

(7) Par la suite, nous avons appris la distinction entre «logiciel libre» et «graticiel» («freeware»). Le terme «graticiel» signifie logiciel que vous êtes libre de redistribuer, mais généralement, vous n'êtes pas libre de l'étudier ou de changer le code source; donc la plupart d'entre eux ne sont pas des logiciels libres. Voir la page Termes prêtant à confusion, que vous devriez éviter pour plus d'explications.

Pour les questions et requêtes relatives à la FSF & GNU: gnu@gnu.org.

Copyright © 1985, 1993, 2003, 2005 Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

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

Le Manifeste GNU.

Richard Stallman

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

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

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

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

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

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

Nous essayerons de supporter UUCP, MIT Chaosnet, et les protocoles de l'Internet pour la communication.

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

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

Pourquoi j'ai dû écrire GNU

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

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

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

Disponibilité de GNU

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

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

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

Notes

(1) Le choix des mots ici était irréfléchi. L'idée était que personne n'aurait à payer l'autorisation d'utiliser le système GNU. Mais cela n'était pas clair, et les gens ont souvent compris que les copies de GNU devaient toujours être distribuées peu chères ou gratuitement. Cela n'a jamais été l'intention; plus tard, le manifeste mentionne pour les sociétés la possibilité de fournir un service de distribution rentable. Par la suite, j'ai appris à bien faire la distinction entre «free» dans le sens de libre, et «free» dans le sens de gratuit. Le logiciel libre est un programme que les utilisateurs ont la liberté de distribuer et de modifier. Certains utilisateurs peuvent obtenir des copies gratuitement, tandis que d'autres les paieront et si cela peut rapporter de quoi financer l'amélioration de programmes, tant mieux. L'important est que toute personne détenant une copie a le droit de l'utiliser en collaboration avec d'autres.

(2) Voilà un autre endroit où je n'ai pas fait la distinction entre les deux définitions de «libre». La phrase telle quelle n'est pas fautive, vous pouvez obtenir des copies de logiciels GNU gratuitement, par vos amis ou par l'Internet. Mais elle suggère effectivement la mauvaise idée. [N.d.t. - En français, la distinction entre «libre» et «gratuit» est évidente.]

(3) De telles sociétés existent actuellement.

(4) Depuis dix ans, la Free Software Foundation a levé la plupart de ses fonds à travers un service de distribution, bien qu'elle soit une association plutôt qu'une société. Vous pouvez commander des logiciels, des livres, ... à la FSF.

(5) Un groupe de sociétés a réuni des fonds vers 1991 pour financer la maintenance du compilateur C de GNU.

(6) Dans les années 80, je n'avais pas encore réalisé à quel point il était déroutant de parler du «problème» de la «propriété intellectuelle». Ce terme est évidemment partial; plus subtil dans le fait qu'il mélange diverses lois disparates qui traitent de problèmes très différents. De nos jours, je presse les gens à rejeter totalement le terme «propriété intellectuelle», de peur qu'il ne conduise d'autres personnes à supposer que ces lois forment une solution cohérente. La façon d'être clair est de parler de brevets, de droits d'auteur, et de marques déposées séparément. Voir des explications plus détaillées sur la manière dont ce terme sème la confusion et le parti-pris.

«Les programmeurs doivent bien gagner leur pain.»

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

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

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

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

Les utilisateurs ayant des besoins en commun, peuvent créer des groupes d'utilisateurs et payer des cotisations. Un tel groupe pourrait faire appel à une société de développement pour écrire les programmes spécifiques pour ses membres.

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

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

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

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

Les conséquences:

* La communauté des utilisateurs soutient le développement des logiciels.

* Cette communauté décide du niveau du soutien nécessaire.

* Pour les utilisateurs qui se soucient de quels projets profitent de leur participation, ils pourront les choisir eux-mêmes.

À terme, rendre les programmes libres est un pas vers le monde d'après pénurie,

Pourquoi beaucoup de programmeurs veulent contribuer.

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

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

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

Comment vous pouvez contribuer

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

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

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

J'ai trouvé de nombreux programmeurs impatients de contribuer à mi-temps pour GNU. Pour la plupart des projets, un tel travail distribué à temps partiel serait très difficile à coordonner; les diverses parties codées indépendamment ne fonctionneraient pas ensemble. Mais ce problème n'existe pas dans le cas du projet de remplacement d'Unix. Un système Unix complet contient des centaines d'utilitaires, chacun étant documenté séparément. La plupart des spécifications des interfaces sont

déterminées par la compatibilité avec Unix. Si chaque collaborateur peut écrire un remplacement compatible pour un seul utilitaire Unix et l'intégrer proprement à la place de l'original sur un système Unix, il s'ensuit que ces utilitaires fonctionneront ensemble sans problème. Même en faisant quelques concessions à la loi de Murphy qui créera quelques problèmes inattendus, l'assemblage de ces composants sera une tâche réalisable. (Le noyau demandera quand même une communication plus soutenue et sera réalisé par un petit groupe.)

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

Pourquoi tous les utilisateurs en bénéficieront.

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

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

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

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

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

Les mesures pour faire payer les licences des programmes et de leurs copies génèrent toujours un coût important pour la société en général, à cause des mécanismes nécessaires pour calculer combien (c'est-à-dire quels programmes) chacun doit payer. Et il faudrait un État policier pour appliquer parfaitement ces mesures. Prenons une

que cela fonctionne toujours de cette façon. Si les coureurs oublient le pourquoi de la récompense, et deviennent obsédés par la victoire, quelles que soient les méthodes employées, ils risquent de trouver d'autres stratégies telles qu'agresser les autres concurrents. Si tous les coureurs s'engageaient dans un combat, ils finiraient tous en retard.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

«La compétition permet de mieux faire les choses.»

Le paradigme de la compétition est une course: en récompensant le vainqueur, nous encourageons tout le monde à courir plus vite. Quand le capitalisme fonctionne réellement de cette façon, tout marche bien; mais ses partisans ont tort s'ils pensent

station orbitale, où l'air doit être fabriqué à un coût important: facturer chaque litre inspiré peut être justifié, mais porter un masque/compteur toute la journée et toute la nuit est intolérable même si on a de quoi payer la facture. Et les caméras de surveillance placées partout pour vérifier que vous ne retirez jamais le masque/compteur seraient inacceptables. Il vaut mieux financer la fabrication de l'air avec une taxe par personne et se débarrasser des masques.

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

Quelques objections facilement contrées aux objectifs de GNU

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

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

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

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

Si votre société a besoin d'un support fiable, la seule façon est d'avoir toutes les sources et tous les outils nécessaires. À partir de là, vous pouvez engager n'importe quelle personne qualifiée pour régler les problèmes. Vous n'êtes pas à la merci d'une seule personne. Avec Unix, le prix des sources rend cette solution inabordable pour la plupart des sociétés. Avec GNU ce sera facile. Il serait éventuellement concevable qu'il n'y ait personne de disponible, mais les modalités de distribution ne sont pas responsables de ce problème. GNU ne propose pas de régler tous les problèmes, mais seulement quelques-uns.

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

De tels services pourraient être proposés par des sociétés qui ne font que des cours d'initiation et des réparations. S'il est vrai que les utilisateurs préfèrent dépenser de l'argent pour un logiciel intégrant un service après-vente, ils seront aussi d'accord pour payer simplement le service, ayant obtenu le logiciel gratuitement. Les sociétés de service se feront concurrence sur la qualité et le prix de leurs prestations. Les

utilisateurs ne seront pas limités à une société particulière. En même temps, ceux d'entre nous qui n'ont pas besoin du service pourront utiliser le logiciel sans payer le service.

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

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

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

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

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

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

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

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

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

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

Il n'y a rien de mal à vouloir être payé pour son travail, ou à chercher à augmenter ses revenus, tant que l'on n'utilise pas de méthodes destructives. Mais les méthodes pratiquées dans le domaine du logiciel sont fondées sur la destruction. Extraire de l'argent aux utilisateurs d'un programme en limitant son utilisation est destructeur, car ces restrictions réduisent l'utilité du programme. Ce qui à son tour réduit la richesse apportée par ce programme à l'humanité. Quand le choix de limiter est délibéré, les conséquences néfastes qui en découlent sont de la destruction délibérée.

La raison pour laquelle un bon citoyen ne doit pas utiliser de telles méthodes destructrices pour augmenter sa richesse personnelle est que si tout le monde faisait de même, il y aurait un appauvrissement général dû à la destruction mutuelle. C'est ce que l'on appelle la morale kantienne, ou la Règle d'or. Puisque je n'apprécie pas les conséquences qui adviennent si tout le monde fait de la rétention d'informations, je ne dois pas trouver acceptable qu'un individu le fasse. Plus précisément, le désir d'être récompensé pour sa création ne justifie pas que l'on prive le monde en général de toute ou partie de cette créativité.

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

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

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

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

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

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