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