

UNIX™ & DEMYSTIFIED XENIX®

LEE PAUL CLUKEY

37221
49

UNIXTM & DEMYSTIFIED XENIX[®]

LEE PAUL CLUKEY

TAB TAB BOOKS Inc.
Blue Ridge Summit, PA 17214

FIRST EDITION

FIRST PRINTING

Copyright © 1985 by TAB BOOKS Inc
Printed in the United States of America

Reproduction or publication of the content in any manner, without express permission of the publisher, is prohibited. No liability is assumed with respect to the use of the information herein.

Library of Congress Cataloging in Publication Data

Clukey, Lee Paul.
UNIX and XENIX demystified.

On t.p. the registered trademark symbol "TM" is superscript following "UNIX" in title and registered trademark symbol "R" is superscript following "XENIX" in title.

Includes index.

1. UNIX (Computer operating system) 2. XENIX (Computer operating system). I. Title.

QA76.8 C57 1985 001.64'2 85-2658

ISBN 0-8306-0874-5

ISBN 0-8306-1874-0 (pbk.)

Should You Read This Book?

Unix and Xenix Demystified is applicable to Unix and Unix-based systems like IBM's PC/IX, as well as to the popular Xenix, which is currently in use on a great many microcomputers and which will be used to operate the newly announced IBM Personal Computer AT. If you are apprehensive about Unix, you will find that *Unix and Xenix Demystified* provides easy-to-read and understandable explanations about Unix and what you will need to know about operating a computer with a Unix system.

This book has been written to help managers, computer salesmen, and individuals considering purchasing or who have recently purchased a computer operating with Unix, to gain a basic understanding of Unix without getting involved with the more stringent technical details found in more advanced Unix textbooks.

It is written for the person who uses a computer for word processing, accounting, and other office application programs, but who would like to gain a better understanding of what Unix is doing in their computer or who would like to learn how to begin taking greater advantage of the capabilities available in their Unix system.

Unix and Xenix Demystified is based on an actual Unix instruction course. It is the only Unix text dedicated to a tutorial presentation of the Unix system. The text began as class notes to a "Beginning Unix" class taught at the Computer Training Centers, in Los Angeles, California. Our instructors are currently teaching Unix at UCLA. Educators will find the book an excellent beginning course for teaching Unix, including step-by-step instructions for laboratory exercises.

The appendices contain some useful information, such as a composite listing of Unix commands from Version 7, System III and V, Berkeley 4.2, Xenix, PC/IX, and UniPlus+. Appendix C provides a cross-reference matrix of the systems in which these commands may be found. You will find this information very useful for comparing the various Unix systems, to determine which commands have been added to each version of Unix and the differences between these systems.

At this point thanks are in order to Microsoft Corporation for providing the material on Xenix which is reprinted in Appendix D. Thanks especially to my wife for enduring countless late-night hours of writing, rewriting, reading, and editing. And finally, thanks to Irv Barilia and Tony Turgeon at the Computer Training Center in Thousand Oaks for technical consultation. These are just a few of the people who helped make *Unix and Xenix Demystified* possible.

Introduction: Learning to Use Unix

The aura surrounding the Unix system causes many potential users of Unix to shy away from it. For example, Unix training has been costing upwards of a thousand dollars for a week-long series of classes, and it is currently taught at only a few schools and colleges throughout the country. In comparison to DOS, which is taught in less than a day and seldom costs more than a hundred dollars, Unix appears as if it must be difficult to learn to use.

Unix textbooks are often lengthy, some as long as 400 to 500 pages; their appearance would suggest that it is important, if not required, to read and master it all in order to learn to use Unix. Even more lengthy are the Unix programmers' manuals from Bell Laboratories (AT&T), which are the bible on Unix. They are so lengthy and difficult to understand that even experienced computer users not only shy away from Unix, but run from it.

Opponents of Unix print banner headlines in magazine articles warning that Unix is very large, very cryptic, "user-unfriendly," and on and on. They claim Unix is a program written by programmers for programmers, and that it is totally unsuited for microcomputer installations or usage.

The pro-Unix articles also have not done much to help persuade users to learn to use Unix. Some of the so-called "Introduction to Unix" articles make Unix look more like an intermediate- to advanced-level course in computer programming.

It should therefore come as no surprise that Unix has been shunned by so many people. Had it not been for an error in my purchase order, I too would probably be standing among the ranks, throwing rocks at Unix, singing hymns of praise to DOS and CP/M, and bragging about making a wise decision.

However, after having learned to use Unix, I feel it was definitely worth the time. I think that computer installations using Unix are offered vastly superior capabilities and better software packages. I also found it as easy to learn to use Unix as lesser systems. My only problem was finding written material to study. I was not about to spend a thousand dollars, and at the time there were only three Unix books available.

My purpose for writing this book is to help you over the most difficult hurdles in getting started with Unix, first by explaining how to go about it, and second by providing a logical and complete introduction to the system. This will prepare you for reading more advanced textbooks and using the Unix programmers' manuals.

In the process I hope I will also be able to dispel the myth that the Unix system is difficult to learn. I believe that the Unix myth, like all myths, will no longer maintain its mystical air once it is exposed to the light of understanding.

By familiarizing you with the operation of the Unix system and by explaining what you need to know to use Unix effectively, this book will convince you that learning to use Unix in your everyday business needs is within the competence of anyone capable of learning to use a word processor or electronic spreadsheet program.

The presentation approach I take gives you a simplified overview of the Unix system before getting involved in any details on using Unix itself. If you first understand what is going on in your system, as well as the significance of this information to your operating needs, a discussion on how to use Unix will be of much more import. This is the main reason that the text is divided into two parts, a tutorial and lab exercise sessions. Although invariably you will learn more by using Unix than you ever will by just reading about it, going through the motions without understanding the significance of what it is you are doing, why you are doing it, and/or of what value the exercises will eventually have for you will limit your ability to retain this information.

Pushing buttons on your terminal keyboard and having the described responses appear on the terminal screen is satisfying. You will soon realize, however, that because the commands have no meaning to you, you will have to relearn these commands several times before they become a part of your vocabulary. You will find that your retention will increase if you reverse the learning process, learning first what you are doing before you do it.

I have tried to write this book for the beginning user in environments other than software development; it is sometimes difficult, however, to know just where to end a technical discussion and how much detail is too much. It is impossible to find the exact point that will satisfy all readers. Thus, when I felt that the detail might be beyond the needs of the beginning user or cause some apprehension, yet be necessary to the basic understanding of the subject, I have isolated it in such a manner that it will not interfere with the presentation of the main line of the subject.

Through reading *Unix and Xenix Demystified*, you will learn that it is necessary to learn to use only a small fraction of the Unix commands to derive extensive benefits from it. In fact, in the typical office installation the services provided by the Unix system probably will not even be apparent to you. From what you will see on your terminal, your system could be operating on something called

"Whatzis." Ninety percent or more of the time you spend operating your computer will be spent with application programs such as word processors; the remaining 10 percent, tasks like adding users or moving files between users, are likely to be performed with system administration programs provided by the computer manufacturer, programs designed for individuals who have little or no computer operating experience.

You might thus ask, "Why should I bother learning about Unix at all?" The answer is that at some time it may be necessary for you to correct a problem or effect some changes to your files that would be difficult or even impossible to do with the program functions available to you in the manufacturers system administration program, the word processor or electronic spreadsheet, or other programs that you have on your computer. If you have some familiarity with the Unix operating system itself, you will be more capable of correcting such problems.

An Outline for This Book

When I begin to read a new textbook I usually start by examining the table of contents of the book in order to get an overall perspective of the subject. This gives me a kind of road map of the material and helps me to develop an overview of the subject, an overview I can use to relate the detailed components.

Unfortunately, most tables of contents are usually filled with nonsignificant chapter and subject titles. Sometimes they are too cute; "The Old Shell Game," for example, is not very descriptive to a new user. Other times they are too detailed. For this reason I have included a graphic overview (Fig. I-1), which will enable you to see, almost at a glance, what is in *Unix and Xenix Demystified*, and the interrelationships among the various subjects.

The diagram, read from the top down, provides an overview of the Unix system with respect to the modular components which make up the Unix system, as follows:

Utilities. These are shown divided into eight categories, based on their specific functions and services.

Operating System. This is composed of a kernel, system calls, and a shell, which in turn contains three types of commands:

Built-in commands

Metacharacters

Conditional commands

The first type of command is included in the shell in order to expedite processing frequently used commands. The second and last types are used in conjunction with the development and entry of complex Unix commands and shell programming.

Subroutine Libraries. As the name suggests, these contain libraries of subroutines used principally in the development of C- and FORTRAN-based programs and in conjunction with the operation of the system.

Special Device Files. These are most often used to control flow of data between the computer and the peripheral devices attached to the computer, e.g., printers, disk drives, modems, etc.

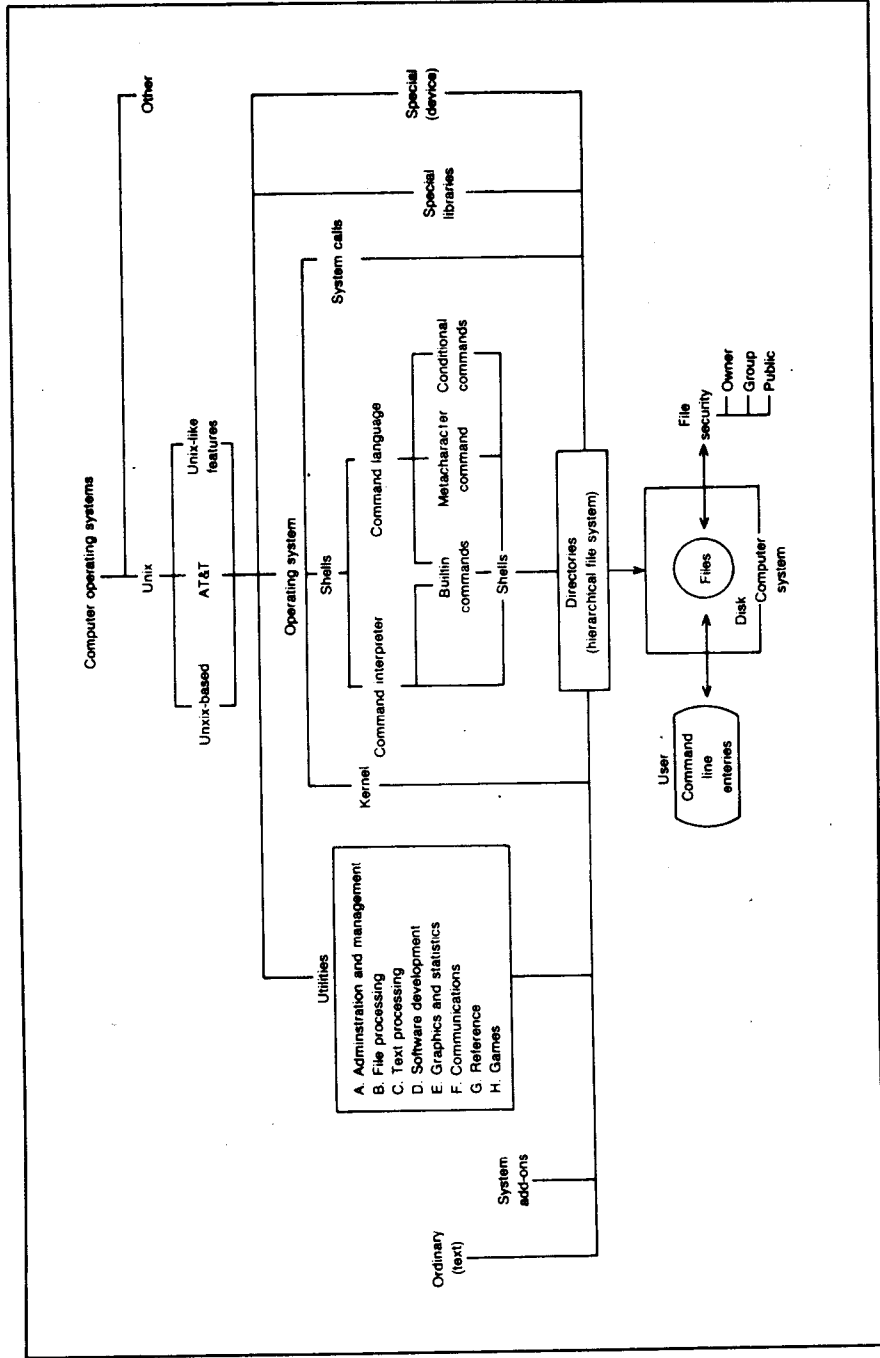


Fig. 1-1. A graphic overview of Unix and Xenix Demystified.

From the bottom up, the diagram displays the use of the Unix system. The user creates a command line—the entry of a command at the terminal—which in turn calls upon the files in the computer's disk storage to operate the computer. The files in the system are stored in a hierarchically structured arrangement, based upon the disk address locations contained in the directories.

All data in the computer—utilities, the Unix operating system itself, the libraries of subroutines, special device files, system add-ons such as word processors as well as ordinary text—are all contained in files on the computer's disk. All of the locations for these files are contained in directories.

The composite diagram shows the interrelationship among the discussions on the physical breakdown of a Unix system and the interaction involved in using the Unix system to operate your computer.

Contents

Should You Read This Book?	vi
Introduction: Learning to Use Unix	viii
PART 1 A TUTORIAL APPROACH	1
Chapter 1 Introduction to Unix Systems	2
Computer Operating Systems	2
Utilities and the Unix System	5
Generic Unix Systems	7
AT&T Unix Systems—Unix-based Systems—Unix-like Systems	
How Big Is a Unix System?	12
Unix System Add-Ons	13
Systems Nomenclature	14
Chapter 2 The Structure of the Unix System	15
The Unix Operating System	15
The Kernel—System Calls—The Shell	
Unix Utilities	21
Applications Programs	22
Software Levels	22
Chapter 3 The Unix File System	26
Auxiliary Memory	26
The File	29
Ordinary Files—Directory Files—Special Files—File Contents—File Names	
The Hierarchical File Structure	32

Building the Hierarchical Structure—Purpose of the Structure—The Hierarchical Structure in Operation—Directories in the File Structure—Typical Top-Level Directories	
Pathnames	40
File Security	42
Registered File Owner—The Group	
Linking Files	45
Mountable File Systems	47
Summary	47
Chapter 4 Commands	48
Shells	48
System Shells—Menu Shells—Application Program Shells—Selecting a Shell	
The Command	55
Command Formats—Command Components—Command Terminology	
Types of Commands	61
Utilities—Shell Commands—System Calls—Subroutines and Libraries	
Summary	62
Chapter 5 Shell Command Language	63
Metacharacters	63
The Command Line	64
Using the Shell Command Language	65
Multiple-Command Command Lines—Pipes—Pipe Lines—Filters—The Tee Command—Background Processes—Subcommands and Command Groupings	
Chapter 6 Introduction to Shell Programming	73
Command Files vs. Shell Scripts	74
Running a Shell Program	74
Shell Conditional Commands	76
Summary	76
Chapter 7 Tutorial Summary	77
PART 2 HANDS-ON UNIX	82
Chapter 8 Introduction to Lab Exercises	83
An Overview	83
Some Things You Should Know	84
The Shell Prompt—Command Interpretation—Entering a Command—Halting a Command in Process—Control Keys	
Chapter 9 Login and System-Level Security	88
Login Name	89
Password	90
Logged-On Acknowledgment	91
Log Off	92
Chapter 10 Locating Directories and Files	93
Who Am I	93
Print Working Directory	94
Change Working Directory	95
List Directory Contents	97
Changing Directories with a Metacharacter	105

Chapter 11 Making Files and Directories	109
Making Files with the Redirection Command 110	
Making Files with Appended Redirection 115	
Making Files with the Move Command 117	
Making a File with the Copy Command 117	
Making a Directory 120	
File Creation Using Pathnames 124	
Chapter 12 File Security	129
Changing File Security 129	
File Copying Permission 133	
Changing File Ownership 135	
Chapter 13 Linking Files	139
Chapter 14 Using Shell Metacharacters	144
Redirect Input 144	
Multiple-Command Command Lines 145	
Pipes 146	
Pipe Lines—Filters—The Tee Command	
Summary 156	
Chapter 15 Making and Using Shell Programs	157
Using Shell Variables 159	
Shell Programs with Pipes 166	
Chapter 16 Removing Files and Directories	170
Appendix A: Lab Exercise Command Summary	178
Appendix B: Consolidated Directory of Unix Utilities	188
Appendix C: Unix Utility Cross-Reference Matrix	204
Appendix D: The Xenix System	230
Index	242

78906.54

B1

Part 1

A Tutorial Approach

Chapter 1

Introduction to Unix Systems

"Unix systems are a coalition of computer programs which have been combined into one interactive system."

"Unix systems have the peculiar characteristic of becoming an industry standard that in and of itself lacks standardization."

"Seldom are two Unix systems the same."

These three statements express why Unix systems are so versatile and flexible—and at the same time so difficult to define and describe for the beginning user.

COMPUTER OPERATING SYSTEMS

The computer, terminal (keyboard and monitor), printer, telephone modems, and all of the other electronic gadgets you may have connected to your computer are referred to as the *system hardware* or the *computer system* (Fig. 1-1).

Computer systems in themselves have no capabilities to perform accounting or word processing, or solve engineering problems, any more than a telephone can dial a telephone number by itself. All of the capabilities you attribute to a computer are derived from the *software* (computer programs) that you install in your computer (Fig. 1-2).

If, for example, you have ever used an automatic telephone dialing unit, you know that you must first enter the series of numbers that make up the frequently called telephone numbers you wish to have available for use in the auto-dialer, before the unit can dial the number automatically for you. The series of numbers you enter into the unit can be likened to a computer program. It contains the

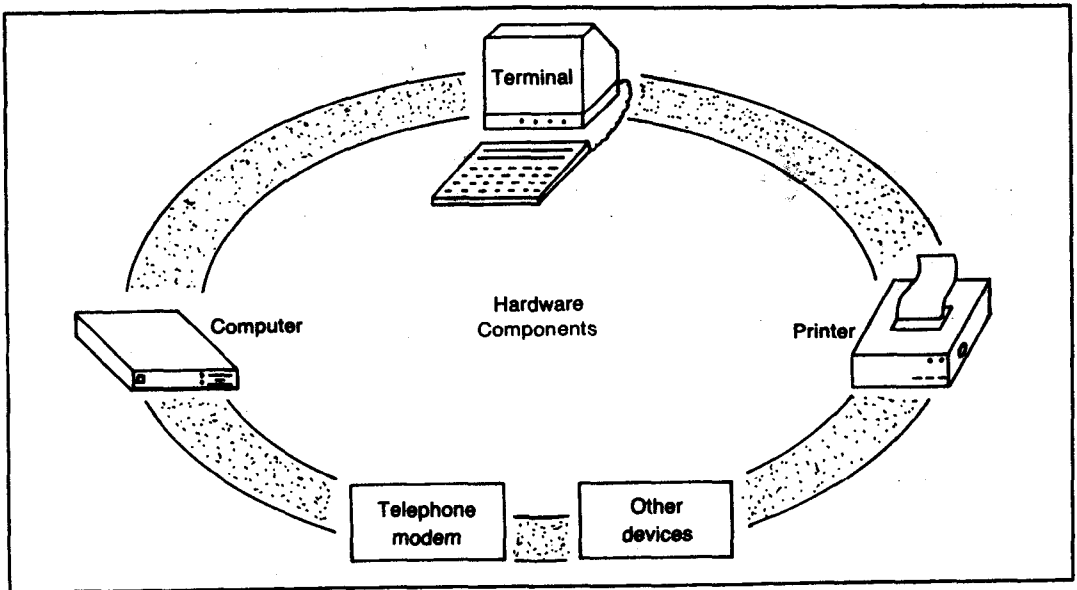


Fig. 1-1. The computer system.

dialing instructions (the telephone number) which, digit by digit, makes the unit place the call for you.

Computer programs are similar to the telephone number in our analogy. They are considerably more complex than a telephone number, but they perform the same type of operation in the computer as the telephone number does in the automatic dialing unit. Computer programs contain lines upon lines of instructions which together may serve as a word processor, an electronic spreadsheet, a database manager, etc. Like the telephone number entered into the automatic dialing unit, the purpose of the computer program is to make the machine perform some job for you automatically.

There are many families of computer programs you can install in your computer system. Word processors, accounting programs, and electronic spreadsheets are of the family called *application computer programs*. You are probably most familiar with this family of software because they are always in the spotlight. They probably are the major reasons you bought your computer.

A software family with which you may not be as familiar is the family called *computer operating systems*. You are most probably not familiar with computer operating systems because their operation is not readily apparent to you. However, if it were not for the work performed by the computer operating system, none of the more apparent application program software would be able to run on your computer.

The computer operating system (or simply the *operating system*) is the fundamental software required for operating any computer system. Unix, from Bell Laboratories (AT&T), is just one of the many operating systems available on the market. A few of the others whose names you may know are:

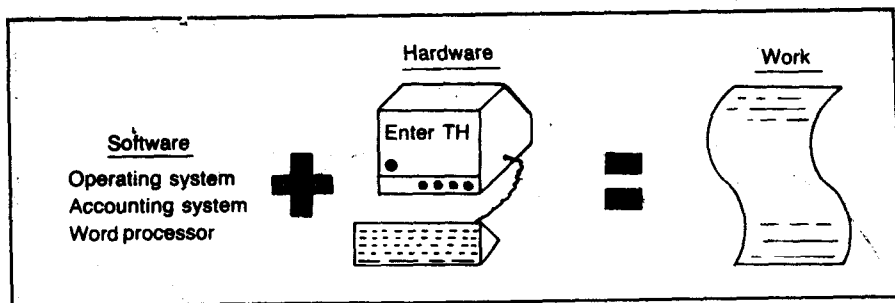


Fig. 1-2. Software drives the computer system.

CP M
MP M
VMS
MS-DOS
DOS
CAPS
CP M-86
MP M-86
PC-DOS

CP M-80
TRSDOS
Apple DOS
Pick
DOSPLUS
LDOS
Apple ProDOS
Tele DOS
Commodore 64

An operating system, in very simple terms, is a library of computer programs, each of which provides the operating instructions that drive each basic computer system operating activity (Fig. 1-3). The computer programs in a Unix operating system contain the instructions for such basic operating activities as:

- Controlling the movement of data into and out of the computer's memory, called *random access memory* or *RAM*.
- Supervising the operation of the computer's central processing unit (CPU), keeping all programs isolated from one another, so that one program will not interfere with the operation of another.
- Sending data to the peripheral devices connected to the computer, such as to the computer's printer for a printout, or to the terminal monitor, for visual display, etc.
- Managing the storage of data in the computer's *auxiliary memory* (disk and tape drives), for example making sure that one file is not inadvertently erased the information in another.
- Running the other computer programs installed in the computer, such as commands, word processors, accounting programs, database programs, etc.
- By means of a security system, keeping all data and programs uniquely identified, keeping unauthorized individuals from gaining access to the system or users from tampering with one another's files, and keeping a program that has not been debugged from running rampant through the computer and destroying other files and programs.