

# Die Geschichte von UNIX - vom Mainframe zum iMac

Andreas Krennmair

a.krennmair@aon.at

<http://synflood.at/linuxwochen/>

## 1. Vor Unix

Ende der 50er, Anfang der 60er Jahre waren sog. Stapelverarbeitungssysteme üblich, z.B. die *IBM 7094*. Es konnte nur ein Programm gleichzeitig laufen, und es wurden zwei zusätzliche Computer benötigt, um die Ein- und Ausgabe entsprechend durchzuführen. Bei der *IBM 7094* wurde eine *IBM 1401* verwendet, um den Inhalt von Lochkarten auf ein Eingabeband zu schreiben, welches in einen Bandler der *IBM 7094* eingelegt wurde. Dann wurde die Verarbeitung gestartet, die Ausgabe wurde auf ein Ausgabeband geschrieben, die dann mit Hilfe einer zweiten *IBM 1401* ausgedruckt wurde. Dieses Verfahren war sehr aufwändig, teuer und ineffizient, da bei I/O Rechenzeit verschwendet wurde. 1961 wurde das erste Time-sharing-System für eine modifizierte *IBM 7094* mit 64 KByte RAM entwickelt (CTSS - Compatible Time Sharing System). Bis zu 30 Benutzer konnten über serielle Terminals, auch verbunden über Modems, auf einem solchem Rechner arbeiten.

1965 entschied sich Bell Labs, eine General Electric GE-645 zu kaufen, und gemeinsam mit General Electric und dem MIT in Boston ein Timesharing-System mit damals neuartigen Konzepten zu entwickeln. Ein fataler Fehler, wie sich später herausstellte, da GE und das MIT Rechner vom Typ GE-635 einsetzten, und so die GE-645 eine GE-635 emulieren mußte. Als Entwicklungssprache für dieses Betriebssystem, genannt Multics (Multiplexed Information and Computing Service), wurde erstmals eine Hochsprache gewählt, nämlich *PL/I*. Die Entwicklung wurde unter CTSS durchgeführt. Bis der *PL/I*-Compiler verfügbar war, schrieb das Entwicklungsteam die ca. 3000-seitige Spezifikation für Multics. Der Compiler verzögerte sich allerdings, und eine abgespeckte *PL/I*-Version (PLM) wurde stattdessen eingesetzt. Der Compiler war allerdings extrem langsam. Und auch das System selbst war unbenutzbar, da es noch zu instabil und langsam war. 1968/69 kam das Projekt in finanzielle Schwierigkeiten, und es wurde öfters angedroht, das Projekt abzubrechen. Daraufhin stieg Bell Labs aus dem Projekt aus.

## 2. Unix V1 - V7

Etwa 1969 war das Apollo-Raumfahrtprogramm der NASA in vollem Gange. Also suchte sich Ken Thompson, einer der ehemaligen Multics-Entwickler, einen gebrauchten Rechner ( *DEC PDP-7* ), um darauf raumfahrtbezogene Programme zu entwickeln (Orbit-Berechnungen von Satelliten, Mondkalender, Computerspiele). Allerdings war die Programmentwicklung primitiv, daher entschloss er sich, ein eigenes Betriebssystem dafür zu schreiben. Da dieses System im Gegensatz zu Multics nur ein Zweibenutzerbetriebssystem war, bei dem jeder User nur einen Prozess gleichzeitig ausführen konnte, schlug Brian Kernighan vor, es Unics zu nennen. Aufgrund von Begrenzungen bei der Dateinamenlänge von *GECOS*, das als Entwicklungsplattform diente, wurde das System aber *Unix* genannt. Die grundsätzlichen Konzepte von Multics wurden übernommen, z.B. ein Dateisystem mit Baumstruktur. Ken Thompson brachte jedoch auch seine eigenen Ideen ein, die er während der Multics-Entwicklung gehabt hatte, jedoch nicht den Weg in Multics fanden. Dieses System wurde zuerst nur innerhalb der Bell Labs eingesetzt, so z.B. als Dokumentenverarbeitungssystem für die Patentabteilung. Als Software wurde das Dokumentformatierungssystem *roff* verwendet, das heute unter Unix in den Modifikation *troff* und *nroff* immer noch zur Formatierung von Dokumentation und Manpages verwendet wird.

Ursprünglich waren alle Unix-Programme inkl. der Kernel selbst noch in Assembler geschrieben worden. Also entschied sich Ken Thompson, einen *FORTRAN-Compiler* zu schreiben, ohne den seiner Meinung nach ein System wertlos wäre. Nach etwa einer Woche entschied er sich allerdings, eine neue Programmiersprache namens B zu entwickeln, beeinflusst von der Programmiersprache *BCPL* (Basic

Combined Programming Language). In B wurden dann einige Systemtools neu geschrieben. 1971 bekam das Team endlich eine neue PDP-11, eine byte-adressierbare Architektur, während B wortorientiert war. Also entwickelte Dennis Ritchie B zur Programmiersprache C weiter, indem er u.a. Typen hinzufügte. 1973 wurde der Unix Kernel komplett in C neugeschrieben ("Version 4") und war damit ein sehr portables System geworden, was später sehr hilfreich sein sollte. Im Oktober desselben Jahres stellten Ken Thompson und Dennis Ritchie das Betriebssystem Unix, mittlerweile ein Multiuser-Multitasking-Betriebssystem, erstmals der breiten Öffentlichkeit vor, und das Interesse dafür explodierte förmlich. Etwa zur selben Zeit wurde AT&T verdächtigt, monopolistische Bestrebungen zu haben, und es war ihnen deswegen verboten, Handel mit Computer- Hard- und Software zu betreiben. Deswegen wurde Unix kostenfrei allen interessierten Universitäten zur Verfügung gestellt, allerdings ohne jegliche Produktunterstützung.

1975 ging Ken Thompson zurück zur Universität Berkeley, wo er studiert hatte, um eine einjährige Gastprofessur abzuhalten. Dort stellten zwei Studenten, Chuck Haley und Bill Joy, das Pascal-System, das er entwickelt hatte, fertig, und entwickelten etwa gleichzeitig den Texteditor em ("Editor for Mortals" = "Editor für Sterbliche") zum Editor ex weiter. Für ex wurde ein visuelles Terminal-Interface programmiert, das heute noch unter dem Namen "vi" bekannt ist und in verbesserten Versionen wohl der am häufigsten verwendete Unix-Editor ist. Visuelle Editoren waren damals etwas revolutionäres, deshalb musste Bill Joy diese großartige Entwicklung extra mit folgenden Worten umschreiben: "When using vi the screen of your terminal acts as a window into the file which you are editing. Changes which you make to the file are reflected in what you see." Bill Joy und Chuck Haley begannen dann, Unix in einer eigenen Version weiterzuentwickeln. Die "Berkeley System Distribution" (BSD) war geboren.

### 3. Unix auf breiter Front

Als Standard-Betriebssystem konnte sich Unix erst etwa Ende der 70er Jahre durchsetzen. Vorher wurde Arbeit entweder auf IBM Mainframes oder auf diversen PDP-Maschinen mit den Betriebssystemen RSX-11, RT-11, RSTS/E, TOPS-10/20 oder Incompatible Timesharing System (ITS) durchgeführt, Unix galt eher als Spielzeug. 1978 brachte DEC den VAX heraus, als Nachfolger für die veralteten PDP-11-Minicomputer gedacht. Standardmäßig wurden diese Rechner mit dem Betriebssystem VMS ausgeliefert, Unix konnte aber v.a. aufgrund der vorhandenen Kompatibilität zur PDP-11 leicht darauf portiert werden ("32/V Unix"). Vor allem 4.2BSD, das im Jahr 1982 veröffentlicht wurde, war auf den VAX-Maschinen sehr beliebt.

Etwa zur gleichen Zeit kamen Mikrocomputer auf, erschwingliche Rechner für den Büro-Einsatz und Privatpersonen. Die qualitativ hochwertigeren Rechner nannten sich "Workstations" und liefen alle mit einer eigenen Unix-Version. Dadurch wurde der Unix-Markt extrem aufgesplittert, und auf einmal existierten unzählige verschiedene Varianten, die alle zueinander **größtenteils** kompatibel waren. Immer wieder traten jedoch subtile Inkompatibilitäten auf. Manche Systeme waren auf Basis von BSD entwickelt worden, die meisten jedoch verwendeten den System V Quellcode. Microsoft nahm für ihre Unix-Variante Xenix den System-III-Sourcecode her, und lizenzierte Xenix an diverse andere Firmen weiter, z.B. SCO oder Intel.

Dann, Ende der 80er Jahre wurde das Preis-/Leistungsverhältnis der IBM-kompatiblen PCs immer besser. Und alle liefen mit einem primitiven Betriebssystem namens DOS, das schon etwa zur Anfangszeit von Unix als primitiv gegolten hätte. Als grafische Oberfläche war nur MS-Windows, später auch das Betriebssystem OS/2 von IBM und Microsoft verfügbar. Die PCs setzten sich gegenüber den Unix-Rechnern v.a. deswegen durch, weil sich die Unix-Hersteller auf keine einheitlichen Standards bei Programmier-API und grafischer Benutzeroberfläche einigen konnten. Die geplante Unix-GUI, CDE, kam zu spät, war zu langsam und zu teuer, und auch der POSIX-Standard für die Unix-API setzte sich erst langsam durch.

Ein Beispiel für eine Unix-Workstation ist der NeXT Cube, eine Highend- Workstation der Firma NeXT Inc. Next Inc. war die Firma von Steve Jobs, nachdem er Apple verlassen hatte. Technisch waren der NeXT Cube und das dazugehörige Betriebssystem NeXTstep genial: etwa war dies einer der ersten Rechner, in den eigene Grafikbeschleuniger-Karten eingebaut wurden. Das Betriebssystem baute auf dem Mach-Mikrokern auf, kombiniert mit einem BSD-Kernel. Was die anderen Unixe jedoch nicht hatten, war die grafische Oberfläche mit einem sehr hohen Bedienkomfort. Die interne Grafikdarstellung erfolgte über Display Postscript, welches den Einsatz von Postscript-Druckern **sehr** vereinfachte. NeXTstep war auch von

Haus aus voll netzwerkfähig, und kam standardmäßig schon mit einem Mailclient. Desweiteren wurde ein Rechner von NeXT eingesetzt, um das World Wide Web zu entwickeln: der erste Webserver war eine Maschine von NeXT. Rechner von NeXT setzten sich aber aufgrund ihres hohen Preises nie wirklich durch. NeXT Inc. wurde 1996 von Apple aufgekauft.

#### **4. Unix-interne Entwicklungen**

Etwa 1979 bemerkte AT&T, dass sie mit Unix ein überaus wertvolles Produkt besaßen, und gab plötzlich den Sourcecode von Unix nicht mehr weiter, der zum Firmengeheimnis wurde. Unix wurde kommerzialisiert. Viele Leute meinten, dieser Weg wäre falsch, das Teilen von Sourcecode würde die Unix-Entwicklung produktiver machen. Aus dieser Entwicklung entstanden auch zwei freie Unix-Klone, nämlich "Idris" und "Minix", letzteres von Andrew S. Tanenbaum, der Minix auch in seinem Buch "Operating Systems - Design and Implementation" genau beschreibt.

1983 wurde das GNU Projekt gegründet, mit dem Ziel, einen komplett freien Unix-Klon zu entwickeln (GNU = Gnu's Not Unix; rekursives Akronym). Alle Unix-Systemutilities wurden neu geschrieben, und oft sogar verbessert. Das Projekt brachte effiziente Compiler u.a. für C, C++, Fortran, Pascal und Lisp sowie einen mit einem Lisp-Dialekt programmierbaren Editor heraus (EMACS), der schon vorher für ITS existiert hatte. 1990 wurde mit der Entwicklung des Kernels auf Basis des Mach Mikrokernels begonnen und später "The Hurd" genannt. Die Entwicklung ist immer noch nicht abgeschlossen, die aktuellste Version ist 0.2, und noch ist der Kernel langsam und enthält kaum Treiber. Es existiert bereits eine eigene Hurd-Distribution ("Debian GNU/Hurd"), die jedoch noch extrem fehlerhaft ist.

#### **5. "Holy Wars"**

Für viele Unix-Benutzer ist Unix mehr als nur als ein Betriebssystem, sondern eine Art Subkultur, die manchmal sogar religiöse Züge annimmt. Aus diesem Grund gibt es in verschiedensten Bereichen sog. "Holy Wars", die bis in die 70er Jahre zurückreichen. Streitereien von damals sind z.B. BSD gegen SysV Release 4 oder vi gegen emacs (Editor des ITS, der auf Unix portiert wurde). Aus den 80ern stammen Diskussion wie X Window System gegen Konsole, und falls X, welcher Window Manager der beste wäre. Aktuelle Fragen sind z.B. Monolithischer Kernel oder Mikrokern, oder welches Desktop Environment das bessere wäre, CDE, KDE, Gnome, oder gar der CDE-Klon XFCE.

#### **6. Die letzten 10 Jahre**

1991 lernte Linus Torvalds auf der Universität Helsinki Unix kennen. Allerdings besaß er nur einen 80386-Rechner ohne Coprozessor, und dafür war nur Minix gratis verfügbar, das jedoch einige Einschränkungen sowohl technischer als lizenztechnischer Natur hatte. Da er nicht das Geld für ein kommerzielles Unix hatte, und auch auf keinen Fall DOS und Windows einsetzen wollte, begann er, einen eigenen Unix-Klon zu programmieren, und entschied, den Kernel im Internet unter dem Namen "Linux" (Linus Unix) frei zur Verfügung zu stellen. Schnell fanden sich andere Programmierer, die von diesem Projekt begeistert waren, und halfen selbst mit, Linux weiterzuentwickeln. Die ersten Linux-Versionen wurden noch unter Minix kompiliert, mit dem GNU C Compiler. Überhaupt wurden später nur noch die GNU Utilities eingesetzt, und Linux ersetzte damit quasi den GNU-Kernel "The Hurd". Linux gilt heutzutage als sehr stabiles Betriebssystem, das auf vielen verschiedenen Hardware-Plattformen läuft (i386, PowerPC, m68k, Alpha, Sparc,...) und auch von großen Firmen (IBM, SGI, ...) unterstützt wird.

#### **7. \*BSD**

Anfang der 90er Jahre hatte Novell sämtliche Rechte an Unix erworben und begannen damit, ihr Copyright auf diversen Sourcecode, der sich in BSD befand, geltend zu machen. Aus diesem Grund entstanden zwei verschiedene BSD-Versionen: 4.4BSD und 4.4BSD-Encumbered. Für die Encumbered-Version musste man eine teure Unix-Source-Lizenz erwerben. 4.4BSD wurde dann fast vollständig von Copyright-behaftetem Sourcecode entfernt und größtenteils neu programmiert. 4.4BSD endete mit den Versionen

Lite1 und Lite2, die das Ende der BSD-Entwicklung in Berkeley markieren.

Bill Jolitz schaffte es schließlich, 6 Dateien, auf die noch Novell das Copyright hatte, neu zu entwickeln, und BSD auf den 80386 zu portieren. Er nannte diese Version 386/BSD. Jedoch war sie extrem fehlerbehaftet, und Jolitz kam mit der Entwicklung nicht mehr nach. Aus den inoffiziellen "386/BSD-Patchkits" entstanden schließlich zwei Projekte, nämlich NetBSD und FreeBSD.

### **NetBSD**

NetBSD wurde 1993 gegründet, um 386/BSD weiterzuentwickeln. Die Ziele waren schnell definiert: das wohl portabelste Betriebssystem zu entwickeln. Dies ist dem Projekt auch gelungen: der Kernel läuft auf über 40 verschiedenen Rechnertypen, und das sogar stabil. Berühmtheit erlangte das Projekte v.a., als es gelang NetBSD auf die Sega Dreamcast zu portieren.

### **FreeBSD**

FreeBSD ist wie NetBSD ein Nachfolger von 386/BSD, jedoch mit anderen Projektzielen: der Sourcecode sollte möglichst frei sein, und FreeBSD sollte eine gute Performance auf PC-Hardware erreichen. Aus diesem Grund existiert nur eine Portierung auf Alpha-Hardware, noch dazu mit sehr eingeschränkter Funktionalität.

### **OpenBSD**

Etwa 1995 zerstritt sich der NetBSD-Entwickler Theo de Raadt mit anderen NetBSD-Mitgliedern, weil diese ihn wegen seiner Unfreundlichkeit auf diversen Mailinglisten aus dem Projekt ausschließen wollten. Theo stieg daraufhin aus NetBSD aus, und gründete sein eigenes Projekt mit dem Namen OpenBSD. OpenBSD hatte andere Ziele als NetBSD. So wird OpenBSD aktiv nach potentiellen Sicherheitslücken durchsucht. Außerdem wird eine Integration von starker Verschlüsselung angestrebt. Aus dieser Bestrebung ist z.B. das OpenSSH-Projekt entstanden, welches eine freie Version der SSH ("Secure Shell") entwickelt.

### **Darwin**

1998 kündigte Apple den Nachfolger für das betagte MacOS-Betriebssystem an. MacOS X sollte wie NeXTstep auf Mach und BSD aufbauen. Als BSD wurde FreeBSD 3.0 ausgewählt und angepasst. Der Kernel selbst und das Userland dafür sind frei verfügbar, und werden aktiv unter dem Namen "Darwin" weiterentwickelt.

MacOS X ist technisch mit NeXTstep ziemlich ähnlich, anstatt Display Postscript gibt es nun Display PDF, und zusätzliche Multimedia-Fähigkeit und eine aufwändige grafische Oberfläche wurden ebenfalls hinzugefügt.

## **8. Die Zukunft**

Die Zukunft von Unix wird spannend. So ist in fast jedem Gebiet der IT Unix verfügbar, z.B. für Realtime-Anwendungen, Supercomputer, Web-Server, DB-Server, File-Server, Workstations, Clustering, Hand-Helds und den Desktop-Bereich.