

UNIX Operating System – The Powerful and Stable Operating System

Subhashini P

Asst. professor, t. john college

Abstract- UNIX is a general-purpose, multi-user, multi-tasking interactive operating system for the Interdata 8/32 computers and the Digital Equipment Corporation PDP-11. The UNIX operating system has become quite famous since its beginning in 1969, running on machines of variable processing power from microprocessors to mainframes and providing a common execution environment setting across them. The system is split into two elements. The Primary part consists of programs and services that have made the UNIX OS environment so famous and popular. The second part consists of the operating system which supports these programs and services. UNIX also provides sufficient mechanisms for security that have to be configured and administered. It provides backup and restore methods in case of system crash. Unix also facilitates the users to send and receive messages and mails which contributes to Unix system communication.

Index Terms- : Unix, Multi-user, Shell, Kernel, Mach Kernel.

I. INTRODUCTION

UNIX and C language have modified the method people used and learned to program with computers. Even though technology changes quick, certain approaches to technology usually stay unchanged. UNIX is one in every of them. Even though UNIX OS was created by the scientific and engineering community, its robustness led to its widespread acceptance in the business market. The UNIX operating system was designed to let variety of programmers to access the system at the same time and also share its resources.

In general, the UNIX operating system consists of three components: the kernel, the shell, and the programs.

1.1 The kernel

If we imagine the UNIX operating system in forms of layers, the lowest layer is the kernel. It interfaces

directly with the computer hardware and is accountable for managing and allocating the resources available to programs. It allocates processor time and memory to every program that is run [1]. The kernel also conjointly provides an interface to programs where they will access files, devices, and network.

1.2 The shell

The shell acts as an associate interface between the user and the kernel. Once a user logs in, the corresponding login program checks the username and password, and then starts another program known as shell. The shell is nothing but a command line interpreter (CLI). It interprets the commands which the user types and executes them [1]. The commands are nothing but programs. Once the programs terminate, the control is returned back to the shell and the user receives another prompt on the system (\$), indicating that the user can go with another command. Here are few of the features:

1) Tab Completion – When we type part of the name of a command, filename or directory and press the [Tab] key, the shell can complete the rest of the name automatically. Suppose, if the shell finds more than one name which begins with those same letters you have typed, it will show you all of the possibilities starting with that combination.

2) History - The shell will always keep a list of the commands you have typed in. Suppose, if you want to repeat a command, then up and down arrow keys can be used to scroll up and down the list.

1.3 The programs

One of the important features of UNIX OS is that it consists of a variety of small programs to meet various needs. Typically, every of those programs does one thing and they do it well. Generally, these

programs operate on top of the shell, but they also additionally interface directly with the kernel.

II. HISTORY

Unix was first developed in Bell Laboratories of AT & T. In 1960's, AT & T's along with MIT and GE company jointly worked on a project called Multics. The motive of this project was to develop a new operating system which has multi-user, multi-tasking and multi-hierarchy file system. But due to insufficient growth of the project, AT & T called off the Multics project. Later in 1969, Dennis Ritchie and Ken Thompson developed an operating system called UNICS. Since this was developed in assembly language, it was not portable [2].

Unics was invented by Brian Kernighan, which was later changed to name Unix. To make this OS portable, Thompson developed Unix in B language. Later, Dennis Ritchie rewrote Unix in C language. AT & T developed most of the Unix versions. During 1980's, Microsoft built its first Unix system for a 16-bit microcomputer. In 1989, Santa Cruz Operations developed their own version of OS called SCO Unix. In 1991, Linus from Finland, developed a system called Linux. It was based on Unix. Now, Linux is one of the operating system which is most popular and widely used.

III. BASICS OF UNIX OPERATING SYSTEM

The Unix shell is a command line interpreter or a command programming language that gives an interface to the Unix operating system. It contains many mechanisms that are found in algorithmic languages like control-flow variables, primitives, and parameter passing. Constructs like for, if, while and case are also accessible. Two-way communication is provided between the shell and commands. The environment that the commands run can be changed by the shell. Pipes can also be used which redirects the output of first command as input to the second command.

Since Unix OS was written in c language, it makes it easy to read, understand, change, and move to different systems. Ritchie estimates that the primary system in C was twenty to forty percent larger and slower as a result of it had been not written in assembly language, however the benefits of

employing a higher-level language outweigh the disadvantages. it's an easy programme that has the facility to produce the services that users wish. It provides primitives that let advanced programs to be designed from easier programs. It uses a hierarchical file system that enables straightforward maintenance and economical implementation. It uses a regular format for files, the computer memory unit stream, creating application programs easier to jot down. It provides an easy, consistent interface to peripheral devices. It's a multi-user, multi-process system; every user will execute many processes at the same time. It hides the machine design from the user, creating it easier to jot down programs that run on completely different hardware implementations.

Features of UNIX:

The following are the advantages of Unix Features.

Portability:

The system is written in high level programming language creating it easier to read, understand, change and, thus move to different machines. The code is modified and complied on a brand new machine. Customers will then select from a good form of hardware merchants while not being fastened in with a specific vendor.

Machine-independence:

The System hides the machine design from the user, creating it easier to put in writing applications that may run on micros and mainframes.

Multi-Tasking:

Unix could be a powerful multi-tasking operating system; it means that once a active task in in method, there is a cooccurring background method operating too. UNIX operating system handles these active and background threads with efficiency and manages the system resources during a fair-share manner.

Multi-User:

UNIX could be a multi-user system designed to support a group of users at the same time. The system permits for the sharing of process power and peripheral resources, at a same time providing wonderful safety features.

Hierarchical File System:

UNIX uses a stratified file structure to store info. This structure has the most flexibility in grouping info in an exceedingly approach that reflects its natural state. It permits for straightforward maintenance and economical implementation.

UNIX shell:

UNIX includes a easy computer programme referred to as the shell that has the ability to supply the services that the user needs. It protects the user from having to understand the inner hardware details.

Pipes and Filters:

UNIX has facilities referred to as Pipes and Filters which enable the user to make complicated programs from easy programs.

Utilities:

UNIX has over two hundred utility programs for varied functions. New utilities are often designed effortlessly by combining existing utilities.

Software Development Tools:

UNIX offers a superb style of tools for computer code development for all phases, from program editing to software maintenance.

IV. ARCHITECTURE OF UNIX OPERATING SYSTEM

It has been noted that the UNIX operating system supports the illusions that the file system has "places" which processes have "lite." The 2 entities, files and processes, are the 2 central ideas within the UNIX operating system model. System calls appear as if normal function calls in C programs, and libraries will map these function calls to the primitives required to enter the OS. Assembly level language programs might invoke system calls directly while not a system call library, however. Programs oftentimes use different libraries like the standard I/O library to supply a lot of subtle use of the system calls. The libraries are joined with the programs at compile time and are therefore a part of the user program for functions of this discussion. The figure shows that the internal part of the unix OS is the hardware which can be directly accessed only by the kernel. Any processing functions will be carried out by the kernel. The shell which is the outer part of the kernel acts as a command line interpreter, that takes the input from the users. Any commands that is given by the user will be interpreted by the shell and will be forwarded to the kernel for further processing which may require some hardware resources [3].

Processes move with the file system via a particular set of system calls, like open (to open a file for reading or writing), close, read, write, stat (query the attributes of a file), chown (change the owner of the

file), and chmod (change the access permissions of a file). The file system accesses file information employing a buffering mechanism that regulates information flow between the kernel and auxiliary storage devices. The buffering mechanism interacts with block I/O device drivers to initiate information transfer to and from the kernel. Device drivers are the kernel modules that manages the operator of peripheral devices. Block I/O devices are random access storage devices or else, their device drivers build them seem to be random access storage devices to the rest of the system.

For example, a tape driver could permit the kernel. to treat a tape unit as a random access memory device. The file system conjointly interacts directly with "raw" I/O device drivers while not the intervention of buffering mechanism. Raw devices, typically referred to as character devices, embrace all devices that aren't block devices. The process subsystem is chargeable for process synchronization inter process communication, memory management, and scheduling of process. The file system and therefore the process control subsystem will read all the executable files before it is loaded into memory for execution. Few system calls that are used to control processes are fork (which is used to create a new process called child process), exec (which starts executing the child process), wait (which makes the parent process to wait until the child process finishes its execution), exit (the system call generated once the child process completes its execution).

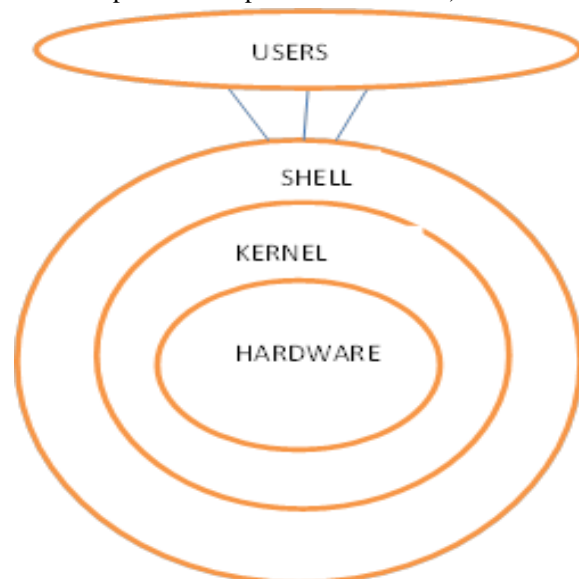


Fig. 4.1: Architecture of Unix Operating System

Unix File System: Unix Operating system has a hierarchical tree structure which has the top most directory as root directory. This root directory is indicated by a / (slash). This tree structure will have many children directories but has only one parent directory.

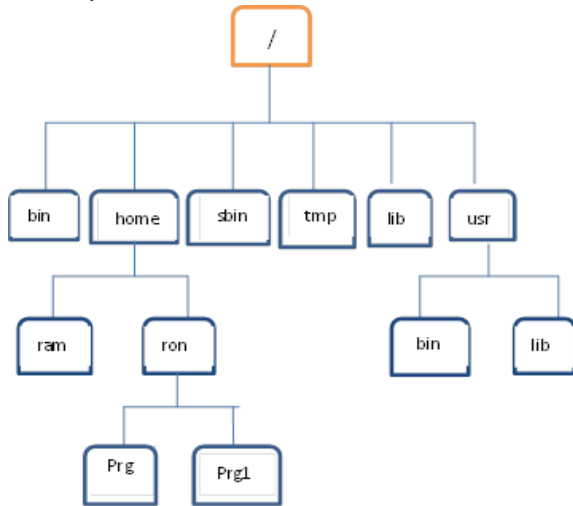


Fig 4.2: Unix File System

V. UNIX SECURITY

System security is very much essential especially for the multi-user systems like Unix OS. It is one of the responsibilities of the system admin to take care of the system security. The other name for system admin is super user or root user. The super user must be concerned on security issues such as preventing the system from unauthorized access, making a back up of file system in case of disk crash and having secure network communications.

The most important one here is password security. For every user of unix system, the super user creates an account, which has a username and password. Using this info, the users can login to the unix system. All the information about the users that was used to create the account will be stored in the file /etc/passwd. This file contains an entry for each corresponding user account in the system. This entry consists of 7 fields of information which is separated from each other by a colon (:). The 7 fields information are: login name, password, user identification number, group identification number, user's info, home directory and the shell [1].

Example:

Subha:X:250:182:Mrs.Subhashini.P:/home/subha:/bin/sh

Here, the password is X which shows that the encrypted password is stored in another separate file called as /etc/shadow file. If the encrypted password is stored in the same file, it is possible to obtain the password by decrypting it. So, for security reasons, the encrypted password will be stored in /etc/shadow file and this file can be read, modified only by the super user. This file has more information about the password. It has 8 fields of information about the password separated by a colon. The 8 fields of info are login name of the user, encrypted password, the date of the last change of password, the min number of days after which the user has to change his/her password, the max number of days after which the user must change his/her password, the number of days remaining after which the password date will expire, after the expiry date the number of days for the account to be disabled, the exact date on which the account will be disabled by default.

Any file that is very confidential can be stored on unix system by encrypting it using the crypt command. Crypt command can be used for encrypting and decrypting the file using a key. File encryption is based on a simple substitution method. It uses DES algorithm. After the file is been encrypted by the key, decryption of the file is done using crypt command with the same key.

```

$crypt abc <prog> prog.crypt
$crypt abc <prog.crypt> prog
    
```

The first command shows that the crypt command uses abc as the key to encrypt the file prog which results in the encrypted prog.crypt file. The second example shows that the crypt command is used with the same key to decrypt the file.

Crypt command can also be used without the key. In that case, the shell will accept the key interactively from the user and it will not be echoed.

```

$crypt <prog> prog.crypt
Key #not echoed
    
```

VI. BACKUP AND RESTORE

All data that we store on the system is important. Suppose if the data is lost due to some unfortunate situations like system crash etc, there should be a backup taken by the users. And store all the data in some backup device like floppy, tape or even hard

disk. The process of storing all the information at frequent intervals of time is known as backup or creating an archive. The process of acquiring back all the data from an archive is called restoring or extracting the data. To do this, Unix provides backup utilities such as tar and cpio command.

tar command (tape archive) is used to take backup and extract the files. cpio command (copy input-output) is used to copy the files to and from a backup device such as floppy disk, magnetic tape or the hard disk [4].

VII. UNIX SYSTEM COMMUNICATION

Unix also provides facilities to communicate with other users. There are two types of communication: offline and online communication. In offline communication, one user can send messages to the other user's terminal using the write command. But before using write command, the user's has to execute mesg command to give write permission. Once the write permission is given, the users can send and receive messages using write command. The two criteria for sending and receiving messages are both the users have to be logged in and mesg command has to be executed by the users. Online communication is possible through electronic mail. This is possible using mail command. Mail command is used both for sending and receiving mails from and to the users. Here, it is not required that the receiver has to be logged in when the sender sends the mail. If the receiver is not available at the moment, mail notification will be displayed after the user logs in. This is possible using mail command. Mail command is used both for sending and receiving mails from and to the users. Here, it is not required that the receiver has to be logged in when the sender sends the mail. If the receiver is not available at the moment, mail notification will be displayed after the user logs in.

VIII. CONCLUSION

This paper illustrates the importance of unix operating system. The two basic components of unix are kernel and shell. Kernel is referred to the heart of the operating system and shell is the command line interpreter. The commands given by the users are interpreted by the shell and forwards it to the kernel for further processing. The kernel will execute the

command and displays the output on the screen. The most important features of unix are multi-user, multi-tasking and multi-hierarchy file system. Unix provides backup facility using tar and cpio command. Unix also facilitates communication between the users using write command and mail command.

REFERENCES

- [1] https://www.researchgate.net/publication/236347617_Analysis_of_the_Unix_Operating_System_And_Improvement_Of_The_Password_Authentication_Technique.
- [2] <http://www.cs.miami.edu/home/schulz/CSC322.pdf>
- [3] <https://www.oreilly.com/openbook/utp/UnixTextProcessing.pdf>
- [4] https://www.tutorialspoint.com/unix/unix_tutorial.pdf

AUTHOR PROFILE

Subhashini P completed Master in Computer Network Engineering at Mangalore Institute of Technology. She is currently working as Assistant Professor in the department of Computer Application at T. John College, Bangalore.