

Deux hommes et un ordinateur

David J. du Colombier Jean-Baptiste Campesato

30 décembre 2006



Cet article n'est pas présenté ici sous sa forme finalisée et est encore en cours de rédaction. Par conséquent, de nombreuses erreurs peuvent encore subsister et plusieurs parties n'ont pas encore été écrites ou sont encore en cours de rédaction.

Ce document est propriété de ses auteurs, et toute reproduction est proscrite.

Table des matières

1	Introduction	4
2	MULTICS	4
3	Unix Time-Sharing System^[3]	4
3.1	La naissance d'Unix	4
3.2	Unix Time-Sharing System ^[4]	4
3.3	Le langage B ^[5]	5
3.4	Le langage C ^[7]	5
3.5	L'évolution d'Unix Time-Sharing System	5
3.6	Unix Time-Sharing System 6, 7 et UNIX 32V	6
4	UNIX System	6
4.1	Les débuts de l'Unix de AT&T	6
4.2	UNIX System V	6
4.3	UnixWare	7
5	BSD, <i>Berkeley Software Distribution</i>^[11]	7
5.1	1BSD et 2BSD	7
5.2	3BSD et 4BSD	7
5.3	USL et BSDI	7
6	XENIX	8
6.1	Microsoft	8
6.2	Santa Cruz Operation	8
7	Le tournant des laboratoires Bell	8
7.1	Unix Time-Sharing System	8
7.2	Plan 9 from Bell Labs ^[12]	8
7.3	Inferno	9
8	"The Unix Wars"^[14]	10
9	MINIX et Linux	10
9.1	MINIX ^[15]	10
9.2	Linux	10

Résumé

Cet article aborde que certains points essentiels de l'histoire d'Unix, il ne s'agit que d'une approche. Si vous souhaitez obtenir plus d'informations sur l'histoire d'Unix nous vous conseillons la lecture des papiers recueillis par Dennis Ritchie disponibles sur sa page personnelle, ainsi que la lecture du livre *A Quarter Century of Unix* de Peter H. Salus.

1 Introduction

En 1964, les laboratoires Bell étaient une branche de Bell System appartenant à la firme AT&T. Cette dernière avait le monopole sur la téléphonie aux États Unis. Les laboratoires Bell furent les premiers à se spécialiser dans ce domaine, ils mirent au point de nombreuses inventions, par exemple, le premier ordinateur digital en 1939 ainsi que le transistor en 1947 (John Bardeen, Walter Brattain et William Shockley)[1, 2].

2 MULTICS

Le MIT, les Laboratoires Bell et General Electric se regroupèrent dans l'intention de développer un système d'exploitation disposant de fonctionnalités nouvelles, comme la gestion de terminaux distants, un système de fichier hiérarchisé (gérant les noms longs, les liens symboliques...), la segmentation, la pagination, la mémoire virtuelle... Ce système se vit nommer MULTICS pour *MULTiplexed Information and Computing Service*. Il hérita aussi de concepts de CTSS (*Compatible Time-Sharing System*) comme la gestion de plusieurs utilisateurs avec le temps partagé, ou l'invite de commande.

En 1966, Ken Thompson diplômé à Berkeley rejoignit les laboratoires Bell et fut appelé à travailler sur MULTICS.

3 Unix Time-Sharing System^[3]

3.1 La naissance d'Unix

Malgré les efforts des laboratoires Bell, de General Electric et du MIT, MULTICS ne devint jamais un système correctement utilisable. Suite à cet échec, les laboratoires Bell se retirèrent du projet à l'été 1969.

Ken Thompson récupéra un ordinateur DEC (*Digital Equipment Corporation*) PDP-7 inutilisé dans son laboratoire et eut l'idée de l'utiliser pour son jeu Space Travel.

Ceci était plus un logiciel de simulation du système solaire qu'un véritable jeu.

Ken Thompson débuta son projet par la création d'un petit système d'exploitation écrit en assembleur inspiré des concepts de MULTICS, puis réalisa le portage de Space Travel.

Le noyau ainsi que les outils rudimentaires furent assemblés sous le système d'exploitation GECOS (*General Electric Comprehensive Operating System*) sur cette même machine.

Ce système d'exploitation simple offrait des perspectives intéressantes par rapport à MULTICS et devint finalement un projet à part entière qui éveilla la curiosité au sein des laboratoires Bell.

En 1969, le premier Unix venait de paraître, cette version fut d'abord nommée par un nom approchant de *Ken's new system*.

Quelques mois plus tard Brian Kernighan proposa le nom *UNICS* suite à un jeu de mot avec *MULTICS*, mais ce nom ne fut pas retenu. Cependant il donna naissance au nom UNIX dont personne ne se souvient de l'origine.

3.2 Unix Time-Sharing System^[4]

À la fin de l'été 1971, suite à une demande datant de 1970, l'équipe s'occupant d'Unix a obtenu un PDP-11/20. Une version simplifiée d'Unix fut écrite pour un compilateur croisé (*cross-compiler*)

sous PDP-7, puis Unix fut entièrement porté. En 1971, le premier Unix Time-Sharing System (pour préciser qu'il s'agit d'un système d'exploitation multi-utilisateur et multitâche) était né.

3.3 Le langage B^[5]

Tous les programmes pour Unix étaient écrits en assembleur, qui n'était pas un langage agréable d'utilisation. De plus les programmes devaient contenir tout ce dont ils avaient besoin pour s'exécuter, il n'y avait pas de lien pour l'utilisation de bibliothèques. D'où la nécessité d'utiliser des langages de haut niveau. La première idée fut d'utiliser TMG (de McClure), mais à la sortie du langage TMG, Ken Thompson estima qu'il n'offrait pas ce dont ils avaient besoin pour programmer un système. Il décida de porter Fortran sous PDP-7 mais finalement il développa le B dans les années 1969-1970 avec l'aide de Dennis Ritchie. Le langage BCPL^[6] (de Martin Richards de l'université de Cambridge) a grandement influencé le développement du B.

Le compilateur B pour PDP-7 fut d'abord écrit en TMG avant d'être réécrit en B, le compilateur croisé (*cross-compiler*) B pour PDP-11, tournant sous PDP-7, fut directement écrit en B.

Finalement Ken Thompson n'a pas réécrit Unix Time-Sharing System en B bien qu'il en ait temporairement eu envie ([3] section *High-level languages*, deuxième partie.).

3.4 Le langage C^[7]

Le B comportait plusieurs problèmes. Notamment au niveau du typage, le B (ainsi que BCPL) n'avait qu'un seul type qui était de la taille des mots (*word*) correspondant à l'architecture, ensuite l'arithmétique sur les flottants n'était possible que parce qu'un *mot* était capable de contenir un flottant, or ce n'est plus le cas avec le PDP-11/20 16bits. Il devint donc nécessaire de créer des types pour gérer les *bytes* (ou *octets*, codé sur 8 bits), les caractères ainsi qu'un type pour la gestion des flottants (le fabricant, *Digital Equipment Corporation*, avait promis que la gestion des flottants serait disponible, notamment avec le PDP-11/45).

De plus le compilateur B donnait des programmes lents à cause de la technique du *threaded code*^[9]. C'est pourquoi en 1971 Dennis Ritchie modifia le B, il ajouta un type caractère (*char*) et réécrivit un compilateur pour PDP-11. A ses débuts le C eut pour nom NB (*new B*).

Dennis Ritchie continua ses modifications jusqu'en 1973. Puis de 1972 à 1977 Alan Snyder, Steven C. Johnson, Michael Lesk et Ken Thompson contribuèrent aussi au C. Durant cette période le langage fut de plus en plus utilisé (notamment avec le compilateur portable de Johnson, *Johnson's portable compiler*) et est en constante croissance.

En 1978, Brian Kernighan et Dennis Ritchie écrivirent un livre qui fit office de standard^[10].

3.5 L'évolution d'Unix Time-Sharing System

Lors de la sortie de la seconde édition, le 12 juin 1972, le nombre d'installations au sein du laboratoire s'élevait à dix machines.

La troisième édition, finalisée en février 1973 apporta une fonction majeure, la notion de tube (*pipe*). Un tube consiste à lancer plusieurs processus en parallèle en permettant une interaction entre eux.

À cette date, le nombre d'installation s'élevait à seize ordinateurs. Ces deux premières éditions d'Unix Time-Sharing System étaient écrites en langage assembleur pour PDP-11/20, la troisième pour PDP-11/45, ce qui rendait le système très difficile à porter sur une autre architecture que le PDP-11.

Ainsi Ken Thompson prit la décision de réécrire son système en utilisant le langage C, créé par Dennis Ritchie quelques mois plus tôt.

La quatrième édition vit ainsi le jour en novembre 1973. Le nombre d'installation s'était élevé à 30 machines.

3.6 Unix Time-Sharing System 6, 7 et UNIX 32V

Unix Time-Sharing System 6 apparut en 1975, BSD de Berkeley distribua des modifications et des ajouts pour cette édition.

Quant à la septième édition, elle parut en 1979 avec le Bourne Shell (*sh*) de Steve Bourne. Cette même année Unix Time-Sharing System fut porté sur plate-forme VAX sous le nom d'UNIX 32V. Unix Time-Sharing System 6, 7 et 32V furent les derniers Unix Time-Sharing System réalisés entièrement par les chercheurs des laboratoires Bell.

4 UNIX System

4.1 Les débuts de l'Unix de AT&T

Le succès d'Unix Time-Sharing System attira l'attention de AT&T, qui décida de réorganiser les laboratoires Bell autour de son nouveau projet.

Le groupe chargé du développement d'Unix Time-Sharing System, dirigée par Ken Thompson, fut nommé *Computer Research Group* (CRG).

Un second groupé nommé *UNIX System Group* (USG) fut chargé du support. Un troisième groupe, le *Programmer's WorkBench* (PWB) fut impliqué dans le développement de technologies liés à Unix et développa son propre système.

Ces trois groupes furent intégrés au sein du *UNIX System Development Lab* dès 1983. La branche de Bell Labs implantée à Columbus (dans l'Ohio), responsable des *Operations Support Systems* participa également à la création d'une variante d'Unix, Columbus Unix (CB).

AT&T fut dans un premier temps incapable de profiter commercialement du travail effectué par les laboratoires Bell. Par crainte que AT&T dispose d'un trop grand monopole, le Département de Justice des États Unis prit la décision d'interdire à AT&T la commercialisation de services autres qu'en rapport avec les télécommunications.

En 1982, AT&T parvint à obtenir l'accord de commercialiser son système d'exploitation en tant qu'outil pour la téléphonie. Ce système d'exploitation fut nommé UNIX System III. Il est issu des recherches effectués sur PWB et CB et tourne sur plate-forme VAX.

4.2 UNIX System V

En 1983, AT&T sort une seconde version de son système d'exploitation commercial, UNIX System V. La principale avancée de cette version est l'implémentation du paquetage IPC, développé par la branche des laboratoires Bell de Columbus.

En 1984 apparaît une première mise à jour, UNIX System V Release 2, qui apporte entre autre le Bourne Shell (*sh*) de Steve Bourne. En même temps est publiée le *System V Interface Définition*, manuel décrivant les spécifications qui permettent de réaliser un système d'exploitation compatible et qui sera mis à jour par la suite à chaque nouvelle version de System V.

En 1986, UNIX System V Release 3 apportant notamment le mécanisme de communication STREAM issu d'Unix Time-Sharing System 8, est suivie par UNIX System V Release 3.2 qui apporte les fonctions réseaux, ainsi que l'incorporation de quelques fonctionnalités appartenant à XENIX.

En 1988, voit le jour la plus importante avancée avec l'arrivée de System V Release 4. Cette version majeure résulte de la fusion de de BSD, SunOS et System V. Parmi les améliorations, on peut citer l'utilisation du protocole TCP/IP de Berkeley, le C ANSI des laboratoires Bell, l'interpréteur de commandes ksh de AT&T, X11 du MIT, l'environnement OPEN LOOK et le Network File System (NFS) de Sun.

Deux mises à jour mineures ont suivi, UNIX System V Release 4.1 et UNIX System V Release 4.2 en 1992, appelé UnixWare 1.0 par Novell.

4.3 UnixWare

En 1993, la branche *UNIX Systems Operations* en charge du développement d'UNIX System V fut vendue à la société Novell. UNIX System V Release 4.2 fut nommé UnixWare 1.0.

5 BSD, *Berkeley Software Distribution*^[11]

5.1 1BSD et 2BSD

En novembre 1973, Ken Thompson et Dennis Ritchie allèrent à l'université de Purdue afin de présenter la première spécification concernant Unix. C'est suite à cette conférence que Bob Fabry, qui était professeur à Berkeley, l'université de Californie, eut envie de tester Unix Time-Sharing System.

En Janvier 1974 l'université reçut Unix Time-Sharing System 4 et l'étudiant Keith Standiford commença à l'installer sur le PDP-11/45 que venait d'acquérir Berkeley. Cependant tout ne se déroula pas comme prévu, Berkeley dut faire appel à Ken Thompson, le matériel dont disposait Berkeley n'était pas le même que celui en possession des laboratoires Bell.

Au début de l'année 1977, Bill Joy fonda la *Berkeley Software Distribution*, il y créa 1BSD (Pour *First Berkeley Software Distribution*), qui n'était pas un système mais un ensemble de modifications et d'ajouts (Notamment un compilateur Pascal et un éditeur de ligne, ex) pour Unix Time-Sharing System 6. En 1978, il sortit 2BSD qui contenait, entre autre, l'éditeur de texte *vi* (amélioration de ex).

Le développement de 2BSD pour plateforme PDP-11 est resté en activité jusqu'en 2000, sous le nom de 2.11BSD.

5.2 3BSD et 4BSD

La troisième édition de BSD sortit en mars 1980, cette édition est désormais basée sur Unix 32V et inclut les modifications apportés par 1BSD et 2BSD. Il s'agit désormais d'un système d'exploitation à part entièrement fonctionnait sur plateforme VAX. En avril 1982, le protocole TCP/IP était implémenté dans la version 4.1aBSD, mais elle ne fut pas distribuée. A la fin du printemps Bill Joy rejoignit Sun Microsystems, laissant à Sam Leffler la gestion du projet.

En août 1983 4.2BSD vit le jour et connut une gloire rapide.

La version 4.3BSD Reno sortit en 1990, à partir de maintenant BSD fonctionnait sous les architectures MIPS de DEC et SPARC de Sun. Elle donna naissance à BSD Net/2 (4.3BSD-Lite), en 1991, qui lui même permit à BSD/386 (1992), 386BSD (1992) et NetBSD (1993) de voir le jour. FreeBSD apparut en 1993, basé sur 386BSD, puis OpenBSD apparut en 1996 suite à une scission entre les développeurs de NetBSD.

Du code de 4.4BSD-Lite est présent dans NetBSD 1.0 et FreeBSD 2.0 (en 1995) ainsi que du code de 4.4BSD-Lite Release 2 dans 2.11BSD Patch 335, FreeBSD 3.0, NetBSD 1.3 et OpenBSD 2.3 (en 1998), ils ont inclus les dernières fonctionnalités apportées par Berkeley.

Le développement de 4.3BSD-Tahoe pour plateforme VAX est resté en activité jusqu'en 2003, sous le nom de 4.3BSD-Quasijarus.

5.3 USL et BSDI

Suite à la sortie de 4.4BSD Alpha en 1992 et 4.4BSD en 1993, la compagnie BSDI (*Berkeley Software Design, Incorporated*) se mit à vendre ces versions. AT&T fit un procès à BSDI car BSD contenait du code venant d'AT&T, BSDI dirent qu'ils n'étaient pas responsable du code contenu dans BSD. AT&T poursuivit donc le groupe BSD chez Berkeley qui fut forcé à modifier BSD afin qu'il ne contienne aucun code issu de AT&T. Cette nouvelle version fut nommée 4.4BSD-Lite et sortie en 1994, auquel suivi 4.4BSD-Lite Release 2, une version corrigée, en 1995.

Par la suite, Berkeley cessa le développement de son système d'exploitation, comptant sur le développement de dérivés comme FreeBSD et NetBSD, tous deux initialement basés sur 386BSD.

6 XENIX

6.1 Microsoft

En juin 1979, Microsoft acheta une licence d'Unix Time-Sharing System 7 puis annonça, en août 1980, un portage d'Unix sur machines 16 bits sous le nom de XENIX, la première version sortie le 25 août 1980. Microsoft vendait des licences aux sociétés afin qu'elles portent XENIX sur leurs machines (par exemple, dès 1983 Santa Cruz Operation proposait un portage pour les processeurs Intel 8086 sous le nom de Santa Cruz Operation XENIX).

XENIX 2.0 sortit en 1985, cette version fut basé sur System V Release 3.2.

6.2 Santa Cruz Operation

En 1987, Microsoft vendit XENIX à Santa Cruz Operation (*Santa Cruz Operation*) contre 25% de leur capital. XENIX devint SCO UNIX en 1989 (Santa Cruz Operation UNIX System V/386 Release).

Santa Cruz Operation acheta UnixWare à Novell en 1995, Santa Cruz Operation mit donc sur le marché SCO OpenServer basé sur SCO UNIX et UnixWare.

7 Le tournant des laboratoires Bell

7.1 Unix Time-Sharing System

En 1985 voit le jour la huitième édition de Unix Time-Sharing System. Développée à partir de 4.1cBSD d' l'université de Berkeley, ce fut la première édition à incorporer d'éléments provenant de systèmes d'exploitation dérivés. Tout comme Unix 32V, cette édition fonctionnait sur la nouvelle architecture VAX de DEC, mais profitait d'avantage des possibilités offertes par le matériel grâce au travail fournit par Bekerley. Cette édition fut la première à disposer du protocole TCP/IP de Berkeley, mais préférait aux Sockets un nouveau système d'interface de communication appelé STREAM. Plusieurs éléments, dont notamment l'interface STREAM, furent utilisés par la suite par AT&T dans UNIX System V.

La huitième édition d'Unix Time-Sharing System fut la dernière édition à être présentée sous la forme d'une distribution complète et cohérente et a avoir été diffusée. La neuvième édition parue en 1986 et la dixième en 1989 représentaient en réalité l'édition des manuels, le système étant développée continuellement et ne fut jamais redistribué en dehors des laboratoires Bell. Le but initial de ces trois dernières édition furent d'une part la continuation de la recherche logicielle sous Unix, mais surtout de profiter du développement effectué conjointement avec Berkeley et parvenir à corriger les erreurs.

Le développement d'Unix Time-Sharing System cessa aux alentours des années 1990 lorsque l'architecture VAX est devenue obsolète et que leur nouveau système, Plan 9 est devenu un système suffisamment stable pour le remplacer. Plusieurs logiciels initialement écrits pour Unix Time-Sharing System, comme par exemple l'éditeur de texte Sam de Rob Pike et le shell rc de Tom Duff furent portés sur Plan 9.

7.2 Plan 9 from Bell Labs^[12]

Vers le milieu des années 1980, les traditionnels gros serveurs exploités par de multiples utilisateurs au travers de terminaux laissaient peu à peu place à un nouveau genre d'ordinateur : les stations de travail. Ces machines étaient individuelles, disposaient d'une interface graphique et étaient reliés au réseau de manière à pouvoir interagir entre elles et leurs différents utilisateurs.

Unix Time-Sharing System était un système inadapté à ce nouveau genre d'utilisation, et ne possédait ni de protocole performant pour travailler en réseau, ni d'interface graphique. De plus, de nombreuses fonctions d'Unix Time-Sharing System avaient été influencés par le matériel sur lequel il reposait. Ceci est le cas, par exemple, du terminal VT100 de DEC, qui ne disposait que

d'une seule entrée pour l'écriture et l'insertion de commandes.

Quelques années auparavant, Bill Joy à l'université de Berkeley avait créé le protocole de communication TCP/IP pour 4BSD pour plateforme VAX, puis plus tard, chez Sun, l'interface graphique SunView pour SunOS pour les stations de travail Sun équipés de Motorola 68000. Cependant les chercheurs des laboratoires Bell craignirent une complexification importante du système par l'apport d'ajouts qui n'avaient pas été pensées dans l'élaboration du système de base.

Les problèmes d'Unix Time-Sharing System semblaient trop profonds pour être corrigés, le développement d'un nouveau système d'exploitation débuta vers 1987, sous la direction de Rob Pike et Dennis Ritchie.

Ce système d'exploitation s'appelle *Plan 9 from Bell Labs*, son nom est inspiré du célèbre film de Ed Wood, *Plan 9 from Outer Space*, qualifié comme le pire film de tous les temps. Communément appelé Plan 9, ce système est un système d'exploitation moderne qui fonctionne selon une architecture distribuée, et est conçu de manière à être portable. Il fut écrit dans un premier temps dans le langage Alef (dérivé du C) puis dans un langage issu d'une simplification de l'ANSI C (standard C89).

Un système d'exploitation distribué permet aux ressources des différentes machines connectés au réseau (serveurs de processeurs, serveurs de stockage, stations, terminaux) d'être exploités de manière à pouvoir fonctionner ensemble, et ce de manière transparente pour l'utilisateur. En outre, les principes de base d'Unix ont été renforcés, le système est réellement *tout fichier*, et le principe du *tout réseau* a été ajouté. Ce système repose sur le protocole 9P, qui sera plus tard, à la sortie de la quatrième édition, remplacé par 9P2000.

Parmi les nombreux utilitaires dans Plan 9, on peut compter le gestionnaire de fenêtre 8 1/2 de Rob Pike, plus tard remplacé par Rio, l'environnement de développement acme, l'éditeur de texte sam, la console à entrées multiples et à contenu éditable. Plan 9 fut le premier système à utiliser l'encodage de caractère UTF-8, développé par Ken Thompson.

Plan 9 fut développé en interne au sein des laboratoires Bell de 1987 à 1993. La première édition publique sortie en 1993, puis la deuxième édition en 1995.

Plan 9 était vendu sans limite d'utilisation et accompagné de ses sources pour la somme de 350 dollars américains. En février 1996, les laboratoires Bell se séparèrent de AT&T pour fonder leur propre groupe spécialisé dans le réseau informatique, Lucent Technologies.

Suite à la création de Lucent Technologies, la troisième édition sortie en 2000 sous une licence semi-libre et gratuitement, et la quatrième édition suivit en 2002. Depuis 2003, Plan 9 est soumis à une licence libre, reconnue par l'Open Source Initiative. Depuis, malgré le départ de la plupart des développeurs suite à des problèmes avec les dirigeants de Lucent Technologies, le système est activement développé par une communauté de passionnés.

Plan 9 est aujourd'hui capable de fonctionner sur les architectures Motorola MC68000, MIPS, SPARC, Intel i386, Intel i960, DEC Alpha, IBM PowerPC, Acorn ARM, AMD 29000, IBM PC et sur le mini-ordinateur iPAQ de Compaq. Il est développé chez les Laboratoires Bell par Russell Cox et Jim McKie.

7.3 Inferno

Issu des mêmes principes qui animent le développement de Plan 9, Inferno est un système d'exploitation destiné à l'embarqué créé initialement par Rob Pike et Phil Winterbottom au début des années 1990.

Le nom Inferno est issu du nom du premier volume de *La Divine Comédie*, écrit par Dante Alighieri. Inferno est écrit dans le langage Limbo, un langage semi-interprété, reposant sur des pseudo-binaires multi-plâteformes exécutés d'une manière transparente à travers une machine virtuelle.

Ce langage, dans son objectif, est similaire au langage Java de Sun Microsystems développé à la même époque, mais ils évoluèrent chacun dans leur propre voie. Inferno est capable de fonctionner aussi bien nativement que hébergé sur un autre système d'exploitation comme FreeBSD, IRIX, Linux, Mac OS X, Windows NT, 2000, XP, Plan 9 from Bell Labs et Solaris. Inferno est un des rares systèmes d'exploitation offrant la capacité de fonctionner sur une machine disposant d'un méga-octet d'espace disque, d'un méga-octet de mémoire vive et un processeur dénué de Memory

Management Unit.

En 1996, la société anglaise Vita Nuova[13] fut créée dans le but de promouvoir le système d'exploitation Inferno et assure aujourd'hui son développement.

8 "The Unix Wars"[14]

Des sociétés achetèrent des licences d'UNIX System V afin de développer leurs Unix propriétaires, ce qui créa un grand nombre d'Unix avec des caractéristiques distinctes.

En 1987, AT&T et Sun Microsystems décidèrent de s'allier afin de développer un unique Unix contenant les meilleures caractéristiques de leurs Unix. Le fait que Bill Joy (*voir BSD*) soit cofondateur de Sun Microsystems permettait à la firme d'accéder aux développements de BSD. C'est ainsi qu'en 1988 UNIX System V Release 4 vit le jour en offrant certaines caractéristiques de SunOS (Notamment NFS, TCP/IP, Open Look, NeWS.)

Cependant les concurrents craignirent que Sun Microsystems ait un avantage et fondèrent l'OSF (*Open Software Foundation*) en 1988 et écrivirent leur propre Unix (*OSF/1*) qui devait être indépendant d'AT&T, ils possédaient, entre autre, X11 et Motif. Finalement l'OSF échoua et fut repris par DEC. Cependant X11 et Motif s'imposèrent.

AT&T mit fin à l'Unix Wars en 1991 en proposant de vendre Unix Time-Sharing System (jusque la septième édition), UNIX System et *UNIX Systems Laboratories* (anciennement *UNIX System Development Lab*, puis *UNIX Systems Operations*). En 1992 Novell Corporation envoya une lettre pour l'achat, qui fut exécuté le 16 juin 1993. Novell transféra les droits de la marque *Unix* et les spécifications à X/Open, puis en 1995, Novell Corporation vendit certains droits d'UNIX System V à Santa Cruz Operation (*Santa Cruz Operation*).

En 1996, X/Open et l'OSF fusionnèrent afin de créer *The Open Group*.

9 MINIX et Linux

9.1 MINIX[15]

Andrew Tanenbaum est un ancien chercheur des laboratoires Bell, à l'époque où il y travaillait il ne s'intéressait guère aux systèmes d'exploitation[16]. A la fin des années 1970 et au début des années 1980, il passa plusieurs été avec le groupe de chercheurs travaillant sur Unix.

Il devint professeur au département informatique de l'Université Libre d'Amsterdam, son enseignement concerne principalement les systèmes d'exploitation (Il a d'ailleurs écrit de nombreux livres à ce sujet). A l'époque il utilisait pour ses cours un livre de John Lions sur Unix Time-Sharing System 6, cependant AT&T interdit l'utilisation des sources d'Unix pour les cours, c'est pourquoi il se mit à créer son propre Unix, entièrement programmé par lui, et donc indépendant d'AT&T, MINIX. Le code fut publié en 1987. Il fit une seconde édition, compatible POSIX, avec Al Woodhull. MINIX est un micro-kernel simple, documenté et orienté pour l'enseignement, c'est pourquoi A. Tanenbaum refusa d'ajouter certaines fonctionnalités à son système.

Minix est toujours en développement avec la branche 3 sortie le 24 octobre 2005.

9.2 Linux

En 1991, Linus Benedict Torvalds était étudiant à l'Université d'Helsinki (Finlande). Son université utilisait un Unix et il voulait avoir le même système chez lui, c'est pourquoi il était utilisateur de MINIX. Il décida de commencer son propre système, en tant que loisir, il cherchait à rajouter des fonctionnalités par rapport à MINIX[17]. La première version de Linux sorti le 17 septembre 1991. Cependant Linux était un noyau monolithique (Contrairement à MINIX).

Parallèlement un projet visant à créer un système libre, GNU (*GNU is Not Unix*), était en développement, il ne manquait plus que le noyau, le leur n'étant pas fini. Les membres du projet GNU décidèrent donc d'utiliser Linux comme noyau dès 1992.

Références

- [1] *Bell Labs' Top Innovations*
<http://www.bell-labs.com/about/history/timeline.html> (Anglais)
- [2] *Milestones in AT&T History*
<http://www.att.com/history/milestones.html> (Anglais)
- [3] *The Evolution of the Unix Time-sharing System* par D. M. Ritchie (1979-1980)
<http://cm.bell-labs.com/cm/cs/who/dmr/hist.html> (Anglais)
- [4] *The Unix Time-Sharing System* par D. M. Ritchie et K. Thompson (1974)
<http://cm.bell-labs.com/cm/cs/who/dmr/cacm.html> (Anglais)
- [5] *Users' Reference to B* par Ken Thompson
<http://cm.bell-labs.com/cm/cs/who/dmr/kbman.html> (Anglais)
- [6] Page personnelle de Martin Richards
<http://www.cl.cam.ac.uk/users/mr/> (Anglais)
- [7] *The Development of the C Language* par Dennis M. Ritchie (1993)
<http://cm.bell-labs.com/cm/cs/who/dmr/chist.html> (Anglais)
- [8] Interview de Brian Kernighan
<http://www.cs.cmu.edu/~mihaib/kernighan-interview/> (Anglais)
- [9] *Threaded Code* par Anton Ertl
<http://www.complang.tuwien.ac.at/forth/threaded-code.html> (Anglais)
- [10] *The C Programming Language* (1979) par B. W. Kernighan and D. M. Ritchie
<http://cm.bell-labs.com/cm/cs/cbook/> (Anglais)
ISBN 0-13-110362-8 (paperback), 0-13-110370-9 (hardback).
- [11] *Twenty Years of Berkeley Unix* (1999) par Marshall Kirk McKusick
<http://www.oreilly.com/catalog/opensources/book/kirkmck.html> (Anglais)
- [12] *Plan 9 from Bell Labs*
<http://www.cs.bell-labs.com/plan9dist/> (Anglais)
- [13] *Vita Nuova*
<http://www.vitanuova.com/> (Anglais)
- [14] *The Unix wars*
<http://www.bell-labs.com/history/Unix/wars.html> (Anglais)
- [15] Minix
<http://www.cs.vu.nl/~ast/minix.html> (Anglais)
- [16] *Who wrote Linux*
<http://www.cs.vu.nl/~ast/brown/> (Anglais)
- [17] *What would you like to see most in minix? (1991) L. Torvalds*
<http://groups.google.com/group/comp.os.minix/browse.thread/thread/76536d1fb451ac60/0b46287f469f6ef5> (Anglais)