

## Introduction

LINUX is a Unix-like and mostly POSIX-compliant computer operating system assembled under the model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released on 5 October 1991 by Linus Torvalds.

The Free Software Foundation uses the name GNU/Linux to describe the operating system. Linux is the first truly free Unix-like operating system. The underlying GNU Project was launched in 1983 by Richard Stallman. It began originally to develop a Unix-compatible operating system called GNU, intended to be entirely free software.

Many programs and utilities were contributed by developers around the world and, by 1991, most of the components of the system were ready. Yet, still missing was the kernel, which is called Linux, was combined afterwards with the GNU system to produce a complete free operating system. Linux Kernel was bundled with many softwares from various distributors and it gave rise to many flavors of LINUX.

## Features Of Linux OS:

1. Open Source- Free
2. Multitasking
3. Multi-User
4. Portable
5. Reliability
6. Scalability
7. Networking & Security

## File Directory

**/** This directory is called superuser home directory. It's at the top of file system structure. All other directories are mounted under it.

**/boot** This directory contains the kernel image file. It also contains the files related to booting the system such as bootloader.

**/etc** This contains the whole system configuration file.

**/home** This is the all users home directory.

**/mnt** This is a generic mount point under which you mount your filesystems such as cdrom, floppy.

**/proc** This is not a real file system, but a virtual file system. This directory is empty until the proc file system is mounted.

**/sys** Modern Linux distributions include a /sys directory as a virtual filesystem (sysfs, comparable to /proc, which is a procfs). It stores and allows modification of the devices connected to the system.

**/dev** This directory contains the devices nodes through which the operating system can access hardware and software devices on the system.

**/bin** This directory contains the command used by superusers and normal users.

**/sbin** This directory contains the command used by superusers only.

**/lib** This contains the library required for running the additional applications and running the Linux kernel. It also contains the library essential for binaries in /bin and /sbin.

## Basic Commands

root@haktuts:~\$ pwd

[pwd will show the present working directory]

root@haktuts:~\$ ls

[ls will show the list of files and directories of present working directory]

root@haktuts:~\$ ls b\*

[ls b\* will show the list of files and directories starting with b and ending with anything]

root@haktuts:~\$ ls -l

[listing of all files and directories along with attributes (i.e. more detailed formats like file permission, file size etc.)]

root@haktuts:~\$ ls -a

[listing of all files and directories, along with hidden files or files starting with periods (i.e. file starting with . like \*.bashrc)]

root@haktuts:~\$ ls -R

[shows the listing of file and directory in tree structure]

root@haktuts:~\$ cat Haktuts.txt

[users can use the command cat to see the preview of Haktuts.txt file]

root@haktuts:~\$ cat > Hak.txt

[used to create files]

root@haktuts:~\$ cat Haktuts.txt Hak.txt >>new.txt

[transfer the content of Haktuts.txt and Hak.txt to new.txt file]

root@haktuts:~\$ cd

[to change the directory]

root@haktuts:~\$ cd com

[go to directory name com]

root@haktuts:~\$ cd ..

[change the directory one level back]

root@haktuts:~\$ cd ../../

[change the directory two levels back]

root@haktuts:~\$ cd ~

[change the directory to home directory]

root@haktuts:~\$ Vi

[will create a blank file]

root@haktuts:~\$ Vi

[will create multiple files at once]

root@haktuts:~\$ Vi file1.txt

[to open file1.txt using Vi] [i- to insert text in file1.txt] [press Shift+: and then type wq to save and quit]

: q to exit without any changes in file

: x to save changes and exit]

root@haktuts:~\$ mkdir com

[will create a directory name com]

root@haktuts:~\$ mkdir com lap mob

[create multiple directories]

root@haktuts:~\$ mkdir -p com/lap/mob

[create a directory with their subdirectories]

```
root@haktuts:~$ cd -
[go to last working directory] [eg:currently user in /home/Haktuts/Desktop directory and user last
working directory is /root/Desktop. when a user types cd - the user goes directly to /root/Desktop
directory]
root@haktuts:~$ rm Haktuts.txt
[rm remove the file Haktuts.txt]
root@haktuts:~$ rmdir com
[rmdir removes the directory name called com – only if the com directory is empty]
root@haktuts:~$ rm -rf com
[rm -rf remove the directory and also the content/data available in the directory]
root@haktuts:~$ cp Hak.txt /root/Desktop
[copy the file Hak.txt to /root/Desktop Directory]
root@haktuts:~$ cp -r com /root/Desktop
[copy the directory com to /root/Desktop directory]
root@haktuts:~$ mv Haktuts.txt /home
[move the file Haktuts.txt to /home directory]
root@haktuts:~$ mv Hak.txt Hak2.txt
[rename the file Hak.txt to Hak2.txt]
```

## Advantages and Disadvantages

In the 1990s, Microsoft Windows dominated the market for desktop computers. Linux, however, is becoming a competitor for desktop operating systems. The main difference between Windows and open-source software is that the code for Windows is highly secret where its competitor's distribution is an open-source operating system that allows anyone to download it, change it, and contribute to the internal kernel.

### Advantages of Open-Source

Linux was one of the first open-source technologies, but many programmers have contributed and added software that's completely open-source for any user. This means that you can download the source code and change it any way you like. Some developers have restrictions on how you can distribute the code. For instance, some developers allow you to change the code, but you cannot distribute it for money.

One main advantage of [open-source](#) technologies such as Linux is the wide range of options available to users and the increased security. With Linux being open-source, several distributions are available to the end-user. Debian, Fedora, Ubuntu and Mint are just a few of the distributions available to end users, and these distributions are completely free to download.

Security is the other main advantage. Several whitehat hackers have contributed to the overall security of Linux, and by making the source available to anyone, security experts can help identify any main security flaws in the operating system. The advantage over operating systems such as Windows is that security flaws are caught before they become an issue for the public.

## **Disadvantages of Open-Source**

Because Linux does not dominate the market like Windows, there are some disadvantages to using the operating system. First, it's more difficult to find applications to support your needs. This is an issue for mostly businesses, but more programmers are developing applications that are supported by Linux. Many more applications are available for the working world compared to what was available a decade ago.

One main issue with Linux is drivers. Before you can install any hardware component in your computer, you must make sure the hardware has drivers available. Hardware manufacturers usually write drivers for Windows, but not all brands write drivers for Linux. This means that some of your hardware might not be compatible with Linux if you decide to switch.

Support for open-source can also be an issue. While there are plenty of Windows support people, Linux is not supported out-of-the-box. The way Linux distribution companies make money is through their support channels. This means that companies must pay fees for support, if they cannot solve an issue. However, there are plenty of forums and blogs that support Linux issues. If your company has a good Linux administrator, the administrator can typically find answers through one of these free channels without paying for support.

Before you decide on open-source technology, make sure you have the resources and personnel to support the software.

## **Preparing to Install Linux**

Presents information you need to know and tasks you need to perform before installing Linux. It helps you make certain that your IBM-compatible PC meets the minimum hardware requirements for Linux. It shows you how to document your system configuration so that you can respond to questions presented by the Linux install procedure. Finally, it shows you how to prepare your hard disk for Linux.

### **Minimum Hardware Requirements**

Linux supports a wide range of PC hardware; but not even Linux supports every known device and system. Your PC must meet certain minimum requirements in order to run Linux. The following sections present these minimum requirements; however, for the latest and most complete information, you should check the Debian Project web site at <http://www.debian.org/>. The Debian web site will also help you determine if Linux supports all the devices installed in your system.

### **Central Processing Unit (CPU)**

Linux does not support the Intel 286 and earlier processors. However, it fully supports the Intel 80386, 80486, Pentium, Pentium Pro, Pentium II, and Pentium III processors. Nevertheless, some users feel that their 80386 Linux systems respond sluggishly, particularly when running X. So, if

you want optimum performance, you should install Linux on a PC having an 80486 processor or better.

Linux also supports non-Intel processors such as the Cyrix 6x86 and the AMD K5 and K6. Most Linux users have systems that use Intel chips; if your system uses a non-Intel chip, you may find it more difficult to resolve possible problems.

## **Motherboard**

Linux supports the standard ISA, EISA, PCI, and VESA (VLB) system buses used on most IBM-compatible PCs. Linux recently gained support for IBM's MCA bus, used in IBM's PS/2 series of computers. However, at the time of this writing, Debian GNU/Linux does not yet support the MCA bus. If you have an IBM PS/2, you may be unable to install Debian GNU/Linux (check the Debian Project web site for the latest available information on support for the MCA bus).

Your motherboard should include at least 16 MB of RAM for optimum Linux performance. Some users have managed to coax Linux into working on systems with as little as 4 MB of RAM. However, if your system has less than 16 MB of RAM, you probably won't be pleased with its performance. If you plan to run X, you may wish to install more than 16 MB of RAM - perhaps 64 MB. Although X operates well with 16 MB of RAM, you can open more windows and switch between them more quickly if you have additional memory.

A handful of motherboards presents special problems when installing Linux. Generally, the problem stems from a bad BIOS, for which a fix is often available. Check the Debian Project web site for details.

## **Drives**

An anonymous wag once quipped that one can never be too thin, too rich, or have too much hard disk space. Fortunately, Linux is not too hungry for disk space. To install and use Linux, you should have at least 250 MB of free hard disk space. (The minimum is about 100 MB, but installing Linux on a system with so little disk space will compel you to omit many useful applications and will leave you with little room to work.)

More realistically, if you plan to use your Linux system as a workstation, you should have at least 600 MB of free disk space; if you plan to use your Linux system as a server, you should have at least 1.6 GB (1,600 MB) of free disk space.

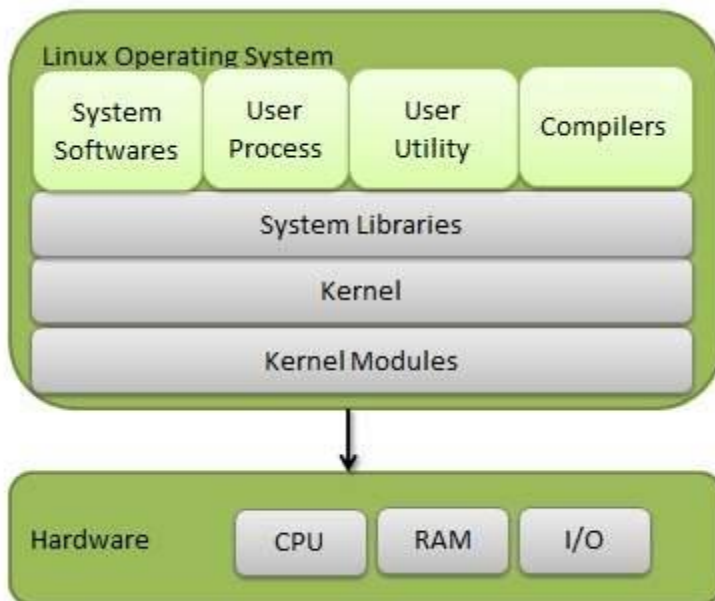
For convenient installation using the CD-ROM included with this book, your system should include an IDE or SCSI CD-ROM drive. It's also possible to install Linux from a PCMCIA CD-ROM drive, an FTP site, an NFS server, an SMB shared volume, or a hard drive. Consult the Debian Project web site for details.

Linux is one of popular version of UNIX operating System. It is open source as its source code is freely available. It is free to use. Linux was designed considering UNIX compatibility. Its functionality list is quite similar to that of UNIX.

### Components of Linux System

Linux Operating System has primarily three components

- **Kernel** – Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
- **System Library** – System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implement most of the functionalities of the operating system and do not requires kernel module's code access rights.
- **System Utility** – System Utility programs are responsible to do specialized, individual level tasks.



### Kernel Mode vs User Mode

Kernel component code executes in a special privileged mode called **kernel mode** with full access to all resources of the computer. This code represents a single process, executes in single address space and do not require any context switch and hence is very efficient and fast. Kernel runs each

processes and provides system services to processes, provides protected access to hardware to processes.

Support code which is not required to run in kernel mode is in System Library. User programs and other system programs works in **User Mode** which has no access to system hardware and kernel code. User programs/ utilities use System libraries to access Kernel functions to get system's low level tasks.

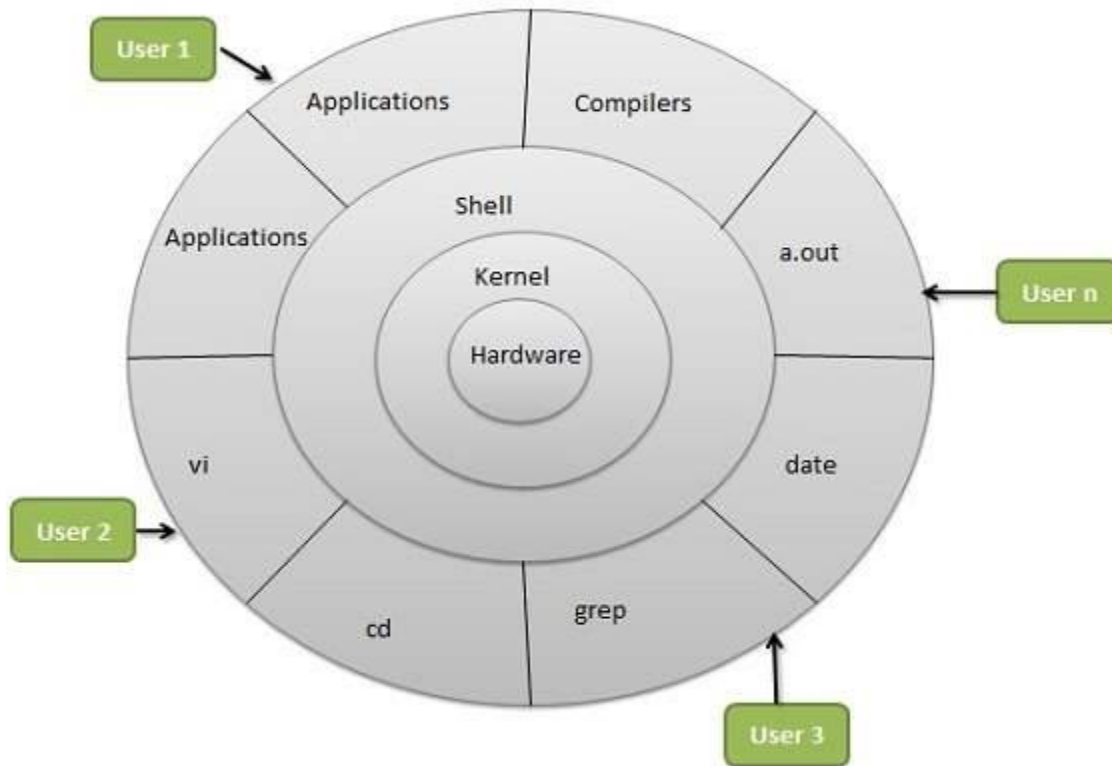
#### Basic Features

Following are some of the important features of Linux Operating System.

- **Portable** – Portability means software can works on different types of hardware in same way. Linux kernel and application programs supports their installation on any kind of hardware platform.
- **Open Source** – Linux source code is freely available and it is community based development project. Multiple teams work in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
- **Multi-User** – Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.
- **Multiprogramming** – Linux is a multiprogramming system means multiple applications can run at same time.
- **Hierarchical File System** – Linux provides a standard file structure in which system files/ user files are arranged.
- **Shell** – Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs. etc.
- **Security** – Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

#### Architecture

The following illustration shows the architecture of a Linux system –



The architecture of a Linux System consists of the following layers –

- **Hardware layer** – Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
- **Kernel** – It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
- **Shell** – An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
- **Utilities** – Utility programs that provide the user most of the functionalities of an operating systems.

## Commands

### 1. ls

The ls command - the list command - functions in the [Linux terminal](#) to show all of the major directories filed under a given file system. For example, the command:

#### ls /applications

...will show the user all of the folders stored in the overall applications folder.



The ls command is used for viewing files, folders and directories.

## 2. cd

The cd command - change directory - will allow the user to change between file directories. As the name command name suggest, you would use the cd command to circulate between two different directories. For example, if you wanted to change from the home directory to the Arora directory, you would input the following command:

**cd/arora/applications**

As you might have noted, the path name listed lists in reverse order.

Logically **cd/arora/applications** reads change to the arora directory which is stored in the applications directory. All Linux commands follow a logical path.

## 3. mv

The mv command - move - allows a user to move a file to another folder or directory. Just like dragging a file located on a PC desktop to a folder stored within the "Documents" folder, the mv command functions in the same manner. An example of the mv command is:

**mv/arora/applications/majorapps /arora/applications/minorapps**

The first part of the command **mv/arora/applications/majorapps** lists the application to be moved. In this case, arora. The second part of the command **/arora/applications/minorapps** lists where arora will be moved to - from majorapps to minorapps.

## 4. man

The man command - the manual command - is used to show the manual of the inputted command. Just like a film on the nature of film, the man command is the meta command of the [Linux CLI](#). Inputting the man command will show you all information about the command you are using. An example:

**man cd**

The inputting command will show the manual or all relevant information for the change directory command.

## 5. mkdir

The mkdir - make directory - command allows the user to make a new directory. Just like making a new directory within a PC or Mac desktop environment, the mkdir command makes new directories in a Linux environment. An example of the mkdir command

### **mkdir testdirectory**

The example command made the directory "testdirectory".

### **6. rmdir**

The rmdir - remove directory - command allows the user to remove an existing command using the Linux CLI. An example of the rmdir command:

### **rmdir testdirectory**

The example command removed the directory "testdirectory".

**It should be noted: both the mkdir and rmdir commands make and remove directories. They do not make files and they will also not remove a directory which has files in it. The mkdir will make an empty directory and the rmdir command will remove an empty directory.**

### **7. touch**

The touch command - a.k.a. the make file command - allows users to make files using the Linux CLI. Just as the mkdir command makes directories, the touch command makes files. Just as you would make a .doc or a .txt using a PC desktop, the touch command makes empty files. An example of the touch command:

### **touch testfile.txt**

The example touch command effectively created the file testfile.txt. As noted by the extension, the file created is a .txt or text file. To equate, a .txt file in Linux is akin to a .txt notebook file within a Windows or Mac OS.

### **8. rm**

The rm command - remove - like the rmdir command is meant to remove files from your Linux OS. Whereas the rmdir command will remove directories and files held within, the rm command will delete created files. An example of the rm command:

### **rm testfile.txt**

The aforementioned command removed testfile.txt. Interestingly, whereas the rmdir command will only delete an empty directory, the rm command will remove both files and directories with files in it. This said, the rm command carries more weight than the rmdir command and should be used with more specificity.

### 9. locate

The locate - a.k.a. find - command is meant to find a file within the Linux OS. If you don't know the name of a certain file or you aren't sure where the file is saved and stored, the locate command comes in handy. A locate command example:

```
locate -i *red*house**city*
```

The stated command will locate an file with the a file name containing "Red", "House" and "City". A note on the input: the use of "-i" tells the system to search for a file unpecific of capitalization (Linux functions in lower case). The use of the asterik "\*" signifies searching for a wildcard. A wildcard tells the system to pull any and all files containing the search criteria.

By specifying -i with wildcards, the locate CLI command will pull back all files containing your search criteria effectivley casting the widest search net the system will allow.

### 10. clear

The clear command does exactly what it says. When your Linux CLI gets all mucked up with various readouts and information, the clear command clears the screen and wipes the board clean. Using the clear command will take the user back to the start prompt of whatever directory you are currently operating in. To use the clear command simply type **clear**.