

Du logiciel libre aux ressources collaboratives libres

Sébastien BLONDEEL
blondeel@april.org

La Lune des Pirates — vendredi 11 avril 2003

APRIL — <http://www.april.org>

Table des matières

1	Introduction	4
2	Aspects techniques	4
2.1	Code source, compilation	5
2.2	Système d'exploitation, applications	8
3	Aspects juridiques	8
3.1	Théorie	8
3.2	Limites	9
4	Les grandes familles de licences	10
4.1	Le logiciel privé	10
4.2	Le logiciel propriétaire	10
4.3	Le logiciel libre	10
4.3.1	Les licences copyleft	11
4.3.2	Les autres licences	11
5	La liberté selon la FSF	12
5.1	Les quatre critères de la liberté	12
5.2	Les types d'œuvres et les conséquences	13
6	Historique	13
6.1	Internet	13
6.2	GNU	14
6.3	Linux	14
6.4	Open Source	15
7	Aspects humains et psychologiques	15
7.1	La motivation, la concurrence	15
7.2	Les communautés	16

8 Aspects industriels	16
8.1 Les business plans	16
8.2 Intérêts du logiciel libre	16
8.3 Le coût du passage	17
9 Aspects administratifs	18
10 Les prochains défis à relever	18
10.1 Documentation technique	18
10.2 Ne pas se laisser distancer	19
10.3 Éduquer	19
10.4 Les brevets logiciels	20
11 Quelques exemples dans d'autres domaines	21
11.1 Jeux	21
11.2 Encyclopédies	22
11.3 Littérature	23
11.4 Autres	24
12 Bibliographie	25

1 Introduction

Les présentations du logiciel libre auxquelles j'ai assisté pêchaient souvent par excès de technicité. Quand le public convient (ex : DESS d'informatique), cela se comprend, et c'est même souhaitable : on parle leur jargon pour faire passer plus d'idées en moins de mots, et on se concentre sur les sujets qui leur tiennent à cœur ou qu'ils sont susceptibles de rencontrer dans leur future vie professionnelle.

J'essaie souvent de me mettre à la place de ceux à qui on parle, en faisant abstraction de mes connaissances techniques ou de ma connaissance du sujet, et je crains que souvent on perde tout le monde en quelques minutes et pérore des heures sans que cela ne les touche plus.

On peut bien sûr retrouver dans le logiciel libre des idées chères aux communistes, aux altermondialistes, aux soixante-huitards, aux libéraux, et j'en oublie : je connais des gens marqués dans l'un de ces courants d'idées qui sont motivés par le fait que le logiciel libre porte leur credo, et qui n'en parlent qu'à travers ce dernier.

Il s'agit quand même de domaines et d'idées fondamentalement apolitiques, ou transpolitiques, et je vais tâcher de vous les expliquer, pour qu'ensuite chacun d'entre vous fasse éventuellement sa petite cuisine mentale et s'y retrouve.

J'ai le défaut d'avoir fait des études d'informatique : j'ai beau jeu de dire aux autres que ces choses sont faciles et pas effrayantes. Je connais des gens d'autres spécialités (physique, chimie, maths, histoire, droit, sociologie, ...) qui ont peu à m'envier en matières d'aisance sur les ordinateurs, et de toutes manières ce n'est pas vraiment tout ceci que j'ai étudié dans mon cursus. En France, le terme «informatique» recouvre un peu tout et n'importe quoi ; je l'utiliserai dans le sens «asservir la machine à l'homme, comprendre ce qui est bête et méchant et répétitif et le lui faire faire».

2 Aspects techniques

Le logiciel libre (dans cet exposé on utilisera ce terme au sens de la Free Software Foundation, et il sera défini précisément plus loin), s'il a de nombreux aspects et dimensions humaines, psychologiques, sociales, sociétales, économiques, etc., se définit avant tout par des critères techniques.

2.1 Code source, compilation

Les logiciels sont écrits par des hommes. Au début c'était difficile et les contraintes des premières machines étaient lourdes. Mais le temps a passé et les programmeurs disposent maintenant d'environnements et de langages de programmation conviviaux pour eux — plus intuitifs, et éloignés de la froideur mécanique de la machine.

L'aspect de la programmation destiné à être manipulé et partagé par des hommes s'appelle le code source. Les machines utilisent, pour des raisons historiques ou d'économie de place ou de vitesse, des codes binaires.

La transformation du code source en code binaire s'appelle la compilation. Elle est menée par un compilateur, et elle détruit des méta-informations : on ne peut pas remonter. On dispose de strictement moins de compréhension de ce qui s'est passé dans l'esprit du programmeur si l'on ne dispose que du code binaire ; on en est presque réduit à un rôle de consommateur passif. J'ai déjà illustré cela par des ateliers grand public, et récemment et régulièrement, il m'arrive à titre personnel de vérifier l'intérêt de disposer du code source.

Le code source est un peu comme la recette de cuisine : un gâteau une fois sorti du four s'adressera directement aux papilles gustatives, qui en apprécieront quasi principalement les qualités. Il sera très difficile d'en retrouver la recette, et d'en fabriquer une version un peu plus sucrée ou en remplaçant ceci par cela. Au contraire, le rédacteur de la recette aura émaillé cette dernière de commentaires et de conseils pour la préparation, qui disparaîtront lors de la réalisation de la recette.

Il n'est pas nécessaire d'avoir étudié l'informatique pour comprendre un code source. Exemple :

```
#include <stdio.h>

int main () {
    int compteur;
    for (compteur = 1; compteur <= 10; compteur = compteur + 1) {
        int carre, cube;
        carre = compteur * compteur;
        cube = carre * compteur;
        printf("%d\t%d\t%d\n", compteur, carre, cube);
    }
}
```

À l'exécution on obtient :

1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

Le code binaire a l'allure suivante :

```

00000000: 7f45 4c46 0101 0100 0000 0000 0000 0000 .ELF.....
00000100: 0200 0300 0100 0000 1083 0408 3400 0000 .....4...
00000200: 1408 0000 0000 0000 3400 2000 0600 2800 .....4. ...(.
00000300: 1900 1800 0600 0000 3400 0000 3480 0408 .....4...4...
00000400: 3480 0408 c000 0000 c000 0000 0500 0000 4.....
[...]
00005b00: f682 0408 0683 0408 0000 0000 0047 4343 .....GCC
00005c00: 3a20 2847 4e55 2920 322e 3935 2e34 2032 : (GNU) 2.95.4 2
00005d00: 3030 3131 3030 3220 2844 6562 6961 6e20 0011002 (Debian
00005e00: 7072 6572 656c 6561 7365 2900 0047 4343 prerelease)..GCC
00005f00: 3a20 2847 4e55 2920 322e 3935 2e34 2032 : (GNU) 2.95.4 2
00006000: 3030 3131 3030 3220 2844 6562 6961 6e20 0011002 (Debian
00006100: 7072 6572 656c 6561 7365 2900 0047 4343 prerelease)..GCC
[...]
0000bb00: 0700 0000 0000 0000 0000 0000 dc06 0000 .....
0000bc00: 7800 0000 0000 0000 0000 0000 0100 0000 x.....
0000bd00: 0000 0000 0100 0000 0300 0000 0000 0000 .....
0000be00: 0000 0000 5407 0000 bf00 0000 0000 0000 ....T.....
0000bf00: 0000 0000 0100 0000 0000 0000 .....

```

Un minimum d'honnêteté intellectuelle m'oblige à signaler que si l'on a la chance de disposer d'un code binaire non «strippé» on peut le faire parler de manière un peu plus convaincante en utilisant un débogueur. On obtient alors :

```

(gdb) disassemble main
Dump of assembler code for function main:
0x80483f0 <main>:      push   %ebp
0x80483f1 <main+1>:      mov    %esp,%ebp

```

```
0x80483f3 <main+3>:    sub    $0x18,%esp
0x80483f6 <main+6>:    nop
0x80483f7 <main+7>:    movl   $0x1,0xffffffff(%ebp)
0x80483fe <main+14>:   mov    %esi,%esi
0x8048400 <main+16>:   cmpl  $0xa,0xffffffff(%ebp)
0x8048404 <main+20>:   jle   0x8048408 <main+24>
0x8048406 <main+22>:   jmp   0x8048440 <main+80>
0x8048408 <main+24>:   mov   0xffffffff(%ebp),%eax
0x804840b <main+27>:   imul  0xffffffff(%ebp),%eax
0x804840f <main+31>:   mov   %eax,0xffffffff8(%ebp)
0x8048412 <main+34>:   mov   0xffffffff8(%ebp),%eax
0x8048415 <main+37>:   imul  0xffffffff(%ebp),%eax
0x8048419 <main+41>:   mov   %eax,0xffffffff4(%ebp)
0x804841c <main+44>:   mov   0xffffffff4(%ebp),%eax
0x804841f <main+47>:   push  %eax
0x8048420 <main+48>:   mov   0xffffffff8(%ebp),%eax
0x8048423 <main+51>:   push  %eax
0x8048424 <main+52>:   mov   0xffffffffc(%ebp),%eax
0x8048427 <main+55>:   push  %eax
0x8048428 <main+56>:   push  $0x80484a4
0x804842d <main+61>:   call  0x8048300 <printf>
0x8048432 <main+66>:   add   $0x10,%esp
0x8048435 <main+69>:   incl  0xffffffffc(%ebp)
0x8048438 <main+72>:   jmp   0x8048400 <main+16>
0x804843a <main+74>:   lea  0x0(%esi),%esi
0x8048440 <main+80>:   leave
0x8048441 <main+81>:   ret
0x8048442 <main+82>:   lea  0x0(%esi,1),%esi
0x8048449 <main+89>:   lea  0x0(%edi,1),%edi
End of assembler dump.
```

Travaillant quasi-exclusivement avec du logiciel libre, je connais mal les technique de déplombage et de décompilation. Vous aurez cependant compris qu'il est bien plus difficile de comprendre ce que le programme fait, et de le modifier pour le corriger, le faire évoluer, ou l'adapter aux besoins, si on ne dispose pas de son code source.

Comment procéderiez-vous pour aller jusque 12 et non 10, en disposant :

- du code source ?
- du code binaire ?
- d'un déboggeur ?

Même question pour calculer les puissances quatre, travailler par incréments de deux, etc.

Cet exemple peut sembler bête et taillé sur mesure. Il en est exactement de même avec la plupart des programmes. Nul besoin d'assimiler toute l'architecture de l'arborescence de fichiers pour intervenir sur un point de détail précis. J'ai un atelier à ce sujet utilisant le jeu affenspiel, et récemment à titre personnel j'ai compris une limitation un peu laconique d'un jeu vidéo de cette manière.

Ne diffuser que la version binaire, compilée, d'un programme revient donc à ériger un obstacle gratuit et artificiel pour restreindre la souplesse d'utilisation de ce produit par autrui.

2.2 Système d'exploitation, applications

Le système d'exploitation est assez distinctement séparé des applications sur les systèmes Unix. Il s'agit de la couche de virtualisation logicielle qui s'interface entre les programmes et les ressources de la machine : elle a pour tâche de reconnaître le matériel et le gérer.

Ainsi, on accédera à n'importe quel modèle de moniteur, de carte vidéo ou de disque dur de la même manière, et le même programme fonctionnera sur des systèmes différents.

Les applications sont les programmes destinés à l'utilisateur.

Le logiciel libre est traditionnellement bon et riche en logiciels plus proches du système, mais il se dote peu à peu d'une bibliothèque de programmes applicatifs de plus en plus fournie et convaincante.

3 Aspects juridiques

3.1 Théorie

I-A-N-A-L. C'est une mise en garde classique en anglais : je ne suis pas un avocat. Vous trouverez sans dommage un mathématicien, professeur ou étudiant, pour vous soutenir en s'engageant que deux et deux font quatre, ou qu'en plaçant des poutres de cette manière votre toiture supportera sans problème telle épaisseur de neige.

Je crains qu'il soit autrement plus difficile de trouver un juriste qui soutienne quoi que ce soit. Les lois changent au gré des rapports de force, les juges les interprètent

de manières diverses, et on observe même des revirements de jurisprudence. J'en ai conclu — peut-être de manière simplificatrice et abusive — qu'il n'est nulle vérité absolue dans ce domaine.

J'exposerai ici ce que j'ai compris et retenu de diverses lectures et exposés sur le sujet. Il s'agira évidemment de droit américain, puisque c'est historiquement et économiquement le territoire de développement privilégié des questions logicielles.

Un logiciel est soumis dès sa création, même s'il ne sort pas du tiroir, au droit du copyright. Ce copyright donne par défaut tous les droits classiques à son auteur, tels que celui d'en interdire la production de copies ou d'exiger une taxe à chaque copie effectuée.

Il peut également soumettre l'utilisation du logiciel à des conditions plus ou moins aberrantes ou discriminatoires.

Ce droit se vend ; on n'en conserve alors plus rien. Ni droit de repentir ni droit moral. C'est le transfert de copyright.

Si l'on souhaite placer un logiciel sous des conditions d'utilisation et de diffusion différentes, on peut l'accompagner d'une licence, que l'utilisateur final devra ou non, selon les pays ou les époques, accepter sciemment et explicitement.

Les œuvres tombées ou placées dans le domaine public peuvent être utilisées par quiconque de la manière qu'il le souhaite. Il peut les revendre sous son propre copyright, les travestir, les modifier, etc.

Le logiciel libre n'est pas libre de droits. Les auteurs de logiciels libres, même s'ils émettent des réserves quant à l'arsenal juridique actuellement en place sur les questions de copyright, exploitent ce dernier pour arriver à leurs fins. Un logiciel libre utilise donc le droit du copyright pour imposer certaines conditions de diffusion à tous. Si on ne les accepte pas, par défaut on n'a aucun droit sur la création intellectuelle d'autrui, et on retombe dans le schéma classique, autrement plus restrictif.

3.2 Limites

En France et en Europe on a d'autres droits, le droit latin notamment prévoit une partie patrimoniale et une partie morale au droit d'auteur.

Il y a également la question des langues (en France une traduction en français est obligatoire, le contrat de licence devrait donc peut-être l'être pour avoir une quelconque valeur).

4 Les grandes familles de licences

4.1 Le logiciel privé

Le logiciel privé est le logiciel majoritaire en volume de par le monde. C'est du logiciel que l'on produit et garde par devers soi (ou remet contre rétribution au commanditaire ou client qui a passé commande dans le cas d'une société de services). Il n'est jamais diffusé à des tiers, les questions de conditions de diffusion ne se posent donc pas.

4.2 Le logiciel propriétaire

On appelle logiciel propriétaire tout logiciel qui n'est pas libre. Il s'agit probablement du logiciel auquel vous êtes habitués : il est fréquent qu'il soit payant, et/ou fourni sans code source, et/ou avec interdiction de le recopier et de le diffuser à des tiers, de manière lucrative ou non.

4.3 Le logiciel libre

Le logiciel libre est un terme parfois galvaudé (et l'Open Source Initiative n'ayant pas réussi à déposer la marque, cette autre dénomination n'apporte pas non plus la solution).

Les entités principales traitant et théorisant sur le sujet sont la Free Software Foundation, ancienne (créée en 1984), solide et réfléchie, et l'Open Source Initiative, plus récente et brouillon.

Toutes deux proposent des définitions permettant de décider si la licence d'un logiciel en fait un logiciel libre (FSF) ou un logiciel Open Source (OSI). Leurs sites Web respectifs dressent également la liste des licences les plus répandues qu'ils ont analysées et les classent en deux catégories.

L'examen de ces listes montre que ces deux définitions formelles, fort différentes (la définition du logiciel libre compte quatre conditions fort simples, inchangées depuis toujours et la définition de l'Open Source compte une petite dizaine de conditions tarabiscotées et en est à la version 1.9) sont quasiment équivalentes (l'APSL, licence peu utilisée qui nie le droit à la vie privée, est compatible OS mais pas libre au sens FSF).

On traite donc des mêmes objets sous des noms différents. Cet éclairage a quand même son importance quant aux éléments mis en avant : la FSF traite avant tout

de la liberté de l'utilisateur, l'OSI adopte un point de vue de technicien. Leurs slogans pourraient être «Liberté, Égalité, Fraternité» et «Rentabilité, Efficacité, Fiabilité».

Il existe plusieurs dizaines de licences de logiciels libres, plus ou moins heureuses, plus ou moins utilisées, répandues et importantes, et elles se divisent en deux grandes classes.

4.3.1 Les licences copyleft

Copyleft, ou «gauche d'auteur» est un jeu de mot expliquant que le principe du copyright est utilisé contre l'esprit original du copyright : une œuvre diffusée sous licence copyleft est fournie avec certaines libertés (comme celles de l'utiliser, la copier, la modifier, la diffuser) **et l'obligation** de ne jamais priver autrui de ces mêmes libertés.

Cela implique notamment que toute copie remise à un tiers devra être placée sous la même licence, et proposer le code source si ce tiers le demande.

L'exemple le plus connu de ces licences est la licence phare de la FSF, la GPL, ou General Public License de GNU, couvrant la majorité des projets logiciels libres. La FSF propose également une licence LGPL, originellement pour les bibliothèques de programmation et dont l'utilisation est désormais découragée (elle permet de faire une édition de liens avec des programmes propriétaires sous certaines conditions), et la GFDL, qui est le pendant de la GPL pour les documentations techniques.

Il est en particulier possible à un tiers de vendre un programme sous GPL, d'en tirer de substantiels profits, et de n'en rien reverser à l'auteur original. Ceci est voulu : plus le logiciel dispose de médias de diffusion, plus il sera accessible à tous. Le but de la FSF n'est pas de protéger les auteurs ou d'assurer leurs moyens de subsistance. Qui plus est, dans la plupart des cas cela impliquerait des comptes d'apothicaire impossibles à mener à bien, notamment dans le cas de nombreux auteurs ou des compilations. Une association pourra ainsi sans problème graver des CD et les revendre à ceux qui ont un accès Internet limité sans aucune papeterie.

4.3.2 Les autres licences

Parmi les autres licences de logiciels libres, les plus importantes historiquement et en volume sont les licences dites BSD ou X, très courtes et simples, utilisées par de nombreux projets logiciels et notamment par les BSD libres (FreeBSD,

NetBSD, OpenBSD).

La grande différence est qu'il est possible de partir d'un logiciel sous ces licences, de le retravailler, et de le diffuser sous forme propriétaire, binaire seulement, en se contentant de le signaler dans un petit coin de la documentation. C'est ce qu'a fait Microsoft dans des Windows récents pour quelques programmes, et c'est ce qu'a fait en particulier Apple avec Mac OS X, inspiré des BSD libres.

L'utilisateur peu averti ne sera donc pas protégé malgré lui, et pourra faire l'erreur de se laisser séduire par une version propriétaire d'un tel programme, pour lequel seul son éditeur pourra ensuite, éventuellement, lui proposer des correctifs.

5 La liberté selon la FSF

La FSF propose une définition simple et directe de la liberté du logiciel : un logiciel est libre si et seulement si il propose quatre libertés précises à son utilisateur.

5.1 Les quatre critères de la liberté

1. Liberté d'utiliser le logiciel, pour tout moyen. Ceci est acquis par défaut et ne nécessite nullement que l'utilisateur accepte implicitement ou explicitement la licence.
2. Liberté d'étudier le logiciel et son fonctionnement et de l'adapter à ses besoins. Il faut donc pour cela pouvoir accéder au code source. C'est l'«Égalité» de tous devant le logiciel.
3. Liberté de faire des copies du programme et de les remettre à ses amis, liberté d'aider le voisin. C'est la composante «Fraternité», et c'est une composante importante des idées véhiculées par la FSF. Les questions de rentabilité ou de financement des programmeurs ou des éditeurs de programmes passent au second plan derrière cela : ce n'est pas le problème de la FSF, même si elle peut suggérer des réponses.
4. Liberté d'améliorer le programme et d'en diffuser des versions modifiées pour que tous en profitent. Ainsi on crée une dynamique de communauté, ce que le créateur de la FSF a toujours voulu suite au démantèlement de son laboratoire de copains.

5.2 Les types d'œuvres et les conséquences

Richard Stallman distingue plusieurs types d'œuvres, et pense qu'elles relèvent de problématiques différentes dans le droit du copyright :

- les œuvres **fonctionnelles**, telles que les recettes de cuisine, les documentations techniques, les logiciels. Elles sont interchangeableables, tant que la nouvelle version remplit le même besoin. Il convient de les diffuser selon les conditions du logiciel libre, et si possible sous copyleft.
- les œuvres **d'opinion**, telles que les mémoires, les manifestes, les prises de position. Il convient d'en autoriser la copie verbatim, gratuitement ou non, pour en encourager la diffusion, mais toute modification du contenu est évidemment interdite.
- les œuvres **artistiques**, pour lesquelles la situation est moins claire. De nombreux classiques sont inspirés d'œuvres de leur époque depuis tombées dans l'oubli : Shakespeare, La Fontaine, ... On peut souhaiter en autoriser la diffusion non commerciale par exemple ; elles ne sont en effet pas interchangeableables.
-

6 Historique

Le logiciel libre est un exemple unique dans l'histoire de l'humanité de collaboration internationale par un grand nombre d'individus qui aboutit à un résultat très riche et complexe. Il y a d'un point de vue ingénierie de nombreuses pertes, déchets, redites, mais cela garantit une certaine qualité et la sélection naturelle s'exerce pour conserver en général les meilleures solutions.

Ceci est rendu possible grâce à Internet, qui permet de copier autant de fois que l'on veut, pour un prix dérisoire et d'un bout à l'autre de la planète, des volumes d'information d'une valeur ajoutée parfois considérable.

6.1 Internet

Internet et Unix ont démarré dans les années 1970. Internet était d'abord un réseau militaire, conçu de manière décentralisée pour résister à une attaque nucléaire. Il s'est peu à peu répandu dans les universités dans les années 1980 puis le grand public dans les années 1990.

Quel que soit votre centre d'intérêt, aussi pointu et spécialisé soit-il, vous serez pratiquement assuré de trouver sur Internet d'autres mordus qui le partageront et

auront déjà mis en place des documents et médias de communication à ce sujet. Qu'ils soient aux antipodes n'a presque aucune importance : c'est ce qu'on appelle le Village Global. La géographie ne vous contraint plus.

Si tu apportes une orange et moi une pomme et que nous les partageons, nous avons chacun deux demi-fruits. Si tu apportes une idée et moi un concept et que nous partageons, nous avons chacun deux informations. Tout est là.

6.2 GNU

Au début des années 1980, Richard Stallman a eu la douleur de voir l'industrie naissante de la micro-informatique vider le laboratoire d'intelligence artificielle du MIT de tous ses amis, et créer en quelque sorte le logiciel propriétaire.

Jusque là le logiciel était naturellement montré aux copains, partagé, fourni avec les machines. Quand il s'est avéré que c'était un poste lucratif, la donne a changé.

Il a résolu de changer le monde et s'est retroussé les manches, en commençant de rien. Il a codé pièce par pièce un système Unix libre et complet, GNU, sous les moqueries de ceux qui pensaient à raison qu'il s'attelait à une tâche titanesque. C'était compter sans son grand talent de programmeur, qui lui a valu de nombreux prix, et le fait que de plus en plus de monde l'a rejoint dans cette tâche.

Tous les outils système étaient prêts au début des années 1990, à l'exception du noyau, cœur du système. RMS et le projet GNU ont alors fait un choix techniquement osé, et mauvais tactiquement, et leur noyau Hurd n'est toujours pas pleinement fonctionnel de nos jours.

6.3 Linux

En 1991, un jeune étudiant en informatique finlandais appelé Linus Torvalds, souhaitant disposer de la puissance d'Unix sur son petit ordinateur personnel, a commencé la programmation d'un noyau selon d'autres choix techniques.

Il a rapidement trouvé que GNU complétait fort bien son œuvre. Le rêve de Stallman prenait corps : il était possible d'utiliser un ordinateur en ne faisant appel qu'à des logiciels libres.

C'est la raison pour laquelle la FSF souhaite que nous nous référions au «système Linux» sous le nom «GNU/Linux».

6.4 Open Source

Internet se développait à grands pas aux USA, et l'Europe suivait avec quelques années de retard : la presse avait trouvé un nouvel os, et les fournisseurs d'accès grand public se multipliaient dans les années 1990.

C'est le cocktail magique qui a permis que tout explose, et les projets applicatifs ambitieux se sont peu à peu montés sur la base GNU/Linux.

Une poignée d'Américains souhaitait redorer l'image de marque et améliorer la visibilité de «Linux» et a créé l'«Open Source Initiative» pour éviter d'utiliser le terme ambigu «free» en anglais en 1998, suite à une décision de Netscape de libérer leur navigateur dans leur guerre avec Internet Explorer, le navigateur de Microsoft.

7 Aspects humains et psychologiques

7.1 La motivation, la concurrence

Tout le monde a accès aux mêmes informations — évidemment, ceux qui ont plus d'expérience, qui ont écrit le code eux-même, ou qui ont des bases théoriques plus solides ou un esprit plus vif sont avantagés. Tout le monde travaille donc dans la plus parfaite transparence, et seuls les critères techniques sont utilisés pour les jugements de valeur.

Les projets se tirent parfois la bourre (tels que GNOME et KDE, deux environnements de bureau). Des programmeurs doués font des concours de chevilles enflées. C'est l'émulation généralisée.

De plus, travailler sur un projet copyleft est une motivation essentielle pour certains, qui vivraient mal que leur contribution tombe sous des fourches caudines.

Les passagers clandestins sont nombreux, et on est toujours le passager clandestin d'un autre projet. Qu'importe : aussi incroyable que cela pouvait paraître dans les années 1980, au début de GNU, il est possible que des milliards de dollars d'équivalent industriel en développement logiciel s'écrivent tous seuls. Et jusqu'à présent il a toujours existé suffisamment de contributeurs pour cette dynamique continue ; après tout les passagers clandestins ne privent personne de rien.

7.2 Les communautés

C'est ainsi que se montent et évoluent un certain nombre de petites communautés autour de projets logiciels. Quelques chefs de projet autoritaires et charismatiques, une poignée de développeurs principaux, et une armée de débogueurs, utilisateurs exprimant plus ou moins bien leurs souhaits, les constituent.

Aussi incroyable que cela paraisse, tout cela tourne, et c'est expliqué dans les essais d'Eric Raymond sur CatB.org, notamment dans *The Cathedral And The Bazaar*.

8 Aspects industriels

Nous avons vu que le logiciel libre avait une définition technique et éthique, et que les aspects économiques passaient au second plan. Question de priorités.

Il n'en demeure pas moins que ce bouillon de culture psychologique a des effets secondaires très intéressants dans un contexte industriel.

8.1 Les business plans

Il est beaucoup plus difficile, presque impossible, de faire de l'argent en vendant des licences de logiciels.

Et alors ?

Environ 5% des programmeurs dans le monde ont un salaire qui dépend directement de ce type de vente. La plupart des autres font du service, de la maintenance, ou développent des programmes à façon (du logiciel privé).

De plus, les incompatibilités entre les différents systèmes, impossibles à éviter en l'absence du code source pour vérifier que personne ne triche et de toutes manières tolérées par les législateurs, ont abouti à une situation de *Winner Takes All*, avec une oligopole confinant au monopole. D'un point de vue économique, les ventes de licences constituent des fuites de capitaux pour engraisser des géants. C'est une forme d'impôt nouveau, régulièrement prélevé.

8.2 Intérêts du logiciel libre

Quoi que certains en pensent, je ne suis pas convaincu que la liberté soit un élément qui ait la moindre importance dans aucune entreprise d'une certaine taille.

Le logiciel libre confère cependant trois indépendances, toutes trois fort intéressantes :

- une indépendance **technique** : si le fournisseur change ses conditions de maintenance ou son service après-vente de manière unilatérale, ou se comporte de manière insupportable, il est envisageable d'en changer si l'on dispose du code source (surtout si ce dernier correspond à un logiciel libre non privé, publié sur Internet et susceptible d'être connu par d'autres). On pourra au pire faire l'effort de rentrer dans le code.

La revue des pairs ne confère pas une qualité parfaite et maximale : on trouve régulièrement des failles de sécurité grotesques, conséquences d'erreurs d'inattention que personne n'a pris la peine de lire par la suite. Le corpus est énorme ; on compte des millions voire des milliards de lignes de code. Mais une telle sécurité n'est possible que pour quelques agences gouvernementales richement dotées car elle suppose une revue exhaustive et poussée du code par des spécialistes.

Dans le cas pratique qui nous intéresse, le logiciel libre est la solution approchée la meilleure qu'on puisse espérer étant données les contraintes du monde physique et les ressources limitées.

Les communautés réagissent très vivement aux bogues les plus graves ainsi qu'à la plupart des remarques ou des questions, et produisent des logiciels généralement de fort bonne tenue.

Un particulier pourra tenter sa chance dans le code, poser la question sur Internet, ou demander au cousin ou au fils de la concierge de lui donner un coup de main. Il ne sera plus frustré par des services techniques incompétents qui se fichent du monde.

- une indépendance **politique** par rapport aux pays des grands éditeurs de logiciels, qui peut avoir son importance pour les États ou les grandes entreprises.
- une indépendance **économique** enfin, car il est souvent possible et facile d'obtenir du logiciel libre de manière gratuite, sans être l'otage des éditeurs, de leurs mises à jours, de leurs bogues savamment mis en place pour nous forcer la main, et de leurs prétendues assistances.

8.3 Le coût du passage

Si on a la malchance de ne pas partir de zéro et d'avoir un corpus documentaire ou des habitudes prises sur systèmes propriétaires, le passage au logiciel libre sera plus difficile et nécessitera un certain investissement ou une motivation solide.

Après tout, les conseillers en communication des éditeurs propriétaires distillent au compte-gouttes les couleuvres qu'ils font avaler année après année à leurs uti-

lisateurs, et prennent garde de procéder progressivement, sans jamais franchir la ligne jaune. Il faut un peu de recul et d'analyse pour voir la tendance générale et craindre pour l'avenir du traitement d'informations, qui occupera un part de plus en plus centrale dans les vies.

9 Aspects administratifs

Les habitudes sont dures à perdre, l'informatique fait peur à tout le monde, ennuie les politiciens, et les lobbies du logiciel propriétaire ont compris depuis les années 1980 qu'il ne fallait plus rigoler. Ils mettent en place des cellules de surveillance et accordent une extrême attention à ces questions, tous les moyens sont bons. Il y a énormément d'argent et de pouvoir en jeu.

Certains souhaitent imposer par voie légale, par exemple dans les appels d'offre, le logiciel libre. Après tout, on impose bien aux industriels de l'agro-alimentaire d'afficher la liste de leurs ingrédients. Les changements de gouvernements et les pressions aidant, on fait souvent trois pas en avant pour deux pas en arrière.

Donnons une liste non exhaustive de liens et de lieux où les discussions ou des annonces publiques ont lieu ou ont eu lieu sur ces sujets :

- Projets de loi en France et en Belgique : <http://wiki.april.org/Terminologie>
- Projets de loi en Belgique : <http://wiki.ael.be/index.php/ProjetOrdonnanceLibre>
- L'Inde se déclare neutre : <http://groups.google.com/groups?selm=Pine.LNX.4.50.0304040352100.7055-100000%40news.goa.bytesforall.org>
- Extrémadurie : <http://www.washingtonpost.com/ac2/wp-dyn/A59197-2002Nov2>
- Résolution de l'université de Buffalo à New York : http://orange.math.buffalo.edu/csc/resolution2_april2003_approved.html
- Costa Rica : <http://groups.google.com/groups?selm=Pine.LNX.4.50.0304080220090.3447-100000%40news.goa.bytesforall.org&output=gplain>

10 Les prochains défis à relever

10.1 Documentation technique

Richard Stallman avait 15 ans d'avance en 1983, et de nos jours il souhaite étendre le mouvement du logiciel libre aux encyclopédies, aux manuels scolaires, et aux documentations techniques de manière générale.

10.2 Ne pas se laisser distancer

Les logiciels propriétaires font de la surenchère de fonctionnalités et d'apparence. La plupart du temps, c'est inutile : à quoi bon avoir des boutons en 3D et des reflets ? Qui utilise la plupart des fonctionnalités des mammoth logiciels qu'il utilise ? Les ordinateurs de la conquête spatiale de 1969 étaient moins puissants que nos actuelles calculatrices, les monstres que nous utilisons pour jouer à de petits jeux futiles leur feraient bien envie.

Ce discours a beau être juste, c'est un combat d'arrière-garde. Quoi qu'on dise, il faut être sexy pour avoir une chance de tenir le haut du pavé, et ne pas se laisser distancer même par les nouveautés futiles.

Il faut également veiller à comprendre en permanence les formats de sauvegarde abscons utilisés par les logiciels propriétaires, évidemment non ou mal documentés, ainsi que les protocoles de communication réseau si l'on veut avoir une chance d'interopérer.

10.3 Éduquer

Chaque génération, chaque classe d'âge, compte une fraction d'individus doués pour l'ordinateur et la chose informatique. J'ai eu la chance de manipuler des micros 8 bits des années 1980. Même s'ils étaient peu puissants et si leurs langages de programmation me font aujourd'hui sourire, ils m'ont formé et permis de faire mes premières armes. Une majorité des étudiants et des professionnels de l'informatique sont des bébés-micros.

Une génération entières a été gâchée dans les années 1990 par le fait que les seuls ordinateurs vendus dans le commerce ne fournissaient pas par défaut d'environnement de programmation. Ils en étaient réduits à exercer leur curiosité et leur malice à des stupidités telles que la manipulation astucieuse des suites bureautiques ou des outils de configuration du système. Quel gâchis.

Il sera plus difficile de sensibiliser les utilisateurs passifs, qui ne veulent pas entendre parler de choses compliquées et pour qui l'ordinateur n'est qu'un outil capricieux mais incontournable, à ces questions. L'intérêt leur sera moins évident et immédiat, et le coût du passage les rebutera. Je n'aurai jamais le temps d'exciter leur curiosité et leur patience suffisamment pour leur expliquer suffisamment de choses sur lesquels je souhaiterais qu'ils s'interrogent avant d'être classé doux dingue illuminé.

Ils forment néanmoins l'essentiel du peloton, Et ce sont peut-être d'autres exemples, tels que ceux que je vais maintenant donner, qui les aideront à comprendre l'im-

portance et la portée de la maîtrise du traitement automatique des informations.

10.4 Les brevets logiciels

Les brevets logiciels sont une autre perversion moderne des anciennes lois ayant mis en place les patentes et le droit d'auteur. Ils sont copieusement détournés de leur but premier par les grandes multinationales avec la complicité de bureaux des brevets qui ne s'acquittent pas de leur tâche. Ils permettent d'écraser un astucieux petit concurrent avec une menace de procès (qui serait perdu par la grande entreprise s'il était mené à terme et si le petit avait les moyens de se défendre). Certains permettent l'évasion fiscale, et ne sont ainsi jamais utilisés en justice. Ils représentent un très mauvais moyen d'évaluer l'innovation et le capital intellectuel d'une entreprise, mais c'est l'un des seuls dont on dispose : on préfère donc l'utiliser plutôt que de remettre en question les procédures d'évaluation.

Le copyright ne protège pas les idées, mais leur expression. On peut écrire un livre d'histoire libre relatant les mêmes événements qu'un livre d'histoire propriétaire à condition de le rédiger à nouveau.

Le brevet logiciel protège les idées, et parfois les plus simples et les plus absurdes. Un Australien aurait récemment déposé un brevet sur la roue, en la décrivant en des termes pompeux. Les exemples grotesques abondent.

Le brevet logiciel est une épée de Damoclès, un instrument de terreur : il est impossible de savoir ou de garantir qu'un code n'en viole aucun, et le détenteur du brevet peut frapper à tout instant, y compris des années plus tard. Le logiciel libre est particulièrement fragile car il est très aisé de montrer, en examinant le code source publiquement affiché, qu'il utilise une technique brevetée.

Un brevet pourra interdire à tout logiciel libre d'implémenter une technique de codage ou décodage d'image, de musique, de vidéo, interdisant ainsi au logiciel libre de proposer les mêmes fonctionnalités que le logiciel propriétaire.

Si l'on avait appliqué les idées du brevet logiciel à la musique des siècles passés, Bach ou Beethoven n'auraient rien pu écrire : leurs innovations ne pouvaient s'exprimer qu'au sein d'une œuvre empruntant aux autres morceaux de l'époque, des idées musicales.

Le brevet logiciel n'est pas l'ennemi du logiciel libre, c'est l'ennemi de l'innovation. Aucune étude économique ne conclut qu'il peut la favoriser, c'est pourtant le leitmotiv sans cesse répété par ses défenseurs. C'est la consécration de la loi du plus puissant, au détriment de l'intérêt de la majorité. Il concerne toutes les PME, qu'elles fassent ou non du logiciel, libre ou non.

Les multinationales américaines font pression depuis plusieurs années pour que l'Europe rende le logiciel brevetable. Il ne l'était pas, même si cela était régulièrement contourné et transgressé. Eurolinux, FFII et EUCD.info sont des sites Web d'information sur ce sujet ou des sujets proches.

Note : tout ceci ne préjuge en rien de l'intérêt ou non du brevet dans d'autres domaines que le logiciel.

11 Quelques exemples dans d'autres domaines

Le logiciel système est rébarbatif et réservé aux passionnés de technique et autres professionnels de l'informatique.

Même les logiciels applicatifs sont ennuyeux : un traitement de textes écrit des lettres, un tableur fait de la comptabilité.

Le logiciel n'est pas une fin en soi. De même que ces émissions de télévision qui parlent d'autres émissions de télévision dans une spirale nombriliste vaine et sans fin, le média de l'ordinateur et de l'Internet n'a d'intérêt que s'il parle de la vraie vie.

L'alchimie créative Internet + communautés réparties en villages globaux + motivation psychologique de participer plus vite, plus haut, plus fort que le voisin n'est pas propre au logiciel libre, même si pour l'instant c'est sans doute le domaine où les créations sont les plus convaincantes.

Elle vaut pour toutes les informations à forte valeur ajoutée qui peuvent circuler sur Internet et être compilées par une communauté éparse mais organisée.

11.1 Jeux

Commençons par une exception : les jeux vidéo. Ce sont techniquement des logiciels, mais ils ont pour but de divertir, de détendre. Ne nous voilons pas la face, ne jouons pas les sainte-nitouches en feignant réserver cette activité légère aux enfants : de plus en plus d'adultes jouent au jeux vidéo, qu'ils ont connus petits, et les autres jouent à d'autres jeux, guère plus intellectuellement stimulant, mais socialement mieux acceptés.

L'homme est fait pour jouer, c'est le pêché originel qui l'a condamné au travail. Dont acte !

Il me faut reconnaître ici que les jeux en logiciel libre ne sont pas vendeurs. His-

toriquement, les fous d'informatique ne se sont pas privés de coder des amusettes en parallèle des logiciels système. De manière peu inattendue, ces divertissements reflétaient leurs goûts, qui ne sont en aucun cas ceux du grand public. Ils ont d'exceptionnelles qualités ludiques, on peut se passionner pour des caractères se déplaçant sur l'écran en noir et blanc et y consacrer des nuits et des mois, mais vous m'accorderez que ce n'est pas vendeur.

La situation a progressé depuis, et on trouve quelques jeux plus présentables, capables d'exciter l'intérêt sans d'abord obliger à s'y plonger et à franchir l'obstacle psychologique d'une interface fruste. Ce sont des jeux en 2D, au niveau technique de ce qui se faisait dans les années 1980.

Les développeurs sont souvent bénévoles, travaillant sur leur temps libres (quelques autres projets de logiciels libres majeurs sont partiellement financés par de grandes sociétés). Ils n'ont absolument pas accès aux moyens et aux studios nécessaires à la création des jeux modernes.

Les jeux les plus vendeurs (et non pas forcément les plus ludiques) sont à mon sens FreeCiv, clone de Civilization, et FreeCraft, clone de WarCraft II.

11.2 Encyclopédies

C'est la nouvelle frontière. Un responsable de Wikipedia a même promis que d'ici 10 ans les encyclopédies propriétaires auraient vécu et qu'il leur souhaitait de crever, car les encyclopédies libres auraient été complétées et bien meilleures.

Nupedia est une encyclopédie très sérieuse, avec un long processus de rédaction, revue, correction, commentaires, par des spécialistes des domaines. Elle compte pour l'instant quelques dizaines d'articles.

Wikipedia est un projet qui a vu le jour suite à une frustration de la lenteur de la progression de Nupedia. Wiki est un protocole permettant d'écrire un document à plusieurs sur interface Web ; certains des liens donnés ici sont des Wiki. Elle a rapidement eu un succès foudroyant, et sa version anglaise compte depuis le 15 janvier 2001, plus de 114 mille articles à ce jour, dont la plupart sont très corrects.

Des traductions partielles existent en plusieurs langues, mais à ce niveau il est bien plus facile de lire l'anglais : c'est après tout la langue qui permet à des gens du monde entier de communiquer, et les Français sont particulièrement favorisés dans l'apprentissage de l'anglais étant donnée sa similarité avec leur langue maternelle.

On trouve encore une «encyclopédie» musicale, présentant des partitions. Les règles de droit d'auteur sévissant en musique sont encore plus dures que dans d'autres domaines, cette encyclopédie se résume donc à d'anciennes partitions,

qui ont plusieurs siècles, mais à ce jour le projet Mutopia compte 274 morceaux.

11.3 Littérature

Aux États-Unis, aucune œuvre n'est tombée dans le domaine public depuis la deuxième guerre mondiale, et certains prédisent que plus jamais, plus aucune œuvre n'y tombera : les majors concernées exercent en effet un lobby intense pour prolonger régulièrement la durée du copyright, avec effet rétroactif. Cela s'est produit plusieurs fois au vingtième siècle, à chaque fois que le personnage de Mickey Mouse menaçait de pouvoir être utilisé librement sans l'autorisation de Disney — il a donné son nom à ce phénomène. Ce ne sont pas les seuls ; la France a procédé de même dans les années 1990, passant de 50 à 70 ans après la mort de l'auteur avec effet rétroactif, et *le Petit Prince* n'est toujours pas domaine public à cause de cela.

De plus, la charge de la preuve est malheureusement dans le camp de celui qui souhaite utiliser une œuvre : à lui de vérifier la date de la mort de l'auteur, s'il a été prisonnier de guerre, les autres cas particuliers. Les États, qui prêtent le concours de leur force publique au respect des ponctions des ayants-droits et de leurs caprices en matière de représentations de l'œuvre de leur grand-père ou aïeul, ne proposent pas pour autant une liste exhaustive et officielle des œuvres protégées et de l'instant précis où elles tombent dans le domaine public. Dans le doute, il convient donc d'attendre un peu plus...

Le projet Gutenberg, démarré dans les années 1970, est le fait d'un homme qui a eu l'idée géniale de numériser la littérature américaine domaine public. Ils ont commencé doucement, les ordinateurs de l'époque disposant de faibles capacités, par les textes qui leur tiennent à cœur : déclaration d'indépendance, divers discours de présidents célèbres. Ils ont ensuite passé près de 10 ans sur la seule Bible, et depuis les années 1990 tout s'emballe et le nombre de livres disponibles croît sans cesse, avec une liste d'attente de promesses sur plusieurs années. Fin 2002, il y avait plus de 6200 livres numérisés, dans plusieurs langues. Je regrette toutefois leurs choix techniques en matière de représentation des accents, en anglais ou dans les langues étrangères comme le français.

Le principe du projet Gutenberg est qu'une personne se charge entièrement d'un livre qui lui plaît, ce qui peut être très long. C'est alors que voici peu, un autre homme a eu une idée géniale : distribuer le travail pour chaque livre, page par page, sur l'Internet. Distributed Proofreading étaient né. Le principe est de masicoter des volumes anciens mais sans valeur, de mettre le tas de feuilles qui en résulte dans un chargeur automatique de scanner, et de lancer le programme de

reconnaissance optique de caractères (ROC). En quelques minutes, on dispose d'une version texte des pages de très bonne qualité. Les photos et leur ROC sont ensuite mises sur le serveur et relues et corrigées manuellement par ceux qui s'inscrivent pour cela. Après deux passes et la mise en place de certaines conventions de codage, les pages de texte sont rassemblées par des volontaires.

L'équivalent français de ces questions est l'Association des Bibliothécaires Universels mais elle semble morte depuis plus d'un an et je regrette leurs contraintes en matière de licence. Le projet Gutenberg s'attaque désormais aux livres en langues étrangères dont le français de toutes façons. Signalons également que le projet Gallica de la BNF propose la numérisation de nombreux livres sous forme d'images, et de certains sous forme de texte.

11.4 Autres

DMOZ est un répertoire (*directory*) collaboratif : il permet d'organiser les informations disponibles sur le Web de manière hiérarchique, avec des commentaires sur la valeur des pages ainsi pointées.

Usenet est un sous-réseau d'Internet très actif depuis les années 1970. Au début, il ne fonctionnait pas toujours en temps réel, et les messages qui circulaient sur ses groupes de discussion atteignaient parfois les différents endroits du globe avec beaucoup de retard. On y trouve actuellement des milliers de groupes, traitant de tous les sujets possibles. De nombreux groupes ont compilé au fil du temps des Foires Aux Questions et des archives, cristallisant et synthétisant en quelque sorte les discussions tenues. On trouvera la plupart de ces documents sur FAQS.org.

On apprendra ainsi la route, hectomètre par hectomètre, permettant de se rendre de Londres au lieu de tournage du *Prisonnier*. Ou encore la liste des briques de Lego disponibles et leurs tailles au dixième de millimètre près. On trouvera des centaines de rumeurs qui ont circulé et les conclusions de l'enquête à leur sujet. Les professeurs de mathématiques soucieux d'égayer leurs exercices ainsi que les animateurs de colonies de vacances s'y inspireront avec joie. La somme de connaissances ainsi consignées semble infinie — mais attention de ne pas oublier l'esprit critique au vestiaire. J'ai appris le point de vue des Créationnistes des USA : ils soutiennent que la Terre a seulement quelques milliers d'années, et que Dieu dans la toute-puissance qui le caractérise a mis en place des fossiles et réglé leur carbone 14 pour tromper les chercheurs qui mettraient Sa parole en doute.

12 Bibliographie

(Ajouter **http://www.** ou **http://** devant les adresses données. On trouvera sur **http://linuxshop.ru/linuxbegin/win-lin-soft-en/** une liste de programmes libres pour remplacer un certain nombre de produits propriétaires)

APRIL — **april.org**. Association pour la Promotion et la Recherche en Informatique Libre. Partenaire français de la FSF.

AFUL — **aful.org**. Association Francophone des Utilisateurs de Linux et de Logiciels Libres. Plus récente qu'APRIL, plus orientée Open Source, ciblant plus les institutions et le lobbying.

GNU — **gnu.org**. GNU's Not Unix. Projet de Richard Stallman de construire un système compatible Unix entièrement libre.

FSF — **fsf.org**. Free Software Foundation.

FSF Europe — **fsfeurope.org**. Free Software Foundation, bureaux et actions en Europe.

FSF France — **france.fsfeurope.org**. Free Software Foundation Europe. branche française.

Stallman.org — **stallman.org**. Écrits et prises de position de Richard Stallman sur de nombreux sujets annexes et connexes aux logiciels et aux ordinateurs.

Open Source Initiative — **opensource.org**. Campagne de marketing du logiciel libre sur des critères techniques.

NUPEDIA — **nupedia.org**. Encyclopédie libre avec des procédures très strictes, comptant quelques dizaines d'articles.

WikiPedia — **wikipedia.org**. Encyclopédie libre beaucoup plus souple, comptant plus de cent mille articles.

Mutopia — **mutopiaproject.org**. Projet de rassemblement des partitions musicales de domaine public.

FreeCraft — **freecraft.org**. Jeu libre, clone de WarCraft II, capable d'en utiliser les graphiques mais disposant de jolis graphiques propres.

FreeCiv — **freeciv.org**. Jeu libre, clone de Civilization.

Mozilla — **mozilla.org**. Navigateur libre, initialement issu de Netscape Navigator, libéré en 1998.

KDE — **kde.org**. Environnement de bureau libre et tous ses outils.

GNOME — **gnome.org**. Environnement de bureau libre et tous ses outils.

Bochs — **bochs.sf.net**. Émulateur de machine virtuelle, permettant d'employer un autre système d'exploitation à l'intérieur d'une fenêtre.

Wine — **winehq.com**. Émulateur de Microsoft Windows permettant d'employer des applications Microsoft Windows sur d'autres systèmes.

GIMP — **gimp.org**. Programme de dessin d'art, fonctionnellement équivalent à PhotoShop.

Gnumeric — **gnnumeric.org**. Tableur libre, capable de lire et d'exporter des données au format Excel.

Open Office — **openoffice.org**. Suite bureautique libre, capable de lire et d'exporter des données au format Microsoft Word.

SourceForge — **sourceforge.net**. Site Web regroupant près de soixante mille projets de développement libres et plus de six cent mille utilisateurs répertoriés.

Savannah — **savannah.gnu.org**. Site Web regroupant plus de 1500 projets de développement libres et plus de quinze mille utilisateurs répertoriés.

Knoppix — **knoppix.org**. CD amorçable d'une distribution GNU-Linux s'exécutant entièrement en mémoire, sans rien inscrire sur le disque dur, avec détection automatique du matériel.

DemoLinux — **demolinux.org**. CD amorçable d'une distribution GNU-Linux s'exécutant entièrement en mémoire, sans rien inscrire sur le disque dur, avec détection automatique du matériel.

Debian — **debian.org**. Distribution de GNU-Linux comptant plus de 8700 paquetages logiciels et développée de manière associative, avec un contrat social.

Linux Mandrake — **linux-mandrake.com**. Distribution de GNU-Linux mise au point par une société parisienne.

Red Hat — **redhat.com**. Distribution de GNU-Linux mise au point par une société californienne.

FreeBSD — **freebsd.org**. Système BSD libre.

NetBSD — **netbsd.org**. Système BSD libre.

OpenBSD — **openbsd.org**. Système BSD libre.

Kernel.org — **kernel.org**. Site de distribution officiel du noyau Linux.

Hurd — **www.gnu.org/software/hurd**. Noyau du système GNU.

Eurolinux — **eurolinux.org**. Site d'information sur les brevets logiciels.

FFII — **ffii.org**. Site d'information sur les brevets logiciels.

EUCD — **eucd.info**. Site d'information sur les menaces concernant la copie privée.

Project Gutenberg — **promo.net/pg**. Site Web centralisant la numérisation des œuvres littéraires de domaine public, principalement en langue anglaise.

Distributed Proofreading — **texts01.archive.org/dp**. Site Web distribuant le travail de relecture de la reconnaissance de caractères de pages numérisées pour le projet Gutenberg.

ABU — **abu.cnam.fr**. Site Web centralisant la numérisation des œuvres littéraires de domaine public, principalement en langue française.

Gallica — **gallica.bnf.fr**. Organe de la bibliothèque nationale de France proposant de nombreux livres sous forme numérique, images et/ou texte.

DMOZ — **dmoz.org**. Open Directory Project : projet de répertoire libre du Web.

CatB.org — **catb.org**. Essais et réflexions sur le logiciel libre et Open Source, ses modes de développement, etc.

FAQs.org — **faqs.org**. Collection de foires aux questions et d'archives de groupes de discussion Usenet.