

Unix: A history and a Memoir

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 14 janvier 2020

<https://www.bortzmeyer.org/unix-history-memoir.html>

Auteur(s) : Brian Kernighan

ISBN n°9781695978553

Éditeur : Kindle Direct Publishing

Publié en 2020

Ce livre est une histoire du système d'exploitation Unix, par une des personnes qui ont suivi l'aventure d'Unix depuis le début, Brian Kernighan.

D'abord, Kernighan écrit très bien, il a un vrai talent pour tout expliquer, y compris les questions informatiques complexes (comme le fameux tube, une des inventions les plus marquantes dans Unix). Ensuite, il a lui-même travaillé au développement d'Unix (il a notamment participé à la création du langage C, dans lequel Unix a été rapidement réécrit, après ses débuts en langage d'assemblage.) Cela lui permet de faire revivre une époque aujourd'hui bien distante.

Il y a fort longtemps, dans un pays lointain, pas du tout dans la Silicon Valley mais à Murray Hill, un gentil roi, la compagnie AT&T avait installé son service de recherche et développement, les Bell Labs. Les Bell Labs (qui existent encore aujourd'hui sous ce nom mais ne sont plus que l'ombre de ce qu'ils étaient) sont devenus une légende de la physique, de la mathématique et de l'informatique, avec pas moins de neuf prix Nobel obtenus (non, l'invention d'Unix n'a pas été récompensée par un prix Nobel.)

C'est dans ce pays merveilleux que deux informaticiens, Ken Thompson et Dennis Ritchie, après le semi-échec du projet Multics, et le retrait des Bell Labs de ce travail, se sont attaqués à une de ces tâches où tous les gens raisonnables vous disent « c'est complètement irréaliste, jeune homme, vous n'êtes pas sérieux ». Ils ont écrit un système d'exploitation et Unix était né, prêt à conquérir le monde. Ce livre est l'histoire d'Unix. Elle est pleine de rebondissements, de crises, de discussions.

Pour rappeler l'importance d'Unix, il faut se souvenir que beaucoup de choses qui nous semblent aujourd'hui évidentes en informatique ne l'étaient pas à l'époque. Par exemple, la grande majorité des systèmes d'exploitation imposaient de fixer une taille maximale à un fichier avant sa création. Si elle était trop faible, on devait re-crée un autre fichier et copier les données. Si cette taille était trop élevée, on gaspillait de l'espace disque. Unix a mis fin à cela et, aujourd'hui, cela va de soi. De même Unix a unifié

les différents types de fichiers. Avant Unix, plusieurs systèmes d'exploitation avaient des commandes différentes pour copier un fichier contenant un programme Cobol et un fichier de données !

L'atmosphère très spéciale des Bell Labs, informelle, avec peu de bureaucratie, un accent mis sur les compétences et pas sur les titres (une méritocratie, une vraie) a beaucoup aidé à développer un système d'exploitation à succès. Kernighan raconte beaucoup d'histoires amusantes, mais consacre également du temps à l'analyse des facteurs importants dans le succès des Bell Labs. Il insiste sur les facteurs physiques (« *geography is destiny* ») : tout le monde sur le même site, et un mélange de bureaux fermés (même les stagiaires avaient leur propre bureau fermé, loin de l'*open space* bruyant empêchant la concentration) et de pièces communes où on pouvait aller quand on voulait, discuter et interagir avec les autres. Les Bell Labs ont été un cas peut-être unique, où toutes les conditions étaient réunies au même endroit, pour produire une étonnante quantité d'inventions géniales. Le tout était aidé par un financement stable et un management qui laissait les chercheurs tranquilles. Il est curieux (et triste) de noter qu'une entreprise 100 % capitaliste comme AT&T donnait plus de liberté et de stabilité financière à ses chercheurs qu'une université publique d'aujourd'hui, où les chercheurs doivent passer tout leur temps en travail administratif, en évaluation, et en recherche d'argent.

Aux Bell Labs, il était fréquent pour un chercheur de travailler sur plusieurs sujets de front et le livre de Kernighan donne une petite idée de la variété des sujets. Anecdote personnelle : j'ai utilisé (très peu !) le système Ratfor que l'auteur avait écrit, quand je faisais du calcul numérique.

Une particularité d'Unix est en effet la profusion d'outils pour informaticiens qui ont été développés sur ce système. L'auteur consacre de nombreuses pages à ces outils en insistant sur le fait que le groupe Unix des Bell Labs maîtrisait toujours la théorie et la pratique. Chaque membre du groupe pouvait écrire une thèse en informatique théorique, et inventer puis programmer des outils utiles. Mais on oublie souvent que les premiers utilisateurs d'Unix n'étaient pas que des informaticiens. Le premier argument « de vente » d'Unix auprès de la direction des Bell Labs était ses capacités de... traitement de texte. Le service des brevets de Bell Labs déposait beaucoup de brevets, et leur préparation prenait un temps fou. En bons informaticiens, les auteurs d'Unix ont automatisé une grande partie des tâches, et les juristes se sont mis à préparer les demandes de brevets sur Unix...

De nos jours, on associe souvent Unix au logiciel libre, puisque Linux, FreeBSD et bien d'autres héritiers de l'Unix original sont libres. Mais à la grande époque des Bell Labs, ces considérations politiques étaient absentes. Kernighan n'en parle jamais et, au contraire, insiste sur le verrouillage de bien des innovations par des brevets. C'est en raison de la licence restrictive de l'Unix d'AT&T que des systèmes comme Linux ou FreeBSD n'ont plus une seule ligne du code original d'AT&T : il a fallu tout réécrire pour échapper aux avocats.

Kernighan ne fait pas que raconter des anecdotes édifiantes. Il corrige également quelques légendes. Par exemple, le fameux commentaire dans le code source d'Unix « *You are not expected to understand this* » ne veut pas du tout dire « lecteur, tu es stupide, laisse ce code aux pros » mais « il n'est pas nécessaire de comprendre ce bout de code pour comprendre le reste ».

Vous ne serez pas surpris d'apprendre que le livre a été composé sur Unix, avec groff et Ghostscript.