

# Conférence de Richard Stallman

*Transcription de la conférence de Richard Matthew Stallman (rms) du mardi 10 novembre 1998, Université Paris 8, France, à l'invitation de l'association APRIL, devant plus de 200 personnes. (Amphithéâtre 3, 19 h 30 / 22 h 30).*

*Richard Stallman a donné sa conférence en français.*

*Cette transcription est aussi fidèle que possible, seul les tournures de phrases ont été réécrites. Toutes erreurs sont les erreurs des transcripteurs : Frédéric Couchet et Sébastien Blondeel.*

---

## 1. Retranscription de la conférence

Je peux décrire l'idée du logiciel libre en 3 mots : liberté, égalité et fraternité. [ Applaudissements ]

Liberté :

La liberté de faire des copies, de diffuser des copies, de donner des copies aux autres, aux copains, aux gens qui travaillent avec vous, aux inconnus. La liberté de faire des changements pour que le logiciel serve à vos besoins. La liberté de publier des versions améliorées telles que la société entière en reçoive les bienfaits.

Fraternité :

Avec le logiciel libre, nous encourageons tout le monde à coopérer, à aider les uns et les autres.

Egalité :

Tout le monde possède les mêmes libertés en utilisant le logiciel, il n'y a pas de situation ordinaire où un patron est tout puissant sur ce logiciel, et tout le reste du monde est complètement impuissant, tout à fait restreint en utilisant ce logiciel.

Mais comment je suis arrivé à cette idée ?

Quand je suis devenu informaticien dans les années 70, j'ai commencé à travailler au laboratoire d'intelligence artificielle du M.I.T (Massachusetts Institute of Technology). J'étais employé comme programmeur système, mon boulot était d'améliorer le système de n'importe quelle manière, de corriger des erreurs, d'ajouter des fonctionnalités, toutes les choses que nous pouvions faire pour améliorer le système de manière utile, nous le faisons.

Et c'était dans une communauté de partage du logiciel, qui comprenait plusieurs universités et parfois des coopérations avec des fabricants d'ordinateurs, comme Digital. Et, si une fois vous aviez vu sur l'écran de quelqu'un d'autre quelque chose d'intéressant, vous pouviez lui dire :

*« quel programme est-ce ? »*

et il disait

*« ah c'est le programme nouveau de quelque chose, le code source est dans ce répertoire »*

Et vous pouviez lancer le programme, et l'utiliser, vous pouviez changer le programme, ajouter d'autres fonctions, vous pouviez couper des morceaux de programmes pour les mettre dans des programmes nouveaux que vous écriviez, toutes les choses utiles à faire avec le logiciel.

Donc, la coopération passive était normale. On pouvait toujours supposer recevoir la coopération passive au moins; la coopération active pas toujours, les autres étant toujours très occupés. Mais, au moins, ils ne faisaient rien pour vous en empêcher. Par exemple, une fois le labo a eu besoin d'un assembleur pour le code de PDP 11, qui fonctionnait sur le PDP 10. Au lieu d'écrire un tel programme,

je suis allé à Harvard où ils utilisaient le système d'exploitation de Digital. Ils avaient tel assembleur qui fonctionnait sur le système de Digital. Donc, j'ai fait une copie du code source qui n'avait pas de notice de droits d'auteurs. Je l'ai apporté au M.I.T et j'ai fait des changements pour qu'il fonctionne sur notre système d'exploitation : Le système de partage temps incompatible ou ITS. J'ai ajouté des fonctions de macros dans l'assembleur, comme ça nous l'avons utilisé pendant des années. Puis, quelqu'un qui travaillait à la société BBN voulait un tel assembleur pour leur système d'exploitation qui s'appelait Tenex. Donc, il a pris le code source de ma version, il a fait des changements pour le faire marcher sur Tenex et ajouter davantage de fonctions de macros. J'avais envie de ces autres fonctions, nous avons fait une version combinée qui fonctionnait sur les deux systèmes avec des conditionnelles, et qui avait toutes les fonctions de macros. Ensuite, nous avons fait la maintenance de cette version collectivement pour les deux systèmes, pendant plusieurs années. Et, le procédé de maintenance du logiciel fonctionnait comme ça, toutes les choses qui existaient étaient toujours disponibles dans notre communauté.

Mais, une fois, j'ai vu ce que c'est d'utiliser un programme dont on ne connaît pas le code source. Ça s'est passé quand Xerox a donné au M.I.T une imprimante Laser. C'était la première que nous voyions, c'était une photocopieuse adaptée pour être une imprimante, c'était la première génération des imprimantes Laser. Et elle fonctionnait très vite, avec des points très petits. C'était vraiment bien, mais pas tout à fait fiable. Parfois, plusieurs fois par heure, elle se bloquait. Donc, elle était un peu difficile à utiliser.

Notre ancienne imprimante avait le même problème, mais l'ancienne était contrôlée par du logiciel libre, donc nous possédions le code source pour ce programme, et nous avons pu ajouter des fonctionnalités spéciales pour nous débrouiller avec ces problèmes. Il fallait attendre longtemps pour la fin de l'impression des fichiers. Une fois le processus terminé, le programme de contrôle de l'imprimante envoyait un message au PDP 10 pour indiquer que cette impression était terminée, puis le message était affiché sur l'écran de l'utilisateur pour indiquer *"ce fichier là est terminé"*. Donc, vous pouviez récupérer les papiers sans délai supplémentaire, sans attendre sans savoir si c'était terminé ou non . Puis, il y avait une fonction, que j'ai ajoutée moi-même, quand l'imprimante se plantait et était bloquée, le programme de contrôle envoyait un message *"l'imprimante est bloquée"*, et le PDP 10 affichait ce message sur les écrans des utilisateurs qui attendait à ce moment de l'impression. Si vous aviez reçu ce message, vous saviez que vous deviez y aller pour corriger le problème et vite, car vous ne vouliez pas attendre que quelqu'un d'autre le corrige. Donc 30 secondes après que le problème arrive, on voyait arriver deux ou trois personnes pour le corriger, l'imprimante marchait presque tout le temps en dépit des problèmes. Nous avons ajouté le feedback, en incluant l'utilisateur dans le système de retour, nous sommes arrivés à un fonctionnement fiable.

Avec la nouvelle imprimante, nous n'avons pas pu le faire car Xerox ne nous a pas donné le code source du programme, nous ne pouvions pas faire de changements dans le programme. Nous étions capables d'écrire ces fonctions mais nous étions bloqués volontairement par Xerox, nous étions prisonniers d'un logiciel qui a mal fonctionné pendant plusieurs années. Il fallait attendre peut-être une heure pour une imprimante capable d'imprimer en dix minutes, ou en 3 minutes. Mais tout le monde avait l'habitude d'attendre une heure, parce que nous avons appris que qu'il était improbable que l'impression se termine dans une plus courte période de temps. C'était dégueulasse. Un jour, j'ai entendu dire qu'un chercheur de l'université de Carnegie Mellon avait une copie du code source de ce programme, plus tard je suis allé à Pittsburg, à son bureau, lui demander :

*« voulez-vous me donner une copie de ce programme »*

Il m'a répondu :

*« non j'ai promis de ne pas vous le donner »*

La parole m'a manqué, je suis sorti sans dire un mot. Parce que je ne savais pas répondre à une telle

action honteuse. C'était dommage pour le labo, car nous n'avons jamais reçu le code source de ce programme, nous ne pouvions jamais corriger les problèmes, et l'imprimante a mal fonctionné pendant toutes les années où nous l'avons utilisée. Pour moi, c'était une bonne chose, d'une manière paradoxale, parce que ce chercheur n'avait pas fait cela seulement à moi, il n'a pas refusé la coopération seulement à moi, parce qu'il l'a fait à vous aussi [ rms pointe quelqu'un dans le public ], il a refusé de vous donner une copie à vous je pense [ rms pointe quelqu'un d'autre dans le public ]. Et à vous aussi je crois [ il désigne encore quelqu'un d'autre ], et je suppose vous aussi, et probablement vous tous dans cette salle. Il a trahi presque tout le monde pour recevoir quelque chose pour soi-même seulement. Il a fait la promesse de refuser aux autres ce qu'il voulait pour lui-même, ce qui est l'immoralité fondamentale : refuser aux autres quelque chose d'utile qu'on désire pour lui-même. Il a trahi tout le monde en signant un contrat de non diffusion.

C'était la première occasion où j'ai rencontré un tel contrat. Et ça a été très bon pour moi de l'avoir expérimenté en tant que victime. Parce que comme ça, je savais qu'il y avait des victimes. La plupart des gens découvrent ces contrats quand ils sont invités à les signer, et ainsi ils peuvent se dire qu'il n'y a pas de victimes :

*« je ne fais de tort à personne; si je ne le fais pas, quelqu'un d'autre le ferait à ma place; ils ne l'auront jamais de toute façon, donc ils ne perdent rien »*

toutes ces choses que quelqu'un peut se dire pour ignorer quelque chose d'immoral. Mais moi je ne pouvais pas le faire, parce que je savais ce que c'est d'être victime, je savais bien qu'il peut coûter cher aux autres de signer un tel contrat, et donc, quand quelqu'un m'a invité à signer un tel contrat, j'ai dit :

*« non, je ne peux pas le faire, ma conscience est contre, parce que si je signe ce contrat, je ferais quelque chose de mauvais pour les autres, et je ne veux pas être coupable. Donc, je me passe de ce logiciel, merci beaucoup mais non. Je le désire mais je ne peux pas l'accepter en ces termes. »*

Mais quelque chose s'est produit presque en même temps. Les deux événements ont eu lieu au début des années 80. L'autre problème est que Digital a arrêté la fabrication des PDP 10, et notre système d'exploitation était dépassé, car il était écrit en assembleur pour PDP 10. Il était complètement inutile pour tout autre ordinateur, et il n'y avait pas d'autre système libre. Donc, pour acheter une machine nouvelle et commencer à l'utiliser, il fallait obtenir un système propriétaire par l'intermédiaire d'un contrat de non-diffusion. Donc, il n'y avait pas de chemin moral pour utiliser un ordinateur, sans trahir le reste du monde.

Qu'est-ce que je pouvais faire ? J'avais un choix difficile. J'aurais pu m'adapter au monde, continuer ma carrière d'informaticien avec un système propriétaire en signant des contrats de non-diffusion et, comme ça, j'aurais pu gagner de l'argent, j'aurais pu m'amuser à écrire du code mais, à la fin, je savais que je devrais dire : *« j'ai passé ma vie à construire des murs pour la division des gens, pour séparer les gens, pour empêcher la coopération entre les gens »*

Et ce serait une vie gaspillée. C'était le choix facile. Il y avait aussi un choix un peu pénible : arrêter d'être informaticien, arrêter d'utiliser les ordinateurs. Triste, mais moral. Parce qu'il y a actuellement plusieurs millions de citoyens qui ne fabriquent pas de logiciels, qui n'utilisent pas de logiciels ou d'ordinateurs. Il est possible de vivre sans utiliser les ordinateurs. Il y a des boulots, pas de boulots où j'ai des talents spéciaux, des boulots que je pouvais faire bien sûr. Mais, cela aurait été quelque part un gaspillage de mes talents. Et aussi, comme ça je ne pouvais plus m'amuser. J'ai donc cherché un autre choix. J'avais l'idée que pour pouvoir avoir une communauté de partage du logiciel, il fallait un système d'exploitation libre, parce que la première collection de logiciels qu'il faut pour utiliser un ordinateur c'est un système d'exploitation. Avec un système d'exploitation, on peut faire beaucoup de choses avec un ordinateur. Sans système d'exploitation, on ne peut rien faire. Le système d'exploitation fait toute la différence entre pouvoir utiliser et ne pas pouvoir utiliser un ordinateur. Ma spécialité était les systèmes d'exploitation, donc si quelqu'un dans le monde pouvait écrire un tel système, c'était moi. Donc

pourquoi ne pas faire l'effort, faire un essai pour changer la situation et écrire un système d'exploitation libre. Je savais que si je réussissais, je pouvais créer de nouveau une telle communauté, et si je ratais, au moins j'aurais fait un effort qui mérite d'être fait. Donc, j'ai décidé d'écrire un système d'exploitation libre qui nous permettrait d'avoir une communauté dans laquelle la coopération est permise.

La question suivante était la conception technique du système. J'ai décidé de faire un système compatible avec Unix et suivre l'organisation d'Unix pour plusieurs raisons. Je venais de voir tout un système d'exploitation devenir inutile, inutilisable parce qu'il avait été écrit pour l'architecture d'une machine. Donc, je savais que le nouveau système devait être portable. A cette époque (1983), il n'y avait qu'un système portable et vraiment utilisé sur des architectures diverses, c'était Unix, ou au moins un système que je connaissais. Et puis, il y avait beaucoup d'intérêt pour Unix à cette époque, beaucoup d'utilisateurs, et je savais que si j'écrivais un système compatible Unix, il y aurait beaucoup de monde pour l'utiliser. Mais si j'avais pris les meilleures idées dans tous les systèmes que j'ai étudié, ou auxquels j'avais participé, et mes propres idées, j'aurais écrit un système incompatible avec les autres, et les utilisateurs m'auraient dit :

*« c'est beau mais je ne veux pas réécrire tous mes programmes, donc non merci ».*

Mon but n'était pas seulement de m'amuser à réécrire un système, mon but était de donner la liberté aux gens, et aussi de créer une communauté. Il fallait prendre le chemin le plus simple pour arriver à construire cette communauté. C'était très important d'être compatible avec Unix.

Ensuite, il fallait un nom, et comme je suis hacker, j'avais besoin d'un nom rigolo et un peu méchant. Il y avait une tradition des hackers où, quand on écrit un programme qui est compatible avec un autre programme, on peut donner au nouveau programme un nom qui est un acronyme récursif qui dit "*ce programme n'est pas l'autre, il ressemble à l'autre mais il n'est pas l'autre*". Ainsi, dans les années 60 et 70, il y avait beaucoup d'éditeurs TECO et, pour la plupart, ils étaient appelés quelque chose-TECO, mais un hacker très doué a eu l'idée d'appeler son programme TINT (Tint Is Not TECO). Et puis, par la suite j'ai écrit le premier éditeur Emacs, il y avait beaucoup d'éditeur Emacs tous compatibles, la plupart appelés quelque chose-Emacs, il y en avait un appelé FINE (Fine Is Not Emacs). Un autre SINE (Sine Is Not Emacs). Et puis EINE (Eine Is Not Emacs). Plus tard, EINE fut presque réécrit entièrement, mais pas tout à fait, et la version nouvelle était appelée ZWEI (ZWEI Was Eine Initially). Donc, je cherchais un nom de ce type, mais j'ai bientôt vu qu'il n'y a pas d'acronyme de 4 lettres qui fait un mot. Donc, j'ai cherché encore et j'ai trouvé l'acronyme de 3 lettres GNU (GNU's Not Unix). En anglais, le mot gnu est très rigolo, car le 'g' muet en début de mot est bizarre même pour l'anglais. Il y a des chansons, des blagues au sujet de ce mot. Evidemment, c'est le mot que je devais choisir. J'avais un nom, je pouvais commencer le projet, mais d'abord, je cherchais de l'argent pour payer des gens pour l'écrire. J'ai écrit une description du projet pour le montrer aux dirigeants des fabricants d'ordinateurs et ils ont répondu, parfois avec de l'intérêt au départ, mais sans donner suite. Après quelques mois, je me suis dit :

*« inutile d'attendre, je dois commencer sans argent »*

donc j'ai démissionné du M.I.T et j'ai commencé à écrire le système GNU en janvier 1984. Heureusement, le chef du labo m'a permis de continuer à utiliser les machines du labo. Je suppose que j'aurais pu trouver ailleurs des machines à utiliser, car le programme Emacs que j'avais écrit était assez renommé, mais comme ça, je n'avais pas à chercher. Je devais démissionner car parfois, les universités prennent des logiciels écrits par leurs employés pour les vendre comme produits propriétaires, et vu que mon but était de donner la liberté aux gens, si j'avais écrit des programmes et si je les avais vus convertis en produits propriétaires, l'effort aurait été totalement inutile. Avoir écrit seulement un programme populaire ne compte pas, cela ne donne pas la liberté aux gens. Et pour empêcher l'université de le faire, je devais démissionner.

Donc, j'ai commencé à écrire ; le premier programme que j'ai écrit et qui était utile en soi était l'Emacs de GNU. Je l'ai commencé en septembre 1984 et, au commencement de 1985, il était utile de premier

niveau. Il y avait des personnes qui voulaient des copies et j'avais le problème de la diffusion du programme. Bien sûr, je l'ai mis sur un serveur FTP, mais parmi les gens intéressés par ce programme, la plupart n'étaient pas sur le réseau. Donc, comment pouvaient-ils recevoir des copies. J'aurais pu dire :

*« cherchez un ami qui est sur le réseau et qui veut faire une copie pour vous »*

Je n'aurais pas eu de travail, mais je n'avais pas d'emploi et j'avais besoin de trouver d'autres moyens de gagner de l'argent, donc j'ai eu l'idée de vendre des bandes. J'ai fait une annonce sur le réseau, *« je vends les bandes pour 150 \$ »*. Les commandes ont commencé à arriver peu à peu et plus fréquemment et, en juillet, j'ai peut-être reçu 10 commandes, et je pouvais presque vivre avec ces commandes seulement.

Puis, à ce moment, je me suis dit qu'il était évident pour tout le monde que le projet GNU pouvait écrire des logiciels, que ce n'était pas seulement des paroles. J'avais l'idée que cela valait la peine de chercher à nouveau des donations. Nous avons donc créé la Free Software Foundation, en octobre 1985, pour recevoir des donations détaxées et aussi pour vendre les bandes à ma place. La plupart de l'argent venait de la vente, ce qui était surprenant. En effet, la vente de choses dont tout le monde peut faire des copies, ça paraît impossible à quelques-uns. On me dit, si le logiciel est libre, vous ne pouvez en vendre qu'un exemplaire, et celui qui a acheté cet exemplaire va vendre des copies à plus bas prix, comme ça tout le monde achètera à lui plutôt qu'à vous, vous ne vendrez jamais d'autres copies. Vous devez donc demander un prix équivalent au coût de développement pour la première copie, ce qui est évidemment impossible. Donc, l'idée est stupide. Pendant ce temps, la fondation vendait beaucoup de copies. Donc, pourquoi cela ne marche pas selon la théorie économique ?

*« On peut uniquement spéculer ».*

Mais évidemment, la théorie n'est pas tout à fait correcte. J'aime bien contredire les lois de l'économie. Parce qu'il y a d'autres personnes qui vendent des copies du même logiciel et qui demandent un prix moins élevé, c'est vrai, car nous cherchons toujours à avoir le prix le plus grand. Néanmoins, il y en a qui nous en achètent.

Je propose que vous nous en achetiez pour soutenir l'écriture du logiciel libre ou au moins quand vous achetez des CD-ROM de logiciels libres, je vous propose de vérifier si le vendeur donne des subventions à l'écriture de logiciels libres. Sinon, vous pouvez choisir un autre vendeur, pour donner l'impulsion à la subvention de l'écriture de logiciels libres. Nous vendons aussi des manuels, des livres joliment imprimés.

Pendant des années, nous avons écrit plusieurs programmes, car dans un système d'exploitation de type Unix, il faut beaucoup de composants, de programmes pour avoir tout un système. Au commencement, nous pouvions écrire n'importe quel composant car nous en avons besoin de presque tous. Plus tard, il était utile d'écrire une liste de ce qui manquait encore, la task-list de GNU. Et puis, nous avons cherché des personnes intéressées pour écrire ces programmes de manière à aboutir à un système complet. Peu à peu, nous nous approchions d'un système complet. Vers 1990 presque tout le système existait, il manquait seulement le noyau. Pas besoin de commencer par un composant particulier, car quand on a besoin de tous les composants, pourquoi commencer par celui-là ? Il faut tout écrire, donc l'ordre n'importe pas. Nous avons commencé un noyau mais j'ai fait une erreur de conception. J'ai supposé qu'en utilisant un micro-noyau, nous évitions la tâche la plus difficile, la tâche de debugging de quelque chose qui marchait directement sur la machine sans outils de correction. Mais, en fait, la conception technique que j'ai choisie était très difficile à achever, avec plusieurs processus asynchrones qui communiquent, et donc des problèmes de synchronisation, donc l'effort a duré plusieurs années. En même temps, Linus Torvalds a écrit un noyau de style classique, qui s'appelle Linux. En combinant Linux avec le système GNU, d'autres personnes sont arrivées à un système complet qui pouvait marcher sur une machine, qui était tout à fait libre, ce qui était le but du projet GNU et c'est presque le même système que GNU. C'est une version modifiée du système GNU par substitution d'un noyau par un

autre.

Question du public : Pourquoi ne pas avoir fait un système Lisp ?

Parce que sur une machine de conception normale, sans fonctions spéciales pour faire une machine Lisp, on ne pouvait pas avoir la vitesse et la détection des erreurs de type comme la machine Lisp du M.I.T. Et j'ai considéré que la vitesse était essentielle, personne n'utiliserait un système plus lent. Il fallait donc choisir un système qui permettrait d'avoir la vitesse ordinaire. Aussi, je savais que pour avoir la vitesse ordinaire dans un système Lisp, il faut se passer de la détection des erreurs de type, donc le système peut facilement se planter à chaque fois qu'il y a une erreur de type de donnée. Il faut avoir un noyau au dessous pour récupérer les erreurs dans le programme de Lisp, de façon à ce que l'on puisse continuer à utiliser le système. Moi je suis arrivé à un système de conception comme l'Unix, et j'ai donc accepté l'idée d'écrire un système comme l'Unix et d'avoir Lisp au-dessus.

[ Note sur les machines LISP: le problème du LISP est qu'il doit vérifier le type des données manipulées, ce qui se fait généralement à l'exécution (runtime), et ralentissait les programmes, à moins d'utiliser des processeurs spécialisés (machines LISP) fort onéreux qui effectuaient la vérification des types en parallèle avec l'exécution. Les fabricants de machines LISP, qui ont propriétérisé le LISP du MIT tous fait faillite face aux machines C meilleur marché, et dont le logiciel système (UNIX) était \*relativement\* "libre". On sait aujourd'hui implémenter des compilateurs LISP efficaces sans avoir à recourir à du matériel spécialisé, en utilisant des techniques d'analyse statique des programmes et d'inférence de type. Merci à Faré pour la précision. ]

[ RMS a fini de répondre à la question, et il reprend le cours de sa conférence ]

On estime à près de 10 millions d'utilisateurs de GNU/Linux, ce qui est suffisant pour estimer que le projet GNU est un succès, car ces gens sont libres. Ils possèdent les trois niveaux de liberté.

Liberté de faire des changements, des améliorations pour leurs propres besoins, comme nous ne pouvions pas le faire au M.I.T avec l'imprimante de Xerox. Si on n'a pas cette liberté, j'ai déjà expliqué les problèmes pratiques que cela entraîne mais il y a aussi des problèmes psycho-sociaux conséquents aux problèmes pratiques, délibérément imposés par un autre. Quand on veut vraiment faire son travail parce qu'on a l'idée que c'est important de le faire, et que quelqu'un vous dit « *vous ne pouvez rien faire car quelqu'un d'autre en a fermé les outils* » on change la mentalité. On arrive à une mentalité de « *bien je suis payé si je peux travailler ou non, j'ai un livre à lire. Si un jour on me permet de travailler, je recommence, mais aujourd'hui je suis payé pour lire* ». Je n'aime pas cette mentalité. Je trouve que c'est un problème social si la situation est telle, de forcer les gens à cette attitude. Et c'est comme ça si on ne peut pas travailler facilement parce qu'on vous a empêché de faire des changements nécessaires dans les logiciels que vous utilisez. C'est le niveau premier de la liberté.

Le deuxième niveau est la liberté d'aider les autres à diffuser des copies. La liberté de coopérer avec ses amis. Parce que l'acte fondamental de coopération parmi les utilisateurs des ordinateurs est de donner des copies du logiciel. Si vous possédez quelque chose qui peut aider la communauté, et que vous rencontrez quelqu'un qui en a besoin, vous le lui donnez. Quoi de plus naturel que cela. Mais il y en a qui vous appellent "pirates" si vous traitez vos amis comme des amis.

Je trouve que si on n'a pas cette liberté, il y a des problèmes pratiques. Parce que si un propriétaire arrive à imposer un système où il faut, pour chaque personne, soit payer, soit ne pas l'utiliser, il y en auraient qui acceptent de payer, mais aussi d'autres qui ne payent pas et ne l'utilisent pas.

Ainsi, le programme est partiellement gaspillé. Pour arriver à un niveau spécifique de qualité il faut réaliser le même travail, qu'il y ait mille utilisateurs ou un million d'utilisateurs. C'est exactement le même travail. Si un système propriétaire impose un découragement financier (des restrictions financières) pour l'utilisation du programme, ainsi il y aura moins d'utilisateurs pour un programme qui a nécessité le même travail ! Donc le travail est partiellement gaspillé.

C'est le problème pratique, mais le problème psychologique survient quand les gens sont insultés et que la coopération est qualifiée de piraterie. Parce que toute société est basée sur la bonne volonté, la coopération avec les autres. Et l faut toujours faire des efforts pour encourager cet esprit dans la société, parce que la nature humaine est complexe !

Plusieurs sentiments cohabitent dans la nature humaine : il y'a les bonnes volontés et les mauvaises volontés. Pour avoir une bonne société, il faut prendre des mesures pour encourager la bonne volonté. Quand j'allais à l'école, quand j'étais très petit, l'institutrice disait toujours :

*« il faut partager ses bonbons ! »*

[rires]

Maintenant, le gouvernement des États-Unis propose que les institutrices disent :

*« Oh non ! ne partage pas ! Partager c'est la piraterie ! Il ne faut jamais partager avec les autres enfants ! »*

public : [ Il y a une grosse différence, c'est que dans le logiciel libre on peut garder son bonbon et le donner à la fois ]

Oui, oui c'est beaucoup plus facile de partager le logiciel ! mais peut-être c'est pour ça qu'il veut l'empêcher, c'est totalement stupide ! Mais en tout cas, toute société où les gens peuvent avoir une bonne manière de vivre est basée sur la bonne volonté.

Bien sûr il y'a d'autres théories.

Il y avait une théorie qui disait qu'on n'a pas besoin de bonne volonté avec la planification centrale, on peut commander chaque personne de faire son boulot et tout pourrait bien fonctionner ainsi. Mais, pour la plupart, nous trouvons aujourd'hui que cette théorie a échoué.

Il y a aussi une autre théorie, concurrente, qui dit qu'il n'y a pas besoin de la bonne volonté parce que le dieu, la main invisible, peut tout commander tel que même des citoyens totalement avides doivent coopérer de manière efficace, même sans bonne volonté. Mais il faut seulement ouvrir les yeux pour voir que ce genre de société ne marche pas, et est totalement détestable. Donc, j'espère que vous êtes d'accord qu'il faut de la bonne volonté. Et quand le gouvernement et les puissants, les forces puissantes de la société, disent que la coopération est de la piraterie, qu'est-ce que cela entraîne ? Il y'a des gens qui refusent le partage des copies, disant :

*« Ah, c'est contre la loi ! Il faut toujours obéir aux lois ! »*

[grimace de dégoût, rires].

Voici un dégât psycho-social. Il y en a d'autres qui disent :

*« Je voudrais vous donner un copie, mais j'ai peur. »*

Une société avec la peur persuasive, persuadée de peur est recherchée maintenant par les propriétaires du logiciel. Parce qu'il faut que la peur soit persuasive pour empêcher les gens de donner des copies, c'est un dégât psycho-social. Egalement, il y en a qui disent :

*« {air narquois et complice} Ah oui, je ... je te di... je te donne une copie pirate ...hin hin hin... »*

[rires]

Ces gens coopèrent mais ils croient que la coopération est leur faiblesse, il croient qu'ils devraient refuser la coopération mais ils ne sont pas assez forts. C'est triste, ils font la chose correcte mais ils ont néanmoins un dégât psycho-social : être convaincu que la coopération est mauvaise.

Donc ça suffit pour le deuxième niveau.

Le troisième niveau de la liberté est la liberté d'aider à construire la société, construire la communauté

en ajoutant des fonctions aux logiciels et en publiant des versions améliorées telles que tout le monde puisse les utiliser.

C'est très important pour le développement du logiciel libre. Parce que c'est le processus à travers lequel le logiciel se développe très vite. Il y en a qui me disent, même aujourd'hui :

*« Ah, si le logiciel est libre, il manquerait de main-d'oeuvre, personne ne l'écrirait ! »*

Pourquoi ? Mais la vérité est empirique. Les faits empiriques nous disent que beaucoup de gens travaillent sur le logiciel libre. En fait, une fois qu'un programme libre est accepté, il y aura beaucoup de monde qui contribuera aux améliorations. En fait, il peut arriver que l'afflux de contributions est si grand que l'auteur ne peut presque pas l'utiliser. C'est un problème que Microsoft n'a jamais.

[rires]

Donc si cette liberté est ôtée, le problème pratique touche la vitesse de développement. Mais il y a aussi un problème psycho-social qui est la conséquence qui touche l'esprit de la coopération scientifique, qui était si fort que les scientifiques parfois pouvaient coopérer même quand leur pays était en guerre. J'ai lu qu'une fois les soldats américains qui avaient débarqué sur une île de l'océan Pacifique, pendant la deuxième guerre mondiale, ont trouvé un bâtiment avec un note :

*« Aux soldats américains. Ce bâtiment est un labo de biologie marine. Nous avons mis en ordre tous nos échantillons et rapports pour que les scientifiques américains puissent continuer nos travaux. »*

Parce que pour eux, ils travaillaient pour l'humanité. Pour la science, pour toute l'humanité. Pas pour le Japon seulement. Ils voulaient que leurs travaux soient utiles à l'humanité, quel que soit le résultat de la guerre.

Mais aujourd'hui, nous vivons dans un état de guerre civile entre des petits groupes, dans chaque pays. Où chaque groupe agit pour bloquer les autres, empêcher les autres, entraver les autres.

C'est triste.

Donc ce sont les trois niveaux de la liberté qui font la définition du logiciel libre. Un programme est libre pour vous, l'utilisateur, si vous possédez ces trois libertés.

La définition doit être formulée en termes d'un utilisateur spécifié parce que parfois un programme est libre pour quelques utilisateurs et n'est pas libre pour d'autres. Je peux l'expliquer avec un exemple, l'exemple du système de fenêtrage « X ». Le système X était écrit au M.I.T. et diffusé par le M.I.T. comme logiciel libre. Si vous aviez reçu une copie du M.I.T., vous aviez tous ces libertés.

Mais il y avait aussi des sociétés fabriquant d'ordinateurs qui avaient leur version du système Unix, et avaient l'habitude de ne jamais donner la liberté aux utilisateurs. Donc, ils ont pris le système de fenêtrage X, l'ont compilé pour leur système et ils ont diffusé des copies comme produit propriétaire avec exactement le même manque de liberté qu'avec l'Unix. Pour le même programme ou presque.

Si vous étiez utilisateur à cette époque, si vous saviez qu'il existait le problème de la liberté, et si vous aviez envie d'avoir ces libertés, vous pouviez chercher la version du M.I.T. et l'installer sur la machine. Si cette version fonctionnait, bien, vous pouviez avoir le système de fenêtrage X avec la liberté. Mais peut-être, cette version n'aurait pas fonctionner sur votre système. En ce cas, le système de fenêtrage X n'était pas pour vous, il ne restait pour vous que du logiciel propriétaire.

Pour les auteurs du système X c'était bon parce que leur but était d'avoir un succès. Mais mon but était de créer une communauté avec la liberté. Donc, pour moi, ce résultat aurait été un échec ! un échec total ! Mon travail aurait été gaspillé avec ce résultat. Donc j'ai cherché un moyen de l'interdire. Et le moyen que j'ai trouvé est la « gauche d'auteur »

[rires] « copyleft ».



L'idée est qu'au lieu de faire comme avec le système X, tout le monde peut faire des copies, tout le monde peut faire des changements, tout le monde peut diffuser des copies, même avec des changements, MAIS toute copie diffusée doit donner toutes ces libertés, les mêmes libertés que je vous donne. Vous devez donner les mêmes liberté aux autres quand vous diffusez des copies. Et comme ça, avec chaque copie, vient la liberté. Il n'y a pas de copie sans liberté. Nulle part.

L'idée de « gauche d'auteur » est un peu générale, comme l'idée de compilateur. Pour utiliser un compilateur, il faut avoir une implémentation spécifiée du compilateur. C'est pareil pour la « gauche d'auteur », il faut une implémentation.

Nous avons plusieurs implémentations de « gauche d'auteur ». C'elle que nous utilisons le plus souvent, c'est la Licence Générale Publique de GNU, mais il y en a d'autres, et pour les manuels libres nous avons des gauches d'auteur, des implémentations de « gauche d'auteur » très simples, pas comme la GPL. Parce que pour un logiciel libre il faut une documentation libre, des manuels libres aussi. C'est ce qui est très important, parce que dans la communauté actuellement, la plupart des gens ne le comprennent pas donc je veux mettre insister sur ce point. Je veux insister sur ce point, je vais insister sur ce point, parce que la documentation est partie essentielle du logiciel. Chaque programme doit arriver avec sa documentation complète. Sur l'ordinateur mais aussi, si vous le voulez, sur papier. Bien sûr, normalement il faut payer pour les copies papier, parce cela coûte quelque chose de les imprimer.

Mais... c'est du logiciel libre, pas gratuit !

Donc, aucun problème s'il faut payer pour les copies. Mais aussi, vous devez être libres de les imprimer vous-même ! Mais le point sur lequel je veux insister est que le logiciel doit être fourni avec la documentation complète. Et cette documentation doit être libre aussi. Parce que quand vous faites des copies de programmes, et quand vous diffusez des copies, il faut diffuser le manuel avec. Et quand vous faites des changements dans le programme, si vous ajoutez des fonctionnalités, ou si vous supprimez des fonctionnalités ou changez de fonctionnalités, pour faire du bon travail il faut changer le manuel aussi, pour correspondre à la version nouvelle du programme. Donc, vous devez être libre de le faire. Si le manuel ne dispose pas de la liberté de faire des changements, il n'est pas utile pour notre communauté. Donc les manuels, seulement les manuels libres, existent pour notre communauté, juste comme seulement les programmes libres existent pour notre communauté. Actuellement, il nous manque la documentation libre et pour plusieurs raisons.

Bien sûr il est plus facile d'attirer des programmeurs, des informaticiens qui savent écrire des programmes que des gens qui savent écrire des manuels. Mais aussi, il y a un grand problème pour les auteurs du logiciel, quand ils savent écrire des manuels, ils rendent les manuels propriétaires ! Ils ne contribuent pas à distribuer les manuels à notre communauté. Et c'est ce problème que je veux corriger en insistant sur ce point. Parce que si tous les programmeurs qui conçoivent le logiciel libre et qui savent écrire des bons manuels écrivaient des manuels libres, la situation serait meilleure. Donc, il faut savoir que les manuels disponibles dans les librairies, s'ils ne sont pas libres, ne sont pas bons. Ils n'appartiennent pas à notre communauté, ils ne couvrent pas les besoins. Nous avons besoin des manuels libres.

Donc, je vous propose ceci : si vous trouvez un manuel libre en vente dans une librairie, vous l'achetez. Mais si vous trouvez un manuel propriétaire, ne l'achetez pas. Parce que pour les éditeurs qui publient des manuels libres, il est bon de leur donner de l'argent en achetant des copies. Nous pouvons leur souhaiter le succès. Et le profit. Mais pour les éditeurs qui éditent les manuels propriétaires, il faut leur souhaiter l'échec.

[rires]

La fondation du logiciel libre, vend des manuels libres pour gagner de l'argent, et j'espère que les éditeurs commerciaux feront de même.

J'espère que, si vous êtes informaticien, vous écrirez du logiciel libre. Parce que si vous avez bénéficié des bienfaits de notre communauté, j'espère que vous éprouverez une dette morale, un devoir moral de contribuer à la communauté. Si vous n'êtes pas informaticien, si vous ne pouvez pas écrire du logiciel, mais si vous savez écrire des textes, peut-être pourriez-vous écrire des manuels, ce qui est même plus important. Ou si vous savez faire de l'organisation bureaucratique, nous avons aussi besoin de cela. Des organisations comme APRIL pourraient vous donner l'occasion de travailler de manière utile

[rires].

Mais, en tout cas, c'est le projet de la communauté. Le système GNU/Linux appartient à la communauté. C'est votre système, à vous tous. Et j'espère que vous chercherez à construire, à bâtir la communauté.

Récemment, j'ai fait une conférence où quelqu'un m'a demandé :

*« qu'est-ce que Microsoft pense du logiciel libre ? »*

et j'ai dit :

*« que je sache, il ne pense rien ! Il ne fait pas attention au logiciel libre »*

[rires]

Il paraît que ce n'est plus le cas. Il existe des documents où Microsoft fait des plans pour combattre le logiciel libre. Et ils ont deux stratégies :

- utiliser les protocoles et les formats de données, formats de fichiers, secrets

et également

- utiliser les brevets

contre nous. Je suppose que parfois, ces stratégies fonctionneront. Parce que nous ne pouvons pas toujours nous défendre, c'est trop difficile. Mais que pouvons-nous faire ?

Contre les protocoles et formats secrets, il faut l'ingénierie en revers (reverse engineering). Et, il y a deux moyens de le faire : on peut faire des expériences : on peut entrer des données et regarder la sortie, entrer quelque chose d'autre et regarder la sortie et enfin, on comprend les règles de transformation, ou avec les protocoles, on peut regarder les messages qui passent entre deux programmes pour comprendre enfin le protocole. Ce moyen est légal partout.

Mais aussi il y a un autre moyen, très convenable parfois, qui est la décompilation.

[rires]

Aux États-Unis la décompilation est illégale ou légale, selon quel court a pris la décision. Donc, il est difficile de savoir si l'action est légale ou non. Mais en Europe, il y a une loi, une loi explicite, qui dit que c'est légal. Par conséquent, il est très important de rassembler les ressources pour faire ce travail en Europe. Et j'espère que les organisations de logiciel libre en Europe peuvent se concentrer sur ces efforts très importants. Quand ils gagnent assez d'argent pour embaucher quelqu'un, ils peuvent choisir une tâche de décompilation pour comprendre un format secret, ou un protocole secret et puis écrire un programme compatible. Selon la directive européenne de droit d'auteur sur le logiciel.

Pour les brevets, ils sont comme des mines. On peut prendre des précautions mais parfois une mine explose. Il est certain qu'il y aura des mines qui s'exploreront sur nous. Et il y aura des choses que nous ne pourrons pas écrire, il y aura des fonctionnalités que nous ne pourrons pas offrir aux utilisateurs. Pendant peut-être quatre ans, ou dix ans, ou même vingt ans [ La période est fonction des cas particuliers ].

Qu'est-ce que nous pouvons faire ? Si nous valorisons la liberté, nous pouvons choisir un logiciel libre

qui n'offre pas cette fonctionnalité, plutôt qu'un programme propriétaire qui l'offre. Parce que la liberté est plus importante qu'une fonctionnalité.

[À quoi vous pensez ?]

Ah on peut pas savoir ! Ah oui il y en a peut-être avec les « winmodems »,

[aaaaah]

Pour faire tourner les modems pour Windows, il faut d'abord déchiffrer un langage de contrôle secret. Mais une fois que c'est fait, il y a le problème de brevet sur les vitesses au-dessus de 9600. Donc, peut-être nous ne pourrions pas supporter les plus grandes vitesses. Qu'est-ce que nous pouvons faire ? Nous pouvons acheter d'autres modems. Oui ! Si la communauté est forte, si la communauté met en valeur la liberté surtout, nous allons vendre d'autres modems. Mais si nous mettons en valeur surtout la facilité, je sais bien ce que les gens feront : ils achèteront des winmodems et les utiliseront avec les « drivers » propriétaires. Donc, c'est comme une armée. Si l'armée a l'esprit de s'unifier pour se battre, ils vaincront. Mais s'ils sont prêts à se rendre peu à peu, ils perdront.

Donc, il faut surtout diffuser l'idée de la liberté, la liberté de la coopération, qui est la plus importante des libertés parce que c'est la liberté de faire quelque chose de bon pour les autres, même plus importante que la liberté de faire quelque chose pour soi-même.

Et avec cette force morale nous pourrions enfin changer les lois des brevets. Un jour, quand nous aurons peut-être cent millions d'utilisateurs, nous pourrions peut-être changer les lois de brevets, dans plusieurs pays et oublier à jamais ce problème.

Mais si au lieu de cela les gens choisissent des programmes propriétaires qui offrent des fonctionnalités brevetées, alors nous perdrons.

Maintenant je veux présenter mon ami Saint... IGNUcius.

[ brouhaha. RMS se change. rires. applaudissements. RMS s'avance vers le premier rang, où un journaliste du Virus Informatique tape sur son ordinateur portable. Il pose la main sur l'écran. ]

« *I bless your computer my son !* »

[rires, applaudissements]

« *Je bénis... ton ordinateur, mon enfant !* »

[rires]

Je peux bénir un ordinateur et faire l'exorcisme de tout logiciel propriétaire !

[rires]

[le propriétaire de la machine palabre] C'est... c'est déjà béni ! [oui, il n'y a pas le moindre logiciel propriétaire]

Pour être un saint dans l'Église de l'Emacs, [rires] il faut vivre une vie pure.

Il faut se passer de tout logiciel propriétaire. Heureusement, être célibataire n'est pas obligé.

[rires, applaudissements]

Donc c'est bien meilleur que les autres églises !

[rires]

Donc j'espère que parmi vous, il y a des saints. Peut-être un jour vous aurez l'auréole, comme moi. [ l'auréole de rms était d'origine un plateau d'un disque dur ]

## 2. Questions du public

[ Comment est perçu GNU et puis la Free Software Foundation auprès du grand public aux États-Unis ? ]

Je ne sais pas ! Aujourd'hui je crois que le grand public ignore le projet GNU et la fondation, parce que les utilisateurs qui utilisent notre système l'appellent « Linux » et pensent qu'il a été écrit par Linus Torvalds, au début de l'année 91. C'est triste ! Je vous demande quand vous parlez du système entier de l'appeler **GNU/Linux**. Comme ça on peut citer Linus Torvalds et le projet GNU en même temps.

[applaudissements]

[je vais dire Linux une dernière fois parce que j'ai une question justement...! Que pensez-vous de l'amalgame est créé entre Linux et le logiciel libre ? C'est-à-dire que, actuellement, il y a un courant qui court et n'importe quel logiciel qui tourne sous Linux ... je vais en citer un]

sur GNU/Linux !

[rires]

[c'est là ma question, ouais c'est celui que j'ai ici, il y a pas de problème. En fait n'importe quelle application qui tourne sous Linux et propriétaire, comme Quake, ou n'importe laquelle, est considérée comme étant du logiciel libre]

C'est incorrect ! C'est stupide !

[je comprends bien, mais quelles sont les solutions que nous pourrions envisager ?]

On m'a demandé qu'est-ce que je pense de la situation où des programmes propriétaires qui tournent sur le système GNU/Linux (il a dit « Linux » mais c'était un erreur [rires] parce que il s'agit du système, pas du noyau. C'est le système sur lequel un programme tourne, pas le noyau seulement.) Mais le problème est que des programmes propriétaires qui tournent sur GNU/Linux sont considérés comme faisant partie de notre communauté ! Ce qui n'est pas vrai, parce qu'une application propriétaire porte des chaînes exactement comme un système propriétaire. Nous avons besoin d'applications libres, comme de systèmes libres. Mais qu'est-ce que nous pouvons faire pour changer cette situation ? Il faut l'expliquer aux gens comme je le fais maintenant. Vous pouvez l'expliquer aux autres. Merci !

[quelle est la position du projet GNU vis-à-vis du langage Java ?]

La position du projet GNU vis-à-vis le langage Java. Notre position vis-à-vis d'un langage est que si les utilisateurs veulent utiliser un langage, nous voulons le supporter. Je ne suis pas fana de Java, je suis fana de Lisp.

[rires, applaudissements]

Et n'importe quel langage avec une syntaxe algébrique est inférieur au Lisp.

[rires, Faré crie sa joie]

[Le système d'exploitation Hurd ?]

HURD n'est pas un système d'exploitation, HURD et Mach forment un noyau. Le système est beaucoup plus grand : il inclut Emacs, GCC, GDB, LIBC, BASH, ... Euh, mais je n'ai pas entendu la question, j'ai entendu quelque chose de Hurd, mais quelle est la question ?

[oui, aujourd'hui, dans la communauté du logiciel libre, quel rôle occupe le noyau Hurd ?]

Hurd est en développement, il ne fonctionne pas suffisamment bien pour être utilisé pour des tâches ordinaires. Nous avons sorti un système minimal qui fonctionne pour le développement du logiciel avec Hurd. Le GNU... « the Hurd », c'est le noyau de GNU. Mais il n'est pas prêt à être utilisé, seulement

pour faire des expériences sur le noyau, pour l'exploration du développement du noyau.

[charlie: j'ai une question qui est psychosociologique, qui est pas une question euh... informatique. Dans votre parcours, y'a eu donc cette problème d'imprimante de Rank Xerox qui vous a rendu...]

Pas Rank Xerox, c'était aux États-Unis !

[...d'accord, la question que je me pose...]

Et je déteste cette bande dessinée !

[d'accord] [rires]

Shhh !

[la question que je me pose c'est la question suivante : vous avez commencé votre intervention sur « Liberté, Égalité, Fraternité », vous avez parlé de la liberté et de la fraternité, on n'a pas parlé de l'égalité mais ce n'est pas gênant, par contre, votre discours ressemble énormément à la pensée des Lumières français du XVIIIème siècle, comme Condorcet, donc je voudrais savoir si dans votre formation culturelle, par delà la formation informatique, ce qui vous a influencé à défendre l'éthique d'un échange qui se situe au niveau d'une liberté contre les féodalités ?]

[brouhaha. shh!]

Je ne sais pas exactement. Je sais que j'ai eu l'idée de me battre contre la tyrannie, et l'injustice. Mais je ne savais pas le faire, je n'avais pas l'occasion de le faire. Mais j'avais toujours l'idée que c'est la vocation la plus pure d'un être humain. Où je l'ai trouvé je ne sais pas. Mais je n'ai jamais eu l'idée de m'adapter aux opinions des autres en plus de ce que je trouve moral ou éthique.

[charlie: mais il y a quand même un problème qui est formel c'est que vous avez, vous vous êtes référé à la communauté scientifique, par delà la problématique du logiciel, puisque l'informatique est l'une des activités humaines de la seconde moitié du XXème siècle dominantes, il y a quand même un danger pour la pérennité des connaissances ]

Pérennité ? peut-être... je crois que je ne comprends pas... je ne peux pas comprendre cette question [rires. shh.] Parce que je ne suis pas éduqué comme philosophe. Et pour la plupart...

[charlie: alors quels sont vos rapports avec les scientifiques qui ne s'intéressent pas à l'informatique et qui voient dans votre le combat le même combat pour que puissent exister encore des réseaux de scientifiques ?]

Je n'ai pas beaucoup de rapports avec eux. Parfois quand je rencontre des scientifiques, je leur dis

*« ah ! vous avez trouvé cette théorie, c'est merveilleux, je vous admire ! »*

et ils me disent

*« vous avez écrit l'Emacs ! »*

[rires]

ce qui m'étonne un peu parce que pour moi, avoir écrit un programme c'est presque rien, par rapport aux théories de la physique. Comme réussite intellectuelle, la mienne est très petite, et très facile. Les réussites en physique sont très difficiles et très grandes.

[charlie: mais peut-être que leur problème est identique...]

Peut-être... J'ai l'idée...

[...à celui que les informaticiens...]

Je vois l'analogie entre des questions mais je ne fais rien en ce sens parce que j'ai mon champ où il y a trop de travail pour moi. Donc ...

[Mais c'est important, il y a un philosophe qui s'appelle Ivan Illic?? qui a dit que ... boîte noire ... ! Il y a un philosophe qui est né à Vienne qui s'appelle Ivan Illic ?? qui dénonce le progrès enfin... dans la structure actuelle parce que les choses sont des boîtes noires, les gens sont manipulés par les objets qu'ils achètent en quelque sorte.]

Je suis d'accord avec lui. Je n'ai jamais lu ce livre. Il est possible que j'aie entendu parler de ces idées. Les idées sont répétées parmi la société. Peut-être qu'elles me sont parvenues et peut-être m'ont-elles influencé.

[non mais ce que je pense c'est que ...]

Mais je suppose que cette question est infructueuse.

[si ! parce que...]

Parce que je n'ai rien à dire de plus. Quel dommage !

[rires, applaudissements]

C'est dommage, mais je vais demander d'autres questions, où je peux répondre de manière un peu utile

[(le même) vous êtes trop modeste, parce que ...] [brouhaha, protestations, cris « la liberté ! »]

J'applique ces idées dans mon champ.

[oui... mais c'est ça qui...]

je suppose que tout l'assistance connaît ces idées déjà, ce sont de bonnes idées, mais pourquoi poser cette question parce que je ne peux rien dire de spécial

[(le même) c'est là le danger parce que je pense que dans différents domaines il y a des apôtres ...]

Bien ! peut-être pouvez VOUS faire des travaux sur la liaison entre eux et nous. Peut-être pouvez VOUS le faire. Il est inutile de ME proposer de le faire, je n'ai plus de temps !

[charlie: alors pourquoi avoir créé au début, une GPL aussi pour les musiciens ?] [ Nb : rms n'a jamais écrit une GPL pour la musique ]

Parce que la musique c'est autre sujet, et les besoins dans ce champ sont différents. Le logiciel est quelque chose de fonctionnel. On utilise le logiciel pour faire un travail ou pour s'amuser. Mais on ne s'amuse pas en regardant le logiciel. On s'amuse par le résultat du fonctionnement du logiciel. Donc, les logiciels pour s'amuser et les logiciels sérieux, tous les deux sont fonctionnels. Un peu comme les recettes.

Avant les ordinateurs, ce qui était le plus proche du logiciel c'était les recettes. Vous savez que les gens qui cuisinent partagent toujours des recettes et changent des recettes. Il est très important dans la vie des gens qui cuisinent, de partager, de changer des recettes. Donc, c'est la même chose pour le logiciel : la liberté de diffuser une version modifiée est très importante. Mais pour la musique, n'est pas la même chose, parce que la musique ne fonctionne pas. On perçoit la musique, mais on n'utilise pas la musique. Le besoin fonctionnel de faire des changements dans un morceau de musique ou dans un enregistrement de musique n'existe pas de la même façon.

Il y a bien sûr le processus « folk » de changement peu à peu de la musique ou d'autres oeuvres culturelles. On peut discuter de l'importance de ce processus mais le besoin très fort de faire des changements et de donner ces changements aux autres n'existe pas dans le domaine des oeuvres culturelles. Donc la question morale n'est pas la même. Elle est en partie semblable, parce que la liberté de donner des copies aux autres est très importante. Si quelqu'un vous en empêche ou vous interdit de faire une copie pour votre ami, c'est détestable. C'est de la tyrannie d'imposer un tel système, il faut des châtiments draconiens pour convaincre les gens ne pas partager la musique. Mais vu qu'il n'y a pas le même besoin de diffuser des versions modifiées, le cas est beaucoup plus simple, on peut avoir un

système où diffuser des versions exactes, des copies exactes, est permis, mais diffuser des copies modifiées est interdit. Et comme ça on peut avoir un système où les noms des musiciens sont toujours gardés sur les copies et comme ça on pourra avoir, je crois dans vingt ans, il sera possible pour les musiciens de gagner leur vie en ayant sur leur machine Web une boîte qui dit :

*« Cliquez ici pour me donner un dollar. Si vous aimez mes morceaux, si vous aimez ma musique, que vous venez d'écouter, cliquez ici pour me donner un dollar. »*

Un dollar, c'est pas beaucoup. Il y aura beaucoup de gens qui donneront un dollar. Et je crois que les musiciens qui sont ainsi acceptés, dont les oeuvres sont écoutées, recevront l'argent. Sans que personne ne soit forcé, à travers un système volontaire.

Ah ! il y a quelqu'un, par là, qui n'a rien demandé.

[la question a peut-être été déjà posée, donc si elle a été posée, ne pas y répondre, actuellement il y a des éditeurs, qui disaient qu'ils pensaient mettre leurs CD-ROMs ou leurs ouvrages, entre guillemets, en « libres d'accès et libres de droits » et il y a certains écrivains qui commencent à penser à mettre leurs ouvrages sous GPL. Il y a actuellement des conflits énormes avec des droits, avec l'AMI européen et mondiaux et autour des droits d'auteur. Est-ce que la GPL...]

Je ne connais pas ces conflits, mais je ne propose pas la GPL pour autre chose que des programmes. Elle peut fonctionner pour d'autres choses. Mais ce n'est pas le genre de gauche d'auteur que je propose, que je suggère pour d'autres types d'écriture comme les manuels ou les essais ou les articles.

[ma question est brève. C'est est-ce que dans ce cas là la GPL pour les informaticiens est quelque chose qui entre guillemets peut être un « cadre », est-ce que ça vous intéresserait de participer au développement, plus seulement de la GPL, mais à son concept, pour d'autres choses, qui pourraient, entre guillemets avec l'avenir et les accélérations des réseaux, prendre une valeur qui pourrait être importante ?]

Je n'ai pas entendu la question. Peut-être quelqu'un peut la résumer.

[il a demandé si vous participeriez à l'extension du concept de la GPL à d'autres domaines ?]

Je l'ai déjà fait !

[(le même) oui !]

J'ai utilisé une forme de gauche d'auteur pendant presque 15 ans pour des manuels, qui ne sont pas du logiciel, qui sont de l'écriture. Mais pour les manuels, j'ai expliqué pourquoi il est important d'autoriser la publication de versions modifiées. Mais nous le ferons avec un autre forme de gauche d'auteur, beaucoup plus simple que la GPL. Parce que le cas n'est pas le même, et une forme de gauche d'auteur plus simple peut fonctionner pour les manuels, mais ne peut pas fonctionner pour le logiciel. Pour d'autres types d'écriture, comme les articles sur la philosophie ou la politique, ou notre position légale ou comme l'histoire de notre projet, je ne donne pas l'autorisation de faire des versions modifiées parce que, dans ce cas il n'y a aucune raison d'avoir cette liberté. Elle existe dans le cas d'un manuel de documentation du logiciel, parce qu'il faut le changer avec le logiciel.

[ça, c'est ce qu'en france, on appelle le « droit moral de l'auteur sur son oeuvre » dans le milieu littéraire, ce que vous décrivez là]

beuh, non ! de façon légale c'est pas la même chose mais c'est une idée peut-être proche, dans l'esprit.

[est-ce que la fondation a encouragé des travaux ou qui sont en cours pour unifier les distributions de GNU/Linux ?]

Non, pas spécialement. Nous sommes d'accord avec ce projet mais ce n'est pas nous qui l'avons fait.

[est-ce qu'il y a une position par rapport à la licence publique de Netscape ?]

Cette licence, la licence de Netscape, rend libre le logiciel mais a deux problèmes, pour lesquels je suggère de ne pas l'utiliser :

- un problème est que cette licence n'est pas une gauche d'auteur. il est très facile d'ajouter des fonctionnalités au programme avec du code propriétaire. Netscape le voulait comme ça, ce n'était pas par hasard, mais c'était fait volontairement, aussi je pense que c'était une mauvaise idée.

- l'autre problème est que cette licence est incompatible avec la licence GPL. Donc, si un programme A est sorti sous la GPL, et un programme B est sorti sous la licence de Netscape, il est illégal de combiner les deux.

Donc, je vous propose de vous passer de la licence de Netscape.

Il est peut-être question que Netscape sorte son logiciel sous les deux licences en même temps. Et je l'espère.

[est-ce qu'on peut étendre le concept d'informatique libre, de logiciel libre plutôt, à l'informatique en entier, et notamment aux services, par exemple le service de consultation, pour trouver des solutions techniques à des projets,...]

Je ne vois pas d'analogie, parce que normalement, si quelqu'un vous pose une question et vous paye pour répondre aux questions, normalement il n'est pas interdit de le dire aux autres, donc c'est libre, l'information que vous donnez est libre. Il n'y a pas de problème, pas besoin d'un mouvement de service libre, parce que c'est presque toujours libre. Ce qui est bien. C'est bien si on n'a pas besoin de se battre.

[ Vous avez encouragé les utilisateurs de la communauté du logiciel libre à promouvoir le logiciel libre. Est-ce que vous considérez que l'action politique ou du moins la sensibilisation des décideurs, des hommes politiques, est importante ?]

Oui c'est important. Peut-être que maintenant c'est possible, notre communauté est assez grande pour pouvoir commencer à le faire. Parce que vous savez que l'argent c'est bien pour convaincre les décideurs gouvernementaux. Et pour discuter de la question avec de l'argent, il faut beaucoup beaucoup beaucoup de gens. Peut-être arrivons-nous à avoir assez de gens pour commencer. Si vous connaissez des députés, peut-être pouvez-vous commencer à parler avec eux de ce sujet. Oui, mais je crois que peut-être d'ici 5 ans nous aurons la possibilité de gagner l'exemption de brevets.

[Si je ne me trompe pas, il y a une version de projet GNUStep pour Windows NT. Pourquoi développer un logiciel libre pour un système propriétaire ?]

Euh, je ne trouve pas que c'est important, mais je ne suis pas contre. Il y a deux questions : supposons qu'un programmeur développe un programme libre seulement pour un système propriétaire. Je le trouve un peu idiot. Pourquoi le faire ? Exactement ! Si on veut la liberté, il faut utiliser un système libre. Donc il est triste que quelqu'un fasse beaucoup d'efforts pour écrire un programme libre qui ne marche pas sur les systèmes libres.

Et l'autre question : si vous avez écrit un programme pour un système libre, et que vous voulez maintenant le faire marcher également sur Windows, en ce cas je dis :

*« si vous voulez le faire, pourquoi pas ? c'est à vous ! »*

je ne fais pas d'efforts moi-même pour faire tourner mes logiciels sur Windows. Parce que cela ne m'intéresse pas beaucoup. C'est un détour, mais s'il y en a d'autres qui veulent le faire, je ne suis pas contre. Je l'accepte.

[FIN DE LA BANDE, ensuite encore 1 heure de questions/réponses environ]