

OVERVIEW OF UNIX

Pooja Nayak, Rajat Wason, Sahil Mudgal
*Student(b.tech VIth sem) Department of Computer science
Dronacharya College Of Engineering,Gurgaon-123506*

Abstract- This paper discusses about UNIX operating System's Components ,Architecture. The 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 • Security 1. Multitasking Capability Many computers do just one thing at a time, as anyone who uses a PC . "UNIX was the first operating system designed to run on 'dissimilar' computers by converting most hardware specific commands in machine language into an independent programming language called 'C ' ". UNIX was the basis of AT&T's telephone system and the government's wide area network system. Then it became the basis of communication between engineers and scientists, and eventually the basis of communication for everyone worldwide. It has held this remarkable spot since 1969.

I. INTRODUCTION

WHAT IS UNIX?

- UNIX is a computer operating system.
- An operating system is the program that controls all the other parts of a computer system, both the hardware and the software. It allocates the computer's resources and schedules tasks. It allows you to make use of the facilities provided by the system. Every computer requires an operating system.
- UNIX is a multi-user, multi-tasking operating system. Multiple users may have multiple tasks running simultaneously. This is very different from PC operating systems such as MS-DOS or MS-Windows (which allows multiple tasks to be carried out simultaneously but not multiple users).
- UNIX is a machine independent operating system. Not specific to just one type of computer hardware. Designed from the beginning to be independent of the computer hardware.

- UNIX is a software development environment. Was born in and designed to function within this type of environment.

WHY UNIX?

- Hardware independence
 - operating system code is written in C language rather than a specific assembly language
 - operating system software can be easily moved from one hardware system to another
 - UNIX applications can be easily moved to other UNIX machines. Porting is usually as simple as transfer of the source and a recompile
- Productive environment for software development
 - rich set of tools
 - versatile command language
- Distributed processing and multi-tasking.

UNIX COMPONENTS:

- **Kernel**
 - The core of the UNIX system. Loaded at system start up (boot). Memory-resident control program.
 - Manages the entire resources of the system, presenting them to you and every other user as a coherent system. Provides service to user applications such as device management, process scheduling, etc.
 - You do not need to know anything about the kernel in order to use a UNIX system. These details are provided for your information only.

- **Shell**

- Whenever you login to a Unix system you are placed in a shell program. The shell's prompt is usually visible at the cursor's position on your screen. To get your work done, you enter commands at this prompt.
- The shell is a command interpreter; it takes each command and passes it to the operating system kernel to be acted upon. It then displays the results of this operation on your screen.
- Several shells are usually available on any UNIX system, each with its own strengths and weaknesses.
- Each shell also includes its own programming language. Command files, called "shell scripts" are used to accomplish a series of tasks.

- **Utilities**

- UNIX provides several hundred utility programs, often referred to as commands.
- Accomplish universal functions
 - editing
 - file maintenance
 - printing
 - sorting
 - programming support
 - online info
 - etc.
- Modular: single functions can be grouped to perform more complex tasks.

II. UNIX COMMANDS

Files:

- **ls** --- lists your files
- **ls -l** --- lists your files in 'long format', which contains lots of useful information, e.g. the exact size of the file, who owns the file and who has the right to look at it, and when it was last modified.
- **ls -a** --- lists all files, including the ones whose filenames begin in a dot, which you do not always want to see.

There are many more options, for example to list files by size, by date, recursively etc.

- **more filename** --- shows the first part of a file, just as much as will fit on one screen. Just hit the space bar to see more or **q** to quit. You can use */pattern* to search for a pattern.
- **emacs filename** --- is an editor that lets you create and edit a file
- **mv filename1 filename2** --- moves a file (i.e. gives it a different name, or moves it into a different directory (see below)
- **cp filename1 filename2** --- copies a file
- **rm filename** --- removes a file. It is wise to use the option **rm -i**, which will ask you for confirmation before actually deleting anything.
- **diff filename1 filename2** --- compares files, and shows where they differ
- **wc filename** --- tells you how many lines, words, and characters there are in a file
- **chmod options filename** --- lets you change the read, write, and execute permissions on your files. The default is that only you can look at them and change them, but you may sometimes want to change these permissions. For example, **chmod o+r filename** will make the file readable for everyone, and **chmod o-r filename** will make it unreadable for others again. Note that for someone to be able to actually look at the file the directories it is in need to be at least executable.
 - **lprm jobnumber** --- remove something from the printer queue. You can find the job number by using **lpq**. Theoretically you also have to specify a printer name, but this isn't necessary as long as you use your default printer in the department.
 - **genscript** --- converts plain text files into postscript for printing, and gives you some options for formatting. Consider making an alias like **alias ecop 'genscript -2 -r \!* | lpr -h -Pvalkyr'** to print two pages on one piece of paper.

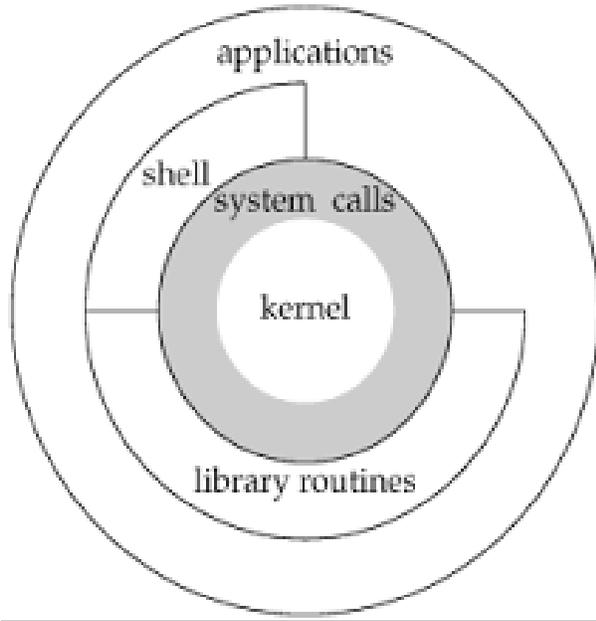
- **dvips filename** --- print **.dvi** files (i.e. files produced by LaTeX). You can use **dviselect** to print only selected pages.

Directories:

Directories, like folders on a Macintosh, are used to group files together in a hierarchical structure.

- **mkdir dirname** --- make a new directory
- **cd dirname** --- change directory. You basically 'go' to another directory, and you will see the files in that directory when you do 'ls'. You always start out in your 'home directory', and you can get back there by typing 'cd' without arguments. 'cd ..' will get you one level up from your current position.
- **pwd** --- tells you where you currently are.

III. UNIX ARCHITECTURE



Benefits for Application Developers

A single standard for the UNIX operating system means:

- Improved portability.
- Faster development through the increased number of standard interfaces.
- More innovation is possible, due to the reduced time spent porting applications.

Benefits for Users

The Single UNIX Specification will evolve and develop in response to market needs protecting users

investment in existing systems and applications. The availability of the UNIX system from multiple suppliers gives users freedom of choice rather than being locked in to a single supplier.

IV. CONCLUSION

The UNIX system has been around for a long time, and many people may remember it as it existed in the previous decades. Many IT professionals who encountered UNIX systems in the past found it uncompromising. While its power was impressive, its command-line interface required technical competence, its syntax was not intuitive, and its interface was unfriendly.

Moreover, in the UNIX system's early days, security was virtually nonexistent. Subsequently, the UNIX system became the first operating system to suffer attacks mounted over the nascent Internet. As the UNIX system matured, however, the organization of security shifted from centralized to distributed authentication and authorization systems.

Today, these perceptions are only of historical interest.

Now, a single Graphical User Interface is shipped and supported by all major vendors has replaced command-line syntax, and security systems, up to and including B1, provide appropriate controls over access to the UNIX system.

SUMMARY

When the history of the information age is written, the extraordinary dynamics of the UNIX system marketplace will be seen as playing an important role. The UNIX system was developed at just the right time and place to be the critical enabler for a revolution in information technology. Client/server architectures, the Internet, object databases, heterogeneous transaction processing, and Web computing all emerged on the shoulders of the UNIX system.

Most importantly, the UNIX system continues to be a driving force for innovation because of its commitment to standards. When proprietary differences are set aside, and with the wide implementation of the Single UNIX Specification they are set aside, suppliers compete by adding value. This fundamental tenet is the reason that the UNIX

system has thrived - and will continue to thrive in the years to come.

REFERENCES

1. www.unix-manuals.com
2. www.tldp.org/HOWTO/Online-Troubleshooting.../unixrefs.html
3. www.cs.jhu.edu/~joanne/unixRC.com
4. www.tutorialspoint.com/unix/unix-quick-guide.com
5. www.unix.stackexchange.com