4.2BSD and 4.3BSD as Examples of the UNIX System

JOHN S. QUARTERMAN, ABRAHAM SILBERSCHATZ, and JAMES L. PETERSON

Department of Computer Sciences, University of Texas, Austin, Texas 78712

This paper presents an in-depth examination of the 4.2 Berkeley Software Distribution, Virtual VAX-11 Version (4.2BSD), which is a version of the UNIX™ Time-Sharing System. There are notes throughout on 4.3BSD, the forthcoming system from the University of California at Berkeley. We trace the historical development of the UNIX system from its conception in 1969 until today, and describe the design principles that have guided this development. We then present the internal data structures and algorithms used by the kernel to support the user interface. In particular, we describe process management, memory management, the file system, the I/O system, and communications. These are treated in as much detail as the UNIX licenses will allow. We conclude with a brief description of the user interface and a set of bibliographic notes.

Categories and Subject Descriptors: C.2.4 [**Computer–Communication Networks**]: Distributed Systems—*distributed applications*; D.4.0 [**Operating Systems**]: General—*UNIX*; D.4.7 [**Operating Systems**]: Organization and Design—*interactive systems*; K.2 [**History of Computing**]: Software—*UNIX*

General Terms: Algorithms, Design, Human Factors, Performance, Reliability, Security

Additional Key Words and Phrases: Flexibility, portability, simplicity

INTRODUCTION

This paper presents an in-depth examination of the *4.2BSD* operating system, the research *UNIX*[1] system developed for the Defense Advanced Research Projects Agency (DARPA) by the University of California at Berkeley. We have chosen 4.2BSD over UNIX System V (the UNIX system currently being licensed by AT&T) because concepts such as internetworking and demand paging are implemented in 4.2BSD but not in System V. Where 4.3BSD, the forthcoming system from Berkeley, differs functionally from 4.2BSD in the areas of interest, such differences are noted.

This paper is not a critique of the design and implementation of 4.2BSD or UNIX; it is an explanation. For comparisons of System V and 4.2BSD, see the literature, particularly the references given in Section 1.1, p. 380. Such comparisons are mostly beyond the scope of this paper.

The *VAX*[2] implementation is used because 4.2BSD was developed on the VAX,

[1] UNIX is a trademark of AT&T Bell Laboratories.

[2] VAX, PDP, TOPS-20, and VMS are trademarks of Digital Equipment Corporation.

Chapter 14 of *Operating Systems Concepts, Second Edition*, by J. L. Peterson and A. Silberschatz (© 1985 by Addison-Wesley, Reading, Massachusetts) and this article were both derived from an earlier common manuscript by J. S. Quarterman. Consequently they share some text. Common portions are reprinted with the permission of Addison-Wesley.

Author's present address: James L. Peterson, MCC, 9430 Research Blvd., Austin, Texas 78759.

CONTENTS

and that machine still represents a convenient point of reference, despite the recent proliferation of implementations on other hardware (such as the Motorola 68020 or National Semiconductor 32032). Also, details of implementation for non-VAX systems are usually proprietary to the companies that did them. And space does not permit examination of every implementation on every kind of hardware.

This paper is not a tutorial on how to use UNIX or 4.2BSD. It is assumed that the reader knows how to use the UNIX system. The presentation is closely limited to a technical examination of traditional operating system and networking concepts, most of which are implemented in the *kernel*. Students of operating systems and novice systems programmers (the intended readership) should find the organization and content appropriate.

The novice UNIX user will want to read Section 7 on the user interface before delving into the sections on kernel details. That section is as brief as possible, because the user interface and user programs in general are (regardless of their importance to the utility and popularity of the system) beyond the proper scope of this paper. Reading one of the several good books on using UNIX (see Section 8, Bibliographic Notes) would be good preparation for reading the paper.

The paper begins with a very brief overview of the history of the system and some description of the design philosophy behind it. The other sections cover process management, memory management, the file system, the I/O system, communications, and certain features of the user interface that distinguish the system. The paper concludes with a set of bibliographic notes.

1. OVERVIEW

This section is concerned with the history and design of the UNIX system, which was initially developed at Bell Laboratories as a private research project of two programmers. Its original elegant design and developments of the past fifteen years have made it an important and powerful operating system. We trace the history of the system E [Compton 1985; Ritchie 1978, 1984a, 1984b] and relate its design principles.

1.1 History

The first version of UNIX was developed at Bell Laboratories in 1969 by Ken Thompson to use an otherwise idle PDP-7. He was soon joined by Dennis Ritchie, and the two of them have since been the largest influence on what is commonly known as Research UNIX.

Ritchie, Thompson, and other early Research UNIX developers had previously worked on the Multics project [Peirce 1985], and Multics [Organick 1975] was a strong influence on the newer operating system. Even the name UNIX is merely a pun on Multics, indicating that in areas where Multics attempted to do many things, UNIX tries to do one thing well. The basic organization of the file system, the idea of the command interpreter (the shell) being a user process, the use of a process per command, the original line-editing characters # and @, and many other things come directly from Multics.

Ideas from various other operating systems, such as Massachusetts Institute of Technology's CTSS, have also been used. The *fork* operation comes from Berkeley's GENIE (XCS-940) operating system.

The Research UNIX systems include *UNIX Time-Sharing System, Sixth Edition* (commonly known as *Version 6*), which was the first version widely available outside Bell Laboratories (in 1976) and ran on the PDP-11. (These version numbers correspond to the edition numbers of the UNIX Programmer's Manual that were current when the distributions were made.) Multiprogramming was added before Version 6, and after the system was rewritten in a high-level programming language, C [Kernighan 1978; Ritchie et al. 1978]. C was designed and implemented for this purpose by Dennis Ritchie. It is descended [Rosler 1984] from the language B, designed and implemented by Ken Thompson. B was itself descended from BCPL. C continues to evolve [Stroustrup 1984; Tuthill 1985a].

The first portable UNIX system was *UNIX Time-Sharing System, Seventh Edition* (*Version 7*), which ran on the PDP-11 and the Interdata 8/32, and had a VAX variety called *UNIX/32V Time-Sharing System Version 1.0* (*32V*). The system currently in development by the Research group at AT&T Bell Laboratories is *UNIX Time-Sharing System, Eighth Edition* (*Version 8*).

After the distribution of Version 7 in 1978, the Research group gave external distributions over to the UNIX Support Group (USG). USG had previously distributed such systems as *UNIX Programmer's Work Bench* (*PWB*) internally, and sometimes externally as well [Mohr 1985].

Their first external distribution after Version 7 was *UNIX System III* (*System III*), in 1982, which incorporated features of Version 7, 32V, and also of several UNIX systems developed by groups other than the Reseach group. Features of UNIX/RT (a real-time UNIX system) were included, as well as many features from PWB. USG released *UNIX System V* (*System V*) in 1983; it is largely derived from System III. The divestiture of the various Bell Operating Companies from AT&T has left AT&T in a position to market System V [Wilson 1985] aggressively. USG has metamorphosed into the UNIX System Development Laboratory (USDL), whose current distribution is *UNIX System V Release 2* (*V.2*), released in 1984.

The ease with which the UNIX system can be modified has led to development work at numerous organizations such as Rand, Bolt, Beranek and Newman (BBN), the University of Illinois, Harvard, Purdue, and even DEC. But the most influential of the non-Bell Laboratories and non-AT&T UNIX development groups has been the University of California at Berkeley [McKusick 1985]. UNIX software from Berkeley is released in so-called *Berkeley Software Distributions* (BSD), hence the generic numbers *2BSD* for the later PDP-11 distributions and *4BSD* for the later VAX distributions.

Many of the features of the 4BSD terminal drivers are from TENEX/TOPS-20,[3] and efficiency improvements have been made as a result of comparisons with VMS.

The first Berkeley VAX UNIX work was the addition to 32V of virtual memory, demand paging, and page replacement in 1979 by Bill Joy and Ozalp Bagaoğlu to produce *3BSD*. The large virtual memory space of 3BSD allowed the development of very large programs, such as Berkeley's own *Franz Lisp*. This memory management work convinced DARPA to fund Berkeley for the later development of a standard UNIX system for government use (4BSD).

One of the goals of this project was to provide support for the DARPA Internet networking protocols TCP/IP. This was done in a general manner, and it is possible to communicate among diverse network facilities, ranging from local networks (such

[3] TENEX is a registered trademark of BBN.

as Ethernets[4] and token rings) to long-haul networks (such as DARPA's ARPANET).

It is sometimes convenient to refer to the Berkeley VAX UNIX systems following 3BSD as 4BSD, although there were actually several releases (indicated by decimal points in the release numbers), most notably *4.1BSD*. 4BSD was the operating system of choice for VAXs from the beginning until the release of System III (1979–1982) and remains so for many research or networking installations. Most organizations would buy a 32V license and order 4BSD from Berkeley without ever bothering to get a 32V tape. Many installations inside the Bell System ran 4.1BSD (many still do, and many others run 4.2BSD).

The 4BSD work for DARPA was guided by a steering committee, which included many notable people from the UNIX and networking communities. *4.2BSD*, first distributed in 1983, is the culmination of the original Berkeley DARPA UNIX project, although further research proceeds at Berkeley.

Berkeley was not the only organization involved in the development of 4.2BSD. Contributions (such as autoconfiguration, job control, and disk quotas) came from numerous universities and other organizations in Australia, Canada, Europe, and the United States. A few ideas, such as the *fcntl* system call, were taken from System V. (Licensing and pricing considerations have prevented the use of any actual code from System III or System V in 4BSD.) Not only are many contributions included in the distributions proper, but there is an accompanying set of user-contributed software, which is carried on the tapes containing the 4BSD distributions. The system was tested on the M68000-based workstation by Sun Microsystems, Inc., before its initial distribution. This simultaneous development contributed to the ease of further ports of 4.2BSD.

Berkeley accepts mail about bugs and their fixes at a well-known electronic address, and the consulting company mt.Xinu distributes a bug list compiled from such submissions. Many of the bug fixes may be incorporated in future distributions. There is constant discussion of UNIX in general (including 4.2BSD) in the DARPA Internet mailing list UNIX-WIZARDS, which appears on the USENET network as the news group *net.unix-wizards*; both the Internet and USENET are international in scope. There is another USENET news group dedicated to 4BSD bugs. While few ideas appear to be accepted by Berkeley directly from these lists and news groups (probably because of the difficulty of sifting through the sheer volume of submissions), discussions in them sometimes lead to new facilities being written that are later accepted.

Figure 1 is a sketch of the evolution of the several main branches of the UNIX system, especially those leading to 4.2BSD and System V [Chambers and Quarterman 1983; Uniejewski 1985]. The dates given are approximate, and there is no attempt to show all influences. Some of the systems named in the figure are not mentioned in the text, but are included to better show the relations among the ones that are discussed in the text.

We are aware at this writing of the imminent release of 4.3BSD and System V Release 2 Version 4. There are few functional changes in the kernel in 4.3BSD, although there are many performance improvements [Cabreral et al. 1985; Leffler et al. 1984; McKusick et al. 1985]. (Some of these 4.3BSD changes are noted in sections throughout this paper.) Although System V Release 2 Version 4 does introduce paging [Jung 1985; Miller 1984] (including copy-on-write and shared memory) to System V, there are few other functional changes.

Dozens of computer manufacturers,[5] including almost all of those usually

[4] Ethernet is a trademark of Xerox Corporation.

[5] These include at least Altos, Amdahl, Apollo, AT&T, Burroughs, Callan, Celerity, Codata, Convergent Technologies, Convex, COSI, Cray, Cromemco, Data General, DEC, Denelcor, Dual Systems, ELXSI, Encore, Flexible, Gould, Heurikon, Hewlett Packard, Honeywell, IBM, Integrated Business Computers, Integrated Solutions, Intel, Interactive Systems, Logical MicroComputer, Medical Informatics, NBI, NCR, National Semiconductor, Onyx, Pacific Computer, Parallel, Perkin-Elmer, Plexus, Pyramid, R Systems, Radio Shack, Ridge, Sequent, Silicon Graphics, Sperry, Sun Microsystems, Tektronix, Visual Technology, and Wicat.

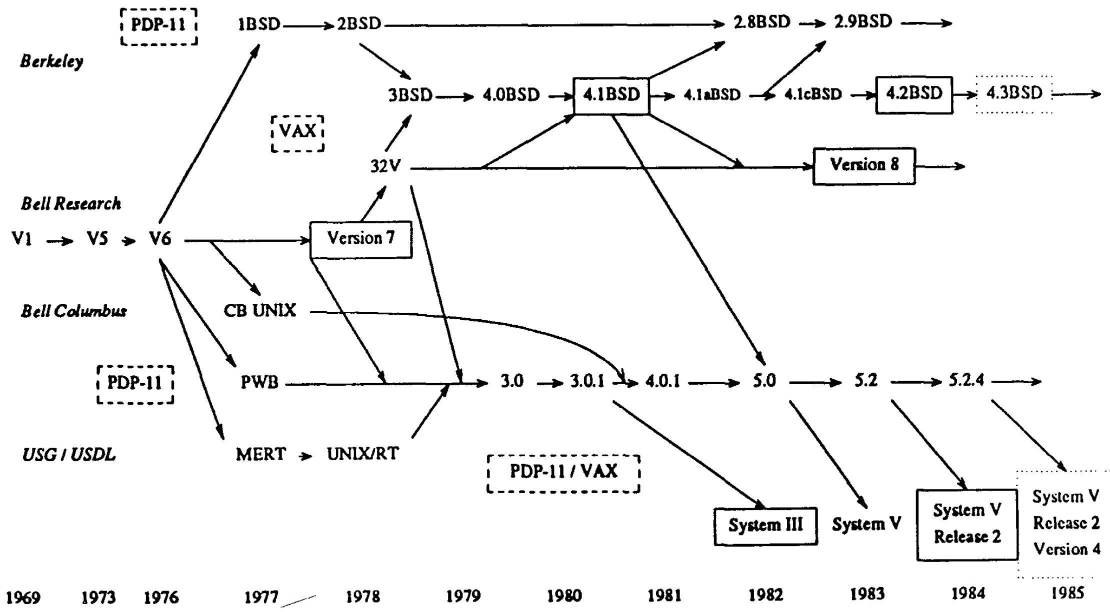

Figure 1. UNIX history.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.