

Le danger des brevets logiciels par Richard Stallman

Parmi les nombreux fronts où est engagé le *Libre* il y a celui complexe, épineux et crucial des brevets logiciels (dossier suivi notamment *chez nous* par l'April).

Voici la position de Richard Stallman^[1], donnée lors d'une conférence en Nouvelle-Zélande.



Le danger des brevets logiciels

URL d'origine du document (GNU.org)

*Richard Stallman - version du 24 avril 2012 - Gnu.org - Creative Commons By-Nd
(Traduction : Mathieu Adoutte en collaboration avec Framalang)*

Retranscription de la conférence donnée par Richard M. Stallman le 8 octobre 2009 à l'Université Victoria de Wellington

SF : Je m'appelle Susy Frankel, et je tiens à vous souhaiter la bienvenue, en mon

nom et en celui de Meredith Kolsky Lewis, à cette conférence organisée par le Centre de droit international des affaires de Nouvelle-Zélande. C'est à Brenda Chawner, membre de l'École de gestion de l'information de l'Université de Wellington (plutôt que du Centre que je viens de nommer, qui fait partie de la Faculté de droit) que revient le mérite d'avoir fait revenir Richard Stallman en Nouvelle-Zélande et d'avoir organisé son programme, notamment son étape de ce soir, ici à Wellington. Malheureusement, retenue par ses activités d'enseignement universitaire, elle n'a pu se joindre à nous.

C'est donc à moi que revient le plaisir de vous accueillir à cette conférence sur « Le danger des brevets logiciels ». Richard Stallman propose une série d'interventions sur différents sujets, et si nous avons choisi celui-ci, avec Brenda, c'est parce que pour la première fois dans l'histoire de la Nouvelle-Zélande, nous avons un débat qui tire un peu en longueur, mais qui est important, sur la réforme du droit des brevets, et vous êtes nombreux dans la salle à y participer. Cela nous a donc semblé particulièrement de circonstance. Merci, Richard, de nous avoir fait cette proposition.

On ne présente plus Richard Stallman. Néanmoins, pour ceux d'entre vous qui n'ont jamais entendu parler de lui, il a lancé le développement du système d'exploitation GNU. Je ne savais pas comment prononcer GNU, alors je suis allé sur Youtube (que ferions-nous sans Youtube)...

RMS : Oh, vous ne devriez pas promouvoir YouTube, ils diffusent leurs vidéos dans un format breveté.

SF : C'est vrai. Je l'évoquais seulement pour aborder ce point : doit-on prononcer G N U ou GNU ?

RMS : C'est sur Wikipedia. (GNU se prononce comme « gnou » en français)

SF : Oui, mais sur YouTube j'ai pu vous entendre le prononcer en direct. Le plus important néanmoins, c'est qu'il n'est pas propriétaire. Mais ce qui nous intéresse le plus, c'est que Richard a reçu de nombreuses distinctions pour son travail. Celle que je préfère, et par conséquent la seule que je mentionnerai, est le prix Takeda pour le progrès social et économique, et j'imagine que c'est bien de cela que nous allons parler ce soir. Je vous demande donc de vous joindre à moi pour accueillir Richard.

RMS : Avant tout, je voudrais préciser que l'une des raisons qui me poussent à boire ceci (une canette ou une bouteille de cola qui n'est pas du Coca-Cola) est qu'il y a un boycott mondial de la société Coca-Cola pour avoir assassiné des représentants syndicaux en Colombie. Consultez le site killercoke.org. Il ne s'agit pas des conséquences de la consommation du produit ; après tout, il en va de même pour beaucoup d'autres produits. Il s'agit de meurtre pur et simple. C'est pourquoi, avant d'acheter une boisson, lisez le texte en petits caractères pour vérifier si elle est fabriquée par Coca-Cola.

Je suis connu avant tout pour avoir initié le mouvement du logiciel libre et dirigé le développement du système d'exploitation GNU - bien que la plupart des personnes qui utilisent ce système croient à tort qu'il s'agit de Linux et pensent qu'il a été inventé par quelqu'un d'autre dix ans plus tard. Mais ce n'est pas de cela que je vais parler aujourd'hui. Je suis ici pour vous parler d'une menace juridique qui plane sur tous les développeurs, les utilisateurs et les éditeurs de logiciels : la menace des brevets - sur des idées informatiques, ou des techniques informatiques, c'est-à-dire des idées portant sur quelque chose de réalisable par un ordinateur.

Pour comprendre le problème, il faut tout d'abord prendre conscience du fait que le droit des brevets n'a rien à voir avec le copyright ; ce sont deux champs complètement différents. Vous pouvez être sûr que rien de ce que vous savez sur l'un ne s'applique à l'autre.

Par exemple, chaque fois que quelqu'un utilise le terme « propriété intellectuelle », il renforce la confusion, car ce terme mélange non seulement ces deux branches du droit, mais également une douzaine d'autres. Ces branches sont toutes différentes, et il en résulte que tout énoncé qui porte sur la « propriété intellectuelle » est parfaitement confus - soit parce que l'auteur de l'énoncé est lui-même confus, soit parce qu'il souhaite embrouiller les autres. Mais au final, que ce soit accidentel ou malveillant, cela accroît la confusion.

Vous pouvez échapper à cette confusion en refusant toute affirmation qui utilise ce terme. Pour porter le moindre jugement utile sur ces lois, il faut commencer par bien les distinguer, et les aborder séparément afin de comprendre quels sont réellement les effets de chacune, puis d'en tirer des conclusions. Je vais donc parler de brevets logiciels, et de ce qui se passe dans les pays qui ont autorisé le droit des brevets à restreindre les logiciels.

A quoi sert un brevet ? Un brevet est un monopole explicite sur l'utilisation d'une idée, attribué par un État. Tout brevet contient une partie intitulée les revendications, qui décrit précisément ce que vous n'avez pas le droit de faire (bien qu'elles soient généralement rédigées en des termes que vous ne pouvez probablement pas comprendre). Il faut lutter pour trouver ce que signifient exactement ces interdictions, et il peut y en avoir plusieurs pages en petits caractères.

Un brevet dure habituellement 20 ans, ce qui représente une durée relativement longue dans notre domaine. Il y a 20 ans, le World Wide Web n'existait pas - le secteur qui concentre aujourd'hui l'essentiel des usages de l'ordinateur n'existait pas il y a 20 ans. Et évidemment, si l'on compare avec l'informatique d'il y a 20 ans, tout ce que les gens y font est nouveau, au moins partiellement. Si des brevets avaient été déposés à l'époque, nous n'aurions pas le droit de faire toutes ces choses que nous faisons aujourd'hui ; elles pourraient toutes nous être interdites, dans les pays qui ont été suffisamment stupides pour mettre en place ce système.

La plupart du temps, ceux qui décrivent le fonctionnement du système de brevet sont aussi ceux qui en tirent profit. Qu'il s'agisse d'avocats spécialisés dans le droit des brevets, ou de personnes travaillant pour l'Office des brevets ou pour une mégacorporation, ils veulent vous convaincre que ce système est bon.

The Economist a un jour décrit le système de brevet comme « une loterie chronophage ». Si vous avez déjà vu une publicité pour une loterie, vous comprenez comment ça marche : ils s'attardent sur les infimes chances de gain, et ils passent sous silence l'immense probabilité de perdre. De cette manière, ils donnent systématiquement une représentation faussée de la réalité, sans pour autant mentir explicitement.

Il en va de même pour le discours sur le système des brevets : on vous parle de ce qui se passe lorsque vous avez un brevet dans la poche, et que de temps en temps vous pouvez le mettre sous le nez de quelqu'un en lui disant : « File-moi ton argent. »

Pour rectifier cette distorsion, je vais vous décrire l'envers du décor, le point de vue de la victime - ce qui se passe pour ceux qui veulent développer ou publier ou utiliser un logiciel. Vous vivez dans la crainte qu'un jour quelqu'un vienne et

brandisse un brevet en vous disant : « File-moi ton argent. »

Si vous voulez développer des logiciels dans un pays qui reconnaît les brevets logiciels, et que vous souhaitez respecter le droit des brevets, que devez-vous faire ?

Vous pourriez tenter de faire une liste de toutes les idées qu'on peut trouver dans le programme que vous allez écrire. Mais évidemment, lorsque vous commencez juste à écrire le programme, vous n'avez aucune idée de ce qui va y figurer. Et même après avoir terminé le programme, vous seriez bien incapable de dresser une telle liste.

Cela tient au fait que vous avez conçu le programme avec une approche bien spécifique. Votre approche est déterminée par votre structure mentale, et de ce fait vous n'êtes pas capable de percevoir les structures mentales qui permettraient à d'autres personnes de comprendre le même programme ; il vous manque un œil neuf. Vous avez conçu le programme avec une structure spécifique à l'esprit ; quelqu'un d'autre qui découvre le programme pourrait l'aborder avec une structure différente, reposant sur d'autres idées, et vous n'êtes pas en mesure de dire quelles pourraient être ces idées. Il n'en reste pas moins que ces idées sont mises en œuvre dans votre programme, et que si l'une d'entre elles est brevetée, votre programme peut être interdit.

Imaginons par exemple qu'il existe des brevets sur les idées graphiques, et que vous vouliez dessiner un carré. Il est facile de comprendre que s'il existe un brevet sur les traits horizontaux, vous ne pouvez pas dessiner un carré. Le trait horizontal fait partie des idées mises en œuvre dans votre dessin. Mais vous n'avez peut-être pas pensé au fait que quelqu'un avec un brevet sur les coins pointant vers le bas peut aussi vous poursuivre, parce qu'il peut prendre votre dessin, et le tourner de 45 degrés. Et maintenant, votre carré a un coin vers le bas.

Vous ne pourrez donc jamais faire une liste de toutes les idées qui, si elles étaient brevetées, vous empêcheraient de réaliser votre programme.

Ce que vous pouvez faire, c'est tenter de lister toutes les idées déjà brevetées qui pourraient figurer dans votre programme. En fait non, vous ne pouvez pas, car les demandes de brevet sont gardées secrètes pendant au moins 18 mois, il est donc possible qu'un brevet ait été déposé à l'Office des brevets sans que personne n'en

sache rien. Et il ne s'agit pas d'une hypothèse théorique.

Par exemple, en 1984 a été écrit *compress*, un programme permettant de compresser des fichiers en utilisant l'algorithme de compression de données LZW. À l'époque, il n'y avait pas de brevet sur cet algorithme de compression. L'auteur du programme avait récupéré l'algorithme dans un article de revue. C'était l'époque où nous pensions encore que les revues de recherche informatique servaient à publier des algorithmes pour que tout le monde puisse les utiliser.

Il a écrit ce programme, l'a publié, et en 1985 un brevet a été accordé pour cet algorithme. Le titulaire du brevet a été malin, il n'est pas allé tout de suite voir les gens pour leur dire de cesser d'utiliser le brevet. Il s'est dit : « Laissons-les creuser leur tombe un peu plus. » Quelques années plus tard, il a commencé à menacer les gens. Il est devenu évident que nous ne pouvions plus utiliser *compress*, j'ai donc demandé à tout le monde de proposer d'autres algorithmes que nous pourrions utiliser pour compresser des fichiers.

Quelqu'un m'a écrit pour me dire : « J'ai conçu un autre algorithme de compression qui marche encore mieux, j'ai écrit un programme, j'aimerais t'en faire cadeau. » Une semaine avant de le publier, je tombe sur la rubrique « brevets » du *New-York Times*, que je consulte rarement - je ne la regarde pas plus de deux fois par an - et je peux y lire que quelqu'un a obtenu un brevet pour une « nouvelle méthode de compression des données ». Je me suis donc dit qu'il valait mieux vérifier, et en effet le brevet prohibait le programme que nous nous apprêtions à publier. Mais cela aurait pu être pire : le brevet aurait pu être accordé un an plus tard, ou deux, ou trois, ou cinq.

Toujours est-il que quelqu'un a fini par trouver un algorithme de compression encore meilleur, qui a servi pour le programme *gzip*, et tous ceux qui avaient besoin de compresser des fichiers sont passés à *gzip*. Ça pourrait ressembler à un happy end, mais comme je vous l'expliquerai tout à l'heure, tout n'est pas si parfait.

Il n'est donc pas possible de savoir quels sont les brevets en cours d'examen, même si l'un d'eux peut entraîner l'interdiction de votre travail lorsqu'il sera publié. Mais il est toutefois possible de connaître les brevets déjà attribués. Ils sont tous publiés par l'Office des brevets. Le seul problème, c'est que vous ne

pourrez jamais les lire tous, il y en a beaucoup trop.

Il doit y avoir des centaines de milliers de brevets logiciels aux États-Unis, les suivre tous constituerait une tâche écrasante. Vous allez donc devoir vous restreindre aux brevets pertinents. Et vous allez sans doute trouver de très nombreux brevets pertinents. Mais rien ne dit que vous allez les trouver tous.

Par exemple, dans les années 80 et 90, il existait un brevet sur le « recalcul dans l'ordre naturel » dans les tableurs. Quelqu'un m'en a demandé une copie un jour, j'ai donc regardé dans notre fichier qui liste les numéros de brevets. Puis j'ai ouvert le tiroir correspondant, j'ai récupéré la version papier du brevet, et je l'ai photocopiée pour la lui envoyer. Quand il l'a reçue, il m'a dit : « Je pense que tu t'es trompé de brevet, tu m'as envoyé un truc concernant les compilateurs. » Je me suis dit que peut-être le numéro de brevet était faux. J'ai vérifié, et de manière surprenante il faisait référence à une « méthode pour compiler des formules en code objet ». J'ai commencé à lire, pour voir s'il s'agissait du bon brevet. J'ai lu les revendications, et il s'agissait bien du brevet sur le recalcul dans l'ordre naturel, mais il n'utilisait jamais ce terme. Il n'utilisait jamais le mot « tableur ». En fait, ce que le brevet interdisait, c'était une douzaine de manières différentes d'implémenter un tri topologique - toutes les manières auxquelles ils avaient pu penser. Mais le terme « tri topologique » n'était pas utilisé.

Donc, si vous étiez par exemple en train de développer un tableur, et que vous ayez recherché les brevets pertinents, vous en auriez sans doute trouvé un certain nombre, mais pas celui-ci. Jusqu'à ce qu'un jour, au cours d'une conversation, vous disiez « Oh, je travaille sur un tableur », et que la personne vous réponde « Ah bon ? Tu es au courant que plusieurs autres entreprises qui font des tableurs sont en procès ? » C'est à ce moment-là que vous l'auriez découvert.

Certes, il est impossible de trouver tous les brevets en cherchant, mais on peut en trouver un bon nombre. Encore faut-il comprendre ce qu'ils veulent dire, ce qui n'a rien d'évident, car les brevets sont écrits dans un langage obscur dont il est difficile de saisir le véritable sens. Il va donc falloir passer beaucoup de temps à discuter avec un avocat hors de prix, à expliquer ce que vous voulez faire, afin que l'avocat puisse vous dire si vous avez le droit ou non.

Les détenteurs de brevets eux-mêmes sont bien souvent incapables de savoir ce que recouvrent leurs brevets. Par exemple, un certain Paul Heckel a conçu un

programme pour afficher beaucoup de données sur un petit écran, et en s'appuyant sur une paire d'idées utilisées dans ce programme, il a obtenu deux brevets.

J'ai essayé un jour de trouver une manière simple d'exprimer ce que recouvrait la revendication numéro 1 de l'un de ces brevets. Je me suis rendu compte que je ne pouvais trouver d'autre formulation que celle du brevet lui-même. Or, j'avais beau tenter, il m'était impossible de me mettre cette formulation entièrement en tête.

Heckel non plus n'arrivait pas à suivre. Lorsqu'il a découvert HyperCard, il n'y a vu qu'un programme très différent du sien. Il ne s'est pas rendu compte que la formulation utilisée dans son propre brevet pouvait lui permettre d'interdire HyperCard. Mais ça n'a pas échappé à son avocat. Il a donc menacé Apple de poursuites. Puis il a menacé les clients d'Apple. Et finalement, Apple a conclu un arrangement avec lui, et comme cet arrangement est secret, il nous est impossible de savoir qui a vraiment gagné. Et ceci est juste un exemple parmi d'autres de la difficulté à comprendre ce qu'un brevet interdit ou non.

Lors d'une de mes précédentes conférences sur ce sujet, il se trouve qu'Heckel était présent dans le public. Arrivé à ce point de la conférence, il a bondi en s'écriant : « Tout ceci est faux ! C'est juste que j'ignorais l'étendue de ma protection. » Ce à quoi j'ai répondu : « Oui, c'est exactement ce que j'ai dit. » Il s'est rassis, mais si j'avais répondu non il aurait sans doute trouvé moyen de continuer la dispute.

Toujours est-il que si vous consultez un avocat, à l'issue d'une longue et coûteuse conversation il vous répondra sans doute quelque chose du genre :

Si vous faites quelque chose dans ce domaine-ci, vous êtes presque certain de perdre un procès, si vous faites quelque chose dans ce domaine-là, il y a une chance considérable de perdre un procès, et si vous voulez vraiment vous mettre à l'abri, ne touchez pas à tel et tel domaine. Ceci dit, en cas de procès, l'issue est largement aléatoire.

Alors, maintenant que vous avez des règles claires et prévisibles pour conduire vos affaires, que pouvez-vous vraiment faire ? En définitive, face à un brevet, il n'y a que trois actions possibles : soit vous l'évitez, soit vous obtenez une licence, soit vous le faites annuler. Je vais aborder ces points un par un.

Tout d'abord, il y a la possibilité d'éviter le brevet, ce qui signifie en clair : ne pas mettre en œuvre ce qu'il interdit. Bien sûr, comme il est difficile de dire ce qu'il interdit, il sera presque impossible de déterminer ce qu'il faut faire pour l'éviter.

Il y a quelques années, Kodak a poursuivi Sun pour infraction à l'un de ses brevets relatifs à la programmation orientée objet, et Sun contestait cette infraction. Le tribunal a fini par trancher qu'il y avait bel et bien infraction. Mais lorsqu'on examine le brevet, il est totalement impossible de dire si cette décision est fondée ou non. Personne ne peut vraiment dire ce que le brevet recouvre, mais Sun a quand même dû payer des centaines de millions de dollars pour avoir enfreint une règle complètement incompréhensible.

Parfois, cependant, il est possible de déterminer ce qui est interdit. Et il peut d'agir d'un algorithme.

J'ai ainsi vu un brevet sur quelque chose ressemblant à la « transformée de Fourier rapide », mais deux fois plus rapide. Si la TFR classique est suffisamment rapide pour vos besoins, vous pouvez facilement vous passer de ce brevet. La plupart du temps il n'y aura donc pas de problème. Mais il se peut qu'à l'occasion, vous souhaitiez faire un programme qui utilise constamment des TFR, et qui ne peut fonctionner qu'en utilisant l'algorithme le plus rapide. Là, vous ne pouvez pas l'éviter, ou alors vous pourriez attendre quelques années que les ordinateurs soient plus puissants. Mais bon, l'hypothèse reste exceptionnelle. La plupart du temps il est facile d'esquiver ce type de brevet.

Parfois, en revanche, il est impossible de contourner un brevet sur un algorithme. Prenez par exemple l'algorithme de compression de données LZW. Comme je vous l'ai dit, nous avons fini par trouver un meilleur algorithme de compression, et tous ceux qui veulent compresser des fichiers se sont mis à utiliser le programme *gzip* qui repose sur ce meilleur algorithme. Si vous voulez juste compresser un fichier et le décompresser plus tard, vous pouvez indiquer aux gens qu'ils doivent utiliser tel ou tel programme pour le décompresser ; au final vous pouvez utiliser n'importe quel programme basé sur n'importe quel algorithme, la seule chose qui vous importe c'est qu'il soit efficace

Mais LZW sert aussi à d'autres choses. Par exemple, le langage PostScript fait appel à des opérateurs de compression et de décompression LZW. Il serait inutile de faire appel à un autre algorithme, même plus efficace, car le format de

données ne serait plus le même. Ces algorithmes ne sont pas interopérables. Si vous compressez avec l'algorithme utilisé dans *gzip*, vous ne pouvez pas décompresser avec LZW. En définitive, quel que soit votre algorithme et quelle que soit son efficacité, si ce n'est pas LZW, vous ne pourrez pas implémenter PostScript conformément aux spécifications.

Mais j'ai remarqué que les utilisateurs demandent rarement à leur imprimante de compresser des choses. En général, ils veulent seulement que leur imprimante soit capable de décompresser. Et j'ai aussi remarqué que les deux brevets sur l'algorithme LZW sont écrits de telle manière que si votre système ne fait que décompresser, alors ce n'est pas interdit. Ces brevets ont été formulés de manière à couvrir la compression, et ils contiennent des revendications portant sur la compression et la décompression, mais aucune revendication ne couvre la seule décompression. Je me suis donc rendu compte que si nous implémentions juste la décompression LZW, nous serions à l'abri. Et, bien que cela ne corresponde pas aux spécifications, cela répondrait largement aux besoins des utilisateurs. C'est ainsi que nous nous sommes faufiletés entre les deux brevets.

Il y a aussi le format GIF, pour les images. Il fait également appel à l'algorithme LZW. Il n'a pas fallu longtemps pour que les gens élaborent un autre format de fichier, du nom de PNG, ce qui signifie « PNG N'est pas GIF ». Je crois qu'il utilise l'algorithme *gzip*. Et nous avons commencé à dire à tout le monde : « N'utilisez pas le format GIF, c'est dangereux. Passez à PNG. » Et les utilisateurs nous ont dit : « D'accord, on y pensera, mais pour l'instant les navigateurs ne l'implémentent pas. » Ce à quoi les développeurs de navigateurs ont répondu : « On l'implémentera un jour, mais pour l'instant il n'y a pas de demande forte de la part des utilisateurs. »

Il est facile de comprendre ce qui s'était passé : GIF était devenu un standard de fait. En pratique, demander aux gens d'abandonner leur standard de fait en faveur d'un autre format revient à demander à tout le monde en Nouvelle-Zélande de parler hongrois. Les gens vont vous dire : « Pas de problème, je m'y mets dès que tout le monde le parle. » Et au final nous ne sommes jamais parvenus à convaincre les gens d'arrêter d'utiliser GIF, en dépit du fait que l'un des détenteurs de brevets passe son temps à faire le tour des administrateurs de sites web en les menaçant de poursuites s'ils ne peuvent prouver que tous les GIF du site ont été réalisés avec un programme sous licence.

GIF était donc un piège dangereux menaçant une large part de notre communauté. Nous pensions avoir trouvé une alternative au format GIF, à savoir JPEG, mais quelqu'un - je crois qu'il s'agissait d'une personne ayant tout juste acheté des brevets dans le but d'exercer des menaces - nous a dit : « En regardant mon portefeuille de brevets, j'en ai trouvé un qui couvre le format JPEG. »

JPEG n'était pas un standard de fait. Il s'agissait d'un standard officiel, élaboré par un organisme de standardisation. Et cet organisme avait lui aussi un avocat, qui a dit qu'il ne pensait pas que le brevet couvre le format JPEG.

Alors, qui a raison ? Eh bien, le titulaire du brevet a poursuivi un groupe de sociétés, et s'il y avait eu une décision, on aurait pu savoir ce qu'il en était. Mais je n'ai jamais entendu parler de décision, et je ne suis pas sûr qu'il y en ait jamais eu. Je pense qu'ils ont conclu un règlement amiable, et ce règlement est probablement secret, ce qui signifie que nous ne saurons jamais qui avait raison.

Les cas dont j'ai parlé ici sont relativement modestes : un seul brevet en ce qui concerne JPEG, deux pour l'algorithme LZW utilisé pour GIF. Vous vous demandez peut-être pourquoi il y avait deux brevets sur le même algorithme. Normalement, cela n'est pas possible, pourtant c'est arrivé. Cela tient au fait que les personnes examinant les brevets n'ont pas le temps d'étudier et de comparer tout ce qui devrait l'être, parce que les délais sont très courts. Et comme les algorithmes ne sont au final que des mathématiques, il est impossible de limiter le choix de brevets et de demandes de brevets à comparer.

Vous voyez, dans le domaine industriel, ils peuvent se référer à ce qui se passe réellement dans le monde physique pour circonscrire les choses. Par exemple dans le domaine de l'ingénierie chimique, on peut se demander : « Quelles sont les substances utilisées ? Quelles sont les produits obtenus ? » Si deux demandes de brevet diffèrent sur ce point, il s'agit donc de deux inventions différentes, et il n'y a pas de problème. Mais en mathématiques, des choses identiques peuvent être présentées de manière très différentes, et à moins de les comparer en détail vous ne vous rendrez jamais compte qu'il s'agit d'une seule et même chose. Et, de ce fait, il n'est pas rare de voir la même chose être brevetée à plusieurs reprises dans le domaine du logiciel.

Vous vous souvenez de l'histoire de ce programme qui a été tué par un brevet

avant même que nous ne l'ayons publié ? Eh bien, cet algorithme était également breveté deux fois. Donc, dans un domaine extrêmement réduit, nous avons déjà vu ce phénomène se produire à plusieurs reprises. Je pense vous avoir expliqué pourquoi.

Mais un ou deux brevets constituent une situation plutôt simple. Prenons MPEG-2, le format vidéo. J'ai vu une liste de plus de 70 brevets couvrant ce format, et les négociations pour permettre à quelqu'un d'obtenir une licence ont pris plus de temps que le développement du standard lui-même. Le comité JPEG voulait développer un nouveau standard plus moderne, ils ont fini par abandonner. Ce n'est pas faisable, ont-ils dit. Il y a trop de brevets.

Parfois, c'est carrément une fonctionnalité qui est brevetée, et la seule façon d'éviter le brevet consiste à ne pas implémenter cette fonctionnalité. Ainsi, les utilisateurs du logiciel de traitement de texte XyWrite ont une fois reçu par la poste une mise à jour supprimant une fonctionnalité. Il s'agissait de la possibilité de définir une liste d'abréviations. Par exemple, si vous définissiez qu'« exp » était l'abréviation d'« expérience », il suffisait de taper « exp-espace », ou « exp-virgule », pour qu'« exp » soit automatiquement remplacé par « expérience ».

Mais quelqu'un disposant d'un brevet sur cette fonctionnalité les a menacés, et ils ont conclu que la seule solution consistait à retirer cette fonctionnalité. Et ils ont envoyé à tous les utilisateurs une mise à jour la retirant.

Mais ils m'ont également contacté, car mon éditeur de texte Emacs proposait ce type de fonctionnalité depuis la fin des années 70. Et comme elle était décrite dans le manuel d'Emacs, ils espéraient que je pourrais les aider à faire tomber ce brevet. Je suis heureux de savoir que j'ai eu au moins une fois dans ma vie une idée qui soit brevetable, je regrette juste que ce soit quelqu'un d'autre qui l'ait brevetée.

Par chance, ce brevet fut finalement invalidé, en partie grâce au fait que j'avais publiquement annoncé un usage antérieur de cette fonctionnalité. Mais en attendant, ils ont dû retirer la fonctionnalité.

Retirer une ou deux fonctionnalités, ce n'est pas une catastrophe. Mais quand il s'agit de 50 fonctionnalités, les gens vont finir par dire : « Ce programme n'est pas bon, il lui manque toutes les fonctionnalités qui m'intéressent. » Ce n'est donc pas une solution viable. Et parfois un brevet est si général qu'il balaie un champ

entier, comme par exemple le brevet sur le chiffrement par clé publique, qui interdit de fait tout chiffrement par clé publique pendant 10 ans.

Voilà pour la possibilité d'esquiver un brevet. C'est souvent possible, mais pas toujours, et il y a une limite au nombre de brevets qu'il est possible d'éviter.

Qu'en est-il de la deuxième solution, obtenir une licence pour le brevet ?

Le titulaire du brevet n'est pas tenu de vous accorder une licence. Cela ne dépend que de lui. Il peut très bien dire : « Je veux juste vous couler. » Une fois, j'ai reçu une lettre de quelqu'un dont l'entreprise familiale avait pour activité la création de jeux de casino, qui étaient bien sûr informatisés, et il avait été menacé par un titulaire de brevet qui voulait lui faire mettre la clé sous la porte. Il m'a envoyé le brevet. La revendication numéro 1 ressemblait à quelque chose du genre « un réseau de plusieurs ordinateurs contenant chacun plusieurs jeux et qui permet plusieurs parties simultanément ».

Je suis à peu près sûr que dans les années 80, il existait une université avec des stations de travail en réseau, dotées d'un quelconque système de fenêtrage. Il aurait suffi qu'ils installent plusieurs jeux pour qu'il soit possible d'afficher plusieurs parties simultanément. Il s'agit de quelque chose de tellement trivial et d'inintéressant que jamais personne ne se serait donné la peine d'écrire un article à ce sujet. Ce n'était pas assez intéressant pour constituer un article, mais ça l'était suffisamment pour constituer un brevet. Si vous avez compris que vous pouvez obtenir un monopole sur cette opération triviale, vous pouvez faire fermer vos concurrents.

Mais pourquoi est-ce que l'Office des brevets accorde tant de brevets qui nous semblent absurdes et triviaux ?

Ce n'est pas parce que les examinateurs de brevets sont stupides. C'est parce qu'ils suivent un processus, que ce processus est régi par des règles, et que ces règles aboutissent à ce résultat.

Vous voyez, si quelqu'un a construit une machine qui fait quelque chose une seule fois, et que quelqu'un d'autre construit une machine qui fait la même chose, mais N fois, pour nous il s'agit juste d'une boucle `for`, mais pour l'Office des brevets il s'agit d'une invention. S'il y a des machines qui peuvent faire A, et des machines qui peuvent faire B, et que quelqu'un conçoit une machine qui peut faire A ou B,

pour nous il s'agit d'une condition *if-then-else* (si-alors-sinon), mais pour l'Office des brevets, il s'agit d'une invention. Ils ont des conditions très peu restrictives, et ils s'en tiennent à ces conditions, et il en découle des brevets qui, pour nous, semblent absurdes et triviaux. Qu'ils soient juridiquement valides, je n'en sais rien. Mais n'importe quel programmeur rigole en les voyant.

Toujours est-il que je n'ai rien pu lui proposer pour se défendre, il a donc dû fermer boutique. Mais la plupart des détenteurs de brevets vous proposeront une licence, qui est susceptible de vous coûter très cher.

Cependant certains développeurs de logiciels n'ont pas grand mal à obtenir des licences, la plupart du temps. Il s'agit des mégacorporations. Quel que soit le domaine, les mégacorporations possèdent en général la moitié des brevets, elles s'accordent entre elles des licences croisées, et elles peuvent contraindre qui que ce soit d'autre à leur accorder des licences croisées. Au final, elles obtiennent sans trop de mal des licences pour quasiment tous les brevets.

IBM a écrit un article à ce sujet dans son magazine interne, *Think* - dans le numéro 5 de 1990, je crois - décrivant les avantages qu'IBM tirait de son portefeuille de près de 9 000 brevets américains (c'était à l'époque, maintenant ils en ont plus de 45 000). L'un de ces avantages, disaient-ils, est qu'ils en tiraient des revenus, mais ils soulignaient que le principal avantage - d'un ordre de grandeur supérieur - était « l'accès aux brevets des autres », par le biais des licences croisées.

Cela veut dire que puisque qu'IBM, avec sa pléthore de brevets, peut contraindre n'importe qui à lui accorder des licences croisées, cette entreprise échappe à tous les problèmes que le système de brevets cause à toutes les autres. Et c'est pour cela qu'IBM est favorable aux brevets logiciels. C'est pour cela que les mégacorporations sont dans leur ensemble favorable aux brevets logiciels : parce qu'elles savent que par le jeu des licences croisées, elles feront partie d'un petit club très fermé, assis au sommet de la montagne. Et nous, nous serons tout en bas, et il n'y aura aucun moyen de grimper. Vous savez, si vous êtes un génie, vous pouvez fonder votre petite entreprise, et obtenir quelques brevets, mais vous ne jouerez jamais dans la même catégorie qu'IBM, quels que soient vos efforts.

Beaucoup d'entreprises disent à leurs salariés : « Décrochez-nous des brevets, c'est juste pour que nous puissions nous défendre. » Et ce qu'elles veulent dire,

c'est : « Nous les utiliserons pour obtenir des licences croisées. » Mais en fait ça ne marche pas. Ce n'est pas une stratégie efficace si vous n'avez qu'un petit nombre de brevets.

Imaginons par exemple que vous ayez trois brevets. L'un concerne ceci, l'autre concerne cela, et soudain quelqu'un vous brandit son brevet sous le nez. Vos trois brevets ne vont servir à rien, car aucun d'entre eux ne concerne cette personne. En revanche, tôt ou tard, quelqu'un dans votre entreprise va se rendre compte que votre brevet concerne d'autres entreprises, et va l'utiliser pour les menacer et leur extorquer de l'argent, quand bien même ces entreprises n'ont pas attaqué la vôtre.

Donc, si votre employeur vous dit « Nous avons besoin de brevets pour nous défendre, aidez-nous à en obtenir », je vous conseille de répondre ceci :

Chef, je vous fais confiance, et je suis certain que vous n'utiliserez ces brevets que pour défendre l'entreprise en cas d'attaque. Mais je ne sais pas qui sera à la tête de l'entreprise dans 5 ans. Pour autant que je sache, elle peut se faire racheter par Microsoft. Je ne peux donc pas compter sur la promesse de l'entreprise de n'utiliser ces brevets que de manière défensive, sauf si elle s'engage par écrit. Commencez par vous engager par écrit à ce que tout brevet que je fournis à l'entreprise ne soit utilisé qu'en cas de légitime défense, et jamais de manière agressive, et alors je serai en mesure d'apporter des brevets à l'entreprise en ayant la conscience tranquille.

Il serait intéressant d'aborder ce problème sur la liste de diffusion de l'entreprise, et pas seulement en privé avec votre patron.

L'autre chose qui peut arriver, c'est que la société fasse faillite et que ses actifs soient liquidés, brevets compris, et que ces brevets soient rachetés par quelqu'un qui les utilise à des fins peu scrupuleuses.

Il est fondamental de comprendre cette pratique des licences croisées, car c'est elle qui permet de détruire l'argument des promoteurs du brevet logiciel selon lequel les brevets sont nécessaires pour protéger les pauvres petits génies. Ils vous présentent une histoire cousue d'invéraisemblances.

Jetons-y un œil. Dans leur histoire, un brillant concepteur de ce-que-vous-voulez,

qui a travaillé des années tout seul dans son grenier, a découvert une meilleure technique pour faire un-truc-important. Et maintenant que cette technique est au point, il veut se lancer dans les affaires et produire ce-truc-important en série, et comme son idée est géniale, son entreprise va forcément réussir. Problème : les grandes entreprises vont lui faire une concurrence acharnée et lui voler tout son marché. Et du coup, son entreprise va couler et il se retrouvera sur la paille.

Examinons les hypothèses fantaisistes que l'on trouve ici.

Tout d'abord, il est peu probable qu'il ait eu son idée géniale tout seul. Dans le domaine des hautes technologies, la plupart des innovations reposent sur une équipe travaillant dans un même domaine, et échangeant avec des gens de ce domaine. Mais pour autant, ce n'est pas impossible en soi.

L'hypothèse suivante est qu'il va se lancer dans les affaires, et qu'elles vont bien marcher. Le fait qu'il soit un ingénieur brillant n'implique en rien que ce soit un bon homme d'affaire. La plupart des entreprises - 95%, je crois - font faillite au cours des premières années. Donc peu importe le reste, c'est ce qui risque de lui arriver.

Bon, mais imaginons qu'en plus d'être un ingénieur brillant, qui a découvert quelque chose de génial tout seul dans son coin, ce soit aussi un homme d'affaire de génie. S'il est doué pour les affaires, son entreprise s'en sortira peut-être. Après tout, toutes les nouvelles entreprises ne coulent pas, un certain nombre prospèrent. S'il a le sens des affaires, peut-être qu'au lieu d'affronter les grandes entreprises sur leur terrain, il tentera de se placer là où les petites entreprises sont plus performantes et peuvent réussir. Il y parviendra peut-être. Mais supposons qu'il finisse par échouer. S'il est vraiment brillant, et qu'il est doué pour les affaires, je suis certain qu'il ne mourra pas de faim, parce que quelqu'un voudra l'embaucher.

Tout ceci ne tient pas debout. Mais poursuivons.

On arrive au moment où le système des brevets va « protéger » notre pauvre petit génie, parce qu'il va pouvoir breveter sa technique. Et quand IBM va venir lui faire concurrence, il va dire : « IBM, vous n'avez pas le droit de me faire concurrence, j'ai un brevet. » Et IBM va dire : « Oh non, encore un brevet ! »

Voilà ce qui va vraiment se passer :

IBM va dire : « Oh, comme c'est mignon, vous avez un brevet. Eh bien nous, nous avons celui-ci, et celui-ci, et celui-ci, et celui-ci, et ils couvrent tous d'autres idées implémentées dans votre produit, et si vous pensez que vous pouvez vous battre avec nous sur tous ceux-là, nous en sortirons d'autres. Donc, nous allons signer un accord de licences croisées, et personne ne sera lésé. » Comme notre génie est doué pour les affaires, il va vite se rendre compte qu'il n'a pas le choix. Il va signer un accord de licences croisées, comme le font tous ceux à qui IBM le demande. Ce qui veut dire qu'IBM va avoir « accès » à son brevet, et pourra donc lui faire librement concurrence comme si le brevet n'existait pas, ce qui signifie au final que l'hypothétique protection dont il bénéficie au travers de son brevet n'est qu'un leurre. Il ne pourra jamais en bénéficier.

Le brevet le « protégera » peut-être de concurrents comme vous et moi, mais pas d'IBM, pas des mégacorporations qui, dans la fable, sont justement celles qui le menacent. De toute façon, lorsque des mégacorporations envoient leurs lobbyistes défendre une législation qui est censée protéger leurs petits concurrents contre leur influence, vous pouvez être sûr que l'argumentation va être biaisée. Si c'était vraiment ce qui allait se passer, elles seraient contre. Mais ça permet de comprendre pourquoi les brevets logiciels ne marchent pas.

Même IBM ne peut pas toujours se comporter ainsi. Certaines entreprises, que l'on appelle parfois des « trolls des brevets », ont pour seul modèle économique d'utiliser les brevets pour soutirer de l'argent à ceux qui produisent réellement quelque chose.

Les avocats spécialisés en droit des brevets nous expliquent à quel point il est merveilleux que notre domaine ait des brevets. Mais il n'y a pas de brevet dans le leur. Il n'y a pas de brevet sur la manière d'envoyer ou de rédiger une lettre de menaces, pas de brevet sur comment déposer un recours, pas de brevet sur comment convaincre un juge ou un jury. Ainsi donc, même IBM ne peut contraindre un troll des brevets à accepter des licences croisées. Mais IBM se dit : « Nos concurrents vont devoir payer eux aussi, cela fait partie des charges incompressibles, on peut s'en accommoder. » IBM et les autres mégacorporations considèrent que la suprématie qu'elles retirent de leurs brevets sur l'ensemble de leur activité vaut bien d'avoir à payer quelques parasites. C'est pour cela qu'elles défendent les brevets logiciels.

Il existe aussi certains développeurs de logiciels qui ont beaucoup de mal à

obtenir des licences, ce sont les développeurs de logiciels libres. Cela tient au fait que les licences contiennent généralement des conditions que nous sommes dans l'impossibilité de remplir, comme par exemple le paiement pour chaque exemplaire distribué. Lorsque les utilisateurs sont libres de copier le logiciel et de distribuer les copies, personne ne peut connaître le nombre de copies en circulation.

Si quelqu'un me propose une licence pour un brevet au prix d'un millionième de dollar par exemplaire distribué, il se peut que j'aie de quoi payer. Mais je n'ai aucun moyen de savoir si cela représente 50 dollars, ou 49, ou un autre montant, car je n'ai aucun moyen de compter le nombre de copies que les gens ont faites.

Un titulaire de brevet n'est pas obligé de demander le paiement en fonction du nombre d'exemplaires distribués. Il peut très bien proposer une licence pour un montant fixe, mais ce montant a tendance à être très élevé, du type 100 000 \$.

Or, si nous avons pu développer tant de logiciels respectueux des libertés, c'est parce que nous pouvons développer des logiciels sans argent. Mais nous ne pouvons pas payer sans argent. Si nous sommes obligés de payer pour avoir le privilège d'écrire des logiciels pour le public, nous risquons de ne pas en faire beaucoup.

Voilà pour la possibilité d'obtenir une licence pour un brevet. La dernière solution consiste à faire annuler le brevet. Si le pays reconnaît les brevets logiciels et les autorise, la seule question est de savoir si tel ou tel brevet respecte bien les conditions de validité. Aller devant les juges ne sert à rien si vous n'avez pas un argument solide qui vous permettra de l'emporter.

De quel argument peut-il s'agir ? Il faut apporter la preuve que, bien avant que le brevet ne soit déposé, d'autres personnes avaient eu la même idée. Et il faut trouver des preuves aujourd'hui qui montrent que l'idée était publique à l'époque. Ainsi, les dés ont été jetés des années plus tôt, et s'ils vous sont favorables, et si vous êtes en mesure d'apporter cette preuve aujourd'hui, alors vous disposez d'un argument qui peut vous permettre de contester le brevet et d'obtenir son annulation. Ça peut marcher.

Ce genre d'affaire peut coûter cher. De ce fait, un brevet probablement invalide constitue malgré tout un moyen de pression très efficace si vous n'avez pas beaucoup d'argent. Beaucoup de gens n'ont pas les moyens de défendre leurs

droit. Ceux qui peuvent se le permettre constituent l'exception.

Voilà les trois possibilités qui peuvent se présenter à vous, chaque fois qu'un brevet interdit quelque chose dans votre programme. Toutes ne sont pas toujours ouvertes, cela dépend de chaque cas particulier, et parfois aucune d'entre elles n'est envisageable. Lorsque cela se produit, votre projet est condamné.

Mais dans la plupart des pays les avocats nous conseillent de « ne pas rechercher d'avance les brevets », la raison étant que les sanctions pour infraction à un brevet sont plus importantes s'il est établi que l'on avait connaissance de l'existence du brevet. Ce qu'ils nous disent, c'est : « Gardez les yeux fermés. N'essayez pas de vous renseigner sur les brevets, décidez de la conception de votre programme en aveugle, et priez. »

Évidemment, vous ne marchez pas sur un brevet à chaque étape de la conception. Il ne va probablement rien vous arriver. Mais il y a tellement de pas à faire pour traverser le champ de mines qu'au final il est peu probable que vous y arriviez sans encombre. Et bien sûr, les titulaires de brevet ne se présentent pas tous au même moment, de sorte que vous ne pouvez jamais savoir combien il y en aura.

Le titulaire du brevet sur le recalcul dans l'ordre naturel demandait 5% du montant brut de chaque tableur vendu. Il n'est pas inconcevable de payer quelques licences de ce type, mais que faire quand le titulaire de brevet n° 20 se présente, et vous demande les 5 derniers pourcent ? Ou le titulaire du brevet n° 21 ?

Les professionnels du secteur trouvent cette histoire amusante, mais absurde, car dans la réalité votre activité s'écroule bien avant d'en arriver là. Il suffit de deux ou trois licences de ce type pour couler votre entreprise. Vous n'arriveriez jamais à 20. Mais comme ils se présentent un par un, vous ne pourrez jamais savoir combien il y en aura.

Les brevets logiciels sont un immense gâchis. Non seulement ils constituent une jungle pour les développeurs de logiciels, mais en plus ils restreignent la liberté de chaque utilisateur d'ordinateur, car chaque brevet logiciel restreint ce que vous pouvez faire avec un ordinateur.

La situation est très différente de celle des brevets sur les moteurs de voiture, par exemple. Ces brevets n'affectent que les entreprises qui fabriquent des voitures,

ils ne restreignent pas votre liberté ni la mienne. Mais les brevets logiciels touchent tous ceux qui utilisent un ordinateur. Il est donc impossible de les examiner sous un angle purement économique. Cette question ne peut pas être tranchée en termes purement économiques. Quelque chose de plus fondamental est en jeu.

Mais même sur le plan purement économique, le système est contre-performant, car sa raison d'être originelle est de promouvoir l'innovation. L'idée était qu'en créant une incitation artificielle à rendre une idée publique, cela stimulerait l'innovation. Au final c'est exactement l'inverse qui se produit, car l'essentiel du travail de fabrication d'un logiciel n'est pas d'inventer des idées nouvelles, mais de mettre en œuvre, conjointement dans un seul programme, des milliers d'idées différentes. Et c'est ce que les brevets logiciels empêchent de faire, et en cela ils sont économiquement contre-performants.

Il existe même des études économiques qui le prouvent, et qui montrent comment, dans un domaine où la plupart des innovations sont incrémentales, un système de brevets peut en réalité diminuer les investissements en recherche et développement. Et ils entravent l'innovation de bien d'autres manières. Donc, même si on laisse de côté le fait que les brevets logiciels sont injustes, même si l'on s'en tient à une approche purement économique, les brevets restent néfastes.

Parfois, les gens nous répondent : « Les autres disciplines vivent avec les brevets depuis des décennies et elles s'y sont faites, pourquoi faudrait-il faire une exception pour vous ? »

Cette question repose sur un présupposé absurde. Cela revient à dire « D'autres personnes ont un cancer, pourquoi pas vous ? » De mon point de vue, chaque fois qu'une personne évite un cancer, c'est une bonne chose, quel que soit le sort des autres. Cette question est absurde, parce qu'elle sous-entend qu'il faudrait que nous souffrions tous des dommages causés par les brevets.

Mais elle contient implicitement une question qui, elle, est pertinente : « Existe-t-il des différences entre les disciplines telles qu'un bon système de brevets pour l'une peut être mauvais pour l'autre ? »

Or il existe une différence fondamentale entre disciplines, quant au nombre de brevets nécessaires pour bloquer ou couvrir un produit donné.

Je suis en train d'essayer de nous débarrasser d'une représentation simpliste mais fréquente, qui voudrait qu'à chaque produit corresponde un brevet, et que ce brevet couvre la structure générale du produit. Selon cette idée, si vous concevez un nouveau produit, il est impossible qu'il soit déjà breveté, et vous pourrez tenter d'obtenir « le brevet » pour ce produit.

Ce n'est pas ainsi que les choses fonctionnent. Ou peut-être au XIXe siècle, mais plus aujourd'hui. En fait, chaque discipline peut être placée sur une échelle représentant le nombre de brevets par produit. Tout en bas de l'échelle, il suffit d'un brevet pour couvrir un produit, mais il n'existe plus de discipline qui fonctionne ainsi aujourd'hui ; les disciplines sont à divers niveaux de l'échelle.

Le secteur qui serait le plus bas sur l'échelle serait l'industrie pharmaceutique. Il y a quelques dizaines d'années, il y avait réellement un brevet par médicament, à un instant donné tout du moins, car le brevet portait sur l'ensemble de la formule chimique correspondant à une substance donnée. À l'époque, si vous aviez développé un nouveau médicament, vous pouviez être certain qu'il n'était pas déjà breveté par quelqu'un d'autre, et vous pouviez obtenir un brevet correspondant à ce médicament particulier.

Mais ce n'est plus comme ça que ça marche. Il y a désormais des brevets très généraux, de sorte que même si vous mettez au point une nouvelle molécule, il se peut qu'elle soit interdite parce qu'elle est couverte par un brevet plus large.

Il se peut qu'il y ait deux ou trois brevets de ce type qui couvrent votre molécule, mais il n'y en aura pas des centaines. Cela tient au fait que nos capacités dans le domaine de la biochimie sont si limitées que personne n'est capable de combiner autant d'idées différentes dans une seule molécule qui serait utilisable en médecine. Si vous parvenez à en combiner deux, c'est déjà un exploit, pour notre niveau de connaissances. Mais dans les autres disciplines il n'est pas rare de combiner un plus grand nombre d'idées pour réaliser une seule chose.

Les logiciels se situent tout en haut de l'échelle. Un logiciel repose sur la combinaison de plus d'idées que n'importe quel autre produit, du fait essentiellement que notre discipline est plus simple que les autres. Je prends pour hypothèse que l'intelligence des gens travaillant dans notre domaine vaut celle de ceux travaillant sur le monde physique. Nous ne sommes pas meilleurs qu'eux, c'est juste que notre discipline est fondamentalement plus simple que la leur, car

nous travaillons avec les mathématiques.

Un programme est fait d'éléments mathématiques, qui ont tous une définition, alors que les objets physiques n'ont pas de définition. La matière se comporte comme elle se comporte, et parfois elle est perverse, et votre invention ne fonctionne pas comme elle était « censée » fonctionner. Pas de chance. Vous ne pouvez pas prétendre que c'est parce que la matière est boguée, et qu'il faudrait patcher l'univers physique. Alors que nous, les programmeurs, nous pouvons construire un édifice qui repose sur une ligne mathématique d'épaisseur nulle, et il tiendra debout car rien ne pèse rien.

Nous n'avons pas à affronter toutes les complications du monde physique.

Par exemple, quand je mets une condition `if` (si) à l'intérieur d'une boucle `while` (tant que),

- je n'ai pas à craindre que si cette boucle `while` se répète à la mauvaise fréquence, la condition `if` entre en résonance et se brise ;
- je n'ai pas à me préoccuper du fait que si elle boucle trop rapidement (disons des millions de fois par seconde), elle est susceptible de générer des signaux électromagnétiques qui pourraient fausser les valeurs ailleurs dans le programme ;
- je n'ai pas à craindre que des fluides corrosifs présents dans l'environnement s'infiltrent entre le `if` et le `while`, et commencent à les ronger, jusqu'à ce que le signal ne passe plus.
- je n'ai pas à me demander comment la chaleur générée par mon `if` va pouvoir s'évacuer via mon `while` pour ne pas qu'il grille ; et
- je n'ai pas à me demander comment je vais pouvoir démonter et remplacer le `if`, s'il fini par se briser, par griller ou par se corroder, afin de remettre le programme en état de marche.

Je n'ai même pas à me demander comment je vais pouvoir insérer le `if` dans le `while` pour chaque exemplaire du programme. Je n'ai pas besoin de concevoir une usine pour fabriquer des copies de mon programme, car il existe un certain nombre de commandes générales qui permettent de copier tout et n'importe quoi.

Si je veux graver des copies sur CD, il me suffit de préparer une image, et il existe pour ce faire un programme, qui me permet de réaliser une image à partir de n'importe quelles données. Je peux graver un CD et l'envoyer à une usine, qui se

chargera de dupliquer ce que je lui envoie. Je n'ai pas besoin de concevoir une usine différente pour chaque chose que je veux reproduire.

Or, dans l'ingénierie physique, c'est bien souvent ce que vous devez faire. Vous devez concevoir vos produits de sorte qu'ils puissent être fabriqués. Et il est encore plus compliqué de concevoir l'usine que de concevoir le produit, et il faut ensuite dépenser des millions de dollars pour construire l'usine. Avec toutes ces contraintes, vous n'allez pas pouvoir concevoir un produit qui réunisse beaucoup d'idées différentes et qui fonctionne.

Un objet physique composé d'un million d'éléments uniques représente un projet titanesque. Un programme composé d'un million d'éléments différents, ce n'est rien. Cela représente quelques centaines de milliers de lignes de code, soit quelques années de travail pour un petit groupe de personnes, donc ce n'est pas énorme. En conséquence, le système de brevets pèse beaucoup plus lourdement sur notre discipline que sur toutes celles qui sont freinées par la perversité de la matière.

Un avocat a réalisé une étude portant sur un programme particulièrement volumineux, à savoir le noyau Linux, qui est utilisé conjointement au système d'exploitation GNU que j'ai lancé. C'était il y a 5 ans ; il a trouvé 283 brevets américains qui apparemment interdisent chacun un type de calcul présent quelque part dans le code de Linux. A l'époque, j'ai lu quelque part que Linux représentait 0,25% de l'ensemble du système. Donc, si vous multipliez ce chiffre par 300 ou 400, vous obtenez une estimation du nombre de brevets qui sont susceptibles d'interdire quelque chose dans l'ensemble du système, soit à peu près 100 000. Il s'agit d'une estimation très approximative, mais nous n'avons aucune information plus précise, car le seul fait de chercher à savoir représenterait une tâche titanesque.

Cet avocat n'a pas publié la liste des brevets, de peur que cela n'expose les développeurs du noyau Linux à des sanctions plus importantes en cas de litige. Il ne cherchait pas à les mettre en difficulté, il voulait juste démontrer la gravité du problème que représente l'inflation des brevets.

Les programmeurs comprennent tout de suite de quoi il s'agit, mais les politiciens ne comprennent pas grand chose à l'informatique. Ils s'imaginent que les brevets ressemblent à une version renforcée du droit d'auteur. Ils s'imaginent que

puisque les développeurs ne sont pas menacés par le droit d'auteur sur leur travail, ils ne seront pas menacés non plus par les brevets. Ils s'imaginent que puisque vous obtenez un droit d'auteur lorsque vous écrivez un programme, vous pourriez aussi bien obtenir un brevet. C'est complètement faux. Comment leur faire comprendre l'effet qu'auraient les brevets ? L'effet qu'ils ont, dans des pays comme les États-Unis ?

J'ai souvent recours à une analogie entre les programmes et les symphonies. Voici pourquoi :

Une symphonie, comme un programme, combine de nombreuses idées. Une symphonie combine de nombreuses idées musicales. Mais il ne suffit pas de choisir une série d'idées et de dire : « Voilà ma combinaison d'idées, ça vous plaît ? » Pour que cela fonctionne, il faut les implémenter. Vous ne pouvez pas juste dresser une liste d'idées musicales, et demander « Ça vous plaît ? », car on ne peut pas entendre cette liste. Il faut écrire des notes, qui représentent la conjonction de ces idées.

Le plus difficile est de choisir des notes qui donnent un résultat final harmonieux ; la plupart d'entre nous en est incapable. Bien sûr, nous sommes tous capables de choisir des idées musicales dans une liste, mais nous serions bien en peine d'écrire une symphonie qui rassemble harmonieusement ces idées. Seuls quelques-uns d'entre nous ont ce talent. C'est cela qui nous limite. Je pourrais probablement inventer quelques idées musicales, mais je ne saurais pas comment en tirer quoi que ce soit.

Imaginez que nous sommes au XVIIIe siècle, et que les gouvernements européens décident de mettre en place un système de brevets sur les idées musicales pour promouvoir l'innovation dans le domaine de la musique symphonique, et que n'importe quelle idée musicale décrite sous forme de mots puisse être brevetée.

Par exemple, le fait d'utiliser comme motif une suite de notes donnée pourra être breveté, de même qu'une progression d'accords, ou une trame rythmique, ou l'utilisation de certains instruments, ou un format de répétitions dans un mouvement. N'importe quelle idée musicale qui pourrait être traduite en mots serait brevetable.

Imaginez maintenant que nous sommes en 1800, vous êtes Beethoven, et vous voulez écrire une symphonie. Vous allez vous rendre compte qu'il est beaucoup

plus difficile d'écrire une symphonie pour laquelle vous n'aurez pas de procès que d'écrire une symphonie qui soit belle, car il faudra vous tailler un chemin dans la jungle des brevets existants. Et si vous vous plaignez de cet état de fait, les titulaires de brevets vous répondront : « Oh Beethoven, tu es jaloux parce que c'est nous qui avons eu ces idées en premier. Tu n'as qu'à chercher un peu et trouver des idées originales. »

Beethoven avait beaucoup d'idées originales. La raison pour laquelle il est considéré comme un grand compositeur, c'est justement parce qu'il a eu beaucoup d'idées originales, et qu'il a su les mettre en musique de manière efficace, c'est-à-dire en les combinant avec beaucoup d'autres idées très répandues. En introduisant dans une composition quelques idées nouvelles, au milieu de beaucoup d'idées plus anciennes et plus classiques, il obtenait des morceaux novateurs, mais pas au point que les gens ne puissent plus les comprendre.

À nos oreilles, la musique de Beethoven n'a rien de révolutionnaire. Apparemment c'était le cas au XIXe, mais comme il a su mêler ses idées nouvelles à d'autres mieux acceptées, il a pu donner aux gens une chance de s'y adapter. Et c'est ce qu'ils ont fait, c'est pourquoi aujourd'hui cette musique ne nous pose aucun problème. Mais personne, pas même un génie comme Beethoven, n'est capable de réinventer la musique à partir de rien, sans faire appel à aucune des idées de son temps, tout en aboutissant à quelque chose que les gens ont envie d'écouter. Et personne n'est assez génial pour réinventer l'informatique à partir de zéro, sans faire appel à aucune idée de son temps, tout en obtenant quelque chose que les gens ont envie d'utiliser.

Quand le contexte technologique change très rapidement, vous finissez par vous retrouver dans une situation où ce qui a été fait il y a vingt ans ne correspond plus à rien. Il y a vingt ans le World Wide Web n'existait pas. Bien sûr, on pouvait faire plein de choses avec un ordinateur, à l'époque, mais ce que les gens veulent aujourd'hui, ce sont des choses qui fonctionnent avec le World Wide Web. Et il n'est pas possible de faire cela en utilisant seulement des idées qui datent d'il y a vingt ans. Et j'imagine que le contexte technologique va continuer à évoluer, offrant de nouvelles occasions à certains d'obtenir des brevets qui nuisent à l'ensemble de la discipline.

Les grandes entreprises elles-mêmes le font. Par exemple, il y a quelques années,

Microsoft a décidé de créer un faux standard ouvert pour les documents, et a obtenu son approbation en corrompant l'ISO (Organisation internationale de normalisation). Le format était conçu à partir d'un brevet que Microsoft avait déposé. Microsoft est suffisamment puissante pour pouvoir partir d'un brevet, et concevoir un format ou un protocole qui utilise ce brevet (que ce soit utile ou non), de telle manière qu'il n'existe aucun moyen d'être compatible, sauf à utiliser la même idée. Ensuite, avec ou sans l'aide d'organismes de normalisation corrompus, Microsoft peut en faire un standard de fait. Par son seul poids, elle peut inciter les utilisateurs à recourir à ce format, ce qui lui donne une mainmise au niveau mondial. Il faut montrer aux hommes politiques ce qui se passe réellement dans ces domaines. Il faut leur expliquer pourquoi cela n'est pas bon.

J'ai entendu dire que la raison pour laquelle la Nouvelle-Zélande veut mettre en place des brevets logiciels est qu'une grande entreprise cherche à obtenir des monopoles. Limiter les libertés de chacun afin qu'une seule entreprise puisse augmenter ses profits, c'est être l'antithèse absolue d'un homme d'Etat.

J'aimerais maintenant passer aux questions-réponses.

Q : Quelle est l'alternative ?

RMS : Pas de brevets logiciels. Je sais que cela fonctionne bien. Je travaillais dans cette discipline avant l'apparition des brevets logiciels. Les gens développaient des logiciels et les distribuaient de différentes manières, sans avoir à craindre un procès de la part d'un titulaire de brevet. Les brevets logiciels sont une réponse à un faux problème, il n'y a donc pas à rechercher d'autres solutions.

Q : Comment les développeurs sont-ils récompensés ?

RMS : Il existe de nombreux moyens. Mais les brevets logiciels n'ont rien à voir avec cela. N'oubliez pas que si vous êtes un développeur de logiciels, les brevets logiciels ne vous aideront pas à obtenir ce que vous cherchez à obtenir.

Il existe différents types de développeurs de logiciels, qui recherchent des choses différentes. Dans les années 80, j'ai développé certains logiciels importants, et la récompense que je recherchais était de voir plus de gens utiliser un ordinateur en toute liberté. J'ai obtenu cette récompense, au moins partiellement - tout le monde ne dispose pas de cette liberté. Mais les brevets logiciels n'auraient fait que m'en empêcher.

D'autres personnes développent des programmes parce qu'ils veulent gagner de l'argent. Pour eux aussi, les brevets logiciels sont une menace, car vous n'allez rien gagner si le titulaire d'un brevet vous oblige à tout lui donner ou vous fait fermer.

Q : Comment faire pour lutter contre le plagiat et...

RMS : Le plagiat n'a rien à voir là-dedans. Absolument rien à voir.

Le plagiat représente le fait de copier le texte d'une œuvre en prétendant l'avoir écrit soi-même. Mais les brevets ne concernent en rien le texte d'une œuvre. Ils n'ont aucun lien avec le plagiat.

Si vous écrivez une œuvre, et que cette œuvre contient certaines idées (ce qui est toujours le cas), il n'y a aucune raison de supposer que les brevets concernant ces idées vous appartiennent. Plus vraisemblablement, ils appartiennent à beaucoup d'autres personnes, la plupart étant des mégacorporations, et elles sont toutes en position de vous poursuivre. Vous n'avez même pas à vous soucier du risque de plagiat. Avant même d'en être arrivé au point où quelqu'un d'autre pourrait vous copier, vous vous serez fait plumer.

Vous confondez brevets et droit d'auteur, j'en ai peur. Les deux n'ont rien à voir. Je vous ai expliqué ce que le système de brevets faisait au logiciel, mais je pense que vous ne me croyez pas parce que vous mélangez droit d'auteur et brevet. Vous supposez que ce que fait l'un, l'autre le fait aussi ; or ce n'est pas le cas. Si vous écrivez du code, le droit d'auteur sur ce code vous appartient, mais si ce code met en œuvre des idées, et que certaines d'entre elles sont brevetées par d'autres, ces derniers peuvent vous poursuivre.

Avec le droit d'auteur, lorsque vous écrivez le code vous-même, vous n'avez pas à craindre que quelqu'un d'autre vienne vous poursuivre, car le droit d'auteur n'interdit que la copie. En fait, même si vous écrivez du code totalement identique au code de quelqu'un d'autre, si vous prouvez que vous ne l'avez pas recopié, ce sera un moyen de défense car le droit d'auteur ne restreint que le fait de recopier. Il ne s'intéresse qu'à la paternité d'une œuvre et non aux idées qui y sont développées, donc son objet est fondamentalement distinct de celui des brevets, et ses conséquences sont radicalement différentes.

Je ne suis pas toujours d'accord avec la manière dont les gens utilisent le droit

d'auteur, et je me suis exprimé à ce sujet. Mais c'est une question totalement différente, sans aucun lien avec celle d'aujourd'hui. Si vous pensez que le droit des brevets aide ceux qui développent des logiciels, cela veut dire que vous avez une image complètement fautive de ce que fait le droit des brevets.

Q : Ne vous méprenez pas. Je suis de votre côté.

RMS : OK, mais vous avez quand même une image faussée. Je ne vous le reproche pas, car vous avez été victime de désinformation.

Q : J'écris des logiciels à des fins commerciales ; suis-je protégé si je les considère comme des boîtes noires et que je les garde secrets ?

RMS : Je ne veux pas discuter de ce problème, car je suis contre ces pratiques, je pense qu'elles sont contraires à l'éthique, mais il s'agit d'un problème distinct.

Q : Je comprends.

RMS : Je ne veux pas changer de sujet, et faire l'éloge de quelque chose que je désapprouve. Mais comme il s'agit d'un sujet différent, je préfère ne pas l'aborder.

Q : Notre Fondation pour la recherche, la science et la technologie, qui doit probablement être l'équivalent de votre Fondation nationale pour la science, offre des bourses de recherche et développement, et l'un des points sur lesquels elle insiste particulièrement est que les idées qu'elle a contribué à financer devraient faire l'objet de brevets.

RMS : Cela ne devrait pas être le cas dans le domaine des logiciels, car les idées informatiques ne devraient pas pouvoir être brevetées par qui que ce soit. Mais ce que vous voyez ici, plus généralement, n'est qu'un exemple de plus de la corruption généralisée de notre société, qui place les fins commerciales au-dessus de toutes les autres. Je ne suis pas communiste, et je ne souhaite pas abolir le commerce, mais lorsqu'on en arrive au commerce par-dessus tout, dans tous les domaines de l'existence, cela me semble dangereux.

Q : Richard, si vous vous adressiez à la Fondation, peut-être pourriez-vous leur proposer d'autres solutions pour qu'un petit pays comme la Nouvelle-Zélande puisse gagner de l'argent au travers des logiciels ?

RMS : Les brevets logiciels n'aident personne à gagner de l'argent avec des logiciels. Ce qu'ils signifient, c'est que vous risquez un procès si vous essayez.

Q : Et cela empêche la Nouvelle-Zélande de construire son économie en s'appuyant sur les logiciels.

RMS : Désolé, mais quand vous dites « cela », je ne comprends pas bien à quoi vous faites référence. Avec les brevets logiciels, ce que vous décrivez devient compliqué pour tout le monde. Si la Nouvelle-Zélande autorise les brevets logiciels, il sera difficile pour qui que ce soit dans le pays de développer des programmes et de les distribuer, à cause du risque de procès. Les brevets logiciels n'ont rien à voir avec le fait de développer un logiciel et de s'en servir.

Q : Donc, en terme de développement économique, la Nouvelle-Zélande serait mieux protégée en n'ayant pas de droit des brevets.

RMS : Oui. Vous voyez, chaque pays a son propre système de brevets, et ils fonctionnent tous de manière indépendante, sauf entre les pays qui ont signé des traités qui disent : « Si vous avez un brevet dans tel pays, vous pouvez venir chez nous avec votre demande de brevet, et nous la prendront en compte à la date à laquelle vous l'avez faite là-bas. » Mais à part ça, chaque pays a ses propres critères concernant ce qui est brevetable, et possède ses propres séries de brevets.

Il en découle que, dans la mesure où les États-Unis autorisent les brevets logiciels et pas la Nouvelle-Zélande, n'importe qui dans le monde, y compris des Néo-Zélandais, peut obtenir des brevets américains et poursuivre de pauvres Américains chez eux. Mais personne ne peut obtenir de brevet permettant de poursuivre un Néo-Zélandais chez lui. Vous pouvez être certain que (si la Nouvelle-Zélande les autorise) presque tous les brevets logiciels appartiendront à des étrangers qui les utiliseront pour matraquer n'importe quel développeur néo-zélandais dès que l'occasion se présentera.

Q : Depuis l'affaire Hughes Aircraft, je crois que c'était dans les années 1990...

RMS : Je ne connais pas cette affaire.

Q : Eh bien en fait la Nouvelle-Zélande autorise les brevets logiciels. Ce n'est pas comme si nous entrions dans un territoire vierge, cela existe déjà.

RMS : Je ne sais pas, mais j'avais cru comprendre qu'il allait y avoir une décision au niveau législatif sur l'opportunité ou non d'autoriser les brevets logiciels. Ceci dit, les offices de brevets se montrent souvent réceptifs au lobbying que les mégacorporations exercent au travers de l'OMPI.

L'OMPI, comme le nom le laisse supposer, œuvre dans le mauvais sens, car l'usage du terme « propriété intellectuelle » ne fait qu'accroître la confusion. L'OMPI tire une grande partie de ses ressources des mégacorporations et utilise ces ressources pour inviter les responsables des offices de brevets à des séminaires dans des destinations paradisiaques. Ce qu'on leur apprend dans ces séminaires, c'est à contourner la loi pour accorder des brevets dans des domaines où ils sont normalement interdits.

Dans de nombreux pays, il existe des lois et une jurisprudence qui posent que les logiciels en tant que tels ne peuvent être brevetés, que les algorithmes ne peuvent être brevetés, ou que les algorithmes « mathématiques » (personne ne sait exactement ce qui rend un algorithme mathématique ou non) ne peuvent être brevetés, et il existe divers autres critères qui, interprétés normalement, devraient exclure les logiciels du champ des brevets. Mais les offices de brevets tordent la loi pour les autoriser malgré tout.

Par exemple, de nombreuses inventions sont en réalité des brevets logiciels, mais sont décrites comme un système incluant un processeur, de la mémoire, des interfaces d'entrée/sortie et d'acquisition des instructions, ainsi que des moyens d'effectuer un calcul particulier. Au final, ce qui est décrit dans le brevet, ce sont les différents éléments d'un ordinateur classique, mais cela leur permet de dire : « C'est un système physique que nous souhaitons breveter. » En réalité, cela revient à breveter un logiciel installé sur un ordinateur. Les subterfuges utilisés sont légion.

Les offices de brevets cherchent généralement à détourner la loi pour accorder plus de brevets. Aux États-Unis, les brevets logiciels ont été créés en 1982 par une décision de la Cour d'appel compétente pour les affaires de brevets, qui a mal interprété une décision de la Cour suprême rendue l'année précédente et l'a appliquée à mauvais escient. Dans une décision récente, il semble que la Cour d'appel ait enfin admis qu'elle s'est trompée depuis le début, et il est possible que cette décision nous débarrasse de tous les brevets logiciels, à moins qu'elle ne soit renversée par la Cour suprême. La Cour suprême est en train de l'examiner,

et nous devrions savoir dans moins d'un an si nous avons gagné ou perdu.

Q : Dans l'hypothèse où cette affaire se terminerait en faveur des brevets, existe-t-il aux États-Unis un mouvement pour promouvoir une solution législative ?

RMS : Oui, et cela fait à peu près 19 ans que je milite en faveur de cette solution. C'est un combat que nous menons dans de nombreux pays.

Q : Où placeriez-vous dans votre univers le cas de I4i ?

RMS : Je n'ai aucune idée de ce dont il s'agit.

Q : Il s'agit de l'affaire dans laquelle Microsoft a presque dû cesser de commercialiser Word, parce que le logiciel enfreignait un brevet canadien.

RMS : Ah oui, ça. C'est juste un exemple qui illustre le danger que représentent les brevets logiciels pour tous les développeurs. Je n'aime pas ce que fait Microsoft, mais c'est un autre problème. Il n'est pas bon que quelqu'un puisse poursuivre un développeur de logiciels et dire : « Vous ne pouvez pas distribuer tel logiciel. »

Q : Le monde dans lequel nous vivons n'est évidemment pas parfait, et nous nous heurtons quelquefois aux brevets logiciels. Pensez-vous qu'il faudrait accorder un privilège aux chercheurs, leur permettant d'ignorer les brevets logiciels de la même manière que la législation sur le droit d'auteur leur permet d'effectuer des recherches sur des œuvres protégées par ce dernier ?

RMS : Non, chercher une solution partielle est une erreur, car nos chances de mettre en place une solution complète sont bien plus élevées. Toutes les personnes impliquées dans le développement et la distribution de logiciels, à l'exception de celles qui travaillent dans les mégacorporations, vont se rallier au rejet total des brevets logiciels lorsqu'elles verront à quel point ils sont dangereux. En revanche, proposer une exception pour certaines catégories particulières ne ralliera que les membres de cette catégorie. Ces solutions partielles sont des leurres. Les gens disent : « Bon, on ne peut pas résoudre le problème une bonne fois pour toutes, j'abandonne. Je propose une solution partielle. » Mais ces solutions partielles ne mettront pas les développeurs de logiciels à l'abri.

Q : Cependant vous ne vous opposeriez pas à une solution partielle, pas

nécessairement limitée aux brevets logiciels, comme par exemple l'utilisation à des fins d'expérimentation qui pourrait être une bonne solution pour les brevets pharmaceutiques ?

RMS : Je ne m'y opposerais pas.

Q : Mais ce que vous dites, juste pour être bien clair, c'est que vous ne pensez pas que ce soit applicable au logiciel.

RMS : Une solution qui ne sauve que certains d'entre nous, ou seulement certaines activités, ou élimine seulement la moitié des brevets logiciels, cela revient à dire : « On pourrait peut-être enlever la moitié des mines du champs de mines. » C'est un progrès, mais ça n'élimine pas le danger pour autant.

Q : Vous avez parlé de ce sujet aux quatre coins de la planète. Quel a été l'impact ? Certains gouvernements ont-ils apporté des changements, ou renoncé aux brevets logiciels ?

RMS : Certains. En Inde, il y a quelques années, il y a eu une tentative pour autoriser explicitement les brevets logiciels dans la loi. Le projet a été abandonné. Il y a quelques années, les États-Unis ont proposé un traité commercial, un traité de « libre-exploitation » à l'Amérique Latine. Ce traité a été bloqué par le président du Brésil, qui s'est opposé aux brevets logiciels et à d'autres dispositions insidieuses concernant l'informatique, et cela a fait capoter l'ensemble du traité. Il semble que c'était le seul point que les États-Unis tenaient à imposer au reste du continent. Mais on ne peut tuer ces projets pour de bon. Certaines entreprises ont des équipes à plein temps qui recherchent des moyens de subvertir tel ou tel pays.

Q : Dispose-t-on de données chiffrées sur ce qui se passe au plan économique dans les communautés innovantes des pays dénués de droit des brevets ?

RMS : Il n'y en a pas. Il est presque impossible de mesurer ce genre de choses. En réalité, je ne devrais pas dire qu'il n'y en a pas. Il y en a un peu. Il est très difficile de mesurer l'effet du système de brevets, car vous allez comparer la réalité avec un monde fictif, et il n'y a aucun moyen de savoir ce qui se passerait vraiment.

Ce que je peux dire, c'est qu'avant l'apparition des brevets logiciels, il y avait

beaucoup de développement logiciel. Pas autant qu'aujourd'hui, bien sûr, car il n'y avait pas autant d'utilisateurs d'ordinateurs.

Combien d'utilisateurs d'ordinateurs y avait-il en 1982, même aux États-Unis ? C'était une toute petite partie de la population. Mais il y avait des développeurs de logiciels. Ils ne disaient pas : « Nous avons absolument besoin de brevets. » Ils ne se retrouvaient pas poursuivis pour infraction à un brevet après avoir développé un programme. Mais le peu de recherche économique que j'aie pu voir montre qu'apparemment les brevets logiciels ont entraîné non pas un accroissement de la recherche, mais un transfert de ressources de la recherche vers les brevets.

Q : Pensez-vous qu'il puisse y avoir un regain d'intérêt pour les secrets de fabrication ?

RMS : Non. Avant l'apparition des brevets logiciels, de nombreux développeurs gardaient les détails de leurs programmes secrets. Mais habituellement, ils ne gardaient pas secrètes les idées générales, parce qu'ils se rendaient compte que la majeure partie du travail de développement d'un bon logiciel résidait, non pas dans l'élaboration d'idées générales, mais dans la mise en œuvre conjointe de nombreuses idées. Ils publiaient donc - ou laissaient leurs salariés publier - les nouvelles idées intéressantes qu'ils avaient eues dans des revues universitaires. Maintenant, ils brevètent ces idées nouvelles. Cela n'a rien à voir avec le fait de développer des programmes utiles, et partager certaines idées avec d'autres ne leur donne pas un programme. Par ailleurs, les milliers d'idées que vous avez combinées dans votre programme sont de toute façon bien connues, pour la plupart.

Q : Pour renforcer ce que vous venez de dire, j'ai entendu récemment une interview de l'un des fondateurs de PayPal, qui disait que son succès reposait à 5% sur des idées et à 95% sur leur mise en œuvre, ce qui confirme votre point.

RMS : Je suis d'accord.

SF : Très bien. Richard a ici des autocollants qui sont gratuits (free), je crois.

RMS : Gratis^[2]. Et ceux-là sont à vendre.

SF : Vous êtes les bienvenus si vous souhaitez nous rejoindre. Ce fut un débat très

constructif, merci Richard.

Notes

[1] Crédit photo : Amine Ghrabi (Creative Commons By-Nc)

[2] Le mot anglais free peut signifier « libre » (comme dans « libre expression » ou « logiciel libre »), ou bien « gratuit ». Cela pose problème dans l'interprétation de free software, c'est pourquoi RMS utilise le mot gratis quand il s'agit de gratuité. ?