



Les logiciels libres concernent l'ensemble des citoyens. Avancées et maîtrise technologique, utilisations, dimensions économiques et politiques, enjeux... État des lieux.



Bernard Lang.

sociation francophone des utilisateurs de logiciels libres (AFUL), c'est bien volontiers que je réponds à la sollicitation du CAES de présenter un bref état des lieux aux collègues du CNRS.

L'immatériel

Des siècles de développement économique fondé essentiellement sur la création et la distribution de biens matériels ont forgé notre perception des relations économiques et sociales. La principale caractéristique de ces biens matériels est qu'ils sont souvent rares, coûteux et difficiles à partager. On ne peut être deux à manger la même part de gâteau ou à conduire la même automobile.

La question des logiciels libres est aujourd'hui encore assez méconnue. Et fort vaste. Directeur de recherche à l'Institut national de la recherche en informatique et en automatique (INRIA), membre fondateur de l'As-

Produire et diffuser ces biens est coûteux et complexe. Coûteux en main-d'œuvre, en ressources matérielles et énergétiques. La gratuité n'est pas de ce monde matériel. Si l'automatisation de la production semble repousser ces difficultés, ce n'est en fait qu'une illusion car nous heurtons aujourd'hui aux limites de notre environnement planétaire. Nos ressources matérielles ne peuvent que rester rares et chères. Notre réflexion, comme nos structures économiques, restent donc principalement consacrées à l'organisation de la pénurie, à la détermination de la production possible et tolérable, à la façon de répartir les richesses disponibles.

Depuis un peu plus de vingt ans, quelques « illuminés » remettent en cause cette vision... et ils le font avec succès. Avec succès parce qu'ils se placent en dehors du monde matériel, dans le monde immatériel des idées, de la connaissance, du logiciel. Si deux personnes ont chacune un objet et qu'elles les échangent, elles ont toujours un objet chacune. Mais si deux personnes ont chacune une idée ou une chanson et qu'elles les échangent, elles repartent chacune avec deux idées, ou deux chansons. Les biens immatériels ont cette propriété merveilleuse de pouvoir se dupliquer et se transmettre à coût nul. Un professeur qui transmet son savoir enrichit ses étudiants sans s'appauvrir. C'est vrai du savoir humain, et c'est aussi vrai des logiciels qui sont en quelque sorte le savoir des machines.

Cela n'a pas toujours été vrai. Pendant longtemps, pour être transmises, les idées avaient besoin d'un support matériel (livre, partition, film...), donc coûteux. L'informatique a changé tout cela par la numérisation des savoirs et des œuvres, leur codage sous une forme électronique que l'on peut reproduire et diffuser à un coût quasi nul. Cette dématérialisation a changé la donne.

Les logiciels

Jusqu'à la fin des années 1970, on se souciait assez peu de la propriété des logiciels et de leur commercialisation, en raison de la

diversité des types de machines et du relativement faible marché que représentait le parc disponible. L'information circulait facilement, et il était courant pour les utilisateurs de pouvoir modifier les programmes pour les adapter à leurs besoins. Avec l'apparition des mini (1970) et surtout des micro-ordinateurs vers 1980, le marché s'est brutalement élargi : la vente des logiciels devint une source de profits considérables que l'on voulait protéger en les contrôlant.

Le contrôle des logiciels s'exerce de plusieurs façons. Juridiquement, la propriété des logiciels est gérée par le droit d'auteur, comme la plupart des créations de l'esprit. Les seuls droits dont dispose l'utilisateur sont ceux qui lui sont explicitement concédés dans un texte appelé licence qui doit nécessairement accompagner tout logiciel diffusé. Typiquement, les créateurs ou éditeurs de logiciels commerciaux exercent leur droits de propriété en n'autorisant que la mise en œuvre du logiciel sur un nombre limité d'ordinateurs, et en interdisant toute modification et toute redistribution du logiciel. Le déploiement de ces « logiciels propriétaires » a eu pour effet d'exclure les programmeurs indépendants ou universitaires de toute participation libre à l'évolution de leur outil, ou même à son simple entretien.

Le contrôle sur les logiciels s'exerce aussi techniquement par le secret. En effet, un logiciel se présente généralement sous deux formes : le code source et le code objet. Le code source, lisible par les programmeurs, est la forme utilisée pour réaliser et modifier le logiciel : c'est la « partition » du « compositeur » du programme. Le code objet est la version utilisée par la machine. Elle est produite automatiquement à partir du code source par des programmes spéciaux appelés compilateurs. C'est une forme du programme qui est quasi illisible pour un être humain, et que l'on enregistre sur les CD et DVD destinés à être lus par l'ordinateur, un peu comme la musique est enregistrée sur un CD pour être jouée par un lecteur. Mais, s'il est relativement facile de reconstituer la partition quand on a la musique, il est essentiellement impossible de reconstituer le code source à partir du code objet. On peut donc aussi contrôler un logiciel en gardant secret son code source.

C'est ainsi qu'un chercheur du Massachusetts Institute of Technology, le célèbre MIT, ayant eu besoin de modifier le programme de contrôle d'une imprimante pour l'adapter à un nouvel usage, s'est vu opposer une fin de non-recevoir par le fabricant. Cela se passait au début des années 1980 et ce chercheur, Richard M. Stallman, prit alors conscience de cette évolution des rapports de force et de cette exclusion nouvelle des utilisateurs. Mais son coup de génie fut d'être le premier à comprendre que, comme la recherche scientifique, la création de logiciels ne relève pas nécessairement d'une activité industrielle et commerciale, et que la collaboration de programmeurs compétents est la seule chose qui compte.

Le sabotier et le programmeur

Supposons qu'un sabotier génial invente un nouveau modèle de sabot ultraléger et confortable. Il décide d'en faire profiter l'humani-

té. Pour produire des sabots pour tout le monde, même sans bénéfices, il faut construire une usine, acheter des machines, embaucher des ouvriers, acheter du bois et de l'énergie, acheter des camions pour livrer les produits, etc. Tout cela coûte, et il faudra donc aussi toute une infrastructure de gestion, et faire payer les sabots, même si c'est à prix coûtant, pour rentrer dans les frais de production. Le sabot gratuit n'existe pas.

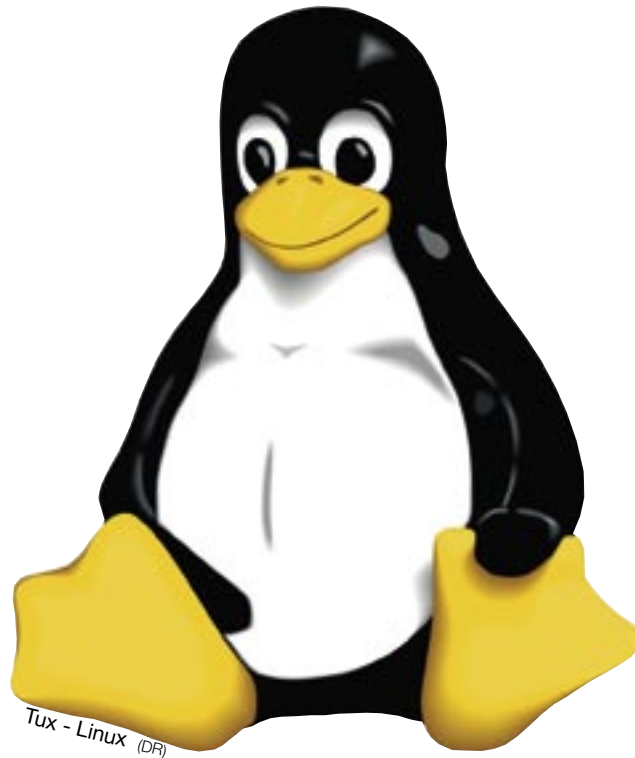
En revanche, si un programmeur génial invente un nouveau logiciel et souhaite le mettre à la disposition de tous, il peut le faire sans aucune activité commerciale, tout simplement en le diffusant sur l'Internet.

C'était une partie de l'idée de Stallman, même si la faible diffusion de l'Internet à l'époque (autour de 1983) l'a obligé à utiliser aussi des supports magnétiques pendant quelques années. La gratuité n'était pas vraiment ce qui l'intéressait. Chercheur de formation, Stallman connaissait la valeur de l'échange et de la collaboration dans la création scientifique et technique. Sa grande idée fut de trouver le moyen de perpétuer cette tradition universitaire dans le monde de la création des logiciels. À la différence des diffuseurs de logiciels propriétaires, il choisit de diffuser les logiciels en fournissant le code source qui permet de les comprendre, accompagnés d'une licence qui permet à quiconque d'utiliser, de modifier et de redistribuer ces logiciels. L'idée du « logiciel libre » était née. Était-elle autre chose que la liberté des idées et cette liberté universitaire de les faire évoluer librement ?

De cette liberté naissent des communautés qui collaborent à l'élaboration de nouvelles ressources et de nouvelles idées, parfois par l'échange et la complémentarité, parfois par la concurrence sur des propositions qui s'opposent. Le logiciel libre, c'est la possibilité pour tout utilisateur qui en a la compétence d'apporter ses innovations et de tenter de les faire adopter par la communauté. Et justement, les travaux de certains économistes, notamment d'Eric von Hippel, du MIT, montrent que dans bien des domaines les innovations utiles sont imaginées par les premiers intéressés : les utilisateurs. Un des exemples favoris de von Hippel concerne les articles de sport. Pour les articles de sport, principalement produits en usine, le fabricant est obligé de rechercher des innovations chez les utilisateurs avant de créer à grands frais une nouvelle ligne de produits. Pour les logiciels, la boucle économique est simplifiée car l'utilisateur qui apporte une amélioration peut directement la mettre lui-même à la disposition de tous.

Et le libre fut

Richard Stallman décida donc de contrer la tendance propriétaire en créant une fondation pour le logiciel libre, la Free Software Foundation (FSF), non par hostilité au commerce, mais parce qu'il s'oppose à tout ce qui entrave la liberté des échanges et de la créativité ou limite arbitrairement l'accès à la connaissance et à la culture. La première tâche de la FSF fut l'élaboration du projet GNU, projet incroyablement ambitieux pour ses moyens, visant à créer un environnement informatique complet. Bien que s'appuyant sur le modèle d'Unix, le plus en faveur à l'époque chez les universitaires qui y ont largement contribué, le nom du projet GNU



signifie par dérision *GNU's Not Unix* (« GNU n'est pas Unix »). En effet, le nom Unix était alors la propriété, en tant que marque, de la société ATT et il fallait éviter tout problème juridique.

La FSF et le projet GNU purent bénéficier de la collaboration de nombreux universitaires qui y retrouvaient la liberté de collaboration et d'échange que leur permettaient de moins en moins une industrie en cours de cloisonnement dans des bunkers juridiques destinés à défendre la « propriété intellectuelle » de chacun. Enthousiasmés par le projet, bien d'autres décidèrent de contribuer et, dix ans plus tard, la FSF se retrouvait à la tête d'un ensemble complet d'outils logiciels de développement, ainsi que de nombreuses applications, sans parler d'un nombre considérable de logiciels libres réalisés à son exemple, mais indépendamment de la FSF.

Au début des années 1990, ne manquait plus que le noyau du système GNU, c'est-à-dire le logiciel central qui gère le fonctionnement intime de la machine et qui permet de mettre en œuvre outils logiciels et applications. C'est alors qu'un étudiant finlandais, Linus Torvalds, commença de son côté à développer une version du noyau d'Unix pour son PC, faute de pouvoir s'offrir une trop coûteuse version commerciale. Ayant réalisé un mini prototype, il demanda de l'aide sur Internet (qui commençait à prendre son essor) et fut satisfait au-delà de toute espérance. Très rapidement, une équipe mondiale de collaborateurs se constitua autour de lui et, en trois à quatre ans, un noyau Unix parfaitement opérationnel, baptisé Linux par l'équipe, apportait le soubassement que n'avait pu encore réaliser la FSF. Le système GNU-Linux était disponible.

Les raisons d'un succès

Petit à petit, les logiciels libres s'imposèrent dans de nombreux usages, non pas tant en raison de leur gratuité – qui ne pouvait aider GNU-Linux dans un contexte où l'on doit payer Windows de toute façon lors de l'acquisition d'une machine –, mais en raison de leur qualité technique, de leur sécurité et de la liberté laissée aux utilisateurs.

D'ailleurs, un logiciel libre n'est pas nécessairement gratuit, et aucune licence libre n'interdit de diffuser le logiciel moyennant paiement. Nombre de sociétés développent des logiciels libres et les vendent. Dans la mesure où le prix en est raisonnable, il est souvent plus simple pour un utilisateur – notamment une entreprise – d'acheter la dernière mouture à une source de confiance plutôt que de faire des recherches pour une version gratuite sur l'Internet. Et il est sain de soutenir celui qui vous fournit un service.

L'avantage principal du logiciel libre est de rétablir la concurrence. Le marché du logiciel est naturellement fort peu concurrentiel dans la durée pour des raisons structurelles incontournables, à la fois techniques et économiques, notamment la faiblesse des coûts de production qui exacerbe l'importance des facteurs d'échelle, ainsi que des effets de réseau massifs, c'est-à-dire le besoin d'uniformité des modes de communication humains et techniques (ce que l'on appelle « interopérabilité »). Tout concourt à ce qu'un acteur dominant le soit de plus en plus, de façon irréversible, et on constate effectivement une monopolisation rapide des différents secteurs du logiciel.

Le logiciel libre s'est révélé être le seul capable de remettre en cause la monopolisation de la création de logiciel. Par sa nature même, il rend public ses modes de communication et permet donc de créer facilement des logiciels concurrents mais compatibles. En fait, il pousse la concurrence beaucoup plus loin, puisque quiconque peut reprendre un logiciel libre existant, et le modifier pour faire sa propre version – voire la commercialiser – qui peut compléter ou être en concurrence avec la version initiale. C'est un mode de développement très semblable à celui de la recherche scientifique, où coopération et concurrence sont étroitement mêlées, entre les hommes et entre leurs travaux. En économie comme en recherche, coopération et concurrence sont les deux mamelles de l'innovation, de la qualité et de l'adaptation. C'est aussi vrai dans l'évolution des espèces.

Rupture économique

La rupture économique du logiciel libre se décline pour toutes les autres créations immatérielles : musique, films, œuvres écrites, et autres. Mais le problème se présente chaque fois de façon un peu différente. La principale difficulté, surtout dans le secteur artistique, est de pouvoir rémunérer les créateurs sans mettre toute la filière, notamment les créateurs et le public, sous la coupe d'intermédiaires qui ont des intérêts très divergents. Le logiciel souffre moins de ce problème.

Un autre cas intéressant est celui des publications scientifiques, écrites gratuitement par des universitaires pour des universitaires. L'Internet permettant de gérer efficacement le problème de la reproduction et de la diffusion, la communauté scientifique est en train de reprendre le contrôle d'un système qui était devenu un moyen pour quelques grands éditeurs de ponctionner lourdement les budgets éducation-recherche. L'accès libre aux publications ne peut que favoriser leur impact au bénéfice d'une plus grande efficacité de la recherche et de l'enseignement.

Qui paye ?

Initialement, les logiciels libres ont été développés par des universitaires et par des bénévoles. Mais cela passe-t-il à l'échelle des besoins de la planète ? L'une des principales motivations des développeurs initiaux était tout simplement de créer ce dont ils avaient besoin. Et comme ce besoin était souvent partagé, d'autres participaient alors à la création, mutualisant le travail de chacun au bénéfice de tous. Mais rien n'impose que ce modèle soit limité à des personnes physiques. Actuellement, on constate que nombre d'organisations publiques ou privées, qui ont souvent développé des logiciels pour leur besoins internes, choisissent de les diffuser en logiciels libres, justement pour susciter cette même mutualisation au bénéfice de tous.

Cela revêt en pratique plusieurs formes. L'association ADUL-LACT, par exemple, a été créée pour organiser cette mutualisation dans les administrations et les collectivités territoriales. L'Open Source Development Laboratory (OSDL) réunit plutôt des entreprises. La mutualisation peut aussi se faire par le biais de sociétés de service qui font du développement à façon pour leurs clients, sous licence libre. Chaque nouveau client peut s'appuyer sur le travail déjà réalisé pour ses prédécesseurs, et ne payer que pour les développements nouveaux. En outre, comme le logiciel est libre, un client peut toujours mettre plusieurs fournisseurs en concurrence, à tout moment. Bien entendu, le travail est fait par des programmeurs, qui sont payés pour leur travail. De même, les sociétés de service sont payées pour leur travail. D'une certaine façon, toutes les organisations utilisatrices se comportent comme des fournisseurs bénévoles de logiciels libres, mais qui sous-traitent en fait le travail technique à des tiers, employés ou entreprises sous contrat. Ceci n'est bien sûr qu'un aperçu, car les choses sont en fait souvent plus complexes et bien plus variées.

Pouvoir et rentes

La mutualisation de la satisfaction d'un besoin n'est pas en soi originale. C'est précisément ce que fait tout fournisseur pour ses clients. La vraie question est de savoir qui contrôle et décide des besoins à satisfaire et de la façon de le faire : le fournisseur ou les clients. Quand il y a concurrence, les clients peuvent faire

Nous aussi, au CAES...

Les logiciels libres ont été introduits voici cinq ans au CAES. Premier emploi : le système d'exploitation des serveurs, notamment ceux qui gèrent les services Internet et nos serveurs Web (Zope, Plone, Apache). Aujourd'hui, la moitié des serveurs que nous utilisons sont sous FreeBSD.

Nous avons aussi fait le choix de partager notre logiciel de gestion intégrée (Ganaël®) en le mettant sous licence GNU-GPL. Dans le cadre de nos accords de partenariat, l'Adas Inra et le CAES de l'Inserm peuvent ainsi bénéficier des développements réalisés par notre service Informatique. La veille est permanente : dès que nous trouvons des logiciels libres répondant à nos besoins, nous les utilisons.

M. A.

pression sur les fournisseurs pour qu'ils répondent à leurs demandes. Quand les fournisseurs forment des monopoles, des oligopoles ou des cartels – ce qui est la situation habituelle dans les technologies de l'information et de la communication (TIC), du logiciel au téléphone – on constate que les fournisseurs imposent des solutions qui sont avantageuses surtout pour eux, au détriment de l'économie, de la société et de l'innovation.

La création mutualisée de solutions libres est une façon efficace de rétablir la concurrence et la primauté de la demande sur l'offre, compte tenu des caractéristiques techno-économiques du logiciel. Mais les seigneurs des TIC ne se laissent pas faire. Le terme « seigneurs » est approprié, car leur modèle économique est analogue à celui des seigneurs du Moyen Âge, qui rançonnaient les commerçants empruntant les voies de communication dans la vallée sous leur contrôle. La vallée, c'est maintenant un standard de communication, contrôlé notamment par les brevets, qui, en empêchant la concurrence, permet de taxer chaque utilisateur, obligé par le contexte économique d'utiliser ce standard, bon ou mauvais.

Cette économie de rente est courante dans les télécommunications, et c'est aussi la première motivation des tentatives actuelles pour étendre la brevetabilité au logiciel. Le brevet logiciel permet de contrôler les communications entre programmes, ou entre utilisateurs, par exemple en brevetant le format des documents électroniques. Par ce contrôle de l'interopérabilité, un éditeur de logiciel peut empêcher tout développement d'un produit concurrent. D'autant plus que le brevet – qui est légalement un privilège... comme il sied à des seigneurs – est économiquement incompatible avec les logiciels libres, car il introduit un coût unitaire qui n'existe pas naturellement pour les biens immatériels.

Les enjeux

Les enjeux du logiciel libre sont nombreux. Économiquement, ils sont accessibles à tous, et permettent de réduire la fracture numérique : entre les citoyens, pour l'éducation notamment, mais aussi entre les pays. Les pays en développement ont ainsi accès à une ressource bon marché et qu'il peuvent maîtriser techniquement, voire développer localement. C'est d'ailleurs tout aussi important pour les économies développées, comme en témoigne le succès des logiciels libres en Europe et au Japon. D'un point de vue écologique, on constate qu'ils permettent de mieux maîtriser la pérennité des matériels, et donc l'impact écologique de leur renouvellement.

Le terme « logiciel libre » amène souvent la question de savoir ce qui est libre. Pour certains c'est le programmeur, pour d'autres c'est l'utilisateur. Peut-être est-ce encore plus le logiciel lui-même, qui peut acquérir une existence indépendante de ses créateurs, et donc se développer grâce à toutes les opportunités qui se présentent, de même que la science libre (on dit plutôt « ouverte ») se développe dans un tissu de liberté, profitant de tous les cerveaux compétents qui se mettent à sa disposition.

Mais la liberté est indivisible, et le logiciel libre est aussi la protection de nos libertés au sens le plus traditionnel. Les logiciels sont maintenant omniprésents dans notre société, dans nos activités, dans nos infrastructures techniques. Or ces infrastructures décident très largement de ce qui est permis, de ce qui est possible. Les villes qui se battent pour une autoroute ou un TGV le savent bien.

Les choix techniques qui président aux infrastructures de l'Internet, des réseaux administratifs, des réseaux de santé et autres sont des choix de long terme, qui régiront pour une large part notre façon de vivre, au même titre que les lois votées au Parlement. « *Le code fait loi* », nous dit Lawrence Lessig, professeur de droit constitutionnel à l'université de Stanford, en faisant référence au code des programmes informatiques. Le logiciel gère les communications et donc aussi les échanges culturels, la vie privée des gens, la sécurité des entreprises et celle des pays. Face aux logiciels contrôlés par des intérêts privés, le logiciel libre apporte cette transparence indispensable aux (infra)structures d'une société démocratique et la maîtrise qu'exige sa souveraineté. ●

Bernard Lang

Bernard Lang est directeur de recherche à l'Institut national de la recherche en informatique et en automatique (INRIA) et membre fondateur de l'Association francophone des utilisateurs de logiciels libres (AFUL).



Pour en savoir plus

- Autres articles de Bernard Lang : <http://pauillac.inria.fr/~lang/ecrits/>
- Quelques pointeurs :
 - Association des développeurs et utilisateurs de logiciels libres pour les administrations et les collectivités territoriales (ADULLACT) : <http://adullact.org/>
 - Association francophone des utilisateurs de logiciels libres (AFUL) : <http://aful.org/>
 - Association pour la promotion et la recherche en informatique libre (APRIL) : <http://april.org/>
 - CeCILL, la licence libre CEA/CNRS/INRIA : <http://cecill.info/>