UNIX SYSTEM ADMINISTRATION
Kochan, Stephen G.
Wood, Patrick H.
First Edition Eight Printing
Hayden Books A Division of Macmillan Computer Publishing
1986/1990
# of Pages: 320
English
0-8104-6289-3

Notes

PREFACE

If you run a UNIX operating system or compatible, this book is for you.

It doesn't matter whether that system is a tiny IBM PC/XT running VENIX or XENIX, and you're the only user/ It doesn't matter whether it's a VAX-11/780 under UNIX System V, with a population of 75, and 8 modern lines plus 5 printers. *Every* UNIX system needs someone to take care of it, look after its needs, and sometimes give it love and affection. This someone is usually called the System Administrator (or SA) – a rather formal sounding title for such a caring position. This someone is probably you!

We'll show you how to do it all as painlessly as possible. Together, we've pulled SA duty on systems ranging from the tiny AT&T UNIX PC to giant Amdahl mainframes, including just about everything in between.

...

Counting individual manufacturers' changes, there are literally dozens of variations of the UNIX system available on the market today, even though only about 10 official versions exist. These include the Sixth, Seventh, and Eight Editions (often referred to as Versions 6,7, and 8), PWB, System III, System V, System V Release 2, System V Release 3, Berkeley 4.1, Berkeley 4.2, and Berkeley 4.3 This list does not include variants of the above such as PC/IX, UniPlus+, Ultrix, VENIX, XENIX, or any of the UNIX look-alikes such as Idris or Coherent.

Due to the many variations in Bell Labs and commercial UNIX versions, your system may differ in its commands or locations of commands as presented in this book. Since we can't possible cover everything, out of necessity we have to limit this book to descendant of the original AT&T Bell Labs UNIX System. We have used the AT&T System 5 and 5.2 manuals with occasional reference to other well known ports including Unisoft's. While you will undoubtedly, learn enough here to be able to keep a Berkeley (BSD) system running, there are enough differences between BSD and AT&T UNIX that you could possibly get confused. If you are trying to run a BSD system using this book, keep an open mind and prepare yourself for a good number of files to be in different places.

...

Certain UNIX traditions, such as leaving the vowels out of command names, are not always for the best. Even so, using a traditional UNIX method to solve system administration problems is a good idea because it is almost always the best method, and it's usually easier to maintain afterward than other methods.

...

Imagine the fun that would develop if you went to another machine and found the the SA there had changed all the major command names and directories. While the fact that this can be done shows the flexibility of the UNIX system, such a change would not be to anyone's benefit.

...

The UNIX system furnished with your machine is called the standard distribution. The system administrator's tools are always in **/etc**, major user commands in **/usr** and so on.

Page 1-2:

THE SYSTEM ADMINISTRATOR'S OVERVIEW OF UNIX

What's So Special About UNIX?

UNIX is a very special operating system. It is not a manufacturer's system, steeped in hardware dependence and tied by a binary umbilical cord to the whims and fancies of one company. Neither is it limited in application – it is an ideal system for writing, programming, and communication. It is easily the most complete operating system in existence today.

Because UNIX is such a comprehensive system, a well-rounded approach to UNIX system administration is necessary. Remember the parable about the blind men who examined the elephant? One felt the trunk and concluded that an elephant was shaped ike a serpent. Another felt the legs and determined that elephant were shaped like tree trunks. Both were party right, but neither understood the whole elephant. In this chapter we're going to look at UNIX from a lot of differrnet angles – UNIX past to UNIX present, a user's point of view and a system administrator's perspective, UNIX internals, and hardware and software considerations. In short, we're going to try to get an overall view of the beast.

What is an Operating System?

The stock description of an operating system is that it is a large body of software that acts as a "traffic cop" and directs the flow of information to and from the hardware. Which is techincally true, it brings us back to the elephant-as-serpent analogy. Any operating system, especially one as powerful as UNIX, is made of a number of different parts:

- *Device drivers*, or programs that stand at the lowest level and allow control of actual hardware devices. They negotiate between the kernel and the hardware bus.
- *A scheduler*, that decides which user programs are to run, when, and for how long. On a primitive operating system such as CP/M-80 or MS-DOS, there is no scheduler because only one program runs under the control of the operating system at a time.
- *Memory management*, which determines how much memory to allocate to each program. If not enough memory is available to run a given program, it will move other programs (or parts of them) to temporary disk storage as necessary. This is know as *swapping* (or *paging*).
- *The file system*, a structure used to locate and store programs and files on disk.

- *System programs*, the software accessible directly by users to allow manipulation of files and devices in various ways for getting basic work done. These may include utility programs, text editors, language compilers, debuggers, and shells.

Shells

A *shell* is a command processor that is the actual interface between the UNIX kernel and the user. It is the shell that runs commands when you type their names, expands "wildcard" character such as ***** and **?**, and takes care of redirecting input and output. There are two common shells available. The first is the *Bourne shell* (written by Stephen R. Bourne), which is the "standard" shell supplied with most UNIX systems. Bourne shell command syntax is reminiscent of the Algol language.

The *C shell* was developed by William Joy at the University of California at Berkeley. As the name implies, command programs written in C shell resemble programs written in the C language.

...

The Kernel

What is generally referred to as the UNIX *kernel* include the scheduler, memory, management routines, and device drivers, as well as a large number of built-in system functions that are hidden from the casual user. These function, known as system calls, are actually sub-programs or primitive functions that are accessible to user programs. The kernel is simply a program called **unix** (or **xenix** or **venix**) that is loaded in when the system is started and runs continuously until shutdown.

Page 3:

C – The Key to UNIX Portability

The traditional view of UNIX is a set of layered spheres, each sphere representing a different level of the system.

...

UNIX is portable because most of it is written in the high-level C language with only a small portion in the native assembler code. A device driver is divided into two parts; the top end written in C and the bottom end is written in both C and assembler. Surprisingly, 90% of the UNIX kernel is written in C. Assembler sections of the kernel are kept to a minimum, each module consisting of only a few lines of code called by a larger C module. Addressing registers, saving and restoring context (the CPU registers and immediate stack) is done from C, not assembler. C is the key to UNIX portability.

...

Originally, the only version of UNIX outside of Bell Laboratories was the DEC PDP-11 port. Now there are many AT&T-supported commercial UNIX ports, including the DEC VAX machines, the Motorola 68000 family of processors, the Intel 80286 family of processors, the National Semiconductor 16032 and 32032 family of processors, and Amdahl's UTS UNIX running on IBM architecture machines. At the small end of the scale is the little IBM PC and its arms of UNIX derivatives, look-alike, and work-alike systems (VENIX, XENIX, and Idris, to mention three).

Page 5:

Thus, using UNIX you can write a book, typeset it, and send it to a publisher, ready for final printing.

Page 10:

The Standardization of UNIX at 5.2

UNIX emerged commercially from AT&T as Version 7. It continued to develop, but other versions of UNIX quickly proliferated. Berkeley UNIX split away at Version 6. Some UNIX look-alikes stopped at System III. As a system administrator, you should be aware that there are many versions of UNIX out there at various stages of development.

At this writing, UNIX System V Release 2 (SVR2 or just UNIX 5.2) is the current AT&T standard. At&T paid for ports to all major processors and architectures, and developed a standardization suite to assure uniformity.

...

AT&T has spent millions of dollars advertising System V as the standard for the UNIX system.

...

Since it isn't tied to the volatile personal computer market as much as to the burgeoning business sector, UNIX has a bright future.