La première erreur : « établir l'équilibre »

Le marchandage du droit d'auteur place le public d'abord : le bénéfice pour les lecteurs est une fin en soi; les bénéfices (s'il y en a) pour les éditeurs sont seulement un moyen pour arriver à cette fin. Les intérêts des lecteurs et les intérêts des éditeurs sont qualitativement inégaux en priorité. La première étape dans la mauvaise interprétation au sujet du droit d'auteur est l'élévation des éditeurs au même rang d'importance que les lecteurs.

Il est souvent dit que la loi américaine sur le droit d'auteur entend « établir l'équilibre » entre les intérêts des éditeurs et des lecteurs. Ceux qui citent cette interprétation la présente comme une reformulation de la position de base fixée dans la Constitution; en d'autres termes, elle est supposée être équivalente au marchandage du droit d'auteur.

Mais les deux interprétations sont loin d'être équivalentes; elles sont différentes conceptuellement ainsi que dans leurs implications. Le concept de l'équilibre suppose que les intérêts des lecteurs et des éditeurs se différencient en importance seulement quantitativement, dans le « poids » que nous leur donnons, et dans les cas auxquels ils s'appliquent. Le terme « parties prenantes » est souvent utilisé pour formuler le problème de cette façon; il suppose que toutes sortes d'intérêts dans une décision politique sont d'égale importance. Cette vision des choses rejette la distinction qualitative entre les intérêts des lecteurs et des éditeurs qui est à l'origine de la participation du gouvernement au marchandage du droit d'auteur.

Les conséquences de cette altération ne sont pas négligeables, car la grande protection du public dans le marchandage du droit d'auteur — l'idée que les privilèges de droit d'auteur ne se justifient qu'au nom des lecteurs, jamais au nom des éditeurs — est abandonnée par l'interprétation de « l'équilibre ». Puisque l'intérêt des éditeurs est considéré comme une fin en soi, il peut justifier des privilèges de droit d'auteur; en d'autres termes, le concept de « l'équilibre » dit que les privilèges peuvent être justifiés en d'autre nom que celui du public.

En pratique, la conséquence du concept de « l'équilibre » est d'outrepasser la charge de la justification pour les changements dans la loi sur le droit d'auteur. Le marchandage du droit d'auteur met à la charge des éditeurs de convaincre les lecteurs de céder certaines libertés. Le concept de l'équilibre révoque cette charge, pour ainsi dire, car il n'y a généralement pas de doute que les éditeurs profiteraient de privilèges supplémentaires. Aussi, à moins que le préjudice pour les lecteurs ne soit avéré, suffisant pour « compenser » ce bénéfice, nous sommes amenés à conclure que les éditeurs sont autorisés à obtenir presque tous les privilèges qu'ils réclament.

Puisque l'idée « d'établir l'équilibre » entre les éditeurs et les lecteurs dénie la primauté des lecteurs à laquelle ils sont habilités, nous devons la rejeter.

Équilibrer contre quoi ?

Quand le gouvernement achète quelque chose pour le public, il agit pour le compte du public; sa responsabilité est d'obtenir la meilleure négociation possible — meilleure pour le public, pas pour l'autre partie prenante du contrat.

Par exemple, lorsqu'il signe des contrats avec des sociétés de construction pour construire des autoroutes, le gouvernement vise à dépenser le moins possible de l'argent public. Les agences du gouvernement font des appels d'offres pour faire baisser le prix.

En pratique, le prix ne peut être nul, car les entrepreneurs ne feront pas d'offres si basses. Bien qu'ils ne soient pas assujettis à une considération particulière, Ils ont les mêmes droits que les citoyens dans une société libre, y compris celui de refuser des contrats désavantageux; même l'offre la plus basse sera

largement suffisante pour un entrepreneur pour faire de l'argent. Donc, il y a en effet un équilibre, en quelque sorte. Mais il ne s'agit pas d'un équilibre délibéré de deux intérêts, chacun revendiquant une considération spéciale. Il s'agit d'un équilibre entre un objectif public et les forces du marché. Le gouvernement essaie d'obtenir pour les contribuables automobilistes la meilleure transaction qu'il peut dans le contexte d'une société libre et d'un marché libre.

Dans le marchandage du droit d'auteur, le gouvernement dépense notre liberté au lieu de notre argent. La liberté est plus précieuse que l'argent, aussi, la responsabilité du gouvernement pour dépenser notre liberté avec sagesse et parcimonie est par conséquent même plus grande que sa responsabilité pour dépenser notre argent. Les gouvernements ne doivent jamais mettre les intérêts des éditeurs sur le même pied d'égalité que la liberté publique.

Pas « équilibre » mais « compromis »

L'idée d'équilibre des intérêts des lecteurs contre les intérêts des éditeurs est la mauvaise voie pour juger de la politique du droit d'auteur, mais il y a en effet deux intérêts à soupeser : deux intérêts **des lecteurs**. Les lecteurs ont un intérêt dans leur propre liberté en utilisant des œuvres publiées; selon les circonstances, ils peuvent aussi avoir un intérêt en encourageant la publication par un système d'incitation.

Le mot « équilibre », dans les discussions sur les droits d'auteur, est devenu un raccourci pour l'idée « d'établir un équilibre » entre les lecteurs et les éditeurs. Pour cette raison, utiliser le mot « équilibre » en ce qui concerne les deux intérêts des lecteurs sèmerait la confusion — nous avons besoin d'un autre terme.

En général, quand une partie a deux buts en conflit partiellement, et ne peut atteindre complètement aucun d'eux, nous appelons cela un « compromis ». Par conséquent, plutôt que de parler « d'établir le bon équilibre » entre parties, nous parlerons de « trouver le bon compromis entre dépenser notre liberté et la garder ».

La seconde erreur : maximiser le rendement

La seconde erreur dans la politique du droit d'auteur consiste à adopter l'objectif de maximiser — pas seulement d'augmenter — le nombre d'œuvres publiées. Le concept erroné « d'établir l'équilibre » élevaient les éditeurs à parité avec les lecteurs; cette seconde erreur les place bien au-dessus des lecteurs.

Quand nous achetons quelque chose, nous n'achetons généralement pas toute la quantité en stock ou le modèle le plus cher. Au lieu de cela, nous conservons des fonds pour d'autres achats, en achetant seulement ce dont nous avons besoin pour chaque bien particulier, en choisissant un modèle de qualité suffisante plutôt que de la meilleure qualité. Le principe des rendements décroissants suggère que dépenser tout notre argent pour un bien particulier est vraisemblablement une répartition des ressources inefficace; nous choisissons généralement de garder de l'argent pour une autre utilisation.

Les rendements décroissants s'appliquent au droit d'auteur comme à n'importe quel autre achat. Les premières libertés que nous devrions négocier sont celles qui nous manqueront le moins, tout en donnant le plus grand encouragement à la publication. À mesure que nous négocions des libertés additionnelles qui touchent notre vie quotidienne, nous trouvons que chaque négociation est un plus grand sacrifice que le précédent, alors qu'elle apporte un plus petit accroissement de l'activité littéraire. Bien avant que l'accroissement ne devienne nul, nous pouvons tout à fait dire que cela ne vaut pas le prix de cet accroissement; nous pourrions alors établir un marchandage dont le résultat global serait d'augmenter le nombre de publications, mais pas en le poussant à l'extrême.

Accepter l'objectif de maximiser la publication rejette par avance toutes ces négociations plus sages et plus avantageuses — il dicte que le public doit céder presque toutes ses libertés pour utiliser des œuvres publiées, seulement pour un petit peu plus de publication.

La rhétorique de la maximisation

En pratique, l'objectif de maximiser la publication sans tenir compte de ce que cela coûte pour la liberté est soutenu par une rhétorique très répandue qui affirme que la copie publique est illégitime, injuste et

mauvaise intrinsèquement. Par exemple, les éditeurs appellent les gens qui copient « pirates », un terme diffamatoire conçu pour assimiler le partage de l'information avec votre voisin à l'attaque d'un bateau. (Ce terme diffamatoire était autrefois utilisé par les auteurs pour décrire les éditeurs qui avaient trouvé des voies légales pour publier des éditions non autorisées; son usage moderne par les éditeurs est presque tout le contraire). Cette rhétorique rejette directement la base constitutionnelle du droit d'auteur, mais se présente elle-même comme représentante de la tradition incontestée du système légal américain.

La rhétorique du « pirate » est typiquement acceptée car elle inonde les médias de sorte que peu de gens réalisent qu'elle est radicale. C'est efficace car, si la copie par le public est fondamentalement illégitime, nous ne pouvons pas objecter à la demande des éditeurs que nous abdiquions notre liberté pour ce faire. En d'autres termes, quand le public est défié de montrer pourquoi les éditeurs ne devraient pas recevoir de pouvoir supplémentaire, la raison la plus importante de toutes — « Nous voulons copier » — est disqualifiée d'avance.

Cela ne laisse aucun moyen d'argumenter contre l'augmentation de pouvoir du droit d'auteur excepté l'utilisation d'enjeux secondaires. De ce fait, l'opposition à des pouvoirs de droits d'auteur plus forts aujourd'hui, ne cite presque exclusivement que des enjeux secondaires, et n'ose jamais citer la liberté de distribuer des copies comme une valeur publique légitime.

En pratique, le principe de maximisation permet aux éditeurs pour argumenter que « Une certaine pratique réduit nos ventes — ou nous pensons que cela pourrait être le cas — aussi, nous supposons que cela diminue la publication en proportion inconnue, et par conséquent, cette pratique doit être interdite ». Nous sommes amenés à la conclusion inadmissible que le bien public se mesure à l'aune des ventes des éditeurs : Ce qui est bon pour General Media est bon pour les États-Unis.

La troisième erreur : maximiser le pouvoir des éditeurs

Une fois que les éditeurs auront obtenu l'assentiment à l'objectif politique de maximiser le rendement de la publication à n'importe quel coût, leur prochaine étape sera de conclure que cela requiert de leur donner le maximum de pouvoirs possible — en faisant couvrir par le droit d'auteur chaque utilisation imaginable d'une œuvre, ou en appliquant quelque autre instrument juridique tel que les licences « sous plastique » à l'effet équivalent. Cet objectif, qui entraîne l'abolition de « l'usage raisonnable » et le « le droit de la première vente », est martelé à tous les niveaux possibles du gouvernement, des États-Unis aux instances internationales.

Cette démarche est erronée car des règles strictes sur le droit d'auteur empêchent la création de nouvelles œuvres utiles. Par exemple, Shakespeare empruntait les intrigues de quelques unes de ses pièces d'autres pièces publiées quelques dizaines d'années auparavant, aussi, si la loi actuelle sur le droit d'auteur avait été en application, ses pièces auraient été illégales.

Même si nous voulions le plus grand taux de publication possible, sans tenir compte de ce que cela coûterait au public, maximiser le pouvoir des éditeurs serait un mauvais moyen d'y parvenir. Comme moyen de promouvoir le progrès, ce serait auto-destructeur.

Les conséquences de ces trois erreurs

La tendance actuelle dans la législation sur le droit d'auteur est de donner aux éditeurs des pouvoirs plus étendus sur des périodes plus longues. Le concept de base du droit d'auteur, tel qu'il ressort déformé de la suite des erreurs commises, offre rarement une base pour dire non. Les législateurs font semblant de s'intéresser à l'idée que le droit d'auteur doive servir le public, alors qu'en fait, il donne aux éditeurs tout ce qu'ils demandent.

Par exemple, voici ce que dit le sénateur Hatch en présentant S. 483, un projet de loi de 1995 pour augmenter le bail du droit d'auteur de vingt ans :

« Je crois que nous en sommes arrivés maintenant au point sur la question de savoir si la période actuelle du droit d'auteur protège de façon adéquate les intérêts des auteurs et sur sa question relative de savoir si la durée de protection fournit une motivation suffisante pour la création de nouvelles œuvres d'auteurs. »

Ce projet de loi étendait le droit d'auteur sur les œuvres déjà publiées écrites depuis les années 1920. Cette modification était un cadeau aux éditeurs sans bénéfice possible pour le public, puisqu'il y a aucun moyen d'augmenter rétroactivement le nombre de livres publiés jusqu'à lors. Cependant, cela vaut au public une liberté qui est significative aujourd'hui — la liberté de redistribuer des livres de cette période.

Le projet de loi étendait aussi les droits d'auteur d'œuvres à venir. Pour les œuvres destinées à la location, le droit d'auteur durerait quatre-vingt-quinze ans au lieu des soixante-quinze ans actuels. Théoriquement, cela aurait augmenté l'incitation à écrire de nouvelles œuvres; mais tout éditeur qui aurait réclamé le besoin de cette surprime, devrait justifier la revendication avec des projections de bilans pour l'année 2075.

Inutile de dire que le Congrès n'a pas remis en question les arguments des éditeurs : une loi étendant le droit d'auteur fut promulguée en 1998. Elle fut appelée Sonny Bono Copyright Term Extension Act, d'après le nom de l'un de ses soutiens qui mourut un peu plus tôt cette année-là. Sa veuve, qui bénéficiait du reste de sa période de droit d'auteur, fit cette déclaration :

« En fait, Sonny voulait que le droit d'auteur dure à jamais. J'ai été informée par la direction qu'un tel changement violerait la Constitution. Je vous invite tous à travailler avec moi pour renforcer les lois sur le droit d'auteur par tous les moyens à notre disposition. Comme vous le savez, il y a aussi la proposition de Jack Valenti pour que cela dure à jamais moins un jour. Peut-être que la commission examinera cette proposition au prochain Congrès. »

La Cour suprême a consenti à auditionner une affaire qui cherchait à renverser la loi sur le fondement que l'extension rétroactive échouait à servir le but de la Constitution sur la promotion du progrès.

Une autre loi, passée en 1996, a rendu criminel de faire plusieurs copies de n'importe quelle œuvre publiée, même si c'est pour faire plaisir à vos amis. Auparavant, ce n'était pas du tout un crime aux États-Unis.

Une loi encore pire, la Digital Millennium Copyright Act (DMCA) (loi sur le droit d'auteur du millénaire numérique), a été conçue pour ranimer la protection contre la copie (ce que détestent les utilisateurs d'ordinateurs) en rendant criminel de casser la protection contre la copie, ou même de publier toute information sur le moyen de casser cette protection. Cette loi devrait s'appeler « Domination by Media Corporations Act » (la loi sur la domination par les compagnies de médias) parce qu'elle offre effectivement aux éditeurs la chance d'écrire leur propre loi sur le droit d'auteur. Elle dit qu'ils peuvent imposer toute restriction quelle que soit sur l'utilisation de l'œuvre, et ces restrictions prennent force de loi pour peu que l'œuvre contienne un moyen d'encryptage ou de gestionnaire de licence pour les faire respecter.

Un des arguments proposés pour ce projet de loi était que cela mettrait en œuvre un récent traité pour augmenter les pouvoirs du droit d'auteur. Le traité fut promulgué par l'Organisation mondiale de la propriété intellectuelle (OMPI), un organisme dominé par les intérêts des détenteurs de droits d'auteurs et de brevets, avec l'aide de la pression exercée par l'administration Clinton; puisque le traité n'augmente que le pouvoir du droit d'auteur, qu'il serve l'intérêt du public dans n'importe quel pays semble douteux. En tout cas, le projet de loi allait bien au-delà de ce que requérait le traité.

Les bibliothèques furent un des piliers de l'opposition à ce projet de loi, spécialement pour les aspects qui bloquent les formes de copie qui sont considérées comme « usage raisonnable ». Comment les éditeurs ont-ils répondu ? L'ancien député Pat Schroeder, maintenant lobbyiste pour l'Association des éditeurs américains, dit que les éditeurs « ne pourraient pas vivre avec ce que [les bibliothèques] demandent ». Puisque les bibliothèques demandaient seulement de préserver une partie du status quo, on pourrait répondre en se demandant comment les éditeurs ont survécu jusqu'à ce jour.

Le membre du Congrès Barney Frank, lors d'une rencontre avec moi et d'autres opposés à ce projet de loi, montra combien la vision du droit d'auteur de la Constitution américaine avait été dédaignée. Il dit que ces nouveaux pouvoirs, appuyés par des peines criminelles, étaient nécessaires et urgents car « l'industrie du film était inquiète », tout autant que « l'industrie de la musique » et d'autres « industries ». Je lui demandais, « Mais est-ce dans l'intérêt du public ? » Sa réponse fut : « Pourquoi parlez-vous de l'intérêt du public ? Ces gens créatifs n'ont pas à abandonner leurs droits dans l'intérêt du public ! ». « L'industrie » a été assimilée aux « gens créatifs » qu'elle emploie, le droit d'auteur a été traité comme lui revenant de droit, et la Constitution a

été mise sens dessus dessous.

La DMCA a été promulguée en 1998. Elle dit que l'usage raisonnable demeure pour la forme légitime, mais autorise les éditeurs à interdire tout logiciel ou matériel avec lesquels vous pourriez le faire. En fait, l'usage raisonnable est interdit.

En se fondant sur cette loi, l'industrie du film a imposé une censure sur le logiciel libre pour la lecture des DVD, et même sur l'information de la façon dont on les lit. En avril 2001, le professeur Edward Felten de l'université de Princeton fut intimidé par des poursuites judiciaires par la Recording Industry Association of America (RIAA) (Association de l'industrie de l'enregistrement d'Amérique) pour retirer un document scientifique exposant ce qu'il avait appris au sujet d'une proposition de système d'encryptage pour restreindre l'accès à de la musique enregistrée.

Nous commençons aussi à voir des livres électroniques qui retirent aux lecteurs beaucoup de leurs libertés traditionnelles — par exemple, la liberté de prêter un livre à un ami, de le vendre à un bouquiniste, de l'emprunter à une bibliothèque, de l'acheter sans donner son nom à une banque de données d'une société, et même la liberté de le lire deux fois. Les livres cryptés limitent généralement toutes ces activités — vous ne pouvez les lire qu'avec un logiciel de décryptage spécial conçu pour vous enfermer.

Je n'achèterai jamais un de ces livres cryptés et verrouillés, et j'espère que vous les rejetterez aussi. Si un livre électronique ne vous donne pas les mêmes libertés qu'un livre papier traditionnel, ne l'acceptez pas !

Quiconque réalise indépendamment un logiciel qui peut lire des livres électroniques cryptés risque des poursuites. Un programmeur russe, Dmitry Sklyarov, a été arrêté en 2001 alors qu'il visitait les États-Unis pour parler à une conférence, car il avait écrit un tel programme en Russie, où il était légal de le faire. Maintenant la Russie prépare également une loi pour l'interdire, et l'Union européenne en a adopté une récemment.

Le marché de masse des livres électroniques a été jusqu'à présent un échec commercial, pas parce que les lecteurs ont choisi de défendre leurs libertés; ils ne sont pas attrayants pour d'autres raisons, comme le fait que les écrans d'ordinateurs ne sont pas des surfaces agréables pour lire. Nous ne pouvons pas compter sur cet heureux accident pour nous protéger sur le long terme; la prochaine tentative pour promouvoir les livres électroniques utilisera le « papier électronique » — des objets ressemblant à des livres dans lesquels des livres électroniques cryptés et restrictifs peuvent être téléchargés. Si cette surface à l'apparence du papier se révèle plus attrayante que les écrans d'affichage, nous aurons à défendre notre liberté afin de la conserver. Pendant ce temps, les livres électroniques font des incursions dans certains créneaux : l'université de New York et d'autres écoles dentaires demandent à leurs étudiants d'acheter leur manuels sous la forme de livres électroniques restrictifs.

Les sociétés de médias ne sont pas satisfaites cependant. En 2001, le sénateur Hollings (investi par Disney) a proposé un projet de loi appelé « Security Systems Standards and Certification Act » (SSSCA)[\[1\]](#) (loi sur les standards des systèmes de sécurité et certification), qui voudrait que tous les ordinateurs (et les autres appareils d'enregistrement numérique et de lecture) aient des systèmes de restrictions de copie rendus obligatoires par le gouvernement. Ceci est leur but ultime, mais la première étape de leur plan est d'interdire tout équipement qui puisse mettre au point la télévision haute définition (HDTV) numérique à moins que celui-ci soit conçu pour qu'il soit impossible au public de le « fausser » (c'est-à-dire, de le modifier pour son propre usage). Puisque le logiciel libre est du logiciel que les utilisateurs peuvent modifier, nous sommes là en face, pour la première fois, d'un projet de loi qui interdit explicitement le logiciel libre pour certains travaux. L'interdiction d'autres travaux suivra sûrement. Si la FCC adopte cette réglementation, des logiciels libres existants tels que GNU Radio seraient censurés.

Bloquer ces projets de lois et réglementations requiert une action politique.[\[2\]](#)

Trouver le bon marchandage

Quelle est la méthode appropriée pour décider de la politique du droit d'auteur ? Si le droit d'auteur est un marchandage fait pour le compte du public, il devrait servir l'intérêt du public avant tout. Le devoir du gouvernement en vendant la liberté du public est de vendre seulement ce qui doit l'être, et de le vendre le plus chèrement possible. À tout le moins, nous devrions réduire autant que possible l'extension de droit d'auteur

tout en maintenant un niveau comparable de publication.

Puisque nous ne pouvons pas trouver ce prix minimum à la liberté par des appels d'offres, comme nous le faisons pour des projets de construction, comment pouvons-nous le trouver ?

Une des méthodes possibles est de réduire les privilèges du droit d'auteur par paliers, et d'observer les résultats. En voyant si et dans quelles mesures les diminutions de publication surviennent, nous apprendrons quel pouvoir de droit d'auteur est réellement nécessaire pour réaliser les objectifs du public. Nous devons juger cela par une observation factuelle, pas par ce que nous prédisent les éditeurs, car ils ont toutes les motivations pour faire des prédictions exagérées de ruine si leurs pouvoirs sont réduits d'une quelconque façon.

La politique du droit d'auteur inclut plusieurs aspects indépendants, qui peuvent être ajustés séparément. Après avoir trouvé le minimum nécessaire pour un aspect politique, il est encore possible de réduire d'autres aspects du droit d'auteur tout en maintenant le niveau de publication désiré.

Un aspect important du droit d'auteur est sa durée, laquelle est maintenant typiquement de l'ordre du siècle. Réduire le monopole de la copie à dix ans, en partant de la date à laquelle l'œuvre est publiée, serait une bonne première étape. Un autre aspect du droit d'auteur, qui couvre la fabrication des œuvres dérivées, pourrait continuer sur une période plus longue.

Pourquoi partir de la date de publication ? Parce que le droit d'auteur sur les œuvres non publiées ne limitent pas directement la liberté des lecteurs; que nous soyons libres de copier une œuvre est discutable quand nous n'avons pas de copies. Aussi, donner une plus longue période pour faire publier une œuvre n'est pas préjudiciable. Les auteurs (qui possèdent généralement le droit d'auteur avant publication) choisissent rarement de retarder la publication seulement pour repousser le terme de leur droit d'auteur.

Pourquoi dix ans ? Parce que c'est une proposition sûre; nous pouvons être assurés pour des raisons pratiques que cette réduction aurait un impact minime sur la viabilité globale de la publication aujourd'hui. Dans la plupart des médias et des genres, les œuvres qui ont du succès sont très rentables en seulement quelques années, et même les œuvres à succès ne sont habituellement plus publiées bien avant dix ans. Même pour les œuvres de référence, dont l'utilité peut aller jusqu'à plusieurs décennies, dix ans de droit d'auteur suffiraient : des éditions mises à jour sont publiées régulièrement, et beaucoup de lecteurs achèteront la publication sous droit d'auteur plutôt qu'une copie vieille de dix ans du domaine public.

Dix ans peuvent être encore plus longs que nécessaire; une fois les choses établies, nous pourrions essayer une autre réduction pour ajuster le système. À une table ronde sur le droit d'auteur lors d'une convention littéraire, où je proposais le bail de dix ans, un célèbre auteur de fiction assis à côté de moi objecta avec véhémence, disant que toute proposition au-delà de cinq ans était intolérable.

Mais nous n'avons pas à appliquer la même durée à toutes les sortes d'œuvres. Maintenir la plus grande uniformité dans la politique du droit d'auteur n'est pas crucial pour l'intérêt public, et la loi sur le droit d'auteur a déjà beaucoup d'exceptions pour des usages et des médias spécifiques. Il serait absurde de payer pour chaque projet d'autoroute le taux nécessaire pour les projets les plus difficiles dans les régions les plus chères du pays; il serait également absurde de « payer » le plus grand prix en terme de liberté pour toutes les sortes d'art, que nous trouvons nécessaire pour un art en particulier.

Aussi, peut-être que les romans, les dictionnaires, les logiciels, les chansons, les symphonies et les films devraient avoir des durées de droit d'auteur différentes, de sorte que nous puissions réduire la durée pour chaque type d'œuvre à ce qui est nécessaire pour tant d'œuvres à publier. Peut-être que les films de plus d'une heure pourraient avoir un droit d'auteur de vingt ans, à cause des dépenses pour les produire. Dans mon propre domaine, la programmation de logiciels, trois ans devraient suffire, car les cycles de production, sont même plus courts que cela.

Un autre aspect du droit d'auteur est l'extension de l'usage raisonnable : la reproduction de tout ou partie d'une œuvre publiée qui est légalement permise même si elle est couverte par le droit d'auteur. La première étape naturelle en réduisant cet aspect du pouvoir du droit d'auteur, est de permettre la copie et la distribution privée et occasionnelle, à but non commercial, en petite quantité pour les particuliers. Ceci éliminerait l'intrusion de la police de droit d'auteur dans la vie privée des gens, mais aurait probablement peu d'effet sur les ventes d'œuvres publiées. (Il peut être nécessaire de prendre d'autres dispositions juridiques

pour s'assurer que les licences sous plastique ne puissent pas être utilisées pour se substituer au droit d'auteur en restreignant de telles copies). L'expérience de Napster montre que nous devrions aussi permettre la reproduction et la distribution non commerciale au public quand tant de gens veulent copier et partager, et trouvent cela utile, seules des mesures draconiennes les arrêteront, et le public mérite ce qu'il veut.

Pour les romans, et en général pour les œuvres destinées au divertissement, la redistribution non commerciale peut être une liberté suffisante pour les lecteurs. Les logiciels étant utilisés à des fins fonctionnelles (pour effectuer des tâches), appellent à des libertés supplémentaires, y compris celle de publier une version améliorée. Voir « Définition du logiciel libre », dans ce livre, pour une explication des libertés que les utilisateurs de logiciels devraient avoir. Mais ce peut être un compromis acceptable que d'avoir ces libertés universellement disponibles après un délai de deux à trois ans à compter de la date de publication du logiciel.

Des changements comme ceux-ci pourraient amener le droit d'auteur à être en phase avec les souhaits du public pour utiliser la technologie numérique pour copier. Les éditeurs trouveront ces propositions sans aucun doute « déséquilibrées »; ils pourront menacer de retirer leurs billes du jeu et de rentrer chez eux, mais ils ne le feront sûrement pas, car le jeu restera profitable et qu'il n'y aura pas d'autres choix.

Considérant les réductions de pouvoir du droit d'auteur, nous devons nous assurer que les sociétés de médias ne les remplaceront pas tout simplement par un contrat de licence utilisateur. Il serait nécessaire d'interdire l'utilisation de contrats qui appliquent des restrictions sur la copie qui vont au-delà de celles comprises dans le droit d'auteur. De telles limitations sur ce que peuvent demander des contrats non négociés de marché de masse sont prévues dans une partie standard du système légal américain.

Remarque personnelle

Je suis un concepteur de logiciels, pas un universitaire. Je me suis intéressé aux problèmes de droits d'auteur car il n'y a pas moyen de les éviter dans le monde des réseaux d'ordinateurs^[3]. En tant qu'utilisateur d'ordinateurs et de réseaux depuis trente ans, je peux évaluer les libertés que nous avons perdues, et celles que nous pourrions perdre bientôt. En tant qu'auteur, je peux rejeter la mystique romantique de l'auteur en tant que créateur semi-divin, souvent citée par les éditeurs pour justifier des pouvoirs de droits d'auteurs accrus pour les auteurs, lesquels signeront leur renonciation à ces droits au bénéfice des éditeurs.

En grande partie, cet article repose sur des faits et des arguments que vous pouvez vérifier, et des propositions sur lesquelles vous pouvez vous faire votre propre opinion. Mais je vous demande d'accepter une seule chose de mon discours : que les auteurs comme moi ne méritent pas de pouvoirs spéciaux à votre détriment. Si vous voulez me récompenser plus tard, pour des logiciels ou des livres que j'aurais écrits, j'accepterai volontiers un chèque — mais s'il vous plaît, n'abandonnez pas votre liberté en mon nom.

Notes

1. Depuis renommé pour l'imprononçable CBDTPA, pour lequel un bon moyen mnémotechnique est, « Consume, But Don't Try Programming Anything », (Consomme, mais n'essaie pas de programmer quoi que ce soit), mais il signifie vraiment « Consumer Broadband and Digital Television Promotion Act » (loi sur la promotion pour la télévision numérique à large bande du consommateur).
2. Si vous voulez aider, je vous recommande les sites web publicknowledge.org et www.eff.org.
3. Internet étant le plus grand des réseaux d'ordinateurs du monde.

Copyright © 2002, 2003, 2007 Free Software Foundation, Inc.,

51 Franklin St, Fifth Floor, Boston, MA 02110, USA

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Cédric Corazza.

Révision : trad-gnu@april.org

Dernière mise à jour : 2009/01/09 09:29:11

Traductions de cette page

[English](#) [en] [Español](#) [es] [Français](#) [fr] [Italiano](#) [it] [Nederlands](#) [nl] [Polski](#) [pl]

Les scientifiques doivent « mettre le droit d'auteur de côté »

par *Richard Stallman*

Cet article est paru dans Nature Webdebates en 2001

Il devrait être évident pour tout un chacun que la littérature scientifique est faite pour diffuser les connaissances scientifiques, et que les revues scientifiques existent pour faciliter ce travail. Par conséquent les règles d'utilisation de la littérature scientifique devraient être de nature à aider à atteindre cet objectif.

Les règles actuelles, appelées droit d'auteur, ont été établies à l'époque de la presse écrite, une méthode de publication de masse inévitablement centralisée. Dans un contexte de presse écrite, le droit d'auteur sur les articles de revues n'était une contrainte que pour les éditeurs — les obligeant à obtenir une autorisation avant de publier un article — et les plagiaires potentiels. Elle aidait les revues à fonctionner et à faire circuler les connaissances, sans gêner le travail utile des scientifiques ou des étudiants, qu'ils soient lecteurs ou auteurs d'articles. Ces règles conviennent bien à ce système.

Mais la technique actuelle de publication scientifique est le Web. Quelles sont les règles susceptibles d'assurer une diffusion maximale des articles et connaissances scientifiques sur le Web ? Il faudrait que les articles soient diffusés sous des formats non-propriétaires, en accès libre pour tous. Et chacun devrait avoir le droit de placer les articles sur un site « miroir », c'est-à-dire de les republier intégralement en indiquant correctement la source.

Ces règles devraient s'appliquer aussi bien aux anciens articles qu'aux nouveaux, lorsqu'ils sont diffusés sous forme électronique. Mais il n'y a pas de nécessité vitale à modifier le système actuel de droits d'auteur en ce qui concerne la publication des revues sur papier parce qu'il n'y a pas de problème dans ce domaine.

Malheureusement, il semble que tout le monde ne soit pas d'accord avec les évidences exprimées au début de cet article. De nombreux éditeurs de revues semblent croire que l'objectif de la littérature scientifique est de leur permettre de publier des revues afin de recueillir des abonnements de scientifiques et d'étudiants. Partager un tel point de vue c'est « confondre la fin et les moyens ».

Leur approche consiste à restreindre l'accès à la simple consultation de la littérature scientifique à ceux qui veulent et peuvent payer pour cela. Ils utilisent le droit d'auteur, qui est toujours en vigueur malgré son inadaptation aux réseaux informatiques, comme excuse pour dissuader les scientifiques de choisir de nouvelles règles.

Dans l'intérêt de la coopération scientifique et de l'avenir de l'humanité, nous devons refuser cette approche à la base — non seulement les systèmes d'obstruction qui ont été mis en place, mais aussi les mauvaises priorités qui les ont inspirés.

Les éditeurs de revues affirment parfois que la mise à disposition en ligne nécessite des serveurs puissants, et qu'ils sont obligés de faire payer l'accès pour financer ces serveurs. Ce « problème » est une conséquence de sa propre « solution ». Que l'on donne à chacun la liberté de publier sur un site miroir, et les bibliothèques du monde entier vont créer des sites miroirs pour faire face à la demande. Cette solution décentralisée réduira la bande passante du réseau et assurera un accès plus rapide, tout en protégeant les données universitaires contre une perte accidentelle.

Les éditeurs disent aussi qu'il faut un accès payant pour rémunérer les rédacteurs. Admettons qu'il faille payer les rédacteurs; c'est l'arbre qui cache la forêt. Les frais de rédaction pour un article moyen varient entre 1% et 3% du coût nécessaire à la recherche qui l'a suscité. Un si petit pourcentage ne justifie pas que l'on fasse obstruction à l'utilisation des résultats.

Au lieu de cela, les frais de rédaction pourraient être couverts, par exemple, par des frais à la page pour les auteurs, qui pourraient les faire financer par les commanditaires de la recherche. Ces commanditaires ne devraient pas s'y opposer dans la mesure où ils financent déjà lourdement cette publication par l'intermédiaire des frais que constitue l'abonnement de la bibliothèque universitaire à la revue. En changeant le modèle économique (transfert des frais de publication vers les commanditaires de la recherche), on peut éliminer le besoin apparent de restreindre la consultation. L'auteur occasionnel qui n'est

pas affilié à une institution ou une entreprise, et qui n'a pas de commanditaire pour sa recherche, pourrait être dispensé de frais de page, qui seraient répercutés sur les auteurs soutenus par des institutions.

Une autre justification pour faire payer l'accès aux publications en ligne c'est de dire qu'il faut financer la conversion des archives imprimées au format électronique. Ce travail doit être fait, mais il nous faut trouver des alternatives qui ne remettent pas en cause le libre accès au résultat. Le travail lui-même ne sera pas plus difficile, ni plus coûteux. Il est contre-productif de numériser des archives et de gâcher le résultat de ce travail en en restreignant l'accès.

La Constitution des États-Unis dit que le droit d'auteur existe « pour promouvoir les progrès de la science ». Quand les droits d'auteur freinent les progrès de la science, la science doit mettre le droit d'auteur de côté.

Copyright © 2001 Richard M. Stallman

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Bruno Menan.

Révision : trad-gnu@april.org

Dernière mise à jour : 2009/02/17 09:28:04

Traductions de cette page

[български](#) [bg] [English](#) [en] [español](#) [es] [français](#) [fr] [Nederlands](#) [nl] [polski](#) [pl] [简体中文](#) [zh-cn]

Qu'est-ce que le copyleft ?

Le *Copyleft* est une façon de rendre un programme ou tout autre œuvre libre, et qui requiert que toutes les versions modifiées et étendues du programme soient libres également.

La manière la plus simple de faire d'un programme un logiciel libre est de le distribuer dans le [domaine public](#), sans copyright. Cela autorise les gens à partager le programme et leurs améliorations si le cœur leur en dit. Mais cela autorise aussi des personnes indélicates à faire du programme un [logiciel propriétaire](#). Ils peuvent très bien y effectuer des changements, juste quelques-uns ou plusieurs, et distribuer le résultat comme un logiciel propriétaire. Ceux qui recevront le programme dans sa forme modifiée n'auront pas la liberté que l'auteur original leur aura donné; l'intermédiaire l'aura fait disparaître.

Dans le [projet GNU](#), notre but est de donner à *tous* les utilisateurs la liberté de redistribuer et de modifier les logiciels GNU. Si des intermédiaires pouvaient enlever cette liberté, nous aurions beaucoup d'utilisateurs, mais ils n'auraient aucune liberté. Alors, au lieu de mettre les logiciels GNU dans le domaine public, nous les mettons sous «copyleft» ou «gauche d'auteur». Le copyleft indique que quiconque les redistribue, avec ou sans modifications, doit aussi transmettre la liberté de les copier et de les modifier. Le copyleft garantit cette liberté pour tous les utilisateurs.

Le copyleft fournit aussi un [encouragement](#) aux autres programmeurs qui veulent ajouter des logiciels libres. Des programmes importants comme le compilateur C++ de GNU n'existent que grâce à lui.

Le copyleft aide aussi les programmeurs qui veulent contribuer [à des améliorations](#) sur des [logiciels libres](#) à obtenir la permission de le faire. Ces programmeurs travaillent souvent pour des entreprises ou des universités qui feraient n'importe quoi pour plus d'argent. Un programmeur pourrait vouloir faire profiter la communauté de ses modifications, mais son employeur pourrait vouloir transformer le travail en un produit propriétaire.

Quand nous expliquons à l'employeur qu'il est illégal de distribuer la version améliorée autrement que comme logiciel libre, celui-ci décide souvent de le distribuer librement plutôt que de le laisser tomber.

Pour mettre un logiciel sous copyleft, nous déclarons d'abord qu'il est sous copyright, ensuite nous ajoutons les conditions de distribution, qui sont un outil légal donnant à chacun le droit d'utiliser, de modifier, et de redistribuer le code du programme, ou *tous les programmes qui en sont dérivés*, mais seulement si les conditions de distribution demeurent inchangées. Ainsi, le code et ses libertés sont légalement indissociables.

Les développeurs de logiciels propriétaires utilisent le copyright pour restreindre la liberté des utilisateurs; nous utilisons le copyleft pour la garantir. C'est pourquoi nous avons inversé le nom, en changeant «copyright» en «copyleft».

Le copyleft est un moyen d'utilisation du copyright du programme. Cela ne signifie pas d'abandonner le copyright ; en fait, faire cela rendrait le copyleft impossible. Le mot « left » (NdT : en anglais, left signifie à la fois laissé, abandonné ou gauche) dans « copyleft » n'est pas une référence au verbe « to leave » (NdT : laisser, abandonner) c'est seulement une référence à la direction inverse de « right » (NdT : droite).

Le copyleft est un concept général, et vous ne pouvez pas utiliser un concept général directement ; vous pouvez seulement utiliser une mise en œuvre spécifique du concept. Dans le projet GNU, les conditions de distribution spécifiques que nous utilisons sont contenues dans la [GNU General Public License](#) ([disponible au format HTML](#), [texte](#), et [Texinfo](#)). La GNU General Public License est appelée la GNU GPL. Il y a également une page [Foire aux questions](#) à propos de la GPL GNU. Vous pouvez lire également [Pourquoi la FSF obtient l'attribution des droits d'auteurs des contributeurs](#).

Une forme alternative de copyleft, la [GNU Lesser General Public License \(LGPL\)](#) ([disponible au format HTML](#), [texte](#), et [Texinfo](#)), s'applique à quelques (mais pas à toutes) bibliothèques GNU. Cette licence était initialement appelée la Library GPL (GPL pour les bibliothèques), mais nous avons changé le nom car l'ancien nom encourageait l'utilisation de cette licence plus souvent qu'elle aurait dû être utilisée. Pour une explication sur les motivations qui nous ont convaincu que ce changement était nécessaire, lire l'article [pourquoi vous ne devriez pas utiliser la LGPL pour votre prochaine bibliothèque](#).

La [GNU Free Documentation License \(FDL\)](#) ([disponible au format HTML](#), [texte](#) et [Texinfo](#)) est une

forme de copyleft conçue pour être utilisée pour un manuel, un livre ou un autre document de manière à assurer à chacun la liberté effective de le copier et de le redistribuer, avec ou sans modifications, de façon commerciale ou non.

La licence appropriée est incluse dans beaucoup de manuels et dans chaque distribution de code source GNU.

Toutes ces licences sont conçues de façon à pouvoir être appliquées à votre programme si vous en détenez le copyright. Vous n'aurez pas à modifier la licence pour le faire, mais seulement à ajouter une copie de la licence à votre programme et des références appropriées dans les fichiers sources qui se réfèrent correctement à la licence.

L'utilisation des mêmes conditions de distribution pour plusieurs programmes différents facilite la copie de code entre ces programmes. Avec les mêmes conditions de distribution, il n'y a plus de souci d'incompatibilité. La LGPL contient une clause qui vous autorise à modifier les conditions de distribution de la GPL ordinaire, ainsi vous pouvez copier du code dans un autre programme couvert par la GPL.

Si vous désirez mettre votre programme sous copyleft avec la GNU GPL ou la GNU LGPL, veuillez lire la [page d'instructions de la GPL](#) comme conseil. Veuillez noter que vous devez reproduire le texte intégral de nos licences, si vous en utilisez une. C'est un tout, et les copies partielles ne sont pas autorisées.

Si vous désirez mettre votre manuel sous copyleft avec la GNU FDL, veuillez lire les instructions à la [fin \(21 ko\)](#) du texte de la FDFL, ainsi que la [page d'instructions de la GFDL](#). De même, les copies partielles ne sont pas autorisées.

Copyright © 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008 Free Software Foundation, Inc.,

51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Benjamin Drieu.

Révision : trad-gnu@april.org

Dernière mise à jour : 2009/03/01 09:29:22

Traductions de cette page

[العربية](#) [ar] [Bosanski](#) [bs] [català](#) [ca] [Česky](#) [cs] [Deutsch](#) [de] [English](#) [en] [español](#) [es] [فارسی](#) [fa] [français](#) [fr]

[Bahasa Indonesia](#) [id] [italiano](#) [it] [日本語](#) [ja] [polski](#) [pl] [português do Brasil](#) [pt-br] [русский](#) [ru]

[српски](#) [sr] [தமிழ்](#) [ta]

Copyleft : Idéalisme pragmatique

Par *Richard Stallman*

Toute décision prise par une personne découle de ses valeurs et buts. Les gens peuvent avoir beaucoup de buts et valeurs différents ; la gloire, le profit, l'amour, la survivance, l'amusement, ou la liberté, ne sont qu'une partie des buts qu'une personne normale peut avoir. Quand ce but est d'aider les autres aussi bien que soi même, cela s'appelle de l'idéalisme.

C'est un but idéaliste qui motive mon travail pour le logiciel libre : propager la liberté et la coopération. Je veux [encourager la diffusion des logiciels libres](#) et le remplacement des logiciels propriétaires qui empêchent la coopération, et rendre ainsi notre société meilleure.

C'est pour cette raison fondamentale que la GNU General Public License est écrite de cette manière - comme un copyleft. Tout code ajouté à un programme couvert par la GPL doit être un programme libre, même s'il est placé dans un fichier séparé. Je mets mon code à disposition pour une utilisation dans des logiciels libres, et pas pour un usage avec des logiciels propriétaires, afin d'encourager ceux qui programment des logiciels à les rendre libres également. Je suppose que, comme les développeurs de logiciels propriétaires utilisent le copyright pour nous empêcher de partager, nous, coopérateurs, pouvons utiliser le copyright pour donner aux autres coopérateurs un avantage bien à eux : ils peuvent utiliser notre code.

Ceux qui utilisent la GNU GPL n'ont pas tous ce but. Il y a plusieurs années, on a demandé à un de mes amis de rééditer un programme copylefté sous des termes non copyleftés, et il a plus ou moins répondu ainsi :

« Parfois je travaille sur des logiciels libres et parfois sur des logiciels propriétaires ; mais quand je travaille sur des logiciels propriétaires, j'exige d'être *payé*. »

Il voulait bien partager son travail avec une communauté qui partageait les logiciels mais ne voyait aucune raison d'aider un entreprise commerciale dont les produits seraient interdits à notre communauté. Son but était différent du mien mais il a décidé que la GNU GPL le satisfaisait également.

Si vous voulez accomplir quelque chose dans le monde, l'idéalisme ne suffit pas ; le choix d'une méthode qui mène à l'accomplissement de ce but est nécessaire. En d'autres termes, vous devez être « pragmatique ». La GPL est-elle pragmatique ? Regardons ses accomplissements.

Considérons GNU C++. Pourquoi existe-t-il un compilateur C++ libre ? Uniquement parce que la GNU GPL indiquait qu'il devait être libre. MCC, un consortium industriel, a développé GNU C++ à partir du compilateur GNU C. En temps normal, MCC rend sa production aussi propriétaire que possible. Mais ils ont fait une interface C++ libre parce que c'était la seule possibilité de l'éditer que leur laissait la GNU GPL. L'interface C++ comportait beaucoup de nouveaux fichiers, mais comme ils étaient prévus pour être liés à GCC, la GPL s'appliquait à eux. Le bénéfice pour notre communauté est évident.

Considérons GNU Objective C. Au début, NeXT voulait rendre cette interface propriétaire ; ils avaient l'intention de l'éditer sous la forme de fichiers .o et de laisser aux utilisateurs le soin de les attacher au reste de GCC, pensant pouvoir ainsi contourner les conditions de la GPL. Mais nos juristes ont dit que cela n'esquivait pas ces conditions et était interdit. Et ainsi l'interface Objective C fut un logiciel libre.

Ces exemples datent de plusieurs années, mais la GNU GPL continue à nous apporter toujours plus de logiciels libres.

Beaucoup de bibliothèques GNU sont couvertes par la GNU Lesser Library General Public License, mais pas toutes. Readline, qui met en œuvre l'édition de ligne de commande, est une des bibliothèques GNU couvertes par la GNU GPL classique. Un jour, j'ai découvert un programme propriétaire conçu pour utiliser Readline, et j'ai dit au développeur que cela était interdit. Il aurait pu sortir du programme l'édition de ligne de commande, mais ce qu'il fit en fait, fut de le rééditer sous la GPL. Maintenant, c'est un logiciel libre.

Les programmeurs qui conçoivent des améliorations pour GCC (ou Emacs, Bash, Linux ou tout programme couvert par la GPL) sont souvent employés par des entreprises ou des universités. Lorsque le programmeur veut remettre son amélioration à la communauté et voir son code dans la prochaine édition, il

est probable que son patron lui dise : « Attendez un peu ; votre code nous appartient ! Nous ne voulons pas le partager ; nous avons décidé de placer votre version améliorée dans un logiciel propriétaire ».

C'est à ce moment qu'intervient la GNU GPL. Le programmeur montre à son patron que ce logiciel propriétaire serait une violation de copyright, et ce dernier réalise qu'il ne lui reste que deux possibilités : publier le code en tant que logiciel libre ou pas du tout. Il laisse presque toujours le programmeur faire ce qu'il voulait initialement et le code est inclus dans la prochaine version.

La GNU GPL n'est pas M. Chic Type. Elle dit « non » à certaines choses qu'on aimerait parfois faire. Certains utilisateurs disent qu'il est dommage que la GPL « exclut » certains développeurs de logiciels propriétaires qui « auraient besoin d'être amenés à la communauté du logiciel libre ».

Mais nous ne les excluons pas de notre communauté ; ils ont choisi de ne pas y entrer. Décider de concevoir des logiciels propriétaires, c'est décider de rester en dehors de notre communauté. Appartenir à notre communauté signifie prendre part à notre coopération ; nous ne pouvons pas « les amener à notre communauté » s'ils ne le désirent pas.

Ce que nous *pouvons* faire, c'est les inciter à nous rejoindre. La GNU GPL est faite pour utiliser comme appâts les logiciels que nous possédons : « Si vous faites des logiciels libres, vous pourrez utiliser ces codes ». Bien sûr, cela ne les convaincra pas tous, mais nous en gagnerons un de temps en temps.

La réalisation de logiciels propriétaires ne participe pas à notre communauté, mais ses développeurs souhaitent souvent une aide de notre part. Les utilisateurs de logiciels libres peuvent offrir des encouragements pour l'égo des développeurs de logiciels libres — reconnaissance et gratitude — mais ils peuvent être tentés lorsqu'une entreprise leur dit : « Laissez-nous juste placer votre logiciel dans notre programme propriétaire et il sera utilisé par des milliers de gens ! » La tentation peut être forte, mais à long terme, nous nous sentons tous mieux si nous y avons résisté.

Il est plus difficile de reconnaître les pressions et tentations quand elles viennent indirectement, au travers d'organisations pour les logiciels libres qui ont adopté une politique satisfaisant aux logiciels propriétaires. Le Consortium X (et son successeur l'Open Group) en sont un exemple : fondé par des sociétés qui font des logiciels propriétaires, ils se sont efforcés de persuader les programmeurs de ne pas utiliser de copyleft pendant des décennies. Maintenant que l'Open Group a [fait de X11R6.4 un logiciel propriétaire](#), ceux d'entre nous qui ont résisté à cette pression sont heureux de l'avoir fait.

[En septembre 1998, plusieurs mois après que X11R6.4 ait été distribué sous une licence non-libre, l'Open Group a revu sa décision et l'a redistribué sous la même licence de logiciel libre non-copylefté que X11R6.3. Merci, l'Open Group ; mais ce revirement ultérieur n'infirme pas les conclusions que nous avons tirées du fait que l'ajout de restrictions était *possible*.]

Pour parler pragmatiquement, avoir une vision à plus long terme affermira votre volonté de résister à cette pression. Si vous concentrez votre attention sur la liberté et la communauté que vous pouvez bâtir en restant ferme, vous trouverez la force de le faire. « Restez debout pour quelque chose ou vous tomberez pour rien ».

Et si des personnes cyniques ridiculisent la liberté, la communauté... si des « réalistes forcenés » disent que le profit est le seul idéal... ignorez-les et utilisez le copyleft tout de même.

Copyright © 1998, 2003 Free Software Foundation, Inc.,
51 Franklin St, Fifth Floor, Boston, MA 02110, USA

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : ?

Révision : trad-gnu@april.org

Dernière mise à jour : 2009/01/22 09:29:10

Traductions de cette page

[Català](#) [ca] [Deutsch](#) [de] [English](#) [en] [Español](#) [es] [Français](#) [fr] [Magyar](#) [hu] [Bahasa Indonesia](#) [id]
[Italiano](#) [it] [日本語](#) [ja] [한국어](#) [ko] [മലയാളം](#) [ml] [Nederlands](#) [nl] [Polski](#) [pl] [português do Brasil](#) [pt-br]
[Русский](#) [ru]

Livre III: À propos des licences



Cette notice est valable pour l'ensemble des articles, relativement:

Copyright © Richard Stallman,

ou Copyright © Free Software Foundation, Inc.,

La reproduction exacte et la distribution intégrale de chaque article est permise sur n'importe quel support d'archivage, pourvu que sa notice soit préservée.

Pouvez-vous faire confiance à votre ordinateur ?

Par Richard Stallman

De qui votre ordinateur devrait-il recevoir ses ordres ? La plupart des personnes pensent que leurs ordinateurs devraient leur obéir, et ne pas obéir à quelqu'un d'autre. Avec un plan qu'elles appellent « Trusted Computing » (« L'informatique de confiance »), de grandes sociétés de médias (y compris du cinéma et de l'industrie du disque), ainsi que des sociétés d'informatique telles que Microsoft et Intel, projettent de faire en sorte que votre ordinateur leur obéisse au lieu de vous obéir. Les programmes propriétaires ont déjà inclus des dispositifs malveillants auparavant, mais ce plan rendrait cette pratique universelle.

Par définition, vous ne contrôlez pas ce que fait un logiciel propriétaire ; vous ne pouvez ni étudier son code source ni le modifier. Il n'est pas étonnant que les hommes d'affaires intelligents trouvent des moyens pour exercer un contrôle sur vos actions et ainsi en tirer avantage à vos dépens. Microsoft l'a déjà fait plusieurs fois : une version de Windows a été conçue pour renseigner Microsoft sur tous les logiciels installés sur votre disque dur ; une mise à jour de « sécurité » récente du lecteur de multimédias de Windows (Windows Media Player) a imposé l'accord des utilisateurs sur de nouvelles restrictions. Mais Microsoft n'est pas seul dans ce cas : le logiciel de partage de musique « KaZaa » est conçu de sorte que les associés de la compagnie KaZaa puisse louer l'utilisation de votre ordinateur à leurs clients. Ces dispositifs malveillants sont souvent secrets, mais même une fois que vous en avez connaissance, il est difficile de les enlever, puisque vous ne disposez pas du code source de l'application.

Dans le passé, il s'agissait d'incidents isolés. « L'informatique de confiance » les rendrait dominants. « L'informatique déloyale » (« Treacherous Computing ») est un nom plus approprié, parce que le projet est conçu pour s'assurer que votre ordinateur vous ésobera systématiquement. En fait, il est conçu pour que votre ordinateur s'arrête de fonctionner comme ordinateur polyvalent. Chaque opération pourra exiger une permission explicite.

L'idée technique fondamentale de « l'informatique déloyale » est que l'ordinateur inclut un dispositif numérique de chiffrement et de signature, et les clefs sont maintenues secrètes (la version Microsoft de ce système s'appelle « Palladium »). Les logiciels propriétaires utiliseront ce dispositif afin de contrôler le lancement de tel ou tel programme, à quels documents ou données vous pourrez accéder, et avec quels programmes vous pourrez lire ou modifier ces documents ou données. Ces logiciels téléchargeront régulièrement de nouvelles règles d'autorisation par Internet, et vous les imposeront. Si vous ne laissez pas votre ordinateur récupérer périodiquement ces nouvelles règles depuis Internet, certaines options cesseront de fonctionner.

Naturellement, Hollywood et l'industrie du disque projettent d'employer l'informatique déloyale pour le « DRM » (Digital Restriction Management - gestion de restrictions numériques), de sorte que des vidéos ou de la musique téléchargées ne puissent être jouées que sur un ordinateur donné. Le partage des fichiers sera totalement impossible, du moins en utilisant les fichiers que vous obtiendriez auprès de ces sociétés et que serez autorisés à lire. Vous, le public, devez avoir la liberté et la possibilité de partager ces informations (je m'attends à ce que quelqu'un trouve une manière de produire des versions non codées, de télécharger et de partager celles-ci, ainsi le DRM ne s'appliquera pas entièrement, mais ce n'est pas une excuse pour laisser ce système s'implanter).

Rendre impossible le partage des fichiers vidéos et musicaux est une mauvaise chose, mais ça pourrait être pire. Il existe des projets pour généraliser ce dispositif aux messages électroniques et aux documents - ayant pour résultat un email qui disparaîtrait au bout de deux semaines, ou des documents qui pourront seulement être lus sur les ordinateurs d'une société mais pas sur ceux d'une autre.

Imaginez que vous recevez un courrier électronique de votre patron vous indiquant de faire quelque chose que vous pensez risqué ; un mois plus tard, lorsque la situation s'envenime, vous ne pouvez plus utiliser ce message pour prouver que la décision n'était pas de vous. « Obtenir l'ordre par écrit » ne vous protège pas quand l'ordre est écrit avec une encre qui disparaît.

Imaginez que vous obtenez un email de votre patron, vous demandant d'agir illégalement ou d'effectuer une action moralement indigne, comme déchiqueter les résultats de l'audit de votre compagnie, ou

permettre à une menace dangereuse pour votre pays de se propager. Aujourd'hui vous pouvez envoyer ce message à un journaliste et lui présenter les faits. Avec « l'informatique déloyale », le journaliste ne pourra pas lire le document ; son ordinateur refusera de lui obéir. L'« informatique déloyale » devient un paradis pour la corruption.

Les logiciels de traitement de texte tels que Word de Microsoft pourraient employer « l'informatique déloyale » quand ils enregistrent vos documents, pour s'assurer qu'aucun autre traitement de texte concurrent ne puisse les lire. Aujourd'hui nous devons deviner les secrets du format Word par des expériences laborieuses afin de programmer des traitements de texte libres qui puissent lire les documents au format Word (les .doc). Si Word chiffre les documents en utilisant « l'informatique déloyale » quand il les enregistre, la communauté du Logiciel Libre n'aura pas la possibilité de développer un programme capable de les lire - et même si nous le pouvions, de tels programmes seraient interdits par la DMCA (Digital Millennium Copyright Act - Loi de copyright du millénaire Numérique).

Les programmes qui utilisent « l'informatique déloyale » téléchargeront régulièrement de nouvelles règles par Internet, et imposeront ces règles automatiquement à votre travail. Si Microsoft ou le gouvernement des États-Unis, n'aime pas ce que vous énoncez dans un document écrit, ils seraient en mesure d'ajouter de nouvelles instructions indiquant à tous les ordinateurs de refuser de lire ce document. Chaque ordinateur obéirait, sitôt qu'il aurait téléchargé les nouvelles instructions. Vos écrits seraient sujets à l'effacement rétroactif « façon 1984 » ; au final, vous ne pourriez même plus les relire.

Vous pourriez penser que vous serez capable de découvrir quelles mauvaises choses fait une application de « l'informatique déloyale », que vous saurez étudier de quelles façons elles sont néfastes, et décider de les accepter ou non. Il faudrait être myope et idiot pour accepter, et il se trouve que la bonne affaire que vous pensez faire ne tiendra pas toujours. À partir du moment où vous devenez dépendant de l'utilisation d'un programme, vous êtes accrochés et ils le savent ; ils peuvent alors se permettre de changer la donne. Quelques applications récupéreront automatiquement les mises à jour qui fonctionneront alors de façon différente - et elles ne vous permettront pas de choisir de mettre à jour ou non.

Aujourd'hui vous pouvez éviter de voir vos libertés restreintes par le logiciel propriétaire en ne l'utilisant pas. Si vous exploitez GNU/Linux ou un autre système d'exploitation libre, et si vous évitez d'y installer des applications propriétaires, alors vous êtes responsable et pouvez décider de ce que fait votre ordinateur. Si un Logiciel Libre contient un dispositif malveillant, des développeurs de la communauté l'enlèveront, et vous pourrez utiliser la version corrigée. Vous pouvez également utiliser des programmes et des applications libres sur les systèmes d'exploitation non-libres ; ceci ne vous procure pas une entière liberté, mais beaucoup d'utilisateurs le font.

L'« informatique déloyale » met l'existence des systèmes d'exploitation libres et des applications libres en danger, parce que vous ne pourrez pas du tout les utiliser. Quelques versions de « l'informatique déloyale » exigeraient que, pour se lancer, le système d'exploitation bénéficie d'une autorisation spécifique, délivrée par une société. Des systèmes d'exploitation libres ne pourront pas être installés. Quelques versions de l'« informatique déloyale » exigeraient que, pour s'exécuter, chaque programme bénéficie d'une autorisation délivrée spécifiquement par le programmeur du système d'exploitation. Vous ne pourriez pas utiliser d'applications libres sur un tel système. Si vous trouviez une façon pour le faire, et le disiez à quelqu'un, cela pourrait être considéré comme un crime.

Il y a déjà des propositions de lois aux États-Unis pour exiger de tous les ordinateurs qu'ils utilisent « l'informatique déloyale », et pour interdire de relier de vieux ordinateurs à Internet. La CBDTPA (nous l'appelons le « Consume But Do Not Try Programming Act » - Consommons mais n'essayons pas de programmer) est l'une d'elles. Mais même s'ils ne vous forcent pas à passer à « l'informatique de confiance » par des lois, les pressions pour l'accepter peuvent être énormes. Aujourd'hui les gens utilisent souvent le format de Word pour s'échanger des documents, bien que cela cause plusieurs problèmes (voir la page [Finissons-en avec les attachements de Word !](#)). Si seulement une machine de « l'informatique déloyale » pouvait lire les documents créés avec la dernière version de Word, beaucoup de personnes l'utiliserait, d'un point de vue individuel (Prends-le ou laisse-le - take it or leave it). Pour s'opposer à l'« informatique déloyale », nous devons nous regrouper et refuser la situation comme un choix collectif.

Pour de plus amples informations au sujet de l'« informatique déloyale », voir la page

<http://www.cl.cam.ac.uk/users/rja14/tcpa-faq.html>.

Bloquer « l'informatique déloyale » exigera d'un grand nombre de citoyens de s'organiser. Nous avons besoin de votre aide ! L'[Electronic Frontier Foundation](#) et la [Public Knowledge](#) font campagne contre « l'informatique déloyale », ainsi que le projet [Digital Speech](#) commandité par la FSF. Veuillez visiter ces sites et ainsi vous pourrez vous inscrire et appuyer leur travail.

Vous pouvez également aider en écrivant aux sièges sociaux d'Intel, IBM, HP/Compaq, ou à tout autre constructeur à qui vous avez acheté un ordinateur, expliquant que vous ne voulez pas subir de pression pour acheter les systèmes informatiques « de confiance » et que vous ne voulez pas qu'ils en produisent. Vous contribuerez ainsi à augmenter la pression des consommateurs sur les constructeurs. Si vous le faites de votre propre chef, envoyez s'il vous plaît les copies de vos lettres aux organismes ci-dessus.

Notes

1. Le projet de GNU distribue le GNU Privacy Guard ([44]GPG), un programme qui permet le chiffrement par clés publiques et signatures numériques, que vous pouvez utiliser pour envoyer des emails sécurisés et privés. Il est utile d'étudier comment GPG diffère de l'« informatique déloyale », et de voir ce qui rend l'un utile et l'autre si dangereux.

Quand quelqu'un emploie GPG pour vous envoyer un document chiffré, et que vous utilisez GPG pour le décoder, le résultat est un document non codé que vous pouvez lire, transférer, copier, et même re-chiffrer pour l'envoyer de manière sécurisée à quelqu'un d'autre. Une application de l'« informatique déloyale » vous laisserait lire les mots sur l'écran, mais ne vous permettrait pas de produire un document non codé que vous pourriez utiliser d'autres manières. GPG, un logiciel libre, aide à mettre en place des dispositifs de sécurité disponibles pour les utilisateurs ; ils l'utilisent. L'« informatique déloyale » est conçue pour imposer des restrictions aux utilisateurs ; dans ce cas c'est elle qui les « utilise ».

2. Les supporters de l'informatique déloyale concentrent leur discours sur son utilisation bénéfique. Ce qu'ils disent est souvent correct, mais sans importance.

Comme la plupart des appareils, les équipements de l'informatique déloyale peuvent être utilisés à des fins qui ne sont pas dommageables. Mais leurs utilisations peuvent être mises en œuvre par d'autres moyens, sans utiliser d'équipement de l'informatique déloyale. La principale différence de l'informatique déloyale pour les utilisateurs est sa fâcheuse conséquence : la manipulation de votre ordinateur pour qu'il travaille contre vous.

Ce qu'ils disent est vrai, et ce que je dis est vrai. Mettez-les ensemble et qu'obtenez-vous ? L'informatique déloyale est un plan pour retirer votre liberté tout en offrant des bénéfices mineurs pour nous distraire de ce que nous perdons.

3. Microsoft présente Palladium comme un dispositif de sécurité, et prétend qu'il protégera vos données contre les virus. Mais ce discours est évidemment faux. Une présentation réalisée par Microsoft Research (le département Recherche/Développement de Microsoft) en octobre 2002 a montré qu'une des caractéristiques de Palladium consiste à permettre aux systèmes d'exploitation et aux applications existantes de continuer de fonctionner ; donc, les virus continueront à faire toutes les choses qu'ils font aujourd'hui.

Quand Microsoft parle de « sécurité » à propos de Palladium, il ne s'agit pas de notre définition de la sécurité : protéger votre machine contre des choses que vous ne voulez pas. Elle signifie protéger vos données stockées sur votre machine contre l'accès par vous, d'une manière que d'autres ne veulent pas. Un diaporama dans la présentation montre plusieurs types d'informations secrètes que Palladium pourrait conserver, y compris des secrets de tiers et des « secrets d'utilisateurs » - mais « secrets d'utilisateurs » est mis entre guillemets, ce qui signifie que Palladium n'est pas véritablement conçu pour ce type d'informations.

Dans la présentation on trouve fréquemment d'autres termes que nous associons habituellement à la notion de sécurité, tels que « attaque », « code malveillant », « spoofing », ainsi que « confiance » (trusted). Aucun d'eux ne désigne ce que signifie normalement ces mots. Une « attaque » ne signifie pas que quelqu'un essaye de vous blesser. Il signifie que vous essayez de copier de la musique. « Code malveillant » signifie un code installé par vous pour faire ce que quelqu'un d'autre ne veut pas que votre machine fasse. « Spoofing » ne signifie pas quelqu'un qui vous dupe, il signifie que vous dupez Palladium. Et ainsi de suite.

4. Un rapport précédent, écrit par les partisans de Palladium a énoncé le principe de base suivant : que celui qui a développé ou a rassemblé l'information devrait avoir le contrôle total de la façon dont vous l'utilisez. Ceci représenterait un renversement révolutionnaire des idées passées de l'éthique et du système légal, et créerait un système de contrôle sans précédent. Les problèmes spécifiques de ces systèmes ne sont aucunement des accidents ; ils résultent du principe de base. C'est ce but que nous devons rejeter.

Copyright © 2002, 2007 Richard Stallman

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Fabien Illide
Révision : trad-gnu@april.org
Dernière mise à jour : 2009/03/06 21:26:07

Traductions de cette page

[العربية](#) [ar] [български](#) [bg] [Deutsch](#) [de] [English](#) [en] [español](#) [es] [فارسی](#) [fa] [français](#) [fr] [עברית](#) [he]
[italiano](#) [it] [한국어](#) [ko] [Nederlands](#) [nl] [polski](#) [pl] [Türkçe](#) [tr] [தமிழ்](#) [ta] [简体中文](#) [zh-cn]
[繁體中文](#) [zh-tw]

Pourquoi les logiciels devraient être libres

Par *Richard Stallman*

(Version du 24 avril 1992)

Introduction

L'existence de logiciels soulève forcément la question de la façon dont devraient être prises les décisions concernant leur usage. Par exemple, supposons qu'une personne ayant une copie d'un programme, rencontre une autre personne qui en voudrait une copie. Il est possible pour eux de copier le programme; qui devrait décider si c'est possible ? Les personnes elles-mêmes ? Ou une tierce personne, son « propriétaire » ?

Les développeurs de logiciels considèrent typiquement que le critère de réponse supposé est la maximisation de leurs profits. Le pouvoir politique des affaires a poussé le gouvernement à adopter à la fois ce critère et la réponse proposée par les développeurs : c'est-à-dire qu'un programme a un propriétaire, en général une société associée à son développement.

J'aimerais considérer la même question en utilisant un critère différent : la prospérité et la liberté du public en général.

Cette réponse ne peut pas être décidée par la loi actuelle — la loi devrait se conformer à la morale, pas l'inverse — ni par l'usage, bien qu'ils puissent suggérer des réponses possibles. La seule façon d'en juger, est d'observer qui bénéficie et qui est lésé en reconnaissant un propriétaire au logiciel, pourquoi et dans quelle mesure. En d'autres termes, nous devrions en évaluer le pour et le contre pour la société dans son ensemble, en prenant en compte aussi bien la liberté individuelle que la production de biens matériels.

Dans cet article, je décrirai les effets d'avoir des propriétaires, et je montrerai que les résultats sont préjudiciables. Ma conclusion est que les programmeurs ont le devoir d'encourager les autres à partager, distribuer, étudier et améliorer les logiciels que nous écrivons : autrement dit, d'écrire des [logiciels libres](#). (1)

Comment les propriétaires justifient leur pouvoir

Ceux-là, qui bénéficient du système actuel où les programmes sont propriétaires, proposent deux arguments en leur faveur : l'argument affectif et l'argument économique.

L'argument affectif ressemble à ceci : « j'ai mis ma sueur, mon cœur, mon âme dans ce programme. Il vient de *moi*, c'est *le mien* ! »

Cet argument ne nécessite pas de réfutation sérieuse. Les programmeurs peuvent cultiver ce sentiment de possession quand ça les arrange; ce n'est pas inévitable. Considérez, par exemple, comment ces mêmes programmeurs cèdent volontiers leurs droits à une grosse entreprise moyennant salaire ; mystérieusement, l'attachement affectif disparaît. Faites le contraste avec ces grands artistes et artisans des temps médiévaux, qui ne signaient même pas de leur nom leurs travaux. Pour eux, le nom de l'artiste n'avait pas d'importance. Ce qui importait, c'était que le travail, et son but afférent, aient été faits. Cette façon de voir à prévalu pendant des centaines d'années.

L'argument économique est du style : « je veux devenir riche (ce qui en général, se dit incorrectement « je veux gagner ma vie »), et si vous ne me permettez pas de devenir riche en programmant, eh bien je ne programmerai pas. Comme tout le monde me ressemble, personne n'écrira de programmes. Et vous serez coincés, car pas de programme du tout ! » Cette menace est généralement déguisée en conseil amical venant de la bouche d'un sage.

J'expliquerai plus tard pourquoi cette menace est du bluff. J'aimerais d'abord mettre le doigt sur une supposition implicite qui est plus évidente dans une autre formulation de l'argument.

Cette formulation part de la comparaison entre l'utilité sociale d'un programme propriétaire et celle où il n'y a pas de programme, pour alors conclure que le développement de logiciels propriétaires est, globalement, bénéfique et qu'il devrait être encouragé. L'erreur, ici, vient de ne comparer que deux résultats

— logiciel propriétaire contre pas de logiciel — et de supposer qu'il n'y a pas d'autres possibilités.

Dans un système reconnaissant la propriété intellectuelle, le développement logiciel est habituellement lié à l'existence d'un propriétaire qui contrôle l'utilisation du logiciel. Tant que ce lien existe, nous devons souvent faire face au choix d'un programme propriétaire ou pas de programme. Cependant, ce lien n'est pas inhérent ou inévitable; c'est une conséquence d'une décision politique, légale et sociale spécifique, que nous contestons : la décision qu'il y ait des propriétaires. Formuler le choix entre logiciel propriétaire contre pas de logiciel, c'est faire une pétition de principe.

L'argument contre le fait qu'il y ait des propriétaires

La question qu'il faut poser, c'est : « Est-ce que le développement de logiciels doit être lié à un propriétaire qui en restreint l'usage ? »

Pour pouvoir en décider, il nous faut juger chaque effet des deux activités suivantes sur la société, *indépendamment* : l'effet du développement de logiciels (indépendamment de ses termes de diffusion), et l'effet de la restriction de son emploi (supposant que le logiciel ait été développé). Si l'une de ces activités est utile et l'autre nocive, nous devrions les dissocier et nous lancer uniquement dans la première.

En d'autres termes, si restreindre la distribution d'un logiciel déjà développé est préjudiciable à la société dans son ensemble, alors un développeur moral rejettera cette activité.

Pour déterminer l'effet de la restriction du partage, nous avons besoin de comparer la valeur, pour la société, d'un programme restreint (par ex. propriétaire) avec le même programme, mais disponible pour tout le monde. Ce qui signifie comparer deux mondes possibles.

Cette analyse met aussi en exergue le simple contre-argument qui est parfois soulevé que « le bénéfice pour le voisin en lui donnant une copie d'un logiciel est annulé par le préjudice fait au propriétaire ». Ce contre-argument suppose que les inconvénients et bénéfices sont équivalents dans leur ampleur. L'analyse implique de comparer ces étendues et elle montre que les bénéfices sont bien plus importants.

Pour mettre en lumière cet argument, prenons un autre domaine d'application : la construction routière.

Il serait possible de financer la construction de toutes les routes avec des péages. Ce qui entraînerait d'avoir des postes de péage à tous les coins de rues. Un tel système inciterait grandement à améliorer l'état des routes. Il aurait aussi comme vertu de faire payer l'usager de la route concernée. Cependant, le péage n'est qu'une entrave artificielle à la fluidité du trafic, artificielle dans le sens où elle n'est pas une conséquence du fonctionnement des routes et des voitures.

Si on compare les routes avec ou sans péage, nous voyons que (si tout se déroule normalement) les routes sans péage sont meilleur marché à construire et à maintenir, plus sûres et plus efficaces à emprunter. (2) Dans les pays pauvres, les postes de péage rendent les routes inaccessibles à bien des citoyens. Les routes sans péage offrent ainsi plus de bénéfices à moindre coût ; elles sont préférables pour la société. C'est pourquoi la société devrait trouver d'autres moyens de financer les routes, sans recourir aux péages. L'usage des routes, une fois construites, devrait être libre.

Quand les partisans des postes de péage les proposent comme étant *simplement* une façon de lever des fonds, ils déforment le choix offert. Les postes de péage, effectivement, permettent de récolter des fonds, mais elles occasionnent aussi autre chose : elles dégradent les routes. La route à péage n'est pas aussi bonne que celle sans péage; nous donner plus de routes ou supérieures sur le plan technique, n'est peut-être pas une amélioration si cela signifie substituer aux routes gratuites des routes à péage.

Bien entendu, construire des routes sans péage coûte de l'argent, que, d'une façon ou d'une autre, le public doit payer. Mais ce n'est pas ce qui, inévitablement, implique des postes de péage. Nous, qui dans un cas comme dans l'autre, devons payer, aurons tout intérêt pour notre porte-monnaie à acheter des routes sans péage.

Je ne suis pas en train de dire que les routes à péage sont pires que pas de route du tout. Ce serait vrai si les péages étaient tels que presque personne n'emprunterait la route — mais ce n'est pas une politique plausible pour un collecteur de péage. Cependant, tant que les postes de péage causeront des gaspillages et

des inconvénients significatifs, il est plus avantageux de lever des fonds de façon moins obstructive.

Pour appliquer ce même argument au développement logiciel, je vais maintenant montrer que d'avoir des « postes de péage » sur des logiciels utiles, coûte cher à la société : cela rend le programme plus coûteux à élaborer, plus cher à distribuer et moins satisfaisant et efficace à utiliser. Je poursuivrai en disant que la construction du programme devra être encouragée autrement. Puis je m'attacherai à présenter d'autres méthodes d'encouragement et de financement du développement logiciel (dans la mesure réellement nécessaire).

Le tort fait en entravant le logiciel

Considérez un instant un programme qui a été développé et dont tous les financements nécessaires à son élaboration ont été pris ; maintenant, la société doit faire le choix entre le rendre propriétaire ou le rendre libre d'utilisation et de partage. Supposez qu'il est désiré que ce programme existe et soit disponible.(3)

Les restrictions sur la distribution et la modification du programme ne facilitent pas son utilisation. Elles ne peuvent qu'interférer. Ainsi, l'effet ne peut avoir qu'un impact négatif. Mais jusqu'à quel point ? Et de quelle manière ?

On distingue trois niveaux de préjudice matériel dans ce genre d'entrave :

- Moins de gens utilisent le programme.
- Aucun des utilisateurs ne peut adapter ou corriger le programme.
- Les autres développeurs ne peuvent rien apprendre du programme ou encore démarrer un nouveau travail en se basant dessus.

Chaque niveau de préjudice matériel est concomitant à un préjudice psycho-social. Cela se réfère aux effets qu'ont les décisions des gens sur leurs sentiments, attitudes et prédispositions futurs. Ces changements dans la façon de penser des gens auront par la suite un effet sur leurs relations avec leurs concitoyens et peuvent avoir des conséquences matérielles.

Les trois niveaux de préjudice matériel gaspillent une partie de la valeur que le programme pourrait offrir, mais ne peuvent la réduire à zéro. S'ils gaspillent la presque totalité de la valeur du programme, alors l'écriture du programme cause du tort à la société, au plus à hauteur de l'effort qu'il a fallu fournir pour écrire ce programme. En effet, un programme dont la vente génère des profits fournit sûrement un net bénéfice matériel direct.

Cependant, tenant compte des préjudices psycho-sociaux concomitants, il n'y a pas de limite au préjudice que peut provoquer le développement d'un logiciel propriétaire.

L'entrave à l'utilisation de programmes

Le premier niveau de préjudice gêne le simple emploi d'un programme. Une copie d'un programme a pratiquement un coût marginal de zéro (et ce prix, vous pouvez le payer en faisant le travail vous-même), ce qui veut dire que, dans un marché libre, elle aurait un prix avoisinant zéro. Une taxe via une licence est un découragement significatif à l'utilisation d'un programme. Si un logiciel largement utile est propriétaire, beaucoup moins de gens l'utiliseront.

Il est facile de montrer que la contribution totale d'un programme à la société est réduite si on lui assigne un propriétaire. Chaque utilisateur potentiel du logiciel, face à la nécessité de le payer pour l'utiliser, peut choisir de payer ou de renoncer à son usage. Si un utilisateur fait le choix de payer le programme, il y a un simple transfert de richesses entre deux parties. Mais chaque fois qu'une personne choisit d'outrepasser l'utilisation du programme, cela cause du tort à cette personne sans que quelqu'un y trouve son compte. La somme de nombres négatifs avec zéro, ça doit être négatif.

Mais cela ne réduit pas la somme de travail qu'il faut pour *développer* le programme. Donc l'efficacité du processus, calculée en divisant la satisfaction des utilisateurs par le coût du développement, en est diminuée.

Ceci reflète la différence cruciale entre la copie de programmes et celle de voitures, de chaises ou de

sandwiches. À part dans la science-fiction, il n'existe pas de machine pouvant reproduire les objets matériels. Mais les programmes sont faciles à copier; n'importe qui peut faire autant de copies que nécessaire, sans gros effort. Ce qui n'est pas vrai dans le cas des objets matériels, vu que la matière est conservée : chaque copie nécessite des matières premières, tout comme la première.

En ce qui concerne les objets matériels, décourager leur usage est logique, car moins d'objets signifie moins de matières premières, moins de travail pour les construire. C'est vrai qu'il faut un coût pour démarrer, pour développer, coût réparti sur l'ensemble de la production. Mais tant que le coût marginal est significatif, y ajouter un pourcentage des coûts de développement ne provoque pas de différence qualitative. Et cela ne nécessite pas de restrictions sur la liberté des utilisateurs.

Cependant, imposer un prix à quelque chose qui, autrement, aurait pu être gratuit, c'est un changement qualitatif. Une taxe imposée, centralisée, sur la distribution de logiciels devient fortement décourageante.

De plus, la production centrale, telle qu'elle est pratiquée de nos jours, est inefficace, y compris dans son rôle de fournisseur de copies de logiciels. Ce système implique d'emballer des disquettes ou des bandes dans un emballage superflu, de les envoyer en nombre dans le monde entier puis de les stocker avant leur vente. Ces coûts sont présentés comme étant le prix à payer pour faire des affaires ; en vérité, ils font partie du gaspillage causé par la présence d'un propriétaire.

Endommager la cohésion sociale

Supposons que vous et votre voisin trouviez utile de faire tourner un certain programme. Dans un souci moral pour votre voisin, vous devriez sentir que la façon correcte d'appréhender la situation, est de permettre une utilisation de ce logiciel par vous deux. S'il n'y avait la possibilité que pour un seul d'entre vous de faire tourner ce programme, il y aurait division ; ni vous, ni votre voisin ne trouveriez cela acceptable.

Signer un accord de licence logicielle typique revient à trahir votre voisin : « je fais la promesse de priver mon voisin de ce programme, ainsi je peux en avoir une copie pour moi-même ». Les gens qui font de tels choix ressentent une pression psychologique interne pour se justifier, en diminuant l'importance d'aider leur voisin — c'est comme cela que le sens civique souffre. C'est un préjudice psycho-social associé au préjudice matériel de décourager l'utilisation du logiciel.

Beaucoup d'utilisateurs reconnaissent inconsciemment le tort de refuser le partage ; ils décident alors d'ignorer les licences et les lois, et de partager tout de même les programmes. Mais ils s'en sentent souvent coupables. Ils savent qu'ils doivent enfreindre la loi pour être de bons voisins, mais ils continuent de penser que les lois font autorité et concluent qu'être un bon voisin (ce qu'ils sont), c'est vilain et honteux. C'est aussi une forme de préjudice psycho-social, mais on peut y échapper en prenant la décision que ces licences et ces lois n'ont pas de force morale.

Les programmeurs souffrent aussi de préjudices psycho-sociaux, sachant que bien des usagers ne pourront utiliser leurs travaux. Ceci conduit à une attitude de cynisme ou de dénégation. Un programmeur peut faire une description enthousiaste du travail qu'il pense techniquement excitant; puis, quand on lui demande « Est-ce qu'il me sera permis de l'utiliser ? », son visage se ferme et il doit bien admettre que non. Pour éviter de se sentir découragé, soit il ignore ce fait la plupart du temps, soit il adopte une attitude cynique afin d'en minimiser l'importance.

Depuis la période reaganienne, la plus grande pénurie, aux États-Unis, ce n'est pas l'innovation technique, mais plutôt la volonté de travailler ensemble pour le bien public. Cela n'a pas de sens d'encourager l'un au détriment de l'autre.

L'entrave à l'adaptation sur mesure des programmes

Le deuxième niveau de préjudice matériel est l'impossibilité d'adapter les programmes. La facilité de modification du logiciel est un de ses grands avantages sur les technologies plus anciennes. Mais la plupart des logiciels commerciaux ne peuvent être modifiés, même après les avoir achetés. Ils sont à prendre ou à laisser, comme une boîte noire, un point c'est tout.

Un programme que vous pouvez exécuter se compose d'une série de nombres dont le sens est obscur. Personne, même un bon programmeur, ne peut aisément changer les nombres afin que le logiciel exécute autre chose.

Normalement, les programmeurs travaillent sur le « code source » d'un programme, écrit dans un langage de programmation comme le Fortran ou le C. Ils utilisent des noms pour désigner les données utilisées et des parties du programme, et ils représentent les opérations par des symboles comme le « + » pour une opération ou le « - » pour une soustraction. Le langage est conçu pour aider les programmeurs à déchiffrer et modifier les programmes. Voici un exemple : il s'agit d'un programme qui calcule la distance entre deux points d'un plan :

```
float
distance (p0, p1)
    struct point p0, p1;
{
    float xdist = p1.x - p0.x;
    float ydist = p1.y - p0.y;
    return sqrt (xdist * xdist + ydist * ydist);
}
```

Voici le même programme sous sa forme exécutable, sur l'ordinateur que j'utilise habituellement :

```
1314258944      -232267772      -231844864      1634862
1411907592      -231844736      2159150         1420296208
-234880989      -234879837      -234879966      -232295424
1644167167      -3214848        1090581031      1962942495
572518958       -803143692      1314803317
```

Le code source est utile (au moins potentiellement) pour chaque utilisateur d'un programme. Mais la plupart des utilisateurs n'ont pas la permission d'avoir des copies du code source. Normalement, le code source d'un programme propriétaire est tenu secret par son propriétaire, empêchant quiconque d'en apprendre quelque chose. L'utilisateur reçoit simplement des fichiers de nombres incompréhensibles que l'ordinateur exécutera. Cela signifie que seul le propriétaire du logiciel peut changer le programme.

Un jour, une amie me dit qu'elle travaillait comme programmeur dans une banque depuis environ six mois, écrivant un programme similaire à quelque chose de commercialement disponible. Elle croyait que, si elle avait accès au code source de ce programme commercial, elle pourrait facilement l'adapter à ses besoins. La banque souhaitait payer pour cela, mais elle n'y fut pas autorisée — le code source était un secret. Elle a dû alors travailler d'arrache-pied pendant six mois, un travail qui compte pour le PNB, mais qui était en fait du gaspillage.

Le Laboratoire d'Intelligence Artificielle du MIT reçut une imprimante graphique comme cadeau de la part de Xerox, aux alentours de 1977. Elle était pilotée par un logiciel libre, auquel nous avons ajouté de nombreuses caractéristiques bien commodes. Par exemple, le logiciel avertissait immédiatement l'utilisateur de la fin du processus d'impression. Si l'imprimante venait à rencontrer un problème, comme un bourrage ou un manque de papier, le programme avertissait de suite tous ceux qui avaient des travaux d'impression en cours. Ces fonctionnalités facilitaient la vie.

Plus tard, Xerox offrit au labo d'I.A. une nouvelle imprimante, plus rapide, une des premières laser. Elle était pilotée par un logiciel propriétaire qui tournait sur un poste dédié et séparé, nous ne pouvions donc ajouter aucune de nos fonctionnalités favorites. On pouvait s'arranger pour envoyer un message quand le travail d'impression se lançait sur le poste dédié, mais pas quand celui-ci se faisait effectivement (et les délais étaient habituellement importants). Il n'y avait aucun moyen de savoir si l'impression était faite; il fallait deviner. Et personne n'était informé d'un bourrage papier, l'imprimante attendait ainsi souvent une heure avant d'être rechargée.

Les programmeurs système du labo d'I.A. étaient capables de corriger de tels problèmes, probablement tout aussi capables que les auteurs du programme. Xerox n'avait pas envie de les corriger et

choisit de nous en empêcher, nous avons donc été forcés de subir les problèmes. Ils n'ont jamais été corrigés.

La plupart des bons programmeurs ont fait l'expérience de cette frustration. La banque pouvait se permettre de résoudre son problème en écrivant un nouveau programme depuis le début, mais un utilisateur lambda, quelle que soit son habileté, ne peut que laisser tomber.

Laisser tomber provoque un préjudice psycho-social — sur l'esprit d'indépendance. C'est démoralisant d'habiter une maison qu'on ne peut réarranger selon ses besoins. Cela conduit à la résignation et au découragement, ce qui peut gagner d'autres aspects de la vie. Les gens qui se sentent ainsi ne sont pas heureux et ne font pas du bon travail.

Imaginez ce que ce serait si les recettes étaient logées à la même enseigne que les logiciels. Vous vous diriez « Voyons, comment modifier cette recette pour qu'il n'y ait plus de sel ? », et le chef cuisinier de vous répondre « Comment oses-tu insulter ma recette, fruit de mon cerveau et de mon palais, en tentant de la modifier ? Tu n'as pas le jugement pour changer ma recette afin qu'elle marche mieux ».

« Mais mon docteur m'a recommandé de ne pas manger salé. Que puis-je faire ? Pouvez-vous en ôter le sel pour moi ? »

« Je serais heureux de le faire ; mes honoraires ne sont que de 50000 dollars ». À partir du moment où le propriétaire a le monopole sur les modifications, les honoraires tendent à gonfler. « De toute façon, je n'ai pas le temps maintenant. Je suis pris par une commission afin de créer une nouvelle recette de biscuits marins pour le Département de la Marine. Je reprendrai contact avec toi d'ici à peu près deux ans ».

L'entrave au développement logiciel

Le troisième niveau de préjudice matériel touche le développement logiciel. Il était autrefois un processus évolutif, où quelqu'un prenait un programme existant et en réécrivait une partie pour ajouter une nouvelle fonctionnalité; puis une autre personne en réécrivait aussi une partie pour y ajouter une autre fonctionnalité. Dans certains cas, cela a continué ainsi sur une période d'une vingtaine d'années. Entre-temps, certaines parties du programme auront été « cannibalisées » pour créer les prémices d'autres programmes.

L'existence de propriétaires empêche ce genre d'évolution, rendant nécessaire de repartir de rien si on veut développer un programme. Cela empêche également les nouveaux praticiens d'étudier les programmes existants pour en apprendre des techniques utiles ou même apprendre comment on structure de gros programmes.

Les propriétaires entravent aussi l'éducation, l'apprentissage. J'ai rencontré de brillants étudiants en informatique qui n'avaient jamais vu le code source d'un gros logiciel. Ils peuvent être bons à écrire de courts programmes, mais ils ne peuvent commencer à apprendre les techniques différentes de l'écriture d'un vaste programme, s'ils ne peuvent observer comment d'autres l'ont fait.

Dans tout domaine intellectuel, on peut atteindre de plus grandes hauteurs en se tenant sur les épaules des autres. Mais ce n'est généralement plus permis dans le domaine logiciel — vous ne pouvez vous tenir sur les épaules que de ceux qui font partie de *vostra propre compagnie*.

Le préjudice psycho-social qui s'y rattache affecte l'esprit de coopération scientifique, qui était autrefois si fort que les scientifiques coopéraient même quand leurs pays étaient en guerre. C'est dans cet esprit que les océanographes japonais, abandonnant leur labo dans une île du Pacifique, ont soigneusement conservé leurs travaux pour les Marines qui commençaient à débarquer, et laissèrent un mot leur demandant d'en prendre bien soin.

Les conflits de profits ont détruit ce que les conflits internationaux avaient épargné. Aujourd'hui, les scientifiques de nombreuses disciplines ne donnent pas assez de détails dans leurs publications, qui permettraient aux autres de reproduire leur expérience. Ils ne publient que ce qui permet au lecteur d'être impressionnés par l'étendue de leurs travaux. C'est particulièrement vrai pour la recherche informatique, où le code source des programmes décrits dans les publications est en général secret.

Peu importe comment le partage est restreint

J'ai parlé des effets d'empêcher les gens de copier, de modifier ou de se baser sur un programme existant. Je n'ai pas précisé comment cette obstruction était réalisée, car cela n'affecte pas la conclusion. Que ce soit par protection contre la copie, copyright, licences, cryptage, cartes ROM ou encore un numéro de série sur le matériel, si cela *réussit* à empêcher l'utilisation, alors il y a préjudice.

Les utilisateurs considèrent certaines de ces méthodes comme plus odieuses que d'autres. Je suggère que les méthodes les plus détestées sont celles qui accomplissent leur objectif.

Les logiciels devraient être libres

J'ai montré comment le fait d'être propriétaire d'un programme, le pouvoir de restreindre sa modification ou sa copie, est une entrave. Ses retombées négatives sont vastes et importantes. Il s'ensuit que la société devrait se passer de propriétaires de logiciels.

Une autre façon de comprendre cela, est que ce dont a besoin la société, c'est de logiciels libres et que les logiciels propriétaires ne sont qu'un pauvre substitut. Encourager le substitut n'est pas une façon rationnelle d'obtenir ce dont nous avons besoin.

Vaclav Havel nous a conseillé de «travailler pour une chose parce qu'elle est bien, pas parce qu'elle a des chances de réussir». Un marché créant des logiciels propriétaires a des chances de réussir selon son propre point de vue, mais ce n'est pas ce qui est bon pour la société.

Pourquoi les gens développeront des logiciels

Si nous éliminons la propriété intellectuelle comme un moyen d'encourager les gens à développer des logiciels, au début, peu de programmes seront développés, mais ils seront plus utiles. Difficile de dire si la satisfaction d'ensemble des utilisateurs sera moindre. Mais si c'est le cas, ou si nous voulons malgré tout l'augmenter, il y a d'autres moyens d'encourager le développement, tout comme il y a des alternatives aux postes de péage pour tirer de l'argent des routes. Mais avant de parler de la façon dont cela peut se faire, je vais d'abord me demander dans quelle mesure un encouragement artificiel est vraiment nécessaire.

Programmer, c'est amusant

Certains domaines professionnels trouvent peu de candidats, sauf pour l'argent; la construction routière, par exemple. Il en est d'autres, touchant aux études ou à l'art, dans lesquelles il y a peu de chance de devenir riche, mais où les gens s'engagent par fascination ou à cause de sa valeur perçue pour la société. On peut y inclure par exemple, les mathématiques logiques, la musique classique et l'archéologie; puis l'organisation politique au sein des travailleurs. Les gens concourent, plus tristement qu'âprement, pour les quelques situations assises disponibles, aucune d'entre elles n'étant vraiment bien solide. Ils paieraient pour avoir la chance de travailler dans un de ces domaines, s'ils le pouvaient.

Un domaine peut se transformer du jour au lendemain, s'il commence à offrir la possibilité de devenir riche. Si un travailleur devient riche, les autres réclament la même opportunité. Bientôt tous demanderont de fortes sommes d'argent pour ce qu'ils avaient l'habitude de faire pour le plaisir. Puis quelques années passent, chaque personne en relation avec ce domaine tournera en dérision l'idée que le travail pourrait être fait sans salaires mirobolants. Ils conseilleront aux acteurs sociaux de s'assurer que de tels salaires soient possibles, en prescrivant des privilèges spéciaux et les pouvoirs, monopoles nécessaires pour que cela puisse se faire.

Ce changement est apparu dans le domaine de la programmation cette dernière décennie. Il y a quinze ans, des articles parlaient d'« accros à l'informatique » : les utilisateurs étaient « connectés » et vivaient modestement. Il était généralement compris que les accros de la programmation pouvaient briser leur couple. Aujourd'hui, il est généralement admis que personne ne ferait de la programmation, sans salaire élevé. Les gens ont oublié ce qu'ils savaient il y a quinze ans.

Même si à un moment précis, la plupart des gens travailleront dans un certain domaine uniquement pour un haut salaire, cela ne durera pas forcément. La tendance peut s'inverser, si la société donne une impulsion. Si nous laissons de côté les possibilités de gros gains, peu de temps après, quand les gens auront

réajusté leurs attitudes, ils auront à nouveau à coeur de travailler dans leur domaine pour la joie de le faire.

La question « comment payer un programmeur » devient plus simple, quand nous réalisons que ce n'est pas la peine de les payer une fortune. Il est plus facile de leur assurer un niveau de vie correct sans plus.

Financer le logiciel libre

Les institutions qui payent les programmeurs n'ont pas besoin d'être des « boîtes à logiciels ». Beaucoup d'autres institutions existantes peuvent le faire.

Les constructeurs de matériels informatiques pensent qu'il est essentiel de supporter le développement logiciel, même s'ils ne peuvent contrôler l'usage du logiciel. En 1970, la plupart de leurs logiciels étaient libres, car ils ne pensaient pas à les entraver. Aujourd'hui, la volonté croissante de se joindre à des consortiums montre leur compréhension que de posséder le logiciel n'est pas ce qui est vraiment important pour eux.

Les universités mènent de nombreux projets de programmation. Aujourd'hui, elles en vendent souvent les résultats, mais, dans les années 70, elles ne le faisaient pas. Douterait-on que les universités développeraient des logiciels libres si elles n'étaient pas autorisées à vendre des logiciels ? Ces projets pourraient recevoir le soutien de contrats gouvernementaux et de bourses qui soutiennent actuellement le développement de logiciels propriétaires.

Il est commun de nos jours que les chercheurs universitaires reçoivent une bourse pour développer un système, qu'ils le développent presque jusqu'à la finalisation, qu'ils le déclarent « fini », puis qu'ils créent des sociétés où ils le finiront effectivement et le rendront utilisable. Parfois, ils déclarent « libre » la version non terminée; s'ils sont vraiment corrompus, ils obtiendront une licence exclusive de la part de l'université. Ce n'est pas un secret, c'est ouvertement admis par les personnes concernées. Pourtant, si les chercheurs n'étaient pas exposés à ces tentations, ils continueraient quand même leurs recherches.

Les programmeurs qui écrivent des logiciels libres peuvent gagner leur vie en vendant des services liés au logiciel. J'ai reçu des honoraires pour le portage du [compilateur GNU C](#) sur un nouveau matériel et pour faire des extensions d'interfaces utilisateurs pour [GNU Emacs](#). (J'ai offert ces améliorations au public, une fois qu'elles ont été réalisées). Je suis aussi payé pour enseigner dans des classes.

Je ne suis pas seul à travailler de cette façon; il existe maintenant une entreprise fructueuse, grandissante qui ne fait pas autrement. Plusieurs autres compagnies offrent aussi un support commercial aux logiciels libres issus du système GNU. C'est le début d'une industrie indépendante du support du logiciel libre, une industrie qui pourrait prendre de fortes proportions, si le logiciel libre devenait courant. Elle offre aux utilisateurs des options qui sont généralement indisponibles dans le cas de logiciels propriétaires, sauf pour les plus riches.

De nouvelles institutions, comme la [Free Software Foundation](#), peuvent aussi financer les programmeurs. La majorité des fonds de la Fondation vient de l'argent récolté dans la vente de bandes par correspondance. Le logiciel présent sur la bande est libre, ce qui signifie que chaque utilisateur est libre de le copier et de le modifier, mais néanmoins, beaucoup payent pour en obtenir des copies. (Rappelez-vous que « free software » veut dire libre, et non gratuit). Certains utilisateurs achètent des bandes, alors qu'ils en possèdent déjà, simplement parce qu'ils sentent que nous méritons cette contribution. La Fondation reçoit aussi des donations considérables de la part de fabricants d'ordinateurs.

La Free Software Foundation est une organisation caritative et ses revenus sont dépensés en employant le plus possible de programmeurs. Si elle avait été érigée en business, distribuant les mêmes logiciels libres au public pour la même somme, elle pourrait maintenant offrir un très bon niveau de vie à son fondateur.

Parce que la Fondation est une organisation caritative, les programmeurs travaillent souvent pour la moitié de qu'ils pourraient toucher ailleurs. Ils le font parce qu'ils sont libres de toute bureaucratie et parce qu'ils ressentent de la satisfaction à savoir que leur travail pourra être utilisé par tous. Et, par-dessus tout, ils le font parce que programmer, c'est passionnant. Ajoutons que des volontaires nous ont écrit nombre de programmes (même des rédacteurs techniques ont récemment commencé à se proposer).

Ce qui confirme que la programmation est parmi les domaines les plus fascinants, au même titre que la musique et les arts. Nous n'avons pas à craindre que plus personne ne veuille programmer.

Que doivent les utilisateurs aux programmeurs ?

Il y a une bonne raison à ce que les utilisateurs de logiciels se sentent obligés moralement à contribuer à leur soutien. Les développeurs de logiciels libres contribuent à l'activité des utilisateurs, c'est à la fois loyal et — sur le long terme — aussi dans l'intérêt des utilisateurs que de les financer.

Cependant, ceci ne s'applique pas aux développeurs de logiciels propriétaires, vu que l'obstructionnisme appelle plutôt une sanction qu'une récompense.

Nous nous trouvons ainsi face à un paradoxe : le développeur de logiciels utiles a droit au soutien des utilisateurs, mais n'importe quelle tentative de transformer cette obligation morale en une exigence, détruit les bases de l'obligation. Un développeur peut soit recevoir une récompense, soit l'exiger, mais pas les deux.

Je crois qu'un développeur moral faisant face à ce paradoxe doit agir de manière à mériter la récompense, mais devrait aussi encourager les utilisateurs à donner volontairement. Tôt ou tard, les utilisateurs apprendront à soutenir les développeurs d'eux-mêmes, tout comme ils ont appris à soutenir les radios et les stations télé indépendantes.

Qu'est-ce que la productivité logicielle ?

Si les logiciels étaient libres, il y aurait toujours des programmeurs, mais peut-être en nombre moindre. Est-ce que cela serait mauvais pour la société ?

Pas forcément. Aujourd'hui, les nations riches ont moins de fermiers qu'en 1900, mais nous ne pensons certainement pas que cela est mauvais pour la société, car ceux qui restent produisent plus de nourriture pour les consommateurs que tous ceux, plus nombreux, jadis. Nous appelons cela l'amélioration de la productivité. Le logiciel libre devrait demander moins de programmeurs pour satisfaire la demande, à cause de l'augmentation de la productivité logicielle à tous niveaux :

- Une utilisation plus large de chaque programme développé.
- La possibilité d'adapter un programme existant pour le personnaliser au lieu de repartir de zéro.
- Une meilleure instruction des programmeurs.
- L'élimination des redondances dans l'effort de développement.

Ceux qui font objection à la coopération dans la mesure où elle diminuerait l'emploi des programmeurs s'opposent en fait à l'accroissement de la productivité. Pourtant les mêmes acceptent souvent la croyance largement répandue que l'industrie logicielle a besoin d'accroître sa productivité. Comment cela se fait-il ?

La « productivité logicielle » peut vouloir dire deux choses : la productivité générale de tout développement logiciel ou la productivité de projets individuels. La productivité générale, c'est ce que la société aimerait améliorer et la voie la plus directe pour le faire est d'éliminer les obstacles artificiels à la coopération qui la réduisent. Mais les chercheurs qui se penchent sur la « productivité logicielle » se focalisent uniquement sur le deuxième sens, limité, du terme, où l'amélioration demande des avancées technologiques difficiles.

Est-ce que la compétition est inévitable ?

Est-il inévitable que les gens se mettent en concurrence, qu'ils essayent de dépasser leurs rivaux dans la société ? Peut-être, oui. Mais la compétition en elle-même n'est pas nocive : ce qui est nocif, c'est le *combat*.

Il y a plusieurs façons de concourir. La compétition peut se présenter comme essayer d'aller plus loin, comme surpasser ce que d'autres ont déjà réalisé. Par exemple, jadis, il y avait concurrence entre les meilleurs programmeurs, afin que l'ordinateur fasse les choses les plus incroyables possible, ou encore à qui écrira le programme le plus court, le plus rapide pour une tâche donnée. Ce genre de compétition est

bénéfique pour tous, *tant que* l'esprit de saine émulation est maintenu.

Une compétition constructive est suffisante pour pousser les gens à de grands efforts. Certains concourent pour être les premiers à avoir visité tous les pays du globe ; il y en a même qui dépensent des fortunes pour cela. Mais ils ne corrompent pas les capitaines de navire pour que leurs rivaux soient échoués une île déserte. Ils se satisfont de laisser le meilleur gagner.

La compétition devient un combat quand les participants commencent à entraver les autres plutôt que de progresser eux-mêmes — c'est-à-dire quand le « que le meilleur gagne » fait la place au « laissez-moi gagner, que je sois le meilleur ou non ». Le logiciel propriétaire est nocif, non parce qu'il est une forme de compétition, mais parce qu'il est une forme de combat au sein des citoyens de notre société.

La compétition dans les affaires n'est pas forcément un combat. Par exemple, lorsque deux épiceries sont en compétition, tout leur effort tend vers l'amélioration de leurs propres services, pas à saboter leur rival. Mais ce n'est pas ce qui démontre un engagement moral dans les affaires; plutôt qu'il existe une faible zone de combat dans ce genre de business, à la limite de la violence physique. Tous les domaines des affaires ne partagent pas cette caractéristique. La rétention d'information utile à tous, c'est une forme de combat.

L'idéologie, dans les affaires, ne prépare pas les gens à résister à la tentation de combattre, dans une compétition. Certaines formes de combat ont été interdites avec les lois anti-trusts, l'interdiction de la publicité mensongère, etc., mais plutôt que de généraliser le rejet du combat, les dirigeants inventent en général d'autres formes de combat qui ne sont pas spécifiquement prohibées. Les ressources de la société sont gaspillées économiquement, à l'instar d'une guerre civile entre factions.

« Pourquoi n'irais-tu pas en Russie ? »

Aux États-Unis, toute personne favorable à autre chose que le plus flagrant laissez-faire égoïste a souvent entendu ce genre de réflexion. On l'entend, par exemple, à l'encontre des partisans d'un système de santé publique, comme on en trouve dans les autres nations industrialisées. Ou encore à propos des partisans d'un soutien public des arts, tout aussi universel chez les nations avancées. L'idée que les citoyens aient une quelconque obligation de participer au bien public est considérée comme du communisme, aux États-Unis. Mais ce terme est-il bien approprié ?

Le communisme, comme il était pratiqué en Union Soviétique, était un système de contrôle centralisé, où toutes les activités étaient passées au crible du régime, soi-disant pour le bien public, mais en fait pour le bien des membres du Parti Communiste. Système où les appareils permettant les copies étaient étroitement gardés, pour empêcher les copies illégales.

Le système américain de la propriété intellectuelle exerce un contrôle central sur la distribution d'un programme et surveille les copieurs grâce à des [systèmes automatiques de protection contre la copie](#), pour empêcher les copies illégales.

Par opposition, je travaille à bâtir un système où les gens sont libres de décider de leurs propres actions; en particulier, libres d'aider leur voisin, de modifier et d'améliorer les outils qu'ils utilisent dans leur vie quotidienne. Un système basé sur la coopération volontaire et la décentralisation.

Du coup, si on doit juger ces points de vue par leurs ressemblances au communisme soviétique, alors ce sont les propriétaires de logiciels qui sont les communistes.

La question des prémisses

Je fais la supposition, dans cet article, que l'utilisateur d'un logiciel n'est pas moins important qu'un auteur ou même l'employeur d'un auteur. Autrement dit, lorsqu'on décide quelle est la meilleure marche à suivre, leurs intérêts et leurs besoins ont autant d'importance.

Cette prémisse n'est pas universellement acceptée. Beaucoup maintiennent que l'employeur d'un auteur est fondamentalement plus important que n'importe qui d'autre. Ils disent, par exemple, que le but, d'avoir des propriétaires de logiciels, est de donner à l'employeur les avantages qui lui sont dûs — indépendamment de l'effet sur le public.

Cela ne sert à rien de prouver ou non ces prémisses. Prouver demande des prémisses partagées. C'est pourquoi la majorité de mon discours s'adresse à ceux qui partagent les prémisses que j'utilise, ou qui, au moins, sont intéressés par leurs conséquences. Pour ceux qui croient que les propriétaires sont plus importants que tous les autres, pour ceux-là, cet article n'est tout simplement pas pertinent.

Mais pourquoi un grand nombre d'Américains accepteraient une prémisse qui élèverait certaines personnes au-dessus des autres ? En partie à cause de la croyance que cette prémisse fait partie des traditions légales de la société américaine. Il y a des gens qui pensent que douter de la prémisse, c'est défier les bases de la société.

Il est important pour ces gens de savoir que cette prémisse ne fait pas partie de notre tradition légale. Ne l'a jamais été.

Ainsi, la Constitution dit que le but du copyright est de « promouvoir le progrès des sciences et des arts utiles ». La cour Suprême l'a élaboré ainsi, énonçant dans la « Fox Film contre Doyal » que « l'intérêt unique des États-Unis ainsi que l'objet principal du monopole [du copyright], résident dans les bénéfices que retire le public du travail des auteurs ».

Nous ne sommes pas obligés d'approuver la Constitution ou la Cour Suprême (il fut un temps où les deux ont approuvé l'esclavage). Leurs positions ne réfutent donc pas la prémisse de la suprématie du propriétaire. Mais j'espère que la conscience qu'il s'agit d'une supposition de la droite radicale, plutôt que d'une tradition reconnue, affaiblira son intérêt.

Conclusion

Nous aimons penser que notre société encourage à aider son voisin ; mais chaque fois que nous récompensons quelqu'un pour son obstructionnisme ou que nous l'admirons pour les richesses qu'il a accumulées ainsi, nous renvoyons le message contraire.

La thésaurisation de logiciels est un exemple de notre volonté d'ignorer le bien-être de la société pour le gain personnel. On peut en voir la trace depuis Ronald Reagan jusqu'à Dick Cheney, depuis Exxon à Enron, en passant par les échecs des banques et des écoles. Nous pouvons la mesurer à l'aune des sans-abri et de la population dans les prisons. L'esprit antisocial se nourrit de lui-même, parce que plus on voit que les autres ne nous tendront pas la main, plus il nous semble futile de les aider. Ainsi, notre société dégénère en jungle.

Si nous ne voulons pas vivre dans une jungle, nous devons changer nos attitudes. Nous devons commencer à lancer le message qu'un bon citoyen est un citoyen qui coopère quand il le faut, que ce n'est pas celui qui réussit en volant les autres. J'espère que le mouvement du logiciel libre contribuera à cela : au moins dans un domaine, nous remplacerons la jungle par un système plus efficace qui encouragera et se nourrira la coopération volontaire.

Notes

1. Le mot « free » dans « free software » signifie libre, et non gratuit [free désigne les deux termes, en anglais] ; le prix payé pour une copie d'un programme libre peut être nul, faible, ou (rarement) assez élevé.
2. Les problèmes de pollution et de congestion du trafic ne modifient pas cette conclusion. Si nous désirons rendre plus coûteuse la conduite afin de la décourager, il n'est pas avantageux de le faire en mettant en place des péages, qui participent et à la pollution et à la congestion. Une taxe sur l'essence serait bien mieux. Pareillement, le désir de renforcer la sécurité en limitant la vitesse maximale n'est pas pertinent; un accès gratuit aux routes améliore la vitesse moyenne en évitant arrêts et retards, quelle que soit la limitation de vitesse.
3. On peut voir un logiciel particulier comme une chose nocive, qui ne devrait être accessible à personne, à l'instar de la base de données d'informations personnelles de Lotus (Marketplace), qui a été retirée des ventes suite à la désapprobation du public. La plus grande partie de mon discours ne s'applique pas à ce cas, mais préférer un propriétaire dans la mesure où cela rendrait le programme moins disponible n'est pas très sensé. Le propriétaire ne le rendra pas *complètement* indisponible, comme on pourrait le souhaiter pour un programme considéré comme nocif.

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Benjamin Drieu.

Révision : trad-gnu@april.org

Dernière mise à jour : 2009/01/22 21:28:10

Traductions de cette page

[Česky](#) [cs] [Deutsch](#) [de] [English](#) [en] [Español](#) [es] [Suomi](#) [fi] [Français](#) [fr] [עברית](#) [he]

[Bahasa Indonesia](#) [id] [Nederlands](#) [nl] [Polski](#) [pl] [português do Brasil](#) [pt-br] [Русский](#) [ru] [Српски](#) [sr]

[தமிழ்](#) [ta] [简体中文](#) [zh-cn] [繁體中文](#) [zh-tw]

Droit d'auteur et mondialisation à l'âge des réseaux informatiques

Ce qui suit est la transcription d'un discours donné au MIT lors du Forum des Communications, le jeudi 19 avril 2001, 17h00 - 19h00

DAVID THORBURN, animateur : Notre intervenant d'aujourd'hui, Richard Stallman, est une figure légendaire dans le monde de l'informatique. Ce fut pour moi une expérience instructive de tenter de trouver un interlocuteur qui lui donne la réplique. Un professeur éminent du MIT m'a confié qu'il faut considérer Stallman comme une figure charismatique de parabole biblique — « Imaginez », m'a-t-il dit, « un Moïse ou un Jérémie. Un Jérémie, plutôt. ». Ce à quoi j'ai rétorqué : « Et bien, voilà qui est admirable ».

Voilà qui est merveilleux, et qui confirme mon impression de l'importance de sa contribution au monde. Pourquoi, dans ce cas, avoir une telle réticence à partager la tribune avec lui ? » Réponse : « Tel Jérémie ou Moïse, il me submergerait, tout simplement. Je refuse de débattre avec lui, mais s'il me fallait nommer cinq personnes vivantes dont les actes ont véritablement bénéficié à l'humanité, Richard Stallman serait l'une d'elles ».

RICHARD STALLMAN : Il me faut commencer par expliquer pourquoi j'ai refusé d'autoriser que ce débat soit diffusé en direct sur le web, au cas où le problème n'est pas apparu clairement : le logiciel utilisé pour la diffusion sur le web oblige l'utilisateur à télécharger un autre logiciel pour recevoir l'émission. Ce logiciel n'est pas libre. Il est disponible gratuitement mais uniquement sous forme exécutable, c'est-à-dire sous la forme d'une suite de nombres cabalistiques.

Ce qu'il fait est secret. On ne peut ni l'étudier, ni le modifier, et certainement pas en publier de version modifiée. Et ce sont là des libertés essentielles dans la définition du « logiciel libre ».

C'est pourquoi, si je souhaite me comporter en prosélyte honnête du logiciel libre, je ne peux pas d'un côté donner des conférences, et de l'autre inciter les gens à utiliser des logiciels qui ne sont pas libres. Cela saperait ma propre cause. Si je ne prends pas mes principes au sérieux, comment m'attendre à ce que d'autres le fassent ?

Cependant, je ne suis pas venu parler de logiciel libre. Après avoir travaillé plusieurs années pour le mouvement du logiciel libre et avec les premiers utilisateurs du système d'exploitation GNU, j'ai commencé à recevoir des invitations pour donner des discours, discours où on me posait des questions telles que : « Les idées de liberté pour les utilisateurs de logiciels se généralisent-elles à d'autres domaines et comment ? »

Et bien sûr, on me posait de sottises questions comme « Le matériel informatique devrait-il être libre ? » « Ce microphone devrait-il être libre ? »

Qu'est-ce que cela peut bien vouloir dire ? Devrait-on avoir la liberté de le copier et de le modifier ? En ce qui concerne les modifications, personne ne peut s'opposer à ce que vous modifiiez un microphone que vous auriez acheté. Quant à la copie, personne ne dispose d'un duplicateur de microphone — de telles choses n'existent que dans les histoires de science-fiction. Un jour, nous disposerons peut-être de nanotechniques d'analyse et d'assemblage, et il sera peut-être possible de copier pour de vrai un objet physique. Alors, les libertés attachées aux objets prendront une véritable importance. Nous verrons des fabricants de produits agricoles tenter d'interdire de dupliquer la nourriture, et il s'agira d'un débat politique de première importance — si de telles nanotechniques voient jamais le jour. J'ignore si ce sera le cas : il ne s'agit pour le moment que de pure spéculation.

Mais cette question se pose pour d'autres types d'informations : toute information stockable sur ordinateur peut être copiée et modifiée. Par conséquent, les problématiques éthiques du logiciel libre, les droits qu'a l'utilisateur de copier et de modifier le logiciel, sont transposables à d'autres types d'informations. Soyons clairs : je ne parle pas des informations privées, telles que les informations personnelles, qu'il n'est jamais question de rendre publiques. Je traite des droits qu'il faudrait donner à quiconque reçoit des copies de données publiées, sans intention ni tentative de les garder secrètes.

Pour exposer mes idées sur la question, je vais passer en revue l'histoire de la distribution de

l'information et du copyright. Jadis, les livres étaient écrits à la main à l'aide d'une plume, et tout lettré pouvait recopier un livre presque aussi efficacement que quiconque. Bien sûr, celui qui exerçait cette activité toute la journée y acquérait probablement des compétences particulières, mais la différence n'était pas énorme. Les copies étant exécutées à la pièce, l'industrialisation du procédé n'apportait rien. La réalisation de dix copies exigeait dix fois plus d'efforts que la réalisation d'une seule copie. Rien non plus n'incitait à la centralisation: on pouvait copier un livre n'importe où.

Cette technique n'impliquant pas la similitude des copies, on ne distinguait pas avec une telle insistance que de nos jours, les processus de création originale et de copie. Il existait des intermédiaires parfaitement raisonnables. On savait ce qu'était un auteur, on savait, par exemple, que telle pièce avait été écrite par Sophocle, mais on pouvait imaginer, entre la rédaction et la copie, d'autres opérations utiles. On pouvait par exemple recopier une partie d'un livre, ajouter de nouveaux passages, recopier une autre partie, ajouter d'autres passages, etc. On appelait cela « écrire un commentaire » — c'était fréquent — et ces commentaires étaient prisés.

On pouvait aussi copier un passage d'un livre, écrire d'autres mots, copier un passage d'un autre livre, écrire d'autres mots, etc. — on appelait cela écrire un recueil. Les recueils étaient fort utiles également. Certaines œuvres sont perdues et ne survivent que par le truchement de fragments repris dans d'autres livres qui ont eu plus de succès que l'original. Peut-être seules les parties les plus intéressantes étaient-elles reprises, et ces sélections étaient abondamment recopiées, alors que personne ne prenait la peine de recopier les textes originaux, jugés trop peu intéressants.

Autant que je sache, rien ne ressemblait alors au copyright. Quiconque souhaitait copier un livre, le pouvait. Plus tard, avec l'invention de l'imprimerie, des livres ont commencé à être imprimés. L'imprimerie ne représentait pas uniquement un progrès quantitatif dans la facilité de copier. Elle affectait de façon différente divers types de copies en introduisant une économie d'échelle intrinsèque. C'était un travail long et fastidieux que de préparer les planches de caractères amovibles, comparée à quoi l'opération d'imprimer la même page plusieurs fois était négligeable. Il en résulta une centralisation et une production en série des livres; ce n'est vraisemblablement qu'en quelques endroits qu'un livre donné pouvait être copié.

Il en résulta également que les lecteurs ordinaires ne pouvaient plus copier efficacement des livres. Cette activité était réservée à ceux qui disposaient d'une presse à imprimer: elle était devenue industrielle.

Cette révolution s'est opérée graduellement, et l'imprimerie n'a pas totalement remplacé la copie manuelle lors des premiers siècles de sa mise en service. Des riches comme des pauvres continuaient de recopier ou de faire recopier des livres à la main. Les riches, pour exhiber leur richesse par le truchement d'exemplaires richement enluminés; les pauvres, par manque d'argent pour acheter un livre imprimé et parce qu'ils avaient le temps. Comme on dit, « le temps n'est pas de l'argent si l'on a du temps à revendre ».

Les copies manuelles ont donc survécu quelque temps. Je pense que c'est au XIX^{ème} siècle que les tarifs de l'imprimerie devinrent suffisamment accessibles pour que même les pauvres instruits puissent s'offrir des livres imprimés.

Le copyright fut développé avec l'essor de l'imprimerie, qu'il réglementa au sens industriel du terme. Il ne limitait pas les droits des lecteurs; il limitait les pouvoirs des éditeurs et des auteurs. En Angleterre, le copyright fut initialement une forme de censure. Il fallait obtenir la permission du gouvernement pour publier un livre. Mais l'idée évolua. Au moment de la constitution des États-Unis d'Amérique, on avait une autre idée du but du copyright, et je pense que cette idée fut reprise en Angleterre également.

Pour la constitution des États-Unis d'Amérique, on proposa que les auteurs obtinssent un copyright, c'est-à-dire un monopole sur la copie de leurs livres. Cette proposition fut rejetée, et remplacée par une proposition crucialement différente: dans le but de promouvoir le progrès, le Congrès pourrait, s'il le souhaitait, établir un système de copyright qui créerait ces monopoles. Si l'on en croit la constitution des États-Unis d'Amérique, les monopoles n'existent pas pour le bien de ceux qui les détiennent; ils n'existent que dans le but de promouvoir le progrès scientifique. On attribue un monopole aux auteurs afin de les inciter à servir la population par leur œuvre.

Le but recherché était donc une augmentation du nombre de livres écrits et publiés, mis à la disposition du public. On pensait que cela inciterait les productions littéraire, scientifique et d'autres

domaines encore, et par voie de conséquence, l'instruction et l'éducation de la société. Voilà le but à servir. La création de monopoles privés n'était qu'un moyen pour servir une fin, et cette fin était le bien public.

À l'âge de l'imprimerie, le copyright était peu douloureux, puisque c'était un règlement industriel. Il ne limitait que les activités des éditeurs et des auteurs. Stricto sensu, les pauvres qui copiaient des livres à la main violaient probablement des copyrights. Mais personne ne les a jamais poursuivis, puisqu'on considérait le copyright comme un règlement industriel.

À l'âge de l'imprimerie, le copyright était facile à faire appliquer car il ne s'exerçait que chez les éditeurs — et par nature, un éditeur cherche à se faire connaître. Si l'on cherche à vendre des livres, il faut dire aux gens où venir les acheter. Il n'était pas nécessaire d'aller faire respecter le copyright dans toutes les chaumières.

Enfin, le copyright fut peut-être bénéfique dans un tel contexte. Aux États-Unis d'Amérique, les spécialistes en droit considèrent le copyright comme un échange, un accord entre le public et les auteurs. Le public abandonne certains de ses droits naturels à la réalisation de copies, et reçoit en échange le bénéfice d'un plus grand nombre de livres créés et produits.

Est-ce un accord avantageux ? Si tout un chacun ne peut réaliser de copies efficacement faute de disposer d'une presse à imprimer — et rares étaient les possesseurs de tels engins — il advient que le grand public abandonne une liberté qu'il ne peut exercer, une liberté impraticable. Si l'on a l'occasion d'échanger une potentialité inutile contre quelque chose de valeur, on réalise un marché gagnant. C'est pourquoi le copyright fut probablement un échange avantageux pour la société à cette époque.

Mais les temps et le contexte changent, et cela remet en question notre évaluation éthique du copyright. Les principes éthiques ne changent en rien avec les progrès techniques; ils sont trop fondamentaux pour être influencés par de telles contingences. Mais toute décision concernant un problème donné dépend des conséquences des choix disponibles, et les conséquences d'un choix donné dépendent du contexte. C'est ce qui se produit dans le domaine du droit du copyright, car l'âge de l'imprimerie prend fin et cède peu à peu la place à l'âge des réseaux informatiques.

Les réseaux informatiques et les techniques numériques de traitement de l'information nous ramènent dans une situation semblable aux temps anciens, où quiconque pouvant lire et utiliser l'information pouvait également la copier, et ce, aussi facilement que n'importe qui. Ce sont des copies parfaites, aussi bonnes que les copies qu'un autre réaliserait. La centralisation et les économies d'échelle introduites par l'imprimerie et les techniques apparentées ne s'appliquent donc plus.

Cette évolution du contexte modifie la manière dont le droit du copyright fonctionne. Il ne fonctionne plus comme un règlement industriel, mais comme une restriction draconienne à l'encontre du public. C'était une restriction imposée aux éditeurs pour le bien des auteurs. C'est devenu, dans la pratique, une restriction imposée au public pour le bien des éditeurs. Le copyright n'était pas douloureux et ne soulevait pas de controverses, puisqu'il ne restreignait pas le public. Aujourd'hui, ce n'est plus le cas. La priorité numéro un des éditeurs est de confiner tout possesseur d'ordinateur. Le copyright était facile à faire appliquer en tant que restriction imposée aux éditeurs, car ils avaient pignon sur rue. C'est maintenant devenu une restriction portant sur tout un chacun. Son application implique surveillance, intrusions, et châtiments sévères, moyens dont nous voyons la mise en lois et en œuvre aux États-Unis d'Amérique et dans d'autres pays.

On pouvait défendre la thèse que le copyright était un marché avantageux pour le public, qui abandonnait des libertés qu'il ne pouvait exercer. Il peut désormais les exercer. Que faire si l'on découvre soudainement une utilité à une potentialité jusque là troquée faute de lui trouver un intérêt ? On peut désormais la consommer, l'utiliser. Que faire ? On n'y renonce plus, on en garde des parties. C'est l'impulsion naturelle. C'est pourquoi, à chaque occasion où il peut exprimer sa préférence, c'est ce que le public fait: il conserve une portion de cette liberté, et l'exerce. Napster en est un exemple frappant: le public décide d'exercer sa liberté de copier au lieu de l'abandonner. La tendance naturelle, pour faire correspondre le droit du copyright aux circonstances actuelles, est de réduire le pouvoir des détenteurs de copyright, réduire les restrictions qu'ils imposent au public, et augmenter les libertés que le public conserve.

Voilà qui déplaît aux éditeurs. Ils recherchent exactement l'inverse. Ils veulent développer le pouvoir du copyright jusqu'à contrôler fermement toute utilisation de l'information. Ces pressions ont fait voter des

lois attribuant aux pouvoirs associés au copyright des extensions sans précédent. Certaines des libertés accordées au public à l'âge de l'imprimerie lui sont désormais retirées.

Examinons par exemple le cas des livres électroniques. C'est un thème tellement à la mode qu'il est difficile d'y échapper. J'ai pris l'avion pour le Brésil et le magazine de bord contenait un article annonçant que d'ici 10 ou 20 ans, nous passerions tous aux livres électroniques. Voilà clairement une campagne financée par quelqu'un. Dans quel but ? Je crois que j'ai deviné. Les livres électroniques sont l'occasion de retirer aux lecteurs des livres imprimés certaines des libertés qu'ils ont réussi à conserver — telles que la liberté de prêter un livre à un ami, de l'emprunter dans une bibliothèque publique, d'en vendre un exemplaire à un magasin de livres d'occasion, d'en acheter un exemplaire de manière anonyme, sans laisser de trace dans une quelconque base de données. Et, qui sait, le droit de le lire deux fois.

Voilà des libertés que les éditeurs souhaiteraient nous retirer, mais qui dans le cas des livres imprimés provoquerait une levée de boucliers car ce serait une prise de pouvoir trop voyante. La stratégie indirecte qu'ils ont trouvée est donc la suivante: :tout d'abord, on obtient de la loi qu'elle retire ces libertés aux livres électroniques à une époque où ils n'existent pas encore, ne provoquant ainsi aucune controverse. Il n'existe pas d'antériorité, d'utilisateurs de livres électroniques habitués à ces libertés et prêts à les défendre. Cette première étape fut atteinte avec le « Digital Millennium Copyright Act » en 1998. Ensuite, on introduit les livres électroniques et peu à peu on incite tout le monde à passer des livres imprimés aux livres électroniques. Finalement, le résultat est que les lecteurs ont perdu ces libertés sans qu'à aucun moment, ils ne s'en soient vu priver et aient donc eu l'occasion de se battre pour les conserver.

Pendant ce temps, on observe des remises en question des libertés de disposer d'autres types d'œuvres. Les films sur DVD, par exemple, sont publiés dans un format chiffré et secret — telle était initialement son intention en tout cas — et la seule manière d'obtenir des majors du cinéma la description du format, afin d'être capable de construire un lecteur de DVD, est de signer un contrat s'engageant à brider le lecteur. Ainsi, il est devenu impossible au public d'exercer les droits qui lui sont garantis par la loi. C'est alors que des programmeurs européens astucieux ont deviné le format des DVD et publié un logiciel libre capable de lire les DVD. Cela a permis aux utilisateurs du système GNU/Linux ayant acquis des DVD de regarder les films qu'ils avaient achetés — exigence parfaitement légitime au demeurant. On devrait pouvoir le faire avec des logiciels libres.

Mais les majors n'étaient pas de cet avis et elles ont porté l'affaire en justice. Voyez-vous, elles ont produit de nombreux films où un savant fou se voyait objecter « Mais Docteur, il y a des choses que l'Homme ne doit pas savoir ». Il faut croire qu'elles ont trop regardé leurs propres films, car elles en sont venues à croire qu'il était mauvais pour l'Homme de connaître le format des DVD. Elles ont ainsi obtenu, par décision de justice, la censure complète de tout logiciel capable de lire les DVD. Il est même interdit désormais de tirer un lien vers un site qui propose cette information en toute légalité, hors des États-Unis d'Amérique. Il a été fait appel de cette décision, et je suis fier d'annoncer, bien que mon rôle reste mineur dans cette bataille, que j'ai signé un rapport technique sur le sujet.

Le gouvernement des États-Unis d'Amérique a pris ouvertement part pour la partie adverse. Cela n'est pas surprenant quand on se rappelle la véritable raison de la mise en place du « Digital Millennium Copyright Act » :le système de financement des campagnes électorales, qui n'est rien d'autre qu'une corruption légalisée, où les grandes entreprises achètent les candidats avant même leur élection. Les élus ne connaissant que trop bien leurs maîtres — ils savent qui ils doivent vraiment servir — ils font passer les lois qui renforcent le pouvoir des grandes entreprises.

Nous ne savons pas ce qu'il adviendra de cette bataille en particulier. L'Australie a depuis voté une loi semblable, et l'Europe est sur le point de le faire; l'intention est clairement de n'oublier aucun pays où cette information pourrait être mise à disposition du public. Mais ce sont les États-Unis d'Amérique qui demeurent champion du monde des entraves au grand public à distribuer les informations préalablement publiées.

Cependant, ce n'est pas le premier pays à faire de ce combat sa priorité. L'Union Soviétique y attachait également une grande importance. Les copies et distributions non autorisées y portaient le nom de samizdat et les Soviétiques avaient mis au point un certain nombre de méthodes pour les contrer. D'abord, des gardes surveillaient tous les copieurs et vérifiaient les documents que les gens copiaient, afin d'interdire toute copie illicite. Deuxièmement, de durs châtements étaient réservés aux contrevenants: :on pouvait être envoyé

en Sibérie. Troisièmement, l'incitation à la dénonciation encourageait chacun à moucharder voisins ou collègues auprès de la police de l'information. Quatrièmement, la responsabilité collective: « Toi! Tu surveilles ce groupe ! Si je prends l'un d'entre eux à réaliser des copies interdites, tu vas en prison. Alors surveille-les bien ». Et cinquièmement, la propagande qui, dès l'enfance, visait à convaincre tout le monde que seul un ignoble ennemi du peuple serait capable de se livrer à de telles activités.

Chacune de ces mesures est en place aux États-Unis d'Amérique en ce moment. D'abord, des gardes qui surveillent les copieurs. Dans les boîtes à copies, des gardes vérifient ce que vous copiez. Mais le coût de gardes pour vérifier ce que vous copiez avec votre ordinateur serait prohibitif; le travail humain coûte trop cher. Alors, on utilise des robots. C'est le but du « Digital Millennium Copyright Act ». Ce logiciel infiltre l'ordinateur: c'est le seul portail qui vous permet d'accéder à certaines informations, et c'est lui qui vous empêche de réaliser des copies.

Il est prévu d'inclure ce logiciel sur tous les disques durs, de sorte que certains fichiers de votre propre disque ne vous soient accessibles qu'après l'accord d'un serveur réseau. Il serait illégal de contourner cette protection, ainsi que d'expliquer à d'autres comment la contourner.

Deuxièmement, les durs châtiments. Voici quelques années, il n'était pas illégal de copier des informations et de les donner à vos amis; cela n'avait jamais été illégal aux États-Unis d'Amérique. C'est depuis devenu un délit, de telle sorte qu'il est maintenant possible de passer des années en prison pour avoir partagé avec autrui.

Troisièmement, les informateurs. Vous avez peut-être vu les publicités à la télévision ou dans le métro de Boston: on incite désormais à balancer les collègues à la police de l'information, qui s'appelle officiellement « Software Publishers Association ».

Quatrièmement, la responsabilité collective. Aux États-Unis d'Amérique, ce but fut atteint en contraignant les fournisseurs d'accès à l'Internet et en les rendant responsables de tout ce que leurs clients publient. La seule manière pour eux de ne pas être tenus responsables est d'appliquer une procédure sans recours consistant à déconnecter ou détruire l'information qui fait l'objet d'un contentieux dans les deux semaines qui suivent la plainte. J'ai entendu qu'il y a quelques jours, un site de protestation intelligent mettant en cause la « City Bank » fut ainsi déconnecté. De nos jours, la présomption d'innocence a vécu: on n'a plus le droit de présenter sa défense, et on se fait tout simplement supprimer le site web.

Enfin, la propagande, dès l'enfance. C'est le but du mot « pirate ». Si vous vous souvenez, ce terme était formellement réservé, voici quelques années, aux éditeurs qui ne payaient pas l'auteur. Son sens a depuis complètement changé et ce mot désigne les membres du public qui échappent au contrôle de l'éditeur. On l'utilise pour convaincre les gens que seul un méchant ennemi du peuple peut se livrer à des activités de copie interdite. Il sous-entend que partager avec son voisin est éthiquement équivalent à lancer un abordage. J'espère que vous refusez cet amalgame et si tel est le cas, j'espère que vous refuserez d'utiliser ce mot dans ce sens.

Ainsi, les éditeurs achètent des lois pour étendre et conforter leur pouvoir. De plus, ils allongent la durée du copyright. La constitution des États-Unis d'Amérique prévoit que l'exercice du copyright doit être limité dans le temps, mais les éditeurs souhaitent tout le contraire. Comme il serait toutefois difficile de faire amender la Constitution, ils ont trouvé une manière détournée d'obtenir le même résultat plus facilement. Tous les 20 ans, ils allongent de 20 ans la durée légale du copyright, avec effet rétroactif. C'est ainsi qu'à tout instant, la durée légale du copyright est finie, et tout copyright donné prendra théoriquement fin à un moment. Mais cette expiration ne sera jamais atteinte dans la pratique, puisque les copyrights sont allongés de 20 ans tous les 20 ans; plus aucune œuvre n'entrera donc jamais dans le domaine public. C'est le principe du « copyright perpétuel ».

La loi qui en 1998 a étendu de 20 ans la durée du copyright est connue sous le nom de « Mickey Mouse Copyright Extension Act » (loi d'extension de copyright « Mickey »), puisque l'un de ses plus virulents défenseurs était Disney. Les gens de Disney se sont rendus compte que le copyright portant sur Mickey allait expirer, événement qu'ils ne souhaitent absolument pas, ce copyright étant très lucratif.

J'étais censé vous parler de « copyright et mondialisation ». Si on étudie la mondialisation, on observe la mise en place d'un certain nombre de politiques au nom de l'efficacité économique ou de

prétendus traités de libre échange — traités dont le véritable but est d'augmenter le pouvoir des grandes entreprises au détriment des lois et des politiques. Ce ne sont pas vraiment des traités de libre échange. Ce sont des traités qui mettent en place un transfert de pouvoir : ils retirent aux citoyens le pouvoir de décision — décisions portant sur leurs propres intérêts ou ceux de leur pays — pour l'attribuer à des entreprises qui ne prendront pas en compte les intérêts de ces citoyens.

Pour ces entreprises, la démocratie est un problème, et ces traités visent à le résoudre. Par exemple, l'ALÉNA prévoit des cas autorisant des entreprises à attaquer en justice un gouvernement étranger pour annuler une loi qu'elles estiment interférer avec leurs profits dans ce pays étranger. C'est ainsi que les entreprises étrangères ont plus de pouvoir que les citoyens du pays.

On observe des tentatives d'étendre ce système au-delà des limites prévues par l'ALÉNA. C'est par exemple l'un des buts de la prétendue zone de libre échange des Amériques que d'étendre ce principe à tous les pays d'Amérique du Sud et des Caraïbes, et les accords multilatéraux sur les investissements avaient pour but d'étendre ce principe au monde entier.

Les traités signés dans les années 1990 ont commencé à imposer le copyright au monde entier, d'une manière plus forte et plus restrictive qu'auparavant. Ce ne sont pas des traités de libre échange. Ce sont des traités de commerce contrôlés par les entreprises et dont le but est d'attribuer aux entreprises le contrôle du commerce mondial, afin d'éliminer tout libre échange.

Au XIX^{ème} siècle, les États-Unis d'Amérique, alors un tout jeune pays, ne reconnaissaient pas les copyrights étrangers. Ce fut une décision prise après mûre réflexion, et c'était une bonne décision. Les débats ont conclu que la reconnaissance par les États-Unis d'Amérique des copyrights étrangers ne serait pas avantageuse, et qu'elle aurait pour effet de vider les caisses sans rien apporter en échange.

On pourrait de nos jours appliquer ce raisonnement aux pays en voie de développement, s'ils n'étaient pas contraints par la puissance de ces mêmes États-Unis d'Amérique d'aller contre leurs intérêts. Dans un tel contexte, c'est en fait une erreur de parler d'intérêts des pays. Je suis sûr que vous avez pour la plupart entendu parler du raisonnement vicié qui tente de mesurer l'intérêt public en ajoutant toutes les richesses individuelles. Si les Américains actifs perdaient un milliard de dollars et que Bill Gates gagnait deux milliards de dollars, cela profiterait-il à la population des États-Unis d'Amérique, de manière générale ? Cela profiterait-il aux États-Unis d'Amérique ? Si l'on ne prend que le total en compte, cela semble une bonne affaire. Cet exemple montre cependant que le seul examen du total ne suffit pas à porter un jugement parce que Bill Gates n'a pas vraiment besoin de deux milliards de dollars supplémentaires, alors que la perte d'un milliard de dollars serait douloureuse pour une population qui ne jouit pas d'un patrimoine comparable. Et bien, dans toute discussion sur ces traités de commerce, quand on entend parler des intérêts de tel ou tel pays, c'est le même calcul qui est fait : en réalité c'est du total des revenus particuliers au sein de chaque pays qu'il est question. On regroupe les riches et les pauvres dans le même calcul. Il s'agit donc d'un prétexte pour appliquer ce même raisonnement vicié afin de vous faire oublier la distribution des richesses au sein du pays et d'éviter que vous vous demandiez si le traité creusera les fossés et rendra cette distribution encore plus inégale — comme ce fut le cas aux États-Unis d'Amérique.

Ce n'est pas l'intérêt des États-Unis d'Amérique qu'on sert en faisant respecter le copyright partout dans le monde. C'est l'intérêt de certains propriétaires et patrons, dont beaucoup se trouvent aux États-Unis d'Amérique. En aucun cas, cela ne sert l'intérêt public.

Que faudrait-il faire, dans ce cas ? Si l'on croit à la justification et au but du copyright tel que précisés dans la constitution des États-Unis d'Amérique — la « promotion du progrès » —, quelles seraient les politiques intelligentes à mettre en place à l'âge des réseaux informatiques ? Clairement, au lieu d'augmenter les pouvoirs de copyright, il nous faut les limiter ou les diminuer, afin d'offrir au grand public un domaine de liberté où les gens puissent faire usage des bénéfices des techniques numériques et des réseaux informatiques. Jusqu'où aller ? C'est une question intéressante, car je ne crois pas qu'il faille abolir complètement le copyright. L'idée de troquer certaines libertés en échange de progrès peut encore être avantageuse à certains niveaux, même si l'exercice traditionnel du copyright abandonne trop de libertés. Si l'on veut réfléchir intelligemment à cette question, il nous faut commencer par reconnaître qu'il n'y a aucune raison de vouloir tout uniformiser. Il n'y a aucune raison de vouloir aboutir aux mêmes accords et aux mêmes compromis pour tous les types d'œuvres.

En réalité, ce n'est déjà pas le cas, car de nombreuses exceptions couvrent le domaine des œuvres musicales. Les lois de copyright traitent la musique très différemment. Mais c'est intelligemment que les éditeurs insistent — arbitrairement — sur l'uniformisation : ils choisissent un cas précis et arguent que dans ce cas particulier, il serait avantageux d'appliquer telle disposition en matière de copyright. Ils expliquent ensuite que par souci d'uniformité, il faut que tout soit régulé par les mêmes dispositions de copyright. Bien sûr, ils choisissent le cas de figure qui leur est le plus favorable, même s'il est rare et pas vraiment très important dans la pratique.

Peut-être bien que ces dispositions sont les meilleures dans ce cas particulier. Il n'est nul besoin de payer le même prix pour tous les achats que l'on réalise : mille dollars pour une voiture neuve serait sans doute une excellente affaire, alors que le même prix pour une brique de lait serait une affaire exécrable. On n'accepte pas de payer le même prix pour tout ce que l'on achète par ailleurs — pourquoi devrait-on l'accepter ici ?

Il nous faut donc distinguer différents types de travaux, et je souhaite vous faire une proposition de classification.

Concernant les recettes, les programmes d'ordinateurs, les manuels et livres de cours, et les œuvres de référence telles que les dictionnaires et les encyclopédies : pour tous ces travaux fonctionnels, je crois que les problèmes qui se posent sont les mêmes que dans le cas du logiciel, et que par conséquent les mêmes conclusions s'appliquent. Les gens devraient même avoir la liberté d'en publier une version modifiée, car il est très utile de modifier les œuvres fonctionnelles. Tout le monde n'a pas les mêmes besoins : si j'ai écrit tel texte pour répondre à une nécessité ou à un souhait personnel, il ne répondra pas forcément à vos besoins, légèrement différents. Vous pourrez donc souhaiter modifier mon travail pour vous le rendre plus adéquat. D'autres pourront alors bénéficier de votre version modifiée, car leurs habitudes seront similaires aux vôtres. Cela fait des siècles que tous les cuisiniers savent cela. Il est normal de donner ses recettes à d'autres, et il est normal d'adapter une recette. Si vous changez une recette et que les amis à qui vous servez le fruit de votre labeur aiment le résultat, ils vous demanderont probablement la recette. Alors peut-être leur donnerez-vous des copies de votre version. C'est exactement ce que nous fîmes, bien plus tard, dans la communauté du logiciel libre.

Voilà pour la première catégorie de travaux. La deuxième catégorie concerne les œuvres visant à rendre compte de la pensée et des idées de certains : le but de ces œuvres est de parler de leurs auteurs. Cette catégorie embrasse, entre autres, les mémoires, les essais d'opinion, les articles scientifiques, les petites annonces, les catalogues de biens à vendre. Toutes ces œuvres ont pour but de communiquer ce qu'une personne pense, a vu, ou croit. Les modifier serait trahir les auteurs; c'est pourquoi modifier ces œuvres n'est pas une activité utile socialement. Par conséquent, seule la copie exacte, ou verbatim, de ces œuvres doit être autorisée.

Il est alors légitime de se demander s'il faut autoriser une activité commerciale autour de telles copies verbatim, ou si la diffusion à but non lucratif suffit. On peut donc distinguer deux activités différentes, afin de les étudier séparément pour chercher des réponses à ces questions — le droit de faire des copies verbatim dans un but non lucratif, et le droit de faire des copies verbatim dans un contexte commercial. C'est un bon compromis que d'appliquer le copyright aux copies verbatim dans un contexte commercial, et d'autoriser sans autre contrainte les copies exactes exécutées et diffusées dans un but non lucratif. De cette manière, les copyright sur les copies verbatim, ainsi que sur toutes les versions modifiées — seul l'auteur pourrait approuver une version modifiée — fournirait une source de revenus de la même manière qu'il finance actuellement la création de ces œuvres, et dans les mêmes limites.

Autoriser les copies exactes à but non lucratif signifie que le copyright ne doit plus s'immiscer dans la vie privée de chacun. Il redevient un règlement industriel, d'application facile et indolore, ne nécessitant plus de châtiments draconiens ni d'informateurs, sous le seul prétexte de le faire respecter. On obtient alors la majeure partie des bénéfices — et on évite la pire des horreurs — du système actuel.

La troisième catégorie renferme les travaux esthétiques ou de loisirs, où le plus important est la sensation ressentie en regardant l'œuvre. La question de la modification est ici très difficile : d'un côté on trouve l'idée que ces œuvres reflètent la vision d'un artiste, et que toute modification viole et brise cette vision. D'autre part, il ne faut pas oublier les processus populaires, où des modifications apportées en chaîne

par de nombreuses personnes produisent parfois un résultat extrêmement riche. Même si ce sont des artistes qui créent, les emprunts à des travaux précédents sont parfois fort utiles. Certaines pièces de Shakespeare se sont inspirées d'œuvres antérieures. Si les lois de copyright aujourd'hui en vigueur avaient alors été appliquées, de telles pièces auraient été illégales. C'est donc une question difficile que de décider comment réguler la publication de versions modifiées d'œuvres esthétiques ou artistiques, et il nous faudra peut-être subdiviser cette troisième catégorie pour résoudre ce problème. Par exemple, on peut imaginer de traiter les scénarios de jeux vidéo d'une manière ; peut-être est-il souhaitable que quiconque puisse en publier des versions modifiées. Mais peut-être qu'un roman doit être traité différemment ; peut-être que toute publication commerciale devrait faire l'objet d'un accord avec l'auteur.

Si la publication commerciale de ces œuvres esthétiques est couverte par le copyright, la plus grande part des revenus actuellement perçus servira encore à financer les auteurs et les musiciens, dans la mesure restreinte où le présent système les soutient, car il est très mauvais de ce point de vue. Cela pourrait donc être un compromis raisonnable, de la même manière que celui que nous avons imaginé pour les œuvres de la deuxième catégorie, les œuvres qui représentent certaines personnes.

Si nous nous projetons dans le futur, à l'âge où les réseaux auront véritablement déployé leurs ailes, et où nous aurons dépassé l'état transitoire actuel, on peut imaginer une autre manière de rémunérer les auteurs pour leurs travaux. Imaginez un système de micro-paiements numériques qui vous permette d'être rémunéré pour votre travail. Imaginez un système de micro-paiements numériques qui permette d'envoyer de l'argent par l'Internet : il existe plusieurs manières d'obtenir un tel résultat en utilisant, par exemple, des procédés cryptographiques de chiffrement. Imaginez encore que les copies exactes de toutes ces œuvres esthétiques soient autorisées, mais que ces œuvres soit écrites de telle manière que toute représentation ou lecture ferait apparaître sur le côté de l'écran une boîte de dialogue disant « Cliquez ici pour envoyer un dollar à l'auteur », ou au musicien, ou au titre que la personne concernée se sera choisi. Cette boîte se trouverait là, sur le côté, sans interférer ni déranger, mais elle vous rappellerait que c'est une bonne idée d'encourager et de financer les auteurs et les musiciens.

Si vous aimez l'œuvre que vous lisez ou écoutez, alors vous vous direz tôt ou tard « Et pourquoi n'enverrais-je pas un dollar aux auteurs ? Un dollar, qu'est-ce pour moi ? Je ne verrai pas la différence ». Et les gens commenceront à envoyer des dollars. Le point positif est que cela fait des copies l'allié des auteurs et des musiciens : toute copie envoyée par courrier électronique à un ami pourra inciter cet ami, lui aussi, à envoyer un dollar aux auteurs. Si vous aimez vraiment l'œuvre, vous pourrez envoyer plusieurs dollars — et ce dollar représente plus que ce que les auteurs perçoivent de nos jours si on achète un livre ou un CD, car seule une infime portion du prix de vente leur parvient. Ces mêmes éditeurs, qui exigent un pouvoir et un contrôle total sur le public au nom des auteurs et des musiciens, paient ces derniers au lance-pierres.

Je vous recommande l'article de Courtney Love, publié dans le magazine « Salon », et traitant des pirates qui envisagent d'utiliser le travail des musiciens sans rémunérer ces derniers. Ces pirates sont les majors du disque, qui reversent en moyenne 4% des ventes aux musiciens. Bien sûr, les musiciens à succès touchent davantage. Ils reçoivent plus de 4% du faramineux chiffre d'affaires qu'ils suscitent — ce qui signifie que la grande majorité des musiciens sous contrat avec les majors du disque perçoivent moins de 4% de leurs ventes.

Voici comment le système fonctionne : la major investit dans une campagne publicitaire et considère ces frais comme une avance aux musiciens, alors même que les musiciens n'en perçoivent rien. Ainsi, ce n'est qu'en théorie qu'une certaine fraction du prix de vente de tout CD revient aux musiciens. En réalité, cette fraction rembourse les frais de la campagne publicitaire, et seuls les musiciens qui vendent beaucoup de disques commencent à toucher des royalties au-delà d'un certain seuil de ventes.

Bien sûr, les musiciens ne signent de tels contrats que dans l'espoir de faire partie des rares chanceux qui auront du succès. C'est en quelque sorte une loterie qu'on leur propose, une tentation. Même s'ils sont bons musicalement, ils peuvent être moins bons en perspicacité et en logique, et ne pas remarquer ce piège. C'est ainsi qu'ils en viennent à signer — et à n'en retirer qu'une campagne publicitaire. Bien, mais pourquoi ne pas leur offrir un autre type de campagne publicitaire, sans passer par un système restrictif et contraignant, un système industriel qui nous bassine de soupe facile à vendre. Pourquoi les musiciens ne s'allieraient-ils pas plutôt la tendance naturelle des auditeurs à partager les morceaux qu'ils aiment ? Avec l'apparition d'une boîte de dialogue incitant à envoyer un dollar aux musiciens, les réseaux informatiques deviendraient leur

vecteur de publicité — publicité qui pour l'instant est le seul bénéfice qu'ils retirent de leurs contrats avec les majors du disque.

Il nous faut reconnaître que le système actuel du copyright ne finance les musiciens que de manière exécration — aussi exécration que la manière dont le commerce international élève le niveau de vie aux Philippines ou en Chine. Voyez ces zones franches où tout le monde travaille dans des ateliers de confection, et où tout y est produit. La mondialisation est un moyen très inefficace d'améliorer le niveau de vie des populations des pays pauvres. Soit un Américain payé 20 dollars de l'heure pour un travail; si on donne ce travail à un Mexicain qui l'effectue, disons, pour 6 dollars par jour, le résultat est qu'on a pris beaucoup d'argent au travailleur américain, pour n'en reverser qu'une infime portion (quelques pourcents) au travailleur mexicain — l'essentiel du bénéfice est pour l'entreprise. Si votre but est d'élever le niveau de vie au Mexique, voilà une bienmauvaise manière de procéder.

Il est intéressant d'observer le même phénomène et la même idée générale dans l'industrie du copyright. Au nom de ces travailleurs qui méritent sans aucun doute de recevoir quelque chose, on propose des mesures qui ne leur reversent qu'une infime portion des bénéfices et qui ont pour effet principal d'étayer le pouvoir qu'exercent les grandes sociétés sur nos vies.

Quand on essaie de remplacer un très bon système, il faut travailler dur pour proposer une solution meilleure encore. Quand on sait que le système actuel est bancal, il n'est pas si difficile que cela de trouver une meilleure solution; de nos jours, la barre n'est pas placée très haut. Il faut toujours garder cela en tête quand on réfléchit aux politiques à appliquer en matière de copyright.

Je pense avoir dit tout ce que j'avais à dire. J'aimerais mentionner que demain (le 20 avril 2001) est, au Canada, le jour de l'opération « appeler son employeur en prétendant être malade ». C'est en effet le début d'un sommet qui doit finaliser les négociations de la zone de libre-échange aux Amériques pour tenter d'étendre le pouvoir des grandes entreprises à d'autres pays encore, et une grande manifestation de protestation est prévue au Québec. On a pu observer des méthodes extrêmes pour écraser cette protestation : de nombreux Américains sont bloqués à la frontière alors qu'ils sont censés pouvoir à tout instant entrer librement au Canada. On a construit, sur un prétexte complètement bidon, un mur-forteresse autour du centre-ville de Québec pour en exclure les protestataires. On a utilisé de nombreuses autres sales méthodes pour contrer la contestation publique de ces traités. S'il nous restait une once de démocratie après le transfert de pouvoirs, ôtés à des gouverneurs démocratiquement élus, et attribués à des entreprises et à des multinationales nullement élues, peut-être bien que maintenant qu'on supprime et opprime les manifestations de protestation, cette infime portion de démocratie qui nous restait ne survivra pas.

J'ai consacré 17 ans de ma vie au logiciel libre et à des causes proches. Je ne l'ai pas fait parce que je pensais que c'était la cause politique la plus importante, mais parce que c'était le domaine où je voyais que je pouvais utiliser mon talent pour faire beaucoup de bien. Mais les causes politiques ont évolué, et de nos jours la cause politique majeure est de résister à la tendance de donner le pouvoir aux entreprises, au détriment des peuples et des gouvernements. Je vois le logiciel libre et les questions apparentées concernant d'autres types d'informations, dont j'ai parlé aujourd'hui, comme un volet de cette cause fondamentale. C'est donc indirectement que j'ai travaillé pour cette cause. J'espère que ma contribution à cet effort n'est pas vaine.

RÉPONSE :

THORBURN : Nous allons accepter les questions et les commentaires du public dans un instant. Je souhaite d'abord apporter une brève réponse générale. Il me semble que les réflexions de Stallman comportent deux éléments clefs, particulièrement forts et importants. Le premier est de reconnaître que les anciennes hypothèses et les vieux usages en matière de copyright ne sont plus appropriés, et sont remis en cause ou sapés par l'avènement de l'ordinateur et des réseaux informatiques. C'est peut-être évident, mais c'est essentiel.

Le deuxième est de comprendre que l'ère numérique nous force à réfléchir à la manière dont on distingue et dont on évalue les différentes formes de travaux intellectuels et créatifs. Stallman a certainement raison de signaler que certaines entreprises intellectuelles justifient plus que d'autres une protection accrue par le copyright. Tenter d'identifier de manière systématique ces différents types ou niveaux de protection par le copyright me semble une manière valable de traiter les problèmes posés aux travaux intellectuels par la venue de l'ordinateur.

Mais je pense détecter un thème supplémentaire, sous-jacent dans le discours de Stallman, un thème pas vraiment lié aux ordinateurs, mais plus largement aux questions d'autorité démocratique et au pouvoir que les gouvernements et les grandes entreprises exercent de façon croissante sur nos vies. Ce côté populiste et anti-mondialisation du discours de Stallman est stimulant mais aussi réducteur, voire simpliste. Et il est peut-être par trop idéaliste. Par exemple, comment un romancier, un poète, un parolier, un musicien, ou l'auteur d'un manuel universitaire survivraient-ils dans ce meilleur des mondes, où l'on encourage les gens à rémunérer les auteurs, sans les y obliger. En d'autres mots, il me semble que le fossé séparant ce qui se pratique actuellement et les possibilités visionnaires sur lesquelles Stallman spéculé est toujours immense.

Je conclurai donc en demandant à Stallman de développer certains aspects de son discours, et en particulier s'il a des idées supplémentaires sur la manière dont ceux que nous appellerons « les créateurs traditionnels » seraient protégés dans le système de copyright qu'il propose.

STALLMAN : Tout d'abord, je dois préciser un point : il ne faut pas utiliser le terme « protection » quand on se réfère à l'action du copyright. Le copyright restreint les gens. Le mot « protection » est un terme de propagande utilisé par les entreprises propriétaires de copyrights. « Protection » signifie empêcher une forme de destruction de quelque chose. Je ne pense pas qu'une chanson soit détruite si un plus grand nombre de copies de cette chanson sont diffusées plus souvent, auprès d'un plus grand nombre d'auditeurs. Je ne pense pas non plus qu'un roman soit détruit si un plus grand nombre de lecteurs en lisent des copies. Aussi n'utiliserai-je pas ce mot. Je pense qu'il incite à s'identifier avec le mauvais bord.

De plus, c'est une très mauvaise idée de réfléchir à la propriété intellectuelle, et ce, pour deux raisons : d'abord, cela présuppose une réponse à la question la plus fondamentale de ce domaine, c'est-à-dire : « Comment ces choses doivent-elles être traitées, doivent-elles être traitées en tant qu'une forme de propriété ? ». Utiliser le terme « propriété intellectuelle » pour décrire ce domaine revient à présupposer qu'on répond affirmativement à cette question, que c'est la bonne manière d'envisager les choses, et qu'aucune autre n'est acceptable.

Ensuite, cela incite à généraliser abusivement. « Propriété intellectuelle » est un terme générique pour plusieurs systèmes juridiques différents, d'origines indépendantes, tels que le copyright, les brevets, les marques déposées, les secrets industriels, et d'autres choses encore. Ce sont là des domaines presque totalement différents : ils n'ont rien en commun. Mais ceux qui entendent le terme « propriété intellectuelle » sont incités à penser incorrectement, et ils imaginent que c'est un principe générique de la propriété intellectuelle qu'on applique à des cas particuliers, aussi supposent-ils que ces divers domaines juridiques sont similaires. Cela a pour conséquences, non seulement de rendre plus confuses les réflexions sur les bonnes manières de procéder, mais aussi d'empêcher de bien comprendre la loi car on suppose que les lois portant sur les copyrights, sur les brevets et sur les marques déposées sont semblables quand en réalité elles diffèrent totalement.

C'est pourquoi, si vous souhaitez encourager une réflexion prudente et une compréhension claire de la loi, il vous faut éviter le terme « propriété intellectuelle ». Parlez de copyrights, parlez de brevets, parlez de marques déposées ou parlez de quoi que ce soit d'autre dont vous voulez parler. Mais ne parlez pas de « propriété intellectuelle ». Toute opinion portant sur la « propriété intellectuelle » est presque forcément erronée. Je n'ai aucune opinion sur la « propriété intellectuelle ». J'ai des opinions sur les copyrights, sur les brevets, et sur les marques déposées, mais ce sont des opinions différentes. Je me suis forgé ces opinions à travers des cheminements de pensée différents, car ce sont des systèmes juridiques complètement différents.

Je me suis permis cette digression parce que je pense que c'est quelque chose d'extrêmement important.

Je vais maintenant répondre à votre question. Bien sûr, nous ne voyons pas bien, actuellement, de quelle manière cela fonctionnerait, si même cela pourrait fonctionner, de demander aux gens de rétribuer volontairement les auteurs et les musiciens qu'ils apprécient. Il est évident toutefois que ce système fonctionnera d'autant mieux que de nombreuses personnes feront partie du réseau — et vous savez que ce nombre décuplera d'ici quelques années. Si nous essayions ce système aujourd'hui, il échouerait peut-être, et cela ne prouverait rien, car il pourrait fort bien fonctionner avec dix fois plus de participants.

D'autre part, nous ne disposons pas encore de ce système de micro-paiement numérique; aussi ne pouvons-nous pas vraiment mettre cela en pratique de nos jours. Tout au plus pouvons-nous essayer des

schémas semblables. On peut s'inscrire à des services où on peut envoyer de l'argent à quelqu'un — des services tels que « Pay Pal ». Mais avant d'être capable d'envoyer de l'argent à quelqu'un à travers « Pay Pal », il faut se farcir tout un galimatias, communiquer des informations personnelles, et le service conserve des traces des paiements effectués. Comment être sûr qu'ils n'abuseront pas de ces informations ?

Même si l'envoi du dollar ne vous décourage pas, la complexité de la procédure le fera peut-être. L'idée générale est qu'il doit être simple comme bonjour de payer quand on en ressent l'envie, de telle sorte que seul le montant transmis pourrait décourager un éventuel mécène, et rien d'autre. Et si ce montant est suffisamment modeste, en quoi vous découragerait-il ? Nous savons par ailleurs les proportions que l'adulation des idoles peut atteindre et nous savons que certains groupes ont encouragé leurs fans à copier et rediffuser leur musique — groupes qui furent, ou sont encore, fort populaires, comme « Grateful Dead ». Ces groupes n'ont eu aucun problème à vivre de leur musique sous prétexte d'encourager leurs fans à en faire des enregistrements ni à échanger ces enregistrements. Ils n'ont même rien perdu en termes de ventes.

Peu à peu, nous quittons l'âge de l'imprimerie et entrons dans l'âge des réseaux informatiques, mais ce changement est graduel. Nombreux sont ceux qui continuent à acheter des disques, et cela continuera probablement de nombreuses années encore, voire toujours. Tant que cela continuera, disposer de copyrights qui s'appliqueront sur les ventes de disques financera les musiciens à peu près aussi bien que de nos jours. Bien sûr, cela n'est pas très efficace — mais au moins, cela ne pourra pas empirer.

DISCUSSION :

QUESTION : [Un commentaire et une question sur le téléchargement, en mentionnant la tentative de Stephen King de vendre l'un de ses romans épisode après épisode, sur le web.]

STALLMAN : Oui, c'est intéressant de savoir ce qu'il a fait et ce qui s'est passé. Quand j'ai entendu parler de ce projet, j'ai été enthousiaste. Je pensais que peut-être Stephen King faisait quelques pas en direction d'un monde qui ne reposait pas sur le contrôle du public dans une poigne de fer. Ensuite j'ai vu qu'il avait en fait écrit pour demander aux gens de payer. Il publiait en réalité un roman par épisodes, en disant : « Si je reçois assez d'argent, je publierai l'épisode suivant ». Mais c'était moins une requête qu'une intimidation. Il disaient substance: « Si vous ne payez pas, vous êtes mauvais. Et si vous êtes trop nombreux à être mauvais, alors je cesserai d'écrire ce roman, tout simplement ».

Ce n'est clairement pas la bonne manière d'inciter le public à vous envoyer de l'argent — il faut faire en sorte d'être aimé, et non craint.

LE MÊME : Pour entrer dans les détails, il fallait qu'un certain pourcentage — j'ai oublié le pourcentage exact, environ 90% — de personnes envoient une certaine somme d'argent — disons, un ou deux dollars, ou une somme de cet ordre. Il fallait taper son nom et son adresse électronique pour télécharger l'épisode et si le pourcentage souhaité n'était pas atteint après le premier chapitre, l'auteur disait qu'il ne publierait pas le chapitre suivant. C'était très contraignant et agressif vis-à-vis de ceux qui téléchargeaient les chapitres.

QUESTION : Un monde sans copyright, où on incite à faire des donations volontaires, n'est-il pas propice aux abus et aux plagiats ?

STALLMAN : Non. Ce n'est pas ce que j'ai proposé. Souvenez-vous, je propose que le copyright couvre les distributions commerciales, et que seules les copies exactes puissent être diffusées de manière non lucrative. Ainsi, quiconque modifierait l'œuvre pour y inclure un pointeur vers son propre siteweb — au lieu d'un pointeur vers le site web de l'auteur original — violerait le copyright et pourrait être poursuivi pour cela, exactement comme c'est le cas aujourd'hui.

QUESTION : Je vois. Dans le monde que vous imaginez, le copyright existe donc toujours ?

STALLMAN : Oui. Comme je l'ai expliqué, pour cette catégorie d'œuvres. Je ne dit pas que tout devrait être permis. Je propose de réduire les pouvoirs conférés aux copyrights, et non pas de les abolir.

THORBURN : Une question à laquelle j'ai pensé pendant que vous parliez, Richard, et qui me revient alors que vous répondez à cette question, est de savoir pourquoi vous n'étudiez pas la manière dont l'ordinateur, en tant que tel, élimine complètement les intermédiaires — de la manière que Stephen King a

refusé de suivre — et peut établir une relation personnelle.

STALLMAN : C'est le cas, en fait, et cette donation volontaire en est une.

THORBURN : Vous pensez que cela ne doit pas impliquer de passer par un éditeur ?

STALLMAN : Absolument. Je pense qu'il ne le faudra pas, voyez-vous, car les éditeurs exploitent les auteurs d'une manière éhontée. Quand on interroge les représentants des éditeurs à ce sujet, ils répondent: « Bien sûr, si un auteur ou si un groupe ne souhaite pas passer par nous, il ne faut pas les y obliger légalement. » Mais en réalité, ils font tout ce qui est en leur pouvoir pour rendre cela impossible. Ils proposent par exemple des formats de médias restreignant la copie — pour publier dans ces formats, il faut passer par les majors car elles refusent de dire aux autres comment procéder. Les éditeurs rêvent donc d'un monde où tous les lecteurs liront ces formats, et où il faudra passer par les éditeurs pour pouvoir faire jouer son œuvre sur ces lecteurs. En fait, bien qu'aucune loi n'interdise à un écrivain ou à un musicien de publier à compte d'auteur, cela ne sera pas possible. N'oublions pas non plus le miroir aux alouettes. Les éditeurs disent mielleusement: « Nous ferons votre promotion et peut-être deviendrez-vous aussi riches et célèbres que les Beatles ». Choisissez n'importe quel groupe à succès, et bien sûr, seule une fraction très restreinte des musiciens atteindra un tel niveau de célébrité. Mais ils peuvent être incités à signer des contrats qui les piégeront à vie.

Les éditeurs respectent fort peu leurs propres contrats avec les auteurs. Par exemple, les contrats d'édition signalent généralement que si un livre est épuisé, les droits futurs repassent à l'auteur — et les éditeurs n'ont presque jamais scrupuleusement respecté cette clause. Souvent, ils y ont été obligés. Et que font-ils maintenant ? Ils commencent à invoquer la publication sur des formats électroniques pour arguer que le livre n'est jamais épuisé ; ainsi ils ne doivent jamais abandonner les droits à l'auteur. Leur idée est d'inciter les auteurs méconnus à signer, suite à quoi ils n'ont plus aucun pouvoir ; seul leur éditeur en a.

QUESTION : Serait-il bon de disposer de licences libres pour divers types de travaux, pour protéger la liberté de tout utilisateur de copier ces œuvres de toute manière appropriée à leur catégorie ?

STALLMAN : Des gens y travaillent. Mais les œuvres non fonctionnelles ne sont pas interchangeables. Prenons par exemple une œuvre fonctionnelle, tel qu'un traitement de textes. Bien. Si quelqu'un réalise un traitement de textes libre, on peut l'utiliser; on n'a plus besoin alors de traitement de textes non libres. Mais je n'irai pas jusqu'à dire qu'une chanson libre remplace toutes les chansons non libres, ou qu'un roman libre remplace tous les romans non libres. Ces catégories d'œuvres sont différentes. C'est pourquoi je pense qu'il nous faut reconnaître que ces lois ne méritent pas d'être respectées. Ce n'est pas mal de partager avec son voisin, et si quelqu'un essaie de vous dire qu'il ne faut pas partager avec son voisin, ne l'écoutez pas.

QUESTION : Concernant les œuvres fonctionnelles, comment, dans votre pensée, équilibrez-vous le besoin d'abolir le copyright et le besoin d'une incitation économique au développement de ces œuvres fonctionnelles ?

STALLMAN : Nous pouvons commencer par remarquer que cette incitation économique est bien moins nécessaire que tout ce qu'on a supposé. Prenez le mouvement du logiciel libre: nous comptons plus de cent mille volontaires qui développent du logiciel libre sur leur temps libre. On peut également observer qu'il existe d'autres manières de gagner de l'argent, sans pour cela reposer sur l'interdiction de copier et de modifier ces œuvres. C'est la leçon intéressante du mouvement du logiciel libre. Non content de fournir une manière d'utiliser un ordinateur tout en conservant la liberté de partager et de coopérer avec autrui, il nous montre aussi que cette hypothèse négative — selon laquelle nul ne ferait jamais de telles choses à moins de disposer de pouvoirs spéciaux pour obliger leurs futurs utilisateurs à les rémunérer — est fausse. Nombreux sont ceux qui feront de telles choses. Considérons, disons, la rédaction de monographies, qui servent de manuels dans de nombreux domaines scientifiques exceptés les plus élémentaires. Leurs auteurs ne gagnent pas d'argent grâce à cette activité. Nous disposons désormais d'un projet d'encyclopédie libre qui est en réalité un projet d'encyclopédie libre et commerciale, et ce projet progresse. Nous avons un projet pour une encyclopédie GNU mais nous l'avons fusionné avec ce projet commercial quand ses membres ont adopté notre licence. En janvier, ils ont adopté la licence de documentation libre de GNU pour tous les articles de leur encyclopédie. Nous avons alors décidé de joindre nos forces et d'exhorter à contribuer à leur projet. Ce

projet s'appelle « GNUPEDIA », et vous trouverez des liens qui vous y mèneront si vous vous rendez à l'URL <http://www.gnu.org/encyclopedia>. Nous avons donc étendu la communauté de développement d'une base libre de connaissances utiles, du logiciel à l'encyclopédie. Je suis confiant et je pense que dans tous ces domaines de travaux fonctionnels, nul n'est besoin d'une incitation économique si forte qu'elle limite et contraint l'utilisation de ces travaux.

THORBURN : Et concernant les deux autres catégories ?

STALLMAN : Quant aux deux autres catégories d'œuvres, je ne sais pas. Je ne sais pas si un jour on écrira des romans sans se soucier de l'argent qu'ils peuvent rapporter. Dans une société ne souffrant plus de disette ni de rareté, je pense que cela serait le cas. Peut-être que pour atteindre cet état de société, il nous faut nous débarrasser du contrôle qu'exercent les grandes entreprises sur notre économie et sur nos lois. Cela revient en quelque sorte au problème de la poule et de l'œuf : par quoi commencer ? Comment obtenir un monde où l'argent n'est plus un besoin vital sans ôter le contrôle des entreprises ? Et comment éliminer ce dernier sans... — Quoi qu'il en soit, je ne sais pas, et c'est pourquoi j'essaie d'abord de proposer un système de copyright qui est un compromis, et qu'ensuite j'imagine comment un système de rémunération volontaire reposant sur un tel système pourrait fournir une source de revenus aux auteurs de ces œuvres.

QUESTION : Comment espérez-vous, dans la réalité, mettre en œuvre un tel système dans un monde étouffé par l'influence que les entreprises exercent sur les hommes politiques américains à cause du système de financement de leur campagne électorale ?

STALLMAN : Je suis à court. J'aimerais savoir. C'est un problème terriblement difficile. Si je savais comment le résoudre, je m'y attacherais, et rien au monde ne pourrait me rendre plus fier.

QUESTION : Comment combattez-vous le contrôle des entreprises ? Quand on calcule les sommes d'argent injectées par les entreprises dans leur lobbying et leurs actions en justice, cela donne le vertige. Je pense que le procès DECSS que vous mentionnez coûte à la défense un million et demi de dollars. Dieu sait ce qu'il coûte du côté des entreprises et de l'accusation. Avez-vous une idée sur la manière de traiter des sommes d'argent aussi gigantesques ?

STALLMAN : J'ai une suggestion. Si je suggérais un boycott complet des films, je pense que ne serais pas écouté. Les gens trouveraient la mesure trop radicale. Je souhaite donc faire une suggestion légèrement différente, qui revient presque au même finalement, et qui est de ne pas aller au cinéma à moins d'avoir une bonne raison de penser que le film est bon. Dans la pratique, cela aura presque le même résultat qu'un boycott inconditionnel des films de Hollywood. Dans la pratique, c'est presque la même chose, mais en intention, c'est très différent. J'ai remarqué que de nombreuses personnes vont au cinéma pour des raisons qui n'ont rien à voir avec le fait qu'elles pensent que le film sera bon. Si on change cela, et si on ne va plus au cinéma que lorsqu'on a une raison sérieuse de penser que le film sera bon, on ôtera aux majors une grande part de leurs revenus.

THORBURN : Une façon de comprendre ce discours aujourd'hui, je pense, est de se rendre compte qu'à chaque apparition d'innovations techniques à même de transformer fortement la société, on assiste à une lutte pour leur contrôle. La situation actuelle ne fait qu'éprouver des événements semblables et passés. Si l'on adopte ce point de vue, nul n'est besoin de se laisser envahir par le désespoir ou même le pessimisme, quant à la situation qui sera atteinte à long terme. Mais à court terme, les luttes pour le contrôle du texte, des images, et de toutes les formes d'information seront probablement douloureuses et intenses. En tant que professeur de médias, j'ai vu par exemple, ces dernières années, mon accès aux images restreint d'une manière sans précédent dans mon expérience. Si je souhaite illustrer un essai de photographies — même si elles sont extraites de films, il est devenu bien plus difficile d'obtenir les autorisations, et les prix ont considérablement augmenté. Et c'est en vain que j'invoque des arguments de recherche et que je mentionne le droit de « l'usage loyal ». Je pense donc, en ces temps de transformations accélérées, qu'il ne faut pas tant craindre les perspectives à long terme que ce qui va se produire à court terme. Dans tous les cas, il nous faut comprendre les événements actuels comme un épisode supplémentaire de la lutte pour le contrôle des ressources technologiques, principe récurrent dans la société occidentale.

Il est également essentiel de comprendre que l'histoire des anciennes technologies est elle-même fort complexe. L'imprimerie a eu par exemple un impact radicalement différent en Espagne, en Angleterre et en

France.

QUESTION : Ce qui m'ennuie souvent, quand j'entends des débats portant sur le copyright, c'est qu'ils commencent souvent par « Nous voulons un demi-tour complet, et nous débarrasser de tout type de contrôle. » Il me semble que les trois catégories présentées suggèrent en partie que le copyright relève d'une certaine sagesse. Certains critiques du système actuel du copyright croient, en fait, qu'il devrait être limité et fonctionner plus à la manière des brevets et des marques déposées, notamment en termes de durée. Je me demande ce que pense notre intervenant de cette stratégie.

STALLMAN : Je suis d'accord que raccourcir la durée du copyright est une bonne idée. Il n'y absolument aucun besoin, pour encourager la publication, que le copyright dure jusqu'à 150 ans — ce qui est parfois le cas avec les lois actuelles. Certaines entreprises nous ont expliqué qu'une période de 75 ans de copyright ne suffisent pas à rendre possible la production d'une œuvre. Je les mets au défi de me présenter des livres de compte et des projections sur les 75 prochaines années pour prouver cette assertion. Ce qu'elles recherchaient, en réalité, c'était à étendre la durée du copyright portant sur les œuvres anciennes, afin de pouvoir continuer à en restreindre l'utilisation. Mais la manière dont on peut encourager la production d'œuvres dans les années 1920 en étendant la durée du copyright de nos jours m'échappe et me dépasse totalement, à moins que ces entreprises ne disposent d'une machine à remonter dans le temps. Ah oui, en effet, elles ont une telle machine dans l'un de leurs films. C'est ce qui a dû détraquer leur raisonnement.

QUESTION : Avez-vous réfléchi à l'extension du concept de « l'usage loyal », et avez-vous abouti à des conclusions et à des nuances que vous accepteriez de partager avec nous ?

STALLMAN : L'idée d'autoriser toute copie exacte des œuvres appartenant à deux des trois catégories que j'ai définies peut être considérée comme une extension de l'usage loyal. C'est plus fort que l'usage loyal tel qu'il existe actuellement. Si vous pensez que le public abandonne certaines libertés en échange d'un peu plus de progrès, alors vous pouvez tracer la limite en plusieurs endroits différents. Quelles libertés abandonner, et quelles libertés conserver ?

QUESTION : Pour étendre un peu le champ de la conversation, je voudrais mentionner le concept de représentation publique, dont on dispose dans certains domaines. Par exemple, les lois sur le copyright ne nous interdisent pas de chanter des chants de Noël lorsque c'est la saison, mais elles interdisent toute représentation publique. Et je me demande s'il ne pourrait pas être utile, au lieu d'étendre l'usage loyal sans limitation à toute copie exacte faite dans un contexte non commercial, de réfléchir à une solution intermédiaire entre cela et la situation actuelle.

STALLMAN : Je pensais que cela suffirait, mais Napster m'a convaincu du contraire, parce que Napster est utilisé dans le contexte d'une redistribution de copies verbatim de manière non commerciale. Le serveur de Napster, en lui-même, est une activité commerciale, mais les gens qui y déposent des morceaux le font de manière non lucrative, et ils auraient tout aussi bien pu déposer ces morceaux sur leur propre site web. L'incroyable excitation et l'intérêt suscités par Napster ont montré la grande utilité de ce mode de distribution. C'est pourquoi je suis désormais convaincu que les gens devraient avoir le droit de redistribuer publiquement et non commercialement des copies exactes de tout.

QUESTION : On m'a récemment suggéré, pour Napster, l'analogie de la bibliothèque publique. Je pense que ceux d'entre vous qui ont écouté les arguments présentés par Napster ont entendu cette analogie. Je me demandais si vous souhaiteriez la commenter. Les défenseurs de ceux qui pensent que Napster devrait continuer et qu'il ne faut pas restreindre ce service disent parfois : « Quand on se rend dans une bibliothèque publique et qu'on emprunte un livre, on ne paie rien, et le même livre peut être emprunté des dizaines voire des centaines de fois, sans argent à la clef. En quoi Napster est-il différent ? ».

STALLMAN : Ce n'est pas exactement la même chose. Mais il faut signaler que les éditeurs veulent transformer les bibliothèques publiques en points de vente où on paie à chaque utilisation. Ils sont donc également opposés aux bibliothèques publiques.

QUESTION : Ces réflexions sur le copyright peuvent-elles suggérer des idées sur les problèmes posés par les lois sur les brevets, notamment sur la possibilité de fabriquer des médicaments génériques et bon marché pour l'Afrique ?

STALLMAN : Non, cela n'a rien à voir. Les problèmes posés par les brevets sont complètement différents des problèmes posés par les copyrights. L'idée qu'ils ont quelque chose de commun est l'une des conséquences malheureuses de l'utilisation du terme de « propriété intellectuelle » et d'encourager les gens à confondre toutes ces questions. Comme vous l'avez entendu, j'ai parlé de problèmes où le prix d'une copie n'est pas crucial. Et quel est le point crucial dans la production de médicaments anti-SIDA pour l'Afrique ? C'est le prix, et rien d'autre.

Les problèmes que j'ai évoqués se posent parce que les technologies numériques donnent à chacun la possibilité de réaliser des copies. Rien ne nous donne à tous la possibilité de copier des médicaments. Je n'ai pas la faculté de copier un médicament en ma possession. En fait, personne n'a cette faculté : ce n'est pas ainsi qu'on fabrique les médicaments. Les médicaments ne peuvent être fabriqués que dans des usines onéreuses et centralisées, qu'ils soient génériques ou importés des États-Unis. Dans tous les cas, ils seront produits dans un petit nombre d'usines, et les problèmes posés concernent simplement leur prix et s'ils sont ou non disponibles à un prix que les Africains peuvent payer.

C'est donc un problème extrêmement important, mais complètement différent. Seul un domaine des brevets pose les mêmes problèmes de liberté de copie, et c'est l'agriculture. En effet, certaines choses brevetées peuvent être — plus ou moins — copiées : le vivant. Le vivant se recopie quand il se reproduit. Il ne s'agit pas nécessairement d'une copie exacte : les gènes sont mélangés. Mais le fait est que depuis des millénaires, les agriculteurs ont exploité cette propriété qu'ont les plantes qu'ils font pousser (ou les animaux qu'ils élèvent) de se recopier. L'agriculture, en un sens, revient à recopier ce qu'on fait pousser (ou élève), et à le recopier d'année en année. Quand des variétés de plantes et d'animaux sont brevetées, quand elles utilisent des gènes brevetés, on interdit de fait aux agriculteurs d'exploiter cette propriété de la nature.

Une variété ainsi brevetée poussait sur le champ d'un agriculteur canadien. Ce dernier disait « Ce n'est pas délibéré de ma part. Le vent a soufflé, et les gènes de ces pollens se sont retrouvés dans mes plantes. » On lui a répondu que cela n'avait pas d'importance ; lui a quand même fallu détruire sa récolte. Ce fut un exemple extrême de coalition entre un gouvernement et un monopole.

Je pense donc, par application des mêmes principes que j'ai appliqués aux copies sur ordinateur, les agriculteurs devraient avoir un droit inaliénable de replanter leurs graines et d'élever leurs troupeaux. On peut imaginer des brevets concernant les producteurs de graines, mais ils ne devraient pas s'appliquer aux agriculteurs.

QUESTION : Les licences ne suffisent pas à rendre un modèle viable. Pouvez-vous développer ce point ?

STALLMAN : Absolument. Voyez-vous, je ne connais pas les réponses. Mais je pense que l'idéalisme est une partie cruciale du développement d'informations libres et fonctionnelles. Il faut que les gens pensent qu'il est important que cette information soit libre, que seule une information libre peut être pleinement exploitée. Qu'une information dont l'utilisation est restreinte ne peut être pleinement exploitée. Il faut que les gens pensent que les informations non libres sont une tentative de les diviser et de les maintenir dans un état de dépendance, la tête sous l'eau. Alors seulement peuvent-ils avoir l'idée de se donner la main pour produire les informations qu'ils veulent utiliser, sans qu'elles soient sous le contrôle d'un puissant qui leur dicterait leur conduite.

Voilà un élément qui accélère prodigieusement les choses. Je ne sais pas combien cela fonctionnera dans divers domaines, mais je pense pouvoir imaginer une manière de mener cela à bien dans le domaine de l'éducation — et des manuels. Nombreux sont les professeurs de par le monde qui ne travaillent pas dans des universités prestigieuses — mais au lycée, ou dans des universités de seconde zone — et qui n'écrivent et ne publient pas, faute de demande. Mais ils sont souvent très intelligents et instruits. Nombreux sont ceux qui connaissent bien leur matière et qui pourraient écrire des manuels portant sur de nombreux sujets, les partager avec le monde, et recevoir énormément de commentaires, de remerciements et d'encouragements de la part de ceux qui en auront bénéficié.

QUESTION : C'est ce à quoi je pensais. Je connais l'histoire de l'éducation ; c'est mon métier: je m'occupe de projets éducatifs et électroniques. Mais ce qui est étrange, c'est que je ne connais aucun exemple de cela. En connaissez-vous ?

STALLMAN : Non. J'ai commencé à proposer cette idée d'encyclopédie et de ressources éducatives libres voici quelques années, et je pensais alors que cela prendrait une décennie avant de s'ébranler. Nous avons déjà une encyclopédie en bonne voie de réalisation. Les choses avancent plus vite que ce que j'imaginai. Je pense qu'on a besoin de quelques professeurs qui se mettraient à écrire quelques manuels libres. Écrivez un manuel sur votre matière préférée — n'en écrivez même qu'une portion. Écrivez quelques chapitres et incitez d'autres personnes à écrire les chapitres restants.

QUESTION : Je voyais plus loin encore. Ce qui est important, dans votre idée de structure, c'est que quelqu'un crée une infrastructure à laquelle tout le monde peut ensuite contribuer. Je ne connais aucune infrastructure ciblée sur l'enfance, de niveau maternelle et primaire, où on puisse contribuer de la manière que vous proposez.

Je peux obtenir les informations dont j'ai besoin de nombreux endroits différents, mais ces informations n'étant pas publiées sous des licences libres, je ne peux en faire usage dans un manuel libre.

STALLMAN : En fait, le copyright ne couvre pas les faits, mais la manière dont ils sont rédigés. On peut donc apprendre un domaine à partir de la source qu'on veut et ensuite écrire un manuel, qu'on peut publier sous une licence libre si tel est notre désir.

QUESTION : Mais je ne peux rédiger seul tous les manuels dont un étudiant aura besoin à l'école.

STALLMAN : C'est vrai. Et je n'ai pas, moi non plus, écrit un système d'exploitation libre complet. J'en ai écrit quelques morceaux, et j'ai invité d'autres personnes à me rejoindre et à écrire les morceaux manquants. J'ai donc montré l'exemple. J'ai dit « Je suis cette voie-là. Suivez-moi et nous y arriverons ensemble ». Et on m'a suivi en nombre suffisant, de telle sorte que nous avons atteint ce but désormais. Si vous prenez le problème en vous demandant comment vous allez vous y prendre pour abattre cette quantité gigantesque de travail, il y a de quoi être découragé. Ne regardez donc pas les choses ainsi ! Pensez plutôt que vous allez faire un pas et qu'après ce pas, d'autres vous rejoindront et feront quelques pas et qu'ensemble, vous arriverez enfin au but que vous vous êtes fixé.

Si on suppose que l'humanité ne se détruit pas, le travail réalisé aujourd'hui dans le but de produire une infrastructure éducative libre et des ressources scolaires libres pour tous, subsistera aussi longtemps que l'humanité vivra. Si cela doit prendre 20 ans, et alors ? Ne pensez donc pas en termes de taille de l'ensemble de la tâche. Pensez en termes du morceau que vous allez accomplir. Cela montrera l'exemple à suivre, et d'autres s'occuperont des autres morceaux.

*Copyright © 2001, 2007, 2008 Free Software Foundation, Inc.,
51 Franklin St, Fifth Floor, Boston, MA 02110, USA*

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

*Traduction : Sébastien Blondeel
Révision : trad-gnu@april.org*

Dernière mise à jour : 2009/01/19 09:27:59

Traductions de cette page

[Česky](#) [cs] [English](#) [en] [Español](#) [es] [Français](#) [fr] [Italiano](#) [it] [português do Brasil](#) [pt-br]

Logiciel Libre : Liberté et coopération

Transcription du
discours de Richard M. Stallman,
«Logiciel Libre : Liberté et coopération»
Université de New York,
le 29 mai 2001.

URETSKY : Je suis Mike Uretsky. Je travaille à l'école de commerce Stern. Je suis aussi un des co-directeurs du centre pour la technologie de pointe. Et au nom de tous au Centre pour la technologie de pointe, je veux vous souhaiter la bienvenue. Je voudrais faire quelques commentaires avant de passer la parole à Ed qui présentera l'orateur.

Le rôle d'une université est d'être un lieu de débats et de permettre des discussions intéressantes. Et le rôle d'une grande université est d'offrir des discussions particulièrement intéressantes. Cet exposé particulier, ce séminaire, répond parfaitement à cet impératif. Je trouve la discussion sur l'Open Source particulièrement intéressante. D'une certaine manière...*[rires]*

STALLMAN : Je fais du logiciel libre. L'Open Source c'est un autre mouvement. *[rires et applaudissements]*

URETSKY : Quand j'ai débuté dans ce champ disciplinaire dans les années 60, fondamentalement, les logiciels étaient libres. Puis nous sommes rentrés dans des cycles. Cela débuta par la liberté puis les fabricants de logiciels pour étendre leur marché les ont poussés dans d'autres directions. À l'arrivée du PC ce fut la même chose dans le domaine du développement. Il y a un philosophe français très intéressant, Pierre Lévy, qui parle d'un mouvement dans cette direction et parle du mouvement dans le cyberspace pas seulement en relation avec la technologie mais aussi avec la restructuration sociale et politique, à travers un changement des types de relations qui va améliorer le bien-être de l'humanité, débat qui traverse de nombreuses disciplines qui travaillent généralement en solo à l'université. Nous espérons donc de très intéressantes discussions. Ed ?

SCHONBERG : - Je suis Ed Schonberg du département d'informatique de l'Institut Courant. Laissez-moi vous souhaiter la bienvenue pour cet événement. Les présentateurs sont généralement, et particulièrement, un aspect inutile des présentations publiques, mais, dans ce cas, ils servent un but utile comme le propos de Mike vient facilement de le prouver. Parce qu'un présentateur par exemple, par des commentaires inappropriés peut permettre à l'orateur de corriger *[rires]* et préciser considérablement les paramètres du débat.

Aussi laissez-moi faire la présentation la plus brève possible de quelqu'un qui n'en n'a pas besoin. Richard est le parfait exemple de quelqu'un qui, agissant localement, commença à penser globalement les problèmes d'inaccessibilité du code source des pilotes d'imprimante au laboratoire d'Intelligence Artificielle il y a bien des années. Il a développé une philosophie cohérente qui nous a tous forcé à réexaminer nos idées sur la façon dont le logiciel est produit, sur ce que signifie la propriété intellectuelle, et sur ce que représente la communauté du logiciel. Bienvenue à Richard Stallman. *[applaudissements]*

STALLMAN : - Quelqu'un peut-il me prêter une montre ? *[rires]* Merci. Bien, je voudrais remercier Microsoft de me donner l'occasion aujourd'hui *[rires]* d'être ici. Ces dernières semaines, je me sentais comme un auteur dont le livre a été fortuitement interdit quelque part. *[rires]* Sauf que tous les articles sur lui mentionnent un nom d'auteur erroné, parce que Microsoft décrit la Licence GNU GPL comme une licence Open Source et la majorité de la couverture de presse a suivi. La plupart des gens ignore, innocemment bien sûr, que notre travail n'a rien à voir avec l'Open Source, et qu'en réalité nous l'avons mené bien avant que l'on ne l'estampille du terme « Open source ».

Nous sommes dans le Mouvement pour le logiciel Libre (FSF) et je vais vous parler de ce qu'est le mouvement pour le logiciel libre, ce qu'il signifie, ce que nous avons fait, et puisque nous sommes réunis par une école de commerce, je vous en dirai un peu plus qu'à l'habitude sur les relations du logiciel libre et du commerce ou d'autres champs de la vie sociale.

Bien, certains d'entre vous n'écriront peut être jamais de logiciels mais vous cuisinez peut-être. Et si vous cuisinez, à moins que vous ne soyez un grand chef, vous utilisez probablement des recettes. Et si vous utilisez des recettes vous avez probablement fait l'expérience de recevoir la copie d'une recette de la part d'un ami qui la partage avec vous. Et vous avez sans doute fait également l'expérience - à moins d'être un néophyte complet - de changer cette recette. Il y a des choses que l'on n'est pas obligé de faire exactement. Vous pouvez laisser tomber certains ingrédients. Ajouter des champignons parce que vous aimez les champignons. Mettre un peu moins de sel parce que votre médecin vous a recommandé de supprimer le sel, que sais-je...? Vous pouvez même faire des changements plus importants selon vos talents. Et si vous avez fait des changements dans une recette et que vos amis l'ont appréciée, l'un d'entre eux vous a peut-être dit « hé, pourrais-je avoir la recette ? ». Et alors vous faites une copie pour votre ami. C'est une chose qu'on fait naturellement avec des recettes de toutes sortes.

En fait, une recette ressemble beaucoup à un programme informatique. Un programme d'ordinateur est comme une recette. Une série d'étapes à mener pour obtenir le résultat que vous attendez. Alors il est tout naturel de faire la même chose avec un programme. Donner une copie à un ami. Apporter des modifications parce que le travail pour lequel il a été écrit n'est pas tout à fait ce que vous voulez. Il a fait du bon travail pour quelqu'un d'autre mais votre travail est différent et après l'avoir changé il est probable qu'il puisse servir à d'autres. Peut-être qu'ils ont à faire un travail comme le vôtre, alors ils vous demandent une copie, et si vous êtes une personne aimable, vous donnez votre copie. C'est une façon de se comporter convenablement.

Alors imaginez que les recettes soient enfermées dans des boîtes noires. Vous ne pourriez pas savoir les ingrédients qu'elles utilisent, encore moins les changer. Et imaginez, si vous faisiez une copie pour un ami qu'on vous traiterait de pirate et qu'on essaierait de vous mettre en prison pour des années. Ce serait un énorme tollé de la part de tous ceux qui sont habitués à partager des recettes de cuisine. Mais c'est exactement ce qui se passe dans le monde du logiciel propriétaire. Un monde dans lequel on empêche et on interdit un comportement correct envers les autres personnes.

Maintenant, pourquoi ai-je remarqué cela ? Je l'ai remarqué parce que j'ai eu la bonne fortune dans les années 70 de faire partie d'une communauté d'informaticiens qui partageaient les logiciels. On pourrait faire remonter ses racines aux origines de l'informatique mais dans les années 70 c'était plutôt rare. En fait c'était en quelque sorte un cas extrême, parce que dans le laboratoire dans lequel je travaillais, le système d'exploitation avait été développé par les gens de cette communauté et nous le partagions avec quiconque. N'importe qui était le bienvenu, pouvait regarder et prendre une copie quel qu'en soit l'usage. Il n'y avait pas de notice de copyright sur ces programmes. Et nous étions confiants dans ce mode de vie. Nous n'avions pas lutté pour cela, nous vivions comme cela. Nous pensions continuer comme cela. Il y avait du logiciel libre mais pas de mouvement pour le logiciel libre.

Mais alors, notre communauté a été détruite par une série de calamités. Et à la fin, elle fut balayée. L'ordinateur PDP-10 que nous utilisions pour tout notre travail fut abandonné. Et notre système d'exploitation, le Système de partage du temps incompatible (Incompatible Timesharing System), qui avait été écrit à partir des années 60, était à cette époque en langage assembleur, ce qui le rendait spécifique à un type d'architecture d'ordinateur, ce qui fait que tout notre travail est tombé en poussière. Les 20 ans de travail de notre communauté sont tombés en poussière.

Mais avant que cela n'arrive, une expérience m'a préparé et m'a aidé à voir ce qu'il fallait faire. À un certain moment, Xerox a donné au laboratoire pour l'Intelligence Artificielle où je travaillais, une imprimante laser, et c'était un beau cadeau car c'était la première fois qu'en dehors de Xerox quelqu'un possédait une imprimante laser. Elle était très rapide, une page à la seconde, très bien par bien des aspects, mais elle n'était pas fiable, parce qu'en fait c'était un copieur rapide qui avait été modifié pour devenir une imprimante. Vous savez les copieurs tombent en panne mais il y a sur place quelqu'un pour les réparer. L'imprimante tomba en panne mais personne ne l'a vu aussi resta-t-elle en panne longtemps.

Nous avons bien une idée pour résoudre ce problème. Faire en sorte que chaque fois que l'imprimante tombe en panne elle avertisse notre ordinateur et les utilisateurs qui attendent une sortie d'imprimante. Car bien sûr, si vous attendez une sortie d'imprimante et que vous savez qu'elle est en panne vous n'allez pas rester assis pour l'éternité, vous irez réparer.

Mais à ce moment nous avons été entravés par le fait que le pilote de l'imprimante n'était pas un

logiciel libre. Il était livré avec mais c'était un programme binaire. Nous n'avions pas le code source. Xerox ne nous avait pas autorisé à avoir le source. Si bien que malgré nos talents d'informaticiens, (nous avons écrit notre système d'exploitation), nous ne pouvions ajouter cette fonction au pilote d'imprimante.

Nous devions simplement souffrir en attendant; cela prenait une ou deux heures pour avoir son impression car la machine était en panne la plupart du temps. De temps à autre vous attendiez en vous disant « je sais que ça va planter. Je vais attendre une heure et aller chercher mon texte » et alors vous vous aperceviez que la machine était en panne tout le temps et que personne d'autre n'avait réparé. Alors vous la répariez et attendiez une nouvelle demi-heure. Ensuite vous reveniez et vous voyiez qu'elle était tombée en panne avant même de commencer votre impression. Elle imprimait trois minutes et tombait en panne pour 30 minutes. Frustration jusque là ! Mais le pire était de savoir que nous aurions pu la réparer, mais quelqu'un par pur égoïsme nous bloquait en nous empêchant d'améliorer son programme. D'où un ressentiment, bien sûr...

Et alors j'ai entendu dire que quelqu'un avait une copie de ce programme à l'université Carnegie-Mellon. En visite là bas un peu plus tard, je me rends à son bureau et je dis : « Salut, je suis du MIT, pourrais-je avoir une copie du code source de l'imprimante ? » et il répond : « Non, j'ai promis de ne pas vous donner de copie » *[rires]*. J'étais soufflé. J'étais si... J'étais si en colère et je ne savais pas quoi faire pour réparer cette injustice. Tout ce que je pus penser c'était sortir. J'ai tourné les talons et je suis sorti. Peut-être que j'ai claqué la porte... *[rires]* Et j'y ai repensé plus tard parce que j'ai réalisé que je n'étais pas simplement en face d'un fait isolé mais d'un phénomène social qui était important et affectait beaucoup de gens.

Pour moi, par chance ce n'était qu'un échantillon, mais d'autres gens étaient obligés de vivre avec ça tout le temps. Et j'y ai repensé plus longuement. Vous voyez, il avait promis de refuser de coopérer avec nous, ses collègues au MIT. Il nous avait trahis. Mais il ne l'avait pas fait qu'à nous. Il y a des chances qu'il vous l'ait fait à vous aussi. *[rires]* et à vous aussi. Et certainement à une bonne partie de ceux qui sont dans cette salle, à l'exception de ceux qui n'étaient pas nés en 1980. Il avait promis de ne pas coopérer avec la population de la planète Terre. Il avait signé un accord de non-divulcation.

C'était ma première confrontation avec un accord de non-divulcation et cela m'a appris une importante leçon, une leçon qui est importante parce que la plupart des programmeurs ne l'apprennent jamais. Vous voyez, c'était ma première rencontre avec un accord de non-divulcation et j'en étais victime. Moi et mon laboratoire entier nous étions victimes. Et la leçon que j'ai apprise c'est que les accords de non-divulcation font des victimes.

Ils ne sont pas innocents, ils ne sont pas inoffensifs. La plupart des programmeurs rencontrent un accord de non-divulcation lorsqu'ils sont invités à en signer un. Et il y a toujours une sorte de tentation, un bénéfice qu'ils auront s'ils signent. Alors ils s'inventent des excuses. Ils disent : « De toute façon, il n'aura pas de copie, alors pourquoi ne rejoindrais-je pas la conspiration pour l'en priver ? ». Ils disent : « Ça se fait toujours comme ça, qui suis-je pour m'y opposer ? ». Ils disent : « Si je ne signe pas quelqu'un d'autre le fera. ». Diverses excuses pour tromper leur conscience.

Mais quand on m'a invité à signer un accord de non-divulcation, ma conscience était déjà en éveil. Elle se rappelait comme j'étais en colère lorsque quelqu'un avait promis de ne pas m'aider, moi et mon labo, à résoudre notre problème. Je ne pouvais pas retourner ma veste et faire la même chose à quelqu'un qui ne m'avait fait aucun mal. Vous savez si quelqu'un me demandait de promettre de ne pas partager une information utile avec un ennemi détesté je le ferais. Si quelqu'un a fait quelque chose de mal il le mérite. Mais des étrangers... Ils ne m'ont fait aucun mal. Comment pourraient-ils mériter ce genre de mauvais traitement. On ne peut pas se mettre à maltraiter tout le monde ou n'importe qui, sinon vous devenez un prédateur de la société. Alors j'ai dit : « Merci de m'offrir ce beau package logiciel, mais je ne peux l'accepter en bonne conscience aux conditions que vous exigez, aussi vais-je m'en passer. Merci beaucoup. ». Ainsi, je n'ai jamais consciemment signé d'accord de non-divulcation pour une information technique utile tel qu'un programme.

Cela dit, il y a d'autres sortes d'informations qui posent d'autres problèmes éthiques. Par exemple, il y a les informations personnelles. Vous savez, si vous voulez me parler de ce qui se passe entre vous et votre petit ami et que vous me demandez de n'en parler à personne, je peux accepter de garder le secret pour vous, parce que ce n'est pas une information technique utile en général.

En fait, ce n'est probablement pas utile généralement. *[rires]* Il y a une petite chance, c'est possible,

que vous puissiez me révéler une merveilleuse nouvelle technique sexuelle [*rires*] et j'éprouverais le devoir moral [*rires*] de la révéler au reste de l'humanité, pour que chacun puisse en profiter.

Donc, je devrais lever ma promesse. Si ce sont juste des détails sur qui veut ceci et qui est en colère contre untel, des choses comme ça, du feuilleton télé, cela je peux le garder confidentiellement... Mais quelque chose dont l'humanité bénéficierait, je ne peux le garder pour moi. Vous voyez, le but de la science et de la technologie est de vivre mieux. Si nous décidons de garder cette information, si nous la gardons secrète, nous trahissons la mission de notre spécialité. Et ceci j'ai décidé de ne pas le faire.

Mais, pendant ce temps ma communauté s'était effondrée et c'était terrible et ça me laissait dans une mauvaise situation. Tout notre système était obsolète parce que le PDP-10 était obsolète et je ne pouvais plus travailler en tant que développeur de système d'exploitation comme je l'avais fait. Cela dépendait de l'appartenance à la communauté utilisant ce logiciel et l'améliorant. Cela n'était plus possible et cela m'amena à un dilemme moral. Qu'allais-je faire ?

Parce que la possibilité la plus évidente signifiait aller contre la décision que j'avais prise. La plus évidente possibilité était de m'adapter au changement du monde. Accepter que les choses soient différentes et que je n'avais qu'à abandonner ces principes et commencer à signer des accords de non-divulgaration pour des systèmes d'exploitation propriétaires et probablement écrire des logiciels propriétaires à mon tour. Mais j'ai réalisé que même si de cette façon j'avais un moyen de m'amuser à coder et de gagner de l'argent en même temps, surtout si je faisais ça ailleurs qu'au MIT, à la fin j'aurais dû me retourner sur ma carrière et dire : « J'ai passé ma vie à construire des murs pour diviser les gens ». Et j'aurais eu honte de ma vie.

Alors, j'ai cherché une autre alternative et il y en avait une évidente. Je pouvais quitter l'informatique et faire autre chose. Je n'avais aucun autre talent mais j'aurais pu être serveur [*rires*]. Pas dans un restaurant chic, ils n'auraient pas voulu de moi, mais j'aurais pu être serveur quelque part. De nombreux programmeurs disent : « Les gens qui emploient des programmeurs exigent ceci, cela, si je ne le fais pas je mourrai de faim ». C'est le mot qu'ils utilisent littéralement. Bon, comme serveur je ne risquais pas de mourir de faim [*rires*]. Réellement les programmeurs ne courent aucun danger mais c'est important voyez-vous, car vous pouvez justifier de faire quelque chose qui blesse autrui en disant : « Sinon quelque chose de pire va m'arriver ». Si vous êtes vraiment sur le point de crever de faim vous pouvez justifier d'écrire du logiciel propriétaire [*rires*]. Et si quelqu'un vous menace d'une arme je dirais même que c'est pardonnable [*rires*]. Mais j'avais trouvé une façon de survivre sans enfreindre mon éthique aussi cette excuse était-elle irrecevable.

Cependant, je réalisais qu'être serveur ne serait pas drôle pour moi et que ce serait gâcher mes talents de programmeur. Je devais éviter de mal utiliser mes talents. Écrire des logiciels propriétaires aurait été mal utiliser mes talents. Encourager les autres à vivre dans un monde de logiciels propriétaires aurait signifié mal utiliser mes talents. Aussi, il valait mieux les gâcher que les utiliser à mauvais escient, mais ce n'était pas encore vraiment bon.

Pour toutes ces raisons, j'ai cherché une autre alternative. Que pouvait faire un développeur de systèmes d'exploitation pour rendre le monde meilleur. J'ai alors réalisé qu'un développeur de système d'exploitation, c'était exactement ce qu'il fallait. Le problème pour moi, et tous les autres c'était que tous les systèmes d'exploitation disponibles pour les ordinateurs modernes étaient propriétaires. Les systèmes d'exploitation libres étaient pour de vieux ordinateurs, obsolètes, n'est-ce pas ? Si vous vouliez un ordinateur moderne vous étiez obligé d'adopter un système d'exploitation propriétaire. Alors si un développeur écrivait un autre système d'exploitation et disait : « Venez tous et partagez cela, vous êtes les bienvenus » cela permettrait à chacun de sortir du dilemme, cela offrirait une nouvelle alternative. J'ai réalisé que je pouvais faire quelque chose qui résoudrait le problème. J'avais les talents requis et c'était la chose la plus utile que je pourrais faire de ma vie. Et c'était un problème que personne d'autre n'essayait de résoudre. J'étais là, de pire en pire, et j'étais seul. Alors j'ai senti : « Je suis élu. Je dois travailler là-dessus. Si ce n'est pas moi, qui ? ». J'ai donc décidé de développer un système d'exploitation libre ou de mourir. De vieillesse bien sûr. [*rires*]

Alors je devais décider quelle sorte de système d'exploitation ce serait. Il y a quelques décisions techniques à prendre. J'ai décidé de faire un système compatible avec Unix pour un certain nombre de raisons. En premier lieu je venais de voir un système que j'adorais devenir obsolète parce qu'il était écrit pour un type spécial d'ordinateur. Je ne voulais pas que cela se reproduise. Nous avons besoin d'un système portable. Si je suivais le schéma d'Unix, j'avais toute chance de faire un système portable et utilisable. Mieux,

ce [nouveau système devait être] compatible dans le détail. La raison en est que les utilisateurs détestent les changements incompatibles. Si j'avais conçu le système de la façon que je préfère -- ce que j'aurais adoré, j'en suis sûr -- j'aurais produit quelque chose d'incompatible. Les détails auraient été différents. Donc, si j'avais conçu le système ainsi les gens m'auraient dit : « Bien, c'est très joli mais c'est incompatible. Ce sera trop d'efforts de changer. Nous ne pouvons nous permettre trop d'efforts pour utiliser votre système au lieu d'Unix, alors nous garderons Unix ». Voilà ce qu'ils auraient dit.

Maintenant, si je voulais créer une communauté où il y aurait des gens, des gens utilisant ce nouveau système et bénéficiant de la liberté et de la coopération, je devais faire un système que les gens utiliseraient, qu'ils trouveraient facile à adopter, qui ne serait pas en échec dès le départ. Rendre ce système compatible avec Unix a entraîné les premières décisions de conception, parce qu'Unix consiste en de nombreux morceaux et ils communiquent à travers des interfaces plus ou moins documentées. Alors si vous voulez être compatible avec Unix, il vous faut remplacer chaque morceau, un par un, par un morceau compatible. Les décisions de conception sont dans un morceau et elles peuvent être prises plus tard quelque soit la personne qui écrive le morceau ; on ne devait pas prendre toutes les décisions dès le départ.

La seule chose que nous devons faire pour commencer le travail était de trouver un nom pour le système. Nous les hackers, nous cherchons toujours des noms drôles ou méchants pour un programme, parce que penser aux gens qui s'amuse du nom c'est la moitié du plaisir de l'écriture [*rires*]. Nous avons aussi une tradition d'acronymes récursifs qui consiste à dire que le programme qu'on crée est similaire à un programme existant. On peut lui donner un nom récursif qui dit celui-ci n'est pas celui-là. Par exemple, il y avait beaucoup d'éditeurs de texte TICO dans les années 60 et 70 et ils étaient généralement appelés quelque chose ou autre TICO. A cette époque un hacker malin appela le sien Tint, pour Tint Is Not Tico, le premier acronyme récursif. En 1975, j'ai développé le premier éditeur de texte EMACS et il y a eu de nombreuses imitations d'EMACS, beaucoup étaient appelés quelque chose Emacs, mais l'un était nommé FINE (N.d.T : « BIEN! ») pour Fine is not Emacs. Et il y a eu SINE pour Sine is not Emacs, et IINA pour Ina Is Not Emacs, et il y a eu MINCE pour Mince Is Not Complete Emacs [*rires*], c'était une imitation incomplète. Ensuite IINA a été complètement réécrit et la nouvelle version a été appelée ZWII pour Zwii Was Ina Initially [*rires*].

Alors j'ai cherché un acronyme récursif pour Quelque Chose n'est pas Unix (N.d.T : Something is not Unix). Et j'ai essayé les 26 lettres mais aucune n'était un mot [*rires*]. Hmm, essayons autre chose. J'ai fait une contraction. De cette façon, je pouvais avoir un acronyme de trois lettres pour Something's Not Unix. J'ai essayé des lettres et j'en suis arrivé au mot « GNU ». Le mot GNU (N.d.T : gnou) est le plus drôle de la langue anglaise [*rires*]. C'était ça. Bien sûr, la raison de cette drôlerie vient du fait que selon le dictionnaire, il doit se dire New (N.d.T : nouveau). Voyez-vous ? Et c'est pourquoi les gens l'utilisent pour de nombreux jeux de mots.

Laissez-moi vous dire que c'est le nom d'un animal d'Afrique. Et la prononciation africaine avait un clic à l'intérieur [*rires*]. Les colonisateurs européens quand ils arrivèrent là-bas n'entendaient pas apprendre la prononciation d'un clic. Alors ils l'ont laissé dehors et écrit un G qui signifiait : « il y a un autre son qui est supposé être là mais que nous ne prononçons pas ». [*rires*]. Cette nuit, je pars en Afrique du Sud et je leur ai demandé de me trouver quelqu'un qui puisse m'apprendre à prononcer les clics. [*rires*]. Ainsi, je saurai prononcer correctement le GNU quand il s'agit de l'animal.

Mais en ce qui concerne le nom de notre système la prononciation correcte est Gueu-nou, prononcez le « g » dur. Si vous parlez du « New operating system » (N.d.T. : nouveau système d'exploitation), vous embrouillez l'esprit des gens, parce que cela fait 17 ans que nous travaillons dessus et il n'est plus du tout « new » !

Mais il est toujours et sera toujours GNU; peu importe le nombre de gens qui l'appellent Linux par erreur. [*rires*]

Ainsi en janvier 84, je quitte mon job au MIT pour commencer à écrire des morceaux de GNU. Tout de même, ils ont été assez sympa pour me laisser utiliser leurs installations. Et, à cette époque, je pensais que j'écrirais tous les morceaux du système GNU en entier, et que je dirais : « Venez et prenez! » et que les gens commenceraient à l'utiliser. Cela ne se passa pas comme cela. Les premiers morceaux que j'ai écrits étaient juste aussi bons que les originaux avec moins de bogues [...] mais ils n'étaient pas terriblement excitants. Personne en particulier ne souhaitait l'utiliser ni l'installer. Mais alors en septembre 84, j'ai commencé à

écrire GNU Emacs, qui était ma seconde implémentation d'Emacs, et au début 85, il fonctionnait. Je pouvais l'utiliser pour mon travail d'édition, ce qui était un soulagement car je n'avais aucune intention d'utiliser VI, l'éditeur Unix. *[rires]*

Avant cela, je faisais mon travail d'édition sur une autre machine et je sauvegardais les fichiers sur le réseau pour pouvoir les tester. Mais quand GNU Emacs a fonctionné assez bien pour que je puisse l'utiliser, d'autres personnes ont voulu l'utiliser aussi.

J'ai dû travailler les détails de la distribution. Bien sûr j'ai mis une copie sur le FTP anonyme, et c'était bien pour les gens qui étaient sur le net, ils pouvaient prendre un fichier tar, mais beaucoup de programmeurs n'étaient pas sur le net en 85. Ils m'envoyaient des mails. « Puis-je avoir une copie ? » je devais savoir quoi leur répondre. J'aurais pu dire, je veux passer plus de temps à écrire des logiciels GNU et pas à enregistrer des bandes, aussi trouvez vous un ami qui a accès au net et qui vous le téléchargera et vous l'enregistrera sur bande. Et je suis sûr que les gens auraient trouvé ces amis tôt ou tard, vous savez... Ils auraient eu des copies. Mais, je n'avais pas de travail. En fait, je n'ai eu aucune profession depuis mon départ du MIT en 84. Aussi, je cherchais une façon de gagner de l'argent par mon travail sur le logiciel libre, donc j'ai commencé un commerce de logiciel libre. J'ai annoncé: « envoyez moi 150 dollars et je vous posterai une bande d'Emacs. » Et les commandes ont commencé à tomber, et vers le milieu de l'année elles dégringolaient...

Je recevais 8 à 10 commandes par mois. J'aurais pu en cas de nécessité vivre seulement de cela parce que j'ai toujours vécu simplement; je vis comme un étudiant à la base. Et j'aime ça car cela signifie que l'argent ne me dicte pas ce que je dois faire. Je peux faire ce qui me paraît important. Cela m'a libéré de faire ce qui semblait « rentable ». Aussi faites un effort pour éviter d'être englué dans les habitudes dispendieuses de l'American way of life, typique des américains typiques. Parce que si vous faites cela, les gens avec de l'argent vous dicteront quoi faire de votre vie. Vous ne pourrez pas faire ce qui est réellement important pour vous.

// les lignes suivantes ne se comprennent que si l'on sait qu'en anglais logiciel libre et logiciel gratuit se disent free software l'un comme l'autre.//

Bon, c'était bien mais les gens me disaient : « Qu'entendez vous par free software si cela coûte 150 dollars ? ». *[rires]* La raison de cette question était la confusion induite par l'ambiguïté du mot « free » en anglais. Le premier sens se réfère au prix et un autre se réfère à la liberté. Quand je parle de logiciel libre, je me réfère à la liberté et non au prix. [...] en tous cas je n'aurais pas passé autant d'années de ma vie pour faire gagner moins d'argent aux programmeurs. Ce n'est pas mon but. Je suis moi même programmeur et je ne m'offusque pas de gagner de l'argent. Je ne passerais pas ma vie à en gagner mais je ne refuse pas d'en gagner. Et je ne suis pas, l'éthique est la même pour tous, je ne suis pas contre le fait qu'un autre programmeur en gagne. Je ne veux pas faire baisser les prix. Ce n'est pas du tout le problème. L'enjeu c'est la liberté. La liberté de chaque personne qui utilise un logiciel, qu'on soit programmeur ou non.

À ce moment je dois vous donner une définition de ce qu'est le logiciel libre. Je préfère aller au réel car dire simplement : « Je crois en la liberté » est vide de sens. Il y a tant de libertés différentes en lesquelles croire, et qui sont en conflit l'une avec l'autre, que la vraie question politique est : « Quelles sont les libertés importantes, celles dont tout le monde doit être sûr de les avoir ? ».

Et maintenant je vais vous donner ma réponse dans ce domaine particulier qu'est l'usage du logiciel. Un programme est libre pour vous, utilisateur particulier, si vous bénéficiez des libertés suivantes :

- premièrement, la liberté zéro est la liberté d'utiliser un logiciel pour n'importe quel usage, de n'importe quelle manière.
- La liberté un est la liberté de s'aider soi-même en changeant le programme pour répondre à ses besoins.
- La liberté deux est celle d'aider son prochain en distribuant des copies du programme.
- Et la liberté trois est celle d'aider à bâtir votre communauté en publiant une version améliorée pour que les autres puissent bénéficier de votre travail.

Si vous avez toutes ces libertés, le programme est un logiciel libre, pour vous ; et c'est crucial. C'est pourquoi je le dis comme ça. J'expliquerai pourquoi plus tard quand je parlerai de la Licence Publique Générale GNU, mais pour le moment j'explique ce qu'est un logiciel libre, ce qui est une question plus

fondamentale.

La liberté zéro est assez évidente. Si vous n'êtes même pas autorisé à faire fonctionner le programme de la façon qui vous plaît, c'est un programme sacrément restrictif ! La plupart des programmes vous donnent la liberté zéro. Et la liberté zéro découle, légalement, des libertés un, deux et trois -- c'est la façon dont la loi sur le copyright fonctionne. Ainsi les libertés qui distinguent le logiciel libre du logiciel ordinaire sont les libertés un, deux et trois ; je vais donc en parler plus en détail et je dirai en quoi elles sont importantes.

La liberté un est celle de modifier le logiciel pour l'adapter à ses besoins. Cela peut signifier corriger des bogues. Cela peut signifier ajouter de nouvelles fonctions. Cela peut signifier porter le logiciel sur un autre type de système informatique. Cela peut signifier traduire tous les messages d'erreur en navajo. Vous devriez pouvoir apporter toutes les modifications que vous voulez librement.

Cela dit, il est évident que les programmeurs professionnels peuvent utiliser cette liberté de façon très effective mais ils ne sont pas les seuls. N'importe quelle personne d'intelligence normale peut apprendre un peu de programmation. Vous savez, il y a des travaux difficiles, des travaux faciles et tout le monde n'apprend pas suffisamment pour faire les travaux difficiles, mais beaucoup peuvent apprendre assez pour faire des travaux faciles, de la même façon que, il y a 50 ans, beaucoup, vraiment beaucoup d'américains ont appris à réparer leur voiture ce qui a permis aux USA d'avoir une armée motorisée pendant la seconde guerre mondiale et de gagner. Alors, chose très importante, avoir beaucoup de bricoleurs.

Et si vous refusez d'apprendre la technologie, cela veut dire que vous avez probablement beaucoup d'amis et que vous êtes doué dans l'art de les obliger à vous rendre service. [*rires*]. Certains d'entre eux sont probablement informaticiens. Alors vous pouvez demander à un de vos amis informaticiens : « Pourrais-tu changer ceci pour moi ?, et cette fonction ? » donc beaucoup de gens peuvent en bénéficier.

Si d'aventure vous n'avez pas cette liberté, ceci cause un préjudice matériel et pratique à la société. Cela fait de vous un prisonnier de votre logiciel. J'ai expliqué comment c'était dans le cas de l'imprimante laser. Vous savez, elle marchait mal et nous ne pouvions la réparer parce que nous étions prisonniers de notre logiciel.

Mais cela affecte aussi le moral des gens. Si l'ordinateur est constamment frustrant et que les gens l'utilisent, leur vies vont devenir frustrantes et s'ils l'utilisent dans leur métier, leur métier va devenir frustrant ; ils vont détester leur métier. Vous savez, les gens se protègent de la frustration en décidant de s'en fiche. Vous en arrivez à des gens qui disent : « Bon, j'ai fait acte de présence au boulot, c'est tout ce que j'ai à faire. Si je ne peux pas progresser ce n'est pas mon affaire, c'est l'affaire du patron ». Et quand ça arrive c'est mauvais pour ces gens, c'est mauvais pour la société toute entière. C'est la liberté un, la liberté de s'aider soi-même.

La liberté deux est celle d'aider votre prochain, en distribuant des copies du programme. Pour des êtres qui pensent et qui apprennent, partager un savoir utile est un acte fondamental d'amitié. Quand ces êtres utilisent des ordinateurs, cet acte d'amitié prend la forme d'un partage de logiciel. Les amis partagent entre eux. Les amis s'aident mutuellement. C'est la nature de l'amitié. Et en fait, l'essence de la bonne volonté -- la disposition à vouloir aider son prochain volontairement -- est la ressource la plus importante de la société. Elle fait la différence entre une société vivable et une jungle où chacun s'entre-dévore. Cette importance a été reconnue par les grandes religions du monde depuis des milliers d'années et elles essayent explicitement d'encourager cette attitude.

Quand j'allais à la maternelle, les enseignants essayaient de nous apprendre cette attitude, l'esprit de partage, en nous la faisant pratiquer. Ils pensaient que si nous le faisons nous l'apprendrions. Alors ils disaient : « Si tu apportes des bonbons à l'école, tu ne peux pas tout garder pour toi, tu dois les partager avec d'autres enfants ». En nous éduquant, la société a fait en sorte d'apprendre cet esprit de coopération. Et pourquoi faut-il faire cela ? Parce que les gens ne sont pas totalement coopératifs. C'est un aspect de la nature humaine mais il y en a d'autres. Il y en a beaucoup. Aussi, si vous voulez une société meilleure, vous devez travailler à encourager l'esprit de partage. Vous savez, ça ne sera jamais à 100%. Ça se comprend. Les gens doivent aussi prendre soin d'eux-mêmes. Mais si nous le rendons plus fort nous nous en porterons tous mieux.

De nos jours, selon le gouvernement des États-Unis, les enseignants sont supposés faire exactement

le contraire. « Oh Johnny, tu as apporté un programme à l'école ! Bien, ne le partage pas ! Oh non, le partage c'est mal ! Le partage ça veut dire que tu es un pirate.

Qu'entendent-ils par le mot « pirate » ? Qu'aider son voisin est l'équivalent moral d'une attaque de bateau. *[rires]*.

Que diraient Jésus et Bouddha à ce sujet ?

Prenez vos chefs religieux favoris... Je ne sais pas... Peut-être Manson aurait dit quelque chose de différent. *[rires]*. Qui sait ce que L. Ron Hubbard aurait dit, mais...

QUESTION : - *[inaudible]*

STALLMAN : - Bien sûr, il est mort. Mais il ne l'admettent pas. Quoi ?

QUESTION : - Les autres aussi sont morts. *[rires]*. Charles Manson aussi est mort. *[rires]*. Ils sont morts, Jésus est mort, Bouddha est mort...

STALLMAN : - Oui, c'est vrai. *[rires]*. De ce point de vue Ron Hubbard n'est pas pire que les autres. *[rires]*. De toute façon... *[inaudible]*

QUESTION : - L. Ron Hubbard utilisait du logiciel libre. Cela l'a libéré de ZANU. *[rires]*

STALLMAN : - Bon quoi qu'il en soit, je pense que c'est vraiment la plus importante raison pour laquelle les logiciels doivent être libres. Nous ne pouvons nous permettre de polluer la ressource la plus importante de la société. C'est vrai que ce n'est pas une ressource physique comme l'air propre et l'eau propre. C'est une ressource psycho-sociale, mais c'est tout aussi réel et cela fait une formidable différence pour nos vies.

Les actions que nous menons influencent les pensées des autres. Quand nous clamons alentour : « Ne partagez pas avec les autres », s'ils nous entendent nous avons eu un effet sur la société et pas un bon effet. C'est la liberté Deux, celle d'aider son voisin.

Oh, du coup si vous n'avez pas cette liberté cela ne cause pas seulement un préjudice à cette ressource psycho-sociale, cela porte préjudice de manière matérielle et pratique. Si le programme a un propriétaire et que le propriétaire s'arrange pour que chaque utilisateur doive payer pour s'en servir, certaines personnes diront : « Pas d'importance, je m'en passerai ». Et c'est un gâchis délibéré. Ce qui est intéressant avec les logiciels c'est que ce n'est pas parce que vous avez moins d'utilisateurs que vous devez produire moins. Si moins de gens achètent des voitures, vous fabriquerez moins de voitures. Là il y a une économie. Il y a des ressources à allouer ou non à la fabrication des voitures. Aussi vous pouvez dire qu'avoir un prix pour une voiture est une bonne chose. Cela évite que les gens gaspillent leurs ressources dans l'achat de voitures dont ils n'ont pas vraiment besoin. Mais si chaque nouvelle voiture n'utilisait aucune ressource, on n'aurait aucun intérêt à économiser sur la fabrication des voitures. Bien, pour les objets physiques, comme les voitures, il faudra toujours des ressources pour en faire une de plus. Chaque exemplaire additionnel. Mais pour les logiciels cela n'est pas vrai.

N'importe qui peut faire une copie. Et c'est complètement banal de le faire. Cela ne consomme aucune ressource sauf un tout petit peu d'électricité. Il n'y a rien à économiser, aucune ressource à allouer ailleurs. Vous trouvez souvent des gens qui prennent l'économie pour argument -- les conséquences du raisonnement économique basé sur des prémisses qui ne s'appliquent pas à l'informatique, et qui prétendent les transposer, en supposant que les résultats seront valides -- dans ce cas l'argument n'est basé sur rien de valable dans le domaine du logiciel. C'est très important de voir comment on arrive à une conclusion et de quelles prémisses elle dépend pour voir si elle est valide. Donc, liberté Deux, la liberté d'aider son voisin.

La liberté Trois est celle d'aider à bâtir votre communauté en publiant une version améliorée du logiciel. Les gens me disaient souvent que si le logiciel est gratuit, personne ne sera payé et donc pourquoi travailler ? [...] Aujourd'hui nous pouvons comparer cette théorie avec les faits empiriques et constater que des centaines de gens sont payés pour faire du logiciel libre et plus de 100 000 le font comme volontaires. Il y a plein de gens qui font des logiciels libres pour différents motifs.

Quand j'ai démarré le premier GNU Emacs -- le premier morceau de GNU que les gens ont réellement voulu utiliser -- et quand il a commencé à y avoir des utilisateurs, après un certain temps j'ai eu un

message disant : « Je pense que j'ai vu un bogue dans le source, et voici une solution ». Et j'ai eu un autre message : « Voici du code pour ajouter une nouvelle fonction ». Et une nouvelle correction, et une nouvelle fonction. Et une autre et une autre -- jusqu'à ce qu'elles se déversent sur moi si vite qu'utiliser toute cette aide devenait un vrai travail. Microsoft n'a pas ce problème. *[rires]*

En fin de compte, des gens ont remarqué ce phénomène. Vous voyez, dans les années 80, beaucoup parmi nous pensaient que le logiciel libre ne serait peut-être pas aussi bon que le non-libre, parce que nous n'aurions pas assez d'argent pour payer des gens. Et bien sûr, les gens qui comme moi accordent de la valeur à la communauté et à la liberté ont dit : « Nous utiliserons des logiciels libres tout de même ». Cela vaut le coup de faire quelque sacrifice au niveau technique pour avoir la liberté. Mais ce que les gens ont vu dans les années 90, c'est que nos logiciels étaient en fait meilleurs. Ils étaient plus puissants et plus fiables que les alternatives propriétaires.

Au début des années 90 quelqu'un a trouvé un moyen de mesurer scientifiquement la fiabilité d'un logiciel. Voilà ce qu'il a fait. Il a pris plusieurs logiciels qui faisaient le même travail, exactement le même travail sur différents systèmes. Parce qu'il y avait certains utilitaires basiques de type Unix. Et le travail qu'ils faisaient, nous le savons, imitait plus ou moins la même chose, ou il suivait les spécifications POSIX, donc c'étaient tous les mêmes en terme de travail mais ils étaient écrits et maintenus par des gens différents et développés séparément. Le code était différent. Alors on a dit, bien, faisons marcher ces programmes avec des données aléatoires et mesurons quand ils crashent ou combien de temps ils tiennent. Alors on a mesuré et les programmes les plus fiables étaient les programmes GNU. Toutes les alternatives propriétaires étaient moins fiables. Alors, il l'a publié et l'a dit à tous les développeurs, et quelques années plus tard il a fait les mêmes expériences avec les dernières versions et il a obtenu le même résultat. Les versions GNU étaient plus fiables. Vous savez il y a des cliniques pour le cancer qui utilisent le système GNU parce qu'il est très fiable. Et la fiabilité est très importante pour elles.

Quoi qu'il en soit, il y a même un groupe de gens qui se concentrent sur cet avantage particulier, et la plus importante raison qu'ils en donnent, est le fait que les utilisateurs puissent faire ces diverses choses et avoir ces libertés. Si vous m'avez écouté, vous aurez noté que je parle d'enjeux éthiques et du type de société où nous voulons vivre, ce qui fait une bonne société, autant que des bénéfices matériels. C'est cela le Mouvement pour le Logiciel Libre.

L'autre groupe de gens qui est appelé Mouvement Open Source, ne parle que des bénéfices pratiques. Ils nient que c'est un enjeu de principe. Ils nient que les gens aient un droit à cette liberté de partager avec son prochain et de voir ce que le programme fait, et de le modifier s'il ne leur plaît pas. Ils disent cependant que c'est une chose utile que les gens aient ces droits. Alors ils vont voir des sociétés et leur disent : « Vous savez, vous pourriez faire plus d'argent si vous laissez les gens faire cela ». Aussi vous voyez que jusqu'à un certain point ils mènent les gens dans la même direction mais pour des raisons fondamentalement, philosophiquement, totalement différentes. Parce que sur l'enjeu le plus profond, l'enjeu éthique, les deux mouvements ne sont pas d'accord. Dans le Mouvement pour le Logiciel Libre on dit : « Vous êtes titulaires de ces libertés. Personne ne devrait vous empêcher de faire ces choses ». Dans le mouvement Open Source on dit : « Oui, on peut vous l'interdire mais nous allons essayer de les convaincre de vous laisser faire. » Bien, ils ont apporté leur contribution, ils ont convaincu un certain nombre de sociétés d'apporter des logiciels substantiels dans la communauté des logiciels libres. Aussi le mouvement Open Source a-t-il contribué substantiellement à notre communauté. Et donc nous travaillons ensemble sur des projets pratiques. Mais philosophiquement il y a un désaccord considérable.

Malheureusement, c'est le mouvement Open Source qui reçoit le plus d'aide du monde des affaires et beaucoup d'articles sur notre travail le décrivent comme Open Source et beaucoup de gens pensent innocemment que nous faisons tous partie du mouvement Open Source. C'est pour cela que je mentionne cette distinction, je veux que vous soyez conscients que le mouvement pour le Logiciel Libre qui a amené notre communauté à l'existence et développé le système d'exploitation libre est toujours là et que nous défendons encore cette philosophie éthique. Je tiens à ce que vous soyez au courant de façon à ce que vous ne désinformiez pas quelqu'un d'autre sans le savoir.

Mais c'est aussi pour que vous puissiez vous situer. Vous savez, c'est à vous de voir quel mouvement vous soutenez. Vous serez peut-être d'accord avec le Mouvement pour le Logiciel Libre et avec mes vues. Vous serez peut-être d'accord avec le mouvement Open Source. Vous serez peut être en désaccord avec les

deux. Vous décidez de votre position sur ces enjeux politiques. Mais si vous êtes d'accord avec le mouvement pour le Logiciel Libre, si vous voyez qu'il y a là un enjeu, que les gens dont les vies sont contrôlées et dirigées par cette décision ont aussi leur mot à dire -- alors j'espère que vous direz que vous êtes d'accord avec le mouvement pour le Logiciel libre et une façon de le faire est d'utiliser le terme Logiciel Libre et juste d'aider les gens à savoir qu'il existe.

Donc, la liberté Trois est très importante pratiquement et psycho-sociologiquement. Si vous n'avez pas cette liberté cela amène un préjudice matériel et pratique parce que le développement communautaire n'a pas lieu et nous ne pouvons pas faire des logiciels puissants et fiables. Mais cela amène un préjudice psychosocial qui affecte l'esprit de coopération scientifique -- l'idée que nous travaillons ensemble à l'avancement du savoir humain. Vous savez le progrès scientifique dépend de façon cruciale de la capacité des gens à travailler ensemble. Et pourtant même de nos jours vous trouvez souvent chaque petit groupe de scientifiques en train d'agir comme s'il était en guerre avec un autre gang de scientifiques et d'ingénieurs. Et s'ils ne partagent pas les uns avec les autres ils sont tous freinés.

Ainsi voilà les trois libertés qui distinguent le logiciel libre du logiciel ordinaire. La Liberté Un est celle de s'aider soi-même, d'apporter des changements en fonction de ses besoins propres. La liberté Deux est celle d'aider son prochain en distribuant des copies. Et la Liberté Trois est la liberté d'aider à construire sa communauté en apportant des modifications et en les publiant pour l'usage des autres gens. Si vous avez toutes ces libertés, ce logiciel est libre pour vous. Maintenant pourquoi est-ce que je définis cela en terme d'utilisateur particulier ? Est-ce que c'est du logiciel libre pour *vous* ? Est-ce que c'est du logiciel libre pour *vous* ? Oui ?

QUESTION : - Pouvez-vous expliquer un peu la différence entre les libertés Deux et Trois ?

STALLMAN : - Bien, elles sont certainement liées. Parce que si vous n'avez pas la liberté de redistribuer vous avez encore moins la liberté de distribuer une version modifiée. Mais ce sont des activités différentes.

QUESTION : - Oh.

STALLMAN : - La liberté Deux c'est -- lisez-le, que vous pouvez faire une copie exacte et la donner à vos amis... Ou peut-être vous faites des copies exactes, vous les vendez à des amis pour qu'ils puissent les utiliser. La liberté Trois c'est quand vous apportez des améliorations ou au moins quand vous pensez qu'il y a des améliorations et que quelques personnes sont d'accord avec vous. Voilà c'est cela la différence. Oh, d'ailleurs, un point crucial. Les libertés Un et Trois dépendent de l'accès au code source. Parce que changer un programme binaire c'est extrêmement difficile. *[rires]* Même des changements très insignifiants comme d'utiliser quatre chiffres pour la date. *[rires]* Si vous n'avez pas le source... Aussi pour des raisons pratiques l'accès au code source est une pré-condition, un prérequis du Logiciel Libre.

Bien, pourquoi définir le Logiciel Libre comme Logiciel Libre « pour vous » ? La raison est que le même programme peut être libre pour quelques personnes et non-libres pour d'autres. Cela pourrait sembler paradoxal. Laissez-moi vous donner un exemple de cette situation. Un très grand exemple -- peut-être le plus grand exemple sur ce problème a été le système X-Window qui a été développé au MIT et publié sous une licence qui en a fait un logiciel libre. Si vous aviez la version MIT avec la licence MIT, vous aviez les libertés Un, Deux et Trois. C'était du logiciel libre pour vous.

Mais parmi ceux qui avaient des copies, il y avait divers fabricants d'ordinateurs qui distribuaient des systèmes Unix et ils firent les changements nécessaires pour que X fonctionne sur leurs systèmes. Vous savez, probablement quelques centaines de lignes sur les centaines de milliers d'X. Et quand ils le compilaient ils plaçaient les programmes binaires parmi les autres programmes binaires de leurs systèmes Unix et distribuaient le tout avec la même clause de non-divulgaration. Alors des milliers de gens ont eu ces copies. Ils avaient le système X-Window mais aucune de ces libertés et ce n'était plus du logiciel libre pour eux.

Ainsi, le fait que le système X-Window soit du logiciel libre dépendait de l'endroit où l'on faisait la mesure. Si vous faisiez la mesure à la sortie du groupe de développeurs vous disiez : « J'ai observé toutes ces libertés, c'est du logiciel libre ». Si vous faisiez la mesure parmi les utilisateurs vous disiez : « Mmh, la plupart des utilisateurs n'ont pas ces libertés, ce n'est pas du logiciel libre ». Bon, les gens qui développaient X-Window n'y voyaient pas de problème car leur principal souci était la popularité, l'ego, essentiellement. Ils voulaient un grand succès professionnel. Ils voulaient sentir : « Aah, un tas de gens utilisent nos logiciels ! »

et c'était vrai, un tas de gens utilisaient leurs logiciels, mais ils n'avaient pas la liberté.

Bien, dans le projet GNU, si la même chose arrivait à un logiciel GNU ce serait un échec car notre but n'est pas simplement d'être populaires. Notre but est de donner aux gens la liberté, d'encourager la coopération et de permettre aux gens de coopérer. Rappelez-vous. Ne forcez personne à coopérer mais faites en sorte que chacun puisse coopérer. Tout le monde a le droit de le faire s'il le souhaite. Si des millions de personnes utilisaient des versions non-libres de GNU, ça ne serait pas du tout un succès, l'ensemble aurait été perverti et détourné de son but.

Donc, j'ai cherché un moyen d'empêcher que cela arrive. La méthode que j'ai adoptée est appelée « copyleft ». Ça s'appelle copyleft car c'est comme prendre un copyright et le retourner. [rises]. Légalement le copyleft fonctionne sur la base du copyright. Nous utilisons la loi existante sur le copyright mais nous l'utilisons pour atteindre un but très différent. Voici ce que nous faisons. Nous disons : « Ce programme est copyrighté ». Et bien sûr, par défaut, cela signifie qu'il est interdit de le copier, de le distribuer et de le modifier. Mais alors nous disons : « Vous êtes autorisé à en distribuer des copies, vous êtes autorisé à le modifier, vous êtes autorisé à en distribuer des copies modifiées et étendues. Changez-le comme vous le voulez ».

Mais il y a une condition. Et la condition bien sûr -- la condition est la raison pour laquelle nous nous sommes donnés tout ce mal. La condition dit :

« Chaque fois que vous distribuez quelque chose qui contient un morceau de ce programme, vous devez distribuer le tout aux mêmes conditions. Ni plus, ni moins. » Donc, vous pouvez modifier le programme et le distribuer mais les gens qui l'auront reçu de vous bénéficieront de la même liberté que celle que vous avez reçue de nous. Et pas seulement pour des parties de ce programme, les extraits que vous avez pris, mais aussi pour tous les autres morceaux de votre programme. L'intégralité de ce programme doit être libre pour eux.

Les libertés de redistribuer et de modifier le programme deviennent des droits inaliénables -- un concept hérité de la Déclaration d'indépendance (N.d.T. : américaine). Des droits dont nous nous assurons qu'ils ne peuvent vous être retirés. Et bien sûr, la licence spécifique qui incarne l'idée du copyleft est la GNU General Public License (N.d.T : Licence Publique Générale GNU, GPL). Une licence controversée car elle a la force de dire non à ceux qui voudraient être les parasites de notre communauté.

Il y a beaucoup de gens qui n'apprécient pas nos idéaux de liberté. Et ils seraient très contents de prendre le travail que nous avons fait et d'en faire une base pour la distribution de logiciel non-libres, et d'inciter les gens à abandonner leur liberté. Et le résultat serait, si nous les laissons faire, que nous développerions des programmes libres et nous devrions être constamment concurrencés par des versions améliorées de nos propres programmes. Ça ne serait pas drôle. Et, beaucoup de gens pensent : « Je suis volontaire pour donner mon temps afin de contribuer à ma communauté mais pourquoi contribuer à un programme propriétaire de telle ou telle société. Vous savez certaines personnes ne trouvent pas ça forcément mal mais elles veulent être rétribuées pour cela. Moi, je préférerais ne pas le faire du tout. Mais les deux groupes de gens -- ceux qui comme moi disent : « Je ne veux pas aider un programme non-libre à prendre pied dans notre communauté » -- et ceux qui pensent : « Je veux bien améliorer un programme non libre mais ils ont intérêt à me payer », ont une bonne raison d'utiliser la licence GPL. Parce que cela dit aux sociétés : « Vous ne pouvez pas juste prendre mon travail et le redistribuer sans la liberté. Ce que permettent les licences sans copyleft comme la licence de X-Window.

C'est cela la grande distinction entre les deux catégories de logiciel libre -- au point de vue de la licence. Il y a les programmes copyleftés, de façon à ce que la licence défende la liberté du logiciel pour chaque utilisateur. Et il y a les programmes non copyleftés pour lesquels des versions non-libres sont permises. Quelqu'un peut prendre ces programmes et en ôter la liberté. Vous pouvez obtenir ce programme dans une version non-libre.

Et ce problème existe aujourd'hui. Il y a encore des versions non-libres de X-Window utilisées sur des systèmes libres. Il y a même des matériels qui ne sont supportés que par des versions non-libres. Et c'est un problème majeur dans notre communauté. Cependant, je ne dirais pas que X-Window est une mauvaise chose; vous savez, je dirais que les développeurs n'ont pas fait du mieux qu'il pouvaient. Mais ils ont effectivement mis à disposition beaucoup des logiciels que nous pouvons utiliser.

Vous savez, il y a une grande différence entre imparfait et mauvais. Il y a de nombreux degrés entre le bien et le mal. Nous devons résister à la tentation de dire : « Si vous n'avez pas fait absolument du mieux possible, vous ne valez rien ». Les gens qui ont développé X-Window ont fait une grande contribution à notre communauté, mais il y avait mieux à faire. Ils auraient pu copylefter des morceaux du programme et empêcher ces versions non-libres d'être distribuées par d'autres.

Maintenant le fait que la GPL défende votre liberté -- utilise la loi du copyright pour défendre cette liberté -- est la raison pour laquelle Microsoft l'attaque aujourd'hui. Voyez, Microsoft voudrait réellement prendre tout ce code que nous avons écrit et le mettre dans des programmes propriétaires. Avoir quelqu'un qui ajoute quelques améliorations ou simplement des changements incompatibles c'est tout ce qu'il leur faudrait. *[rires]*.

Vous savez, avec la force de marketing de Microsoft, ils n'ont pas vraiment besoin de faire une version meilleure pour nous supplanter. Ils ont juste besoin de la rendre différente et incompatible. Et alors la mettre sur le bureau de tout le monde. Donc ils n'aiment pas la GPL. Parce que la GPL GNU ne leur permet pas de faire cela. Elle ne leur permet pas la stratégie « extend and embrace (N.d.T. : s'étendre et étreindre) ». Elle dit : Si vous voulez partager du code avec nous vous pouvez. Mais vous devrez aussi partager, et de la même façon. Les changements que vous avez faits devront pouvoir être partagés. Ainsi c'est une coopération dans les deux sens, ce qui est une réelle coopération.

Beaucoup de sociétés, même des grosses sociétés comme IBM et HP veulent utiliser nos logiciels dans cet esprit. IBM et HP contribuent à de substantielles améliorations dans les logiciels GNU. Et ils développent d'autres logiciels libres. Mais Microsoft ne veut pas de cela, et ils prétendent que le business ne peut être compatible avec la GPL. Eh bien si le business n'inclut pas IBM et HP et SUN, peut être qu'ils ont raison. *[rires]*. J'en dirai plus ultérieurement sur ce sujet.

Je dois d'abord terminer l'exposé historique. Vous voyez, nous nous sommes organisés en 84, non seulement pour écrire du logiciel libre, mais en fait pour faire quelque chose de plus cohérent: développer un système d'exploitation libre qui ne comprenne que des logiciels libres. Cela signifiait que nous devions l'écrire morceau par morceau. Bien sûr, nous cherchions des raccourcis, le travail était si grand que les gens pensaient qu'on ne finirait jamais. Je pensais qu'il y avait tout de même une chance que nous finissions mais que ça valait la peine d'essayer des raccourcis. Alors on a continué à chercher; n'y aurait-il pas un programme que quelqu'un avait écrit et que nous pourrions adapter, implanter et que nous n'aurions pas à ré-écrire de zéro ? Par exemple le système X-Window. C'est vrai qu'il n'était pas copylefté, mais il était libre et nous pouvions l'utiliser.

En fait j'ai voulu inclure un système de fenêtres dès le premier jour. J'ai écrit quelques systèmes de fenêtrage quand j'étais au MIT avant de commencer GNU. Et alors, avant même qu'Unix ait un système de fenêtres en 1984 j'ai décidé que GNU en aurait un. Mais je n'ai jamais fini d'écrire le système de fenêtres pour GNU car X-Window est arrivé. Et j'ai dit : « Bono ! ». Un gros travail que nous n'aurons pas à faire. Nous utiliserons X. Donc, j'ai dit basiquement prenons X et mettons-le dans le système GNU. Et nous ferons marcher les autres morceaux de GNU avec X lorsque cela sera approprié. Et nous avons trouvé d'autres logiciels qui avaient été écrits par d'autres personnes comme le formateur de texte TeX. Une bibliothèque de code de Berkeley. En ce temps là il y avait l'Unix de Berkeley, mais ce n'était pas un logiciel libre. Cette bibliothèque venait d'un autre groupe à Berkeley, qui faisait des recherches sur la virgule flottante. Et donc nous avons agencé ces morceaux.

En octobre 85 nous avons fondé la FSF (N.d.T. : Free Software Foundation : Fondation pour le Logiciel Libre). Notez donc que le projet GNU est venu avant. La FSF est venue après. Presque deux ans après l'annonce du projet. Et la FSF est une fondation exonérée d'impôt, qui lève des fonds pour promouvoir la liberté de partager et changer les logiciels. Dans les années 80 une des principales choses que nous avons faites avec nos fonds fut d'employer des gens pour écrire des parties de GNU. Et des programmes essentiels tels que le shell et la librairie C ont été écrits comme cela ainsi que des parties d'autres programmes. Le programme « tar » qui est absolument essentiel -- bien que pas du tout excitant fut écrit comme ça. *[rires]*. Je crois que « grep » a été écrit comme ça aussi, si bien que nous nous sommes approchés du but.

Vers 1991, il n'y avait plus qu'une pièce majeure qui manquait et c'était le noyau. Bon, pourquoi ai-je mis le noyau à part ?

Probablement parce que l'ordre dans lequel vous mettez les choses n'a pas d'importance, au moins techniquement. Il faut tout faire de toute façon. Et en partie parce que nous pensions trouver un début de noyau ailleurs. Ce que nous avons fait. Nous avons trouvé Mach qui était développé à l'université Carnegie-Mellon. Et ce n'était pas tout le noyau mais le socle du noyau, aussi devions-nous écrire la partie supérieure, des choses comme le système de fichiers, le code réseau, etc. Fonctionnant par-dessus Mach comme systèmes utilisateurs ils étaient plus faciles à déboguer. Vous pouvez ainsi déboguer avec un débogueur de niveau source. Et ainsi, de cette façon nous pensions être capables d'avoir cette partie supérieure en peu de temps. Ça n'a pas marché de cette façon. Ces processus asynchrones et multi-threads, s'envoyant des messages les uns aux autres se sont avérés très difficiles à déboguer. Et le système basé sur Mach que nous utilisions possédait un environnement de débogage calamiteux. Il n'était pas fiable et avait divers problèmes. Cela nous a pris des années et des années pour faire fonctionner le noyau GNU. Mais heureusement notre communauté n'a pas eu à attendre le noyau GNU. Parce qu'en 1991 Linus Torvalds développa un autre noyau libre appelé Linux. Et il utilisait le vieux schéma du noyau monolithique et il se trouve qu'il réussit à le faire marcher beaucoup plus vite, que nous le nôtre. C'est sûrement une erreur que j'ai faite : la décision de l'architecture. De toute façon, au début on ne savait rien de Linux, car il ne nous a jamais contacté pour en parler. Bien qu'il ait été au courant du projet GNU. Mais il l'a annoncé à d'autres gens et à d'autres endroits sur le net. Alors d'autres gens ont fait le travail de combiner Linux avec le reste du système GNU pour en faire un système d'exploitation libre complet. Essentiellement pour faire la combinaison GNU+Linux.

Mais ils ne réalisaient pas ce qu'ils faisaient. Vous voyez disaient-ils, nous avons un noyau ; regardons et voyons les morceaux que nous pouvons ajouter. Alors ils ont regardé autour, en-dessous et partout et tout ce dont ils avaient besoin était disponible. Quelle bonne fortune dirent-ils. *[rires]*. Tout est là. Nous pouvons trouver tout ce dont nous avons besoin. Prenons toutes ces différentes choses et mettons-les ensemble et nous aurons un système complet. Ils ne savaient pas que la plus grande partie de ce qu'ils trouvaient c'étaient des parties du système GNU. Alors, ils ne réalisaient pas qu'ils plaçaient Linux dans le créneau manquant du système GNU. Ils pensaient qu'ils prenaient Linux et qu'ils en faisaient un système. Alors ils l'ont appelé système Linux.

QUESTION : - *[inaudible]*

STALLMAN : - Vous entend pas -- Quoi ?

QUESTION : - *[inaudible]*

STALLMAN : - Bien je crois que ce n'est juste pas... C'est provincial, vous savez...

QUESTION : - Mais c'est plus une bonne fortune que de trouver X et Mach ?

STALLMAN : - Oui. La différence c'est que les gens qui ont développé X et Mach, n'avaient pas pour but de faire un système d'exploitation libre complet. Nous sommes les seuls à avoir cela. Et c'est notre travail acharné qui a fait exister le système. Nous avons en réalité fait plus de travail que n'importe quel autre projet. Ce n'est pas une coïncidence, car ces gens -- ils ont écrit des parties utiles du système. Mais ils ne l'ont pas fait parce qu'ils voulaient finir le système. Ils avaient d'autres raisons.

Maintenant les gens qui ont développé X, ils pensaient que concevoir un système de fenêtres sur le réseau serait une bonne chose et cela l'était.

Et cela s'est transformé en un système d'exploitation bon et libre. Mais ce n'est pas ce qu'ils espéraient. Ils n'y pensaient même pas. C'est un accident. Un bénéfique accidentel. Je ne dis pas que ce qu'ils ont fait était mauvais, ils ont fait un grand projet libre. C'est une bonne chose. Mais ils n'avaient pas la vision ultime. C'est le projet GNU qui avait la vision.

Et donc, nous sommes ceux dont -- tous les morceaux n'ont pas été faits par d'autres. Nous les avons fait. Sinon nous n'aurions pas eu un système complet. Même quand ils étaient totalement ennuyeux et pas du tout romantiques comme « tar » et « mv ». *[rires]*. Et nous l'avons fait. Ou « ld », vous trouvez qu'il n'y a rien de très excitant dans « ld », mais j'en ai fait un. *[rires]*. Et j'ai fait des efforts pour qu'il utilise un minimum de d'entrées-sorties sur disque de façon à pouvoir supporter des programmes plus gros et être plus rapide. Mais vous voyez, j'aime bien faire du bon boulot, j'aime bien améliorer différentes choses du programme pendant que je le réalise. Mais la raison pour laquelle je l'ai fait n'est pas que j'avais des idées brillantes pour un meilleur « ld ». La raison était que j'avais besoin d'un « ld » qui soit libre. Et nous ne pouvions attendre de

personne d'autre qu'il le fasse. Aussi nous l'avons fait ou avons trouvé quelqu'un pour le faire. Aussi, bien qu'à ce point des milliers de gens dans des projets aient contribué à ce système, il n'y a qu'un projet qui soit à l'origine de ce système et c'est le projet GNU. C'*est* fondamentalement le système GNU, avec d'autres choses ajoutées depuis.

Donc, quoi qu'il en soit, le fait d'appeler ce système Linux a fait du mal au projet GNU car nous ne sommes pas reconnus normalement pour le travail que nous avons fait. Je pense que Linux, le noyau, est un morceau très utile de logiciel libre et je n'ai que de bonnes choses à en dire.

Bon, en fait, je pourrais trouver quelques mauvaises choses à dire mais... *[rires]* mais à la base j'en dis du bien. Toutefois appeler le système GNU, Linux, est juste une erreur. Je vous demanderai de faire le petit effort nécessaire pour appeler le système GNU/Linux et de cette façon, nous aider à en partager le crédit.

QUESTION : - Vous avez besoin d'une mascotte ! Procurez-vous un animal en fourrure ! *[rires]*

STALLMAN : - Nous en avons un.

QUESTION : - Ah bon ?

STALLMAN : - Nous avons un animal : un gnou. *[rires]* Alors, oui, lorsque vous dessinez un pingouin, dessinez un gnou à côté. *[rires]*. Mais gardons les questions pour la fin.

Je dois encore avancer. Alors pourquoi est-ce que je me soucie de tout cela ? Vous savez pourquoi est-ce que je pense que cela vaut la peine de vous ennuyer et peut-être de vous donner une piètre opinion de moi-même *[rires]* en soulevant la question de la reconnaissance. Parce que certaines personnes quand je parle de ça, certaines personnes pensent que je fais cela pour nourrir mon ego, n'est-ce pas ? Bien sûr, je ne vous demande pas de l'appeler « Stallmanix », n'est ce pas ? *[rires et applaudissements]*

Je vous demande de l'appeler GNU parce que je veux que le projet GNU ait du crédit. Et il y a une raison très spécifique à cela, qui est beaucoup plus importante que le fait d'être reconnu. Parce que vous savez de nos jours il y a des gens dans notre communauté et la plupart des gens qui en parlent ou écrivent à son sujet, ne mentionnent même pas GNU, ni ses buts de liberté, ni d'ailleurs ses idéaux politiques et sociaux. Parce que l'endroit d'où cela vient c'est GNU. Les idées associées à Linux, la philosophie est très différente. C'est basiquement le système apolitique de Linus Torvalds. Ainsi quand les gens pensent que l'ensemble du système est Linux, ils tendent à penser : « Oh tout a dû débiter avec Linus Torvalds. C'est sa philosophie que nous devrions examiner attentivement ». Et quand ils entendent parler de la philosophie GNU ils disent : « Mon dieu, c'est bien idéaliste, ça ne doit pas être pratique. Je suis un utilisateur Linux et pas GNU ». *[rires]*

Quelle ironie ! Si seulement ils savaient ! S'ils savaient que le système qu'ils aiment, et dans certains cas adorent et à la folie, c'est notre philosophie idéaliste devenue réalité. Ce n'est pas qu'ils devraient être d'accord avec nous. Mais au moins ils verraient une raison de la prendre un peu au sérieux, d'y penser attentivement, de lui donner une chance. Ils verraient comme c'est lié à leur vie. Vous savez, s'ils réalisaient : « j'utilise le système GNU. Voici la philosophie GNU. C'est grâce à cette philosophie que le système que j'aime existe ». Ils la considéreraient avec un esprit beaucoup plus ouvert. Ça ne veut pas dire que tout le monde serait d'accord. Les gens pensent des choses différentes. C'est OK. Vous savez, les gens doivent décider eux-mêmes. Mais je veux que cette philosophie soit créditée pour les résultats qu'elle a obtenus.

Si vous regardez dans notre communauté, vous verrez que presque partout les institutions appellent notre système Linux. Vous savez les journalistes l'appellent principalement Linux. Ce n'est pas juste mais ils le font. Les sociétés disent que ça donne une image de marque au système. Oh et la plupart des journalistes quand ils écrivent des articles ne l'envisagent pas comme un sujet politique ou social. Ils l'envisagent comme un problème industriel ou commercial ou quelle société a plus ou moins réussi, ce qui est une question mineure pour la société. Et si vous regardez les sociétés qui « emballent » le système GNU/Linux pour les utilisateurs, la plupart d'entre elles l'appellent Linux et elles y ajoutent *toutes* des logiciels non-libres.

Voyez, la GPL stipule que si vous prenez du code d'un programme GÉPÉLisé, et si vous ajoutez du code pour en faire un programme plus grand, ce programme entier devra être placé sous GPL. Mais vous pourriez mettre d'autres programmes séparés sur le même support (de toutes sortes, disque dur, CD, ...) et ils peuvent être sous d'autres licences. Cela est considéré comme une simple agrégation. Et, pour l'essentiel,

distribuer deux programmes à quelqu'un en même temps, nous n'avons rien à dire là-dessus. Mais, en fait, ce n'est pas vrai. J'aimerais que ça soit vrai, que si une société utilise un programme GÉPÉLisé dans un produit l'ensemble du produit doit être du logiciel libre. Mais ce n'est pas -- ça ne va pas jusque là -- ce n'est pas comme ça. Si deux programmes séparés communiquent l'un avec l'autre à distance, comme en s'envoyant des messages, ils sont légalement séparés en général. Alors les sociétés en ajoutant des logiciels non-libres au système, donnent aux utilisateurs une très mauvaise idée philosophiquement et politiquement. Ils disent aux utilisateurs : « C'est bien d'utiliser des logiciels non-libres. Nous les ajoutons même comme bonus ».

Si vous regardez les magazines sur l'utilisation du système GNU/Linux, la plupart ont un titre comme Linux ceci ou Linux cela. Donc ils appellent le système « Linux », la plupart du temps. Et ils sont remplis de publicités pour des programmes non-libres, que vous pouvez faire fonctionner par-dessus le système GNU/Linux. Ces publicités ont un message commun : « Le logiciel non-libre est bon pour vous. Tellement bon, que vous pourriez même *payer* pour l'avoir ». [*rires*]. Et ils appellent ces choses des packages à valeur ajoutée, ce qui en dit long sur leurs valeurs. Ils disent : « Accordez de la valeur au côté pratique, pas à la liberté ». Et je n'adhère pas à ces valeurs, aussi je les appelle packages à liberté soustraite. [*rires*]. Parce que si vous avez installé un système d'exploitation libre vous vivez maintenant dans un monde libre. Vous bénéficiez de la liberté pour laquelle nous avons travaillé pendant des années à vous la donner. Ces packages vous donnent l'occasion de vous attacher à une chaîne.

Et si vous regardez les réunions commerciales autour du système GNU/Linux, elles s'appellent toutes Linux-expo. Et elles sont remplies de stands exposant des logiciels non-libres, donnant le sceau de l'approbation à du logiciel non-libre. Donc où que vous regardiez dans notre communauté, les institutions renforcent le logiciel non-libre, niant totalement l'idée de liberté pour laquelle GNU a été développé. Et le seul moment où les gens rencontrent l'idée de liberté en relation avec GNU et les logiciels libres c'est justement quand vous utilisez le terme Logiciel Libre. C'est pourquoi je vous demande d'appeler le système « GNU/Linux ». S'il vous plaît, rendez les gens conscients d'où et pourquoi le système est venu.

Bien sûr, en utilisant simplement ce nom vous ne ferez pas une explication historique. Vous pouvez taper quatre lettres supplémentaires et écrire « GNU/Linux ». Vous pouvez dire deux syllabes de plus. Mais GNU/Linux c'est moins de syllabes que Windows2000. [*rires*]. Vous n'en dites pas vraiment beaucoup mais vous les préparez de façon à ce que lorsqu'ils entendent parler de GNU, et de ce qu'il en est, ils voient comment ça se rattache à leur vie. Et ça fait, indirectement, une différence énorme. Alors, aidez-nous.

Vous noterez que Microsoft qualifie la GPL de licence Open Source. Ils ne veulent pas que les gens en parlent en terme de liberté. Ils incitent les gens à penser étroitement, en tant que consommateurs. (Et en plus pas très rationnels, comme consommateurs, s'ils choisissent les produits Microsoft). Mais ils ne veulent pas que les gens pensent en tant que citoyens ou hommes d'état. Pour eux c'est défavorable. En tout cas c'est défavorable pour leur modèle économique.

Bon, comment ça se passe pour le logiciel libre. Bon, je peux vous dire comment le logiciel libre est lié à la société. Un second sujet qui pour vous pourrait être de quelque intérêt, c'est comment il est lié à l'économie. Car, en réalité le Logiciel libre est *terriblement* utile à l'économie. Beaucoup de secteurs utilisent du logiciel, dans les pays avancés. Seule une petite fraction en développe. Et le logiciel libre est terriblement avantageux pour une société qui utilise des logiciels car cela veut dire que c'est vous qui avez le contrôle. À la base le logiciel libre signifie que l'utilisateur contrôle ce que fait le programme. Soit individuellement soit collectivement, si on y prête suffisamment d'attention. N'importe quelle personne qui porte de l'attention peut exercer quelque influence. Si ça ne vous intéresse pas vous n'achetez pas, ensuite vous utilisez ce que les autres préfèrent. Mais si vous êtes intéressés alors vous avez votre mot à dire.

Avec les logiciels propriétaires vous n'avez essentiellement rien à dire. Avec le Logiciel libre vous pouvez changer ce que vous voulez changer. Et peu importe qu'il n'y ait pas de programmeurs dans votre société, ça va quand même. Vous savez si vous voulez bouger les cloisons de votre appartement, vous n'avez pas à être une entreprise de maçonnerie, vous n'avez qu'à trouver un maçon et lui demander de prendre en charge ce travail. Et si vous voulez changer les logiciels que vous utilisez vous n'avez pas besoin d'être une société d'informatique, il vous suffit d'aller dans une société d'informatique et de leur dire : « Combien demandez-vous pour mettre en œuvre ces fonctions ? Et pour quand pouvez-vous le faire ? ». Et s'ils ne font pas le travail vous allez voir quelqu'un d'autre.

Il y a un marché libre pour le service. Alors une entreprise qui s'intéresse au service trouvera un avantage terrible dans le logiciel libre. Dans le logiciel propriétaire, le service est un monopole. Parce qu'une société a le code source, ou peut-être quelques sociétés qui ont payé des sommes énormes, si c'est le shared source de Microsoft. Mais ils sont très peu nombreux. Et donc il n'y a pas mille sources de service pour vous. Et cela veut dire, sauf si vous êtes un géant, qu'ils n'en ont rien à faire de vous. Votre société n'est pas assez importante pour qu'ils tiennent à avoir votre clientèle. Une fois que vous utilisez le programme, vous êtes obligé de passer par eux pour l'assistance, parce que migrer pour un autre logiciel est un travail gigantesque. Alors vous finissez par payer pour avoir le privilège de signaler un bogue. *[rires]*. Et une fois que vous avez payé ils vous disent : « OK, nous avons noté le bogue. Dans quelques mois vous pourrez acheter une mise à jour et vous verrez si nous avons réparé ». *[rires]*

Les sociétés de service dans le logiciel libre ne peuvent pas s'en tirer comme cela. Elles doivent satisfaire les consommateurs. Bien sûr vous pouvez avoir beaucoup d'assistance gratis. Vous postez votre problème sur l'internet. Vous pouvez recevoir une réponse le lendemain. Mais ça n'est bien sûr pas garanti. Si vous voulez être sûr, vous avez intérêt à conclure un accord avec une société et à les payer. Et c'est, bien sûr, une des façons dont l'économie du logiciel libre fonctionne.

Un des autres avantages du logiciel libre c'est la sécurité et la protection de la vie privée. (Et cela s'applique aussi aux individus mais je pose le débat dans le domaine des affaires). Vous voyez, quand un programme est propriétaire on ne peut pas dire ce qu'il fait vraiment. Il pourrait avoir des fonctionnalités délibérément implantées que vous ne voudriez pas savoir là. Comme il pourrait y avoir une « backdoor » (N.d.T. : « porte dérobée ») pour laisser le développeur rentrer sur votre machine. Il peut vous espionner et envoyer des informations quelque part. Ce n'est pas inhabituel. Il y a des programmes Microsoft qui ont fait cela. Mais ce n'est pas seulement Microsoft. Il y a d'autres programmes propriétaires qui espionnent l'utilisateur. Et vous ne pouvez même pas le savoir. Et, bien sûr, même dans le cas où le développeur est totalement honnête, un programmeur peut commettre des erreurs. Il pourrait y avoir des bogues qui affectent votre sécurité, ce qui n'est la faute de personne. Mais l'important c'est que si ce n'est pas du logiciel libre vous ne pouvez trouver les erreurs ni les réparer.

Personne n'a le temps de vérifier le source de chaque programme qu'il utilise. Vous n'allez pas faire cela. Mais, avec les logiciels libres il y a une grande communauté et dans cette communauté il y a des gens qui vérifient les choses. Et vous bénéficiez de leurs vérifications. Parce que s'il y a un bogue accidentel (et il y en a de temps en temps dans n'importe quel programme), ils pourront le trouver et le corriger. Et les gens ne placeront vraisemblablement pas un cheval de Troie ou une fonction d'espionnage, s'ils pensent qu'ils peuvent être découverts. Les développeurs propriétaires pensent qu'ils ne seront pas pris. Que cela passera indétecté. Mais un développeur libre devra penser que les gens regarderont et verront s'il y en a. De même, dans notre communauté nous ne pouvons pas faire avaler une fonction que les utilisateurs n'aimeraient pas. Car nous savons que si les utilisateurs ne l'aiment pas, ils feront une version modifiée, sans cette fonction. Et ils utiliseront tous la version modifiée.

En fait on peut raisonner un peu et se projeter suffisamment pour ne pas créer cette fonction. Après tout, si vous écrivez un programme libre, vous voulez que les gens aiment votre version. Vous ne voulez pas y mettre quelque chose que les gens vont détester, et voir une version modifiée prendre le dessus. Ainsi vous réalisez que l'utilisateur est roi, dans le monde du logiciel libre. Dans le monde du logiciel propriétaire, l'utilisateur n'est **pas** roi. Vous n'êtes qu'un consommateur, vous n'avez pas votre mot à dire sur le logiciel que vous utilisez.

Dans cet aspect, le logiciel libre est un nouveau mécanisme pour faire jouer la démocratie. Le professeur Lessig qui est maintenant à Stanford a remarqué que le code était un peu comme une sorte de loi. Celui qui écrit un code que les gens utilisent pour tel ou tel usage écrit les lois qui régissent la vie des gens. Avec le logiciel libre, ces lois sont écrites d'une façon démocratique. Pas comme la démocratie traditionnelle, il n'y a pas une grande élection où l'on demande : « Qui veut que l'on mette telle ou telle fonction ? ». *[rires]*. À la place nous disons : « que ceux qui veulent travailler à mettre en œuvre telle fonction de telle façon, le fassent ». Et si beaucoup de gens le veulent de telle façon, cela se fait ainsi. De cette façon, tout le monde contribue à la décision sociale simplement en avançant dans la direction où on veut aller.

Et vous êtes, personnellement, libre d'aller aussi loin que vous voulez. Une entreprise est libre d'avancer dans une direction autant qu'elle le veut.

Après, vous additionnez toutes ces choses et cela donne la direction où va le logiciel.

Et c'est souvent utile de prendre un morceau d'un programme existant, en général un gros morceau. Et ensuite d'écrire une certaine quantité de logiciel que voulez ajouter. Et faire en sorte que le programme fasse exactement ce que vous voulez qu'il fasse, ce qui vous aurait coûté les yeux de la tête à développer vous même de zéro si vous n'avez pu cannibaliser un programme libre existant.

Un autre résultat de la puissance de l'utilisateur, c'est que nous tendons à être bons en matière de standardisation et de compatibilité. Parce que les utilisateurs aiment ça ! Les utilisateurs rejettent vraisemblablement un programme qui est délibérément incompatible. Bon, maintenant certains groupes d'utilisateurs ont besoin d'une certaine incompatibilité. Et alors, ils l'obtiennent ; c'est très bien. Mais quand les utilisateurs veulent que vous suiviez un standard, les développeurs doivent le suivre. Et nous savons cela. Nous le faisons. Par contraste, si vous regardez les développeurs propriétaires, ils trouvent souvent avantage à *ne pas* suivre un standard, délibérément. Et pas parce qu'ils pensent que cela donnera un avantage à l'utilisateur, mais plutôt pour s'imposer à lui -- l'enfermer. Et vous en trouverez même qui modifient leurs formats de fichiers de temps en temps juste pour obliger les utilisateurs à se procurer la dernière version.

Les archivistes ont un problème maintenant parce que des fichiers écrits sur des ordinateurs il y a des années ne sont plus accessibles. Ils ont été écrits avec des programmes propriétaires qui sont maintenant perdus. S'ils avaient été écrits avec des logiciels libres, ils pourraient être mis à jour et fonctionner. Et ces choses, ces archives, ne seraient plus inaccessibles. Il y a eu des plaintes à ce sujet récemment qui proposaient la solution du logiciel libre. Et en effet, en utilisant un logiciel propriétaire pour stocker vos données, vous mettez la tête dans un nœud coulant.

Donc j'ai parlé de la façon dont le logiciel libre change le monde des affaires. Mais comment affecte-t-il le domaine plus particulier de l'industrie informatique ?

Bien, la réponse est pratiquement pas. Et la raison en est que 90% de l'industrie informatique (d'après ce que j'entends dire) développe du logiciel sur mesure. Du logiciel qui n'est pas destiné à la diffusion. Pour le logiciel sur mesure la question éthique sur libre ou propriétaire ne se pose pas. Vous voyez la question est : « Est-ce que les utilisateurs peuvent changer et redistribuer le logiciel ? ». S'il n'y a qu'un utilisateur et qu'il a ces droits ce n'est pas un problème. Cet utilisateur a le droit de faire toutes ces choses. Aussi un programme qui est conçu pour un utilisateur pour un usage interne est un logiciel libre, du moins si le code source est accessible avec tous les droits.

Et l'enjeu ne se pose pas pour un logiciel embarqué dans une montre ou un four micro-onde, ou dans le système d'allumage d'une voiture. Parce que ce ne sont pas des endroits où télécharger des logiciels à installer. Ce ne sont pas de vrais ordinateurs pour l'utilisateur. Il n'y a pas un enjeu suffisant. Donc, pour la plus grande part, l'industrie informatique continuera ainsi. La chose importante c'est que comme le plupart des emplois sont dans le sur mesure, même s'il n'y avait pas de possibilités pour une économie du logiciel libre, les développeurs libres pourraient quand même trouver un emploi dans le sur mesure. Il y en a tellement, la proportion est si importante.

Mais, il se trouve qu'il y a une industrie du logiciel libre. Il y a des compagnies de logiciel libre. Et à la conférence de presse que je vais faire, quelques uns d'entre-eux vont se joindre à nous. Et, bien sûr, il y a des sociétés qui ne sont pas des sociétés de logiciel libre mais qui développent du logiciel libre très utile. Et le logiciel libre qu'elles produisent est substantiel.

Mais comment travaille l'industrie du logiciel libre ? Eh bien, certains vendent des copies. On est libre de copier un programme mais ils peuvent quand même vendre des centaines de copies par mois. Et d'autres vendent de l'assistance et des services variés. Personnellement dans les années 80, j'ai vendu de l'assistance sur les logiciels libres. Basiquement, pour 200\$ de l'heure je changeais ce que vous vouliez dans les programmes GNU que j'avais écrits. Et oui, c'était un gros tarif, mais c'était pour des programmes que j'avais écrit et les gens pouvaient penser que j'y passerais moins de temps. [rires]. Et j'ai vécu de cela. En fait, j'ai vécu mieux que jamais. J'ai aussi enseigné. Et j'ai continué jusqu'en 1990 où j'ai obtenu une récompense importante et je n'ai plus eu à le faire.

Mais en 1990, la première entreprise de logiciel libre a été formée, qui est CYGNUS support. Et leur travail était essentiellement le même que le mien.

J'aurais certainement pu travailler pour eux si j'en avais eu besoin. Comme ce n'était pas le cas, j'ai pensé qu'il était bon pour le Mouvement que je reste indépendant. De cette façon je pouvais dire de bonnes et de mauvaises choses sur les différentes entreprises de logiciel libre ou non, sans conflit d'intérêt. Je pensais que cela servirait plus le mouvement. Mais si j'avais dû en vivre j'aurais travaillé pour eux. C'est un travail éthique. Il n'y aurait pas eu de raison d'en avoir honte. Et cette société a été rentable dès les premières années. Elle a été fondée avec très peu de capital. Juste l'argent de ses trois fondateurs. Et elle a grossi chaque année jusqu'à ce qu'ils soient trop cupides, cherchent des investisseurs extérieurs et ça s'est mal passé. Mais il a fallu plusieurs années de succès avant qu'ils ne soient trop gourmands. Cela illustre une des choses intéressantes sur le logiciel libre. Il démontre que vous n'avez pas à lever du capital pour vous développer. Je veux dire, c'est utile, cela peut aider. Si vous levez du capital vous pouvez payer des gens pour écrire un bout de logiciel. Mais vous pouvez faire beaucoup avec peu de gens. Et en fait la formidable efficacité du processus de développement du logiciel libre est une des raisons importantes de passer au logiciel libre. Et cela éclaire ce que dit Microsoft: « la GPL GNU est mauvaise », parce qu'elle leur rend difficile l'appel au capital. Et le fait de prendre des logiciels libres et de les intégrer dans leurs programmes sans les partager. À la base nous n'avons pas besoin qu'ils lèvent du capital. Nous ferons le travail. Nous faisons le travail.

Les gens disaient : « Vous ne pourrez jamais avoir un système d'exploitation libre complet ». Maintenant nous en avons fait beaucoup plus. Et je dirais que nous avons devant nous un champ qui englobe l'ensemble du logiciel d'intérêt général. Et ceci dans un monde où 90% des utilisateurs n'utilisent pas encore nos logiciels libres. Bien que dans certains domaines, plus de la moitié des serveurs web du monde utilisent GNU/Linux avec Apache comme serveur web.

QUESTION : - *[inaudible]* Qu'avez vous dit avant Linux ?...

STALLMAN : - J'ai dit GNU/Linux.

QUESTION : - Ah bon ?

STALLMAN : - Oui, si je parle du noyau, je dis Linux. Vous savez, c'est son nom. Le noyau a été écrit par Linus Torvalds. Et nous devons l'appeler par le nom qu'il a choisi, par respect pour l'auteur. Toutefois, en général dans l'industrie la plupart des utilisateurs ne l'utilisent pas. La plupart des utilisateurs ne l'utilisent pas encore. Aussi lorsqu'ils l'utiliseront, nous devrions avoir automatiquement dix fois plus de volontaires et dix fois plus de clients pour l'industrie du logiciel libre qu'il y aura. Je suis très confiant sur le fait que nous *pouvons* faire le travail.

Et c'est très important, parce que Microsoft nous demande d'être désespérés. Ils disent : « La seule façon d'avoir des logiciels qui marchent, la seule façon d'avoir des innovations, c'est de nous donner le pouvoir. Laissez-nous vous dominer. Laissez-nous contrôler ce que vous pouvez faire avec les programmes que vous utilisez de façon à pouvoir vous pressurer beaucoup d'argent, utiliser une certaine fraction de cet argent pour développer, et le reste comme profit ».

Eh bien nous ne devons pas être aussi désespérés. Il ne faut pas être désespéré au point d'abandonner sa liberté. C'est très dangereux.

Une autre chose, c'est que Microsoft -- en fait, pas que Microsoft -- les gens qui n'encouragent pas le logiciel libre, adoptent un système de valeurs où la seule chose qui compte ce sont les bénéfices à court terme. Combien d'argent ferons nous cette année ? Quel travail puis-je faire aujourd'hui ? Pensée à court terme et pensée étroite. Ils présupposent qu'il est ridicule d'imaginer que quelqu'un puisse faire un sacrifice pour la liberté.

Hier même, un tas de gens faisaient des discours sur les Américains qui faisaient des sacrifices pour la liberté de leurs compatriotes. Certains ont fait de grands sacrifices. Ils ont même sacrifié leur vie pour des sortes de liberté dont chacun a, au moins, entendu parler dans notre pays. (Au moins dans certains cas. Je suppose qu'il faut oublier la guerre du Vietnam). *[note de l'éditeur : le jour précédent était le Memorial Day aux USA. C'est un jour où on commémore les héros des guerres].*

Mais heureusement, garder notre liberté d'utiliser des logiciels n'exige pas de grands sacrifices. Juste de petits sacrifices. Comme apprendre à utiliser la ligne de commande, si on n'a pas encore d'interface graphique. Comme faire le travail de cette façon-ci, parce qu'on n'a pas de logiciel libre pour le faire de cette façon-là. Comme payer une société pour développer tel package logiciel en libre, pour que nous puissions en

disposer dans quelques années. Divers petits sacrifices que nous pouvons tous faire. Et dans le long terme, même *nous* en bénéficierons ! En réalité c'est plus un investissement qu'un sacrifice. Il nous faut seulement voir assez loin pour réaliser qu'il est bon de travailler à l'amélioration de la société, sans compter les centimes et les francs de qui bénéficie de cet investissement.

J'en ai fini avec ce point.

Je voudrais mentionner qu'il existe une autre approche de l'économie du logiciel libre qui a été proposée par Tony Stanco, et qu'il appelle FreeDevelopers (N.d.T. : les développeurs libres). Qui implique une certaine structure économique qui espère éventuellement payer une certaine partie de ses profits à des auteurs de logiciels libres qui ont rejoint cette organisation. Et ils espèrent obtenir de grands contrats de développement en Inde, parce qu'ils vont utiliser des logiciels libres là-bas, ce qui leur fera des économies de coût considérables.

Et donc maintenant, je vais passer aux questions.

QUESTION : - [*inaudible*]

STALLMAN : - Pourriez-vous parler plus fort s'il vous plaît ? Je ne peux vraiment pas vous entendre.

QUESTION : - Comment une société comme Microsoft pourrait-elle inclure des contrats pour du logiciel libre ?

STALLMAN : - Bien, en réalité Microsoft est en train de prévoir de transférer beaucoup de ses activités dans les services. Et ce qu'ils veulent faire est sale et dangereux, car ils veulent lier les services avec les programmes. Dans une sorte de zig-zag, vous voyez. Si bien que pour utiliser ce service, vous devrez utiliser un programme Microsoft, ce qui veut dire que vous utiliserez ce service... Et ainsi de suite. Tout est lié ensemble. C'est leur plan.

La chose intéressante est que vendre ces services n'engage pas la question éthique du logiciel libre ou non libre. Ça pourrait être bien, d'avoir cette activité pour les entreprises qui vendent leurs services sur le net. Mais ce qu'ils essaient d'atteindre c'est de verrouiller encore plus -- d'atteindre un monopole encore plus grand -- sur les logiciels et les services. Et cela a été décrit récemment dans un article de Business Week, je crois. Et d'autres gens ont dit que cela allait transformer le net en « Microsoft Ville ».

Et c'est pertinent car vous savez, le tribunal du procès antitrust contre Microsoft a recommandé de casser en deux la compagnie. Mais d'un côté, ça n'a pas de sens; ça ne ferait rien de bon, ni du côté système d'exploitation, ni du côté applications.

Mais ayant lu cet article, je vois une autre façon, efficace celle-là, de séparer Microsoft, avec d'un côté les services et de l'autre le logiciel. Et de les obliger à garder quelque distance. Que la division services publie ses interfaces, pour que tout le monde puisse écrire un programme client pour ces services. Bien sûr, il faudrait payer pour ces services, mais c'est un enjeu tout à fait différent.

Si Microsoft est divisé en deux de cette façon, « services et logiciels », ils ne pourront pas utiliser leurs logiciels pour empêcher la concurrence avec les services. Et ils ne pourront pas utiliser les services pour fausser la concurrence des logiciels Microsoft. Et nous pourrions faire des logiciels libres, que des gens utiliseront peut-être pour utiliser les services Microsoft, mais cela ne nous dérangera pas.

Parce que bien que Microsoft soit la société de logiciels propriétaires qui a subjugué le plus de monde, les autres en ont subjugué moins, mais ce n'est pas faute d'avoir essayé. [*rires*]. Ils n'ont simplement pas si bien réussi. Donc le problème n'est pas Microsoft et uniquement Microsoft. Microsoft est seulement le plus grand exemple du problème que nous voulons résoudre, qui est que le logiciel propriétaire éloigne les utilisateurs de la liberté de coopérer et de former une société éthique. Aussi, il ne faut pas trop focaliser sur Microsoft. Vous savez, bien qu'ils m'aient donné l'occasion d'être ici, ça ne les rend pas plus importants. Ils ne sont pas tout.

QUESTION : - Plus tôt, vous discutiez les différences entre le logiciel Open Source et le Logiciel Libre. Que pensez-vous de la tendance actuelle des distributions GNU/Linux à se limiter à la plateforme Intel ? Et le fait qu'il semble que de moins en moins de programmeurs programment correctement et fassent

des logiciels qui compilent partout ? Et fassent des logiciels qui fonctionnent seulement sur système Intel.

STALLMAN : - Je ne vois pas là d'enjeu éthique. Bien qu'en fait les sociétés qui fabriquent des ordinateurs réalisent parfois des portages de GNU/Linux. HP a fait cela récemment apparemment. Et ils n'ont pas cherché à porter Windows car cela aurait coûté trop cher. Mais avoir GNU/Linux supporté était l'affaire de cinq ingénieurs pendant quelques mois, je crois. C'était facilement faisable.

Maintenant, bien sûr, j'encourage les gens à utiliser « autoconf », un package GNU qui vous aide à rendre vos programmes portables. Je les y encourage. Ou si quelqu'un corrige le bogue qui empêche de compiler sur votre plateforme, à l'incorporer. Mais je ne vois pas là d'enjeu éthique.

QUESTION : - Deux commentaires. Primo : récemment vous avez parlé au MIT. J'ai lu la transcription. Et quelqu'un vous a interpellé sur les brevets. Et vous avez dit: « les brevets sont un autre enjeu. Je n'ai pas de commentaires sur cela. »

STALLMAN : - Exact. En réalité j'ai beaucoup à dire sur les brevets. Ça prendrait une heure. [*rires*].

QUESTION : - Je voulais dire ceci. Il me semble qu'il y a un enjeu. Il y a une raison pour que les sociétés appellent les brevets et les copyrights quelque chose comme la propriété dure. Elles veulent utiliser le pouvoir de l'état pour leur assurer un monopole. Ce qu'il y a de commun, c'est la motivation des sociétés à obtenir un monopole pour leur intérêt privé.

STALLMAN : - Je comprends. Mais il ne reste pas beaucoup de temps alors tant qu'à faire je voudrais répondre à ça.

Vous avez raison de dire que c'est ce qu'elles veulent. Mais il y a une autre raison pour qu'elles veuillent utiliser le terme « propriété intellectuelle. »

C'est qu'elles ne veulent pas que les gens pensent correctement sur l'enjeu des copyrights et des brevets. Parce que la loi sur les droits d'auteur ou copyrights n'est pas du tout la même que celle sur les brevets. Leurs effets sur les logiciels sont totalement différents.

Les brevets logiciels sont des restrictions faites aux programmeurs, leur interdisant d'écrire certaines sortes de programmes. Tandis que le copyright ne fait pas cela. Avec le copyright au moins si vous l'avez écrit vous pouvez le distribuer. Donc il est terriblement important de séparer ces deux questions. Elles ont un petit quelque chose en commun à un très bas niveau. Et tout le reste est différent. Alors s'il vous plaît pour rendre la discussion plus claire, discutez du copyright ou discutez des brevets. Mais ne parlez pas de « propriété intellectuelle ». J'ai des opinions sur les brevets et sur le copyright. [...]

QUESTION : - Je m'interrogeais sur la musique en ligne. Il y a des similarités et des différences...

STALLMAN : - Bien. Je dirais que la liberté minimum dont nous devrions disposer pour *n'importe quelle* information publiée, est le droit de redistribuer non commercialement, verbatim. Pour des travaux fonctionnels, nous avons besoin de la liberté de redistribuer *commercialement* une version modifiée, parce que c'est terriblement utile à la société. Pour des travaux non fonctionnels, vous savez des choses qui doivent être divertissantes ou esthétiques, ou refléter les vues d'une personne, peut-être qu'ils ne doivent pas être modifiés. Et peut être que cela signifie que c'est OK d'avoir un copyright qui couvre toute distribution commerciale.

Rappelez-vous que selon la Constitution des États-Unis, le but du copyright est le bénéfice du public. Il est de modifier la conduite de certaines personnes privées, de façon à ce qu'elles publient plus de livres. Et le bénéfice est que la société discute et apprenne et, vous savez, nous avons la littérature. Nous avons les écrits scientifiques. Le but est d'encourager cela. Les copyrights n'existent pas pour le bien-être des auteurs, ni celui des éditeurs, il existe pour les lecteurs et tous ceux qui bénéficient de la communication d'information, qui survient quand des gens écrivent et d'autres lisent. Et ce but, je l'approuve !

Mais à l'âge des réseaux informatiques la *méthode* n'est plus appropriée, parce qu'elle exige des lois draconiennes qui envahissent l'intimité de chacun et terrorisent tout le monde. Vous savez, des années de prison pour avoir partagé avec son voisin. Ce n'était pas la même chose du temps de la presse écrite. Le copyright était alors une réglementation industrielle. Il s'appliquait aux éditeurs. Maintenant, c'est une restriction imposée par les éditeurs sur le public. Alors la relation de pouvoir a viré à 180°, même si c'est la

même loi.

QUESTION : - Ainsi on peut avoir la même chose, mais en faisant de la musique à partir d'autre musique.

STALLMAN : - Oui, c'est intéressant.

QUESTION : - Et unique. De nouvelles œuvres, c'est encore beaucoup de coopération.

STALLMAN : - Tout à fait. Et je suppose que cela demande un concept de « fair use » (N.d.T. : littéralement « usage raisonnable », un concept juridique propre au copyright américain). Certainement faire un sample de quelques secondes et l'utiliser pour faire une œuvre musicale, ça doit être un « usage raisonnable ». Même l'idée ordinaire d'« usage raisonnable » renferme cela, si vous y pensez. Je ne sais pas si les tribunaux seraient d'accord mais ils devraient. Ce ne serait pas un vrai changement du système tel qu'il a existé.

QUESTION : - Que pensez-vous de la publication des données *publiques* dans des formats propriétaires ?

STALLMAN : - Bien, ça ne devrait pas se faire. Le gouvernement ne devrait jamais exiger des citoyens d'utiliser un programme non-libre pour y accéder, communiquer avec le gouvernement dans n'importe quel sens.

QUESTION : - Je suis, comment diriez-vous, un utilisateur de GNU/Linux...

STALLMAN : Merci. *[rires]*

QUESTION : - Depuis quatre ans. La seule chose qui m'ait parue problématique pour moi, et qui est quelque chose d'essentiel, je crois pour nous tous, est de surfer sur le Web.

STALLMAN : - Oui.

QUESTION : - Une chose qui a été décidément une faiblesse dans l'usage de GNU/Linux a été de surfer sur le Web, parce que le principal outil pour cela, Netscape.

STALLMAN : - n'est pas un logiciel libre...

Laissez moi répondre à cela. [...] Il y a une tendance terrible chez les utilisateurs de GNU/Linux à utiliser Netscape Navigator sur leur système GNU/Linux. Et en fait les distributions commerciales viennent avec. Voilà bien une situation ironique : nous avons travaillé dur pour faire un système d'exploitation libre, et maintenant si vous allez dans un magasin vous pouvez trouver des versions de GNU/Linux (la plupart d'entre elles appelées Linux) et elles ne sont pas libres. Du moins en partie. Il y a Netscape Navigator et peut être d'autres logiciels non libres. Donc il est très difficile de trouver un système libre, sauf si vous savez ce que vous faites. En fait vous pouvez, il suffit de ne pas installer Netscape Navigator.

En fait, il y a des navigateurs libres depuis de nombreuses années. Il y a un navigateur web libre que j'utilise et qui s'appelle « lynx ». C'est un navigateur libre qui n'est pas graphique. Il est en mode texte. Il a l'extraordinaire avantage de ne pas afficher les publicités. *[rires et applaudissements]*

Mais de toute façon il y a un projet de navigateur graphique libre appelé Mozilla, qui est pratiquement au point. Et je l'utilise à l'occasion.

QUESTION : - Konqueror 2.1 est très bon aussi.

STALLMAN : - Très bien. Voilà donc un autre navigateur graphique libre. Donc nous résolvons finalement ce problème.

QUESTION : - Pouvez-vous me parler de la différence philosophique/éthique entre le Logiciel Libre et Open Source ? Pensez-vous que les deux soient irréconciliables ? *[la fin de la question et le début de la réponse ont sauté au changement de cassettes.]*

STALLMAN : - [...] Mais comme je le disais, dans beaucoup de travail pratique, la politique d'une personne ne compte pas. Quand quelqu'un offre son aide pour un projet GNU nous ne lui disons pas : « Vous devez être d'accord avec notre politique ». Nous disons que dans un package GNU vous devez appeler le

système GNU/Linux, et vous devez l'appeler logiciel libre. Ce que vous dites quand vous ne parlez pas du projet GNU, ça vous regarde.

QUESTION : - IBM a commencé une campagne pour les agences gouvernementales pour vendre leurs nouvelles grosses machines en utilisant Linux pour argument de vente et disant à un moment « Linux ».

STALLMAN : - Oui, bien sûr c'est en réalité le système GNU/Linux. [*rires*]

QUESTION : - C'est vrai. Eh bien le responsable des ventes, il n'y connaît rien à GNU.

STALLMAN : - Je dois dire qui ?

QUESTION : - Le grand responsable des ventes.

STALLMAN : - Oh oui. Le problème c'est qu'ils ont déjà décidé ce qu'ils voulaient dire à leur avantage. Et l'enjeu de savoir comment nommer correctement ou précisément n'est pas l'enjeu premier pour une société comme cela. Dans une petite société il y a un patron. Si le patron est enclin à réfléchir sur des choses comme cela il peut prendre une décision. Mais pas dans une corporation géante. C'est dommage, vous savez.

Il y a une autre chose plus réelle à propos de ce que fait IBM. Ils disent qu'ils mettent un milliard de dollars dans Linux. Mais peut-être faut-il mettre des guillemets autour de dans. Parce qu'une partie de cet argent sert à payer des gens pour faire des logiciels libres. C'est réellement une contribution à notre communauté. Mais une autre partie de cet argent sert à fabriquer du logiciel propriétaire ou à porter des logiciels propriétaires vers GNU/Linux, et ce n'est **pas** une contribution à notre communauté. Mais IBM mélange tout cela. Ce pourrait être en partie de la publicité, ce qui est une contribution même si elle est en partie fautive. Donc c'est une situation compliquée. Une partie de ce qu'ils font est une contribution, une autre partie non et une autre partie est entre les deux. On ne peut pas mettre tout cela ensemble et dire « Whee ! Un milliard de dollars d'IBM! ». [*rires*]. C'est simplifier à outrance!

QUESTION : - Pouvez-vous en dire plus sur la pensée qui sous-tend la licence GNU/GPL ?

STALLMAN : - [...] Donc, la pensée qui sous-tend la licence GNU/GPL ? C'est en partie que je voulais protéger la liberté de la communauté des phénomènes que j'ai décrits à propos de X-Window, et qui sont arrivés avec d'autres logiciels libres. En fait quand j'ai pensé à cet enjeu, X-Window n'était pas encore sorti. Mais j'avais vu le problème se réaliser avec d'autres programmes libres. Par exemple « TeX ». Je voulais être sûr que les utilisateurs auraient tous la liberté. Sinon, j'ai réalisé que je pouvais écrire un programme et que peut être beaucoup de gens utiliseraient ce programme, mais ils n'auraient pas la liberté. Alors à quoi bon ?

Mais l'autre enjeu auquel je pensais c'était : je voulais donner le sentiment à la communauté qu'elle n'était pas un paillason, le sentiment qu'elle ne serait pas la proie du premier parasite venu. Si vous n'utilisez pas le copyleft, vous dites en substance : [*voix mièvre*] « prenez mon code. Faites ce que vous voulez. Je ne dis pas non ». Alors n'importe qui arrive et dit : [*voix très ferme*] « Ah je veux faire une version non-libre de ceci. Je le prends ». Et alors, très probablement ils font quelques améliorations. Ces versions non-libres intéressent les utilisateurs et remplacent les versions libres. Et alors qu'est-ce que vous avez accompli ? Vous avez simplement fait une donation à un projet de logiciel propriétaire.

Et quand les gens voient ce qui est arrivé, quand des gens voient : « les autres prennent et ne donnent jamais », ça peut être démoralisant. Et ce n'est pas une pure spéculation. Je l'ai vu arriver. Ça a participé à la disparition de la vieille communauté dont j'ai fait partie dans les années 70. Certaines personnes sont devenues non-coopératives. On pensait qu'elles en profitaient. Elles agissaient en donnant cette impression. Et on a réalisé qu'elles pouvaient utiliser la coopération et ne pas donner en retour. Nous ne pouvions rien y faire. C'était très décourageant. Nous qui ne suivions pas la tendance nous disputions et n'arrivions pas à trouver une idée pour arrêter cela.

Alors la GPL est conçue pour éviter cela. Elle dit : « Vous êtes le bienvenu pour vous joindre à la communauté et utiliser ce code. Vous pouvez l'utiliser de toutes les façons possibles. Mais si vous sortez une version modifiée, vous devez la sortir pour la communauté en tant que logiciel libre ».

Alors en fait, il y a encore bien des façons pour les gens de profiter de notre travail sans y contribuer,

comme ne pas écrire de logiciels. Bien des gens utilisent GNU/Linux et n'écrivent pas de logiciels. Il n'y a pas d'exigences sur ce que vous devez faire pour nous. Mais si vous faites une certaine chose précise vous devez contribuer. Ça signifie que notre communauté n'est pas un paillason. Et je pense que ça donne aux gens la force de sentir : « Oui, nous ne serons pas piétinés par n'importe qui. Nous tiendrons ».

QUESTION : - Oui, ma question portait sur le logiciel libre mais non copylefté. Puisque tout le monde peut le prendre et en faire du logiciel propriétaire, est-ce que quelqu'un ne peut pas le prendre, faire quelques modifications et le placer sous GPL ?

STALLMAN : - Oui, c'est possible.

QUESTION : - Ça rendrait toutes les copies futures GÉPÉLisées.

STALLMAN : - À partir de cette branche. Mais voici pourquoi nous ne le faisons pas :

QUESTION : - Hmmm ?

STALLMAN : - Nous ne faisons pas cela généralement. Laissez-moi vous expliquer pourquoi. [...] Nous pourrions si nous le voulions, prendre X-Window, faire une copie sous GPL et faire des modifications. Mais il y a un groupe beaucoup plus important de gens qui travaillent à son amélioration et qui ne veulent pas le GÉPÉLiser. Si nous faisons cela nous créerions une branche. Et ce n'est pas très sympa comme traitement. Ils font partie de notre communauté. Ils contribuent à notre communauté.

Deuxièmement, cela se retournerait contre nous, parce qu'ils ont fait beaucoup plus de travail sur X que nous n'en ferions. Notre version serait inférieure à la leur et les gens ne l'utiliseraient pas, alors à quoi bon ?

Alors quand une personne apporte une amélioration à X, je dis à cette personne coopérez avec l'équipe de développement de X-Window. Envoyez leur votre travail et laissez les s'en servir. Ils développent une *très* importante pièce de logiciel libre. C'est bon de coopérer avec eux !

QUESTION : - Sauf, considérant X en particulier, il y a deux ans quand ils étaient très loin dans l'Open Source non-libre...

STALLMAN : - En fait ce n'était même pas Open Source. Vous ne pouviez pas distribuer commercialement, ou quelque chose comme ça. [...] Il y avait des restrictions. On ne pouvait pas distribuer commercialement, je crois. Ou on ne pouvait pas distribuer commercialement une version modifiée. Il y avait une restriction considérée comme inacceptable à la fois par le mouvement Open Source et par la FSF.

Et oui, c'est à cela que mène l'utilisation de licence sans copyleft. En fait, le consortium X avait une politique très rigide. Ils disaient : « Si votre programme est copylefté, nous ne le distribuerons pas du tout. Nous ne le mettrons pas dans notre distribution ». Alors un grand nombre de personnes ont été mis sous pression pour ne pas copylefter. Et le résultat c'était que tout leur logiciel était grand ouvert, ensuite. Et ensuite, après avoir demandé aux gens d'être trop permissifs ils ont dit : « Maintenant nous pouvons mettre des restrictions ». Ça n'était pas très éthique de leur part.

Mais, la situation étant donnée, allons-nous gaspiller des ressources pour maintenir une version GPL de X ? Ça n'aurait pas de sens de faire cela. Il y a tant d'autres choses à faire. Laissons-les faire plutôt. Nous pouvons coopérer avec les développeurs de X.

QUESTION : - Avez-vous un commentaire : GNU est-il une marque déposée ? Et est-il pratique d'inclure cela comme faisant partie de la GPL ?

STALLMAN : - Nous cherchons à déposer GNU comme marque déposée mais ça n'a rien à voir avec la GPL ; c'est une longue histoire pour expliquer cela.

QUESTION : - Vous pourriez exiger que la marque déposée soit incluse dans les programmes GPL.

STALLMAN : - Non, je ne pense pas. Les licences ne couvrent que les programmes individuels. Et quand un programme fait partie du projet GNU personne ne ment sur cela. Mais le nom du système dans son ensemble c'est un autre enjeu. C'est un à côté, cela ne vaut pas la peine d'en discuter plus.

QUESTION : - S'il y avait un bouton qui forçait toutes les sociétés à libérer leurs logiciels,

l'utiliserez-vous ?

STALLMAN : - Bien, je ne l'utiliserais que pour les logiciels publiés. Vous savez, je pense que les gens ont le droit d'écrire des logiciels privés et de les utiliser. Cela inclut les entreprises. C'est une question de vie privée. Et c'est vrai qu'il y a des temps où cela peut être néfaste si vous gardez par devers vous quelque chose qui peut être très utile à l'humanité. Mais c'est une sorte de mal différent, même si cela concerne le même secteur.

Mais oui, je pense que tout logiciel publié devrait être libre. Et rappelez-vous que si ce n'est pas un logiciel libre c'est à cause de l'intervention de l'État. L'État intervient pour faire du non-libre. Il crée des pouvoirs spéciaux pour soutenir les propriétaires de programmes, de telle sorte qu'ils ont la police pour nous empêcher d'utiliser les programmes de certaines façons. J'aimerais certainement mettre un terme à cela.

SCHONBERG : - Les interventions de Richard génèrent invariablement une quantité énorme d'énergie intellectuelle. Je suggère qu'une partie soit dirigée pour utiliser et écrire des logiciels libres. Nous allons nous interrompre rapidement. Je voulais dire que Richard a injecté dans la profession, qui est connue dans le public pour son attitude apolitique, un niveau de discussion morale et politique sans précédent. Et nous lui devons beaucoup pour cela [...]. [*applaudissements*].

STALLMAN : - Vous êtes libres de sortir quand vous voulez. Je ne vous retiens pas prisonniers ici vous savez. [...] Ah, un dernier mot, notre site web : www.gnu.org.

Copyright © 2001, 2005, 2006 Richard Stallman

Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Dernière mise à jour : \$Date: 2008/08/15 13:46:24 \$

Traduction : Xavier Dumont

Révision : [trad- gnu@april.org](mailto:trad-gnu@april.org)

Traductions de cette page

[English](#) [en] [Česky](#) [cs] [Français](#) [fr] [Polski](#) [pl]

Termes prêtant à confusion, que vous devriez éviter

Nous vous recommandons d'éviter un certain nombre de mots et de phrases, soit parce qu'ils sont ambigus, soit parce qu'ils suggèrent des opinions avec lesquelles nous espérons que vous n'êtes pas entièrement d'accord.

Veillez aussi consulter [Catégories de logiciels libres](#).

[Autres textes à lire](#) | « [BSD-style](#) » | « [Biens numériques](#) » | « [Commercial](#) » | « [Compensation](#) » | « [Consommateur](#) » | « [Contenu](#) » | « [Créateur](#) » | « [Disponible gratuitement](#) » | « [Donner des logiciels](#) » | « [Écosystème](#) » | « [Fermé](#) » | « [Freeware \(graticiel\)](#) » | « [Gestion des droits numériques](#) » | « [Gratuitement](#) » | « [Hacker](#) » | « [Industrie logicielle](#) » | « [Informatique de confiance](#) » | « [Lecteur MP3](#) » | « [Logiciel commercial](#) » | « [Marché](#) » | « [Ouvert](#) » | « [PC](#) » | « [Photoshop](#) » | « [Piratage](#) » | « [PowerPoint](#) » | « [Propriété Intellectuelle](#) » | « [Protection](#) » | « [RAND](#) » | « [Système LAMP](#) » | « [Système Linux](#) » | « [Vendeur](#) » | « [Vendre des logiciels](#) » | « [Vol](#) »

« BSD-style »

L'expression « BSD-style license » (n.d.t : licence de type BSD) prête à confusion car elle [mélange des licences qui ont des différences importantes](#). Par exemple, la licence originale BSD avec sa clause publicitaire, est incompatible avec la GPL GNU, mais la licence BSD révisée est compatible avec la GPL.

Pour éviter toute confusion, il est préférable de nommer [la licence spécifique en question](#) et d'éviter le terme vague « de type BSD ».

« Fermé »

En décrivant un logiciel non libre comme « fermé », cela se réfère clairement au terme « open source » (source ouvert). Au sein du Mouvement du logiciel libre, [nous voulons éviter d'être confondus avec le plus récent Mouvement open source](#), aussi, nous évitons soigneusement une utilisation qui encouragerait les gens à nous confondre avec eux. Par conséquent, nous évitons de décrire un logiciel non libre comme « fermé ». Nous le qualifions de « non libre » ou « [propriétaire](#) ».

« Commercial »

N'utilisez pas le mot « commercial » comme synonyme de « non libre ». Cela mélange deux problèmes totalement différents.

Un programme est commercial s'il est développé en tant que partie d'une activité commerciale. Un programme commercial peut être libre ou non libre en fonction de sa licence. De la même manière, un programme développé par une école ou un individu peut être libre et non libre, en fonction de sa licence. Les deux questions, quel type d'entité a développé le programme et quelle liberté est laissée aux utilisateurs, sont indépendantes.

Dans la première décennie du mouvement du logiciel libre, les logiciels libres étaient pratiquement tous non commerciaux ; les composants du système d'exploitation GNU/Linux étaient développés par des individus ou par des organisations non commerciales telles que la FSF et les universités. Mais dans les années 1990, le logiciel commercial libre a commencé à apparaître.

Le logiciel commercial libre est une contribution à notre communauté, donc nous l'encourageons. Mais ceux qui pensent que « commercial » signifie « non libre » sont tentés de penser que cette idée est contradictoire et la rejettent en se basant sur une incompréhension. Faites attention à ne pas utiliser le mot « commercial » de cette manière.

« Compensation »

Parler de « compensation pour les auteurs » en connexion avec le copyright suggère les hypothèses que (1) le copyright existe pour le bien des auteurs et que (2) quand nous lisons quelque chose, l'auteur

travaille pour nous et par conséquent, nous devons à l'auteur de l'argent. La première hypothèse est simplement [fausse](#), et la seconde est plutôt outrageuse.

« Consommateur »

Le terme « consommateur », quand il est utilisé pour parler des utilisateurs d'ordinateurs, apporte des suppositions malheureuses.

La théorie économique utilise les termes « producteur » et « consommateur ». Dans ce contexte, ces mots sont appropriés. Mais en décrivant les utilisateurs de logiciels comme des « consommateurs », cela présuppose qu'ils ont un rôle accessoire. Ce terme les assimile à du bétail qui broute passivement ce que d'autres leur donnent.

Cette façon de penser mène à des travestissements comme le CBDTPA « Consumer Broadband and Digital Television Promotion Act » (Loi pour la promotion de la télévision numérique à large bande pour le consommateur) qui nécessiterait des systèmes de restriction de copie pour tout appareil numérique. Si tout ce que font les utilisateurs est « consommer », alors, pourquoi cela les dérangerait-ils ?

La vision économique obtuse des utilisateurs en tant que « consommateurs » va de pair avec l'idée que les œuvres publiées sont des « contenus ».

Pour décrire les gens qui ne sont pas limités à la consommation passive de leurs ordinateurs, nous suggérons des termes comme « particuliers » et « citoyens ».

« Contenu »

Si vous voulez décrire ce qui est à l'intérieur d'une tasse, vous parlez du « contenu » de cette tasse, mais utiliser ce mot pour décrire des œuvres d'auteur écrites ou autres, vaut la peine d'être évité. Cette utilisation relève d'une attitude spécifique à l'encontre de ces œuvres : elles sont interchangeable et leur but est de remplir des boîtes et de faire de l'argent. En effet, ce terme ne respecte pas les œuvres elles-mêmes.

Ceux qui utilisent ce terme sont souvent les éditeurs qui poussent à obtenir des pouvoirs de copyright accrus aux noms des auteurs (« créateurs », comme ils disent) des œuvres. Le terme « contenu » révèle ce qu'ils pensent vraiment. (Consultez [Courtney Love's open letter to Steve Case](#) — Lettre ouverte de Courtney Love à Steve Case — (recherchez « content provider » dans cette page. Hélas, Mlle Love n'est pas consciente que le terme « propriété intellectuelle » est aussi [trompeur](#)).

Quoi qu'il en soit, aussi longtemps que d'autres personnes utiliseront le terme « fournisseurs de contenu » (content providers), les dissidents politiques peuvent tout aussi bien les appeler les « fournisseurs mécontents » (malcontent providers).

Le terme de « gestion de contenu » remporte le prix de la vacuité. Aucun de ces deux mots n'a de signification spécifique ; « contenu » signifie « un type d'information » et « gestion » dans ce contexte « faire quelque chose avec cette information ». Donc un « système de gestion de contenu » est un système pour faire quelque chose avec un type d'information ».

« Créateur »

Le terme « créateur », appliqué aux auteurs, les compare implicitement à un dieu (« le Créateur »). Le terme est utilisé par les éditeurs pour élever le statut moral des auteurs au-dessus de celui des gens ordinaires, pour justifier des pouvoirs de droit d'auteur accrus que les éditeurs exercent au nom des auteurs. Nous recommandons de dire « auteur » à la place. Cependant, dans beaucoup de cas, « détenteurs de droits d'auteur » est ce que vous voulez réellement dire.

« Biens numériques »

Le terme « biens numériques » tel qu'il est appliqué aux copies d'œuvres d'auteur, les force à rentrer dans le schéma de pensée des biens physiques (qui ne peuvent être copiés, et qui par conséquent doivent être fabriqués et vendus).

« Gestion des droits numériques »

La « gestion des droits numériques » logiciels (Digital Rights Management software) se réfère à des schémas techniques pour imposer des restrictions aux utilisateurs d'ordinateurs. L'utilisation du mot « droits » dans ce terme est de la propagande, conçue pour que vous ne vous rendiez pas compte du problème, du point de vue de la minorité qui imposent les restrictions, tout en ignorant le grand nombre de ceux à qui sont imposées ces restrictions.

Parmi les bonnes alternatives : « Gestion des restrictions numériques » (Digital Restrictions Management), « Gestion des restrictions numériques » (Digital Restrictions Malware) et « Menotticiel » (handcuffware).

« Écosystème »

C'est une erreur de décrire notre communauté (ou toute communauté) comme un « écosystème », car ce mot implique l'absence (1) d'intentions et (2) d'éthique. Dans un écosystème, les espèces évoluent en fonction de leur forme. Si l'une est faible, elle s'éteint, et ce n'est ni bien, ni mal. Le terme « écosystème » suggère implicitement une attitude passive : « Ne demandez pas comment les choses *devraient* être, regardez seulement ce qui leur arrive ».

Par contraste, les êtres vivants qui ont une responsabilité éthique peuvent décider de préserver quelque chose qui, seule, tendrait à s'éteindre : comme la société civile, la démocratie, les droits de l'Homme, la paix, la santé publique, ... ou la liberté des utilisateurs d'ordinateurs.

« Gratuitement »

Si vous voulez dire qu'un programme est un logiciel libre, ne dites pas qu'il est disponible gratuitement. Ce terme signifie « zéro franc ». L'important, dans le logiciel libre, c'est la liberté, pas le prix.

Le logiciel libre est souvent distribué gratuitement, par exemple sur des serveurs FTP. Mais des copies de logiciels libres sont aussi disponibles sur des CD-ROM payants et des copies de logiciels propriétaires peuvent être occasionnellement disponibles gratuitement dans des promotions et certains logiciels propriétaires sont normalement disponibles gratuitement pour certains utilisateurs.

Pour éviter toute confusion, vous pouvez dire que le programme est disponible « librement ».

« Disponible gratuitement »

N'utilisez pas « disponible gratuitement » comme synonyme de « logiciel libre ». Ils ne sont pas équivalents. « Disponible gratuitement » signifie que tout le monde peut facilement obtenir une copie. « Logiciel libre » est défini en termes de la liberté des utilisateurs qui ont une copie. Ce sont des réponses à différentes questions.

« Freeware (graticiel) »

Veuillez ne pas utiliser le terme « freeware » comme un synonyme de « logiciel libre ». Le terme « freeware » était souvent utilisé dans les années 80 pour des programmes distribués uniquement sous leur forme exécutable, sans le code source. Aujourd'hui, ce terme n'a pas vraiment de définition acceptée.

Si vous utilisez d'autres langues que l'anglais, essayez d'éviter également d'employer des mots empruntés à l'anglais tels que « free software » ou « freeware ». Essayez d'utiliser le terme le moins ambigu que [votre langue](#) offre, par exemple.

En utilisant un mot de [votre propre langue](#), vous montrez que vous vous référez bien à la liberté et que vous ne vous contentez pas d'imiter un mystérieux concept de marketing venant de l'étranger. La référence à la liberté peut, à première vue, sembler étrange ou perturbante pour vos compatriotes, mais ils s'y habitueront vite et découvriront ainsi le véritable message véhiculé par le logiciel libre.

« Donner des logiciels »

Utiliser le terme « donner » au lieu de « distribuer un programme comme logiciel libre » induit en erreur. Le problème est le même qu'avec « gratuitement » : cela implique que le problème est le prix, pas la liberté. Une façon d'éviter la confusion est de dire « distribuer en tant que logiciel libre ».

« Hacker »

Un hacker est quelqu'un qui [apprécie l'intelligence enjouée](#)—pas nécessairement avec les ordinateurs. Les programmeurs de l'ancienne communauté du logiciel libre des années 60 et 70 du M.I.T. s'identifiaient eux-mêmes comme des hackers. Aux environs de 1980, des journalistes qui découvraient la communauté des hackers, se sont mépris sur ce terme et l'ont assimilé à « casseurs de sécurité ».

Veillez ne pas répandre cette erreur. Les personnes qui cassent la sécurité sont des « crackers ».

« Propriété Intellectuelle »

Les éditeurs et les juristes aiment assimiler le copyright à la « propriété intellectuelle » — un terme qui inclut aussi les brevets, les marques déposées et d'autres zones plus obscures de la loi. Ces lois ont si peu de choses en commun, et diffèrent tant, qu'il serait mal-avisé de faire des généralisations. Il est beaucoup mieux de parler spécifiquement de « copyright », de « brevets » ou de « marques déposées ».

Le terme « propriété intellectuelle » contient un postulat caché : que la façon naturelle de penser à ces divers problèmes est basée sur une analogie avec les objets physiques, et la façon dont nous les envisageons comme étant notre propriété.

Quand il s'agit de copie, cette analogie néglige la différence cruciale entre des objets matériels et l'information : l'information peut être copiée et partagée presque sans effort, alors que ce n'est pas possible avec les objets matériels.

Même le système américain n'accepte pas complètement cette analogie, dans le sens où il ne traite pas les « copyrights » de la même façon que les droits de propriété sur un objet physique.

Pour éviter la méprise et la confusion de ce terme, il est préférable [de ne pas parler ni même de penser en termes de « propriété intellectuelle »](#).

« Système LAMP »

« LAMP » signifie « Linux, Apache, MySQL et PHP » — une combinaison courante de logiciels à utiliser sur un serveur Web, sauf que « Linux » se réfère vraiment au système GNU/Linux. Donc, au lieu de « LAMP » ce devrait être « GLAMP » : « GNU, Linux, Apache, MySQL and PHP »

« Système Linux »

Linux est le nom du noyau que Linus Torvalds a développé à partir de 1991. Le système d'exploitation dans lequel Linux est utilisé est basiquement GNU avec Linux ajouté. Appeler tout le système « Linux » est à la fois injuste et mène à la confusion. Veuillez appeler le système complet [GNU/Linux](#), pour donner à la le crédit au Projet GNU et pour distinguer le système complet du noyau seul.

« Marché »

Il est trompeur de décrire les utilisateurs de logiciels libres ou les utilisateurs de logiciels en général en terme de « marché ».

Cela ne veut pas dire que nous sommes contre les marchés. Si vous avez une société de services sur les logiciels libres, alors vous avez des clients et vous commercez avec eux dans un marché. Aussi longtemps que vous respectez leur liberté, nous vous souhaitons de réussir dans votre marché.

Mais le mouvement du logiciel libre est un mouvement social, pas une entreprise, et la réussite à

laquelle il se destine n'est pas celle d'un marché. Nous essayons de servir le public en lui donnant la liberté — pas en essayant de les éloigner d'un concurrent. Assimiler la campagne pour la liberté à une campagne commerciale est une diminution de la liberté.

« Lecteur MP3 »

Vers la fin des années 90, il est devenu possible de fabriquer des lecteurs audio numériques, portables et compacts. La plupart supportent le codec breveté MP3, mais pas tous. Certains supportent les codecs audio libres de brevet Ogg Vorbis et FLAC, et ne peuvent même pas lire les fichiers encodés en MP3, précisément pour éviter les brevets. Appeler de tels lecteurs « lecteurs MP3 » ne prête pas seulement à confusion, mais cela met aussi MP3 dans une position privilégiée non méritée qui aide les détenteurs du brevet à continuer à attaquer notre communauté. Nous suggérons les termes « lecteur audio numérique », ou simplement « lecteur audio » si le contexte le permet.

« Ouvert »

Veillez éviter d'utiliser le terme « open » (ouvert) ou « open source » comme synonyme de « logiciel libre ». Ils se réfèrent à une [position différente](#), basée sur des valeurs différentes. Le logiciel libre est un mouvement politique ; l'open source, un modèle de développement. Si vous parlez de la position de l'open source, il est juste que vous utilisiez leur nom, mais veuillez ne pas nous affubler nous ou notre travail avec leur slogan (cela conduit les gens à penser que nous partageons leurs vues).

« PC »

Vous pouvez utiliser l'abréviation « PC » pour vous référer à un certain type d'ordinateur, mais veuillez ne pas l'utiliser avec l'implication que l'ordinateur fonctionne sous Microsoft Windows. Si vous installez GNU/Linux sur le même ordinateur, cela reste un PC.

« Photoshop »

Veillez éviter d'utiliser le terme « Phtoshop » comme un verbe, en voulant parler de manipulation d'image ou d'édition d'image en général. Photoshop n'est que le nom d'un éditeur d'image propriétaire particulier, et il existe beaucoup d'autres alternatives libres, comme [GIMP](#) par exemple.

« Piratage »

Les éditeurs assimilent souvent la copie prohibée à de la piraterie. De cette façon, ils laissent entendre que la copie illégale est moralement identique à l'attaque de bateaux en haute mer, et au rapt ou au meurtre de leurs passagers.

Si vous ne croyez pas que la copie illégale est aussi grave que l'enlèvement ou le meurtre, vous devriez préférer ne pas utiliser le mot « piratage » pour la décrire. Des termes neutres comme « copie prohibée » ou « copie illégale » peuvent le remplacer. Quelques uns d'entre nous préfèrent même utiliser un terme positif comme « partage d'informations avec son voisin ».

« PowerPoint »

Veillez éviter d'utiliser le terme « PowerPoint » pour parler de présentation par diapositives. PowerPoint n'est que le nom d'un programme propriétaire particulier pour faire des présentations, et il existe beaucoup d'autres alternatives libres, comme la classe beamer disponible dans toute distribution (La)TeX.

« Protection »

Les avocats des éditeurs adorent utiliser le terme « protection » pour décrire le copyright. Ce mot implique la prévention de la destruction et de la détérioration. Par conséquent, il encourage les gens à s'identifier au propriétaire et à l'éditeur qui bénéficie du copyright, plutôt qu'aux utilisateurs qui sont limités

par lui.

Il est facile d'éviter le mot « protection » et d'utiliser des termes neutres à la place. Par exemple, au lieu de « La protection du copyright dure très longtemps », vous pouvez dire « Le copyright dure très longtemps ».

Si vous voulez critiquer le copyright au lieu de le soutenir, vous pouvez utiliser le terme « restrictions du copyright ».

Le terme « protection » est aussi utilisé pour décrire des fonctionnalités malicieuses, comme dans « protection contre la copie », une fonctionnalité qui interfère avec la copie. D'un point de vue utilisateur, c'est de l'obstruction. Nous appelons donc cela « obstruction à la copie ».

« RAND (Reasonable And Non-Discriminatory) »

Les organismes de standards qui promulguent des standards restreints par des brevets, qui interdisent le logiciel libre, ont typiquement une politique de licence qui nécessite une somme fixe par copie du programme. Ils qualifient souvent de telles licences avec le terme « RAND », qui signifie « reasonable and non-discriminatory » (raisonnable et non discriminatoire).

Ce terme blanchit une classe de licences de brevets qui ne sont normalement ni raisonnables, ni non-discriminatoires. Il est vrai que ces licences ne sont pas discriminatoires à l'encontre de personnes spécifiques, mais elles le sont à l'encontre de la communauté du logiciel libre, et cela les rend déraisonnables. Ainsi, une moitié de « RAND » est trompeuse, l'autre moitié est partielle.

Les organismes de standards devraient reconnaître que ces licences sont discriminatoires, et laisser tomber l'utilisation du terme « raisonnable et non discriminatoire » ou « RAND » pour les décrire. Jusqu'à ce qu'ils le fassent, les autres écrivains qui ne veulent pas se joindre au blanchiment feraient bien de rejeter ce terme. L'accepter et l'utiliser simplement parce que les sociétés qui exercent leur pouvoir par les brevets l'ont répandu, laisse à ces sociétés le soin de dicter les vues que vous exprimez.

Nous suggérons le terme « uniform fee only » (droit uniforme seulement) ou « UFO » en abrégé, à la place. Il est approprié car la seule condition de ces licences est un droit d'accès uniforme.

« Vendre des logiciels »

Le terme « vendre des logiciels » est ambigu. À strictement parler, échanger une copie d'un programme libre pour une somme d'argent est de la « vente » : mais les gens ont l'habitude d'associer le terme « vendre » aux restrictions du logiciel propriétaire et à l'utilisation subséquente du logiciel. Vous pouvez être plus précis et éviter la confusion en parlant soit de « distribution de copie d'un programme gratuitement », soit d'« imposition de restrictions propriétaires sur l'utilisation d'un programme » en fonction de ce que vous voulez dire.

Consulter [Vendre des logiciels libres](#) pour une discussion plus détaillée sur le sujet.

« Industrie logicielle »

Le terme « industrie logicielle » encourage les gens à penser que le logiciel est toujours développé par une sorte d'usine et qu'il est délivré aux consommateurs. La communauté du logiciel libre montre que ce n'est pas le cas. Les sociétés de logiciels existent, et diverses sociétés développent des logiciels libres et/ou non libres, mais celles qui développent des logiciels libres ne sont pas comme des usines.

Le terme « industrie » a été utilisé comme propagande par les avocats en brevets logiciels. Ils appellent le développement logiciel « industrie » et puis essaient d'argumenter que cela signifie qu'il doit être soumis aux monopoles des brevets. [Le Parlement européen, en rejetant les brevets logiciels en 2003, a voté pour définir « industrie » comme « production automatisée de biens ».](#)

« Vol »

Les partisans du copyright utilisent souvent des mots tels que « volé » et « dérobé » pour décrire les infractions au copyright. En même temps, ils nous demandent de considérer la loi comme une référence au niveau de l'éthique : si la copie est interdite, c'est qu'elle doit être mauvaise.

Alors il est intéressant de signaler que la loi (du moins aux États-Unis) ne considère pas les infractions au droit d'auteur comme du « vol »¹. Les avocats du copyright qui utilisent des termes comme « voler » dénaturent l'autorité à laquelle ils font appel.

L'idée que les lois décident de ce qui est juste et de ce qui est mal est en général fausse. Les lois sont une tentative d'accomplir la justice : dire que les lois définissent la justice ou une éthique est prendre le problème à l'envers.

« Informatique de confiance »

[« Informatique de confiance »](#) est le nom donné à un schéma de conception des ordinateurs qui permet aux développeurs d'applications d'avoir confiance en votre ordinateur pour leur obéir plutôt qu'à vous. De leur point de vue, il s'agit « d'informatique de confiance ». De votre point de vue, il s'agit « d'informatique déloyale ».

« Vendeur »

Veillez ne pas utiliser le terme « vendeur » pour vous référer de manière générale à quiconque développe ou distribue un paquetage logiciel. Beaucoup de programmes sont développés afin de vendre des copies, et leurs développeurs sont par conséquent des vendeurs ; cela inclut même certains paquetages de logiciels libres. Cependant, beaucoup de programmes sont développés par des particuliers volontaires, ou par des organisations, qui n'ont pas l'intention de vendre des copies. Leurs développeurs ne sont pas des vendeurs. De même, seuls certains développeurs de distributions GNU/Linux sont des vendeurs.

Veillez aussi consulter [Catégories de logiciels libres](#).

Note:

1. En France, les manquements au droit d'auteur ne sont pas non plus considérés comme du vol.

*Copyright © 1996, 1997, 1998, 1999, 2001, 2002, 2003, 2004, 2007 Free Software Foundation, Inc.,
51 Franklin St, Fifth Floor, Boston, MA 02110, USA*

La reproduction exacte et la distribution intégrale de cet article est permise sur n'importe quel support d'archivage, pourvu que cette notice soit préservée.

Traduction : Benjamin Drieu.

Révision : trad-gnu@april.org

Dernière mise à jour : 2009/02/16 09:28:15

Traductions de cette page

[Català](#) [ca] [Česky](#) [cs] [English](#) [en] [Español](#) [es] [Français](#) [fr] [Italiano](#) [it] [日本語](#) [ja] [Polski](#) [pl] [português do Brasil](#) [pt-br]

[Română](#) [ro] [Русский](#) [ru] [Српски](#) [sr]

Annexe A - Licence de documentation libre GNU

Licence de documentation libre GNU

Disclaimer

This is an unofficial translation of the GNU Free Documentation License into French. It was not published by the Free Software Foundation, and does not legally state the distribution terms for documentation that uses the GNU FDL--only the original English text of the GNU FDL does that. However, we hope that this translation will help French speakers understand the GNU FDL better.

Ceci est une traduction française non officielle de la Licence de documentation libre GNU. Elle n'a pas été publiée par la Free Software Foundation, et ne fixe pas légalement les conditions de redistribution des documents qui l'utilisent -- seul le texte original en anglais le fait. Nous espérons toutefois que cette traduction aidera les francophones à mieux comprendre la FDL GNU.

Traduction française *non officielle* de la GFDL Version 1.1 (Mars 2000)

Copyright original :

Copyright (C) 2000 Free Software Foundation, inc

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Pour la traduction :

Version 1.0 FR (Jean-Luc Fortin, juillet 2000)

Version 1.1 FR (Christian Casteyde, mars 2001)

Version 1.1.1 FR (César Alexanian, mars 2001)

Version 1.1.2r2 FR (Christian Casteyde et César Alexanian, juin 2001)

Chacun est libre de copier et de distribuer des copies conformes de cette Licence, mais nul n'est autorisé à la modifier.

0 - PRÉAMBULE

L'objet de cette Licence est de rendre tout manuel, livre ou autre document écrit « libre » au sens de la liberté d'utilisation, à savoir : assurer à chacun la liberté effective de le copier ou de le redistribuer, avec ou sans modifications, commercialement ou non. En outre, cette Licence garantit à l'auteur et à l'éditeur la reconnaissance de leur travail, sans qu'ils soient pour autant considérés comme responsables des modifications réalisées par des tiers.

Cette Licence est une sorte de « copyleft », ce qui signifie que les travaux dérivés du document d'origine sont eux-mêmes « libres » selon les mêmes termes. Elle complète la Licence Publique Générale GNU, qui est également une Licence copyleft, conçue pour les logiciels libres.

Nous avons conçu cette Licence pour la documentation des logiciels libres, car les logiciels libres ont besoin d'une documentation elle-même libre : un logiciel libre doit être accompagné d'un manuel garantissant les mêmes libertés que celles accordées par le logiciel lui-même. Mais cette Licence n'est pas limitée aux seuls manuels des logiciels ; elle peut être utilisée pour tous les documents écrits, sans distinction particulière relative au sujet traité ou au mode de publication. Nous recommandons l'usage de cette Licence principalement pour les travaux destinés à des fins d'enseignement ou devant servir de documents de référence.

1 - APPLICABILITÉ ET DÉFINITIONS

Cette Licence couvre tout manuel ou tout autre travail écrit contenant une notice de copyright autorisant la redistribution selon les termes de cette Licence. Le mot « **Document** » se réfère ci-après à un tel manuel ou

travail. Toute personne en est par définition concessionnaire et est référencée ci-après par le terme « **Vous** ».

Une « **Versión modifiée** » du Document désigne tout travail en contenant la totalité ou seulement une portion de celui-ci, copiée mot pour mot, modifiée et/ou traduite dans une autre langue.

Une « **Section secondaire** » désigne une annexe au Document, ou toute information indiquant les rapports entre l'auteur ou l'éditeur et le sujet (ou tout autre sujet connexe) du document, sans toutefois être en rapport direct avec le sujet lui-même (par exemple, si le Document est un manuel de mathématiques, une **Section secondaire** ne traitera d'aucune notion mathématique). Cette section peut contenir des informations relatives à l'historique du Document, des sources documentaires, des dispositions légales, commerciales, philosophiques, ou des positions éthiques ou politiques susceptibles de concerner le sujet traité.

Les « **Sections inaltérables** » sont des sections secondaires considérées comme ne pouvant être modifiées et citées comme telles dans la notice légale qui place le Document sous cette Licence.

Les « **Textes de couverture** » sont les textes courts situés sur les pages de couverture avant et arrière du Document, et cités comme tels dans la mention légale de ce Document.

Le terme « **Copie transparente** » désigne une version numérique du Document représentée dans un format dont les spécifications sont publiquement disponibles et dont le contenu peut être visualisé et édité directement et immédiatement par un éditeur de texte quelconque, ou (pour les images composées de pixels) par un programme de traitement d'images quelconque, ou (pour les dessins) par un éditeur de dessins courant. Ce format doit pouvoir être accepté directement ou être convertible facilement dans des formats utilisables directement par des logiciels de formatage de texte. Une copie publiée dans un quelconque format numérique ouvert mais dont la structure a été conçue dans le but exprès de prévenir les modifications ultérieures du Document ou dans le but d'en décourager les lecteurs n'est pas considérée comme une Copie Transparente. Une copie qui n'est pas « **Transparente** » est considérée, par opposition, comme « **Opaque** ».

Le format de fichier texte codé en ASCII générique et n'utilisant pas de balises, les formats de fichiers Texinfo ou LaTeX, les formats de fichiers SGML ou XML utilisant une DTD publiquement accessible, ainsi que les formats de fichiers HTML simple et standard, écrits de telle sorte qu'ils sont modifiables sans outil spécifique, sont des exemples de formats acceptables pour la réalisation de Copies Transparentes. Les formats suivants sont opaques : PostScript, PDF, formats de fichiers propriétaires qui ne peuvent être visualisés ou édités que par des traitements de textes propriétaires, SGML et XML utilisant des DTD et/ou des outils de formatage qui ne sont pas disponibles publiquement, et du code HTML généré par une machine à l'aide d'un traitement de texte quelconque et dans le seul but de la génération d'un format de sortie.

La « **Page de titre** » désigne, pour les ouvrages imprimés, la page de titre elle-même, ainsi que les pages supplémentaires nécessaires pour fournir clairement les informations dont cette Licence impose la présence sur la page de titre. Pour les travaux n'ayant pas de Page de titre comme décrit ci-dessus, la « **Page de titre** » désigne le texte qui s'apparente le plus au titre du document et situé avant le texte principal.

2 - COPIES CONFORMES

Vous pouvez copier et distribuer le **Document** sur tout type de support, commercialement ou non, à condition que cette Licence, la notice de copyright et la notice de la Licence indiquant que cette Licence s'applique à ce **Document** soient reproduits dans toutes les copies, et que vous n'y ajoutiez aucune condition restrictive supplémentaire. Vous ne pouvez pas utiliser un quelconque moyen technique visant à empêcher ou à contrôler la lecture ou la reproduction ultérieure des copies que vous avez créées ou distribuées. Toutefois, vous pouvez solliciter une rétribution en échange des copies. Si vous distribuez une grande quantité de copies, référez-vous aux dispositions de la section 3.

Vous pouvez également prêter des copies, sous les mêmes conditions que celles suscitées, et vous pouvez afficher publiquement des copies de ce **Document**.

3 - COPIES EN NOMBRE

Si vous publiez des copies imprimées de ce **Document** à plus de 100 exemplaires et que la Licence du

Document indique la présence de **Textes de couverture**, vous devez fournir une couverture pour chaque copie, qui présente les **Textes de couverture** des première et dernière pages de couverture du **Document**. Les première et dernière pages de couverture doivent également vous identifier clairement et sans ambiguïté comme étant l'éditeur de ces copies. La première page de couverture doit comporter le titre du **Document** en mots d'importance et de visibilité égales. Vous pouvez ajouter des informations complémentaires sur les pages de couverture. Les copies du **Document** dont seule la couverture a été modifiée peuvent être considérées comme des copies conformes, à condition que le titre du **Document** soit préservé et que les conditions indiquées précédemment soient respectées.

Si les textes devant se trouver sur la couverture sont trop importants pour y tenir de manière claire, vous pouvez ne placer que les premiers sur la première page et placer les suivants sur les pages consécutives.

Si vous publiez plus de 100 **Copies opaques** du **Document**, vous devez soit fournir une **Copie transparente** pour chaque **Copie opaque**, soit préciser ou fournir avec chaque **Copie opaque** une adresse réseau publiquement accessible d'une **Copie transparente** et complète du **Document**, sans aucun ajout ou modification, et à laquelle tout le monde peut accéder en téléchargement anonyme et sans frais, selon des protocoles réseau communs et standards. Si vous choisissez cette dernière option, vous devez prendre les dispositions nécessaires, dans la limite du raisonnable, afin de garantir l'accès non restrictif à la **Copie transparente** durant une année pleine après la diffusion publique de la dernière **Copie opaque** (directement ou *via* vos revendeurs).

Nous recommandons, mais ce n'est pas obligatoire, que vous contactiez l'auteur du **Document** suffisamment tôt avant toute publication d'un grand nombre de copies, afin de lui permettre de vous donner une version à jour du **Document**.

4 - MODIFICATIONS

Vous pouvez copier et distribuer une **Versión modifiée** du **Document** en respectant les conditions des sections 2 et 3 précédentes, à condition de placer cette **Versión modifiée** sous la présente Licence, dans laquelle le terme « **Document** » doit être remplacé par les termes « **Versión modifiée** », donnant ainsi l'autorisation de redistribuer et de modifier cette **Versión modifiée** à quiconque en possède une copie. De plus, vous devez effectuer les actions suivantes dans la **Versión modifiée** :

1. Utiliser sur la **Page de titre** (et sur la page de couverture éventuellement présente) un titre distinct de celui du **Document** d'origine et de toutes ses versions antérieures (qui, si elles existent, doivent être mentionnées dans la section « **Historique** » du **Document**). Vous pouvez utiliser le même titre si l'éditeur d'origine vous en a donné expressément la permission.
2. Mentionner sur la **Page de titre** en tant qu'auteurs une ou plusieurs des personnes ou entités responsables des modifications de la **Versión modifiée**, avec au moins les cinq principaux auteurs du **Document** (ou tous les auteurs s'il y en a moins de cinq).
3. Préciser sur la **Page de titre** le nom de l'éditeur de la **Versión modifiée**, en tant qu'éditeur du **Document**.
4. Préserver intégralement toutes les notices de copyright du **Document**.
5. Ajouter une notice de copyright adjacente aux autres notices pour vos propres modifications.
6. Inclure immédiatement après les notices de copyright une notice donnant à quiconque l'autorisation d'utiliser la **Versión modifiée** selon les termes de cette Licence, sous la forme présentée dans l'annexe indiquée ci-dessous.
7. Préserver dans cette notice la liste complète des **Sections inaltérables** et les **Textes de couverture** donnés avec la notice de la Licence du **Document**.
8. Inclure une copie non modifiée de cette Licence.
9. Préserver la section nommée « **Historique** » et son titre, et y ajouter une nouvelle entrée décrivant le titre, l'année, les nouveaux auteurs et l'éditeur de la **Versión modifiée**, tels que décrits sur la **Page de titre**, ainsi qu'un descriptif des modifications apportées depuis la précédente version.
10. Conserver l'adresse réseau éventuellement indiquée dans le **Document** permettant à quiconque

- d'accéder à une **Copie transparente** du **Document**, ainsi que les adresses réseau indiquées dans le **Document** pour les versions précédentes sur lesquelles le **Document** se base. Ces liens peuvent être placés dans la section « **Historique** ». Vous pouvez ne pas conserver les liens pour un travail datant de plus de quatre ans avant la version courante ou si l'éditeur d'origine vous en accorde la permission.
11. Si une section « **Dédicaces** » ou une section « **Remerciements** » sont présentes, les informations et les appréciations concernant les contributeurs et les personnes auxquelles s'adressent ces remerciements doivent être conservées, ainsi que le titre de ces sections.
 12. Conserver sans modification les **Sections inaltérables** du **Document**, ni dans leurs textes, ni dans leurs titres. Les numéros de sections ne sont pas considérés comme faisant partie du texte des sections.
 13. Effacer toute section intitulée « **Approbatons** ». Une telle section ne peut pas être incluse dans une **Version modifiée**.
 14. Ne pas renommer une section existante sous le titre « **Approbatons** » ou sous un autre titre entrant en conflit avec le titre d'une **Section inaltérable**.

Si la **Version modifiée** contient de nouvelles sections préliminaires ou de nouvelles annexes considérées comme des **Sections secondaires** et que celles-ci ne contiennent aucun élément copié à partir du Document, vous pouvez à votre convenance en désigner une ou plusieurs comme étant des **Sections inaltérables**. Pour ce faire, ajoutez leurs titres dans la liste des **Sections inaltérables** au sein de la notice de Licence de la version Modifiée. Ces titres doivent être distincts des titres des autres sections.

Vous pouvez ajouter une section nommée « **Approbatons** » à condition que ces approbations ne concernent que les modifications ayant donné naissance à la **Version modifiée** (par exemple, comptes rendus de revue du document ou acceptation du texte par une organisation le reconnaissant comme étant la définition d'un standard).

Vous pouvez ajouter un passage comprenant jusqu'à cinq mots en première page de couverture, et jusqu'à vingt-cinq mots en dernière page de couverture, à la liste des **Textes de couverture** de la **Version modifiée**. Il n'est autorisé d'ajouter qu'un seul passage en première et en dernière pages de couverture par personne ou groupe de personnes ou organisation ayant contribué à la modification du **Document**. Si le Document comporte déjà un passage sur la même couverture, ajouté en votre nom ou au nom de l'organisation au nom de laquelle vous agissez, vous ne pouvez pas ajouter de passage supplémentaire ; mais vous pouvez remplacer un ancien passage si vous avez expressément obtenu l'autorisation de l'éditeur de celui-ci.

Cette Licence ne vous donne pas le droit d'utiliser le nom des auteurs et des éditeurs de ce **Document** à des fins publicitaires ou pour prétendre à l'approbation d'une **Version modifiée**.

5 - FUSION DE DOCUMENTS

Vous pouvez fusionner le **Document** avec d'autres documents soumis à cette Licence, suivant les spécifications de la section 4 pour les **Versions modifiées**, à condition d'inclure dans le document résultant toutes les **Sections inaltérables** des documents originaux sans modification, et de toutes les lister dans la liste des **Sections inaltérables** de la notice de Licence du document résultant de la fusion.

Le document résultant de la fusion n'a besoin que d'une seule copie de cette Licence, et les **Sections inaltérables** existant en multiples exemplaires peuvent être remplacées par une copie unique. S'il existe plusieurs **Sections inaltérables** portant le même nom mais de contenu différent, rendez unique le titre de chaque section en ajoutant, à la fin de celui-ci, entre parenthèses, le nom de l'auteur ou de l'éditeur d'origine, ou, à défaut, un numéro unique. Les mêmes modifications doivent être réalisées dans la liste des **Sections inaltérables** de la notice de Licence du document final.

Dans le document résultant de la fusion, vous devez rassembler en une seule toutes les sections « Historique » des documents d'origine. De même, vous devez rassembler les sections « Remerciements » et « Dédicaces ». Vous devez supprimer toutes les sections « Approbatons ».

6 - REGROUPEMENTS DE DOCUMENTS

Vous pouvez créer un regroupement de documents comprenant le **Document** et d'autres documents soumis à cette Licence, et remplacer les copies individuelles de cette Licence des différents documents par une unique copie incluse dans le regroupement de documents, à condition de respecter pour chacun de ces documents l'ensemble des règles de cette Licence concernant les copies conformes.

Vous pouvez extraire un document d'un tel regroupement et le distribuer individuellement sous couvert de cette Licence, à condition d'y inclure une copie de cette Licence et d'en respecter l'ensemble des règles concernant les copies conformes.

7 - AGRÉGATION AVEC DES TRAVAUX INDÉPENDANTS

La compilation du **Document** ou de ses dérivés avec d'autres documents ou travaux séparés et indépendants sur un support de stockage ou sur un média de distribution quelconque ne représente pas une **Version modifiée** du **Document** tant qu'aucun copyright n'est déposé pour cette compilation. Une telle compilation est appelée « agrégat » et cette Licence ne s'applique pas aux autres travaux indépendants compilés avec le **Document** s'ils ne sont pas eux-mêmes des travaux dérivés du **Document**.

Si les exigences de la section 3 concernant les **Textes de couverture** sont applicables à ces copies du **Document**, et si le **Document** représente un volume inférieur à un quart du volume total de l'agrégat, les **Textes de couverture** du **Document** peuvent être placés sur des pages de couverture qui n'encadrent que le **Document** au sein de l'agrégat. Dans le cas contraire, ils doivent apparaître sur les pages de couverture de l'agrégat complet.

8 - TRADUCTION

La traduction est considérée comme une forme de modification, vous pouvez donc distribuer les traductions du **Document** selon les termes de la section 4. Vous devez obtenir l'autorisation spéciale des auteurs des **Sections inaltérables** pour les remplacer par des traductions, mais vous pouvez inclure les traductions des **Sections inaltérables** en plus des textes originaux. Vous pouvez inclure une traduction de cette Licence à condition d'inclure également la version originale en anglais. En cas de contradiction entre la traduction et la version originale en anglais, c'est cette dernière qui prévaut.

9 - RÉVOCATION

Vous ne pouvez pas copier, modifier, sous-licencier ou distribuer le **Document** autrement que selon les termes de cette Licence. Tout autre acte de copie, modification, sous-Licence ou distribution du **Document** est sans objet et vous prive automatiquement des droits que cette Licence vous accorde. En revanche, les personnes qui ont reçu de votre part des copies ou les droits sur le document sous couvert de cette Licence ne voient pas leurs droits révoqués tant qu'elles en respectent les principes.

10 - RÉVISIONS FUTURES DE CETTE LICENCE

La Free Software Foundation peut publier de temps en temps de nouvelles versions révisées de cette Licence. Ces nouvelles versions seront semblables à la présente version dans l'esprit, mais pourront différer sur des points particuliers en fonction de nouvelles questions ou nouveaux problèmes. Voyez <A HREF="<http://www.gnu.org/copyleft/>"><http://www.gnu.org/copyleft/> pour plus de détails.

Chaque version de cette Licence est dotée d'un numéro de version distinct. Si un **Document** spécifie un numéro de version particulier de cette Licence, et porte la mention « ou toute autre version ultérieure », vous pouvez choisir de suivre les termes de la version spécifiée ou ceux de n'importe quelle version ultérieure publiée par la Free Software Foundation. Si aucun numéro de version n'est spécifié, vous pouvez choisir n'importe quelle version officielle publiée par la Free Software Foundation.

Comment utiliser cette Licence pour vos documents

Pour utiliser cette Licence avec un document que vous avez écrit, incorporez une copie du texte de cette Licence en anglais et placez le texte ci-dessous juste après la page de titre :

Copyright (c) **ANNÉE VOTRE NOM**

Permission vous est donnée de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU Free Documentation License, Version 1.1 ou ultérieure publiée par la Free Software Foundation ; avec les sections inaltérables suivantes :

LISTE DES TITRES DES SECTIONS INALTÉRABLES

Avec le texte de première page de couverture suivant :

TEXTE DE PREMIÈRE PAGE DE COUVERTURE

Avec le texte de dernière page de couverture suivant :

TEXTE DE DERNIÈRE PAGE DE COUVERTURE

Une copie de cette Licence est incluse dans la section appelée **GNU Free Documentation License** de ce document.

Si votre Document ne comporte pas de section inaltérable, de textes de première et dernière pages de couverture, veuillez insérer les mentions suivantes dans les sections adéquates :

- pas de section inaltérable -
- pas de texte de première page de couverture -
- pas de texte de dernière page de couverture -

Vous pouvez également fournir une traduction de la Licence GNU FDL dans votre document, mais celle-ci ne doit pas remplacer la version anglaise. La section intitulée **GNU Free Documentation License** doit contenir la version anglaise de la Licence GNU FDL, c'est la seule qui fait foi.

Si votre Document contient des exemples non triviaux de code programme, nous recommandons de distribuer ces exemples en parallèle sous Licence GNU General Public License, qui permet leur usage dans les logiciels libres.

Annexe B - Texte original de la licence GNU FDL

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

SOMMAIRE

- 0. PREAMBLE
 - 1. APPLICABILITY AND DEFINITIONS
 - 2. VERBATIM COPYING
 - 3. COPYING IN QUANTITY
 - 4. MODIFICATIONS
 - 5. COMBINING DOCUMENTS
 - 6. COLLECTIONS OF DOCUMENTS
 - 7. AGGREGATION WITH INDEPENDENT WORKS
 - 8. TRANSLATION
 - 9. TERMINATION
 - 10. FUTURE REVISIONS OF THIS LICENSE
- External links

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively

with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D.** Preserve all the copyright notices of the Document.
- E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H.** Include an unaltered copy of this License.
- I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it

was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

External links

- [Official GNU FDL webpage](http://www.gnu.org/copyleft/)

Copyright © U.C.H Pour la Liberté
Permission vous est donnée de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU Free Documentation License, Version 1.1 ou ultérieure publiée par la Free Software Foundation.
Une copie de cette Licence est incluse dans la section « GNU Free Documentation License » de ce document.

Pour la Liberté

ᠮᠠᠨᠤᠯᠠᠯᠠᠭᠤᠨ

©2009