

# CHAPTER - 1

## INTRODUCTION

### HISTORY OF LINUX:

In earlier days, computers were as big as houses or parks. So we can imagine how difficult it was to operate them. Moreover, every computer has a different operating system which made it completely worse to operate on them. Every software was designed for a specific purpose and was unable to operate on other computer. It was extremely costly and normal people neither can afford it nor can understand it.

In 1969, a team of developers of Bell Labs started a project to make a common software for all the computers and named it as 'Unix'. It was simple and elegant, used 'C' language instead of assembly language and its code was recyclable. As it was recyclable, a part of its code now commonly called 'kernel' was used to develop the operating system and other functions and could be used on different systems. Also its source code was open source. Initially, UNIX was only found in large organizations like government, university, or larger financial corporations with mainframes and minicomputers (*PC is a microcomputer*).

In eighties, many organizations like IBM, HP and dozen other companies started creating their own UNIX. It result in a mess of UNIX dialects (*vocabulary*). Then in **1983, Richard Stallman** developed **GNU project** with the goal to make it freely available UNIX like operating system and to be used by everyone. But his project failed in gaining popularity. Many other UNIX like operating system came into existence but none of them was able to gain popularity.

In **1991, Linus Torvalds** a student at the **University of Helsinki, Finland**, thought to have a freely available academic version of Unix started writing its own code. Later this project became the **Linux kernel**. He wrote this program especially for his own PC as he wanted to use **Unix 386 Intel computer** but couldn't afford it. He did it on **MINIX using GNU C compiler**. GNU C compiler is still the main choice to compile Linux code but other compilers are also used like Intel C compiler. He started it just for fun but ended up with such a large project. Firstly he wanted to name it as 'Freax' but later it became 'Linux'.

He published the Linux kernel under his own license and was restricted to use as commercially. Linux uses most of its tools from GNU software and are under GNU copyright. In 1992, he released the kernel under GNU General Public License.

Today, supercomputers, smart phones, desktop, web servers, tablet, laptops and home appliances like washing machines, DVD players, routers, modems, cars, refrigerators, etc. use Linux Operating System.

### INTRODUCTION OF LINUX:

In the simple language Linux is an operating system. We all are familiar with other operating systems like Microsoft windows, Apple Mac OS, iOS, Google android, etc. just like them Linux is also an operating system.

An operating system is a software that enables communication between computer hardware and software. It conveys input to get processed by the processor and brings output to the hardware to display it. This is the basic function of an operating system.

Linux is around us since mid-90s. It can be used from wristwatches to supercomputers. It is everywhere in our phones, laptops, PCs, cars and even in refrigerators. It is very much famous among the developers and normal computer users.

## **STRUCTURE OF LINUX OPERATING SYSTEM:**

An operating system is a collection of software, each designed for a specific function. Linux OS has following components:

### **1. Kernel:**

Kernel is the core of the operating system. It establishes communication between devices and software. Moreover, it manages the system resources. Basically it has four responsibilities:

#### **☑ Device Management:**

A system has many devices connected to it like CPU, memory device, sound cards, graphic cards, etc. A kernel stores all the data related to all the devices in device driver (without this kernel won't be able to control the devices). Thus kernel knows what a device can do and how to manipulate it to bring out the best performance. It also manages communication between all the devices. Kernel has certain rules that has to be followed by all the devices.

#### **☑ Memory Management:**

Another function that kernel has to manage is the memory management. Kernel keeps a track of used and unused memory and make sure that processes shouldn't manipulate data of each other using virtual memory address.

#### **☑ Process Management:**

In process management kernel assign enough time and gives priorities to processes before handling CPU to other process. It also deals with security and ownership information.

#### **☑ Handling System Calls:**

Handling system calls means a programmer can write a query or ask the kernel to perform a task.

### **2. System Libraries:**

System libraries are special programs that helps in accessing the kernel's features. A kernel has to be triggered to perform a task and this triggering is done by the applications. But applications must know how to place a system call because each kernel has a different set of system calls. Programmers have developed standard library of procedures to communicate with kernel. Each operating system supports these standards and then these are transferred to system calls for that operating system.

Most well-known system library for Linux is glibc (GNU C library).

### **3. System Tools:**

Linux OS has a set of utility tools which are usually simple commands. It is a software which GNU project has written and publish under their open source license so that software is freely available to everyone. With the help of commands we can access our files, edit and manipulate data in our directories or files, change location of files or anything.

### **4. Development Tools:**

With the above three components our OS is running and working. But to update our system we have additional tools and libraries. These additional tools and libraries are written by the programmers and are called tool chain. A tool chain is a vital development tool used by the developers to produce a working application.

### **5. End User Tools:**

These end tools make a system unique for a user. End tools are not required for the operating system but are necessary for a user. Some examples of end tools are graphic design tools, office suites, browsers, multimedia players, etc.

## **ADVANTAGES OF LINUX:**

### **1. Low cost:**

We don't need to spend time and money to obtain licenses since Linux and much of its software come with the GNU General Public License. We can start to work immediately without worrying that our software may stop working anytime because the free trial version expires. Additionally, there are large repositories from which we can freely download high quality software for almost any task we can think of.

### **2. Stability:**

Linux doesn't need to be rebooted periodically to maintain performance levels. It doesn't freeze up or slow down over time due to memory leaks and such. Continuous up-times of hundreds of days (up to a year or more) are not uncommon.

### **3. Performance:**

Linux provides persistent high performance on workstations and on networks. It can handle unusually large numbers of users simultaneously, and can make old computers sufficiently responsive to be useful again.

### **4. Network Friendliness:**

Linux was developed by a group of programmers over the Internet and has therefore strong support for network functionality; client and server systems can be easily set up on any computer running Linux. It can perform tasks such as network backups faster and more reliably than alternative systems.

### **5. Flexibility:**

Linux can be used for high performance server applications, desktop applications, and embedded systems. We can save disk space by only installing the components needed for a particular use. We can restrict the use of specific computers by installing for example only selected office applications instead of the whole suite.

## **6. Compatibility:**

It runs all common UNIX software packages and can process all common file formats.

## **7. Choice:**

The large number of Linux distributions gives us a choice. Each distribution is developed and supported by a different organization. We can pick the one we like best; the core functionalities are the same; most software runs on most distributions.

## **8. Fast And Easy Installation:**

Most Linux distributions come with user-friendly installation and setup programs. Popular Linux distributions come with tools that make installation of additional software very user friendly as well.

## **9. Full Use Of Hard Disk:**

Linux continues work well even when the hard disk is almost full.

## **10. Multitasking:**

Linux is designed to do many things at the same time; e.g. a large printing job in the background won't slow down our other work.

## **11. Security:**

Linux is one of the most secure operating systems. "Walls" and flexible file access permission systems prevent access by unwanted visitors or viruses. Linux users have to option to select and safely download software, free of charge, from online repositories containing thousands of high quality packages. No purchase transactions requiring credit card numbers or other sensitive personal information are necessary.

## **12. Open Source:**

If we develop software that requires knowledge or modification of the operating system code, Linux's source code is at our fingertips. Most Linux applications are Open Source as well.

## **DISADVANTAGES OF LINUX:**

### **1. No Standard Edition:**

There's no standard edition of Linux. Whereas Microsoft offers several different editions of each version of Windows, there are countless variations of Linux. For a new user it can be confusing to work out which is best for us.

### **2. Patchier Support for Drivers:**

Linux has patchier support for drivers (the software which coordinates our hardware and our operating system). This means we'll sometimes find it trickier to get a new device set up.

### **3. Not User Friendly:**

Linux is, for new users at least, not as easy to use as Windows. That's largely because Linux gives us more control, but does mean we'll have to spend some time getting used to the way it works.

#### 4. Hard to Get Desired Help:

Because Linux is neither as popular as Windows, nor a commercial product, support works in a different way. We may have to look harder to find the answer to a problem and, while Linux supporters are more likely to offer help, it may not always match our own level of technical understanding.

#### **CULTURE OF FREE SOFTWARE:**

The culture of free software development from which Linux has thrived (*flourished*) and will continue to thrive. The copyright for Fedora and Red Hat Enterprise Linux systems is covered under the GNU public license. That license, which most free software falls under, provides the following:

- ☑ **Author Rights:** The original author retains the rights to his or her software.
- ☑ **Free Distribution:** People can use the GNU software in their own software, changing and redistributing it as they please. They do, however, have to include the source code with their distribution (or make it easily available).
- ☑ **Copyright Maintained:** Even if we were to repackage and resell the software, the original GNU agreement must be maintained with the software. This means that all future recipients of the software must have the opportunity to change the source code, just as we did.

It is important to remember that there is no warranty on GNU software. If something goes wrong, the original developer of the software has no obligation to fix the problem. However, the Linux culture has provided resources for that event. Experts on the Internet can help us iron out our problems, or we can access one of the many Linux newsgroups or forums to read how others have dealt with their problems and to post our own questions about how to fix ours. Chances are that someone will know what to do, maybe even going so far as to provide the software or configuration file we need.

**NOTE:** *The GNU project uses the term free software to describe the software that is covered by the GNU license. Many Linux proponents tend to use the term open source software to describe software. Although source code availability is part of the GNU license, the GNU project claims that software defined as open source is not the same as free software because it can encompass semi-free programs and even some proprietary programs.*