

# Filesystem Hierarchy

The Filesystem Hierarchy Standard (FHS) is a set of guidelines that define the directory structure of a Linux operating system. This standard specifies the names and locations of directories, files, and other file system objects that are essential for the proper functioning of the Linux operating system.

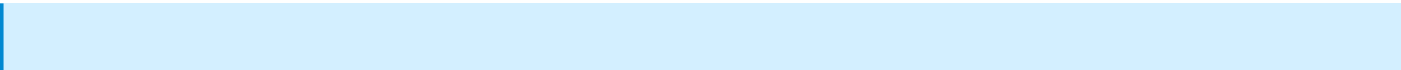
Directory	Description
/bin	binary files of basic commands
/boot	static files of the bootloader
/dev	device files
/etc	host-specific system configuration
/lib	basic dynamic libraries and kernel modules
/media	mount point for removable media
/mnt	for temporarily mounted file system
/opt	additional application programs
/run	relevant data for running processes
/sbin	essential binary files of the system
/srv	data for services
/tmp	temporary files
/usr	secondary hierarchy
/var	variable data

## Root Directory (/)

The root directory is the highest level of the file system hierarchy. It contains all other directories and files in the system. All directories and files in the system are located either directly or indirectly under the root directory.

/bin

## Directory



**NOTE:** On most modern Linux distributions, `/bin` is actually a symbolic link pointing to `/usr/bin`.

The `/bin` directory contains essential binary files that are required for the system to function properly. These files are used by all users of the system, and they include basic system commands such as `ls`, `cp`, and `mv`. The `/bin` directory must not contain any subdirectories. Each installed package places its main executables here.

## `/boot` Directory

The `/boot` directory contains the files needed by the bootloader for the Linux kernel, including the kernel itself, `initrd` (**initial RAM disk**), and boot loader configuration files. The bootloader is the program that loads the operating system when the computer starts up. The files in `/boot` are essential for the bootloader to function properly.

## `/dev` Directory

The `/dev` directory contains device files that represent hardware devices connected to the system. These files allow applications and users to interact with the hardware devices such as keyboards, mice, and disk drives using standard input/output operations.

There's different types of device files:

Type	Description
Character device	Communication with a character device is based on the exchange of characters, such as individual bytes
Block device	Communication with a block device is based on the exchange of entire blocks of data
Pseudo device	Devices that are not corresponding to any real hardware actually present on the system

Each device file in the `/dev` directory represents a specific hardware device. These files are created dynamically and disappear when the device is disconnected.

Some examples of device files:

File	Description
------	-------------

<code>/dev/dm-X</code>	<p>Logical volumes created using LVM (<b>d</b>evice <b>m</b>anager).</p> <ul style="list-style-type: none"> <li>• X = index of device manager volume</li> </ul>
<code>/dev/hdX</code>	<p>IDE hard drives or other storage devices. Common in older systems.</p> <ul style="list-style-type: none"> <li>• X = lower-case letter indicating order of discovery</li> </ul>
<code>/dev/hdXY</code>	<p>Partitions on IDE storage device.</p> <ul style="list-style-type: none"> <li>• X = disk</li> <li>• Y = partition on that disk</li> </ul>
<code>/dev/jsX</code>	<p>Device file for a joystick/gamepad.</p> <ul style="list-style-type: none"> <li>• X = index of a joystick connected to the system</li> </ul>
<code>/dev/mdX</code>	<p>Software RAID devices (<b>m</b>eta <b>d</b>isk).</p> <ul style="list-style-type: none"> <li>• X = meta disk group index</li> </ul>
<code>/dev/mmcblkX</code>	<p>MMC/SD cards and eMMC storage devices.</p> <ul style="list-style-type: none"> <li>• X = index of an MMC storage device</li> </ul>
<code>/dev/nvmeXnY</code>	<p>NVMe solid-state drives.</p> <ul style="list-style-type: none"> <li>• X = controller index</li> <li>• Y = device index on that controller</li> </ul>
<code>/dev/nvmeXnYpZ</code>	<p>Partitions on NVMe storage device.</p> <ul style="list-style-type: none"> <li>• X = controller index</li> <li>• Y = device index on that controller</li> <li>• Z = partition on that device</li> </ul>
<code>/dev/null</code>	<p>Pseudo-device that always outputs EOF (end of file). Output redirected to it is immediately discarded.</p>
<code>/dev/random</code>	<p>Pseudo-device that generates random data (blocking).</p>
<code>/dev/sdX</code>	<p>SCSI or SATA hard drives and other storage devices. Common in modern systems.</p> <ul style="list-style-type: none"> <li>• X = lower-case letter indicating order of discovery.</li> </ul>
<code>/dev/sdXY</code>	<p>Partitions on SCSI or SATA storage device.</p> <ul style="list-style-type: none"> <li>• X = disk</li> <li>• Y = partition on that disk</li> </ul>
<code>/dev/srX</code>	<p>CD-ROM or DVD-ROM drives.</p> <ul style="list-style-type: none"> <li>• X = index of an optical drive connected to the system</li> </ul>

<code>/dev/ttyX</code>	Device file for the terminal. <ul style="list-style-type: none"><li>• X = index of a terminal</li></ul>
<code>/dev/urandom</code>	Pseudo-device that generates random data (non-blocking)
<code>/dev/vdX</code>	Virtual storage device. <ul style="list-style-type: none"><li>• X = lower-case letter indicating order of discovery</li></ul>
<code>/dev/vdXY</code>	Partitions on a virtual storage device. <ul style="list-style-type: none"><li>• X = disk</li><li>• Y = partition on that disk</li></ul>
<code>/dev/videoX</code>	Device file for a webcam. <ul style="list-style-type: none"><li>• X = index of a webcam connected to the system</li></ul>

## `/etc` Directory

The `/etc` (**e**ditable **t**ext **c**onfiguration) directory contains system configuration files. These files are used to configure the system and its applications. Examples of files in this directory include `passwd` and `group` files, which contain user and group information, and `fstab` file, which contains information about file systems that are mounted at boot time. The `/etc` directory is one of the most important directories in the system, as it contains configuration files that enable system administrators to customize the system to their needs.

Some examples of what might be of particular interest:

Directory/File	Description
<code>/etc/opt/</code>	Configuration files for software installed to <code>/opt</code> .
<code>/etc/security/</code>	Configuration files for PAM ( <b>P</b> luggable <b>A</b> uthentication <b>M</b> odules).
<code>/etc/skel/</code>	Configuration files that are copied to a newly created user's <code>/home</code> directory.
<code>/etc/X11/</code>	Configuration files for the X Window System which provides a graphical user interface for Linux.
<code>/etc/crontab</code>	Configuration file uniquely formatted to automate system tasks on a set schedule.
<code>/etc/fstab</code>	Contains information about file systems and partitions automatically mounted at system startup.

Directory/File	Description
<code>/etc/group</code>	Contains information about user groups, including group names and group IDs.
<code>/etc/hosts</code>	Configuration file that maps hostnames to IP addresses.
<code>/etc/logrotate.conf</code>	Configuration file for the <code>logrotate</code> utility, managing system log files.
<code>/etc/passwd</code>	Stores user account information such as usernames, user IDs, home directories, and login shells.
<code>/etc/profile</code>	Contains system-wide environment variables and other startup scripts.
<code>/etc/resolv.conf</code>	DNS resolver file which specifies how the system leverages DNS to resolve hostnames.
<code>/etc/shadow</code>	Contains encrypted user passwords and other password-related information.
<code>/etc/ssh/sshd_config</code>	Configuration file for the SSH server, which allows secure remote access to the system.
<code>/etc/sudoers</code>	Contains rules for allowing or denying <code>sudo</code> privileges to users and groups.

# /home Directory

The `/home` directory contains the personal files for all users in the system. Each user has a subdirectory of their name in the `/home` directory that contains their documents, pictures, music, and individual application settings. Each user's home directory is private, and other users cannot access the files in that directory without the user's permission.

**NOTE:** A user's personal home directory can be shortened to `~`.

**NOTE:** Directories and files starting with a period `.` are considered hidden.

Some examples for directories/files in a user's home directory could include:

Directory/File	Description
<code>~/.config/</code>	Contains configuration files for applications that are specific to a user.
<code>~/.gnupg/</code>	Contains user-specific configuration files for GnuPG (GNU Privacy Guard).

Directory/File	Description
<code>~/.local/</code>	Contains user-specific data files for applications that are installed system-wide.
<code>~/.mozilla/</code>	Contains user-specific configurations and settings for Firefox.
<code>~/.ssh/</code>	Contains user-specific configuration files and private/public keys for SSH (Secure Shell) connections.
<code>~/.bashrc</code> , <code>~/.zshrc</code>	Contains user-specific configurations for the user's terminal emulator.

These hidden directories in a user's home directory are important for ensuring that user-specific data and configuration files are stored in a separate location from system-wide files. This separation enables users to customize their system to their needs without affecting other users or the core system.

# /lib Directory

**NOTE:** On most modern Linux distributions, `/lib` is actually a symbolic link pointing to `/usr/lib`.

The `/lib` directory (or `/lib64` for 64-bit systems) is a crucial component of the operating system. It contains shared libraries that are required by programs in `/bin` and `/sbin` directories, as well as other applications. These libraries provide functionality for many basic system operations, such as networking, file handling, and encryption. Without these libraries, the system would not be able to perform many of its fundamental functions. It is important to ensure that the files in `/lib` are always up-to-date and compatible with the system's other components to guarantee smooth operation.

In addition to the shared libraries, the `/lib` directory contains other important files, such as kernel modules (`/lib/modules/`) and firmware files (`/lib/firmware/`). Kernel modules are small programs that can be loaded and unloaded dynamically into the kernel without having to reboot the system. These modules provide additional functionality to the system, such as support for new hardware or file systems. Firmware files, on the other hand, contain low-level software that is used to control hardware devices, such as network adapters (especially WiFi) or graphics cards.

Directories/Files	Description
<code>/lib/firmware/</code>	Contains firmware files needed by devices to function properly, e.g. WiFi adapters and graphics cards.
<code>/lib/security/</code>	Contains authentication modules for PAM ( <b>P</b> luggable <b>A</b> uthentication <b>M</b> odules).

Directories/Files	Description
<code>/lib/systemd/</code>	Contains files for the systemd system and service manager. Also contains systemd unit files.

# `/media` Directory

The `/media` directory contains mount points for removable media devices such as USB drives and CD/DVDs. When a removable device is mounted, its contents appear in a subdirectory under the `/media` directory. The `/media` directory is used to manage removable media devices, and it allows users to access the files on those devices as if they were part of the file system.

# `/mnt` Directory

The `/mnt` directory is used to temporarily mount file systems. This directory is typically used by system administrators when they need to mount a file system for a short period of time. The `/mnt` directory is used to access file systems that are not part of the main file system, such as network file systems or file systems on removable media devices.

# `/opt` Directory

The `/opt` directory is used for optional software packages that are not part of the main operating system. Examples of software that might be installed in the `/opt` directory include proprietary software packages or third-party software that is not included with the distribution's main software repositories.

Examples of software you might see in this directory:

Directory/File	Description
<code>/opt/1Password/</code>	Password manager from Agile Bits
<code>/opt/discord/</code>	Discord VoIP client
<code>/opt/google/</code>	Google Chrome web browser
<code>/opt/quake3/</code>	Quare 3 Arena from id Software
<code>/opt/spotify/</code>	Spotify music streaming client
<code>/opt/visual-studio-code/</code>	Visual Studio Code editor from Microsoft

# /proc Directory

The `/proc` directory is a virtual directory that contains information about running processes and system resources. This directory provides a way for processes to communicate with the kernel and allows system administrators to monitor system performance. The files in the `/proc` directory are not actual files, but rather a virtual representation of system resources. The `/proc` directory is used by system administrators to monitor system performance and diagnose system problems.

Directory/File	Description
<code>/proc/cpuinfo</code>	Contains information about the CPU, e.g. model name, number of cores, and clock speed.
<code>/proc/meminfo</code>	Contains memory usage information including swap storage use.
<code>/proc/modules</code>	Contains a list of all the modules being used by the kernel.
<code>/proc/swaps</code>	Contains information only about swap storage.
<code>/proc/sys</code>	Contains files that are used to manage kernel parameters.
<code>/proc/version</code>	Contains Linux version information.

# /root Directory

The `/root` directory is the home directory for the root user. This directory contains the root user's personal files and settings. The root user is the most powerful user on the system, and has access to all files and directories on the system.

# /run Directory

The `/run` directory is a unique and important directory in the Linux operating system that contains runtime data that is used by the system and applications. This data is volatile as it is mounted as a `tmpfs` filesystem that is only stored in system memory and its contents are lost when the system is rebooted.

The data stored in the `/run` directory is updated frequently and automatically by the system and applications. The directory is also used as a mount point for systemd, a system and service manager that is used in most modern Linux distributions. Systemd is responsible for managing system services, and it also uses the `/run` directory to store runtime data for these services.



The `/run` directory is also used for user mounted storage mediums like CDs, DVDs, USB drives and network shares.

## `/sbin` Directory

**NOTE:** On most modern Linux distributions, `/sbin` is actually a symbolic link pointing to `/usr/bin`.

The `/sbin` directory contains essential system binaries that are used for system administration tasks, such as configuring the network or managing user accounts. These files are typically used by system administrators and are not intended for everyday use.

## `/srv` Directory

The `/srv` directory is used to store data that is served by the system, and it is typically used by web servers, FTP servers, and other server applications.

Other directories that are used for different purposes include:

Directory/File	Description
<code>/srv/ftp/</code>	Contains files that an FTP server uses to store all of the files that can be accessed by the FTP server.
<code>/srv/www/</code> , <code>/srv/http/</code>	Contains files that are served by a web server (sub-directory is distribution specific).

## `/sys` Directory

The `/sys` directory is a fundamental component of the Linux filesystem hierarchy and an essential tool for hardware management and configuration. This directory is structured in such a way that each hardware device is represented by a directory tree and a set of files, providing users with detailed information about the device's properties and allowing them to adjust, among other things, power management settings, device parameters, and monitor the device's status. The files in the `/sys` directory are not actual files, but rather a virtual representation of hardware devices. They are re-created upon every boot of the system and mounted as a virtual `sysfs` filesystem.

# `/tmp` Directory

The `/tmp` directory is used for temporary files. This directory is typically used by applications to store non-critical files and it is typically cleaned out on a regular basis to prevent the accumulation of unnecessary files. It is writable to any user and is ideal for using it as a scratchpad type of directory. The `/tmp` directory is mounted as a `tmpfs` type of filesystem, which means its contents are stored in the system's memory for fast access. However, this also means that it is generally not recommended putting large files there as to not exhaust the system's memory with large amounts of data.

# `/usr` Directory

The `/usr` (**U**nix **S**ystem **R**esources) directory is used to store programs, libraries, documentation and data that are not part of the core operating system, but that are still important for the system to function. Examples of programs that might be installed in the `/usr` directory include text editors, email clients, and games.

Some other examples of contents of the `/usr` directory:

Directory/File	Description
<code>/usr/bin/</code>	Contains executable binaries of applications
<code>/usr/bin/gcc/</code>	Binaries for compiling all sorts of source code of different programming languages
<code>/usr/include/</code>	Source code header files for including functionality of one application into another during development
<code>/usr/lib/</code>	Shared system libraries, e.g. <code>libc.so</code> (C library), <code>libssl.so</code> (SSL library)
<code>/usr/lib/systemd/</code>	Systemd service unit files
<code>/usr/local/</code>	Typically used to install applications compiled from source, otherwise goes largely unused
<code>/usr/share/</code>	Architecture-independent data files, e.g. game assets, icons, cursors, desktop themes, wallpapers, etc.
<code>/usr/share/doc/</code>	System documentation
<code>/usr/share/locale/</code>	Translations for applications
<code>/usr/share/man/</code>	Application manual pages
<code>/usr/src/</code>	Linux kernel source code files

# `/var` Directory

The `/var` directory contains variable data files. These files are typically files that change frequently, such as log files, locally delivered mail and spooler files for printer job queues.

Some other examples of contents of the `/var` directory:

Directory/File	Description
<code>/var/cache/</code>	Frequently accessed data, e.g. browser cache and downloaded software packages
<code>/var/lib/</code>	Varying application data used by system libraries (e.g. <code>/var/lib/libvirt/images/</code> for <code>libvirt</code> virtual machine disk images)
<code>/var/log/</code>	Log files of all sorts of different applications
<code>/var/mail/</code>	Locally delivered mail messages
<code>/var/mysql/</code>	MySQL database files
<code>/var/spool/</code>	Printer job queue

Revision #6

Created 28 March 2023 21:55:51 by Sebin

Updated 30 March 2023 23:27:40 by Sebin