**Chapter 1**

**UNIX Operating System**

**1. Overview**

The UNIX operating system was designed to let a number of programmers access the computer at the same time and share its resources. The operating system coordinates the use of the computer's resources, allowing one person, for example, to run a spell check program while another creates a document, lets another edit a document while another creates graphics, and lets another user format a document -- all at the same time, with each user oblivious to the activities of the others.

The operating system controls all of the commands from all of the keyboards and all of the data being generated, and permits each user to believe he or she is the only person working on the computer. This real-time sharing of resources makes UNIX one of the most powerful operating systems ever.

Although UNIX was developed by programmers for programmers, it provides an environment so powerful and flexible that it is found in businesses, sciences, academia, and industry. Many telecommunications switches and transmission systems also are controlled by administration and maintenance systems based on UNIX. While initially designed for medium-sized minicomputers, the operating system was soon moved to larger, more powerful mainframe computers. As personal computers grew in popularity, versions of UNIX found their way into these boxes, and a number of companies produce UNIX-based machines for the scientific and programming communities.

**2. Uniqueness of UNIX**

The features that made UNIX a hit from the start are:

* Multitasking capability
* Multi-user capability
* Portability
* UNIX programs
* Library of application software

**2.1. Multitasking**

Many computers do just one thing at a time, as anyone who uses a PC or laptop can attest. Try logging onto our company's network while opening the browser while opening a word processing program. Chances are the processor will freeze for a few seconds while it sorts out the multiple instructions. UNIX, on the other hand, lets a computer do several things at once, such as printing out one file while the user edits another file. This is a major feature for users, since users don't have to wait for one application to end before starting another one.

**2.2. Multi-user**

The same design that permits multitasking permits multiple users to use the computer. The computer can take the commands of a number of users -- determined by the design of the computer -- to run programs, access files, and print documents at the same time. The computer can't tell the printer to print all the requests at once, but it does prioritize the requests to keep everything orderly. It also lets several users access the same document by compartmentalizing the document so that the changes of one user don't override the changes of another user.

**2.3. System portability**

A major contribution of the UNIX system was its portability, permitting it to move from one brand of computer to another with a minimum of code changes. At a time when different computer lines of the same vendor didn't talk to each other -- yet alone machines of multiple vendors that meant a great savings in both hardware and software upgrades. It also meant that the operating system could be upgraded without having all the customer's data inputted again. And new versions of UNIX were backward compatible with older versions, making it easier for companies to upgrade in an orderly manner.

**2.4. UNIX tools**

UNIX comes with hundreds of programs that can divided into two classes:

* Integral utilities that are absolutely necessary for the operation of the computer, such as the command interpreter, and
* Tools those aren’t necessary for the operation of UNIX but provide the user with additional capabilities, such as typesetting capabilities and e-mail.

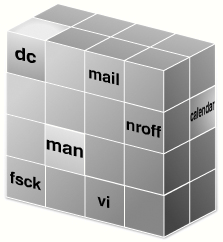


Figure 1.1 - UNIX Modular Structure

Tools can be added or removed from a UNIX system, depending upon the applications required.

**2.4.1. UNIX Communications**

E-mail is commonplace today, but it has only come into its own in the business community within the last 10 years. Not so with UNIX users, who have been enjoying e-mail for several decades. UNIX e-mail at first permitted users on the same computer to communicate with each other via their terminals. Then users on different machines, even made by different vendors, were connected to support e-mail. And finally, UNIX systems around the world were linked into a world wide web decades before the development of today's World Wide Web.

**2.5. Applications libraries**

UNIX as it is known today didn't just develop overnight. Nor were just a few people responsible for it's growth. As soon as it moved from Bell Labs into the universities, every computer programmer worth his or her own salt started developing programs for UNIX. Today there are hundreds of UNIX applications that can be purchased from third-party vendors, in addition to the applications that come with UNIX.

**3. How UNIX is organized**

The UNIX system is functionally organized at three levels:

* The kernel, which schedules tasks and manages storage;
* The shell, which connects and interprets users' commands, calls programs from memory, and executes them; and
* The tools and applications that offer additional functionality to the operating system

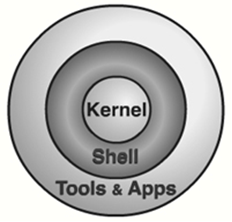


Figure 1.2 - Parts of the UNIX System

The three levels of the UNIX system:

* Kernel
* Shell
* Tools & Applications

**3.1. The kernel**

The heart of the operating system, the kernel controls the hardware and turns part of the system on and off at the programmer’s command. If we ask the computer to list (ls) all the files in a directory, the kernel tells the computer to read all the files in that directory from the disk and display them on our screen.

**3.2. The shell**

There are several types of shell, most notably the command driven Bourne Shell ($sh) and the C Shell($csh) (no pun intended), and menu-driven shells that make it easier for beginners to use. Whatever shell is used, its purpose remains the same -- to act as an interpreter between the user and the computer. The shell also provides the functionality of "pipes," whereby a number of commands can be linked together by a user, permitting the output of one program to become the input to another program.

**C Shell**

The C shell is a command processor that's typically run in a text window, allowing the user to type commands which cause actions. The C shell can also read commands from a file, called a script. Like all Unix shells, it supports filename wildcarding, piping, here documents, command substitution, variables and control structures for condition-testing and iteration. What differentiated the C shell, especially in the 1980s, were its interactive features and overall style. Its new features made it easier and faster to use. The overall style of the language looked more like C and was seen as more readable.

**Korn Shell**

The Korn shell (ksh) is a Unix shell which was developed by David Korn (AT&T Bell Laboratories) in the early 1980s and announced at Toronto USENIX on July 14 1983. ksh is backwards-compatible with the Bourne shell and includes many features of the C shell as well, such as a command history, which was inspired by the requests of Bell Labs users.

The main advantage of ksh over the traditional Unix shell is in its use as a programming language. Since its conception, several features were gradually added, while maintaining strong backwards compatibility with the Bourne shell.

**Bash Shell**

Bash is a Unix shell. it stands for Bourne-again shell, Bash is a POSIX shell with a number of extensions. It is the shell for the GNU operating system from the GNU Project. It can be run on most Unix-like operating systems. It is the default shell on most systems built on top of the Linux kernel as well as on Mac OS X and Darwin.

**3.3. Tools and Applications**

There are hundreds of tools available to UNIX users, although some have been written by third party vendors for specific applications. Typically, tools are grouped into categories for certain functions, such as word processing, business applications, or programming.

**4. Type of Unix OS’s**

