

Sr. No: 11
Name: Soham Desai
XIE ID: 202003021

EXP.No. 1- Case Study: Brief History of UNIX, Unix Architecture; Installation of Unix Operating System

Objective: To understand architecture and installation of Unix Operating System.

Outcome: Understand the architecture and functioning of Unix .

OPERATING SYSTEM

An operating system (commonly abbreviated OS and O/S) is the software component of a computer system that is responsible for the management and coordination of activities and the sharing of the limited resources of the computer. It is a set of programs that provides the interface between user and computer hardware. Controlled allocation of the processors, memories, and I/O devices among the programs are allocated by the operating systems efficiently. So the Operating System is also known as resource allocator. Operating systems offer a number of services to application programs and users. Applications access these services through application programming interfaces (APIs) or system calls.

TYPES OF OPERATING SYSTEM

1. Single User - The system will have its own hard disk, memory, CPU and other resources all dedicated to a single user. E.g. MS-DOS

2. Multi User- The users have access to a multi-user system will have just a terminal and a keyboard. The other resources such as hard disk, printers are centrally located. The user is expected to simply hook onto his account, perform the work, disconnect and leave quietly.

E.g. UNIX

Unix-Linux History

In order to understand the popularity of Linux, we need to travel back in time, about 55 years ago...

Imagine computers as big as houses, even stadiums. While the sizes of those computers posed substantial problems, there was one thing that made this even worse: every computer had a different operating system. Software was always customized to serve a specific purpose, and software for one given system didn't run on another system. Being able to work with one system didn't automatically mean that you could work with another. It was difficult, both for the users and the system administrators. Technologically the world was not quite that advanced, so they had to live with the size for another decade. In 1969, a team of developers in the Bell Labs laboratories started working on a solution for the software problem, to address these compatibility issues.

They developed a new operating system, which was

1. Simple and elegant.
2. Written in the C programming language instead of in assembly code.
3. Able to recycle code.

The Bell Labs developers named their project "UNIX." The code recycling features were very important. Until then, all commercially available computer systems were written in a code specifically developed for one system. UNIX on the other hand needed only a small piece of that special code, which is now commonly named the kernel. This kernel is the only piece of code that needs to be adapted for every specific system and forms the base of the UNIX system. The operating system and all other functions were built around this kernel and written in a higher programming language, C. This language was especially developed for creating the UNIX system. Using this new technique, it was much easier to develop an operating system that could run on many different types of hardware. The software vendors were quick to adapt, since they could sell ten times

more software almost effortlessly.

Weird new situations came in existence: imagine for instance computers from different vendors communicating in the same network, or users working on different systems without the need for extra education to use another computer. UNIX did a great deal to help users become compatible with different systems. Throughout the next couple of decades the development of UNIX continued. More things became possible to do and more hardware and software vendors added support for UNIX to their products.

UNIX was initially found only in very large environments with mainframes and minicomputers (note that a PC is a “micro” computer). You had to work at a university, for the government or for large financial corporations in order to get your hands on a UNIX system. But smaller computers were being developed, and by the end of the 80s, many people had home computers. By that time, there were several versions of UNIX available for the PC architecture, but none of them were truly free and more important: they were all terribly slow, so most people ran MS DOS or Windows 3.1 on their home PCs. By the beginning of the 90s home PCs were finally powerful enough to run a full blown UNIX. While there was an academic UNIX- lookalike called Minix available at the time, its creator, Andrew S. Tanenbaum did not allow modifications that would make it more generally usable. He wanted his system to stay “clean”, since he created it in order to teach computer science with it.

Linus Torvalds, a young man studying computer science at the University of Helsinki, used the Minix, and when he felt too constrained by its limitation, he started to code his own UNIXlookalike operating system. From the start, it was Linus’ goal to have a free system that was completely compliant with the original UNIX. That is why he asked for POSIX standards, POSIX still being the standard for UNIX.

In those days plug-and-play wasn’t invented yet, but so many people were interested in having a UNIX system of their own, that this was only a small obstacle. New drivers became available for all kinds of new

hardware, at a continuously rising speed. Almost as soon as a new piece of hardware became available, someone bought it and submitted it to the Linux test, as the system was gradually being called, releasing more free code for an ever wider range of hardware. These coders didn’t stop at their PCs; every piece of hardware they could find was useful for Linux.

Two years after Linus’ post, there were 12000 Linux users. The project, popular with hobbyists, grew steadily, all the while staying within the bounds of the POSIX standard. All the features of UNIX were added over the next couple of years, resulting in the mature operating system Linux has become today. Linux is a full UNIX clone, fit for use on workstations as well as on middlerange and high-end servers. Today, a lot of the important players in the hardware and software market each have their team of Linux developers; at your local dealers you can even buy preinstalled Linux systems with official support—though there is still some hardware and software not supported.

UNIX FEATURES

- 1. Multi-user system**—Multi-user capability of UNIX allows several users to use the same computer to perform their tasks. Several terminals [Keyboards and Monitors] are connected to a single powerful computer [UNIX server] and each user can work with their terminals.
- 2. Multi-tasking system**—Multitasking is the capability of the operating system to perform various task simultaneously, i.e. a user can run multiple tasks concurrently.
- 3. Programming Facility**—UNIX is highly programmable, the UNIX shell has all the necessary ingredients like conditional and control structures, etc.
- 4. Security**—UNIX allows sharing of data; every user must have a single login name and password. So accessing another user’s data is impossible without his permission.

Sr. No: 11
Name: Soham Desai
XIE ID: 202003021

5. Portability—UNIX is portable because it is written in a high level language.

So, UNIX can be run on different computers.

6. Communication—UNIX supports communication between different terminals of the same server as well as between terminals on different servers. Apart from these features, UNIX has an extensive Tool kit, exhaustive system calls and Libraries and enhanced GUI

ORGANIZATION OF UNIX

The UNIX system is functionally organized at three levels :

1. The kernel, which schedules tasks and manages storage.
2. The shell (Command Interpreter), which connects and interprets users' commands, calls programs from memory, and executes them.
3. The tools and applications that offer additional functionality to the OS.

COMPONENTS OF UNIX OPERATING SYSTEM

Kernel—The kernel is the heart of the system, a collection of programs written in C that directly communicate with the hardware. It manages the system resources, allocates time between user and processes, decides process priorities, and performs all other tasks. It's that part of UNIX system that is loaded into memory when the system is booted. So we can define kernel as the Master program or often called the Operating system.

Shell—The user cannot directly interact with the kernel. During the login of the user, the kernel starts an interactive program for each user. This program is known as shell. It is actually the interface between the user and the kernel. When user gives some command the shell analyses those commands and passes them to the kernel. The kernel then submits these commands to hardware and the required actions are carried out.

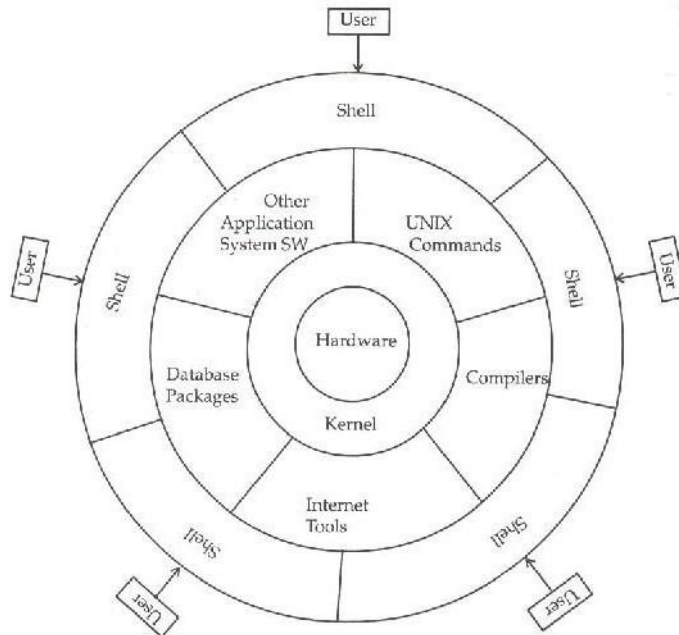


Fig 1.1: Structure of UNIX Operating System

Sr. No: 11
Name: Soham Desai
XIE ID: 202003021

FILES AND DIRECTORIES

A file is a collection of information that is assigned a name that is used to identify that file. The file is always stored in secondary storage medium. A directory is a special type file that contains a list of file names. All files are grouped together into directory for easier access. A directory can have one or more directories in it called sub-directories. In LINUX, files and directories are arranged in a hierarchical manner.

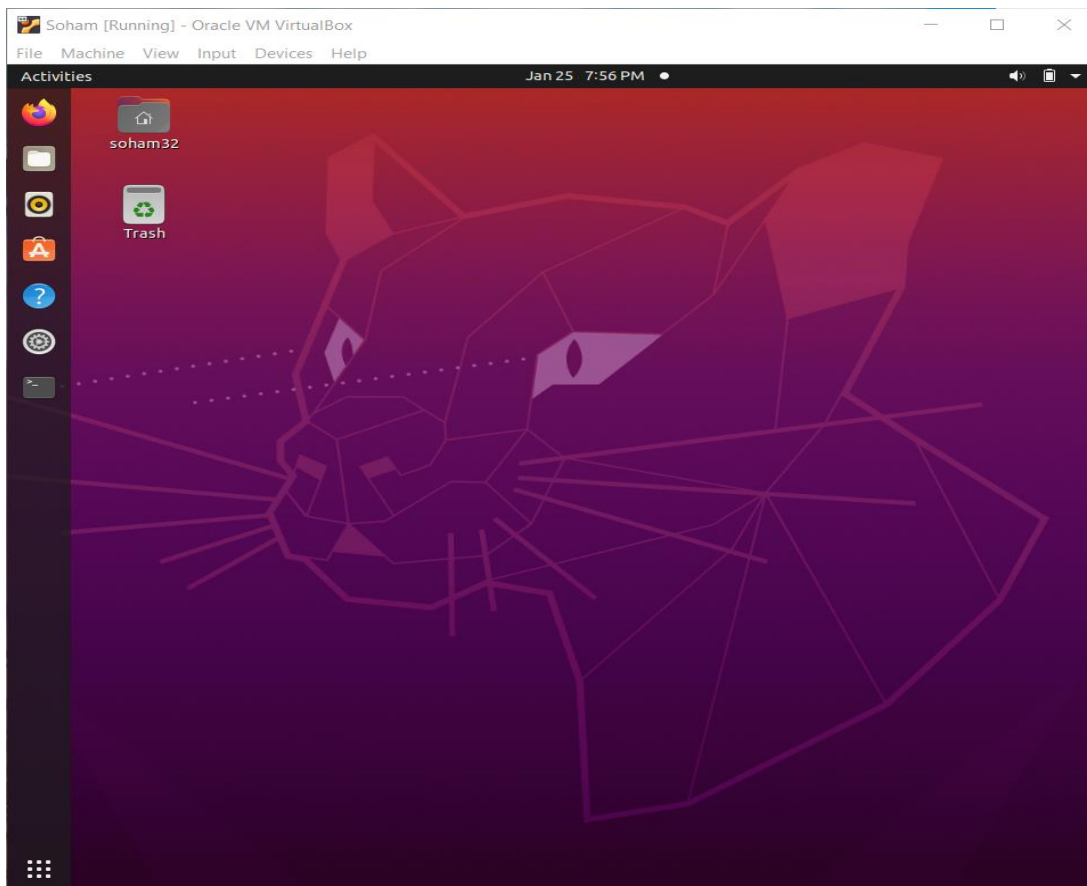
The files can be broadly classified as follows:

Ordinary files- Contains stream of data. All data, text, source programs, object and executable code, commands fall into this category.

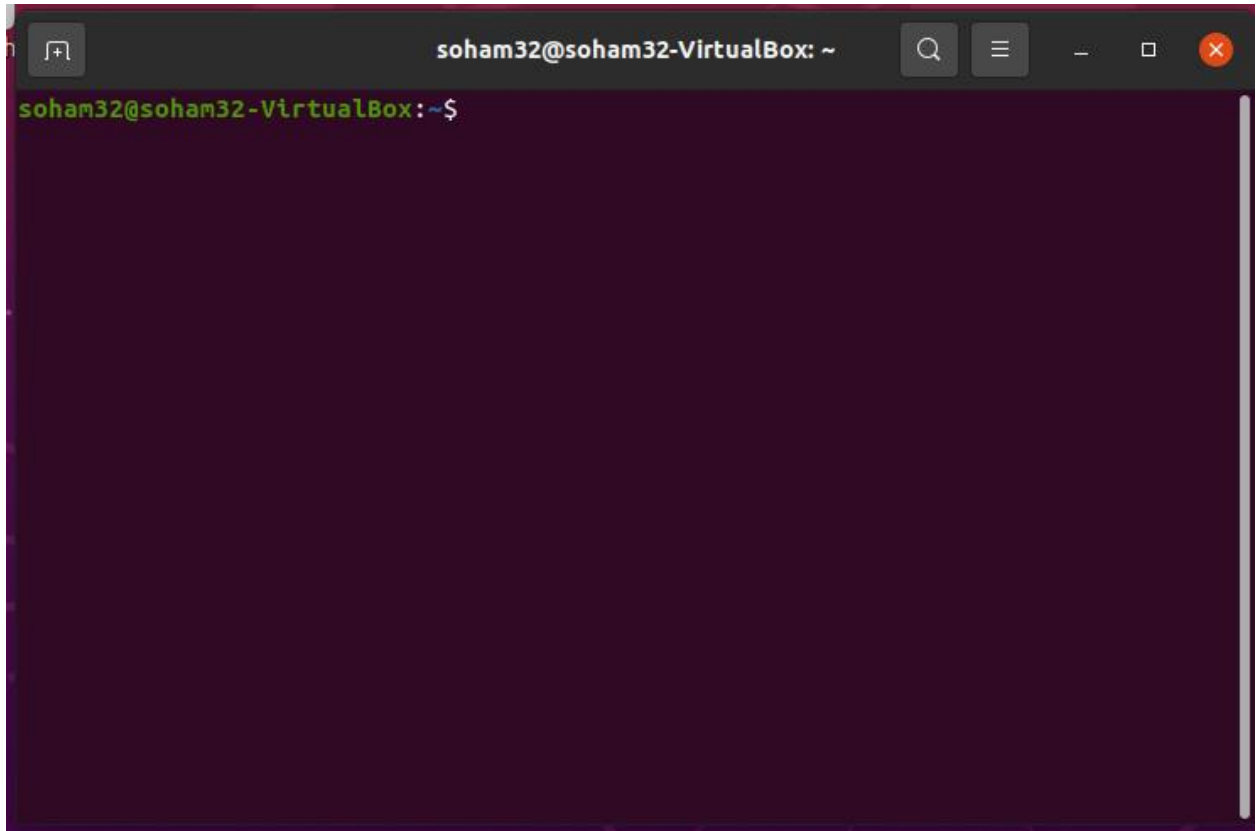
Directory files- Contains no external data. It contains an entry, name of the file and its identification number for each file and subdirectory under that directory. Directory files are not created by the user but by the UNIX system itself.

Device files - Even physical devices are treated as files. These are special in the sense that any output directed to it will be reflected onto the respective device.

Output:



Sr. No: 11
Name: Soham Desai
XIE ID: 202003021



Conclusion - Thus the study of Unix Operating System has been completed successfully.

Viva questions

1. Define Operating System.
2. What is kernel?
3. What is a UNIX shell?
4. What is the difference between multi-user and multi-tasking?
5. How is UNIX different from Linux?
6. List some features of UNIX.
7. What is piping?
8. How are the files and directories organized in UNIX?
9. How does the kernel differentiate device files and ordinary files?
10. What are the types of file system in UNIX?