cd LINUX

(A complete Starter Guide to Linux)

Contents

INTRODUCTION: Linux is a family of open-source UNIX line Operating System. Linux is not an operating system; the kernel inside the operating system is known as the LINUX kernel.

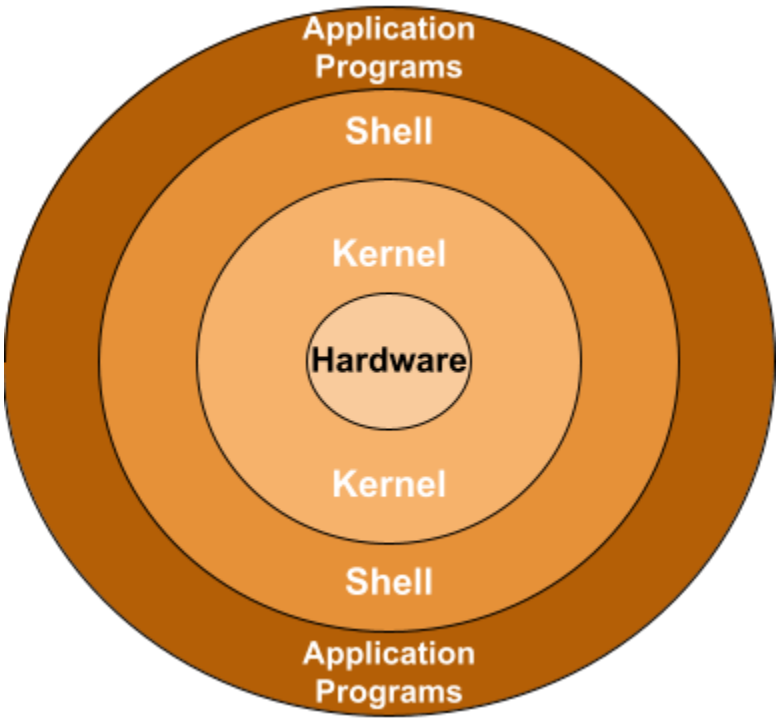
There are various distributions based on the LINUX kernel.

Like - Debian, Ubuntu, Archlinux, Red Hat Enterprise Linux, CentOS, Fedora

Structure of Linux Based Operating Systems: The Linux Based Operating Systems has a layered structure that consists of Hardware, Kernel, Shell, and applications.

- **Hardware:** This layer consists of all the physical parts of the computer. It includes CPU, Main Memory, Storage Devices (HDD, SSD), input and output devices and Networking Hardware.
- **Kernel:** Kernel is the core of all Linux-based Operating Systems. LINUX is the name of the kernel rather than the Operating Systems. Kernels include various subsystems or layers like Virtual File Systems (VFS), Network Stack, Device Drivers, System Call Interface, etc. Its main works are Process Management, Memory Management, Device Management, File System Management, Networking, Resource Allocation, Security, Inter-process Communication and many more.

- **Shell:** The shell is the command line interface that allows users to interact with the kernel. It takes input from the users as commands and then interprets those commands and transmits them to the kernel, which then performs the requested operations. There are also various types of shells. e.g.- Bourne Shell (sh), C Shell (csh), Korn Shell (ksh), Bourne Again Shell (bash), Z Shell(zsh).
- Applications: This layer contains all the software that users need to run to interact with the computer as well as to perform their computing tasks. All the Applications like Web Browsers, Text Editors, Media Players can be found at the Application Layers.



Usage:

- 96% of the top one million servers are powered by Linux.
- All the world's top 500 supercomputers use Linux
- The Android Operating System that powers approximately 85% of the world's mobile phones, is built on top of Linux.

Basic Features:

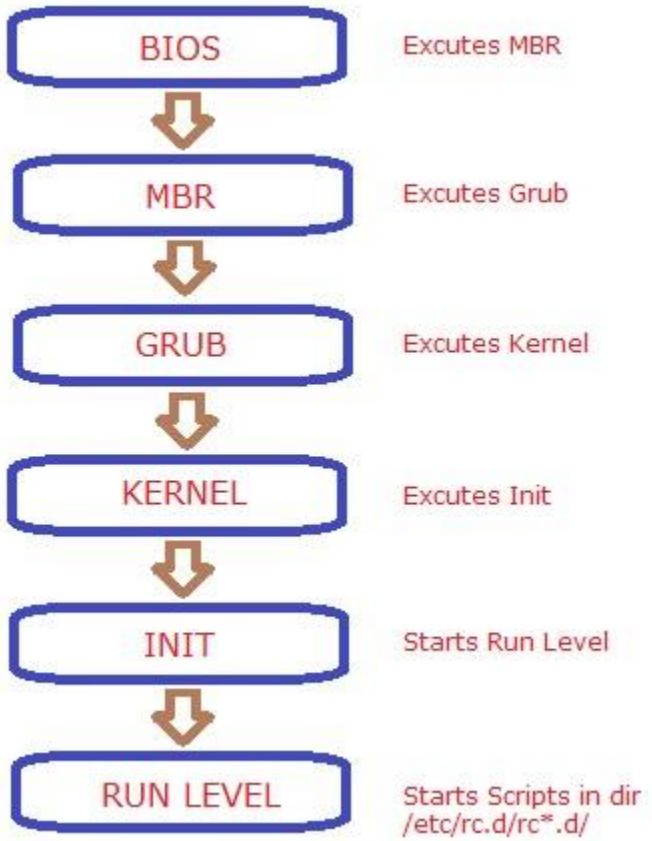
- **Portable:** Linux kernel and application programs support their installation on any kind of hardware platform. That's why Linux Software can work on different platforms in the same way.

- **Open Source:** Linux Source Codes are publicly available, and its development is based on community driven development projects. For this cause Linux is continuously evolving.
- **Hierarchical File System:** Linux provides a standard file structure in which system file or user files are arranged.
- **Multi-User:** Multiple users can access the same resources at the same time if the system is powered by a Linux powered distribution.
- **Multiprogramming:** Multiple applications can run at the same time on Linux.
- **Security:** As Linux is continuously evolving, Linux remains more secure and safe from known vulnerabilities.

Linux Booting Process: The Linux Booting process is a sequence of steps that starts after powering the computer and ends after the system is ready for user interaction.

1. **BIOS/UEFI Initialization:** After powering on the computer, BIOS (Basic Input/Output System) or UEFI (Unified Extensible Firmware Interface) firmware is loaded which performs a **Power-On-Self-Test (POST)** to check essential components like Main Memory, CPU and Connected Devices. Thereafter, the BIOS or UEFI initializes the hardware components.
2. **Bootloader Execution:** The BIOS/UEFI locates the bootloader, which loads the Linux kernel into Main Memory. A boot menu appears to select the preferred OS to boot into. After selecting the preferred OS, the kernel modules and the Initial RAM disk (**initramfs**) into memory.
   In most cases, the bootloader is **GRUB** (Grand Unified Bootloader). There are also many other bootloaders like, Systemd-boot, EFIstub, LILO (Linux Loader), BURG (More graphically appealing bootloader based on GRUB), rEFInd etc.
3. **Kernel Initialization:** The loaded Linux kernel then initializes hardware drivers and mounts the root file system. After setting up essential kernel structure and loading necessary kernel modules, it starts the **init** process (or **systemd** in most modern Linux distributions).
4. **Initramfs and Device Initialization:** The **initramfs** (initial RAM filesystem) provides temporary root filesystem support before the real root filesystem is mounted.
5. **System Initialization:** The **init** process (or **systemd**) starts basic system processes and services, mounts the filesystem, and launches the background daemons. In some cases, **SysVinit** or **Upstart** does all the system initialization work.
6. **Target/Runlevel Execution:** After initialization, the system reaches an operating state, where it determines which processes to start or stop. This operating state is known as **Target** in systemd based distros and **Runlevel** in SysVinit based distros. This process can also refer to as Executing startup scripts

7. **User Login and Shell Access:** After completing all these processes, a login screen or prompt appears for user authentication. After authentication, the user becomes able to access the shell or desktop environment.



Starting up of a Linux Distro

Installation Methods: Linux systems require very low computing power. There are several methods to use Linux Distributions on
computers.

- **Bare Metal:** In this method, the storage device of the computer is formatted, and a fresh Linux installation is done. In this method, Linux is the only Operating System to be able to run on the computer.

- **Dual Boot:** In this method, a Linux Distribution is installed alongside another operating system. It means that the computer can run two different operating systems when required. Multiple operating systems can be installed using this method, but only one operating system can run at a time. Partitioning the secondary storage becomes a major challenge in this process.

- **Live Disk:** Many Linux distributions provide the option to install it on a removable storage device like CD or USB, instead of making any changes to the secondary device. That Linux OS runs in total form that removable device, which is known as Live Disk.
- **Virtual Installation:** Linux can be installed as virtual machines by using some softwares like VMWare Workstation or Oracle VirtualBox. In this method, Linux is installed on top of another Linux-based or non-Linux-based operating system.

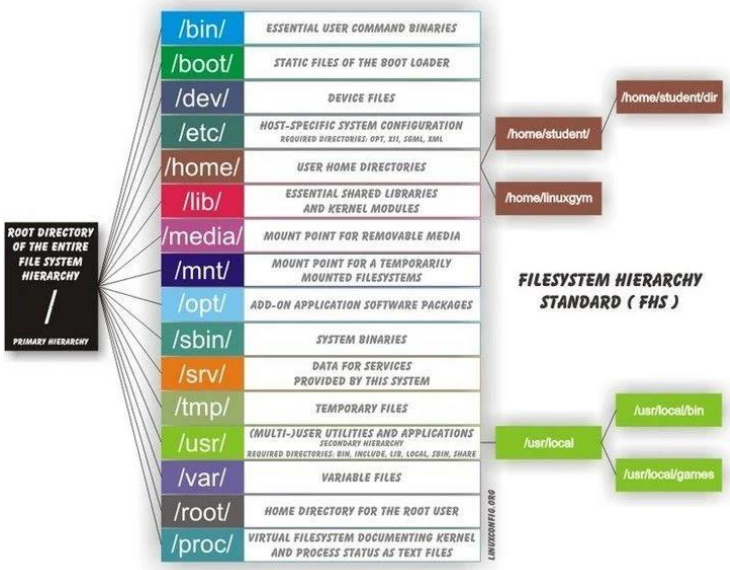
Graphical User Interface (GUI) in Linux: Linux is mostly known for its Command Line Interface (CLI) as it acts as a very lite option to control servers. But for daily usage, a Graphical User Interface is very important. Window Managers and Desktop Environments provide this to Linux.

- **Window Managers:** A Window Manager is a system software that controls the placement and appearance of windows in a GUI. These programs handle the way interaction with individual windows. e.g – i3wm, awesomewm, dwm, hyprland.
- **Desktop Environments:** A Desktop Environment (DE) provides a more complete GUI experience, bundling Window Managers with other essential components to create a unified and user-friendly interface. There are various DEs like – GNOME, KDE Plasma, Deepin, XFCE, LXQT, MATE, Budgie, Enlightenment and many more. Most Linux distributions come with a default Desktop Environment, but users can install their preferred Desktop Environment according to their needs.

File System Hierarchy of Linux: In Linux, File System is a hierarchal Structure that organizes and manages the files and directories on storage devices. It starts with a tree-like structure starting from the root directory (**/**).
All the Directories that come under the root directory (/) signifies a unique purpose.

- **/bin :** All the executable binary programs (file) required during booting, repairing, files required to run into single-user-mode, and other important, basic commands viz., cat, du, df, tar, rpm, wc, history, etc.
- **/boot :** Holds important files during boot-up process, including Linux Kernel.
- **/dev :** Contains device files for all the hardware devices on the machine e.g., CDROM, CPU, etc.
- **/etc :** Contains Application's configuration files, startup, shutdown, start, stop script for every individual program.
- **/home :** Home directory of the users. Every time a new user is created, a directory in the name of user is created within home directory which contains other directories like Desktop, Downloads, Documents, etc.
- **/lib :** The Lib directory contains kernel modules and shared library images required to boot the system and run commands in root file system.

- **/lost+found :** This Directory is installed during installation of Linux, useful for recovering files which may be broken due to unexpected shut-down.
- **/media :** Temporary mount directory is created for removable devices viz., media/CDROM.
- **/mnt :** Temporary mount directory for mounting file system.
- **/opt :** Optional is abbreviated as opt. Contains third party application software. Viz., Java, etc.
- **/proc :** A virtual and pseudo filesystem which contains information about running processes with a particular Process-ID aka PID.
- **/root :** This is the home directory of root user and should never be confused with '/'.
- **/run :** This directory is the only clean solution for early-runtime-dir problem.
- **/sbin :** Contains binary executable programs, required by System Administrator, for Maintenance. Viz., iptables, fdisk, ifconfig, swapon, reboot, etc.
- **/srv :** Service is abbreviated as 'srv'. This directory contains server-specific and service-related files.
- **/sys :** Modern Linux distributions include a /sys directory as a virtual filesystem, which stores and allows modification of the devices connected to the system.
- **/tmp :** System's Temporary Directory, Accessible by users and root. Stores temporary files for user and system, till next boot.
- **/usr :** Contains executable binaries, documentation, source code, libraries for second level program.
- **/var :** Stands for variables. The content of this file is expected to grow. This directory contains log, lock, spool, mail, and temp files.



Filesystem Hierarchy Standard in Linux

Folder and Directory: The terms Folder and Directory are used interchangeably to describe a container that contains files and folders. People call these containers directories in CLI and folders in GUI.

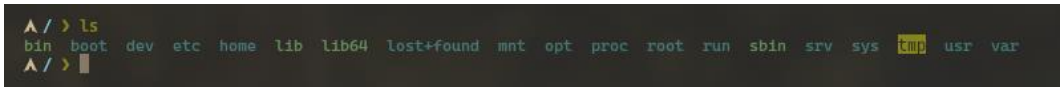
But to be precise, Directory refers to the way a structured list of document files and folders is stored on the computer. Whereas a Folder is a space where files and other folders can be stored.

Basic Linux Commands: Linux is famous for its Command Line Utilities. Any kind of computing task can be performed only using the right commands. Here are some of the most basic yet useful commands listed below. Before starting, it is essential to remember that Linux Shell Commands are Case Sensitive, i.e.- the commands differentiate between uppercase and lowercase letters.

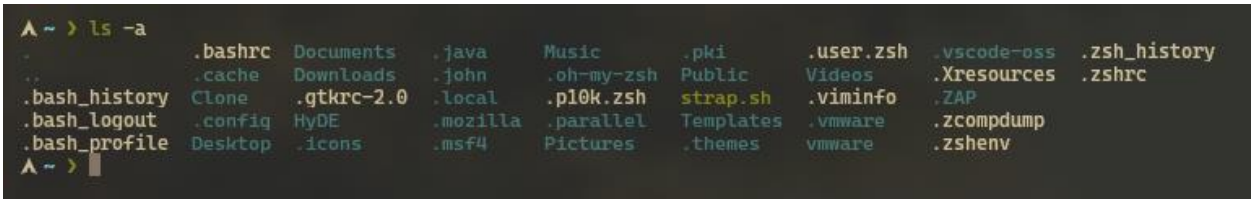
- **whoami:** Shows the current username.

- **pwd:** It is used to know the current directory which the user is in.

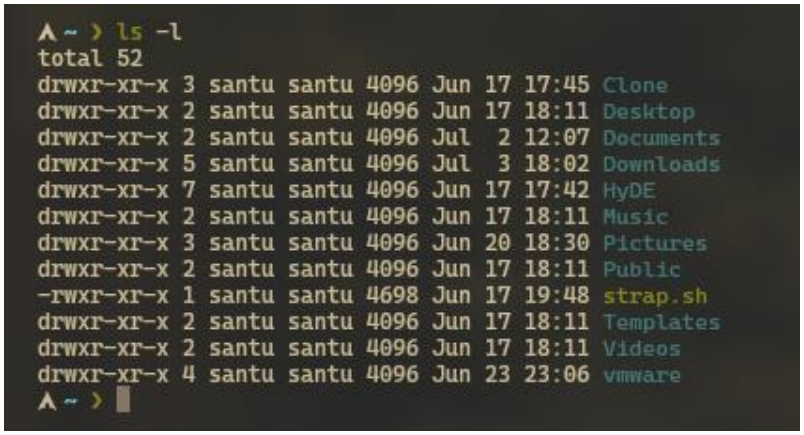
- **ls:** This command is used to list the contents of a directory.



ls –a is used to list all the files in a directory including the hidden files.



ls –l is used to view a detailed list of all contents in a directory.



ls [address of a directory] can be used to see the contents of another directory from the current directory. e.g.- ***ls /home/Downloads*** will list the contents of the **Downloads** directory.

Similarly, there are many more aliases which can be used to perform different content listing operations.

Options	Description
-l	Long listing format of files and directories, one per line
-a	List all hidden files and directories started with '.'
-F	Add a '/' classification at the end of each Directory
-g	List all files and directories with the group name
-i	Print index number of each files and directories
-m	List all file and directories separated by comma ','
-n	List numeric UID and GID of Owner and Groups
-r	List all files and directories in reverse order
-R	Short list all directories
-t	Sorted by modified time, started with the newest file

Alternatively, ***ls –help*** can be used to see other flags.

- **cd:** The cd command is used to change directory.
  Usage:
  cd [address of target directory]
  Example: ***cd /home/Downloads*** this command will take the shell to Downloads directory.
  If we simply type ***cd*** then the shell will change its directory to the home directory.
- **mkdir** and **rmdir:** **mkdir** is used to create a directory, and **rmdir** is used to permanently delete an empty directory.
  Usage:
  mkdir [New Directory Name]
  rmdir [Directory Name]

- **touch:** This command is used to create an empty text file.

Usage:

touch [fileName] or ***touch [address/filename]***

```
~/Documents ❯ touch newFile.txt
~/Documents ❯ ls
newFile.txt
```

- **cat:** The cat command is used to concatenate the contents of a file.
  Usage:
  cat [fileName] or ***cat [address/fileName]***

```
~/Downloads ❯ cat textFile.txt
Hey!!
It's a text file
```

- echo: This command is used to print output on the shell.
  Usage:
  echo "Hello World"

```
~ ❯ echo "Hello World"
Hello World
```

- **man:** This command opens the manual of scripts/commands.
  Usage:
  man [commandName]
- **uname:** uname command is used to get complete OS information. It can be used with different flags to get different information.

```
Usage: uname [OPTION]...
Print certain system information.  With no OPTION, same as -s.

  -a, --all                print all information, in the following order,
                             except omit -p and -i if unknown:
  -s, --kernel-name        print the kernel name
  -n, --nodename           print the network node hostname
  -r, --kernel-release     print the kernel release
  -v, --kernel-version     print the kernel version
  -m, --machine            print the machine hardware name
  -p, --processor          print the processor type (non-portable)
  -i, --hardware-platform  print the hardware platform (non-portable)
  -o, --operating-system   print the operating system
      --help        display this help and exit
      --version     output version information and exit
```

- **cp:** cp is the short form of "copy," the command is used to duplicate files or entire directories from one location of the file system to another.
  Usage:

cp [source] [target] or ***cp [addressOfSource/sourceFile] [adderssOfTarget]***

```
~/Documents ❯ ls
newFile.txt
~/Documents ❯ cat newFile.txt
Hey!!
It's a text File
~/Documents ❯ cp newFile.txt copiedTextFile.txt
~/Documents ❯ ls
copiedTextFile.txt  newFile.txt
~/Documents ❯ cat copiedTextFile.txt
Hey!!
It's a text File
```

```
~ ❯ ls Documents/
file
~ ❯ ls Documents/file
file1  file2  file3
~ ❯ cp -r Documents/file Documents/copiedFile
~ ❯ ls Documents/
copiedFile  file
~ ❯ ls Documents/copiedFile/
file1  file2  file3
```

- **mv:** mv is short form of "Move". In Linux, mv is used to move files or folders (with **–r** flag). **mv** is also used to rename files using CLI.
  Usage:
  mv [targetFile] [targetAddr] or **mv [addrOfTarget/targetFile] [targetAddr]**

```
~ ❯ ls Desktop/
~ ❯ mv Documents/copiedFile/ Desktop/
~ ❯ ls Desktop/
copiedFile
~ ❯ ls Documents/
file
```

  Mv [oldName] [newName] {to rename files of directories}

- **rm:** rm is used to remove files or entire directories (using **–r** flag). Non-empty directories can also be removed using **–rf** flag.
  Usage:
  rm [address/file]
  rm –rf [address/dirName] {to remove non-empty directories}
- **grep:** The grep command is used to find a specific string in a series of outputs. Commonly it is used with pipelining to get the specified strings or data.

Redirection in Linux: In Linux, redirection operation is performed using > operation.

Using **>** operators the output of a variable can be saved to another file. It creates a new file with the input filename but overwrites if the file is already present.

Usage:

echo "Hello World" > hello.txt

cat > fileName

Appending in Linux: Appending is done using **>>** operator. New texts can be added to an existing file using **>>** operator. It creates a new file and does not overwrite a file that is already present; instead, it adds new lines.

Usage:

cat >> filename

Pipelining in Linux: In Linux, Pipelining simply means a sequence of commands connected by the pipe operator (**|**). It allows the output of one command to be used as the input for the next, creating a flow of data from one command to another. Pipelining can be used to process and manipulate data from a large amount.

Usage:

ls | grep a {this will only show file names containing the letter a in it}