

Linux Filesystems

1. Think of a refrigerator that has multiple shelves that can be used for storing various items.
2. These shelves help you organize the grocery items by shape, size, type, etc
3. The same concept applies to a filesystem, which is the embodiment of a method of storing and organizing arbitrary collections of data in a human-usable form.

Different types of filesystems supported by Linux:

- Conventional disk filesystems: **ext3, ext4, XFS, Btrfs, JFS, NTFS, vfat, exfat**, etc.
- Flash storage filesystems: **ubifs, jffs2, yaffs**, etc.
- Database filesystems
- Special purpose filesystems: **procfs, sysfs, tmpfs, squashfs, debugfs, fuse**, etc.

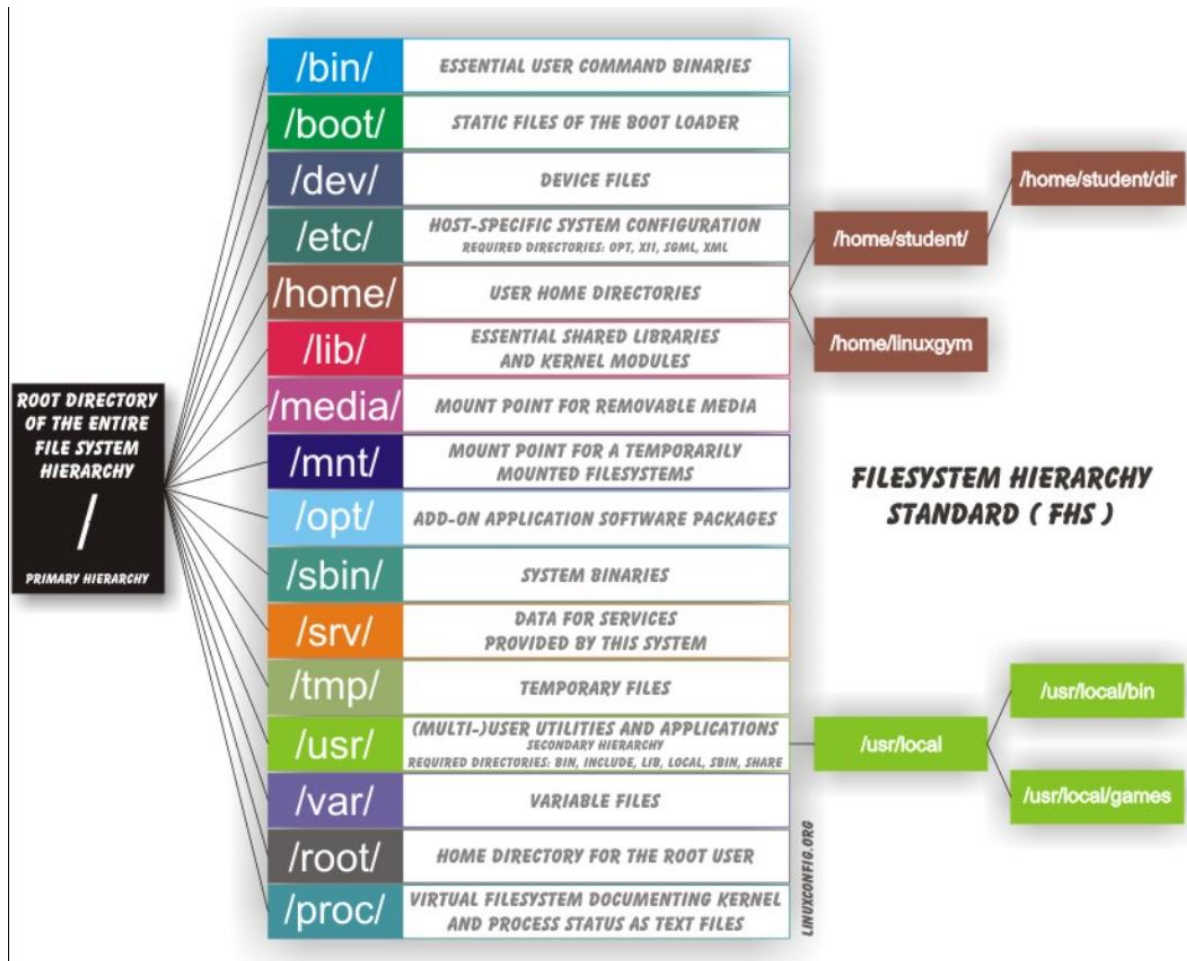
Partitions and Filesystems:

1. A **partition** is a physically contiguous section of a disk, or what appears to be so in some advanced setups.
2. A **filesystem** is a method of storing/finding files on a hard disk (usually in a partition).
3. One can think of a partition as a container in which a filesystem resides.
4. A comparison between filesystems in Windows and Linux is given in the accompanying table:

	Windows	Linux
Partition	Disk1	/dev/sda1
Filesystem Type	NTFS/VFAT	EXT3/EXT4/XFS/BTRFS...
Mounting Parameters	DriveLetter	MountPoint
Base Folder (where OS is stored)	C:\	/

Filesystem Hierarchy Standard:

1. Linux systems store their important files according to a standard layout called the Filesystem Hierarchy Standard (FHS).
2. Linux uses the '/' character to separate paths (unlike Windows, which uses '\'), and does not have drive letters.
3. Multiple drives and/or partitions are mounted as directories in the single filesystem.
4. Removable media such as USB drives and CDs and DVDs will show up as mounted at **/run/media/yourusername/disklabel** for recent Linux systems, or **under /media** for older distributions.
5. For example, if your username is **student** a USB pen drive labeled FEDORA might end up being found at **/run/media/student/FEDORA**, and a file **README.txt** on that disc would be at **/run/media/student/FEDORA/README.txt**.



More About the Filesystem Hierarchy Standard:

1. All Linux filesystem names are case-sensitive, so **/boot**, **/Boot**, and **/BOOT** represent three different directories (or folders).
2. Many distributions distinguish between core utilities needed for proper system operation and other programs, and place the latter in directories under **/usr** (think user).
3. To get a sense for how the other programs are organized, find the **/usr** directory in the diagram from the previous page and compare the subdirectories with those that exist directly under the system root directory (**/**).

The image shows four terminal windows in a virtual machine environment, each displaying the output of a 'tree' command to show directory structures. The background of the terminal windows is a desert landscape.

- Terminal 1 (leftmost):** Shows the directory tree for `/usr`. It lists `bin`, `boot`, `dev`, `etc`, `home`, `lib`, `lib64`, `lost+found`, `media`, `mnt`, `opt`, `proc`, `root`, `run`, `sbin`, `srv`, `sys`, `tmp`, `usr`, and `var`. It reports 20 directories.
- Terminal 2:** Shows the directory tree for `/home`. It lists `student` and its subdirectories: `.cache`, `.ccache`, `.config`, `.dbus`, `Desktop`, `Documents`, `Downloads`, `.emacs.d`, `LFT`, `.local`, `.mozilla`, `Music`, `Pictures`, `Public`, `Templates`, and `Videos`. It reports 17 directories.
- Terminal 3:** Shows the directory tree for `/usr`. It lists `bin`, `etc`, `games`, `include`, `lib`, `lib64`, `libexec`, `local`, `sbin`, `share`, `src`, and `tmp`. It reports 12 directories.
- Terminal 4 (rightmost):** Shows the directory tree for `/var`. It lists `account`, `adm`, `cache`, `crash`, `cvs`, `db`, `empty`, `ftp`, `games`, `gopher`, `kerberos`, `lib`, `local`, `lock`, `log`, `mail`, `nis`, `opt`, `preserve`, `run`, `spool`, `target`, `tmp`, and `yp`. It reports 24 directories.

Summary:

1. A **partition** is a logical part of the disk.
2. A **filesystem** is a method of storing/finding files on a hard disk.
3. By dividing the hard disk into partitions, data can be grouped and separated as needed. When a failure or mistake occurs, only the data in the affected partition will be damaged, while the data on the other partitions will likely survive.
4. The boot process has multiple steps, starting with BIOS, which triggers the boot loader to start up the Linux kernel. From there, the `initramfs` filesystem is invoked, which triggers the `init` program to complete the startup process.
5. Determining the appropriate distribution to deploy requires that you match your specific system needs to the capabilities of the different distributions.