**Linux File System**

[**https://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/foreward.html**](https://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/foreward.html)

### What are filesystems?

A filesystem is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk.   
  
A *filesystem*is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk.

/bin /usr/bin :: Contains user Binaries and commands

/sbin /usr/sbin :: system binaries and admin commands

/etc All config file

/etc/opt Host specifc conifg files

/etc/skel When new user created files from dir copied to user home dir

/dev character device & block device [hard disk/ DVD CD]

/tmp temporary files

/boot boot files

/home user Home dir

/lib shared lib :: Libraries are codes that can be used by the executable binaries

/mnt Mount Dir

/opt Optional software

/root root home Directory

/var variable data files

/var/log/wtmp which logs all logins and logouts into the system

var/log/messages

Contains global system messages, including the messages that are logged during system startup. There are several things that are logged in /var/log/messages including mail, cron, daemon, kern, auth, etc.

/var/log/auth.log

Contains system authorization information, including user logins and authentication machinsm that were used.

/var/log/lastlog

Displays the recent login information for all the users. This is not an ascii file. You should use lastlog command to view the content of this file.

/var/log/secure

Contains information related to authentication and authorization privileges. For example, sshd logs all the messages here, including unsuccessful login.

/proc

Its virtual file system which contains information about currently running process and kernel parameters. . The content of the proc directory is used by a number of tools to get runtime system information.

**/proc/version**  Kernel version, gcc version, and Linux distribution installed

**/proc/uptime**  Uptime information (in seconds).

**/proc/stat**  Record or various statistics kept from last reboot

**/proc/swap**  Information about swap space

**/proc/mounts**  List of all mounts in use by system

**/proc/loadavg**  System load average

**/proc/filesystem**s Current filesystems supported by the kernel

**/proc/cpuinfo**  CPU & no of Processor info

**/proc/meminfo**  MemTotal Total amount of usable RAM

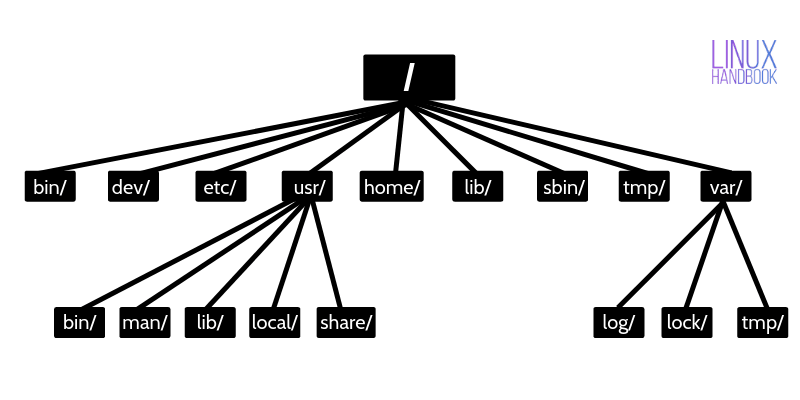
MemFree

Cached

### The root directory /

Everything, all the files and directories, in Linux are located under ‘root’ represented by ‘/’. If you look at the directory structure, you’ll realize that it is similar to a plant’s root.

**All files on a linux system are stored on filesystems which are organized into a single inverted tree of directories, known as file system hierarchy.**



**/bin – Binaries**

* The ‘/bin’ directly contains the executable files of many basic shell commands like ls, [cp](https://linuxhandbook.com/copy-directory-linux/), [cd](https://linuxhandbook.com/cd-command-examples/) etc. Mostly the programs are in binary format here and accessible by [all the users in the Linux system](https://linuxhandbook.com/linux-list-users/).
* For this reason and in contrast to /usr/bin, the binaries in this directory are considered to be essential. The reason for this is that it contains essential system programs that must be available even if only the partition containing / is mounted. This situation may arise should you need to repair other partitions but have no access to shared directories (ie. you are in single user mode and hence have no network access). It also contains programs which boot scripts may depend on.

This directory contains executable programs which are needed in **single user mode** and to bring the system up or repair it.

### /sbin – System binaries

* This is similar to the /bin directory. The only difference is that is contains the binaries that can only be run by root or a sudo user. You can think of the ‘s’ in ‘sbin’ as super or sudo.

Like /bin, this directory holds commands needed to boot the system, but which are usually not executed by normal users.

### /usr – User binaries and program data

In ‘/usr’ go all the executable files, libraries, source of most of the system programs. For this reason, most of the files contained therein is read­only (for the normal user).

This directory is usually mounted from a separate partition. It should hold only shareable, read-only data, so that it can be mounted by various machines running Linux

* **‘/usr/bin’** contains basic user commands

This is the primary directory for executable programs. Most programs exe‐cuted by normal users which are not needed for booting or for repairing

* **‘/usr/sbin’** contains additional commands for the administrator

This directory contains program binaries for system administration which are not essential for the boot process, for mounting /usr, or for system

* **/usr/local**
  + The original idea behind '/usr/local' was to have a separate ('local') '/usr' directory on every machine besides '/usr', which might be just mounted read-only from somewhere else. It copies the structure of '/usr'. These days, '/usr/local' is widely regarded as a good place in which to keep self-compiled or third-party programs.
  + The /usr/local hierarchy is for use by the system administrator when installing software locally. It needs to be safe from being overwritten when the system software is updated. It may be used for programs and data that are shareable amongst a group of hosts, but not found in /usr. Locally installed software must be placed within /usr/local rather than /usr unless it is being installed to replace or upgrade software in /usr.
* ‘/usr/lib’ contains the system libraries
* ‘/usr/share’ contains documentation or common to all libraries, for example ‘/usr/share/man’ contains the text of the manpage

### /etc – Configuration files

The /etc directory contains the core configuration files of the system, use primarily by the administrator and services, such as the password file and networking files.

A "configuration file" is defined as a local file used to control the operation of a program; it must be static and cannot be an executable binary. For this reason, it's a good idea to backup this directory regularly. It will definitely save you a lot of re-configuration later if you re-install or lose your current installation.

If you need to make changes in system configuration (for example changing the hostname), this is where you’ll find the respective files.

* /etc/opt

Host-specific configuration files for add-on applications installed in /opt.

* /etc/skel

When a new user account is created, files from this directory are usually copied into the user's home directory.

### /dev – Device files

There are two types of device files based upon how data written to them and read from them is processed by the operating system and hardware:

Device file allows transparent communication between user space applications and computer hardware.

* Character special files or Character devices
  + Talks to devices in a character by character (1 byte at a time)
  + Examples: Virtual terminals, terminals and serial modems etc
* **Character files** : These are also device files that provide unbuffered serial access to system hardware components. They work by providing a way of communication with devices by transferring data one character at a time.
* Usaully it will be in /dev
* $ ls -l /dev | grep ^c
* Block special files or Block devices
  + Talks to devices 1 block at a time ( 1 block = 512 bytes to 32KB)
  + Examples: Hard disk, DVD/CD ROM, and memory regions etc
* **Block files** : These are device files that provide buffered access to system hardware components. They provide a method of communication with device drivers through the file system. One important aspect about **block** files is that they can transfer a large block of data and information at a given time.
* Usaully it will be in /dev
* $ ls -l /dev | grep ^b

### /proc – Process and kernel files

The ‘/proc’ directory contains the information about currently running processes and kernel parameters. The content of the proc directory is used by a number of tools to get runtime system information.

For example, if you want to [check processor information in Linux](https://linuxhandbook.com/check-cpu-info-linux/), you can simply refer to the file /proc/cpuinfo. You want to [check memory usage of your Linux system](https://linuxhandbook.com/linux-memory-usage/), just look at the content of /proc/meminfo file.

The *proc* file system is a pseudo-file system which is used as an interface to kernel data structures. It is commonly mounted at */proc*. Most of it is read-only, but some files allow kernel variables to be changed.

**Ps uptime & uname .** These commands are that they provide a real time data. Which means each time you run the command, the output will be slightly different. This means its fetching information from a place which is very dynamic in nature and also is fetching from a source which is very much credible and provides a real and updated data each time.

Its stored in RAM (memory)

/proc file system is also accessed by the kernel using VFS. Due to this when a user tries to access a file inside the /proc file system, proc file system creates the content of that file with the help of information in the kernel. This is the reason when you list the directory /proc, most of it is shown with 0 bytes in size, but is populated dynamically when you access it.

Example : $file /proc/meminfo file wil be empty

The output says that the file is empty. But let's try to access the file with any editor like vi, cat or less. So when you access the content, the current values are populated from the kernel. This is the reason, why you get the most current and accurate status of the system from files inside /proc

* **/proc/cmdline** – Kernel command line information.
* **/proc/console** – Information about current consoles including tty.
* **/proc/devices** – Device drivers currently configured for the running kernel.
* **/proc/dma** – Info about current DMA channels.
* **/proc/fb** – Framebuffer devices.
* **/proc/filesystem**s – Current filesystems supported by the kernel.
* **/proc/iomem** – Current system memory map for devices.
* **/proc/ioports** – Registered port regions for input output communication with device.
* **/proc/loadavg** – System load average.
* **/proc/locks** – Files currently locked by kernel.
* **/proc/meminfo** – Info about system memory (see above example).
* **/proc/misc** – Miscellaneous drivers registered for miscellaneous major device.
* **/proc/modules** – Currently loaded kernel modules.
* **/proc/mounts** – List of all mounts in use by system.
* **/proc/partitions** – Detailed info about partitions available to the system.
* **/proc/pci** – Information about every PCI device.
* **/proc/stat** – Record or various statistics kept from last reboot.
* **/proc/swap** – Information about swap space.
* **/proc/uptime** – Uptime information (in seconds).
* **/proc/version** – Kernel version, gcc version, and Linux distribution installed.

### /boot – Boot files

This directory contains everything required for the boot process except for configuration files not needed at boot time (the most notable of those being those that belong to the GRUB boot-loader) and the map installer. Thus, the /boot directory stores data that is used before the kernel begins executing user-mode programs. This may include redundant (back-up) master boot records, sector/system map files, the kernel and other important boot files and data that is not directly edited by hand. Programs necessary to arrange for the boot loader to be able to boot a file are placed in /sbin. Configuration files for boot loaders are placed in /etc.

### /home – User personal data

Linux is a multi-user environment so each user is also assigned a specific directory that is accessible only to them and the system administrator. These are the user home directories, which can be found under '/home/$USER' (~/). It is your playground: everything is at your command, you can write files, delete them, install programs, etc.... Your home directory contains your personal configuration files, the so-called dot files (their name is preceded by a dot). Personal configuration files are usually 'hidden', if you want to see them, you either have to turn on the appropriate option in your file manager or run ls with the -a switch. If there is a conflict between personal and system wide configuration files, the settings in the personal file will preva

### /lib – Shared libraries

Libraries are basically codes that can be used by the executable binaries. The /lib directory holds the libraries needed by the binaries in /bin and /sbin directories.

Libraries needed by the binaries in the /usr/bin and /usr/sbin are located in the directory /usr/lib.

### /media – Mount point for removable media

When you connect a removable media such as USB disk, SD card or DVD, a directory is automatically created under the /media directory for them. You can access the content of the removable media from this directory.

### /mnt – Mount directory

This is similar to the /media directory but instead of automatically mounting the removable media, mnt is used by system administrators to manually mount a filesystem.

This is a generic mount point under which you mount your filesystems or devices. Mounting is the process by which you make a filesystem available to the system. After mounting your files will be accessible under the mount-point. This directory usually contains mount points or sub-directories where you mount your floppy and your CD. You can also create additional mount-points here if you wish. Standard mount points would include /mnt/cdrom and /mnt/floppy. There is no limitation to creating a mount-point anywhere on your system but by convention and for sheer practicality do not litter your file system with mount-points. It should be noted that some distributions like Debian allocate /floppy and /cdrom as mount points while Redhat and Mandrake puts them in /mnt/floppy and /mnt/cdrom respectively.

### /opt – Optional software

Traditionally, the /opt directory is used for installing/storing the files of third-party applications that are not available from the distribution’s repository.

The normal practice is to keep the software code in opt and then link the binary file in the /bin directory so that all the users can run it.

### /root – The home directory of the root

There is /root directory as well and it works as the home directory of the root user. So instead of /home/root, the home of root is located at /root. Do not confuse it with the root directory (/).

### /var – Variable data files

Var, short for variable, is where programs store runtime information like system logging, user tracking, caches, and other files that system programs create and manage.

The files stored here are NOT cleaned automatically and hence it provides a good place for system administrators to look for information about their system behavior. For example, if [you want to check the login history in your Linux system](https://linuxhandbook.com/linux-login-history/), just check the content of the file in /var/log/wtmp.

/var/backups

Directory containing backups of various key system files such as /etc/shadow, /etc/group, /etc/inetd.conf and dpkg.status. They are normally renamed to something like dpkg.status.0, group.bak, gshadow.bak, inetd.conf.bak, passwd.bak, shadow.bak

/var/cache

Is intended for cached data from applications. Such data is locally generated as a result of time-consuming I/O or calculation. This data can generally be regenerated or be restored. Unlike /var/spool, files here can be deleted without data loss. This data remains valid between invocations of the application and rebooting of the system. The existence of a separate directory for cached data allows system administrators to set different disk and backup policies from other directories in /var.

/var/crash

This directory will eventually hold system crash dumps. Currently, system crash dumps are not supported under Linux. However, development is already complete and is awaiting unification into the Linux kernel.

var/lib

Holds dynamic data libraries/files like the rpm/dpkg database and game scores. Furthermore, this hierarchy holds state information pertaining to an application or the system. State information is data that programs modify while they run, and that pertains to one specific host. Users shouldn't ever need to modify files in /var/lib to configure a package's operation. State information is generally used to preserve the condition of an application (or a group of inter-related applications) between invocations and between different instances of the same application. An application (or a group of inter-related applications) use a subdirectory of /var/lib for their data. There is one subdirectory, /var/lib/misc, which is intended for state files that don't need a subdirectory; the other subdirectories should only be present if the application in question is included in the distribution. /var/lib/'name' is the location that must be used for all distribution packaging support. Different distributions may use different names, of course.

/var/lock

Many programs follow a convention to create a lock file in /var/lock to indicate that they are using a particular device or file. This directory holds those lock files (for some devices) and hopefully other programs will notice the lock file and won't attempt to use the device or file.

Lock files should be stored within the /var/lock directory structure. Lock files for devices and other resources shared by multiple applications, such as the serial device lock files that were originally found in either /usr/spool/locks or /usr/spool/uucp, must now be stored in /var/lock. The naming convention which must be used is LCK.. followed by the base name of the device file. For example, to lock /dev/ttyS0 the file LCK..ttyS0 would be created. The format used for the contents of such lock files must be the HDB UUCP lock file format. The HDB format is to store the process identifier (PID) as a ten byte ASCII decimal number, with a trailing newline. For example, if process 1230 holds a lock file, it would contain the eleven characters: space, space, space, space, space, space, one, two, three, zero, and newline.

/var/log

Log files from the system and various programs/services, especially login (/var/log/wtmp, which logs all logins and logouts into the system) and syslog (/var/log/messages, where all kernel and system program message are usually stored). Files in /var/log can often grow indefinitely, and may require cleaning at regular intervals. Something that is now normally managed via log rotation utilities such as 'logrotate'. This utility also allows for the automatic rotation compression, removal and mailing of log files. Logrotate can be set to handle a log file daily, weekly, monthly or when the log file gets to a certain size. Normally, logrotate runs as a daily cron job. This is a good place to start troubleshooting general technical problems.

/var/log/auth.log

Record of all logins and logouts by normal users and system processes.

/var/log/btmp

Log of all attempted bad logins to the system. Accessed via the lastb command.

/var/log/debug

Debugging output from various packages.

/var/log/messages

System logs.

/var/log/utmp

Active user sessions. This is a data file and as such it can not be viewed normally. A human-readable form can be created via the dump-utmp command or through the w, who or users commands.

/var/log/wtmp

Log of all users who have logged into and out of the system. The last command can be used to access a human readable form of this file. It also lists every connection and run-level change.

/var/mail

Contains user mailbox files. The mail files take the form /var/mail/'username' (Note that /var/mail may be a symbolic link to another directory). User mailbox files in this location are stored in the standard UNIX mailbox format. The reason for the location of this directory was to bring the FHS inline with nearly every UNIX implementation (it was previously located in /var/spool/mail). This change is important for inter-operability since a single /var/mail is often shared between multiple hosts and multiple UNIX implementations (despite NFS locking issues).

/var/opt

Variable data of the program packages in /opt must be installed in /var/opt/'package-name', where 'package-name' is the name of the subtree in /opt where the static data from an add-on software package is stored, except where superceded by another file in /etc. No structure is imposed on the internal arrangement of /var/opt/'package-name'.

/var/run

Contains the process identification files (PIDs) of system services and other information about the system that is valid until the system is next booted. For example, /var/run/utmp contains information about users currently logged in.

/var/spool

Holds spool files, for instance for mail, news, and printing (lpd) and other queued work. Spool files store data to be processed after the job currently occupying a device is finished or the appropriate cron job is started. Each different spool has its own subdirectory below /var/spool, e.g., the cron tables are stored in /var/spool/cron/crontabs

/var/tmp

Temporary files that are large or that need to exist for a longer time than what is allowed for /tmp. (Although the system administrator might not allow very old files in /var/tmp either.

### /tmp – Temporary files

This directory contains mostly files that are required temporarily. Many programs use this to create lock files and for temporary storage of data. Do not remove files from this directory unless you know exactly what you are doing! Many of these files are important for currently running programs and deleting them may result in a system crash. Usually it won't contain more than a few KB anyway. On most systems, this directory is cleared out at boot or at shutdown by the local system. The basis for this was historical precedent and common practic