

# Histoire d'Unix

David du Colombier      Jean-Baptiste Campesato

9 juillet 2008

# Table des matières

<b>1</b>	<b>La genèse</b>	<b>2</b>
1.1	Bell Labs . . . . .	2
1.2	Multics . . . . .	3
1.3	Ken's New System . . . . .	3
1.4	Le langage B . . . . .	4
1.5	La naissance d'Unix Time-Sharing System . . . . .	4
1.6	Le langage C . . . . .	6
1.7	L'évolution d'Unix Time-Sharing System . . . . .	6
<b>2</b>	<b>L'expansion</b>	<b>8</b>
2.1	Berkeley Software Distribution . . . . .	8
2.1.1	1BSD et 2BSD . . . . .	8
2.1.2	L'arrivée de l'adressage sur 32 bits . . . . .	9
2.1.3	3BSD et 4BSD . . . . .	9
2.1.4	4BSD à 4.2BSD . . . . .	10
2.1.5	4.3BSD . . . . .	11
2.1.6	BSD Networking Release . . . . .	12
2.1.7	Le procès : USL vs. BSDI . . . . .	14
2.1.8	4.4BSD . . . . .	14
2.2	L'Unix de AT&T . . . . .	15
2.2.1	Les débuts . . . . .	15
2.2.2	UNIX System V . . . . .	16
2.3	L'Unix de Sun Microsystems . . . . .	17
2.3.1	SunOS . . . . .	17
2.3.2	Solaris . . . . .	18
<b>3</b>	<b>L'ouverture</b>	<b>19</b>
3.1	Le tournant de Bell Labs . . . . .	19
3.1.1	La fin d'Unix Time-Sharing System . . . . .	19
3.1.2	Plan 9 from Bell Labs . . . . .	20
3.1.3	Lucent Technologies . . . . .	21
3.1.4	Inferno . . . . .	21
3.2	« The Unix Wars » . . . . .	22

# Chapitre 1

## La genèse

### 1.1 Bell Labs

Bell Telephone Company fut fondé en 1878 par le beau père d'Alexander Graham Bell, Gardiner Greene Hubbard. Il participa également à la mise en place d'une société fille nommée New England Telephone and Telegraph Company. Ces deux entreprises fusionnèrent en 1879 pour former la National Bell Telephone Company. Cette nouvelle société fusionna à son tour en 1880 avec plusieurs autres pour former American Bell Telephone Company. Elle acquit Western Electric en 1881, une filiale spécialisée dans la conception d'équipement téléphoniques. American Bell Telephone Company devenu finalement AT&T (American Telephone and Telegraph Company) en 1885.

AT&T est depuis plus d'un siècle la plus importante compagnie téléphonique des États-Unis.

En 1925, le président de AT&T, Walter Gifford, fonde Bell Labs (Bell Telephone Laboratories) afin de remplacer les laboratoires de recherche de Western Electric. Le siège de Bell Labs se situe à Murray Hill, dans l'état du New Jersey.

En 1956, par crainte que AT&T dispose d'un monopole trop important, le département de justice des États Unis prit la décision d'interdire à AT&T la commercialisation de services autres que ceux rapport avec les télécommunications. Cette interdiction concerne aussi bien AT&T que ses filiales Western Electric et Bell Canada. Le but visé était d'empêcher AT&T d'étendre son monopole sur le marché informatique naissant.

Bell Labs est devenu l'un des plus grand laboratoire de recherche en sciences physiques. Les chercheurs qui y ont travaillé ont été à l'origine de plusieurs avancées technologiques majeures telles que le premier ordinateur digital en 1939, le transistor en 1947, la fibre optique et l'unité de mesure du son, le bel.

## 1.2 Multics

Au début des années 1960, Bell Labs, General Electric et le MIT (Massachusetts Institute of Technology) entreprirent le développement d'un système d'exploitation expérimental. Ce système dénommé Multics (Multiplexed Information and Computing Service) a été conçu pour fonctionner sur les ordinateurs GE-645 de General Electric. Ken Thompson rejoignit le projet en 1966 juste après l'obtention d'un Master EECS (Electrical Engineering and Computer Science) à l'université de Berkeley. Dennis Ritchie fut impliqué à son tour en 1967, après l'obtention d'un doctorat en sciences physiques à l'université de Harvard. Ils travaillèrent tous les deux au centre de recherche de Bell Labs situé à Murray Hill.

Malgré de nombreux efforts et un investissement financier colossal, Multics ne devint jamais un système d'exploitation correctement utilisable. Il était si peu efficace que trois utilisateurs simultanés pouvaient surcharger la machine. Bell Labs se retira du projet en mars 1969.

## 1.3 Ken's New System

Après l'abandon de Multics, Ken Thompson continua de développer sur l'ordinateur GE-645. Il écrivit un jeu vidéo nommé Space Travel. C'était plus un logiciel de simulation astronomique qu'une véritable application ludique. Cependant, les performances désastreuses de Multics se faisaient ressentir et le fonctionnement de la machine coûtait horriblement cher en consommation électrique.

Il récupéra un ordinateur PDP-7 de DEC (Digital Equipment Corporation) inutilisé dans un recoin de l'un des laboratoires où il travaillait. Il eut l'idée d'utiliser cette machine pour faire fonctionner son jeu vidéo. Cependant, le PDP-7 ne disposait que du système d'exploitation GECOS (General Electric Comprehensive Operating System) qui était relativement peu pratique à utiliser. Ce système très rudimentaire ne permettait l'utilisation de la machine qu'à une seule personne à la fois. Il ne disposait que d'un assembleur et d'un chargeur d'applications, c'est à dire qu'il n'était possible que d'écrire un programme et l'exécuter. Ken Thompson s'inspira de son travail effectué sur Multics pour créer un système d'exploitation simple et efficace en langage assembleur. Sous GECOS, il programma un noyau, un interpréteur de commande et un assembleur. Il conçut le système d'exploitation en un mois seulement, en été 1969, lorsqu'il était en vacances avec sa femme chez ses parents.

Le résultat éveilla d'abord la curiosité de son collègue Dennis Ritchie. Le système de Ken Thompson avait quelques similarités avec Multics, mais également de profondes différences. C'était un système très simple fondé sur une multitude de petits programmes qui fonctionnaient ensemble, à l'opposé des gros systèmes monolithiques tels que Multics. Il offrait également des performances très correctes. Ken Thompson nomma son système quelque chose comme « Ken's New System ».

## 1.4 Le langage B

Sur l'ordinateur PDP-7, tous les programmes étaient écrits en langage assembleur. C'est un langage de bas niveau, proche de la machine, et peu agréable à utiliser. De plus, il est dépendant de l'architecture du processeur et non portable. C'est à dire que si l'on souhaitait faire fonctionner un programme existant sur une machine différente, il devait être réécrit entièrement. Il est rapidement devenu nécessaire d'avoir recours à des langages de plus haut niveau, indépendant du matériel. En 1970, Doug McIlroy réalisa une implémentation du langage TMG pour PDP-7. TMG est un système d'écriture de compilateur conçu par Bob McClure. Ken Thompson n'était pas satisfait des fonctionnalités offertes par TMG et préférait un compilateur pour le langage FORTRAN. Finalement, il réalisa le langage B, un dérivé simplifié du langage BCPL, conçu par Martin Richards. Sur le premier système d'exploitation de Ken, B était un interpréteur et ne produisait pas de code machine. L'interpréteur B a d'abord été écrit en langage TMG puis réécrit en langage B. Ce fut le premier langage de haut niveau utilisé sur le système d'exploitation de Ken Thompson, plusieurs applications furent écrites avec ce langage.

## 1.5 La naissance d'Unix Time-Sharing System

En 1970, Joseph Ossanna suggéra l'achat d'un ordinateur DEC PDP-11/20 pour la réalisation d'un projet de traitement de texte. La machine intéressa Dennis Ritchie qui fit une demande à ses supérieurs pour l'obtenir. Max Mathews, le directeur du département de recherche acoustique, apporta les fonds nécessaires. L'administration de Bell Labs comprenait que le traitement de texte était quelque chose d'utile et l'ordinateur fut commandé.

La machine arriva au laboratoire très rapidement, vers la fin de l'été 1970. La machine livrée ne comportait ni disque dur, ni système d'exploitation. Elle ne disposait que d'un lecteur de bandes perforées. Ken Thompson réécrit intégralement son système en assembleur pour PDP-11/20 depuis son PDP-7. Pour se faire, il utilisait un assembleur PDP-11 écrit en langage B sur le PDP-7. Comme la machine ne disposait d'aucun disque dur, Ken Thompson faisait démarrer son systèmes en deux temps. D'abord il chargeait son système avec une première bande, ensuite il la remplaçait par une autre bande accessible en écriture pour fonctionner de la même manière qu'un disque dur. Une fois le système en place, il fut assez décevant. L'assembleur PDP-11 écrit en B était trop lent et du être réécrit en langage assembleur.

Brian Kernighan proposa de nommer ce nouveau système « Unics », un jeu de mots sur « Multics », qui se transforma ensuite en « Unix ». Le nouveau système fut finalement nommé officiellement « Unix Time-Sharing System » pour insister sur le fait qu'il s'agit d'un système à temps partagé, c'est à dire un système qui peut accueillir plusieurs utilisateurs exécutant plusieurs programmes simultanément.

Quelques années plus tôt, Jerome Saltzer avait écrit le logiciel de traitement de texte `runoff` pour le système d'exploitation CTSS. Bob Morris et Ken Thompson le portèrent sur le système GE-635 en l'appelant `roff`. Cette version était un peu différente et plus

simple que l'originale. Enfin Ken Thompson porta roff sur Unix. Bien que ce ne fut pas le but réel, ce logiciel connut un certain succès. Il fut utilisé d'abord par les départements des brevets de Bell Labs (Bell Labs Patent Department) en partageant le PDP-11/20 avec le laboratoire. Finalement le département des brevets acquit la machine et financèrent l'achat d'un PDP-11/45 au laboratoire de recherche. Le PDP-11/45 est une machine similaire et compatible avec le PDP-11/20, mais plus récente.

L'une des premières spécificités d'Unix était d'offrir à l'utilisateur des commandes simples, dont le fonctionnement était facilement prévisible et assimilable. La seconde spécificité était la présence de petits manuels décrivant chacune des soixante commandes présentes dans le système. Ces différents manuels furent regroupés au sein d'un unique ouvrage en papier, précédés d'une préface. L'*Unix Programmer's Manual* parut le 3 novembre 1971.

Peu à peu, différentes branches de AT&T furent dotées de machines tournant sous Unix. Le système commençait à se répandre à une échelle nationale. Le développement poursuivit, de nouvelles commandes furent ajoutées et les erreurs furent corrigées. Unix devenait progressivement de plus en plus performant et efficace.

Le 12 juin 1972 sortit la deuxième édition du *Unix Programmer's Manual*. La préface indiquait « Finally, the number of Unix installations has grown to 10. ».

Le système d'exploitation Unix évolue en permanence, c'est pourquoi on ne parle pas vraiment de « version », mais d'« édition » d'Unix. Une « édition » représente en fait l'état du système tel qu'il était lorsque son manuel d'utilisation a été publié. C'est à dire que par exemple, la première édition d'Unix est le système tel qu'il est décrit dans la première édition du manuel.

Une fonction majeure fut ajoutée, la notion de « pipe ». Un « pipe » consiste à rediriger la sortie d'une première commande vers l'entrée d'une seconde. Cette notion permet aux programmes de fonctionner ensembles.

En février 1973 sortit la troisième édition du *Unix Programmer's Manual*. La préface indiquait « Finally, the number of Unix installations has grown to 16, with more expected. ».

Doug McIlroy remarqua qu'à partir de ce moment naissait une véritable philosophie propre à Unix :

- écrire des programmes qui font une chose et qui la font bien,
- écrire des programmes qui fonctionnent ensembles,
- écrire des programmes qui utilisent des flux en format texte, parce que c'est une interface universelle.

C'est sur cette philosophie que s'appuyèrent les développements futurs.

## 1.6 Le langage C

Le langage B comportait plusieurs problèmes. D'abord, de même que BCPL, B n'avait qu'un seul type pour les variables, la taille des « word », qui était lié à l'architecture. Ensuite l'arithmétique sur les nombre décimaux n'était possible que parce qu'un « word » était autrefois capable de contenir un décimal, or ce n'était plus le cas avec le PDP-11/20. Il devint nécessaire que le langage dispose de différents types de variables pour séparer les entiers, les décimaux et les caractères. Enfin, le compilateur B donnait des programmes lents à cause de la technique du « threaded code ».

En 1971, Dennis Ritchie dû modifier le B, il ajouta le type « char » pour représenter les caractères et écrivit un compilateur pour PDP-11. Ce langage d'abord nommé NB (New B) fut ensuite appelé C.

Dennis Ritchie apporta des modifications jusqu'en 1973. De 1972 à 1977, Alan Snyder, Steven Johnson, Michael Lesk et Ken Thompson ont contribué au développement de ce langage. Durant cette période le langage fut de plus en plus utilisé (notamment avec le compilateur portable de Johnson) et est en constante croissance.

En 1978, Brian Kernighan et Dennis Ritchie publièrent la première version du très célèbre livre *The C Programming Language*, qui fait toujours aujourd'hui office de référence après des programmeurs. On appelle « K&R » la version de C telle qu'elle est décrite dans ce livre. Une seconde édition de ce livre fut publiée le 22 mars 1988 pour rendre le langage conforme au standard ANSI C (ISO/IEC 9899 :1989).

## 1.7 L'évolution d'Unix Time-Sharing System

En novembre 1973 sortit la quatrième édition du *Unix Programmer's Manual*. Le système fut en partie réécrit en langage C. C'est l'évènement qui aura le plus grand impact dans l'avenir du système car cela signifie désormais que le système pourra être porté sur n'importe quel matériel en quelques mois et que les modifications sont simples.

En juin 1975 sortit la cinquième édition du *Unix Programmer's Manual*. Peu à peu, de plus en plus de composants du système furent réécrits en langage C. Elle comporte 25 158 lignes de code en langage C et 31 413 lignes de code en langage assembleur.

En mai 1975 sortit la sixième édition du *Unix Programmer's Manual*. Ce fut la première édition d'Unix à avoir été disponible publiquement à travers le monde, principalement pour les universités. Le système accompagné de ses sources était envoyé sur une bande magnétique à tous ceux qui en faisaient la demande. Cela permettait à quiconque de modifier le système et de redistribuer ses modifications. C'est ce qui permit le début de l'expansion d'Unix. Cette édition comportait 36 373 lignes de code en langage C et 47 584 lignes de code en langage assembleur.

John Lion écrivit un livre appelé *A commentary on the Unix Operating System* commentant en détail les sources du noyau de cette édition. Ce livre fut un des livres d'infor-

matique les plus lu au monde et eu un grand succès auprès des étudiants.

En janvier 1979 sortit la septième édition du *Unix Programmer's Manual*. Ce fut la dernière édition d'Unix à avoir été réalisée intégralement par l'équipe de Bell Labs. Tout le code C fut remis à jour pour suivre le standard tel qu'il est décrit dans le livre *The C Programming Language*. La quasi intégralité du système est désormais écrite en langage C et le système est apte à être porté facilement. L'interpréteur de commande `sh` (Bourne Shell) de Steve Bourne remplace désormais l'interpréteur de Ken Thompson. Tout comme la sixième édition, cette édition fut largement diffusée sous la forme de plusieurs bandes magnétiques. Elle comportait 147 945 lignes de code en langage C et 18 348 lignes de code en langage assembleur.

Un port de la septième édition sur les plateformes VAX, la nouvelle architecture 32 bits de DEC, sortit en mai 1979 sous le nom Unix 32V. Ce système fut développé par John Reiser et Tom London à Bell Labs. Il ne tirait pas vraiment partie des performances offertes par l'architecture VAX, notamment à cause de l'absence d'un système mémoire virtuelle élaboré. Elle comportait 157 937 lignes de code en langage C et 7 161 lignes de code en langage assembleur.

# Chapitre 2

## L'expansion

### 2.1 Berkeley Software Distribution

#### 2.1.1 1BSD et 2BSD

Courant novembre 1973, Ken Thompson et Dennis Ritchie se sont rendus à l'université de Purdue dans le cadre du Symposium on Operating Systems Principles afin d'y présenter la première spécification d'Unix.

Le professeur Bob Fabry de l'université de Berkeley en Californie, fut toute suite intéressé et souhaita obtenir une copie d'Unix pour Berkeley. Cependant à l'époque l'université n'avait pas d'ordinateur capable de supporter Unix, la première chose à faire était donc d'obtenir un PDP-11/45.

Les départements de mathématiques et d'informatique réunirent leurs moyens afin de pouvoir acheter ce PDP-11/45. En janvier 1974 une version d'Unix Time-Sharing System 4 fut reçue et fut installée par l'étudiant Keith Standifort. Cette installation ne fut pas des plus simples, plusieurs problèmes nécessitèrent l'intervention à distance de Ken Thompson. Ces problèmes étaient dus au fait que la machine des Bell Labs était différente de celle de Berkeley, notamment au niveau du contrôleur de disques.

À l'automne 1975 deux nouveaux étudiants, Bill Joy et Chuck Haley, développèrent rapidement un intérêt pour Unix. Ils commencèrent par créer l'éditeur de texte `ex` puis explorèrent le coeur du noyau vers la fin de l'été 1976.

Ainsi, au début de l'année 1977, Bill Joy fonda la Berkeley Software Distribution. La première distribution BSD, qui n'était qu'un ensemble d'ajouts et de modifications, parut avec un compilateur Pascal et l'éditeur de texte `ex`. Une trentaine de copies furent distribuées.

Berkeley acheta des terminaux ADM-3a (des terminaux avec curseur pour modifier différentes zones de l'écran), ce qui permit à Bill Joy de commencer à développer `termcap` (une librairie offrant la possibilité d'utiliser des curseurs pour redessiner le contenu du terminal) puis l'éditeur de texte `vi`.

Au milieu de l'année 1978, 2BSD sortit avec `vi`, `termcap` (compatible avec de nombreux terminaux) et un compilateur Pascal plus robuste. 2BSD était capable de tourner sur des PDP-11/34.

Le développement de 2BSD perdura jusqu'en avril 2000 avec 2.11BSD patch 431.

### 2.1.2 L'arrivée de l'adressage sur 32 bits

Début 1978, le professeur Richard Fateman commença à chercher une machine ayant un espace d'adressage plus grand afin de pouvoir continuer son travail sur Macsyma (Un logiciel de calcul formel d'abord développé sur un PDP-10). Le nouveau VAX-11/780 regroupait tous les critères nécessaires. Plusieurs départements réunirent leurs fonds afin de pouvoir en obtenir un.

Cette nouvelle machine fonctionna d'abord sous VMS de DEC, mais le département d'informatique étant habitué à l'environnement Unix souhaita continuer à l'utiliser. Fateman obtint rapidement une copie d'Unix 32V de John Reiser et de Tom London de Bell Labs.

Ce système proposait un environnement Unix Time-Sharing System 7 mais ne permettait pas d'utiliser la totalité de la capacité de virtualisation de la mémoire du matériel. L'espace d'adressage virtuel était donc, comme sous Unix 7, limité à la taille de la mémoire physique, soit 1 Mo.

Fateman alla voir le professeur Domenico Ferrari pour tenter de faire développer un système de virtualisation de la mémoire par ses étudiants. Ozalp Babaoglu proposa plusieurs implémentations et alla demander de l'aide à Bill Joy afin de mieux comprendre le noyau d'Unix. En janvier 1979 la majorité des bugs furent réparés.

### 2.1.3 3BSD et 4BSD

Suite à cela Bill Joy commença à porter 2BSD. Peter Kessler et Marshall Kirk McKusick portèrent le compilateur pascal alors que Bill Joy porta `ex`, `vi`, l'interpréteur de commandes `cs`h et les autres petits programmes de 2BSD.

La distribution fut prête fin 1979 sous le nom de 3BSD contenant le noyau utilisant la mémoire virtuelle, les outils standards de Unix 32V et les programmes de 1BSD et de 2BSD. En décembre 1979 presque cent copies furent distribuées.

La DARPA (Defense Advanced Research Projects Agency) avait déjà réussi à relier entre eux tous leurs centres majeurs de recherche. Cependant beaucoup de ces machines étaient en fin de vie, il fallut donc garder les logiciels et données de recherche, tâche complexe en raison de la multitude de machines et systèmes d'exploitation différents. Pour les futures générations d'ordinateurs, choisir un seul type de matériel n'était pas possible, cela serait revenu à dépendre d'un fabricant. Le personnel du DARPA a donc

choisit rapidement Unix en tant que standard en raison de sa grande portabilité.

À l'automne 1979 Bob Fabry répondit à le DARPA : il souhaitait utiliser une pré-version de 3BSD. Ainsi suite à la publication de 3BSD en décembre 1979, et à son succès croissant, il put obtenir un contrat de 18 mois à partir d'avril 1980. Il consistait à ajouter des fonctionnalités nécessaires à le DARPA. Bob Fabry fonda le CSRG (Computer Systems Research Group). Bill Joy se joignit au projet aux environs de mars.

Un système de licence fut mis en place grâce aux efforts coordonnés de Bob Fabry, Bob Guffy (AT&T) et d'avocats de Berkeley afin d'obtenir une diffusion officielle d'Unix.

Plusieurs fonctionnalités furent ajoutées : le système de gestion des processus/tâches de Jim Kulp, un système d'auto-réamorçage, un système de fichiers basé sur des blocs de 1 Ko, le support de la dernière VAX (VAX-11/750). 4BSD vit le jour en octobre 1980 avec le compilateur Pascal, l'interpréteur LISP de Franz, et un système de gestion de mails.

Durant ses neuf mois de vies 150 copies de 4BSD furent distribuées, et grâce à la licence mise au point 500 machines l'utilisèrent.

#### 2.1.4 4BSD à 4.2BSD

En juin 1981 4.1BSD fut publié avec des modifications et de nouvelles fonctionnalités comme le système d'auto-configuration de Robert Elz. En deux ans, 400 copies furent distribuées. Ce système ne fut pas nommé 5BSD car AT&T s'y opposa en pensant que les utilisateurs pouvaient confondre 5BSD et UNIX System V.

Le succès de 4.1BSD satisfait le DARPA qui prolongea le contrat de deux ans.

La DARPA souhaite améliorer BSD, notamment en incluant un système de fichier plus rapide pour suivre l'amélioration matérielle, en élargissant l'espace d'adressage à plusieurs gigaoctets, en permettant un système de communication inter-processus plus souple et en ajoutant une gestion réseau afin de permettre aux nouveaux ordinateurs sous BSD de participer à l'ARPAnet.

Duane Adams mit donc sur pied un groupe connu sous le nom de « steering committee » (Comité de pilotage). Ce comité se réunit 2 fois par ans d'avril 1981 à juin 1983. Il était composé de Bob Fabry, Bill Joy, Sam Leffler (université de Californie), Alan Nemeth, Rob Gurwitz (les deux de BBN (Bolt, Beranek et Newman)), Dennis Ritchie (Bell Labs), Keith Lantz (université de Stanford), Rick Rashid (université de Carnegie-Mellon), Bert Halstead (MIT), Dan Lynch (The Information Sciences Institute), Duane Adams, Bob Baker (les deux du DARPA) et Jerry Popek (université de Californie à Los Angeles). Dès 1984 ces réunions furent ouvertes à des tiers.

Un premier document présentant les fonctionnalités à ajouter fut diffusé en Juillet 1981. Cet été, Marshall Kirk McKusick rejoignit le CSRG pour travailler sur le nouveau

système de fichiers et Bill Joy se mit à travailler sur les IPC (un protocole de communication inter-processus). Et en automne 1981 Sam Leffler se rallia au CSRG comme membre à temps complet afin de travailler avec Bill Joy.

Bob Gurwitz proposa une première implémentation de la pile TCP/IP à Berkeley, Joy l'intégra au système et testa ses performances. Bill Joy et Sam Leffler comprirent que la pile TCP/IP devait s'élargir à d'autres protocoles que ceux du DARPA et la modifièrent afin de permettre l'utilisation de plusieurs protocoles en même temps.

Les IPC et la pile TCP/IP permirent de créer plusieurs programmes simples concernant l'accès à des ressources distantes : `rcp`, `rsh`, `rlogin` et `rwho` qui ne devaient être que temporaires.

4.1aBSD fut distribué en avril 1982 pour une utilisation locale.

Le système de fichier de Marshall Kirk McKusick fut intégré au noyau en juin 1982 créant ainsi 4.1bBSD.

À la fin du printemps 1982 Bill Joy annonça sa décision de rejoindre Sun Microsystems.

En Avril 1983 4.1cBSD vit le jour.

En juin 1983, Bob Fabry laissa le contrôle du CSRG aux professeurs Domenico Ferrari et Sysa Graham afin de prendre une année sabbatique. Leffler continua à compléter le système. Le système de quota de Robert Elz fut intégré, l'installation simplifiée, la documentation revue et la recherche de bugs active.

En août 1983 4.2BSD vit le jour.

Suite à 4.2BSD Sam Leffler quitta Berkeley pour Lucasfilm et fut remplacé par Mike Karels qui fut lui même rejoint en décembre 1984 par Marshall Kirk McKusick.

4.2BSD connut une popularité impressionnante avec plus de 1000 licences diffusées en 18 mois. Les vendeurs se mirent à vendre 4.2BSD à la place d'UNIX System V qui n'avait pas de système de fichier rapide. Cependant après quelques années BSD retourna à ses racines universitaires. Les modifications de BSD furent ajoutées au fur et à mesure à UNIX System V qui reprit une position dominante.

### **2.1.5 4.3BSD**

Des critiques à propos de 4.2BSD virent rapidement le jour, notamment à cause de sa lenteur due au fait que les nouvelles fonctionnalités furent ajoutées sans optimisation. Mike Karels et Marshall Kirk McKusick passa deux années à améliorer le système.

Ils annoncèrent en juin 1985 à la conférence USENIX qu'ils espéraient diffuser 4.3BSD lors de l'été.

BBN stoppa cette diffusion car BSD n'avait pas intégré la version finale de leur code réseau.

Quand Mike Karels fut en possession du code de BBN il décida d'intégrer les bonnes idées du code de BBN dans celui de Berkeley plutôt que de le remplacer alors qu'il comptait de nombreuses améliorations et qu'il avait été bien testé.

Des tests effectués par Mike Muuse considéré comme tiers indépendant montrèrent que le code de Berkeley était mieux adapté.

4.3BSD fut diffusé en juin 1986 faisant taire la plupart des critiques concernant les performances.

4.3BSD fut porté sur PDP-11 par Keith Bostic (nouveau au CSRG) sous le nom de 2.11BSD resté en production jusque 2000, c'était un travail remarquable vu les améliorations au niveau de la mémoire des VAX sur les PDP-11.

Les machines VAX étant en fin de vie le projet de Bill Joy, qui consistait à séparer le code dépendant du matériel et le code indépendant, fut réalisé. Ainsi en juin 1988 4.3BSD-Tahoe vit le jour.

Jusque là chaque personne souhaitant obtenir BSD devait acheter une licence à AT&T vu que BSD fut toujours distribué sous forme de sources. Le prix de la licence AT&T devenant de plus en plus élevé les utilisateurs demandèrent à ce que le code réseau de BSD soit diffusé séparément (il était entièrement l'oeuvre de Berkeley, aucune partie ne provenait de Unix 32V). Le code réseau et les programmes allant avec furent vendus dès juin 1989 pour \$1000 sous le nom de Networking Release 1 et sous une licence très peu restrictive (le code pouvait être revendu, modifié, diffusé si les en-têtes de copyright n'étaient pas modifiées et si la documentation accompagnant le code révélait l'utilisation de BSD).

Parallèlement le développement continua : le système de mémoire virtuelle du système d'exploitation Mach, le système de fichier NFS de Sun furent rajoutés donnant naissance à 4.3BSD-Reno début 1990.

## 2.1.6 BSD Networking Release

Lors d'une réunion hebdomadaire du CSRG Keith Bostic parla de la popularité de la version réseau diffusée librement (Networking, Release 1). Il voulait qu'une version améliorée soit mise au point avec plus de code BSD. Mike Karels et Marshall Kirk McKusick lui rappelèrent qu'il fallait ré-implémenter la bibliothèque C, des utilitaires par centaines et le noyau afin de ne plus avoir de code provenant de AT&T.

Mike et Marshall lui proposèrent donc de s'occuper du noyau s'il s'occupait du reste, ce qui leur semblait infaisable.

Keith Bostic devint donc le pionner du développement fédéré par l'internet : il demanda aux utilisateurs de réécrire les utilitaires en n'utilisant que les descriptions publiées, avec pour seule compensation la présence de leur nom dans le système en tant que contributeur BSD.

Malgré un démarrage lent, en 18 mois presque tous les utilitaires furent réécrits.

Karels et Marshall furent donc résignés à travailler avec Keith sur le noyau pour supprimer tout code provenant de AT&T.

Après quelques mois il ne restait que six fichiers, qui semblaient difficilement réécritables, ils décidèrent donc de diffuser ce qu'ils avaient pour l'instant. Ils obtinrent la permission de l'administration une fois l'enquête sur la propriété du code fini.

Ils voulurent premièrement nommer différemment cette version, cependant pour ne pas avoir à refaire écrire une licence par les avocats de l'université, ils décidèrent de la nommer Networking Release 2 en n'effectuant juste une modification de la licence de la Networking Release 1.

La diffusion commença en Juin 1991 avec les mêmes conditions et le même prix que la première fois.

Moins de six mois plus tard, Bill Jolitz remplaça les six fichiers restants contenant du code AT&T. Il diffusa un système compilé amorçable pour l'architecture PC sous le nom 386/BSD via un serveur FTP anonyme.

À cause de son travail, Bill Jolitz ne put gérer les corrections d'erreurs et les améliorations de 386/BSD. Ainsi quelques mois plus tard un groupe d'utilisateurs avides de 386/BSD formèrent fin 1993 le groupe NetBSD dans le but de rassembler leurs travaux et leurs ressources. Les diffusions prirent le nom de NetBSD distribution. Ils décidèrent de porter le système sur le plus d'architectures possibles et continuèrent le développement orienté recherche du CSRG.

Jusqu'en 1998 leur distribution n'était disponible que sur Internet, sans aucun support physique.

Peu de mois après la création du groupe NetBSD, le groupe FreeBSD vit le jour. L'objectif était de ne s'occuper que des architectures PC et d'être accessible au plus de personnes.

En 1995, Theo de Raadt, l'un des fondateurs de NetBSD, entra en désaccord avec le reste de l'équipe. Il donne alors naissance à un projet dérivé qu'il nommera OpenBSD.

## 2.1.7 Le procès : USL vs. BSDI

En plus des projets libres basés sur Networking Release 2 décrits ci-dessus, une société, BSDI (Berkeley Software Design, Incorporated) fut formé dans le but de développer et de distribuer une version commerciale du code. Comme les autres groupes, ils ajoutèrent les six fichiers de Bill Jolitz. En janvier 1992 ils vendaient un paquetage comprenant les sources et les binaires pour \$995. Des publicités demandant d'appeler le numéro 1-800-ITS-Unix et montrant que le système de BSDI permettait d'économiser jusque 99% face à UNIX System V furent diffusées.

USL (Unix System Laboratories), une entreprise provenant de la fission d'AT&T, leur expédia rapidement une lettre demandant à ce que BSDI arrête de présenter leurs produits en tant qu'Unix, notamment avec le numéro de téléphone.

BSDI accepta et toutes les modifications demandées furent appliquées. Cependant USL porta quand même plainte face à BSDI en indiquant que le produit contenait du code USL propriétaire (et touchait donc leurs secrets de construction).

BSDI se défendit en soutenant qu'ils utilisaient un code librement diffusé par l'université de Californie et six fichiers de Bill Jolitz. Le juge demanda donc à USL de reformuler la plainte afin de ne considérer que les six fichiers (BSDI n'était pas responsable de la diffusion des sources de Berkeley), ce qui aurait rendu difficile la poursuite plainte.

USL ne lâcha pas prise en portant plainte une seconde fois mais contre BSDI et l'université de Californie aussi afin d'interdire la vente des produits de BSDI et de la Networking Release 2. Le juge ne retint que deux points et demanda à ce que l'affaire soit traitée dans une cour d'état plutôt que fédérale.

L'université de Californie porta donc plainte contre USL dans leur état. La plainte concernait le fait que USL n'avait pas respecté l'obligation de citer l'université de Californie dans le code BSD utilisé dans UNIX System V.

Peu après, USL fut vendue à Novell par AT&T. Des discussions pour résoudre l'affaire débutèrent pendant l'été 1993. Un arrangement fut conclu en janvier 1994 : trois fichiers de la Networking Release 2 furent supprimés et l'université ajouta les copyrights d'USL dans environ 70 fichiers.

## 2.1.8 4.4BSD

La nouvelle version obtenue fut nommée 4.4BSD-Lite<sup>1</sup> et fut diffusée en juin 1994 sous les mêmes conditions que les distributions Networking. Ainsi la distribution des sources et des binaires était gratuite tant que les copyrights de l'université restaient inchangés et que l'université était citée si quelqu'un réutilisait le code.

---

<sup>1</sup>Lite provenant de l'anglais *lit*, éclairé, et de *light*, allégé.

Une version complète, donc nécessitant la possession d'une licence USL, fut diffusée sous le nom de 4.4BSD-Encumbered.

Les accords passés empêchaient USL de poursuivre une organisation utilisant le code de 4.4BSD-Lite comme base de son système : les groupes BSD (BSDI, NetBSD et FreeBSD) durent prendre 4.4BSD-Lite comme base en y intégrant leurs améliorations. Ceci permit aux groupes de se synchroniser avec le travail du CSRG.

Les bénéfices provenant des 4.4BSD permirent des améliorations et des corrections de bugs. Ces modifications continuèrent pendant deux ans, jusqu'à ce que les rapports de bugs soient très rares.

4.4BSD-Lite, Release 2 sortit en Juin 1995.

Suite à cette version le CSRG fut dissout, après deux décennies à s'occuper de BSD.

## 2.2 L'Unix de AT&T

### 2.2.1 Les débuts

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

Le groupe chargé du développement d'Unix Time-Sharing System, dirigée par Ken Thompson, fut nommé CRG (Computer Research Group). Un second groupe nommé USG (UNIX System Group) fut chargé du support. Un troisième groupe, le PWB (Programmer's WorkBench) fut impliqué dans le développement de technologies liés à Unix et développa son propre système.

Ces trois groupes furent intégrés au sein du USDL (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, appelée CB (Columbus Unix).

AT&T fut dans un premier temps incapable de profiter commercialement du travail effectué par Bell Labs suite à l'interdiction prononcée en 1956 par le département de justice des États Unis.

En 1982, AT&T parvint à obtenir l'accord de commercialiser son système d'exploitation en tant que « outil pour la téléphonie ». Ce nouveau système d'exploitation commercial fut nommé UNIX System III. Ce système développé par l'USG résulte de la fusion des recherches effectués sur Unix 32V, PWB et CB. Il tourne sur plateforme VAX. Cette version comporte 307 479 lignes de code en langage C et 64 953 lignes de code en langage assembleur.

## 2.2.2 UNIX System V

En 1983, AT&T sortit une seconde version majeure de son système d'exploitation commercial, appelée UNIX System V Release 1. Cette version fut développée par l'USDL et était basée sur UNIX System III. Elle inclue plusieurs éléments provenant de BSD comme l'éditeur de texte `vi` et l'interface textuelle `curses`. Elle inclue également les IPC, les sémaphores ainsi que la mémoire partagée.

En 1984 apparaît une première mise à jour, UNIX System V Release 2. Cette dernière apporte entre autre l'interpréteur de commande `sh` issu de Unix Time-Sharing System 7. En même temps est publié le SVID 1 (System V Interface Definition), un 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 UNIX System V.

En 1986, apparaît une deuxième mise à jour, UNIX System V Release 3. Cette dernière profite notamment d'éléments provenant d'Unix Time-Sharing System 8, telles que les streams et RFS. Elle inclue également TLI (Transport Level Interface), une API destinée à accéder à la couche transport d'un réseau, l'appel système, les bibliothèques partagées, ainsi que la pagination de la mémoire sur demande, pour le matériel qui le supportait. Parallèlement est publié le SVID 2.

Cette mise à jour est suivie par UNIX System V Release 3.2. Cette dernière qui apporte les fonctions réseaux, ainsi que l'incorporation de quelques fonctionnalités appartenant au système d'exploitation XENIX (développé par Microsoft puis revendu à SCO (Santa Cruz Operation)).

En 1988, voit le jour la plus importante avancée avec l'arrivée de System V Release 4. Cette mise à jour majeure résulte de la fusion de BSD, SunOS et UNIX System V Release 3. Provenant de BSD, on peut citer le système de fichier FFS (Fast File System), le protocole de communication réseau TCP/IP, les sockets et l'interpréteur de commande `csh` (C Shell) de Bill Joy. Provenant de SunOS, il y a le système de fichier réseau NFS, l'environnement graphique OpenWindows et sa bibliothèque OpenLook et le serveur graphique NeWS. Cette mise à jour apporte également l'interpréteur de commande `ksh` (Korn Shell) de David Korn, le standard ANSI du langage C, l'internationalisation et le support des standards POSIX et X/Open.

Trois mises à jour bien plus mineures ont suivi. UNIX System V Release 4.1 apporte les entrées/sorties asynchrones. UNIX System V Release 4.2, sortie en 1992, apporte le support du système de fichier Veritas FS, la gestion des droits ACL décrite dans POSIX et les modules noyaux chargés dynamiquement. UNIX System V Release 4.2MP apporte le support du multiprocesseur.

UNIX System V Release 4.2 fut appelé UnixWare 1.0 suite à son rachat par Novell.

## 2.3 L'Unix de Sun Microsystems

### 2.3.1 SunOS

En 1981, dans le cadre du projet Stanford University Network, financé principalement par le département informatique de Stanford, Andy Bechtolsheim mit au point l'une des premières stations de travail Unix. Andy Bechtolsheim avait comme objectif initial de réaliser la machine cliente permettant de connecter au nouveau réseau informatique de Stanford. La machine réalisée était composée d'un processeur Motorola 68000 à 8 MHz, 1,2 Mo de RAM et d'un contrôleur de disque dur ST-506, une carte réseau et une puce graphique dotée de 1 Mo de mémoire et capable d'afficher une résolution de 1024x768 pixels en 256 couleurs. Elle tournait sous un port de la septième édition d'Unix pour Motorola 68000 réalisée par la société Unisoft.

Suite à l'énorme succès de son ordinateur, Andy Bechtolsheim, accompagné de Scott McNealy et Vinod Khosla décidèrent de fonder une entreprise de fabrication de stations de travail sous Unix. Scott McNealy prit la direction du projet, Vinod Khosla s'occupa du financement. Andy Bechtolsheim demanda l'aide Bill Joy de Berkeley pour créer un système d'exploitation pour leurs machines. Bill Joy quitta alors le projet BSD. Finalement, Andy Bechtolsheim, Vinod Khosla, Scott McNealy et Bill Joy fondèrent Sun Microsystems en février 1982.

En 1982, la première machine commercialisée par Sun Microsystems, nommée Sun-1 fut assez similaire à la machine fabriquée par Andy Bechtolsheim à Stanford. Elle tournait également sous Unisoft Unix 7.

En 1983, Sun Microsystems commercialisa une machine Sun-1 améliorée, dotée d'une nouvelle carte mère et d'un processeur Motorola 68010. Une nouvelle machine similaire nommée Sun-2 fut également lancée. Ces machines étaient livrées avec un système d'exploitation nommé SunOS 1.0, un port de 4.1BSD sur processeur Motorola, réalisé par Bill Joy. Il ajoute entre autres l'environnement graphique SunView. De nouvelles machines furent par la suite commercialisées et Bill Joy améliora son système d'exploitation.

En mai 1985, SunOS 2.0 vit le jour. Il est désormais basé sur le récent 4.2BSD. SunOS 2.0 apporta des améliorations majeures telles que VFS (Virtual File System) et NFS (Network File System). VFS est un protocole qui permet à différents systèmes de fichier d'être utilisés simultanément sur une machine. NFS est un protocole qui permet d'accéder à un système de fichier à travers le réseau, permettant l'échange et le partage de fichiers entre plusieurs machines.

En février 1986, SunOS 3.0 vit le jour. Il inclut des fonctionnalités provenant de UNIX System V telles que IPC.

En décembre 1988, SunOS 4.0 vit le jour. Il est désormais basé sur 4.3BSD. Il inclut à nouveau des fonctionnalités issues de UNIX System V, notamment stream. SunOS 4.0 apporte des fonctionnalités majeures telles qu'un nouveau système de mémoire virtuelle,

permettant d'éviter les contraintes liés à la RAM, ainsi que les liens dynamiques, permettant aux binaires de prendre beaucoup moins de place. C'est la première version à intégrer l'environnement graphique OpenWindows combinée au serveur graphique vectoriel NeWS.

À partir de 1989, l'architecture Motorola est peu à peu remplacée par l'architecture SPARC développée par Sun Microsystems à partir des travaux effectués sur les processeurs RISC à Berkeley.

En Mars 1990, SunOS 4.1 fut la première version à supporter l'architecture SPARC. Le support du multiprocesseur fut ajouté par la suite.

### 2.3.2 Solaris

Suite à leur partenariat avec AT&T, Sun Microsystems décida d'abandonner leur système d'exploitation SunOS basé sur BSD pour le remplacer par le système d'exploitation Solaris 2, basé sur UNIX System V Release 4.

SunOS 4.1.x fut renommé rétrospectivement Solaris 1.x.

Solaris 2.0 vit le jour en juillet 1992. Les premières versions de Solaris 2 étaient lentes, instables et ne disposait pas dans un premier temps de support du multiprocesseur. SunOS continua à être développé pour cette raison. La dernière version de SunOS, SunOS 4.1.4 (aussi appelée Solaris 1.1.2) vit le jour en novembre 1974.

En novembre 1993, Solaris 2.3 marqua le remplacement de son propre serveur graphique NeWS par son concurrent, X11 du MIT. Cela constitue en quelque sorte un pas en arrière en ce qui concerne la technologie, mais un pas en avant en ce qui concerne la standardisation. X11 est en effet devenu standard sur la totalité des Unix propriétaires ou non. Solaris 2.3 fut la première version de Solaris 2 à avoir atteint une qualité suffisante pour être commercialisée à grand échelle.

En novembre 1995, Solaris 2.5 vit le jour. L'environnement graphique standard CDE fut fourni pour la première fois en complément à OpenWindows.

À partir de Solaris 2.7, le numéro de version « 2. » fut abandonné et le système s'appela Solaris 7 qui sortit en octobre 1998. Ce fut la première version à disposer de programmes compilés en 64 bits pour leur nouvelle architecture SPARC V9.

OpenWindows fut retiré définitivement du système en juin 2002 à la sortie de Solaris 9. L'environnement graphique libre GNOME, issu de Linux, fut ajouté en complément de CDE à partir de Solaris 10 en janvier 2005.

Depuis juin 2005, une partie des sources du système d'exploitation Solaris a été rendu disponible publiquement sous une licence libre appelée CDDL (Common Development and Distribution License).

# Chapitre 3

## L'ouverture

### 3.1 Le tournant de Bell Labs

#### 3.1.1 La fin d'Unix Time-Sharing System

En février 1985 voit le jour la huitième édition de Unix Time-Sharing System. Développée à partir de 4.1cBSD de 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 Berkeley. Cette édition fut la première à disposer du protocole TCP/IP de Berkeley. Un mécanisme d'entrée/sortie très efficace baptisé stream, ainsi qu'un système de communication appelé IPC furent développés. Plusieurs éléments 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. Une deuxième bande magnétique contenait les logiciels de Rob Pike pour le support du terminal graphique Blit, dont notamment un gestionnaire de fenêtres et un éditeur de texte. La neuvième édition parue en septembre 1986 et la dixième en octobre 1989 ne représentaient en réalité que l'édition des manuels, le système étant développée continuellement ne fut jamais redistribué en dehors de Bell Labs. 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.

### 3.1.2 Plan 9 from Bell Labs

Vers le milieu des années quatre vingt, les traditionnels gros serveurs laissaient peu à peu place à un nouveau genre d'ordinateurs : les stations de travail. Désormais, il ne s'agit plus d'une grosse machine sur lequel se connectent plusieurs utilisateurs à partir de terminaux, mais de petits ordinateurs individuels qui partagent leurs ressources à travers le réseau.

Unix Time-Sharing System était un système inadapté à ce nouveau genre d'utilisation. En 1987, un nouveau système d'exploitation est développé. Ken Thompson, Rob Pike, Dave Presotto et Phil Winterbottom sont à l'origine du projet. Chacun s'occupe d'une partie spécifique du système. Ils furent ensuite rapidement rejoint par Tom Killian, Jim McKie, Tom Duff, Howard Trickey, Andrew Hume, Bob Flandrena, Alan Berenbaum, Lorinda Cherry, Bill Cheswick, Sean Dorward, David Gay, Paul Glick, Eric Grosse, John Hobby, Gerard Holzmann, Brian Kernighan, Bart Locanthi, Doug McIlroy, Judy Paone, Sean Quinlan, Bob Restrick, Dennis Ritchie, Bjarne Stroustrup et Cliff Young. Après la seconde édition, Russ Cox, David Hogan, Bruce Ellis, Charles Forsyth, Eric Van Hensbergen et Tad Hunt ont également rejoint le projet. La plupart de ces chercheurs ont auparavant travaillé sur Unix Time-Sharing System. Plan 9 est né dans l'objectif de corriger les défauts de base d'Unix, qui se révélaient trop profonds pour être corrigés. De plus, AT&T avait décidé de commercialiser *UNIX System V*, un système d'exploitation commercial basé sur Unix Time-Sharing System, et rendait donc impossible la réalisation de changements majeurs. L'idée de Plan 9 était de recréer un système similaire, mais en partant de rien et en faisant usage du matériel récent.

Les principaux problèmes d'Unix étaient l'absence de couche réseau élaborée et l'absence d'interface graphique native. Bien que ces éléments furent apportés par la suite avec la création du protocole TCP/IP par Bill Joy de l'université de Berkeley et l'élaboration du serveur graphique X11 par les étudiants du MIT, ils furent très complexes et très mal intégrés dans les dérivés d'Unix qui sont toujours utilisés aujourd'hui. Par exemple, X11 est complètement indépendant du système, il utilise sa propre gestion de la souris, du clavier, de l'affichage, des polices, et de nombreux éléments qui sont pourtant déjà supportés par le noyau du système. Plan 9 a été conçu dès l'origine en tant que système d'exploitation distribué, c'est à dire qu'il peut fonctionner sur une multitude de machines hétérogènes, dispersées physiquement, et reliés en réseau, de manière complètement transparente pour l'utilisateur. Chose qui n'est pas faisable avec les systèmes d'exploitation couramment utilisés aujourd'hui. Dans Plan 9, tout est « système de fichier », c'est à dire que toute ressource, qu'elle soit locale ou distante, est considérée comme un système de fichier, manipulable par des appels système simple : « read » et « write ». Grâce au protocole 9P, le système accède de manière complètement transparente aux ressources, de quelle soit matérielle ou logicielle. Plan 9 était initialement écrit dans le langage de programmation Alef, inventé par Rob Pike. Cependant, il était trop complexe de maintenir le système avec les évolutions du langage et Plan 9 fut réécrit dans un dialecte issu du langage ISO C89. Plan 9 va réellement à l'encontre du modèle de développement de la plupart de systèmes de plus en plus complexes d'aujourd'hui, en proposant des réponses simples aux problèmes de développement. Le système d'exploitation Plan 9 est composé

dans son ensemble d'environ 1 500 000 lignes de code. Linux qui n'est qu'un noyau, totalise à lui seul plus de 4 600 000 lignes de code, le noyau de Plan 9 en comptant à peine 170 000. Ceci montre que l'extrême complexité des systèmes actuels peut être évitée.

Plusieurs systèmes d'exploitation, dont principalement 4.4BSD (ainsi que ses successeurs) et Linux, ont repris des éléments de Plan 9. D'abord, le système de fichier procfs qui permet d'associer chaque ressource d'un ordinateur à un fichier situé dans le répertoire `/proc`, ce qui permet de manipuler avec aisance les ressources et obtenir des informations sur les statuts facilement. Ensuite, l'appel système `rfork` qui permet de créer de nouveaux processus, ainsi que de manipuler des processus en cours d'exécution. Enfin, depuis le noyau 2.6.13, Linux supporte nativement le système de fichier 9fs, permettant d'une part l'interaction avec les systèmes Plan 9 et d'autre par d'être utilisé en tant d'alternative pour le système de fichier réseau NFS de Sun Microsystems ou tout autre système de fichier local.

Plan 9 est le premier système d'exploitation qui, en 1992, fut doté du support complet de l'encodage de caractère UTF-8. Ce dernier a été créé lors d'un dîner un soir de septembre 1992 par Rob Pike et Ken Thompson, non satisfait de l'encodage standard d'Unicode, l'ISO 10646. Aussi appelé UTF-16, cet encodage de caractère avait la particularité de coder tous les caractères sur 16-bit, ce qui se réveillait souvent être un gâchis d'espace mémoire car l'encodage ASCII contenant les caractères latin, utilisé par de nombreuses langues, tient sur 7 bits. L'UTF-8 permet d'encoder chaque caractère sur 7 à 16 bits, selon la taille requise pour le caractère, ce qui permet un important gain de place par rapport à UTF-16. L'UTF-8 a depuis été standardisé dans l'annexe R de la norme ISO 10646. L'UTF-8 est aujourd'hui présent dans de nombreuses distributions Linux, ainsi que quelques systèmes comme Solaris. L'UTF-16, malgré ses défauts, est utilisé notamment dans le langage Java, ainsi qu'en interne dans Windows. L'encodage Windows-1252 étant utilisé pour les noms de fichiers et l'encodage du texte. Mac OS utilise quant à lui un encodage Unicode propriétaire et non compatible avec UTF-8 et UTF-16.

### 3.1.3 Lucent Technologies

En 1995, Bell Labs décidèrent de se séparer de AT&T afin de fonder une entité indépendante. La section commerciale de Bell Labs prit le nom de Lucent Technologies et devint le principal fournisseur d'équipements de télécommunications aux États Unis.

En mars 2006, Lucent Technologie décida de fusionner avec son homologue français Alcatel. L'accord est finalisé en décembre 2006 et la société fut renommée Alcatel-Lucent.

### 3.1.4 Inferno

En 1996, Rob Pike et David Presotto décidèrent de créer un nouveau système d'exploitation développé parallèlement à Plan 9. Le nom « Inferno » provient du nom du premier volume de *La Divine Comédie*, écrit par Dante Alighieri, que Rob Pike était en

train de lire lors de la création du système.

Inferno a pu voir le jour car Lucent Technologies pensait pouvoir le commercialiser. Bien qu'il ait été vendu installé sur quelques téléphones et routeurs réseau, Inferno fut un échec commercial pour Lucent Technologies. En 2000, il fut finalement décidé de revendre Inferno. Vita Nuova Holdings Limited, une récente société anglaise basée sur le campus de York et spécialisée dans la vente de services pour Plan 9 et Inferno acquerra ce dernier pour un million de dollars. Inferno fut dès lors distribué gratuitement et sous une licence d'utilisation libre.

Inferno est un système d'exploitation distribué similaire à Plan 9, à la différence qu'il est programmé dans le langage Limbo. C'est un langage pseudo compilé, qui, de façon similaire à Java, permet aux applications d'être exécutés dans une machine virtuelle, indépendamment de l'architecture de la machine utilisée. Inferno peut soit être hébergé sur un autre système d'exploitation (FreeBSD, Irix, Linux, Mac OS X, Plan 9, Solaris ou Windows NT), soit être installé nativement sur une machine (PC ou PDA). Inferno est en ce sens à Limbo ce que JavaOS était à Java. Limbo est le successeur du langage Alef, qui lui-même, succède au langage Newsqueak. Ces trois langages ont été conçus par Rob Pike en tant que successeur au langage C dont ils héritent une syntaxe similaire. Inferno est aujourd'hui un des rares systèmes d'exploitation offrant la possibilité de fonctionner sur une machine disposant d'un mégaoctet d'espace disque, d'un mégaoctet de mémoire vive et un processeur dépourvu de MMU (Memory Management Unit).

## 3.2 « The Unix Wars »

Au cours des années 1980, plusieurs importants constructeurs informatiques ont acquis des licences d'UNIX System V chez AT&T pour réaliser leurs propres systèmes d'exploitation dérivés. Par exemple HP (Hewlett-Packard) développa HP-UX, IBM développa AIX et Silicon Graphics développa IRIX. Tous ces systèmes d'exploitation étaient directement dérivés de System V, mais tendaient à devenir incompatibles entre eux. Sun Microsystems développa quand à lui SunOS, un système basé sur BSD.

En 1984, les constructeurs européens Bull, ICL, Siemens, Olivetti et Nixdorf fondèrent X/Open, un consortium destiné à promouvoir l'élaboration d'un standard Unix en commun. Ils réalisèrent XPG (*X/Open Portability Guide*), un groupement de manuels visant à décrire une structure à suivre par les systèmes qui permet la réalisation d'applications tournant sur plusieurs systèmes d'exploitation. L'Issue 1 fut publiée en 1984, elle traitait de la base des interfaces système. L'Issue 2 fut publiée en 1987 et ajoutait l'internationalisation, les interfaces des terminaux, les communications inter-processus et différents langages de programmation qui proviennent de standards internationaux. L'Issue 3 fut publiée en 1988 pour converger vers POSIX (Portable Operating System Interface). POSIX est le nom donné au standard IEEE Std 1003.1-1998, un standard similaire à XPG publiée par l'IEEE (Institute of Electrical and Electronics Engineers) en 1998.

En 1987, AT&T et Sun Microsystems décidèrent de s'unir pour la réalisation d'un système en commun. Ils fondèrent l'association UI (Unix International) en 1988. Leur objectif était de réaliser un système d'exploitation disposant des meilleures fonctionnalités issues de BSD et UNIX System V. Le fruit de cet effort de collaboration se retrouve dans UNIX System V Release 4.

Bien que cette initiative fut accueillie positivement par les consommateurs, les constructeurs concurrents voyaient dans cette alliance un avantage certain pour Sun Microsystems. DEC, IBM, HP et d'autres se regroupèrent au sein de l'OSF (Open Software Foundation). Ils entreprirent le développement d'un système d'exploitation en commun basé sur BSD. DEC travailla sur une première version nommée OSF/1 qui sortit en janvier 1991. Elle tournait sur les ordinateurs de DEC disposant d'un processeur Alpha. Ce système était le premier système d'exploitation à disposer du micro-noyau Mach développé à l'université de Carnegie-Mellon. HP a tenté de porter OSF/1 sur ses machines mais le projet fut abandonné.

En 1991, AT&T décide de revendre la propriété de UNIX Systems Laboratories (anciennement UNIX System Development Lab, puis UNIX Systems Operations), c'est à dire la propriété d'Unix Time-Sharing System et UNIX System V. En 1992, Novell fit une offre qui fut conclue le 16 juin 1993. Novell transféra ensuite la marque « UNIX » à X/Open, ainsi qu'une partie des droits sur UNIX System V à SCO.

En mars 1993, UI et l'OSF formèrent l'alliance COSE (Common Open Software Environment) pour mettre fin aux tensions entre les deux groupes. Finalement X/Open et l'OSF fusionnèrent en 1996 pour former The Open Group.

Aujourd'hui, The Open Group propose en partenariat avec IEEE le standard « Single Unix Specification » qui rassemble la plupart des standards qui furent publiés. Solaris et AIX sont les seuls systèmes actuellement certifiés comme conforme à la version actuelle de ce standard mais la plupart des Unix propriétaires ont été certifiés pour des versions plus anciennes. Les systèmes libres tels que Linux, DragonFlyBSD, FreeBSD, OpenBSD ou NetBSD s'en inspirent fortement.

Ce document est diffusé conformément à la licence « Creative Commons : Paternité - Pas d'Utilisation Commerciale - Pas de Modification 3.0 ».

Explications : <http://creativecommons.org/licenses/by-nc-nd/3.0/>

Contrat : <http://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>