**Linux File System:-**

**Root (/)**

* /bin: - contains user binaries, executable files that are used by all users.
* /sbin: - contains system binaries, executable files that are used by root user (admin).
* /etc: - contains all system related configuration files. e.g. user password, Ip address, user id etc.
* /tmp: - contains all temporary files
* /dev: - contains device files.

**\*Note: - Everything in Linux is a File.**

* /var: - contains variable files. A notable directory /var/log where system log files are kept.
* /opt: - contains all third-party application/software.
* /lib: - contains all helpful library files used by the system.
* /media: - contains subdirectories where removable media devices inserted into the computer are mounted. (For example, when we insert a CD into your LINUX system, a directory will automatically created inside the /media directory)
* /home:- contains home directories for users.

**WINDOWS VS LINUX FILE SYSTEM:-**

* In Linux there is a **single hierarchical directory structure**. In Windows, there typically **many partitions with directories** under these partitions.
* In Linux, everything **starts from the root directory,** represented by ‘/’, and then expands into sub directories, in windows it had **various partitions and then directories under those partitions.**
* Linux file systems are **Ext, ReiserFS, Btrfs and XFS** where Windows file systems are **FAT (File Allocation Table) & NTFS (New Technology File System)**.
* On windows, an application might store all its files in **C:\Program Files\Applications**. On Linux, its files would be split between **multiple locations** – its binaries in /usr/bin, its libraries in /usr/lib, and its configuration files in /etc/.
* In windows, you can’t have a file named file and another file named FILE in the same folder. The **windows file system is not case sensitive**, so it treats these names as the same file. On **Linux the file system is case sensitive**. That means you could have files named file, File and FILE in the same folder.
* Windows uses backslashes, just as DOS did. For example, the path to a user’s directory on windows is **C:\Users\Name.** On Linux, the path to a user’s home directory is **/home/name**.
* On Linux applications don’t lock exclusive access to files, as often as they do on Windows. For example, if you are watching a video file in VLC on Windows. After you done watching it, you try to delete it. You will see an error message: you need to stop watching the file in VLC before you can delete it, rename it, or do anything else to it. On Linux, you could generally delete it or modify the video files as it was playing. You won’t see any error.

# **Different File types in Linux:-**

The most used and obvious file types are regular files and directories. the different types of Linux file types and **ls** command identifiers:

* “–“ : regular file
* **d** : directory
* **c** : character device file
* **b** : block device file
* **s** : local socket file
* **l** : symbolic link

## **Regular file**

The regular file is a most common file type found on the Linux system. It governs all different files such us text files, images, binary files, shared libraries, etc. You can create a regular file with the **touch** command.

## **Directory file**

Directory is second most common file type found in Linux. Directory can be created with the **mkdir** command:

As directory can be identified by “d” symbol from the **ls** command output. To remove empty directory use the **rmdir** command.

## **Character device file**

Character and block device files allow users and programs to communicate with hardware peripheral devices.

## **Block device file**

Block devices are similar to character devices. They mostly govern hardware as hard drives, memory, etc.

## **Local domain sockets file**

Local domain sockets are used for communication between processes. Generally, they are used by services such as X windows, syslog and etc.

Sockets can be created by socket system call and removed by the **unlink** or **rm** commands.

## **Symbolic Links file**

With symbolic links an administrator can assign a file or directory multiple identities. Symbolic link can be though of as a pointer to an original file. There are two types of symbolic links:

* hard links
* soft links

The difference between hard and soft links is that soft links use file name as reference and hard links use direct reference to the original file. Furthermore, hard links cannot cross file systems and partitions. To create symbolic soft link we can use **ln -s** command.

To remove symbolic link we can use **unlink** or **rm** command.