

Geschichte von Unix

Andreas Krennmair

a.krennmair@aon.at

1. Vor Unix

Ende der 50er, Anfang der 60er Jahre waren sog. Stapelverarbeitungssysteme ueblich, z.B. die *IBM 7094*. Es konnte nur ein Programm gleichzeitig laufen, und es wurden zwei zusaeztliche Computer benoetigt, um die Ein- und Ausgabe entsprechend durchzufuehren. 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 aufwaendig, teuer und ineffizient, da bei I/O Rechenzeit verschwendet wurde. 1961 wurde das erste Timesharing-System fuer eine modifizierte *IBM 7094* mit 64 KByte RAM entwickelt (CTSS - Compatible Time Sharing System). Bis zu 30 Benutzer konnten ueber serielle Terminals, auch verbunden ueber 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. Als Entwicklungssprache fuer dieses Betriebssystem, genannt Multics (Multiplexed Information and Computing Service) wurde erstmals eine Hochsprache gewaehlt, naemlich *PL/I*, und die Entwicklung wurde unter CTSS durchgefuehrt. Bis der *PL/I*-Compiler verfuegbar war, schrieb das Entwicklungsteam die ca. 3000-seitige Spezifikation fuer Multics. Der Compiler verzoeagerte sich allerdings, und eine abgespeckte *PL/I*-Version (PLM) wurde stattdessen eingesetzt, der Compiler war allerdings extrem langsam. 1968/69 kam das Projekt in finanzielle Schwierigkeiten, und es wurde oefters 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 dafuer zu schreiben. Da dieses System im Gegensatz zu Multics nur ein Zweibenutzerbetriebssystem war, bei dem jeder User nur einen Prozess gleichzeitig ausfuehren konnte, schlug Brian Kernighan vor, es Unics zu nennen. Aufgrund von Begrenzungen bei der Dateinamenlaenge von *GECOS*, das als Entwicklungsplattform diente, wurde das System aber *Unix* genannt. Die grundsaeztlichen Konzepte von Multics wurden aber uebernommen, z.B. ein hierarchisches Dateisystem und "everything is a file", bei dem die Treiberschnittstelle zu Geraten durch eine spezielle Datei repraesentiert wird. Dieses System wurde zuerst nur innerhalb der Bell Labs eingesetzt, so z.B. als Dokumentenverarbeitungssystem fuer die Patentabteilung. Als Software wurde das Dokumentformatierungssystem roff verwendet, das heute unter Unix in den Modifikation troff, groff und nroff immer noch zur Formatierung von Dokumentation und Manpages verwendet wird.

Urspruenglich waren alle Unix-Programme inkl. das System selbst noch in Assembler geschrieben worden. Also entschied sich Ken Thompson, einen *FORTRAN-Compiler* zu schreiben, ohne den seiner Meinung nach ein System wertlos waere. 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, waehrend B wortorientiert war. Also entwickelte Dennis Ritchie B zur Programmiersprache C weiter, indem er u.a. Typen hinzufuegte. 1973

wurde der Unix Kernel komplett in C neugeschrieben ("Version 4") und war damit ein sehr portables System geworden, was spaeter sehr hilfreich sein sollte. Im Oktober des selben Jahres stellten Ken Thompson und Dennis Ritchie das Betriebssystem Unix, mittlerweile ein Multiuser-Multitasking-Betriebssystem, erstmals der breiten Oeffentlichkeit vor, und das Interesse dafuer explodierte foermlich. Etwa zur selben Zeit wurde AT&T verdaechtigt, monopolistische Bestrebungen zu haben, und es war ihnen deswegen verboten, Handel mit Computer- Hard- und Software zu betreiben. Deswegen wurde Unix kostenfrei allen interessierten Universitaeten zur Verfuegung gestellt, allerdings ohne jegliche Produktunterstuetzung.

1975 ging Ken Thompson zurueck zur Universitaet Berkeley. Dort stellten zwei Studenten, Chuck Haley und Bill Joy, das Pascal-System, das Ken Thompson entwickelt hatte, fertig, und entwickelten etwa gleichzeitig den Texteditor em ("Editor for Mortals" = "Editor fuer Sterbliche") zum Editor ex weiter. Fuer ex wurde ein visuelles Terminal-Interface programmiert, das heute noch unter dem Namen "vi" bekannt ist und in verbesserten Versionen wohl der am haeufigsten verwendete Unix-Editor ist. Visuelle Editoren waren damals etwas revolutionaeres, 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 oder Incompatible Timesharing System (ITS) durchgefuehrt, Unix galt eher als Spielzeug. 1978 brachte DEC den VAX heraus, als Nachfolger fuer die veralteten PDP-11/20-Rechner gedacht. Standardmaessig wurden diese Rechner mit dem Betriebssystem VMS ausgeliefert, Unix konnte aber v.a. aufgrund der vorhandenen Kompatibilitaet zur PDP-11 leicht darauf portiert werden ("32V Unix"). Vor allem 4.2BSD, das im Jahr 1982 herausgebracht wurde, war auf den VAX-Maschinen sehr beliebt.

Etwa zur gleichen Zeit kamen Mikrocomputer auf, erschwingliche Rechner fuer den Buero-Einsatz und Privatpersonen. Die qualitativ hochwertigeren Rechner nannten sich "Workstation" und liefen alle mit einer eigenen Unix-Version. Dadurch wurde der Unix-Markt extrem aufgesplittert, und auf einmal existieren unzaehlige verschiedene Varianten, die alle zueinander **groesstenteils** kompatibel waren. Manche Systeme waren auf Basis von BSD entwickelt worden, manche verwendeten den System V Quellcode. Microsoft lizenzierte fuer ihre Unix-Variante Xenix den System-III-Sourcecode, und lizenzierte Xenix an diverse andere Firmen weiter, z.B. SCO oder Intel.

Dann, Ende der 80er Jahre wurden die Workstations leistungsmaessig von Clones des IBM-PC ueberfluegelt. Und alle liefen mit einem primitiven Betriebssystem namens DOS, das schon etwa zur Anfangszeit von Unix als primitiv gegolten haette. Als grafische Oberflaeche war nur MS-Windows, spaeter auch OS/2 von IBM verfuegbar. Die PCs setzten sich gegenueber den Unix-Rechnern v.a. deswegen durch, weil sich die Unix-Hersteller auf keinen einheitlichen Standard bezuegl. grafischer Benutzeroberflaeche einigen konnten. Der Industrie-Standard CDE kam zu spaet, und war auch zu langsam.

4. Unix-interne Entwicklungen

Etwa 1979 bemerkte AT&T, dass sie mit Unix ein ueberaus wertvolles Produkt besaessen, und gab ploetzlich den Sourcecode von Unix nicht mehr weiter, der zum Firmengeheimnis wurde. Unix wurde kommerzialisiert. Viele Leute meinten, dieser Weg waere falsch, das Teilen von Sourcecode wuerde die Unix-Entwicklung produktiver machen. Aus dieser Entwicklung entstanden auch zwei freie Unix-Klone, naemlich "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 gegrundet, 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 u.a. effiziente Compiler fuer C, C++, Fortran,

Pascal und Lisp sowie einen mit einem Lisp-Dialekt programmierbaren Editor heraus (emacs), der schon vorher fuer ITS existiert hatte. 1990 wurde mit der Entwicklung des Kernels auf Basis des Mach Mikrokernelns begonnen und "The Hurd" genannt. Die Entwicklung ist immer noch nicht abgeschlossen, die aktuellste Version ist 0.3, und noch ist der Kernel langsam. Es existiert bereits eine eigene Hurd-Distribution ("Debian GNU/Hurd"), die jedoch noch extrem fehlerhaft ist (mehr als 100000 Bugs).

5. "Holy Wars"

Fuer viele Unix-Benutzer ist Unix mehr als nur als ein Betriebssystem, sondern eine Art Subkultur, die manchmal sogar religioese Zuege annimmt. Aus diesem Grund gibt es in verschiedensten Bereichen sog. "Holy Wars", die bis in die 70er Jahre zurueckreichen. 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 Diskussionen wie X Window System gegen Konsole, und falls X, welcher Window Manager der beste waere. Aktuelle Fragen sind z.B. Monolithischer Kernel oder Mikrokernel, oder welches Desktop Environment das bessere waere, KDE oder Gnome, oder gar der CDE-Klon XFCE.

6. Die letzten 10 Jahre

1991 lernte Linus Torvalds auf der Universitaet Helsinki Unix kennen. Allerdings besass er nur einen 80386-Rechner ohne Coprozessor, und dafuer war nur Minix gratis verfuegbar, das jedoch einige Einschränkungen sowohl technischer als lizentechnischer Natur hatte. Da er nicht das Geld fuer 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 Verfuegung 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. Ueberhaupt wurden spaeter 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 laeuft (i386, PowerPC, m68k, Alpha, Sparc,...) und auch von grossen Firmen (IBM, SGI, ...) unterstuetzt wird.

Waehrend sich die kommerziellen Unix-Entwickler gegenseitig Konkurrenz machten, brachte Microsoft 1995 Windows 95 und 1996 seinen stabileren Bruder Windows NT 4.0 heraus, erstmalig 32-bit-Windows-Versionen mit preemptivem Multitasking und ansprechenden grafischen Oberflaechen fuer die breite Masse. Die Benutzeroberflaeche ist dabei eine Mischung von NeXTStep und MacOS. Mit Windows NT zielte Microsoft ganz klar auf Bereiche ab, in denen Unix vorher dominant war, z.B. Workstation, Web-Server und File-Server. Allerdings zeigte sich sehr schnell, dass auch Windows NT 4.0 und 2000 auch nur ein Unix sein "wollen", Microsoft hat Windows 2000 naemlich nach eigenen Aussagen um "innovative Konzepte" ergaenzt, die unter Unix allerdings schon seit 25 und mehr Jahren existieren, z.B. Hardlinks oder das Mounten von Partitionen und Datentraegern.

7. Die Gegenwart

Die meisten Entwicklungen rund um Unix spielen sich meist in Zusammenhang mit Linux ab. So z.B. hat Sun den kompletten Sourcecode von Solaris 8 offengelegt, als direkte Reaktion auf die freie Verfuegbarkeit von Linux. Aber es existieren neben Linux noch andere freie Unix-Projekte, alle Ableger von BSD. Diese Projekte sehen sich selbst als "die Elite", und haben mit ihren BSD-Varianten einen technischen Erfolg, die Popularitaet ist jedoch bei weitem nicht so gross wie bei Linux. Freie BSD-Varianten gibt es z.Z. vier:

FreeBSD

Ziel des Projektes ist es, ein moeglichst stabiles Betriebssystem fuer PCs zu entwickeln. Als Codebasis wurde 386/BSD 0.1 herangezogen. Mit den Patchsets "SecureBSD" und "TrustedBSD" kann die

Sicherheit von FreeBSD noch zusätzlich verbessert werden.

NetBSD

dieses Projekt hat den wohl portabelsten Kernel. Er laesst sich auf ueber 10 Hardwareplattformen kompilieren und laeuft dort auch stabil (Alpha, ARM, i386, m68k, MIPS, NS32K, PowerPC, Hitachi Super H Prozessorfamilie [Sega Dreamcast], Sparc, VAX). Gegrundet wurde das NetBSD-Projekt im April 1993 und verwendete ebenfalls 386/BSD als Codebasis.

OpenBSD

Dieses Projekt spezialisiert sich auf Sicherheit und im speziellen auf Kryptographie (OpenBSD's Motto lautet "secure out of the box"). So stammt z.B. die bekannteste Implementierung der Secure Shell (ssh) vom OpenSSH-Projekt, das eng mit OpenBSD zusammenarbeitet. Gegrundet wurde das Projekt von einem ehemaligen NetBSD-Entwickler, dem alle Accounts gesperrt wurden, weil er sich mit anderen Entwicklern zerstritten hatte. Spaeter flossen in OpenBSD auch diverse Komponenten von 4.4BSD Lite2 und FreeBSD ein.

Darwin

Darwin basiert auf FreeBSD 3.0 und dem Mach Mikrokern und bildet den Kernel fuer Apple's neues Betriebssystem MacOS X. Der Sourcecode ist frei verfuegbar und genauso portabel wie FreeBSD, deswegen ist moeglicherweise mit einer Portierung von MacOS X auf Intel-basierte Rechner zu rechnen. Apple sponsort das Projekt, und gibt Verbesserungen an das FreeBSD-Projekt zurueck.

8. Die Zukunft

Kuenftige Anwendungsgebiete von Unix werden in Zukunft "embedded" Anwendungen (QNX ist POSIX-kompatibel), Web-Server, Datenbank-Server (IBM), File-Server, High-End-Workstations (z.B. von Sun oder SGI), Clustering, und der Desktop-Bereich sein (MacOS X). Interessante Entwicklungen speziell im Bereich Linux sind z.B. Journaling Filesystems (ext3 von den Linux-Entwicklern, XFS von SGI, JFS von IBM und reiserfs von Hans Reiser), Hand-Helds (Linux auf dem Compaq iPAQ, "Linux-Uhr", ...) und die Verwendung auf Grossrechnern wie des IBM System/390.

9. Unix-Varianten

A/UX, AIX, ArchBSD, System III-V, BSD/OS, Coherent, Darwin, Digital Unix, FreeBSD, HP-UX, Hurd, Linux, MS Xenix, Minix, NetBSD, NeXT/OpenSTEP, OpenBSD, POSIX, PWB/UNIX, QNX, Reliant Unix, SCO OpenServer, SCO Xenix, SINIX, SunOS, Solaris, Ultrix, UNICOS, SCO Unixware, UTS, Venix, Xinu.