

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^[1].

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

systeme 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 Environment (*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 systèmes de télécommunications 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)