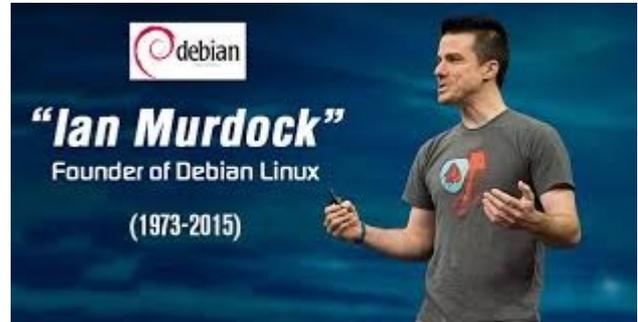


# Comment j'en suis venu à découvrir Linux

par Ian Murdock<sup>1</sup>  
le 17 août 2015

J'ai vu ma première station Sun à l'hiver 1992, alors que j'étais en premier cycle à l'Université Purdue. À ce moment, j'étais étudiant à la Keanner School of Management, et un amour d'enfance pour les ordinateurs fut ravivé par un cours obligatoire de programmation informatique que j'ai suivi durant le semestre d'automne (nous avions le choix entre le COBOL et le FORTRAN — qui même en 1992 semblait très démodé — et j'ai donc pris le COBOL qui me paraissait être le plus « business » des deux).



À peu près dix ans auparavant, mon père, un professeur d'entomologie à Purdue, avait remplacé au travail sa machine à écrire par un Apple II+. Pensant que son fils de neuf ans pourrait s'éclater avec, il l'apporta à la maison un week-end avec un jeu ressemblant à Space Invaders qu'il avait acheté au ComputerLand local. J'ai passé des heures sur cet ordinateur ce week-end-là. J'accompagnais bientôt Papa au labo à chaque opportunité pour passer autant de temps que possible sur l'ordinateur.

Étant un garçon de neuf ans, j'étais attiré au début, de façon prévisible, par les jeux et l'intérêt que je leur portais me conduisit à ma première exposition à la programmation : des magazines d'informatique qui incluaient des listes de code de jeux très simples, que je tapais laborieusement dans l'Apple — et après des heures de travail, j'espérais n'avoir fait aucune erreur (l'Apple II, au moins au sortir de la boîte, utilisait un simple éditeur ligne par ligne, par conséquent revenir en arrière et effectuer des changements était très fastidieux, sans mentionner en premier lieu la recherche d'erreurs).

Peu de temps après, j'ai rencontré Lee Sudlow tandis que je traînais au labo les week-ends. Lee était un des étudiants de deuxième cycle de Papa et avait commencé à utiliser l'Apple pour l'appuyer dans ses expériences. Lee était toujours heureux d'expliquer ce qu'il faisait, alors que je lui tournais autour zyeutant au-dessus de son épaule, son obligeance était sans doute motivée — au moins en partie — par le fait que le morveux de neuf ans scrutant chacun de ses gestes était le fils de son conseiller académique. Ne me rendant pas compte de ce genre de choses, je le regardais avec fascination taper du code dans l'Apple — code qu'il inventait lui-même, et non qu'il lisait dans un magazine informatique.

Entre l'apprentissage par l'exemple à travers l'étude du code des magazines et la tutelle occasionnelle de Lee, j'écrivais peu de temps après des jeux et autres programmes simples, d'abord en Applesoft BASIC et plus tard en assembleur pour 6502. Pour encourager mon intérêt croissant, Papa finit par acheter un Apple IIe pour la maison, et mon histoire d'amour avec l'informatique continua pendant plusieurs autres années. Néanmoins, en entrant dans mes années d'adolescence, l'ordinateur fut progressivement remplacé par des choses plus pressantes, comme le baseball, la musique, les filles, et au milieu des années 1980, la poussière s'amassait sur l'Apple dans le placard de ma chambre aux côtés de mes collections de romans Les Frères Hardy et de figurines Star Wars.

<sup>1</sup> Ian Ashley Murdock (28 April 1973 – 28 December 2015) was a German-born American software engineer, known for being the founder of the Debian project and Progeny Linux Systems, a commercial Linux company.

Mon obsession pour l'informatique est restée dormante durant les six années suivantes, jusqu'à ce qu'elle soit réactivée fortuitement durant ce cours de COBOL à l'automne 1992. Lorsque le cours se termina, j'ai naturellement perdu mon compte sur l'ordinateur central IBM 3090 sur lequel nous effectuions nos exercices et travaux de laboratoire. Heureusement, en tant qu'étudiant, j'avais droit à un compte personnel sur l'une des machines du centre informatique de l'université, soit l'IBM soit l'un des trois mini-ordinateurs Sequent Symmetry qui tournait sous DYNIX, une variante du système d'exploitation UNIX. Un ami m'a convaincu qu'UNIX était plus intéressant et avait davantage un brillant avenir que les machines virtuelles et moniteurs conversationnels d'IBM ; j'ai donc suivi son conseil, et je fis une demande pour un compte sur l'une des machines Sequent. La semaine suivante, j'étais le fier titulaire d'un compte sur sage.cc, complété par l'allocation princière de 500 kilo-octets d'espace disque. (Oui, je suis sarcastique — 500 kilo-octets est un espace misérable même pour 1992. J'ai fini par trouver des moyens d'y faire face).

J'eus un appétit vorace pour UNIX cet hiver. Je passais la plupart de mes soirées dans le sous-sol du bâtiment des mathématiques me prélassant dans la phosphorescente lueur verte des terminaux Z-29, explorant dans les moindres recoins le système UNIX. Un silence sinistre régnait dans ces salles de terminaux, où seul le son du clac, clac, clac de quelques douzaines de claviers persistait, interrompu parfois par un occasionnel chuchotement « Hé, regarde ça... »

Souvent, après une nuit d'exploration, je quittais le bâtiment par un détour pour passer devant une fenêtre de verre plat derrière laquelle le centre informatique abritait ses machines. Je contempiais avec émerveillement le Sequent Symmetry de la taille d'un réfrigérateur que je venais d'utiliser, je scrutais ses lumières clignotantes en sachant que des centaines de personnes se trouvaient encore à l'intérieur, même si ce n'était que virtuellement, grâce à la magie du temps partagé, une technique que les ordinateurs avancés utilisaient pour diviser la puissance de calcul de la machine entre plusieurs utilisateurs, fournissant l'illusion à chacun d'entre eux d'être le seul et l'unique à l'exploiter. Par-dessus tout, j'observais avec envie les opérateurs système assez privilégiés pour s'asseoir de l'autre côté de la fenêtre maniant dans la console le pouvoir tout-puissant du « superuser ».

Insatisfait par les Z-29, je commençais à rôder autour du campus la nuit tombée avec un ami, Jason Balicki, pour voir ce qui pouvait être trouvé d'autre. Jason suivait depuis quelques années le parcours des sciences informatiques, il savait donc où chercher (même si nous faisons nos explorations — cela faisait partie du plaisir — en entrant la nuit dans les bâtiments et en essayant les poignées des différentes salles, qui semblaient contenir des ordinateurs, pour voir si elles étaient restées ouvertes).

J'ai appris que les meilleurs labos se trouvaient dans le bâtiment de « management des sciences de l'ingénieur » (connu sur le campus par son malheureux acronyme, ENAD), où plusieurs salles de terminaux X offraient une interface graphique noir et blanc aux Sequent et autres machines UNIX du campus. Bientôt, mon lieu préféré de « hacking » (un terme que Jason me fit découvrir) se trouva dans un des laboratoires de terminaux X, qui étaient techniquement réservés aux seuls étudiants en ingénierie, une restriction qui n'était protégée par aucun mot de passe — et que nous ignorions scrupuleusement.

Mais la richesse du bâtiment de l'ENAD se trouvait dans ses laboratoires de stations de travail SUN. Contrairement aux modestes Z-29 et même aux terminaux X comparativement avancés, les SUN étaient des œuvres d'art, avec leurs boîtiers aux lignes pures et leurs écrans couleur haute résolution. Jason exposait en outre qu'ils faisaient tourner le meilleur UNIX existant, SunOS, même si les SUN étaient considérablement mieux verrouillés que les terminaux X, exigeant un compte sur

le réseau du département d'ingénierie pour y accéder ; je n'ai donc pas eu l'opportunité de me servir vraiment de SunOS avant un bon moment.

J'accédais également à UNIX depuis la maison via mon ordinateur basé sur un Intel 80286 et un modem 2400 bauds, qui m'épargnait une longue marche vers le labo d'informatique à travers le campus, particulièrement les jours de grand froid. Être capable d'accéder au Sequent depuis la maison était génial, mais je voulais reproduire l'expérience des terminaux X du bâtiment de l'ENAD. De la sorte un jour, en janvier 1993, j'entrepris de trouver un serveur X qui aurait pu tourner sur mon PC. Au cours de cette recherche sur Usenet, je suis tombé sur quelque chose nommé « Linux ».

Linux n'était pas un serveur X bien sûr, mais c'était quelque chose de bien mieux : un système d'exploitation complet pour PC ressemblant à UNIX, quelque chose que je n'avais même pas pu envisager et qui existait bien. Malheureusement, il réclamait au minimum un processeur 386, et mon ordinateur n'avait qu'un 286. J'ai commencé alors à mettre des sous de côté afin d'acquérir une machine suffisamment rapide pour le faire fonctionner, et pendant que j'économisais, je dévorais tout ce que je trouvais sur l'objet de mon désir. Quelques semaines plus tard, je postais un message sur le groupe Usenet d'informatique de Purdue demandant si quelqu'un sur le campus utilisait Linux — et je reçus une réponse d'un étudiant en informatique, Mike Dickey, qui m'invita volontiers à venir voir son installation Linux.

Empli d'inspiration, j'ai acheté une boîte de trente disquettes et j'ai commencé le lent processus de téléchargement de Linux sur celles-ci depuis une salle d'informatique du bâtiment Krannert, même si je devais encore patienter un mois de plus avant de pouvoir m'offrir un ordinateur capable d'effectuer une telle installation. Finalement, je n'ai pas pu attendre. Jason et moi avons trouvé dans l'une des résidences une salle d'informatique, qui n'avait pas été fermée, contenant un seul PC et, un soir de février, au milieu de la nuit, nous nous sommes mis à installer Linux sur cet ordinateur. Je me demande encore parfois ce qu'a dû penser l'étudiant malchanceux arrivé en premier le lendemain matin.

Le noyau Linux avait été créé environ une année et demie auparavant par Linus Torvalds, un étudiant de vingt-et-un ans en premier cycle d'informatique à l'Université d'Helsinki. Passionné par les ordinateurs depuis longtemps, Torvalds avait suivi un parcours à peu près similaire au mien, cependant il avait commencé son parcours en programmation sur un Commodore Vic-20, et n'avait pas été distrait par les intérêts traditionnels d'un adolescent durant les années 1980. La première exposition de Torvalds à UNIX datait de 1990 durant un cours universitaire et, comme moi, ce fut le coup de foudre au premier regard.

Durant l'automne de la même année, Torvalds suivit un cours sur les systèmes d'exploitation qui se basait sur le livre *Operating Systems: Design and Implementation* d'Andrew Tanenbaum, un professeur d'informatique de l'Université libre d'Amsterdam. L'ouvrage de Tanenbaum enseignait les systèmes d'exploitation par une série d'exemples au travers d'un clone d'UNIX pour PC qu'il avait écrit, nommé MINIX ; son livre contenait le code source complet de MINIX — le code de programmation humainement lisible (et modifiable) — ainsi qu'une série de disquettes pour que les lecteurs puissent installer, utiliser, et modifier le système d'exploitation.

Intrigué, Torvalds acheta un PC au début de l'année 1991, et rejoignit la communauté MINIX en plein essor, des dizaines de milliers de membres compétents largement réunis par le groupe de discussion Usenet `comp.os.minix`. Il commença alors non seulement à expérimenter MINIX mais aussi les nouvelles capacités de commutation de contexte du processeur Intel 80386 de son ordinateur. (La commutation de contexte facilite le fonctionnement de plus d'un programme à la

fois sur le processeur, l'un des prérequis d'un système à temps partagé comme celui du Sequant Symmetry, que j'allais découvrir l'année suivante à Purdue). Durant l'été 1991, les expériences de Torvalds avec la commutation de contexte commençaient à évoluer en un noyau de système d'exploitation à part entière, la pièce maîtresse du code d'un système d'exploitation permettant l'accès au processeur, à la mémoire, aux disques ainsi qu'aux autres composants de l'ordinateur, et qui fournit une interface simplifiée à ces fonctions informatiques, qui permettent à des applications complexes d'être écrites plus aisément (Glyn Moody, *Rebel Code: Inside Linux and the Open Source Revolution*, Basic Books, 2002, p. 8, 13, 32-42).

MINIX n'était pas le seul projet existant de système d'exploitation animant les amateurs en 1991, bien qu'il fût l'un des rares à être suffisamment complet pour être utilisé, et l'un des seuls à pouvoir fonctionner sur un modeste ordinateur. Le projet de système d'exploitation le plus connu jusque-là était GNU, présidé par Richard Stallman. Ce dernier, qui programmait depuis le milieu des années 1960 et qui a été développeur système au MIT de 1971 à 1983, était un « hacker » de la vieille école, un individu qui s'implique dans l'informatique pour l'amour même de l'informatique, parfois aussi de façon militante (comme pour Stallman), et pour la croyance que toute information devrait être librement partagée.

Le but du projet GNU était de produire un système d'exploitation libre (non seulement gratuit, mais aussi libre dans le sens qu'il serait librement modifiable) compatible avec UNIX (GNU était une sorte d'acronyme récuratif pour « GNU's Not UNIX », une sorte parce que cette dénomination employait une technique puissante souvent utilisée par les programmeurs appelée récursion, qui implique un calcul s'utilisant lui-même comme l'une de ses variables). Stallman lança le projet GNU en 1983 en réponse au marché fleurissant du logiciel propriétaire, dans lequel le code source n'était pas modifiable, et le plus souvent même inaccessible.

Le logiciel propriétaire était un fait relativement nouveau au début des années 1980 et, pour Stallman, une évolution très perturbante. Jusqu'à cette époque, les logiciels étaient pour la plupart distribués librement avec le matériel informatique, et les hackers partageaient souvent des copies de ce code source avec leurs propres modifications et améliorations. Stallman considérait cette tendance grandissante favorisant le logiciel propriétaire tout simplement comme le premier pas vers un 1984 numérique, où les utilisateurs d'ordinateurs, et finalement toute la société, seraient maintenus prisonniers par les intérêts cupides d'entreprises commerciales, et il était déterminé à stopper ce mouvement.

À la mi-1991, Stallman et un groupe informel de volontaires avaient assemblé quasiment l'ensemble du système d'exploitation GNU — un compilateur, un débogueur, un éditeur, un interpréteur de commandes (ou « shell »), et une variété d'outils et de bibliothèques qui ressemblaient à UNIX, mais en mieux — les versions du projet GNU étaient presque universellement considérées comme supérieures à celles de son éponyme. La seule pièce manquante était le noyau, et une petite équipe venait d'être formée pour écrire ce dernier élément à la Free Software Foundation de Stallman, une organisation à but non lucratif qu'il avait créé en 1985 pour superviser le développement de GNU et servir d'une certaine manière de gardien du logiciel libre. Les hackers à travers le monde pensaient qu'il ne s'agirait que d'une question de temps avant que GNU soit achevé et disponible, et qu'ils disposent enfin d'un système d'exploitation libéré du fardeau propriétaire.

À l'autre bout de la planète, le noyau de système d'exploitation de Torvalds devenait suffisamment complet pour être distribué au monde entier. Dans un message aujourd'hui célèbre publié sur `comp.os.minix` le 25 août 1991, Torvalds écrit :

*Bonjour à tous ceux qui utilisent minix -*

*Je suis en train d'écrire un système d'exploitation (libre) pour des compatibles PC-AT 386/486 (juste un hobby, ça ne sera pas grand et professionnel comme GNU). Cela mijote depuis avril, et ça commence à être prêt. J'aimerais bien avoir des retours sur les choses que les gens aiment ou non dans minix, comme mon système y ressemble d'une certaine manière (même conception du système de fichiers (pour des raisons pratiques) parmi d'autres choses).*

La réponse fut immédiate et très enthousiaste. Alors que tout le monde s'attendait à ce que GNU soit terminé de façon imminente, il n'était toujours pas disponible, au moins dans une forme qui aurait pu être utilisée sans une base échafaudée à partir d'UNIX. Et même si MINIX était populaire, il n'était pas gratuit, bien qu'il fût peu coûteux en comparaison des autres systèmes UNIX. Mais peut-être le plus important, MINIX était conçu d'abord comme une aide à l'enseignement, et non comme un logiciel de production. Tanenbaum avait donc horreur d'inclure les nombreux patchs et modifications du système d'exploitation qui étendaient ses capacités, qu'il recevait massivement tous les jours de hordes de passionnés du monde entier, craignant que leur ajout ne rende MINIX trop compliqué et, en conséquence, plus difficile à comprendre pour ses étudiants.

L'appât que représentait un système d'exploitation pour PC similaire à UNIX, aussi imparfait soit-il, libre et pouvant évoluer à la vitesse souhaitée par sa communauté, était une trop grande tentation pour beaucoup d'utilisateurs MINIX, qui se réunirent rapidement en nombre autour du nouveau système de Torvalds, surnommé à l'automne 1991 « Linux ». Mais Linux n'était qu'un noyau — il nécessitait en plus l'installation d'une variété d'outils et d'applications pour qu'il puisse effectivement faire quelque chose. Heureusement, la plupart de ces outils existaient déjà grâce au projet GNU de Stallman.

En 1992, quelques utilisateurs intrépides débutèrent le rassemblement de collections d'images disquettes qui combinaient Linux et la chaîne d'outils GNU afin de faciliter le fonctionnement du système et l'émergence de nouveaux utilisateurs. Ces collections (plus tard nommées « distributions ») s'amélioraient progressivement, et au moment où j'acquis finalement mon PC en mars 1993, la distribution Soflanding Linux System (ou SLS) s'était développée en une trentaine de disquettes et incluait maintenant une quantité d'applications — et oui, également un logiciel identique à celui qui faisait fonctionner les terminaux X du bâtiment de l'ENAD.

Quoi qu'il en soit, je n'ai jamais trouvé le temps d'essayer de me connecter depuis mon PC au serveur X basé maintenant sur Linux du Sequent, ce qui aurait été terriblement lent à 2400 bauds — plusieurs milliers de fois plus lent que la vitesse actuelle. Je disposais à présent d'un UNIX rien qu'à moi, un UNIX que je pouvais explorer juste là, sur mon bureau. Et cette exploration je l'ai faite, dans un véritable cours intensif UNIX. Lorsque je me suis remis de l'excitation d'être le « superuser », cette puissance indicible que je n'avais pu auparavant qu'admirer au travers d'une vitre, je suis devenu captivé non pas tellement par Linux lui-même mais par le processus qui l'a vu naître — des centaines d'individus hackant depuis leur petit coin du système et utilisant Internet pour échanger du code, et lentement mais sûrement améliorer le système grâce à chacune des modifications.

J'ai alors entrepris de faire une contribution personnelle à cette communauté grandissante, une nouvelle distribution appelée Debian qui serait plus facile d'utilisation et plus robuste, parce qu'elle serait construite et maintenue collaborativement par ses utilisateurs, un peu comme Linux.

