

# De l'hacktivism au web 2.0 – De la révolution à sa dissolution ?

Les mouvements alternatifs d'émancipation portés par le numérique, dont le logiciel libre fait partie, ont été récupérés et domestiqués par le système et sa force marketing sous la bannière et le vocable du [« web 2.0 »](#).



Un web 2.0 qui présente de plus l'avantage de favoriser l'institution d'une sorte de totalitarisme global décentralisé avec notre complicité et toutes les traces personnelles, permanentes et continues, que nous laissons, le plus souvent volontairement, dans les nuages d'Internet.

Avec l'avènement du web 2.0, non seulement vous voyez s'éloigner le rêve d'une *autre* société mais vous renforcez le contrôle et la surveillance de l'actuelle ! Difficile de faire pire en quelque sorte...

Telle n'est pas mon opinion mais mon propre (et donc faillible) résumé d'un article parcouru récemment sur [Indymedia](#) dont le titre exact est ["Become the media!" : de l'hacktivism au web 2.0](#).

Attention, c'est dense, politisé et plein de références à des auteurs qui vous seront peut-être peu familiers si vous ne baignez pas dans une certaine culture intellectuelle « de gauche » (cf [Félix Guattari](#), [Jello Biafra](#), [Walter Benjamin](#), [Jean Baudrillard](#), [Gilles Deleuze](#), [Michel de Certeau](#), [Michel Maffesoli](#)).

Nous en avons reproduit la fin ci-dessous pour vous donner (ou

non) l'envie de le parcourir dans son intégralité<sup>[1]</sup>.

Je ne vous cache pas qu'il m'intéresse d'avoir vos réactions dans les commentaires. En espérant que les uns et les autres sauront s'écouter et échanger en toute sérénité sur un sujet, je le reconnais bien volontiers, un peu *glissant*. Un petit débat courtois et non un gros troll poilu pour le dire autrement ☐

Pour ce qui me concerne, je ne partage pas la radicalité et le pessimisme du propos et j'ai justement l'impression que les actions que nous menons participent modestement à échapper à ce piège. Mais il est vrai que lorsque le « logiciel libre » devient « open source », il prend le risque de perdre en route tout ce qui fait sa [substantifique moelle](#).

## **“Become the media!” : de l'hacktivism au web 2.0 (extraits)**

[URL d'origine du document](#)

*Dr No – 26 juillet 2010 – Indymedia Nantes*

(...)

Quoiqu'il en soit, ce dont il s'agit là encore finalement, avec cette « réappropriation », ce « devenir-media » de la masse et cette « démocratisation » des dispositifs d'informations et de communication, c'est du déploiement toujours plus important d'un macro-système technique, d'un maillage global comme dispositif de socialisation forcée par dressage à la discipline inconsciente d'un code, c'est-à-dire – à l'instar du système électoral ou de la consommation – d'imposition de règles du jeu (ici de la communication) et d'intériorisation de ces règles comme subtil mode de mobilisation et de contrôle social. Indépendamment des contenus qui n'en sont que l'alibi, le médium – le code, le modèle, la forme, le canal, le dispositif, la technique – est

le message, il influe directement sur nos modes de perception sensibles, modifie nos rapports à l'espace et au temps et par conséquent nos modes d'être-au-monde. En l'occurrence, « ce qui est médiatisé, ce n'est pas ce qui passe par la presse, la TV, la radio : c'est ce qui est ressaisi par la forme/signe, articulé en modèles, régi par le code. » La réappropriation du code ne jouant donc là au final que comme « reproduction élargie du système » sous couvert de nouvelles modalités. C'est pourquoi il ne faut jamais sous-estimer la capacité de ce système à intégrer les innovations (même et peut-être surtout si elles se veulent « révolutionnaires ») a fortiori si celui-ci fonctionne sur les principes d'interaction, de réversibilité, de participation et de feed-back comme c'est d'ailleurs le cas aujourd'hui beaucoup plus qu'hier.

**« l'éthique hacker », l'utopie cyberpunk et les expérimentations cyberculturelles, les trouvailles de « l'hacktivism » électronique et de « l'Internet militant », du mouvement des logiciels libres, l'Open Source, l'Open Publishing, le P2P, le Wi-Fi, les média-tactiques alternatives, collaboratives et communautaires elles-mêmes, c'est-à-dire en somme toutes ces « pratiques moléculaires alternatives » que Félix Guattari appelaient de ses vœux pour renverser le pouvoir grandissant de l'ingénierie logicielle et les nouvelles modalités de la « société de contrôle » ont pour la plupart, dans ce qu'elles avaient d'original et novateur, été absorbées et recyclées par celle-ci et les industriels pour donner naissance à ce que l'on peut appeler les nouveaux « agencements post-médiatiques » du web 2.0.**

C'est-à-dire toutes ces nouvelles applications de l'Internet « participatif » et « collaboratif » basé sur le principe du « contenu généré par les utilisateurs », ce qui précisément, on l'aura remarqué, était bien l'idée de « l'open publishing » (publication libre) proposé par le réseau international des sites Indymedia dans l'esprit du partage horizontal de l'information, de la participation et de la collaboration en

vue de favoriser l'auto-organisation des groupes et des individus constitués en « machines de guerre » contre l'axiomatique mondiale exprimée par les Etats.

Un Web 2.0 dit « participatif » et « collaboratif » donc, où effectivement, convergence numérique aidant, la masse devient son propre média (MySpace, Facebook, YouTube, Twitter, Wikis et autres blogs), engendrant à leur tour de nouveaux usages qui inspirent également de nouveaux produits, services et dispositifs reconfigurant de fond en comble notre rapport au monde et nos relations sociales, tout en développant de nouveaux marchés ainsi que de nouveaux « business models » (management 2.0, marketing 2.0, « gratuité », « co-création de valeur », etc.) qui incarnent des changements de paradigmes économiques par où se joue la mutation du capitalisme. Car en effet, force est de constater que les principes du « participatif », du « collaboratif », de la « coopération » et du « partage » sont aujourd'hui devenus les principaux éléments d'un nouvel esprit du capitalisme de l'ère 2.0 fonctionnant par « boucles de récupération » et recyclage écosystémique des singularités comme moteur et dynamique de l'innovation (technologique, économique, culturelle, sociale, etc.). C'est en quelque sorte ce qui se présente plus communément aujourd'hui sous l'appellation d' « innovation ascendante » qui consiste justement pour les entreprises et/ou les institutions à observer, et même à favoriser, les pratiques de réappropriation, investissement, exploration, détournement, expérimentation par les usagers/consommateurs des produits, services et technologies dans le but de réintégrer les éventuelles micro-inventions et les « usages innovants » dans leur propre processus de création et développement industriel, commercial, technocratique, etc.

C'est une dynamique qui s'appuie sur la compréhension des comportements que permet en l'occurrence la « sociologie des usages » et notamment les travaux de Michel de Certeau sur ce qui constitue en quelque sorte les « arts de faire avec » .

Recherche qui se voulait un travail de compréhension et en premier lieu de mise en valeur des arts de vivre la société de consommation, par élaboration de « lignes de fuites » (Deleuze et Guattari) pourrait-on dire, c'est-à-dire plus particulièrement des ruses subtiles, des tactiques de résistance, de contournements, détournement, réappropriation, braconnage, dissimulation, en somme toute la multitude de pratiques inventives et créatives qui se disséminent dans la banalité du quotidien des usagers/consommateurs et que la rationalité occidentale, selon les mots de l'auteur, aurait eu trop tendance à occulter. Et on pourrait voir dans ce travail la saisie de l'essence même de la notion anglo-saxonne de « hacking », de son esprit ou de son éthique élargie à l'ensemble de la société. Quoiqu'il en soit, on le voit bien, ce dont il s'agit avec « l'innovation ascendante » mise en œuvre dans le nouveau paradigme économique des entreprises les plus à l'avant-garde du capitalisme c'est de capter/capturer la puissance créatrice de la socialité de base, l'énergie et le vitalisme qui émergent de ce que Michel Maffesoli appelle la « centralité souterraine ». Dans le même ordre d'idée se développe aujourd'hui dans les milieux du marketing et du management, par le biais des différentes plateformes multimédias de la société en réseaux, le « crowdsourcing » (approvisionnement par la foule) qui consiste pour une entreprise là encore à faire « participer » et « collaborer » directement la foule des internautes comme usagers/consommateurs à la recherche et au développement de nouveaux produits et services, à apporter des améliorations, etc..

Enfin, toutes choses mettant en œuvre un processus communicationnel global s'appuyant sur des dispositifs de « feed-back » et des mécanismes circulaires tout à fait caractéristiques des boucles causales rétroactives qui furent à la base de la modélisation des systèmes cybernétiques qui simulent les lois de la nature et dont la finalité, rappelons-le, est le Contrôle par auto-régulation comme mode de

management et de gouvernance.

Des systèmes de contrôle et de gouvernance de l'ère des machines de « troisième espèce » qui se déploient sur toute l'étendue de la vie quotidienne par le biais de la globalisation d'un méga-réseau engagé dans un processus matriciel. Une « matrice communicationnelle », un maillage systémique à vocation ubiquitaire qui tend par ailleurs à rendre obsolètes les modèles panoptiques de surveillance hyper-centralisés et transcendants de type orwellien qu'incarne la fameuse figure de « Big Brother ». Car en effet, ce à quoi on a de plus en plus nettement affaire aujourd'hui c'est à un processus de capillarisation du Contrôle en quelque sorte et qui tend par là à devenir totalement immanent.

Comme le remarquait déjà pertinemment Jean Baudrillard au début des années 70 « même à long terme, l'impossibilité des mégasystèmes policiers signifie simplement que les systèmes actuels intègrent en eux-mêmes, par le feed-back et l'autorégulation, ces métasystèmes de contrôle désormais inutiles. Ils savent introduire ce qui les nie comme variables supplémentaires. (...) Ils ne cessent donc pas d'être totalitaires : ils réalisent en quelque sorte l'idéal de ce que l'on peut appeler un totalitarisme décentralisé. » Par ailleurs, dans son texte annonçant l'avènement d'une "subjectivité post-médiatique" Félix Guattari rappelait que toutes les anciennes formations de pouvoir et leurs façon de modéliser le monde avaient été déterritorialisées. C'est ainsi, disait-il, que « la monnaie, l'identité, le contrôle social passent sous l'égide de la carte à puce. » Car en effet, ce qui se joue aujourd'hui avec tout ce maillage systémique planétaire, ce déploiement du méga-réseau matriciel à vocation ubiquitaire, c'est un processus de globalisation des « sociétés de Contrôle », fluides, ouvertes, modulaires, multipolaires et à géométrie variable comme installation d'un nouveau régime de domination qui remplacent peu à peu les « sociétés disciplinaires » (Foucault) avec la crise

généralisée des milieux d'enfermement en système clos (familles, écoles, armée, usines, prisons, hôpitaux, etc.) ainsi que l'avait bien vu à la même époque Gilles Deleuze, et où, entre autres choses, les individus deviennent peu à peu des entités « dividualles » encodées comme multiplicité de données dans un macro-système d'information. « Ce sont les sociétés de contrôle qui sont en train de remplacer les sociétés disciplinaires. (...) On ne se trouve plus devant le couple masse-individu. Les individus sont devenus des « dividiuels », et les masses, des échantillons, des données, des marchés ou des « banques ». (...) les sociétés de contrôle opèrent par machines de troisième espèce, machines informatiques et ordinateurs (...). Ce n'est pas une évolution technologique sans être plus profondément une mutation du capitalisme. »

Mutation post-industrielle du capitalisme de plus en plus flexible, flottant, immatériel, sémiotique et cognitif, où le « service de vente » devient le centre ou l'âme de « l'entreprise » qui a remplacé « l'usine » de production désormais démantelée, automatisée, externalisée et assez souvent reléguée en périphérie du tiers-monde à l'instar des grandes enseignes multinationales qui se concentrent sur les logiques de Communication et le développement médiatique, si ce n'est psycho-technique, de leur « image de marque ». « On nous apprend que les entreprises ont une âme, ce qui est bien la nouvelle la plus terrifiante du monde. Le marketing est maintenant l'instrument du contrôle social, et forme la race impudente de nos maîtres » affirmera ainsi sans détours Gilles Deleuze. De même, « il n'y a pas besoin de science-fiction pour concevoir un mécanisme de contrôle qui donne à chaque instant la position d'un élément en milieu ouvert, animal dans une réserve, homme dans une entreprise (collier électronique). Félix Guattari imaginait une ville où chacun pouvait quitter son appartement, sa rue, son quartier, grâce à sa carte électronique (dividuelle) qui faisait lever telle ou telle barrière ; mais aussi bien la carte pouvait être recrachée tel

jour, ou entre telles heures ; ce qui compte n'est pas la barrière, mais l'ordinateur qui repère la position de chacun, licite ou illicite, et opère une modulation universelle. »

Vision qui prend d'autant plus d'importance aujourd'hui avec l'informatisation généralisée de la société, l'injonction à la mobilité, l'hyperconnectivité et les projets de dissémination des technologies numériques et autres puces communicantes (informatique ubiquitaire/ubimedia) dans tout l'environnement physique de nos métropoles postmodernes où peut désormais s'opérer de façon massive, par la grâce de l'ingénierie logicielle, la traçabilité, la géolocalisation, le fichage et le profilage des « dividus » dispersés dans les flux et les réseaux, dans et par lesquels se dispensent désormais leur être-au-monde fantomatique sous « le règne de la Technique planétaire »

## Notes

[1] Crédit photo : [Daniel Zanini H.](#) (Creative Commons)

---

**Combien de futurs hackers  
Apple est-il en train de  
tuer ?**



Le succès actuel de l'écosystème Apple et de son dernier bébé l'[iPad](#) n'en finissent plus de nous interpeller.



Après [Cory Doctorow](#), voici le vibrant témoignage du *vieux* développeur [Mark Pilgrim](#) qui, paradoxe, est devenu ce qu'il est grâce aux anciens ordinateurs d'Apple (cf photo ci-contre<sup>[1]</sup> en plein apprentissage).

Ces ordinateurs étaient ouverts et c'est parce qu'on pouvait les [bidouiller](#) que Mark a pu trouver sa vocation et faire de sa passion son métier.

Ce ne serait plus le cas aujourd'hui. Et de se demander alors combien de Mozart de l'informatique est-on actuellement en train de virtuellement assassiner...

L'informatique est une science jeune mais qui commence à avoir ses *anciens combattants* dont certains cèdent à la tentation du « c'était mieux avant ». Le problème c'est qu'ici c'était effectivement mieux avant !

Ce serait déprimant si le logiciel et le [hardware](#) libres n'existaient pas. Mais encore faudrait-il qu'ils rencontrent massivement la jeune génération. Et malheur à nous si le rendez-vous est manqué !

## Le crépuscule du bidouilleur

### [Tinkerer's Sunset](#)

*Mark Pilgrim – 29 janvier 2010 – DiveIntoMark  
(Traduction Framalang : Loque humaine)*

Quand [DVD Jon](#) fut arrêté après avoir cassé [l'algorithme de chiffrement CSS](#), il a été inculpé « d'intrusion d'ordinateur non-autorisée ». Cela mena alors ses avocats à poser la question suivante : « sur quel ordinateur s'est-il introduit ? ». Réponse du procureur : [« le sien »](#) !

Si cette introduction ne vous a pas fait bondir mieux vaut arrêter dès maintenant la lecture de cet article.

Lorsque j'étais plus jeune, « l'intrusion » était quelque chose que vous pouviez uniquement perpétrer sur les ordinateurs des *autres*. Mais mettons ça de côté, nous y reviendrons plus tard.

Mon père était professeur d'université la plus grande partie de sa vie d'adulte. Une année, il prit un congé sabbatique pour écrire un livre. Il avait suffisamment économisé pour s'acheter un ordinateur et une chose super récente appelé logiciel de traitement de texte. Ainsi il écrivit, il édita, et il écrivit encore. C'était évidemment tellement mieux que de travailler sur une machine à écrire qu'il ne s'est jamais posé la question de savoir si c'était de l'argent bien dépensé ou non.

Il se trouve que sur cet ordinateur, le langage de programmation [BASIC](#) était pré-installé. Vous n'aviez même pas besoin de booter le système d'exploitation à partir d'un disque. Vous allumiez l'ordinateur, appuyiez sur *Ctrl-Reset*, et vous aviez une invite de commande. Et sur cette invite de commande, vous pouviez taper un programme tout entier, puis vous tapiez EXECUTE, et, bordel, ça s'exécutait.

J'avais 10 ans. C'était il y a 27 ans, mais je me souviens encore de ce que j'ai ressenti quand j'ai réalisé que vous pouviez – que je pouvais – faire faire *n'importe quoi* à cet ordinateur en tapant les bons mots dans le bon ordre, en lui disant EXECUTE, et que, bordel, ça s'exécutait.

Cet ordinateur était un [Apple IIe](#).

À l'âge de 12 ans, j'écrivais des programmes BASIC si complexes que l'ordinateur n'avait plus assez de mémoire pour les contenir. À 13 ans, j'écrivais des programmes en [Pascal](#). À 14 ans j'écrivais des programmes en [assembleur](#). À 17 ans, je participais à l'épreuve de Programmation de l'[Olympiade Nationale](#) (et la remportais). À 22 ans, j'étais employé comme programmeur.

Aujourd'hui, je suis un programmeur, un rédacteur technique, et un [hacker](#) au sens de [Hackers and Painters](#). Mais vous ne devenez pas hacker en programmant ; vous devenez hacker en *bidouillant*. C'est le bricolage qui donne ce sens de l'émerveillement.

Vous devez bondir hors du système, abattre les barrières de sécurité, enlever une à une les couches posées par l'ordinateur pour faciliter la vie des gens qui ne veulent pas savoir comment ça marche. Il s'agit d'utiliser [l'éditeur de secteur Copy+](#) pour apprendre comment le disque du système d'exploitation démarre, puis de le modifier de manière à ce que l'ordinateur fasse du bruit à chaque fois qu'il lit un secteur sur le disque. Ou alors d'afficher une page de garde au démarrage avant qu'il liste le catalogue du disque et mène à l'invite de commande. Ou de copier une myriade de merveilleuses commandes du tableau [Peeks & Pokes](#) du magazine [Beagle Bros](#). et d'essayer de comprendre ce que je venais de faire. Juste parce que ça me bottait. Juste parce que c'était fun. Parce que ça effrayait mes parents. Parce que je devais absolument savoir comment tout ceci marchait.

Après, il y a eu un Apple IIgs. Et encore après, un Mac IIci. [MacsBug](#). [ResEdit](#). [Norton Disk Editor](#). Arrêtez-moi si ça vous rappelle quelque chose.

Apple a fait les machines qui ont fait qui je suis. Je suis devenu qui je suis en bidouillant.

Le titre de ce billet est tiré de [« On the iPad »](#) d'Alex

Payne, que je vais citer maintenant dans ses grandes largeurs :

*L'iPad est un objet attractif, fort bien pensé et conçu, mais profondément cynique. C'est une machine de consommation digitale. Or, comme [Tim Bray](#) et [Peter Kirn](#) l'ont fait remarquer, c'est un appareil qui ne favorise pas la créativité...*

*Le tragique avec l'iPad est qu'il semble offrir un meilleur modèle d'informatique pour beaucoup de personnes – peut-être la majorité des gens. Envolés les métaphores et concepts déroutants de ces trente dernières années d'informatique. Envolé la possibilité de tripatouiller et modifier sans but particulier. L'iPad est simple, va droit au but, ne demande pas d'entretien...*

*La chose qui me préoccupe le plus avec l'iPad est la suivante : si j'avais eu un iPad plutôt qu'un vrai ordinateur lorsque j'étais petit, je ne serais jamais devenu un programmeur aujourd'hui. Je n'aurais jamais eu la possibilité d'exécuter n'importe quel programme stupide, potentiellement dangereux, mais hautement éducatif que j'aurais pu télécharger ou écrire. Je n'aurais pas été capable de titiller ResEdit et de supprimer le son du démarrage du Mac de façon à ce que je puisse bricoler sur l'ordinateur à toute heure sans réveiller mes parents.*

Maintenant, je suis conscient que vous allez pouvoir développer vos propres programmes pour l'iPad, comme vous pouvez développer pour l'iPhone aujourd'hui. Tout le monde peut développer ! Tout ce dont vous avez besoin, c'est d'un Mac, [XCode](#), un « simulateur » d'iPhone, et de 99 dollars pour un [certificat de développeur à durée limitée](#). Le « certificat de développeur » est en vrai une [clé cryptographique](#) vous permettant (temporairement) d'accéder (partiellement) à... votre propre ordinateur. Et c'est très bien – tout du moins

exploitable – pour les développeurs d'aujourd'hui, parce qu'ils savent qu'ils sont développeurs. Mais les développeurs de demain ne le savent pas encore. Et sans cette [possibilité de bidouiller](#), certains ne le seront jamais.

(À y réfléchir, [j'avais tort](#) et [Fredrik avait raison](#), car il semblerait que les ordinateurs sous [Chrome OS](#) donneront bien la [possibilité aux développeurs d'exécuter leur propre code en local](#). Je ne connais pas les détails de ce à quoi cela va ressembler, si ça sera un bouton, un interrupteur physique ou autre chose. Mais ça sera là, une plateforme officielle prenant en compte les développeurs d'aujourd'hui et, plus important, les développeurs de demain.)

Et, je sais, je sais, vous pouvez « [jailbreaker](#) » votre iPhone, pour (re)gagner l'accès administrateur, et exécuter n'importe quoi qui, bordel, puisse s'exécuter. Et je n'ai aucun doute sur le fait que quelqu'un trouvera comment « jailbreaker » l'iPad aussi. Mais je ne veux pas vivre dans un monde où il faut forcer l'entrée de son propre ordinateur avant de pouvoir bidouiller. Et je ne veux certainement pas vivre dans un monde où [bidouiller son ordinateur est illégal](#). (Au passage, DVD Jon a été acquitté. Le procureur [a fait appel](#) et il a été acquitté à nouveau. Mais qui a besoin de la loi quand vous avez la cryptographie à clé publique de votre côté ?)

Il était une fois des machines, fabriquées par Apple, qui ont fait de moi ce que je suis.

Je suis devenu ce que je suis en bidouillant. Maintenant, il semble qu'ils fassent tout ce qui est en leur pouvoir pour empêcher *mes* enfants de trouver [ce sens de l'émerveillement](#). Apple a déclaré la guerre aux bidouilleurs. À chaque mise à jour de logiciels, la génération « jailbreakée » précédente cesse de fonctionner, et les gens doivent trouver de nouvelles façons pour entrer de force dans leurs propres ordinateurs. Il n'y aura même pas de *MacBug* pour l'iPad. Il n'y aura pas de

*ResEdit*, ou un éditeur de secteur *Copy ][+*, ou un tableau *Peeks & Pokes* pour l'iPad.

Et c'est une vraie perte. Peut-être pas pour vous, mais pour quelqu'un qui ne le sait pas encore et qui pourrait même ne jamais le savoir.

## Notes

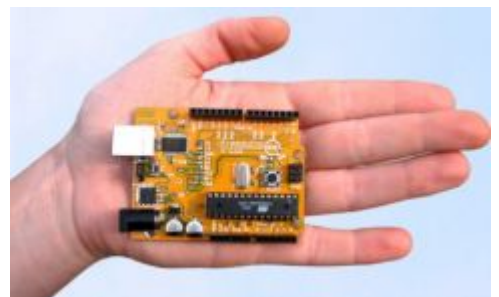
[1] Crédit photo : [Mark Pilgrim](#)

---

**Dis-moi si tu préfères  
bidouiller Arduino ou  
consommer iPad et je te dirai  
qui tu es**

Parmi [la centaine de commentaires](#) provoqués par notre récent article [Pourquoi je n'achèterai pas un iPad](#), on a pu noter une opposition franche entre ceux qui pensaient qu'il était important, voire fondamental, d'avoir la possibilité « d'ouvrir le capot » logiciel et matériel de la bête, et ceux qui n'y voyaient qu'une lubie de geeks passéistes et rétrogrades.

Or aujourd'hui nous allons justement évoquer un drôle d'objet qui accepte d'autant plus volontiers de se *mettre à nu* qu'il sait que c'est sa principale qualité aux yeux de son enthousiaste et créative communauté.



Il s'agit de la carte [Arduino](#) qui est un peu à l'électronique ce que le logiciel libre est à l'informatique, puisque le design, les schémas, les circuits et l'environnement de programmation sont disponibles sous licence libre<sup>[1]</sup>.

Pour vous en dire plus sur cet atypique hardware libre, nous avons choisi de traduire ci-dessous un article de présentation qui fait le parallèle et la liaison avec les hackers ou [bidouilleurs](#) du monde GNU/Linux.

Nous vous suggérons également cette [excellente interview](#) de Alexandra Deschamps-Sonsino, réalisée par Hubert Guillaud pour InternetActu, dont voici quelques larges extraits :

*Arduino est une plateforme de prototypage en électronique. Elle permet aux gens de faire par eux-mêmes, c'est-à-dire de fabriquer des projets interactifs, des objets qui répondent, qui réagissent par exemple à la présence des gens, à leurs mouvements, aux pressions qu'ils y exercent... Arduino relie le monde réel au monde virtuel et vice-versa.*

*Arduino est né en 2005 au sein d'une école de Design en Italie (...). Plusieurs professeurs ressentaient le besoin d'une plateforme technique pour créer des environnements physiques interactifs, utilisables par des gens qui n'avaient pas les compétences techniques pour cela.*

*(...) Arduino permet de faire un lien entre une entrée et une réponse. Il agit comme un cerveau : quand il reçoit telle information, il fait telle chose, selon la manière dont je l'ai équipé ou programmé. Arduino est à la fois du hardware et du software (du matériel et du logiciel). Il se compose d'une carte électronique de quelques centimètres qu'on connecte à un ordinateur à l'aide d'un câble USB. On télécharge un logiciel gratuit sur son ordinateur qui permet de gérer et programmer la puce de la carte Arduino. Une fois programmée, cette puce exécute ce qu'on lui dit. Il n'y a plus qu'à connecter la carte à une batterie et elle fait ce*

*pour quoi elle a été programmée.*

*(...) Au niveau de la communauté, cette plateforme a révolutionné la façon dont les gens pensaient et réfléchissaient à la technologie. Il a permis de ne plus penser la techno de manière abstraite, mais de produire et s'impliquer très rapidement. C'est une plateforme qui coûte peu cher (la carte de base et la puce coûtent une vingtaine d'euros). Toute l'information nécessaire pour accéder au matériel et à son fonctionnement est en ligne, en open source, que ce soit via les forums ou via l'aire de jeux (où la communauté publie codes, plans, tutoriels et astuces). La communauté est désormais forte de quelque 6000 personnes très présentes dans les forums pour accueillir et accompagner les débutants. Il s'est vendu plus de 60 000 cartes Arduino à travers le monde et la distribution est désormais mondiale.*

*(...) L'internet nous a permis de faire plein de choses avec nos vies en ligne... et nous a donné envie de faire la même chose avec les objets de tous les jours.*

*Depuis la révolution industrielle, on a beaucoup créé de dépendances aux produits déjà fabriqués, déjà organisés. Le mouvement DIY (Do It Yourself, Faites-le vous-mêmes) qui se développe depuis 2 ans, réunit une communauté qui ne veut plus accepter des produits tout finis, tout cuits. Cette nouvelle vague de hackers (bidouilleurs) essaye de regarder ce qu'il y a l'intérieur, alors que les conditions d'utilisation n'encouragent pas les gens à regarder ce qu'il y a l'intérieur de ce qu'ils achètent. (...) Le DIY devient un outil pour la microproduction, permettant à chacun de créer son propre business, de fabriquer 20 exemplaires et de voir ce qu'il se passe. Le DIY est finalement important pour sortir du carcan de la mégaproduction. Avant, il fallait un grand marché potentiel pour lancer un produit. Avec l'internet et des plateformes comme Arduino, chacun a accès à sa micro production.*



Arduino s'inscrit donc en plein dans cette approche [DIY \(Do It Yourself\)](#), ou, encore mieux, du DIWO (Do It With Others), que l'on retrouve dans les [Fab lab](#) (lire à ce sujet cet [article](#) de Rue89).

Le professeur que je suis se met à rêver d'une utilisation accrue de ces objets libres dans nos écoles, en particulier en cours de technologie au collège<sup>[2]</sup>.

Plus de curiosité, de créativité, d'esprit critique, d'autonomie, et d'envie d'apprendre, comprendre et entreprendre ensemble, pour moins d'idolâtrie, de passivité et d'individualisme consumériste : une « génération Arduino » plutôt qu'une « génération iPad » en somme...

*PS : Tous les liens de l'article ont été ajoutés par nos soins pour en faciliter la compréhension.*

## Arduino – La révolution matérielle

### [Arduino – the hardware revolution](#)

*Richard Smedley – 23 février 2010 – LinuxUser.co.uk*

*(Traduction Framalang : Yoann, JmpMovAdd, Siltaar et Goofy)*

Chaque année on nous annonce que ce sera « l'année de Linux sur nos écrans d'ordinateur ». Or cette percée tant attendue du logiciel libre chez le grand public tarde à arriver. Mais au moment même où nous guettons des signes d'espoir tels que les ventes de netbooks sous Linux, l'apparition de sites en Drupal ou le développement des téléphones Android (dont une partie est libre), une autre révolution est en marche, dans le monde physique et pourtant pas si éloigné de la sphère d'Internet.

Et voici [Arduino](#) qui fait son entrée : un faible coût, un code source ouvert, une carte matérielle pour le raccordement du monde réel à votre ordinateur, et/ou à tout l'Internet. Que peut-on en faire ? Tout. La seule limite est l'imagination, et

comme vous allez le voir à travers quelques exemples de créations que nous passons en revue ici, l'invention de nouveaux usages est la seule règle.

## Matériel ouvert

Tout comme dans le cas de GNU/Linux, la propagation de ce matériel tient aux raisons suivantes : tout le monde le possède, peut l'améliorer et il donne envie de s'y impliquer. Les plans de référence pour Arduino sont en effet distribués sous licence Creative Commons (le logiciel est quant à lui naturellement sous licence libre en GPL/LGPL), et la société italienne qui est derrière cette plateforme, [Smart Projects](#), accepte avec plaisir les nouveaux collaborateurs et les suggestions alternatives. Les cartes sont réalisées en différents formats, vendues partout dans le monde entier, et si vous souhaitez en fabriquer une vous-même, le Web regorge de modèles différents, quel que soit votre niveau de compétence.

Le nombre de cartes utilisées est estimé à plusieurs centaines de milliers, mais comme dans le cas des distributions Linux, la possibilité de les copier librement rend délicat le décompte précis. Ce qui n'est pas difficile c'est de constater la nature véritablement ouverte des communautés en ligne et l'émergence de nombreuses réunions entre hackers autour des projets Arduino. Ceci a généré un flot continu des projets géniaux menés par toutes sortes de personnes à la fibre créative et artistique. Mais d'abord, un peu d'histoire...

Ceux qui ont de la mémoire et un intérêt pour l'histoire des geeks et du mouvement du logiciel libre se souviennent peut-être du [Tech Model Railroad Club \(TMRC\)](#) – un groupe d'étudiants du MIT créé dans les années 1950 qui s'étaient réunis pour jouer avec les trains électriques. Certains s'intéressaient avant tout aux modèles réduits mais d'autres se passionnaient pour les circuits, l'aiguillage et tout ce qui fait que les trains partent et arrivent à l'heure. C'est

le fameux Signals and Power Subcommittee (*NdT : Sous-comité des signaux et de l'énergie*) qui a mis en œuvre dans les années cinquante et soixante un système de contrôle numérique semi-automatique très brillant, avant d'acquérir un ordinateur [PDP-11](#) en 1970.

Les membres du TMRC ont incarné très tôt la [culture hacker](#), lui donnant son vocabulaire et ses termes de référence. Beaucoup sont devenus des pionniers au sein des premières grandes entreprises d'informatique ([DEC](#), ...). Mais cette culture hacker correspondait bien au stéréotype américain du « [nerd](#) » : le génie sociopathe qui n'arrivait jamais à avoir de petite copine (au TMRC il n'y avait, inévitablement, que des garçons).

Les logiciels libres et la culture hacker ont toujours souffert d'un problème d'image, si bien que la participation féminine dans l'informatique professionnelle a chuté de 50% à 20% pendant les 50 dernières années, certains projets libres ont la proportion dérisoire de 1% de femme. C'est déplorable, les gars, vraiment ! mais il y a des lueurs d'espoir.

## **Au-delà d'Arduino**

Les modules sont basés sur les micro-contrôleurs [Atmel AVR](#) et une conception open source. Il vous est donc facile de faire votre propre Arduino et en fait il existe beaucoup de versions de ce que l'on appelle les [Freeduinos](#) qui ont été créés pour des besoins très différents.

Même le micro-contrôleur Atmel n'est pas indispensable – du moment que l'interface et le langage sont compatibles, on peut bricoler toutes sortes de clones. Il existe aussi des kits pour créer son propre Arduino, vous pouvez même construire votre propre carte si vous êtes à l'aise avec l'électronique embarquée. C'est ce que font finalement certains après des expériences fructueuses avec l'Arduino, bien qu'ils ne soient pas à priori des hackers de systèmes embarqués.

Ainsi la télécommande Arduino pour caméra de Michael Nicholls's, élaborée avec au [Fizzpop](#) hackerspace, est un voyage parmi les oscillateurs et les signaux carrés de contrôle. Chaque projet peut s'avérer aussi amusant qu'éducatif, et en fait, la vie ne devrait-elle pas toujours lier ces deux éléments ? Les télécommandes pour caméra sont un projet populaire, mais ceux qui souhaitent les rendre encore plus petites vont au-delà du projet Arduino, et développent leurs propres cartes mères.

Pour Abdul A Saleh et Aisha Yusuf, le projet Arduino a été une étape puisqu'ils bidouillaient un circuit à brancher sur des radios ordinaires jusqu'à ce qu'ils réalisent qu'un service Web serait plus utile pour leur idée de startup, un moyen de trouver des émissions télé connexes. Leur système peut désormais pointer sur des podcasts au lieu de parcourir les stations de radios, mais « c'est cela qui donne désormais un nouvel élan à de notre projet » indique Yusuf.

En creusant autour de l'univers amical des hackers d'Arduino on trouve plusieurs startups, micro-sociétés et excellentes petites entreprises de constructeurs, vendeurs et formateurs, ainsi que des artistes. Certains, comme [.:oomlout:.](#) entrent dans toutes les catégories à la fois.

Beaucoup sont allés du « suivre la voie du matériel libre », à « poursuivre leur rêves ». Tout comme l'Internet mobile, les ordinateurs portables et les cybercafés ont permis aux créatifs numériques de se lancer en freelance à moindre frais, le bidouilleur de matériel dédié a besoin de son espace de travail partagé à moindre coût, avec si possible plein de collègues créatifs autour. Pour répondre à ce besoin, les [hackerspaces](#) (*NdT : que l'on pourrait éventuellement traduire par « bidouilloires »*) ont finalement vu le jour au Royaume-Uni.

## Hackerspaces

Si le netbook n'a pas complètement fait de 2009 « l'année de Linux dans les ordinateurs grand public », il a vu l'arrivée en retard des hackerspaces sur ses rives, avec des groupes se formant à Birmingham, Brighton, Exeter, Leeds, Liverpool, Londres, Manchester, Shrewsbury, Stoke-on-Trent et York, avec deux groupes distincts coopérant à Manchester. (NdT : le même phénomène s'est produit en France avec au moins cinq hackerspaces rien qu'à Paris – voir [Hackerspace.net](http://Hackerspace.net) et ce [reportage de Rue89](#))

Fabrique le toi-même, ne l'achète pas. L'éthique du hacker sonne bien ces temps-ci, alors que l'intérêt pour les [jardins familiaux](#) va croissant et que les journaux multiplient les dossiers pour nous aider à bâtir des maisons plus écologiques. Ce n'est plus le « fais-le marcher et répare », hérité de nos parents avec le rationnement en temps de guerre, et l'austérité qui a suivi, mais un défi post-société de consommation, pour trouver de la valeur au-delà du « je suis ce que je consomme », par une implication plus profonde dans les choses qui nous entourent. C'est cette implication que l'on retrouve avec les projets Arduino et les réalisations complexes sorties des hackerspaces. Ils témoignent d'une approche vraiment ludique et d'une certaine aisance avec la technologie plutôt que son rejet.

L'Homo sapiens est la seule race définie par les objets dont elle s'entoure, et qui ne peut survivre sans les outils qu'elle fabrique. Des recherches archéologiques ont montré que les néanderthaliens de l'âge de pierre, vivant dans des caves, sans agriculture, et survivant grâce à la chasse et à la cueillette, employaient leurs précieuses heures de temps libre à fabriquer des bijoux et du maquillage.

Il semble que l'envie de jouer, de se parer et de s'amuser soit inhérente à ce que nous sommes. Les hackers et les artistes qui utilisent les modules Arduino pour s'amuser avec

le matériel ne sont ni des fondus de technologie ni des artistes d'avant-garde mais la simple incarnation de l'esprit de notre temps.

## Quelques liens connexes (en vrac)

- [Kit de démarrage Arduino](#) par Hilaire Fernandes d'[Ofset](#) qui conclue ainsi son propos : « reste à imaginer des activités pour des élèves ».
- Vidéo d'une enseignante qui nous présente son cours [Art et Microcontrôleurs](#) à la SAT de Montréal. On a envie d'en être !
- [Arduino, la carte aux trésors ?](#) Un bien joli titre pour ce retour d'expérience (voir aussi [Arduino ou l'électronique créative](#)).
- La [Grotte du Barbu](#) nous présente Arduino. Une vidéo d'une vingtaine de minutes qui ne fait rien pour combattre le cliché du geek mais qui est expliquée très bien en quoi cet objet est intéressant voire fascinant (voir aussi cette [autre vidéo](#)).
- Un petit [clip diaporama](#) sur la bonne ambiance et les réalisations de la [formation Arduino](#) qui a eu lieu à [l'Ecole nationale supérieure de création industrielle \(ENSCI\)](#) en février dernier.
- [Tout ce qui concerne Arduino](#) dans l'excellent magazine [Make](#), chantre du *Do It Yourself*.
- Le livre [Getting Started with Arduino](#) par Massimo Banzi chez O'Reilly. On se rédige un équivalent libre en français chez Framabook ?
- [Taking an Open-Source Approach to Hardware](#), un article du grand Wall Street Journal s'il vous plaît !
- Le site [Hackable Devices](#) ou le supermarché en ligne du matériel libre et ouvert !

*Ne pas hésiter à en ajouter d'autres références dans les commentaires et bien entendu à donner votre avis sur Arduino, son modèle et notre choix discutables de l'opposer ici symboliquement et sociologiquement à l'iPad.*

## Notes

[1] Crédit photo : [Freeduino.org](http://Freeduino.org) (Creative Commons By)

[2] Il est à noter que le groupe toulousain LinuxÉdu (voir ce [billet](#) du Framablog) propose le 5 juin prochain une [découverte d'Arduino](#) parmi les nombreuses autres actions de sa journée de sensibilisation.

---

# Pourquoi je n'achèterai pas un iPad

Hier, vendredi 28 mai, soit deux mois après les États-Unis, Apple a lancé officiellement la commercialisation en France de l'[iPad](#).



L'occasion pour nous de traduire cet article de [Cory Doctorow](#) dont le titre ne souffre d'aucune ambiguïté.

L'ami Cory est l'un de nos plus brillants défenseurs des libertés numériques, et il n'est guère étonnant de le voir ici monter au créneau pour y manifester sa grande perplexité, arguments percutants et convaincants à l'appui.

Avec notre billet [iPad'libertés pour les utilisateurs](#) de la Free Software Foundation, cela nous fait deux bonnes raisons d'expliquer aux adorateurs du Veau d'or<sup>[1]</sup> que sous le vernis clinquant d'une fausse modernité se cache une réalité bien moins reluisante qu'il n'y paraît.

*Remarque : Cette traduction a été entièrement réalisée le samedi 29 mai de 11h à 14h dans le cadre du premier « Traduction », atelier original organisé par l'équipe Framalang et inséré dans l'Ubuntu Party de Paris. [Pour en savoir plus...](#)*

## **Pourquoi je n'achèterai pas un iPad (et pense que vous ne devriez pas non plus)**

**[Why I won't buy an iPad \(and think you shouldn't, either\)](#)**

*Cory Doctorow – 2 avril 2010 – BoingBoing*

*(Traduction Framalang : la quinzaine de personnes présentes au Traduction)*

Voilà dix ans que j'écris des chroniques sur Boing Boing pour y faire découvrir des trucs sympas que d'autres ont créés. La plupart des nouveautés vraiment intéressantes ne sont pas venues de grosses entreprises aux budgets gigantesques, mais d'amateurs qui expérimentent. Des gens qui ont été capables de créer des produits, de les proposer au public et même de les vendre, sans avoir à se soumettre aux diktats d'une seule entreprise qui s'autoproclame gardien de votre téléphone et autres engins high-tech personnels.

Danny O'Brien [explique très bien](#) pourquoi je ne vois aucun intérêt à l'achat d'un iPad – on dirait vraiment le retour de la « révolution » CD-ROM, quand l'industrie du « contenu » proclamait qu'elle allait réinventer les médias, en concevant des produits hors de prix (à fabriquer et à acheter). J'ai commencé ma carrière dans l'informatique en tant que programmeur pour des CD-ROM, et j'ai moi aussi ressenti cet engouement, mais j'ai fini par comprendre que c'était une impasse et que les plateformes ouvertes et les amateurs inventifs finiraient par surpasser les pros roublards et disposant de gros budgets.

Je me rappelle les premiers jours du Web – et les derniers



jours du CD-ROM – quand tout le monde s'accordait à dire que le Web et les PC étaient trop « geek », trop compliqués et trop imprévisibles pour « ma mère » (c'est incroyable le nombre de technophiles qui mettent leur mère plus bas que terre). Si on m'avait donné une action d'AOL à chaque fois qu'on m'a dit que le Web allait mourir parce qu'AOL était simplissime et que le Web était un vrai dépotoir, je serais un gros actionnaire.

Et mes parts ne vaudraient pas grand-chose.

## **Les entreprises dominantes font de piètres révolutionnaires**

Compter sur les entreprises dominantes pour être à l'origine de nos révolutions est une erreur stratégique. Elles ont une fâcheuse tendance à utiliser leurs technologies pour facturer voire interdire tout ce qu'il y a de bien dans leur produit.

Prenez par exemple [l'application Marvel](#) dédiée à l'iPad (jetez juste un coup d'œil, pas plus). Enfant, j'étais fan de comics, et je le suis resté. Ce qui me plaisait par-dessus tout, c'était de les échanger. Il n'existait pas de medium reposant davantage sur les échanges entre gamins pour constituer son public. Et le marché des bédés d'occasion ! C'était – et c'est encore – tout simplement énorme, et essentiel. Combien de fois ai-je farfouillé dans les caisses de bédés d'occasion dans un immense entrepôt poussiéreux pour retrouver des anciens numéros que j'avais ratés, ou de nouveaux titres pour pas cher (dans ma famille, c'est devenu une sorte de tradition qui se perpétue d'une génération à l'autre – le père de ma mère l'emmenait tous les week-ends avec ses frères et sœurs au Dragon Lady Comics sur Queen Street à Toronto pour troquer leurs vieilles bédés contre des nouvelles).

Qu'ont-ils fait chez Marvel pour « améliorer » leurs bandes dessinées ? Ils vous interdisent de donner, vendre ou louer les vôtres. Bravo l'amélioration. Voilà comment ils ont

transformé une expérience de partage exaltante et qui crée du lien, en une activité passive et solitaire, qui isole au lieu de réunir. Bien joué, « Marvsney » (*NdT : Contraction de Marvel et Disney, en référence au récent rachat du premier par le second pour 4 milliards de dollars*).

## **Du matériel infantilisant**

Considérons ensuite l'appareil lui-même : à l'évidence, on s'est creusé la tête pour le concevoir, mais on ressent aussi un grand mépris pour l'utilisateur. Je suis intimement convaincu de la pertinence du [Manifeste du constructeur](#) (*NdT : Maker Manifesto*) : « Si vous ne pouvez pas l'ouvrir, alors ce n'est pas à vous ». Il faut préférer les vis à la colle. Le Apple ][+ d'origine était fourni avec le plan schématique des circuits imprimés, et a donné naissance à une génération de hackers qui bidouillaient leur matériel informatique ou leurs logiciels et ont bousculé le monde dans le bon sens.

Mais, avec l'iPad, il semblerait que [pour Apple](#) le client type soit la maman technophobe et simplette, celle-là même dont on parle si souvent dans l'expression « c'est trop compliqué pour ma mère » (écoutez les pontifes chanter les louanges de l'iPad, ils ne tarderont pas à expliquer qu'on tient enfin quelque chose qui n'est pas trop compliqué pour leur pauvre maman).

La seule interaction que propose l'iPad est celle du simple « consommateur », c'est-à-dire, selon la mémorable définition de William Gibson, « un truc de la taille d'un bébé hippo, couleur patate bouillie vieille d'une semaine, qui vit seul, dans l'obscurité, dans un mobile home, aux alentours de Topeka. Il est recouvert d'yeux, et transpire en permanence. La sueur dégouline et lui pique les yeux. Il n'a pas de bouche... pas d'organes génitaux, et ne peut exprimer ses pulsions rageuses et ses désirs infantiles qu'en changeant de chaîne avec sa télécommande universelle ».

Pour améliorer votre iPad, ne cherchez pas à comprendre comment il fonctionne pour le bricoler, achetez des iApps. Offrir un iPad à vos enfants, ce n'est pas un moyen de leur faire comprendre qu'ils peuvent démonter et réassembler le monde autour d'eux. C'est un moyen de leur dire que même changer les piles c'est une affaire de pros.

Sur ce sujet, il faut absolument lire [l'article de Dale Dougherty](#) sur l'influence d'Hypercard pour toute une génération de jeunes hackers. J'ai effectué mes débuts comme programmeur Hypercard, dont l'invitation douce et intuitive à refaire le monde m'a donné envie d'embrasser une carrière dans l'informatique.

## **Le modèle de la grande distribution s'étend au logiciel**

Intéressons-nous maintenant à l'iStore. Les DRM sont l'alpha et l'oméga d'Apple, alors même que son dirigeant clame partout qu'il les déteste. Apple s'est allié à deux industries (celles du divertissement et des télécoms) qui sont les plus convaincues que vous ne devriez pas être en mesure de modifier vos appareils, d'y installer vos logiciels, d'écrire des applications, et d'outrepasser les instructions envoyées par le vaisseau mère. Apple a construit son activité autour de ces principes. La société utilise des DRM pour contrôler ce que vous pouvez faire sur vos propres appareils, ce qui signifie que les clients d'Apple ne peuvent emmener leur « iContenu » avec eux vers des appareils concurrents, et que ceux qui développent pour Apple ne peuvent vendre à leurs propres conditions.

Le verrouillage de l'iStore ne rend pas meilleure la vie des clients ou des développeurs d'applications . En tant qu'adulte, je veux être capable de choisir ce que j'achète et à qui je fais confiance pour l'évaluer. Je ne veux pas que le Politburo de Cupertino (*NdT : La ville du siège d'Apple*) restreigne mon univers applicatif à ce qu'il choisit

d'autoriser sur sa plateforme. Et en tant que créateur et détenteur de copyright, je ne veux pas d'un unique canal de diffusion contrôlant l'accès à mon public et dictant quel contenu est acceptable. [La dernière fois que j'ai bloqué sur ce sujet](#), Apple s'est répandu en excuses pour le caractère abusif de ses conditions contractuelles, mais la meilleure était : « Pensez-vous vraiment que nous fournirions une plateforme où vous pouvez faire fortune sans aucune contrepartie ? ». J'ai lu cette phrase en imitant la voix de Don Corleone et ça sonnait vraiment bien. Je crois en un marché où la compétition peut prendre place sans que j'aie pour autant à m'agenouiller devant une entreprise qui a érigé un pont-levis entre mes clients et moi.

## **Le journalisme en quête d'une figure paternelle**

Si la presse parle autant de l'iPad, c'est selon moi parce qu'Apple assure le spectacle, et parce dans le monde merveilleux de la presse, chacun cherche une figure paternelle qui lui promettra le retour de son lectorat payant. Toutefois, ce n'est pas seulement parce que les gens peuvent avoir accès gratuitement aux journaux qu'ils ne paient plus. C'est aussi parce que des contenus alternatifs, gratuits et de qualité équivalente, se multiplient. L'ouverture des plateformes a permis une explosion de la quantité de contenus, certains un peu amateurs, d'autres de qualité professionnelle, la plupart mieux ciblés que ne le proposaient les anciens médias. Rupert Murdoch peut menacer tant qu'il le veut de [retirer son contenu de Google](#), je lui dis : Vas-y Rupert, fonce ! Ta fraction de fraction de morceau de pourcentage du Web nous manquera tellement peu qu'on ne le remarquera même pas, et nous n'aurons aucun problème à trouver du contenu pour combler le vide.

La presse techno regorge de gadgets dont les blogueurs spécialisés raffolent (et qui n'intéressent personne d'autre). De même, la presse généraliste est remplie d'articles qui nourrissent le consensus médiatique. Les empires d'hier

pensent faire quelque chose de sacré, vital et surtout *mature*, et ce sont ces *adultes* qui veulent nous extraire de ce bac à sable qu'est le Web, plein de contenus amateurs sans circuits de distribution, afin d'y conclure des accords d'exclusivité. Et nous retournerons alors dans le jardin clôturé qui apporte tant de valeur actionariale à des investisseurs dont le portefeuille n'a pas évolué avec le commerce en ligne.

Mais l'observation attentive du [modèle économique](#) de l'édition sur iPad nous raconte une toute autre histoire : même des ventes astronomiques d'iPad n'arriveront pas vraiment à arrêter l'hémorragie des ventes de l'édition papier. Et ce n'est pas en poussant de grands soupirs et en regrettant le bon temps où tout était verrouillé que les clients reviendront.

## **Les gadgets, ça va ça vient**

Les gadgets, ça va ça vient. L'iPad que vous achetez aujourd'hui va devenir de l'e-pollution dans un an ou deux (moins, si vous décidez de ne pas payer pour qu'on vous change la batterie). Le vrai problème n'est pas dans les fonctionnalités de ce bout de plastique que vous déballez aujourd'hui, mais dans l'infrastructure technique et sociale qui l'accompagne.

Si vous voulez vivre dans un univers créatif où celui qui a une bonne idée peut en faire un programme que vous pourrez installer sur votre appareil, l'iPad n'est pas fait pour vous.

Si vous voulez vivre dans un monde équitable où vous pouvez conserver (ou donner) ce que vous achetez, l'iPad n'est pas fait pour vous.

Si vous voulez écrire du code pour une plateforme où la seule chose qui conditionne votre succès est la satisfactions de vos utilisateurs, l'iPad n'est pas fait pour vous.

## Notes

[1] Crédit photo : [Josh Liba](#) (Creative Commons By)

---

# Socrate et les hackers – Une conférence de Bernard Stiegler

À peine avons-nous mis en ligne le billet [Quand Socrate nous aide à mieux comprendre le logiciel libre](#) qu'un commentaire nous signalait une récente conférence de [Bernard Stiegler](#), dont le titre, séduisant et énigmatique, faisait lui aussi référence au célèbre philosophe grec : « Socrate et les hackers ».

Elle a été donnée le 13 avril dernier à la [Maison des Arts de Malakoff](#).

On y retrouve les thèmes chers à Bernard Stiegler : perte du savoir dans un processus de prolétarianisation généralisée lié au marketing et espoir mis dans la figure de « l'amateur ».

Je ne suis pas un [hacker](#) mais je pense bien faire parti de ces amateurs. Et, du coup, j'aime bien écouter Bernard Stiegler parce qu'il me donne l'impression que je participe à sauver le monde du marasme dans lequel il se trouve ☐

Tout ceci m'a fait repenser à un très lointain article de *Libération* (25 mai 2001). Une [interview](#) de [Pekka Himanen](#)<sup>[1]</sup> par [Florent Latrive](#).

À la question : quel est votre hacker préféré ? Voici ce que le philosophe finlandais avait alors répondu :

« Socrate. Toute son attitude, cette relation passionnée et modeste au savoir, son ouverture d'esprit, sa quête de directions intellectuelles non prévues: l'attitude des Grecs anciens est très similaire à celle des hackers d'aujourd'hui. Platon, son disciple, a fondé la première académie du monde occidental, et c'est le modèle de la recherche scientifique aujourd'hui. C'est aussi celui des hackers passionnés d'ordinateurs... ».

## Bernard Stiegler – Socrate et les hackers



-> La [vidéo](#) au format webm

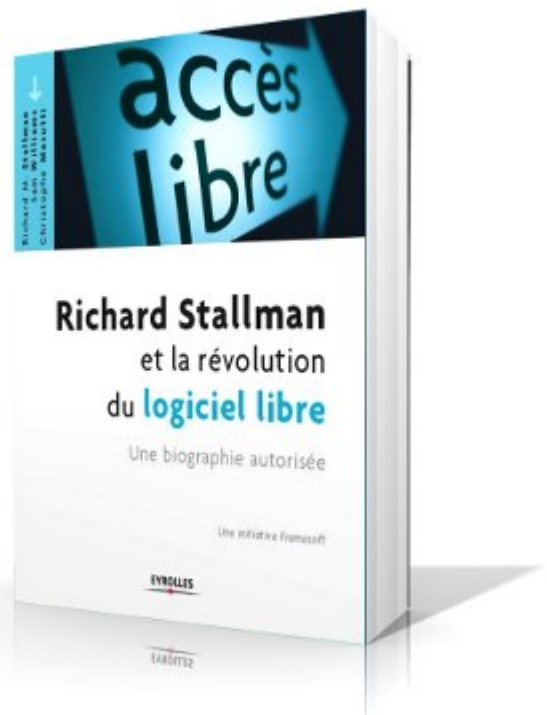
### Notes

[1] À ce propos, c'est vraiment une anomalie de ne plus trouver la moindre trace papier ou numérique de la traduction française de *l'Ethique Hacker* de Pekka Himanen, alors même que c'est clairement l'un des ouvrages les plus importants de la *culture libre*. Il y a une [note de lecture](#) sur Freescape et puis c'est à peu près tout (si ce n'est une indisponibilité récurrente [sur Amazon](#)). L'explication vient du fait qu'Exils, l'éditeur de la traduction n'existe plus, mais c'est bien dommage que personne n'ait pris le relai (et les éventuels droits). Si quelqu'un est assez motivé pour contacter Pekka Himanen et voir avec lui comment on pourrait rééditer son livre en français, qu'il sache que Framasoft et son projet Framabook sont de tout cœur derrière lui.

---

# Il était une fois un livre sur et avec Richard Stallman

Le livre [Richard Stallman et la révolution du logiciel libre – Une biographie autorisée](#), qui est sorti le 21 janvier aux éditions Eyrolles, possède trois auteurs.



Il y a bien entendu Sam Williams, auteur de la version d'origine en anglais. Mais si nous n'étions qu'en face de sa simple traduction, il demeurerait alors le seul auteur de l'ouvrage. Or deux autres noms apparaissent : Richard Stallman lui-même et [Christophe Masutti](#) de Framasoft.

Pourquoi cette originalité et pourquoi méritent-ils tous deux de venir s'associer à Sam Williams ? Vous le saurez en parcourant l'histoire peu commune de ce livre, telle qu'elle a été vécue par Christophe.

Il nous propose une très jolie formule pour expliquer les intentions et les apports de Stallman : il a souhaité « [hacker](#) » le livre.

Avec le même état d'esprit, ou plutôt la même [éthique](#), que lorsqu'il se trouvait, plusieurs dizaines d'années auparavant, jeune programmeur parmi les siens au département de recherche en intelligence artificielle du MIT.



# Il était une fois un livre sur et avec Richard Stallman

*Christophe Masutti – janvier 2010 – GNU Free Documentation License*



Tout a commencé en mars 2007, lorsque Alexis Kauffmann a posté un [message](#) sur le [forum](#) du réseau [Framasoft](#), invitant les volontaires à participer à la traduction du livre de Sam Williams, [Free as in Freedom: Richard Stallman's Crusade for Free Software](#), publié en 2002 chez O'Reilly sous la licence libre [GNU Free Documentation License](#).

Framasoft a une certaine habitude des traductions et s'est même constitué avec le temps une équipe entièrement dédiée à cela, le groupe [Framalang](#). Il se trouve qu'à cette époque Framalang ne chôlait pas et nous ne souhaitons pas leur rajouter une charge de travail supplémentaire avec un livre de quelque 300 pages !

Nous avons donc fait le choix de proposer directement le projet dans un wiki public, et pas n'importe lequel d'entre eux : [Wikisource](#), la bibliothèque libre du réseau [Wikimedia](#). Lors d'une [conférence](#) tenue aux [Rencontres mondiales du logiciel libre 2009](#) de Nantes, Alexis donne plus de détails sur le mode opératoire : quelqu'un avait déjà traduit la préface et le premier chapitre du livre sur un site personnel, ce qui nous a servi de base pour créer la structure de l'ensemble du projet [sur Wikisource](#). L'objectif était bien entendu d'arriver à nos fins mais il s'agissait également d'une expérience, celle d'une ambitieuse traduction collaborative et spontanée ouverte à tous, exactement comme on crée, modifie et améliore les articles de Wikipédia.

Un an plus tard, le bilan était contrasté. Nous avons bénéficié de l'exposition de Wikisource et de nombreuses personnes étaient venues participer. Mais quantitativement il

restait encore beaucoup à faire et qualitativement ce qui avait été fait manquait singulièrement de cohérence et d'harmonie (ne serait-ce que pour se mettre d'accord sur la traduction de tel ou tel terme important et récurrent). Et puis nous butions sur des questions aussi élémentaires que la décision de « clore » un chapitre traduit, ce qui sur un wiki aussi fréquenté ne va pas de soi.

Ne nous en déplaise, il fallait mettre un peu de [« cathédrale dans le bazar »](#), c'est-à-dire un peu de verticalité dans l'horizontalité du projet. Alexis a alors pris la décision de [relancer le projet sur le blog](#) de Framasoft, en invitant les bonnes volontés à se regrouper sur une [liste de discussion](#) dédiée et créée pour l'occasion. Pour ma part, je pris l'initiative d'animer cette liste qui compta rapidement une bonne dizaine d'inscrits (dans le livre vous trouverez en préambule les remerciements à cette liste nominative de participants).

Notre première décision consista à créer ailleurs un deuxième wiki, mais cette fois-ci loin des regards du réseau Wikimedia. Il ne s'agissait pas de jouer les cachottiers, mais nous en étions arrivés à la conclusion qu'il n'était plus possible de continuer à travailler sur Wikisource, dans la mesure où à tout moment une personne externe à la liste pouvait s'en venir tout « bouleverser » (c'est, j'en conviens fort bien, ce qui fait tout le charme de Wikipédia, mais à cette période d'un projet si spécifique nous souhaitions avant toute chose être efficaces et coordonnés dans notre travail).

Nous nous trouvions cependant face à un dilemme : d'un côté la traduction d'origine sur Wikisource restait bien entendu ouverte et continuait sa vie de texte wiki (bien que fort peu active, puisque la liste avait capté une bonne part de l'énergie disponible) et de l'autre côté, le travail sur notre nouveau wiki prenait forme et la traduction avançait plutôt bien. Une fois terminée, notre intention était de reverser la traduction de *Free as in Freedom* dans Wikisource, quitte à

« écraser » les contributions effectuées dans l'intervalle (ces contributions peuvent néanmoins être réhabilitées grâce à l'historique des modifications). Aujourd'hui, on peut donc trouver sur Wikisource cette [traduction française](#) de *Free as in Freedom* publiée par Sam Williams en 2002. Modulo le fait que quelqu'un est peut-être venu en déplacer un mot il y a une heure ☐

Notre traduction avançait donc plutôt bien jusqu'à obtenir une forme convenable à la relecture en novembre 2008, avec en prime la décision définitive d'en faire un volume de plus de la collection de livres libres [Framabook](#).

Une petite parenthèse est ici nécessaire. En effet, nous travaillions depuis peu en étroite collaboration avec l'équipe de [La Poule ou l'Oeuf](#), qui est une chaîne éditoriale en ligne pour la production de livres, pensés comme unités d'une collection, permettant un rendu final de type TeX, PDF, ePub ou HTML. Ce livre était aussi pour nous l'occasion d'implémenter dans le système non seulement l'ouvrage mais aussi la maquette générale de notre collection Framabook. Nous sommes très heureux du résultat car non seulement la Poule ou l'Oeuf a servi pour la mise en page du livre publié chez Eyrolles, mais ce sont désormais tous les framabooks qui vont bénéficier de la puissance de cet outil et des compétences de l'équipe de la Poule ou l'Oeuf.

Parenthèse fermée. Un mois plus tard, en décembre 2008, j'écrivis à Sam Williams pour lui demander une préface. Il accepta, tout en me précisant qu'il aurait bien aimé que Richard Stallman eût participé aux éventuelles modifications de l'ouvrage en anglais mais que finalement cela ne s'était pas produit. À ce moment-là, je ne compris guère l'allusion qui trouva son explication quelques jours plus tard.

Nous réfléchissions également aux illustrations possibles de l'ouvrage. Il y avait une belle [photo](#) de Richard Stallman dans le livre d'origine chez O'Reilly, tirée du [site web personnel](#)

de Richard. Je contacte donc ce dernier, non seulement pour lui demander l'autorisation d'utiliser sa photo, mais aussi pour l'informer que nous comptions publier la traduction en français (une [traduction en italien](#) avait été publiée en 2003).

C'est là que tout a basculé (positivement).

Il faut savoir que le livre *Free as in Freedom* n'a jamais obtenu l'appui formel de Richard Stallman. Pire : Richard aurait déjà affirmé qu'il était hors de question de venir lui demander un autographe sur le livre. On peut légitimement se demander pourquoi... Certes le travail de Sam Williams est d'abord un travail de journaliste, et il dresse un portrait sans concession de la personnalité de Richard Stallman : introverti, controversé, irascible et intransigeant. Tous ceux qui se sont rendus au moins une fois à l'une de ses conférences et qui sont au courant de ses activités, ont une bonne idée de ce que je veux dire.

Mais ce n'est pas sur ce point que Richard Stallman était en désaccord avec l'ouvrage : il y avait des erreurs manifestes voire des interprétations faussées dans les faits concrets relatés dans l'ouvrage. En somme, un travail mené un peu trop vite et sans assez de recul. Par ailleurs, un programmeur de l'envergure de Richard Stallman met un point d'honneur à vouloir reformuler avec exactitude les approximations, même lorsqu'elles ont été commises volontairement dans le but de rendre l'ouvrage plus accessible. Il n'en demeure pas moins que sous le prétexte de l'accessibilité, certaines approximations transformaient carrément le propos ou les événements. La manière dont Richard a agit sur le texte est relatée dans sa préface et lorsque le propos relève notamment de l'interprétation personnelle de Sam Williams, les ajouts de Richard sont clairement identifiés dans le livre par des encarts et des notes de bas de page. Les lecteurs pourront donc se faire une bonne idée des transformations ainsi opérées, quitte à aller voir et comparer avec l'original de

Sam Williams.

Je prends un exemple : lorsque Sam Williams relate la tension entre [Eric S. Raymond](#) et Richard Stallman, on comprend que Raymond accuse Richard d'être la principale cause du manque de réactivité du [projet Hurd](#) (le projet de noyau du [système GNU](#)), et que cette accusation est fondée (on se doute néanmoins que Raymond n'a pas bien digéré la fin de non recevoir pour les modifications de l'éditeur [Emacs](#) qu'il voulait imposer). Pour Williams, et aussi pour Raymond, c'est le « micro-management » à la Stallman, c'est à dire sans concession, qui a freiné Hurd, avec pour conséquence la popularisation du [noyau Linux](#), qui obéit, lui, à un schéma de développement beaucoup plus ouvert. Il serait pourtant simpliste de se contenter de cette interprétation. Richard l'explique autrement, tant en montrant que Hurd n'est pas une mince affaire qu'en montrant aussi que le noyau Linux n'est pas la panacée du point de vue technique comme du point de vue éthique (le plus important à ses yeux).

Bref, suite à mon courriel, Richard me répondit qu'il désirait apporter quelques précisions sur l'épisode [LMI](#) et [Symbolics](#), deux entreprises qui débauchèrent le gros de l'équipe de [hackers](#) du [MIT](#) au début des années 1980. Cet épisode était très important, mais il ne touchait en gros qu'une dizaine de paragraphes dans l'ouvrage. Lorsque j'en fis référence à l'équipe de notre liste de discussion, tout le monde approuva l'idée.

Pourtant, au fil des échanges, Richard me confia qu'il n'avait jamais lu le livre de Sam Williams, et qu'en lisant les chapitres en anglais que je lui envoyais (repris depuis le site personnel de Sam Williams), il ressentait fortement le besoin de le réécrire.

Et tout l'art du hacker se révéla.

Alors que je lui suggérais d'écrire lui-même son autobiographie (d'un certain côté, j'anticipais sur mes

craintes : la retraduction de l'ensemble à partir de toutes ces nouvelles modifications !), il se contenta de me renvoyer des chapitres réécrits, sur lesquels je faisais un [« diff »](#) (une [commande Unix](#) permettant d'afficher les différences entre deux fichiers) pour pouvoir implémenter ces modifications dans la traduction française.

Après tout, qu'est-ce qu'un hacker ? Le lecteur en trouvera une bonne définition historique en annexe de l'ouvrage. L'essentiel est de comprendre que « hacker » signifie surtout améliorer, et qu'un bon hacker qui se respecte ne supporte pas l'imperfection. En toute logique, Richard ressentait tout simplement l'envie irrésistible de « hacker » le livre de Sam Williams. Qui d'autre sinon lui ?

J'énonce tout ceci avec le recul que me permet la parution de l'ouvrage. Dans les faits, je recevais plusieurs courriels par semaine contenant les modifications de Richard. Je les implémentais après les avoir lues, demandé des précisions, et argumenté lorsqu'elles étaient discutables. Bref, un travail soutenu qui nous amena Richard et moi jusqu'au début de l'été 2009.

Un mois auparavant, Alexis avait rencontré [Muriel Shan Sei Fan](#), directrice de la collection *Accès Libre* aux éditions Eyrolles. Et entre la poire et le fromage, il évoqua « l'aventure » de cette traduction qu'il continuait à suivre attentivement. Muriel trouva le projet tant est si bien intéressant qu'elle nous proposa de le publier chez eux.

Nous acceptâmes, mais ce serait vous mentir que d'affirmer que ce fut une décision facile et unanime dans l'équipe. En effet, nous avons, et nous avons encore, nos habitudes chez [InLibroVeritas](#), l'éditeur si particulier et attachant de tous les autres framabooks, avec qui nous travaillons main dans la main depuis des années pour défendre et faire la promotion du logiciel libre et sa culture.

Plusieurs arguments ont cependant pesé dans la balance. Tout d'abord nous n'avions plus affaire cette fois à un livre sur un logiciel libre particulier ([Thunderbird](#), [OpenOffice.org](#), [LaTeX](#), [Ubuntu](#)...). Nous étions face à un livre mutant, une traduction devenue « biographie autorisée » car modifiée et enrichie pour l'occasion par son sujet même. Et pas n'importe quel sujet : la plus illustre des personnalités de la jeune histoire du logiciel libre. Cela méritait assurément de rechercher la plus grande exposition possible. Outre sa capacité de diffusion et distribution, Eyrolles nous offrait aussi son expertise et expérience. Le livre avait été traduit et une première fois relu, mais nous étions néanmoins conscients de sa perfectibilité de par les conditions mêmes de sa réalisation mentionnées plus haut (sachant de plus qu'à ce moment précis de l'histoire Richard n'en avait toujours pas fini avec ses propres modifications). Eyrolles a ainsi retravaillé le texte et mis à disposition du projet plusieurs personnes non seulement pour effectuer de nouvelles relectures mais parfois même pour retraduire certains passages. J'ajoute que nous apprécions la collection pionnière [Accès Libre](#) qui abrite en son sein de nombreux ouvrages de qualité sur le logiciel libre. Et enfin dernier argument et non des moindres : sous notre impulsion, Eyrolles allait pour la première fois publier d'emblée un livre sous licence libre, avec tous les avantages et inconvénients que cela suppose.

Nous nous rencontrâmes in the real life, Muriel, Richard, Alexis et moi, au cours d'un déjeuner en marge des Rencontres mondiales du logiciel libre de Nantes en juillet 2009. Nous discutâmes des modalités de publication et surtout, justement, de la question de la licence. L'ouvrage d'origine étant sous licence GNU Free Documentation License (à cause d'un Stallman insistant, Sam Williams s'en explique à la fin de son livre), sa traduction ne pouvait que l'être également. Or publier sous licence libre n'était pas dans les habitudes d'Eyrolles et cela ne rentrait pas forcément dans les « cases » de ses contrats types (rien que le fait d'interdire la classique

interdiction de photocopier était une nouveauté). De plus nous connaissons les positions sans concession de Stallman dès que l'on touche à ces questions de licence. Mais nous avons néanmoins réussi sans trop de mal à nous mettre d'accord, et il faut rendre ici hommage aux éditions Eyrolles d'avoir su s'adapter et innover.

La dernière ligne droite fut en tout cas aussi passionnante que stressante, avec ses nombreux va-et-vient entre Richard (apportant ses dernières modifications), Eyrolles (éditant à la volée l'ensemble de l'ouvrage), La Poule (obligée à mettre en forme un texte sans cesse en mouvement) et moi (dispersé un peu partout). Toujours est-il que vers la fin décembre 2009, ouf, nous étions prêts et le projet bouclé. Nous méritions tous ce beau cadeau de Noël que nous nous offrions là □

De leur côté, Richard Stallman et [John Sullivan](#) vont très prochainement publier en anglais le livre dans sa nouvelle version, aux [éditions internes](#) à la [Free Software Foundation](#), ajoutant ainsi une dimension supplémentaire au projet. Non seulement nous touchons les lecteurs francophones, mais le monde anglophone pourra aussi se délecter de ce « hack biographique ». Grâce à la licence libre (et aux efforts de quelques uns), le livre, parti des États-Unis, revient à la maison après un détour par la France qui l'aura transformé !

Pour moi, ce livre n'est pas seulement une biographie, même s'il en a l'apparence et la saveur. Il s'agit d'une histoire, celle du mouvement pour le logiciel libre, qui a influencé profondément l'histoire générale de l'informatique depuis la fin des années 1960. On considère généralement cette histoire à travers celle de l'industrie logicielle ou des composants d'ordinateurs. Mais il manque souvent une approche en termes de pratiques d'ingénierie et de circulation de l'information. Le logiciel libre constitue en cela une belle illustration de l'ingénierie logicielle, qui avance non seulement par projet, mais aussi parce qu'elle est fondamentalement un débat permanent sur la disponibilité et le partage de l'information.



Lorsque le partage d'idées est impossible (notamment à cause des brevets) et lorsque les développeurs et les utilisateurs sont restreints dans leurs libertés, alors c'est la société toute entière qui pâti de la pénurie de code et de libertés.

Tous les métiers ont leur déontologie. Les informaticiens ont une éthique et ceux qui la distillent sont les hackers. Par delà ses actes et ses programmes, l'un des principaux mérites de Richard Stallman est d'avoir réussi à concentrer et matérialiser cette éthique dans une simple licence (la fameuse [GNU General Public License](#)), qui va non seulement fonder, défendre et diffuser le logiciel libre mais qui constitue aujourd'hui une référence et une source d'inspiration pour d'autres mouvements émancipateurs. En fait, les programmes ont toujours été libres, et c'est un non-sens éthique qui les a rendu privateurs à un moment donné. L'histoire de l'informatique est heureusement loin d'être terminée.

Celle de ce livre touche par contre à sa fin, puisqu'il sera officiellement publié le 21 janvier prochain sous le titre *Richard Stallman et la révolution du logiciel libre – Une biographie autorisée*. Je remercie chaleureusement tous ceux que cette aventure a mis sur notre chemin. Toutes ces rencontres qui font aussi le sel d'un tel projet. À Framasoft, nous sommes fiers d'avoir pu le mener à son terme. Et malgré le labeur qui n'a pas manqué, ce fut un réel plaisir. Plaisir que nous espérons désormais partager avec le lecteur...

Cette histoire touche donc à sa fin, disais-je, mais, licence libre oblige, rien ne dit qu'il ne subisse pas à l'avenir d'autres métamorphoses. Ne serait-ce que parce que Richard est heureusement toujours parmi nous et qu'il lui reste encore certainement de belles pages à écrire dans le livre de sa vie.

---

# Souhaitons-nous une société d'illettrés numériques ou une société libre ?

On peut s'extasier béatement devant les prétendues capacités technologiques de la nouvelle génération, née une souris dans la main, et baptisée un peu vite les [« digital natives »](#).



Mais s'il ne s'agit que de savoir manier de nouveaux objets, sans conscience, sans recul, et sans compétence ni curiosité pour en *soulever les capots*, alors nous nous mettons peut-être en danger<sup>[1]</sup>.

Or, parmi ces nouveaux objets, il y a les logiciels, dont tout le monde aura noté la place croissante qu'ils occupent dans nos sociétés contemporaines. Nous écarterons d'autant plus facilement ce danger que nous serons toujours plus nombreux à accorder de la valeur à la liberté des logiciels.

C'est la thèse que défend ici [Hugo Roy](#) en évoquant, par analogie avec la démocratie, une approche [systémique](#) de la situation.

*PS : Pour l'anecdote, il s'agit d'une traduction que nous avons entamée sans savoir qu'Hugo Roy était... français ! Du coup c'est la première fois qu'on se retrouve avec une traduction relue par l'auteur même de l'article d'origine !*

## Logiciel Libre, Société Libre : À propos

# de la Démocratie et du Hacking

## [Free Software, Free Society: Of Democracy and Hacking](#)

*Hugo Roy – 8 novembre 2009 – Blog de la FSFE*

*(Traduction Framalang : Gilles Coulais et Hugo Roy)*

Lorsqu'on explique pourquoi le logiciel libre est important, une question revient souvent :

**« Ai-je réellement besoin de la liberté du logiciel ? »**

L'utilité de la liberté du logiciel n'est pas évidente pour tous. Tout le monde n'est pas capable de comprendre le code source d'un programme, et ils sont encore moins nombreux à pouvoir le modifier. Seuls les hackers et les développeurs peuvent en effet jouir pleinement des [quatre libertés](#) d'un logiciel libre. Il est alors difficile de convaincre quelqu'un d'abandonner le logiciel propriétaire pour le simple bénéfice de la liberté, tant qu'il ne comprend pas **l'utilité de cette liberté**.

Il est essentiel de penser ce problème non pas comme un simple engagement envers la liberté, mais plus comme un problème de systèmes.

Tout d'abord, ne pas jouir d'une liberté n'implique pas pour autant qu'on ne bénéficie pas des effets de cette liberté. L'analogie la plus évidente ici sont les systèmes politiques. La Constitution est à la souveraineté ce que la licence des logiciels libres est au droit d'auteur. La Constitution qui définit notre système politique donne à chaque citoyen des libertés et des droits, tel que le droit de se porter candidat à une élection.

Tout le monde peut se présenter à une élection, ce qui ne signifie pas pour autant que tout le monde le fera. Tout le monde n'a pas la compétence ou l'envie de devenir politicien. Cela étant, diriez-vous que la démocratie n'a aucune importance juste parce que vous ne souhaitez pas

personnellement entrer en politique ? Je crois que la plupart des gens ne diraient pas cela.

C'est la même chose avec le logiciel libre. Chacun peut utiliser, partager, étudier ou améliorer le programme. Mais le fait que vous ne le ferez pas ne doit pas vous amener à penser que ce n'est pas important pour vous. C'est important pour le système lui-même. Et plus le système devient important, plus cette liberté prend de la valeur.

À moins, bien sûr, que vous ne partiez du principe que le logiciel n'est pas important, et par conséquent son degré de liberté également. Mais alors, je suggère que vous éteigniez votre ordinateur et que vous arrêtiez de me lire. Prenez un avion et passez le reste de votre vie sur une île déserte.

Regardons maintenant de plus près l'utilité de la liberté logicielle. Alors que de plus en plus de logiciels sont utilisés dans notre société pour faire toujours davantage, nous devrions être de plus en plus nombreux à être capables de comprendre le logiciel. Sauf à vouloir donner à certains un contrôle total sur vous-même. Et alors les autres façonneront le système à votre place, en vue d'obtenir toujours plus de pouvoir au sein de ce système.

C'est la raison pour laquelle nous avons besoin de démocratiser le [hacking](#). Et cette démocratisation viendra naturellement si le logiciel libre est largement utilisé. Donnez aux gens la possibilité d'étudier et d'explorer quelque chose, et ils finiront par le faire, au moins par curiosité, de la même manière que l'Imprimerie a donné aux gens la volonté d'être capable de lire puis d'écrire. Il s'agit évidemment d'un long processus. Mais ce processus peut s'avérer beaucoup plus long si nous utilisons du logiciel propriétaire, un logiciel que vous ne pouvez ni lire, ni modifier, ni partager.

Souhaitons-nous une société d'illettrés numériques ou une

société libre ?

## Notes

[1] Crédit photo : [Glitter Feet](#) (Creative Commons By)

---

# Unix a 40 ans : passé, présent et futur d'un OS révolutionnaire

Séquence Histoire, à l'occasion du quarantième anniversaire de la naissance d'[Unix](#), le mythique système d'exploitation [multitâche](#) et [multi-utilisateur](#), créé en août 1969, dont l'héritage est impressionnant.



Quarante ans, dans la jeune histoire de l'informatique, c'est une véritable épopée. Et le fait qu'il soit toujours évoqué aujourd'hui en dit long sur ses qualités d'origine. On en parle parce que, bien qu'en bout de course, on continue à l'utiliser, mais aussi parce qu'il a engendré et inspiré à sa suite toute une famille de systèmes d'exploitation dont rien moins que BSD, Mac OS X et enfin, évidemment, GNU/Linux.

Dans quelles conditions et contexte Unix est-il né ? Pourquoi un tel succès, une telle longévité et une telle parenté ? Et qu'en est-il de sa situation aujourd'hui ? Autant de questions qui trouveront réponses dans l'article ci-dessous traduit par nos soins<sup>[1]</sup>.

# Unix a 40 ans : Le passé, présent et futur d'un OS révolutionnaire

## [Unix turns 40: The past, present and future of a revolutionary OS](#)

Gary Anthes – 4 juin – *ComputerWorld.com*

(Traduction Framalang : Vincent, Daria et Tyah)

**Après quatre décades, le futur du système d'exploitation s'assombrit, mais son héritage va perdurer.**

Cela fera quarante ans cet été, un programmeur s'asseyait et jetait les bases en un mois ce qui allait devenir l'un des plus importants morceaux de logiciel jamais créé.

En août 1969, [Ken Thompson](#), un programmeur des [Laboratoires Bell](#) (NdT : *Bell Labs*), une filiale d'[AT&T](#), considérait le voyage d'un mois de sa femme et de son fils comme une opportunité pour mettre en pratique ses idées pour un nouveau système d'exploitation. Il écrivait la première version de Unix dans un langage assembleur pour un mini-ordinateur un peu mollasson, un [Digital Equipment Copr. \(DEC\) PDP-7](#), en passant une semaine sur chaque partie : un [système d'exploitation](#), un [shell](#), un [éditeur](#) et un [assembleur](#).



Thompson et un collègue, [Dennis Ritchie](#) (cf photo ci-contre), se trouvaient démotivés depuis que les Bell Labs s'étaient retirés un peu plus tôt dans l'année d'un projet trouble de développement d'un système temps partagé appelé [Multics](#)

(Multiplexed Information and Computing Service). Ils n'avaient aucune envie de rester sur des systèmes d'exploitation de type [batch](#) qui prédominaient à cette époque, et ne voulaient pas non plus réinventer Multics, qu'ils considéraient comme grotesque et peu maniable.

Après avoir débattu autour de plusieurs idées pour un nouveau système, Thompson a écrit la première version de Unix, que le binôme a continué de développer durant les années suivantes avec l'aide de leurs collègues [Doug McIlroy](#), [Joe Ossanna](#) et Rudd Canaday. Quelques-uns des principes de Multics ont été réutilisés dans leur nouveau système d'exploitation, mais la beauté d'Unix alors (et encore maintenant) reposait sur sa philosophie « moins-c'est-plus » (*NdT : less-is-more philosophy*).

« Un OS puissant pour un usage interactif n'a pas besoin d'être coûteux, que ce soit en matériel ou en efforts humains », devaient écrire Ritchie et Thompson cinq ans plus tard dans les communications de l'ACM (CACM), le journal de l'[Association for Computer Machinery](#). « Nous espérons que les utilisateurs de Unix trouveront que les plus importantes caractéristiques de ce système sont sa simplicité, son élégance et sa facilité d'utilisation ».

Apparemment ils y sont arrivés. Unix va évoluer pour devenir une pierre angulaire de l'informatique, largement installé sur les serveurs et les postes de travail dans les universités, les gouvernements et les entreprises. Et son influence s'est étendue bien au-delà de ses déploiements, comme l'a noté l'ACM en 1983, lorsqu'ils ont attribué leur prix le plus élevé à Thompson et Ritchie, le A.M. [Turing Award](#) pour leur contribution à l'informatique : « Le modèle du système Unix a conduit une génération de concepteurs logiciels à repenser leur manière de programmer ».

## **Les premiers pas**

Bien sûr, le succès d'Unix n'est pas arrivé d'un coup. En 1971, il a été porté sur le mini-ordinateur [PDP-11](#), une plateforme plus puissante que le PDP-7 pour laquelle il avait été écrit à l'origine. Des programmes d'édition et de formatage de texte ont été ajoutés, et il a été déployé pour quelques dactylos du département des Brevets des Bell Labs,

ses premiers utilisateurs en dehors de l'équipe de développement.

En 1972, Ritchie écrivait le langage de programmation de haut-niveau C (basé sur un langage précédent de Thompson, le B) ; suite à cela, Thompson réécrivit Unix en C, ce qui permit d'améliorer la portabilité de l'OS sur différents environnements. En cours de chemin, il acquit le nom de Unics (Uniplexed Information and Computing Services), un jeu de mots sur Multics ; l'orthographe se transforma rapidement en Unix.

Il était temps d'envahir le monde. L'article de Ritchie et Thompson de Juillet 1974 dans le CACM, [« Le système temps-partagé UNIX »](#), prit le monde de l'informatique dans un ouragan. Jusqu'alors, Unix avait été confiné à une poignée d'utilisateurs dans les Laboratoires Bell. Mais maintenant, avec l'Association for Computing Machinery derrière lui, un éditeur l'ayant qualifié comme étant *élégant*, Unix était à un moment crucial.

« L'article du CACM avait eu un impact très fort », raconte l'historien de l'informatique [Peter Salus](#) dans son livre *The Daemon, the Gnu and the Penguin (NdT : Le démon, le Gnou et le manchot)*. « Peu de temps après, Ken était assailli de demandes concernant Unix ».

## Un paradis pour les hackers

Thompson et Ritchie étaient des [hackers](#) accomplis, quand ce terme désignait quelqu'un qui combinait une créativité hors pair, une intelligence à toute épreuve et de l'huile de coude pour résoudre des problèmes logiciels que l'on a peine à imaginer.

Leur approche, et le code qu'ils avaient écrit, étaient attractifs pour les programmeurs des universités, et plus tard pour des entreprises qui démarraient sans les méga budgets de IBM, Hewlett-Packard ou Microsoft. Unix était tout ce dont les autres hackers, comme [Bill Joy](#) à l'Université de Californie,



[Rick Rashid](#) à l'Université Carnegie Mellon et [David Korn](#) plus tard aux Laboratoires Bell avaient rêvé.

« Presque depuis le début, le système était capable de se maintenir seul, et il l'avait prouvé », écrivaient Thompson et Ritchie dans l'article du CACM. « Comme tous les programmes sources étaient toujours disponibles et facilement modifiables en ligne, nous étions volontaires pour réviser et modifier le système et ses logiciels lorsque de nouvelles idées étaient inventées, découvertes ou suggérées par d'autres ».

Korn, maintenant membre d'AT&T, travaillait comme programmeur aux Bell Labs dans les années 1970. « L'une des caractéristiques de Unix était que des outils pouvaient être écrits, et de meilleurs outils pouvaient les remplacer », se souvient-il. « Ce n'était pas un monolithe dans lequel il fallait tout approuver ; vous pouviez en fait développer de meilleures versions ». Il a développé l'influent [Korn shell](#), un langage de programmation permettant d'interagir directement avec Unix, maintenant disponible en open source.

Salus, auteur et historien des technologies, se souvient de son travail avec le [langage de programmation APL](#) sur un [mainframe IBM](#) tournant sous System/360, en tant que professeur à l'Université de Toronto dans les années 1970. Mais le lendemain de Noël 1978, un ami à l'Université de Columbia lui fit une démonstration de Unix, tournant sur un mini-ordinateur. « Je me suis exclamé : Oh mon Dieu !, et j'étais totalement converti », raconte Salus.

Il dit que pour lui, l'un des avantages clés de Unix était la fonctionnalité *pipe*, introduite en 1973, qui rendait facile le passage de données d'un programme à l'autre. Le concept de [pipeline](#), inventé par McIlroy des Bell Labs, fut par la suite copié dans de nombreux OS, parmi lesquels les variantes d'Unix, Linux, [DOS](#) et Windows.

Un autre avantage de Unix, le deuxième effet *Woow*, comme le

dit Salus, était qu'il n'avait pas besoin d'un mainframe d'un demi-million de dollars pour tourner. Il avait été écrit sur le petit mini-ordinateur DEC PDP-7, assez primitif, car c'était tout ce que Thompson et Ritchie avaient sous la main en 1969. « Le PDP-7 était presque incapable de quoi que ce soit », se souvient Salus. « J'étais scotché ».

Pas mal de monde a également été rendu accro. Les chercheurs des universités ont adopté Unix car il était assez simple et facilement modifiable, ne demandait pas beaucoup de ressources, et le code source était libre pour une bonne partie. Des startups comme Sun Microsystems et un ensemble d'entreprises maintenant défuntes qui se spécialisaient dans l'informatique scientifique, comme [Multiflow Computer](#), en ont fait leur système d'exploitation de choix pour les mêmes raisons.

## La progéniture d'Unix

Unix a grandi comme un système non-propriétaire car en 1956 AT&T avait été enjoint par un décret fédéral à ne pas s'écarter de sa mission de production de service téléphonique. Il pouvait développer du logiciel, et même le licencier pour un tarif *raisonnable*, mais l'entreprise ne pouvait pas entrer sur le marché de l'informatique.

Unix, qui avait été développé sans encouragement du management, avait d'abord été vu par AT&T comme quelque chose entre une curiosité et un casse-tête légal.

C'est à ce moment là, à la fin des années 1970, AT&T a réalisé qu'ils avaient quelque chose de commercialement intéressant en main. Ses avocats ont commencé à adopter une interprétation plus favorable de l'ordonnance de 1956, comme ils cherchaient des manières de protéger Unix comme un secret de fabrication. A partir de 1979, avec la sortie de la Version 7, les licences Unix interdisaient aux universités d'utiliser le code source Unix pour l'étudier pendant leurs cours.

Pas de problème, se dit alors le professeur d'informatique [Andrew Tanenbaum](#), qui avait utilisé Unix V6 à l'Université Vrije à Amsterdam. En 1987, il écrivit un clone de Unix pour utiliser lors de ses cours, en créant le système d'exploitation open source [Minix](#), qui tournait sur un microprocesseur Intel 80286.

« Minix comprenait toutes les idées d'Unix, et c'était un travail brillant », raconte Salus. « Seul un programmeur majeur, quelqu'un qui comprenait en profondeur les mécanismes internes d'un système d'exploitation, pouvait faire cela ». Minix allait devenir le point de départ pour la création de Linux par [Linus Torvalds](#) en 1991. Si ce n'était pas exactement un clone d'Unix, il s'agissait à n'en pas douter d'un *Unix-like*.

En reculant d'une décade ou à peu près, Bill Joy, qui était un étudiant diplômé et programmeur à l'Université de Californie de Berkeley dans les années 1970, mit la main sur une copie de l'Unix des Laboratoires Bell, et il pensa que c'était une bonne plateforme pour son propre travail sur un compilateur Pascal et un éditeur de texte.

Les modifications et extensions que lui et d'autres firent à Berkeley ont résulté dans la seconde branche majeure de Unix, appelée [Berkeley Software Distribution](#) (BSD) Unix. En mars 1978, Joy envoyait des copies de 1BSD, vendues 50\$.

Ainsi en 1980 il y avait deux lignes majeures de Unix, l'une de Berkeley et l'autre de AT&T, et la scène était montée pour ce qui allait devenir la Guerre des Unix. La bonne nouvelle était que les développeurs de logiciels pouvaient avoir accès de partout au code source Unix, et l'adapter à leurs besoins et caprices. La mauvaise nouvelle c'est qu'ils firent exactement cela. Unix a proliféré, et les variantes ont divergé.

En 1982, Joy a contribué à fonder Sun Microsystems et offrait

une station de travail, la [Sun-1](#), qui utilisait une version de BSD appelée [SunOS](#). (Solaris allait apparaître environ 10 ans plus tard). L'année suivante, AT&T a sorti la première version de [Unix System V](#), un système d'exploitation qui a eu une influence énorme et qui fut la base de [IBM AIX](#) et Hewlett-Packard [HP-UX](#).

## Les Guerres Unix

Au milieu des années 80, les utilisateurs, y compris le gouvernement fédéral, se sont plaints que si en théorie Unix était un système d'exploitation unique et portable, en fait on en était loin. Les éditeurs faisaient semblant d'adhérer à cette réclamation, mais travaillaient jour et nuit pour enfermer les clients avec des fonctionnalités et [API](#) uniques à leur Unix.



En 1987, Unix System Laboratories, qui faisait partie des Laboratoires Bell à cette époque, commença à travailler avec Sun sur un système qui pourrait unifier les deux branches majeures de Unix. Le produit de leur collaboration, appelé Unix System V Release 4.0, fut sorti deux ans plus tard, et combinait des fonctionnalités de System V Release 3, BSD, SunOS et du [Xenix](#) de Microsoft.

Les autres éditeurs Unix craignaient l'alliance AT&T/Sun. Les diverses parties ont formé des organismes *standards* concurrents, qui portaient des noms tels que [X/Open](#), [Open Software Foundation](#) (ou OSF), Unix International et Corporation for Open Systems. Les arguments, contre-arguments et réussites de ces groupes pourraient remplir un livre, mais ils clamaient tous leur désir d'un Unix unifié, tout en se tirant dessus les uns les autres.

Dans un article écrit en 1988, mais non publié, pour la [DARPA](#)

(Defense Advanced Research Projects Agency) (*NdT : Agence pour les Projets de Recherche Avancée de la Défense*), le pionnier des mini-ordinateurs [Gordon Bell](#) disait ceci de la toute nouvelle Open Software Foundation, qui incluait IBM, HP, DEC et d'autres, alliés contre le partenariat AT&T/Sun : « OSF est une manière pour ceux qui n'ont pas d'Unix d'entrer sur ce marché en devenir, tout en maintenant leur musées de code à hautes marges ».

Les [Guerres Unix](#) n'ont pas réussi à calmer les différences ou à créer un réel standard pour le système d'exploitation. Mais en 1993, la communauté Unix était réveillée par Microsoft, sous la forme de [Windows NT](#), un système d'exploitation 32 bits utilisable en entreprise. Ce NT propriétaire visait directement Unix, et devait étendre l'hégémonie de Microsoft sur l'ordinateur de bureau aux centres serveurs et endroits similaires, tenus principalement par les serveurs Sun et autres.

Les utilisateurs Microsoft ont applaudi. Les fournisseurs Unix ont paniqué. Tous les principaux rivaux des Unix majeurs se sont réunis dans une initiative appelé [Common Open Software Environnement](#) (*NdT : Environnement Logiciel Ouvert Commun*), et l'année suivante ont plus ou moins déposé les armes en mixant le groupe [Unix International](#), appuyé par AT&T/Sun avec l'Open Software Foundation. Cette coalition a évolué dans l'actuel [The Open Group](#), organisme certificateur des systèmes Unix, et propriétaire de [Single Unix Specification](#) (SUS), maintenant la définition officielle de « Unix ».

D'un point de vue pratique, ces développements ont pu *standardiser* Unix au maximum possible, en fonction de la concurrence des éditeurs. Mais ils sont sans doute arrivés trop tard pour enrayer la montée d'une marée appelée Linux, le système d'exploitation open source dérivé de Minix du professeur Tanenbaum.

## Mais qu'est ce que « Unix », finalement ?

Unix, vous diront la plupart des gens, est un système d'exploitation écrit il y a plusieurs dizaines d'années aux Laboratoires Bell de AT&T, et ses descendants. Les versions majeures d'Unix aujourd'hui proviennent de deux branches d'un même tronc : l'une émane directement de AT&T et l'autre vient de AT&T via l'Université de Californie de Berkeley. Les branches les plus robustes actuellement sont AIX d'IBM, HP-UX de HP et Solaris de Sun.

Toutefois, The Open Group, qui détient la marque déposée Unix, [définit Unix](#) comme n'importe quel système d'exploitation qui a été certifié par ses services pour se conformer à la Single Unix Specification (SUS) (*NdT : Spécification Unix Unique*). Ceci inclut des OS que l'on ne considère pas généralement comme étant des Unix, tel que [Mac OS X Leopard](#) (qui dérive d'un BSD Unix) et IBM z/OS (qui vient du système d'exploitation mainframe MVS), car ils se conforment au SUS et supportent les APIs SUS. L'idée basique est que c'est un Unix si cela agit comme Unix, quel que soit le code sous-jacent.

Une définition encore plus large d'Unix pourrait inclure les OS [Unix Like](#) (qui ressemblent à Unix), parfois appelé des *clones* d'Unix ou des *sosies*, qui ont copié une bonne partie des idées de Unix mais n'incorporent pas directement du code Unix. Le plus connu d'entre tous étant certainement Linux.

Finalement, même s'il est raisonnable d'appeler Unix un *système d'exploitation*, c'est finalement plus que cela. En plus du noyau de l'OS, les implémentations Unix embarquent typiquement des utilitaires tels que des éditeurs en ligne de commande, des interfaces de programmation applicatives (API), des environnements de développement, des [bibliothèques](#) et de la documentation.

## Le futur d'Unix

Le manque continu de portabilité complète entre les

différentes versions concurrentes d'Unix, de même que l'avantage financier de Linux et Windows sur les processeurs standards [x86](#), vont inciter les organisations informatiques de quitter Unix, suggère un récent [sondage](#) du [Gartner Group](#).

« Les résultats réaffirment l'enthousiasme continu pour Linux comme plate-forme serveur, avec Windows qui croît de la même manière, et Unix qui prend la route d'un long, mais graduel, déclin », décrit le sondage, publié en février 2009.

« Unix a eu un passé long et vivant, et même s'il n'est pas encore parti, il sera de plus en plus sous pression », dit l'analyste du Gartner George Weiss. « Linux est l'Unix de choix stratégique. Même si Linux n'a pas le long héritage de développement, de [tuning](#) et de [tests de stress](#) que Unix a pu avoir, il s'en approche et va bientôt égaler Unix en performance, stabilité et montée en charge », dit-il.

Mais un récent [sondage](#) de ComputerWorld suggère que le retrait d'Unix ne va pas arriver aussi rapidement. Dans cette enquête concernant 130 utilisateurs Unix sur 211 responsables informatiques, 90% ont déclaré que leur entreprise était « très ou extrêmement dépendante » de Unix. Un peu plus de la moitié a dit : « Unix est une plateforme essentielle pour nous et le restera indéfiniment », et seulement 12% ont dit « Nous pensons sortir de Unix dans le futur ». Les économies de coût, essentiellement au travers d'une consolidation de serveurs, étaient citées comme la raison n°1 pour migrer vers autre chose.

Weiss pense que la migration vers des processeurs x86 va s'accélérer à cause de l'avantage du coût du matériel. « Des architectures extensibles horizontalement ; le clustering ; le cloud computing ; la virtualisation sur x86, quand vous combinez toutes ces tendances, le système d'exploitation de choix tourne autour de Linux ou Windows », dit-il.

« Par exemple », déclare Weill, « dans l'annonce récente de

[Cisco](#) de son architecture [Unified Computing](#), vous avez dans un même équipement le réseau, le stockage, le calcul et la mémoire, et vous n'avez pas besoin d'Unix. Vous pouvez utiliser Linux ou Windows sur x86. Ainsi, Intel est en train de gagner la guerre pour le compte de Linux contre Unix ».

The Open Group, propriétaire des Single Unix Specification et organisme certificateur des systèmes Unix, concède peu à Linux, et déclare que Unix est le système de choix pour « le haut du panier en termes de fonctionnalités, montée en charge et performance pour des applications critiques ». Linux, dit-il, tend à être le standard pour les applications plus simples, moins critiques.

Korn, de AT&T, est parmi ceux qui pensent que Unix a encore un bel avenir. Korn dit que la force d'Unix pendant toutes ces années, depuis l'ajout des pipes en 1973, est qu'il peut facilement être éclaté en petits bouts et distribué. Ceci va pousser Unix en avant, dit-il. « La philosophie (du pipeline) fonctionne bien avec le cloud computing, dans lequel vous construisez des petits morceaux réutilisables plutôt qu'une grosse application monolithique ».

## **L'héritage d'Unix**

Quel que soit le sort final d'Unix, le système d'exploitation né dans les Laboratoires Bell il y a 40 ans a établi un héritage qui risque de durer encore quelques dizaines d'années. Il peut revendiquer la paternité d'une longue liste de logiciels populaires, incluant l'offre Unix de IBM, HP, Sun, Apple Mac OS X et Linux. Il a également influencé des systèmes qui ont leurs racines dans Unix, comme Microsoft Windows NT et les versions IBM et Microsoft de DOS.

Unix a permis à de nombreuses entreprises qui démarraient de réussir, en leur offrant une plateforme peu onéreuse sur laquelle construire. C'était une des briques de base pour la construction d'Internet, et il fait partie du cœur des



systemes de telecommunications aujourd'hui. Il a engendré de nombreux concepts d'architecture, comme l'utilisation du pipeline, et le dérivé de Unix, [Mach](#), a contribué énormément à l'informatique scientifique, distribuée et multiprocesseurs.

C'est peut-être l'ACM qui l'a mieux énoncé en 1983 dans sa citation du prix Turing en l'honneur de Thompson et Ritchie sur leur travail sur Unix : « Le génie du système Unix est dans sa structure, qui permet aux programmeurs de bâtir sur le travail des autres ».

## Notes

[1] Crédit photos : 1. [Rudolf Schuba](#) (Creative Commons By) / 2. [Hoshie](#) (Domaine Public) / 3. [Eraserhead1 et Infinity0](#) (Creative Commons By-Sa)

---

# Stallman – How a hacker became a freedom fighter

Une courte interview de [Richard Stallman](#) traduite par Olivier et relu par Gaelix (Framalang).

L'occasion pour moi d'annoncer que la traduction collective de sa biographie [a retrouvé de l'énergie](#). Ce sera pour cet été si tout va bien...

OPINION

**Interview: How a hacker became a freedom fighter**

12 April 2008  
From NewsScientist Print Edition [Subscribe](#) and get 4 free issues.  
Michael Reilly

One of the founding fathers of "free software" and an esteemed elder of the hacking community, **Richard Stallman** has made defending people's freedoms his life's work. That usually means supplying hackers with software and attacking copyright law. But as he tells **Michael Reilly**, his advocacy of personal freedoms extends to the protection of true democracy and of the human rights increasingly being trampled on in the US and elsewhere

**Is it true you used to live in your office?**

Yes it is. I lived there for half of the 1980s and most of the 1990s.

**What made you do that?**



## Entretien : Comment un hacker est devenu un combattant pour la liberté

### [Interview: How a hacker became a freedom fighter](#)

Michael Reilly – 12 avril 2008 – NewsScientist.com

*L'un des pères fondateurs du « logiciel libre » et membre vénérable de la communauté des hackers, Richard Stallman a fait de la défense des libertés individuelles le combat de sa vie. Cela se traduit généralement par la fourniture de logiciels aux hackers et par des attaques contre les lois du copyright. Mais comme il le confie à Michael Reilly, sa défense des libertés personnelles individuelles s'étend à la protection de la vraie démocratie et des droits de l'Homme qui sont de plus en plus piétinés aux USA et ailleurs.*

**Est-il vrai qu'à une époque vous viviez dans votre bureau ?**

Oui c'est exact. J'y ai vécu durant la moitié des années 80 et quasiment durant l'ensemble des années 90.

**Qu'est ce qui vous a poussé à faire ça ?**

C'était pratique et économique. Devoir me déplacer pour rentrer chez moi quand j'avais sommeil était mauvais : d'abord parce que si j'avais sommeil ça pouvait me prendre deux heures pour me motiver et mettre ma veste et mes chaussures et tout ça. Et ensuite, la marche jusqu'à chez moi m'aurait réveillé, donc une fois arrivé je n'aurais quand même pas été au lit.

C'était nettement plus simple de pouvoir dormir à l'endroit où j'étais.

**Pour vous, qu'est ce que veut dire « hacker » ?**

Un hacker est quelqu'un qui apprécie l'intelligence espiègle. Je sais que pour beaucoup de personnes il représente un pirate informatique, mais puisqu'au sein de ma communauté nous nous appelons « hacker » je n'accepterai aucune autre signification. Si vous voulez parler de ces personnes qui cassent les codes de sécurité vous devriez parler de « cracker ». Le terme « hacker » ne se limite pas au domaine des ordinateurs. Au Massachusetts Institute of Technology il existe une ancienne tradition, les gens « hackent » les bâtiments et les lieux publics en y accrochant [le fameux panneau de signalisation « Nerd Crossing »](#) par exemple. Aucune sécurité n'est détournée et c'est espiègle et intelligent.

**À propos de l'espièglerie, quand avez-vous commencé à dire « happy hacking » à la place de « Au revoir » ?**

Quelque part dans les années 70. Je cherchais un moyen de dire au revoir et de souhaiter mes meilleurs vœux aux autres hackers et « happy hacking » semblait parfait pour ça. C'est devenu une habitude.

**Quand êtes-vous passé de hacker à activiste ?**

Cela s'est produit en 1983 quand j'ai initié le mouvement du logiciel libre (NdT : the free software movement). J'étais arrivé à la conclusion que les logiciels libres étaient le seul moyen d'apporter la liberté aux utilisateurs, j'ai donc lancé le mouvement pour provoquer cela.

**Au fond, qu'est ce que le mouvement des logiciels libres ?**

Il provient d'un désir de liberté. Je veux utiliser un ordinateur sans que personne ne contrôle ce que je fais dessus. Et je veux être libre de partager avec vous. Cela

signifie donc que je ne peux pas utiliser les logiciels propriétaires inclus avec la plupart des ordinateurs dans les années 80. Les logiciels propriétaires éloignent les utilisateurs et les rendent impuissants : éloignés parce qu'il est interdit de les partager et impuissants parce qu'ils ne disposent pas du code source. Les développeurs décident donc de ce que fait le logiciel et les utilisateurs n'ont pas leur mot à dire.

Pour changer cet état de fait j'ai écrit le système d'exploitation GNU. Dans le même cadre j'ai rédigé la GNU General Public License, qui garantit que tous les utilisateurs de ce système d'exploitation reçoivent, en plus du logiciel, quatre libertés essentielles : la liberté de faire fonctionner le programme comme ils le souhaitent, la liberté de partager le logiciel avec leurs amis et leurs voisins, la liberté de modifier le programme à leur convenance et la liberté de distribuer leur copie modifiée à tout le monde.

Lorsque [la composante Linux](#) a été ajoutée au système GNU nous avons un système d'exploitation complet et les gens se sont vraiment mis à l'utiliser. Ils ont aussi découvert certains avantages pratiques. C'est un système puissant et stable et bien sûr vous n'aviez pas à payer le droit de l'utiliser, ce qui est un avantage plutôt superficiel mais qui était important aux yeux de nombreuses personnes.

### **Les gens l'ont-ils largement adopté ?**

Le système GNU/Linux est devenu plutôt populaire, même si le souci de liberté ne s'est pas répandu autant que le système lui-même. Beaucoup de gens ont ainsi reçu la liberté sans trop savoir ce que cela représentait. Lorsque les gens ont la liberté mais n'en jouissent pas il est probable qu'ils la perdront. Par exemple, au milieu des années 90, certains distributeurs de GNU/Linux, on en trouvait déjà un certain nombre, ont commencé à ajouter des logiciels propriétaires et disaient « Regardez ce que l'on vous offre ! ». Ils

répandaient principalement le message que les programmes non-libres sont bons. Ce n'est pas ainsi qu'on fera passer le message que la liberté est importante. Cela montre bien que de perdre de vue la liberté a des conséquences concrètes.

**Vous êtes inquiet de la perte de toutes sortes de liberté. Est-ce pour cela que vous avez soutenu Dennis Kucinich dans sa campagne pour devenir le candidat Démocrate aux élections présidentielles ?**

J'ai soutenu son programme de restauration de différents droits de l'Homme, tel que l'habeas corpus, qui a été en partie détruit aux USA. Le président Bush a obtenu le pouvoir de mettre en prison des étrangers simplement en les désignant comme des « combattants ennemis ». Kucinich milite également pour l'arrêt de la torture et des guerres d'agressions. Il aurait mis un terme à l'occupation de l'Irak.

**Quel est la plus grande menace qui pèse sur le monde ?**

Les logiciels libres ne sont pas la première priorité, mais c'est un domaine dans lequel j'ai vu comment apporter ma contribution. Je pense qu'il y a deux graves problèmes. Le premier est le réchauffement climatique et l'environnement. Le second est la démocratie des droits de l'Homme et la séparation de la politique du monde des affaires. La seule façon de restaurer la démocratie est de mettre un terme au pouvoir politique du monde des affaires.

**Comment peut-on y parvenir ?**

Son emprise est si forte, je ne sais pas comment la renverser. Tout ce que je peux dire c'est qu'il faut le faire. Les gens tiennent pour acquis que le monde des affaires ait une grande influence en politique, mais tant que cela sera vrai nous n'auront pas de vraie démocratie.

**Est-ce que certains hommes politiques partagent cette vision ?**

Le président de l'Equateur, Rafael Correa, se bat vraiment pour la démocratie. Il est aussi en faveur des logiciels libres. Je lui ai expliqué le concept en personne et il a compris que cela avait du sens, sur le plan pratique comme sur le plan éthique. C'est un ancien professeur d'économie. Il a flanqué à la porte l'influence des US et des multinationales et a refusé de signer un traité d'échange avec les USA. Et une fois que le traité concernant les bases militaires américaines arrivera à expiration il ne le renouvèlera pas.

**Votre foi dans les logiciels libres vous a amené à considérer des manières de réformer les lois sur le copyright. Comment vous-y prendriez-vous ?**

Avec un copyright modéré. Les gens devraient être libres de redistribuer des copies exactes de quasiment tout, films, CD, ... à leurs amis ou à des inconnus si c'est à but non commercial. Les autres usages devraient toujours être couverts par le copyright.

**Croyez-vous que les gens se sensibiliseront aux logiciels libres et à la liberté en général ?**

Je suis de nature pessimiste. Mais tellement de choses surprenantes se sont produites que je ne pense pas savoir ce qui se passera aux cours des dix prochaines années. Je préfère admettre mon ignorance.