Manifeste GNU (The GNU Manifesto)

Richard Stallman

Texte original : http://www.gnu.org/gnu/manifesto.html
Traduction française : Hache 1999 (révisée en 2002 et 2008)

Le Manifeste GNU *a été écrit par Richard Stallman au commencement du projet GNU et faisait office d'appel à la participation. Dans les quelques années qui ont suivi sa rédaction, des petites mises à jour ont été effectuées afin de rendre compte de l'avancement du projet. Aujourd'hui, cependant, il semble préférable de le donner en l'état, tel que l'ont découvert la plupart de ses lecteurs.*

Depuis la publication initiale, nous nous sommes rendus compte que le texte ici ou là prêtait à des malentendus que des formulations différentes permettraient sans doute d'éviter. Des notes de bas de page ajoutées en 1993 aident à préciser le sens de ces passages.

Pour connaître les logiciels GNU actuellement disponibles, consultez notre site web et notamment notre liste de logiciels. Pour savoir comment apporter votre contribution au projet, consultez http://www.gnu.org/help/help.fr.html.

Qu'est-ce que GNU ? GNU n'est pas Unix !

« GNU », qui est l'acronyme de « GNU's Not Unix » (GNU n'est pas Unix), est le nom du système d'exploitation complet et compatible Unix que je suis en train de développer pour pouvoir le donner gratuitement à quiconque en aurait l'usage.(1) Plusieurs personnes, bénévoles comme moi, m'aident dans cette tâche. Nous avons grand besoin de contributions sous la forme de temps, d'argent, de programmes et d'équipement.

Pour l'instant, nous disposons d'un éditeur de texte Emacs programmable en Lisp, d'un débogueur symbolique, d'un générateur d'analyseurs syntaxiques compatible avec Yacc, d'un éditeur de liens et d'environ trente-cinq utilitaires. Un interpréteur de commandes est presque terminé. Un nouveau compilateur C portable et optimiseur s'est compilé lui-même et pourrait être disponible dès cette année. Un noyau initial existe mais de nombreuses autres fonctions doivent y être ajoutées pour émuler Unix. Lorsque le noyau et le compilateur seront terminés, il sera possible de distribuer un système GNU permettant le développement de logiciel. Notre logiciel de composition sera TeX, mais un Nroff est aussi en chantier. Nous utiliserons également le système libre et portable X Window. Par la suite, nous ajouterons un interpréteur portable Common Lisp, une version du jeu Empire, un tableur et des centaines d'autres choses, ainsi qu'une documentation en ligne. À terme, nous espérons fournir tous les éléments utiles normalement inclus dans un système Unix, et plus encore.

GNU permettra d'exécuter des logiciels Unix, sans pour autant être identique à Unix. Nous tirerons profit de notre expérience d'autres systèmes d'exploitation pour apporter toutes les améliorations qui sembleront souhaitables. En particulier, nous prévoyons d'utiliser des noms de fichier plus longs, des numéros de version de fichier, un système de fichiers tolérant aux pannes, le complètement automatique des noms de fichiers (peut-être), la gestion d'affichage indépendante du terminal, et peut-être, finalement, un système de fenêtrage fondé sur Lisp grâce auquel plusieurs programmes Lisp ou autres logiciels Unix pourront partager un même écran. C et Lisp pourront tous deux être utilisés comme langages de programmation système. Nous tâcherons de prendre en charge les protocoles de communication UUCP, Chaosnet et Internet.

Dans un premier temps, GNU est destiné aux ordinateurs 68000/16000 avec mémoire virtuelle, parce que ce sont les ordinateurs sur lesquels il sera le plus facile de le faire fonctionner. L'effort supplémentaire nécessaire à son déploiement sur de plus petits ordinateurs pourra être fait par quelqu'un qui souhaite l'utiliser sur les ordinateurs concernés.

Pour éviter des confusions malheureuses, merci de prononcer le « G » dans le mot « GNU » quand il s'agit du nom de ce projet. (N.d.T. : En anglais, le nom commun « gnu », qui désigne le gnou, se prononce comme le français « noue ».)

Pourquoi je dois écrire GNU

À mon sens, la règle de réciprocité exige que je partage les programmes que j'apprécie avec les personnes qui apprécient ces mêmes programmes. Les éditeurs de logiciels, cherchant à diviser pour régner, obtiennent des utilisateurs qu'ils renoncent à tout partage. Je refuse de rompre de cette manière la solidarité qui me lie aux autres utilisateurs. Je me sentirais coupable de signer un accord de non-divulgation ou un accord de licence. Pendant des années, j'ai IJuvré au sein de l'Artificial Intelligence Lab (N.d.T. : du MIT) pour résister à de telles tendances et à d'autres attitudes déplaisantes, mais ils ont fini par dépasser les bornes, et il ne m'était plus possible de demeurer dans une institution où de telles choses étaient faites pour moi contre ma volonté.

Afin de pouvoir continuer à utiliser des ordinateurs sans déshonneur, j'ai décidé de constituer un ensemble de logiciels libres suffisant pour que je puisse me passer de tout logiciel non libre. J'ai démissionné du Laboratoire d'IA afin de priver le MIT de tout prétexte légal pour m'empêcher d'offrir GNU gratuitement.

Pourquoi GNU sera compatible avec Unix

Je ne considère pas Unix comme un système idéal, mais ce n'est pas non plus un trop mauvais système. Les fonctions essentielles d'Unix sont bonnes, et je pense pouvoir ajouter ce qui manque à Unix sans les gâter. De plus, adopter un système compatible avec Unix serait commode pour de nombreuses personnes.

Conditions sous lesquelles GNU sera offert

GNU n'est pas dans le domaine public. Chacun sera autorisé à modifier et à reproposer GNU, mais les personnes qui l'offrent ne seront pas autorisées à limiter la façon dont il peut être reproposé. En d'autres termes, des modifications propriétaires ne seront pas autorisées. Je veux garantir que toutes les versions de GNU restent libres.

Pourquoi de nombreux autres programmeurs souhaitent participer à ce projet

J'ai trouvé de nombreux autres programmeurs intéressés par GNU et désireux d'apporter leur aide.

Beaucoup de programmeurs sont mécontents de la commercialisation des logiciels système. Il est possible que cette commercialisation leur permette de gagner plus d'argent, mais elle les met dans une relation de conflit plutôt que de camaraderie avec les autres programmeurs. L'amitié entre programmeurs se manifeste essentiellement dans le partage de programmes ; les accords commerciaux généralement utilisés aujourd'hui interdisent donc aux programmeurs de se traiter mutuellement en amis. L'acheteur du logiciel doit choisir entre l'amitié et le respect de la loi. Naturellement, beaucoup décident que l'amitié est plus importante. Mais ceux qui prennent la loi au sérieux se sentent souvent mal à l'aise face à ce choix. Ils deviennent cyniques et pensent que programmer n'est qu'une façon comme une autre de gagner de l'argent.

En développant et en utilisant GNU plutôt que des logiciels propriétaires, nous pouvons conserver une attitude amicale envers chacun tout en respectant la loi. De plus, GNU sert d'inspiration et de point de ralliement pour tous ceux qui voudraient nous rejoindre dans cette attitude de partage. Nous pouvons en tirer un sentiment d'harmonie qu'exclurait l'utilisation de logiciels non libres. Pour à peu près la moitié des programmeurs que je rencontre, il s'agit d'une satisfaction importante que l'argent ne peut pas remplacer.

Comment vous pouvez participer au projet

(Aujourd'hui, pour connaître les tâches de développement auxquelles vous pouvez participer, consultez la liste des tâches GNU. Pour connaître d'autres façons de participer au projet GNU, 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 auxquelles vous pouvez vous attendre si vous faites don d'ordinateurs est que GNU fonctionnera très bientôt sur ceux-ci. Les ordinateurs doivent être complets, prêts à l'emploi, adaptés à l'utilisation en zone résidentielle, et ne pas nécessiter de systèmes de refroidissement ou d'alimentation sophistiqués.

J'ai trouvé quantité de programmeurs enthousiastes à l'idée de travailler à temps partiel dans le cadre du projet GNU. Pour la plupart des projets, il serait très difficile de coordonner ce type de contributions décentralisées et à temps partiel : les différents morceaux produits séparément ne créeraient pas un tout cohérent. Mais dans le cas particulier de la tâche consistant à remplacer Unix, le problème ne se pose pas. Un système Unix complet contient des centaines d'utilitaires, et chacun d'entre eux a sa propre documentation. La plupart des spécifications d'interface sont définies par la compatibilité avec Unix. Si chaque participant peut créer une version de remplacement compatible d'un seul utilitaire Unix, et la faire fonctionner correctement à la place de la version originale au sein d'un système Unix, alors les utilitaires créés fonctionneront correctement lorsqu'ils seront assemblés. Même en admettant, loi de Murphy oblige, quelques problèmes inattendus, l'assemblage de ces composants restera une tâche réalisable. (Le noyau quant à lui exigera une communication plus étroite et sera confié à un petit groupe soudé.)

Si je reçois des dons financiers, je serai peut-être en mesure d'employer quelques personnes à temps plein ou à temps partiel. Le salaire sera peu élevé par rapport au marché, mais je recherche des personnes pour qui l'esprit de communauté est aussi important que les gains matériels. Il s'agit de permettre à des personnes dévouées de consacrer toute leur énergie au projet GNU en leur évitant d'avoir à gagner leur vie par d'autres moyens.

Pourquoi tous les utilisateurs d'ordinateur en profiteront

Une fois que GNU aura été créé, de bons logiciels système seront accessibles à tous gratuitement, comme l'air que nous respirons.[2]

L'enjeu dépasse de beaucoup la simple économie du prix d'une licence Unix. Il s'agit aussi d'éviter de répéter inutilement le travail de programmation système. Tout le travail peut alors se porter directement sur le progrès du système.

Le code source complet du système sera accessible à tous. Du coup, un utilisateur qui aurait besoin d'une version modifiée du système aura toujours la liberté de réaliser lui-même les modifications ou d'engager n'importe quel programmeur ou société disponible dans ce but. Les utilisateurs ne seront plus à la merci d'une personne ou d'une société possédant le code source et seule à pouvoir apporter des modifications.

Les écoles pourront offrir un environnement d'une plus grande valeur éducative en encourageant les étudiants à étudier et à améliorer le code du système. Le laboratoire informatique de Harvard avait pour politique de n'installer aucun logiciel sur le système si son code source n'était pas publiquement disponible, et appliquait cette politique en refusant effectivement l'installation de certains logiciels. Cela m'a beaucoup inspiré.

Enfin, les coûts consacrés à déterminer la propriété du logiciel système et les limites de son utilisation seront supprimés.

Les dispositions visant à faire payer les gens pour l'utilisation d'un logiciel, y compris l'octroi de licences pour l'utilisation de copies du logiciel, représentent un coût social énorme à cause de la difficulté qu'il y a à déterminer quelle partie (c'est-à-dire quels programmes) une personne doit payer. De plus, seul un État policier serait en mesure de forcer tout le monde à les respecter. Imaginez une station spatiale où la production de l'air est très coûteuse : faire payer chaque personne par litre d'air respiré peut sembler justifié, mais porter jour et nuit un masque mesurant l'air respiré n'est pas tolérable même si on a les moyens de payer la facture. Et les caméras de surveillance installées partout afin de vérifier que personne n'ôte jamais son masque ne sont pas tolérables non plus. Mieux vaut financer la centrale d'air par une taxe personnelle et renoncer aux masques.

Pour un programmeur, copier tout ou partie d'un logiciel est aussi naturel que de respirer, et aussi productif. Il devrait pouvoir le faire aussi librement.

Quelques objections facilement réfutables au projet GNU

« Si le logiciel est gratuit, personne n'en voudra, parce que l'on ne pourra compter sur aucun service. »

« Il est nécessaire de faire payer le logiciel pour financer le service. »

Si les gens préfèrent payer pour obtenir GNU avec un service plutôt que de recevoir GNU gratuitement sans service, alors une entreprise qui fournirait uniquement le service à des personnes ayant obtenu GNU gratuitement devrait être rentable.(3)

Il faut distinguer entre, d'une part, le service qui prend la forme d'un réel travail de programmation et, d'autre part, le service qui consiste en un simple accompagnement. Pour le travail réel, il est impossible de se fier à un fournisseur. Si votre problème ne se pose pas à un nombre suffisant de personnes, le fournisseur vous enverra paître.

Si votre entreprise a besoin d'un service fiable, la seule solution est de disposer de tous les fichiers sources et de tous les outils nécessaires. Vous pouvez alors engager toute personne disponible pour résoudre votre problème, et n'êtes pas à la merci d'un individu quelconque. Avec Unix, le prix des fichiers sources interdit cette approche à la plupart des entreprises. Avec GNU, ce ne sera pas un problème. Il peut toujours arriver qu'aucune personne qualifiée ne soit disponible, mais ce problème n'est pas imputable aux modalités selon lesquelles le logiciel est offert. GNU ne résout pas tous les problèmes du monde, seulement certains d'entre eux.

Pendant ce temps, les utilisateurs non spécialistes ont besoin qu'on les accompagne et qu'on fasse pour eux des choses qu'ils pourraient en principe facilement faire eux-mêmes s'ils savaient comment s'y prendre.

De tels services pourraient être fournis par des prestataires spécialisés dans les services d'accompagnement et de dépannage. Si effectivement les utilisateurs préfèrent payer une certaine somme pour obtenir un logiciel avec un service après-vente, alors ils seront d'accord d'acheter le service après avoir reçu le logiciel gratuitement. Les prestataires de service rivaliseront entre eux pour offrir la meilleure qualité au prix le plus bas et les utilisateurs ne seront pas captifs. Pendant ce temps, ceux d'entre nous qui n'ont pas besoin de service pourront utiliser le logiciel sans avoir à payer pour ce service.

« Il est impossible de toucher beaucoup de gens sans publicité, et il est nécessaire de vendre le logiciel pour financer la publicité. »

« Il est absurde de faire de la publicité pour un logiciel auquel les gens ont librement accès. »

Différentes formes de publicité gratuite ou très bon marché peuvent être utilisées pour informer un certain public d'un projet comme GNU. Il est cependant possible, en effet, que l'on puisse atteindre davantage d'utilisateurs de micro-ordinateurs par le biais de publicité traditionnelle. Dans ce cas, une entreprise qui fait de la publicité pour le service consistant à copier et à envoyer GNU contre une certaine somme d'argent devrait avoir assez de succès pour payer, notamment, sa publicité. De cette façon, seuls les utilisateurs qui profitent de la publicité paient pour celle-ci.

En revanche, si beaucoup obtiennent GNU de leurs amis, et que de telles entreprises n'ont pas de succès, cela montrera que la publicité n'était pas réellement nécessaire pour diffuser GNU. Pourquoi les promoteurs de l'économie de marché ne veulent-ils pas laisser le marché en décider ?(4)

« Ma société compte sur un système d'exploitation propriétaire pour lui donner un avantage concurrentiel. »

GNU retirera le système d'exploitation du champ de la concurrence. Vous ne pourrez pas avoir d'avantage concurrentiel dans ce domaine, mais vos concurrents non plus. Vous rivaliserez dans d'autres domaines, tandis que vous vous rendrez mutuellement service dans celui-ci. Si votre activité consiste à vendre un système d'exploitation, pas de chance : vous n'aimerez pas GNU. Si votre activité est différente, GNU peut vous éviter de vous retrouver dans le monde coûteux du commerce de systèmes d'ex-

ploitation.

J'aimerais que le développement de GNU soit soutenu par les dons de nombreux constructeurs et utilisateurs, réduisant ainsi le coût supporté par chacun.[5]

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

S'il y a une chose qui mérite rétribution, c'est bien le fait de donner quelque chose à la société. La créativité peut être un don pour la société, mais seulement dans la mesure où la société est libre d'utiliser ce qui est créé. Si les programmeurs méritent d'être récompensés parce qu'ils créent des logiciels innovants, alors ils méritent, pour la même raison, d'être punis s'ils limitent l'utilisation de ces logiciels.

« Un programmeur ne peut-il pas demander une rétribution pour sa créativité ? »

Il n'y a rien de mal à vouloir être payé pour son travail, ni à chercher à augmenter ses revenus, pour autant que l'on n'utilise pas de moyens destructeurs. Or aujourd'hui, les moyens que l'on utilise dans le domaine du logiciel reposent sur une destruction.

Tirer de l'argent des utilisateurs d'un logiciel en limitant l'utilisation qui peut en être faite est destructeur, parce que cela réduit la quantité et les modalités d'utilisation du logiciel. Cela réduit la richesse que l'humanité tire du logiciel. Lorsque le choix de limiter est délibéré, les conséquences dommageables qui en résultent relèvent d'une destruction délibérée.

La raison pour laquelle un bon citoyen n'utilise pas de moyens destructeurs pour s'enrichir est que, si cette façon de faire se généralisait, nous serions tous appauvris par cette destruction mutuelle. C'est la morale kantienne, ou règle de réciprocité. Parce que je n'apprécie pas les conséquences qu'aurait une rétention générale de l'information, je suis forcé de considérer comme immoral un tel comportement. Dans le cas qui nous occupe, le désir d'être rétribué pour sa créativité ne justifie pas de priver le monde en général de tout ou partie de cette créativité.

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

Je pourrais répondre que personne n'est forcé à être un programmeur. La plupart d'entre nous serions incapables de gagner de l'argent comme clown de 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. Nous faisons autre chose.

Cette réponse, cela dit, n'est pas satisfaisante, parce qu'elle admet l'hypothèse implicite selon laquelle les programmeurs, si l'on abolit la propriété dans le domaine des logiciels, ne pourront plus recevoir le moindre centime. On fait comme si c'était tout ou rien.

La vraie raison pour laquelle les programmeurs ne mourront pas de faim, c'est qu'il sera toujours possible d'être payé pour programmer ; simplement, pas autant que maintenant.

Limiter les copies n'est pas le seul modèle économique utilisable dans l'industrie du logiciel. C'est le modèle le plus courant parce que c'est celui qui rapporte le plus d'argent. Si ce modèle était interdit, ou rejeté par le client, l'industrie se tournerait vers des modèles économiques différents, moins utilisés aujourd'hui. Il existe toujours de nombreuses façons d'organiser un domaine d'activité donné.

Il est probable que la programmation ne sera pas aussi lucrative dans le nouveau modèle qu'elle l'est aujourd'hui. Mais l'on ne saurait en tirer un argument contre le changement. On ne considère pas comme injuste que les commerciaux aient le salaire qu'ils ont aujourd'hui. Si les programmeurs avaient un salaire similaire, il n'y aurait pas là non plus d'injustice. (En pratique, ils gagneraient tout de même considérablement plus.)

« N'a-t-on pas le droit de maîtriser l'utilisation qui est faite de sa créativité ? »

« Maîtriser l'utilisation qui est faite de ses idées » revient en réalité à maîtriser la vie des autres ; et cela sert généralement à leur rendre la vie plus difficile.

Les personnes qui ont étudié avec attention les questions de propriété intellectuelle[6] (les avocats, par exemple) vous diront qu'il n'existe pas de droit naturel de propriété intellectuelle. Les soi-disant droits de propriété intellectuelle reconnus par le gouvernement ont été créés par des lois particulières dans des buts particuliers.

Ainsi, le système des brevets a été mis en place pour encourager les inventeurs à divulguer les détails de leurs inventions. Son but était d'aider la société plutôt que d'aider les inventeurs. À l'époque, la durée de vie d'un brevet, à savoir 17 ans, était une durée brève comparée à la cadence du progrès technique. Comme les brevets ne sont un problème que chez les constructeurs, pour qui les coûts et les efforts associés aux accords de licence sont peu importants par rapport à ceux de la production, les brevets sont généralement peu nuisibles. Il ne gênent pas la plupart des individus qui utilisent des produits brevetés.

L'idée de droit d'auteur n'existait pas dans l'Antiquité, où les auteurs copiaient fréquemment de longs passages d'autres auteurs. Il s'agissait alors d'une pratique très utile, sans laquelle les œuvres de nombreux auteurs n'auraient pas survécu. Le système du droit d'auteur a été créé expressément dans le but d'encourager les auteurs. Dans le domaine pour lequel il a été inventé — les livres, qui ne pouvaient être copiés de façon économique que dans une imprimerie —, il était peu dommageable et ne gênait pas la plupart des individus lisant des livres.

Tous les droits de propriété intellectuelle ne sont que des autorisations accordées par la société dans l'idée, juste ou fausse, que la société dans son ensemble en bénéficierait. Mais dans chaque situation particulière, nous devons nous poser la question : est-il vraiment souhaitable d'accorder une telle autorisation ? Quel type d'acte donnons-nous à quelqu'un l'autorisation de réaliser ?

Le cas des logiciels aujourd'hui est très différent de celui des livres il y a un siècle. Le fait que le moyen le plus facile de copier un programme est l'échange entre voisins, le fait qu'un programme a à la fois un code source et un code objet distincts l'un de l'autre, et enfin le fait qu'un programme est utilisé plutôt que lu et apprécié, tout cela concourt à créer une situation dans laquelle une personne qui fait respecter un copyright lèse la société dans son ensemble, matériellement et spirituellement ; et dans laquelle personne ne devrait se conduire ainsi, que la loi l'y autorise ou non.

« La concurrence conduit à de meilleures réalisations. »

Le modèle de la concurrence est la course : en récompensant le vainqueur, on encourage tout le monde à courir plus vite. Lorsque le capitalisme fonctionne réellement de cette manière, il remplit sa fonction ; mais ses promoteurs se trompent en supposant qu'il fonctionne toujours de cette manière. Si les coureurs oublient pourquoi la récompense est offerte et recherchent la victoire à tout prix, ils risquent de trouver d'autres stratégies, comme d'agresser les autres concurrents. Si les coureurs se mettent à se battre, ils seront tous retardés.

Le logiciel propriétaire et secret est l'équivalent moral des coureurs qui se battent. Hélas, le seul arbitre dont nous disposions ne semble pas trouver d'inconvénients aux combats, et se contente de les réguler (« Tous les dix mètres, vous avez droit à un coup de feu »). En réalité il devrait les séparer et punir les coureurs qui essaient de se battre.

« Les programmeurs ne vont-ils pas cesser de programmer sans incitation financière ? »

En fait, beaucoup de personnes sont prêtes à programmer sans la moindre incitation financière. La programmation a une fascination irrésistible pour certaines personnes, généralement les plus douées. On ne manque pas de musiciens professionnels qui, même s'ils n'ont aucun espoir de gagner leur vie de cette façon, continuent à jouer.

Mais en réalité cette question, bien qu'elle soit fréquemment posée, ne correspond pas à la situation. Le salaire des programmeurs sera réduit, mais ne sera pas réduit à néant. La vraie question est donc : trouvera-t-on encore des programmeurs si l'incitation financière est moindre ? D'après mon expérience, oui.

Pendant plus de dix ans, beaucoup des meilleurs programmeurs du monde ont travaillé au Laboratoire d'intelligence artificielle pour un salaire bien inférieur à celui qu'ils auraient touché ailleurs. Ils ont obtenu différents types de rétributions non financières : notoriété et considération, notamment. Et la créativité est aussi un plaisir, une rétribution en elle-même.

Ensuite, la plupart sont partis pour faire ailleurs le même travail avec une rémunération importante.

L'expérience nous montre que l'argent n'est pas ce qui motive les programmeurs ; mais que si on leur donne l'occasion d'en gagner beaucoup, ils finissent par s'y attendre et à l'exiger. Les organismes qui paient peu ont du mal à rivaliser avec ceux qui paient beaucoup, mais il y n'a pas de raison que leurs difficultés persistent une fois que ceux qui paient beaucoup ont été exclus.

« Nous avons désespérément besoin des programmeurs. S'ils exigent que nous cessions d'aider nos voisins, nous sommes forcés de leur obéir. »

Le désespoir n'est jamais si grand qu'il faille obéir à ce genre d'exigence. Rappelez-vous : « Des millions pour la défense, mais pas un cent de tribut ! »

« Les programmeurs doivent bien gagner leur vie d'une façon ou d'une autre. »

À court terme, c'est exact. Cependant, les programmeurs peuvent gagner leur vie de toutes sortes de façons, sans vendre le droit d'utiliser un programme. Cette façon de faire s'est imposée parce qu'elle est la plus lucrative pour les programmeurs et les hommes d'affaires, pas parce que c'est la seule façon de gagner sa vie. Il est facile de trouver d'autres solutions si on souhaite les trouver. Voici quelques exemples.

Un fabricant introduisant sur le marché un nouvel ordinateur paiera pour le portage de systèmes d'exploitation sur le nouveau matériel.

Les programmeurs peuvent également se consacrer aux services de formation, d'accompagnement et de maintenance.

Les personnes qui ont des idées originales peuvent distribuer leurs programmes gratuitement[7], demandant des dons aux utilisateurs satisfaits ou vendant des services d'accompagnement. J'ai rencontré des gens qui font cela.

Des utilisateurs ayant des besoins similaires peuvent former des groupes et payer des cotisations. Un groupe de ce genre pourrait signer un contrat avec un prestataire pour le développement des programmes que le groupe souhaite utiliser.

Toutes sortes de développements pourraient, par ailleurs, être financés par une taxe sur le logiciel.

Supposons que chaque personne achetant un ordinateur ait à payer un certain pourcentage du prix au titre de la taxe sur le logiciel. Le gouvernement confierait cette somme à un organisme comme la National Science Foundation pour qu'il finance le développement de logiciel.

Si l'acheteur de l'ordinateur fait lui-même un don à un projet de développement, il reçoit un crédit à valoir sur la taxe. Il peut choisir le projet auquel il adresse son don, son choix s'orientant généralement vers un projet dont il attend une utilité pour lui. Il peut recevoir un crédit à concurrence de la taxe totale qu'il doit.

Le taux de la taxe pourrait être déterminé par un vote de toutes les personnes qui la paient, pondéré par le montant sur lequel ils vont être taxés.

Conséquences :

– La communauté des utilisateurs d'ordinateurs soutient le développement de logiciel.
– Cette communauté décide du niveau de soutien nécessaire.
– Les utilisateurs qui se soucient de savoir à quels projets est attribuée leur part peuvent les choisir eux-mêmes.

À long terme, la libération des programmes est un pas vers le monde d'après pénurie, où personne ne devra travailler dur juste pour gagner sa vie. Les gens seront libres de se consacrer à des activités plaisantes, comme la programmation, après avoir consacré les nécessaires dix heures par semaine aux tâches indispensables comme la législation, le conseil aux familles, la réparation des robots et la recherche d'astéroïdes. Il ne sera plus nécessaire de gagner sa vie en programmant.

Nous avons déjà beaucoup réduit la quantité de travail que la société dans son ensemble doit réaliser par rapport à sa production, mais seule une petite partie de cette réduction s'est traduite en loisirs pour les personnes actives, parce qu'une importante activité non productive est nécessaire pour accompagner l'activité productive. Les causes principales en sont la bureaucratie et la lutte iso-

métrique contre les concurrents. Le logiciel libre réduira notablement ces nuisances dans le domaine de la production de logiciel. C'est une étape nécessaire pour que les gains techniques en productivité se traduisent en réduction du travail.

(1) La formulation utilisée dans ce passage était peu rigoureuse. L'idée était que personne n'aurait à payer pour l'**autorisation** d'utiliser le système GNU. Mais la formulation n'est pas claire, et on interprète souvent le passage comme disant que les copies de GNU devraient toujours être distribuées gratuitement ou presque. L'intention n'a jamais été celle-là ; plus bas, le manifeste mentionne la possibilité que des sociétés proposent le service de distribution de façon rentable. J'ai appris depuis à bien distinguer entre « free » dans le sens de « libre » et « free » dans le sens de « gratuit ». Un logiciel libre est un logiciel que les utilisateurs peuvent offrir et modifier librement. Certains utilisateurs peuvent obtenir des copies gratuitement, tandis que d'autres paieront ; si l'argent collecté sert à améliorer le logiciel, tant mieux. Le point important est que toute personne disposant d'une copie peut librement coopérer avec les autres en utilisant cette copie.
(2) Là aussi, je n'ai pas bien distingué entre les deux sens de « free » (N.d.T. : « libre » et « gratuit »). La phrase telle que je l'ai écrite n'est pas fausse : vous pouvez en effet obtenir gratuitement les logiciels GNU, par vos amis ou par l'Internet. Mais l'idée qu'elle suggère n'est pas la bonne.

(3) Il existe aujourd'hui plusieurs sociétés de ce type.
(4) Pendant 10 ans, la Free Software Foundation a tiré la plupart de son financement d'un service de distribution, bien que ce soit une fondation plutôt qu'une entreprise. Vous pouvez commander différents articles auprès de la FSF.
(5) Un groupe de sociétés informatiques ont 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 évoquer « le problème » de « la propriété intellectuelle » pouvait être source de confusion. Ce terme est à l'évidence tendancieux, et un problème plus subtil est qu'il amalgame des lois diverses et donnant lieu à des problèmes très différents. Aujourd'hui, je propose de rejeter complètement le terme de « propriété intellectuelle », pour éviter de suggérer qu'il s'agit d'un ensemble cohérent de lois. Pour s'exprimer clairement, il faut parler séparément de brevets, de copyright ou de marques déposées. Vous pouvez lire des explications plus détaillées sur la façon dont ce terme suscite la confusion et le parti-pris.
(7) Nous avons par la suite appris à distinguer entre le « logiciel libre » et « logiciel gratuit ». Le terme « logiciel gratuit » (N.d.T. : « freeware ») désigne un logiciel qui peut être librement offert, mais que l'on n'est généralement pas libre d'étudier et dont on ne peut pas librement modifier le code source, de sorte que la plupart du temps il ne s'agit pas de logiciel libre. Vous trouverez une explication plus détaillée à la page Termes prêtant à confusion, que vous devriez éviter.