The File System Hierarchy Standard (FHS)

File System Hierarchy Standard (FHS) provides a set of requirements and guidelines for file and directory placement in Linux and any Unix-like operating system. In other words, defines the main directories and their contents. An FHS-compliant file system makes it easy for users to locate and specific files and directories because everything is in its expected place in the file system. However, the FHS document is the authoritative reference to any FHS-compliant file system, but the standard leaves many areas undefined or extensible.

The idea behind the FHS is to organize the directories so that shareable files could be placed in one place to make easily available for programs while separating from private or exclusive files, or in another aspect we can segregate static and variable files and putting all the static files into a read-only medium while frequently variable files (like logs) could be placed on a fast storage drive.

The separation resulted four important parts in the FHS directory structure - the root `/`, `/usr`, `/var` and `/home`. Each piece has its own functionality and could be placed on different file systems to achieve greater productivity - like using a network based file system for the home directories to made it available all the computers in the network.

The pieces are serving the following common role:

- `root (/)` file system must exists and should be unique for each machine. Commonly it is located on a local disk but it can be placed on ram drives - as we seen on the booting procedure - or could be exists on a network share. It must contain all the necessary files that are required for the booting and mounting. The root file system is only enough to the single user mode which can be used to repair or restore the operating system. Sometimes the BIOS in older PCs can access only the first 1024 cylinders of your hard disk.

  Note

  To make sure that the information in your `/boot` directory is accessible to the BIOS, create a separate disk partition (of only about 100MB) for /boot and make sure that it exists below cylinder 1024.

  Note

  Protecting `/tmp` from the rest of the hard disk by placing it on a separate partition can ensure that applications that need to write to temporary files in /tmp are able to complete their processing, even if the rest of the disk fills up.
- `/usr` file system contains most of the applications and utilities available to users. The content of this part is not machine specific, so having /usr on a separate partition lets you mount that file system as read-only after the operating system has been installed. This prevents attackers from replacing or removing important system applications with their own versions that may cause security problems. A separate /usr partition is also useful if you have diskless workstations on your local network. Using NFS, you can share /usr over the network with those workstations. In this case you need to update your applications only one place not with a one-by-one procedure.
- `/var` file system holds all the variable files like spools (for mail, news, printing), logs, manual pages and temporary files.
- `/home` file system holds all the user data, so the real data in the system. Having a separate /home mount point can help to create backups easier because the other part of the system is not changing so frequently like this one. Also, some people have a separate /home partition so they can reinstall the operating system, erasing the root (/) partition, and simply remounting the /home partition. However, a big /home file system you may want to disassemble into a few more parts to made it easier to administer, like in our University the /home entry is divided into two more parts, one for the employees and one for the students.

While we call the different part as file systems it is not always necessary to put them on a separate file system really. Small, one user environments does not requires all the above separations because one root file system can serve all the purposes. The only one essentialist thing is all the standard names must work.

Now, let's see the most important directories and their meaning:

Table 3.1. The majority of the FHS

root directory	Primary hierarchy root and root directory of the entire file system hierarchy. It can be referenced by the / .
/bin	Essential command **binaries** that need to be available in single user mode and for ***all users***, e.g. cat, ls, cp. (especially files required to boot or rescue the system; mostly small and simple commands). Application programs doesn't have nothing to install here.
/boot	Contains static files required to boot the system, such as the Linux kernel. This directory holds also the boot loaders files.

/dev	The /dev/ directory contains device nodes that either represent devices that are attached to the system or virtual devices that are provided by the kernel. These device nodes are essential for the system to function properly. Devices in this directory and subdirectories are either character or block (discussed previously). Character devices include mouse, keyboard, modem while block devices include hard disk, floppy drive etc.
	Examples of common files in the /dev include:
	`/dev/hda - the master device on primary IDE channel.`
	`/dev/hdb - the slave device on primary IDE channel.`
	`/dev/tty0 - first virtual console.`
	`/dev/tty1 - second virtual console.`
	`/dev/sda - first device on primary SCSI or SATA channel.`
	`/dev/lp0 - first parallel port.`

	Host-specific system-wide configuration files. In early versions of the UNIX Implementation Document from Bell labs, /etc is referred to as the etcetera directory,as this directory historically held everything that did not belong elsewhere. However, the FHS restricts /etc to static configuration files and may not contain binaries. The directory name has been re-designated in various ways. Recent interpretations include backronyms such as "Editable Text Configuration" or "Extended Tool Chest". In contrast with Windows, where the Registry does the black magic with its complicated and undocumented structure, the Unix-like solution is much more comfortable and gives a clear-cut to find and use the configuration entries. Moreover, its more than expected that a misconfigurated system still remain configurable through some settings prevent from properly booting it up. Naturally, the user-specific settings are stored in a different location, usually in hidden subfolders in the user's home directory. Major files and directories:

/etc	• /etc/rc or /etc/rc.d or /etc/rc?.d
	Scripts executed on system initialization or run level change.
	• /etc/passwd
	User database with the following fields: username, user ID (UID), primary group ID (GID), description, like real name, phone number, room number - its name GECOS filed), home directory and login shell.
	`adamkoa:x:500:500:Adamko Attila:/home/adamkoa:/bin/bash`
	• /etc/fstab
	Mounted file systems at system startup (done by the mount -a command) .
	• /etc/group
	Like the /etc/passwd file but describes the available groups on the machine.
	`adamkoa:x:500:`
	`mail:x:12:mail,exim,postfix`
	• /etc/inittab
	Configuration file for the init process. (see Booting section)
	• /etc/issue
	A message to getty - displayed before the login prompt. Usually contains a short system information and a welcome message. The content is configured by the system administrator.
	• /etc/motd
	A daily message displayed after a successful login.
	• /etc/mtab
	The currently mounted file systems.
	• /etc/shadow
	The shadow password file which contains the same as the /etc/passwd but extended with the encoded version of the user's password. Only the root can read.
	• /etc/profile, /etc/csh.login, /etc/csh.cshrc
	Executed by the different shells at logon. This makes available to system administrators to create a unique settings for all the users. More details in the given shell's manual.
	Examples of directories in /etc are the X11/ and skel/:
	`/etc`
	`   |- X11/`
	`   |- skel/`
	The /etc/X11/ directory is for X Window System configuration files, such as xorg.conf. The /etc/skel/directory is for "skeleton" user files, which are used to populate a home directory when a user is first created. Applications also store their configuration files in this directory and may reference them when they are executed.

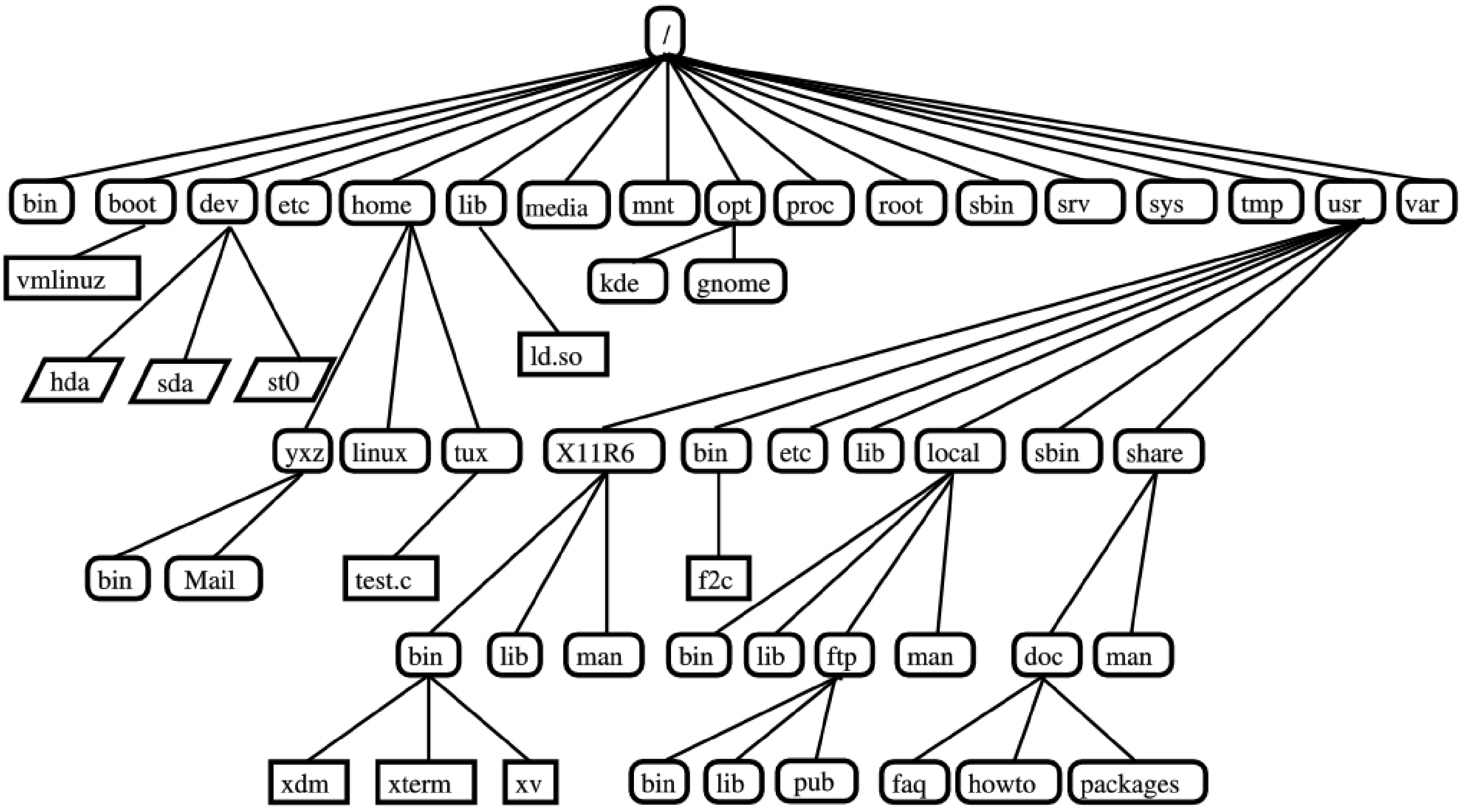
/home	Users' home directories, containing saved files, personal settings, etc. Under this directory the user (owner) has full permissions, outside this directory mostly read permission is granted only. You can reference to your own home directory with the ~ (tilde) letter, and any other users' home directory with the ~`username` syntax.
/lib	The /lib/ directory should contain only those libraries needed to execute the binaries in /bin/ and /sbin/ . These shared library images are particularly important for booting the system and executing commands within the root file system. You'll never find an executable at /bin or /sbin that needs a library that is outside this directory. One can think about the libraries as function libraries, containing the necessary functions for a given functionality. They are similar to the Windows' dynamically loadable libraries - a.k.a. DLLs. Under this directory can be found the kernel modules (device drivers) as well.
/lost+found	Some files and fragment that were "recovered" during the previous fsck [file system check] (Not part of FHS! Each partition has its own lost+found directory.)
/media	Mount points for removable media such as CD-ROMs, USB PenDrives, floppy, ... (appeared in FHS-2.3)
/mnt	Temporarily **m**ou**nt**ed file systems for the current session only. Automatically detected removeable media will be mounted in the /media directory, all the others can go here. Because there is no default location for the volumes like inside Windows drive letter scheme, one can mount anywhere in the directory structure but this point is reserved for it.
/opt	Add-on application software packages which are pre-compiled and non-distribution specific (tar'ed) goes here.
/proc	Virtual file system providing information about processes and kernel information as files. Examples include system memory, cpu information, hardware configuration etc. In Linux, corresponds to the procfs mount.
/root	Home directory for the root user.
/sbin	Stores executables used by the root user. The executables in /sbin/ are used at boot time, for system administration and to perform system recovery operations. Typical examples are init, halt, route, ifup, ifconfig, swapon, etc. Locally-installed system administration programs should be placed into /usr/local/sbin.
/sys	The /sys/ directory utilizes the new sysfs virtual file system specific to the 2.6 kernel. With the increased support for hot plug hardware devices in the 2.6 kernel, the /sys/ directory contains information similarly held in /proc/, but displays a hierarchical view of specific device information in regards to hot plug devices.
/tmp	Directory to hold **Temp**orary files. Mostly not preserved between system reboots. This is the other location which can be written by all the users.

	Secondary hierarchy for user data and contains the majority of (multi-)user utilities and applications. (formerly from **U**NIX**s**ource**r**epository, now from **U**NIX**s**ystem **r**esources) These files are **not**-required to boot or rescue the system and can be shared across multiple machines thus often it has its own partition and is mounted read-only. Its structure similar to the primary hierarchy root.
	At a minimum, the following directories should be subdirectories of /usr/:
	`/usr`
	`   |- bin/ - contains executables for user applications and utilities`
	`   |- etc/ - config files /often empty/`
	`   |- games/`

/usr	`    |- include/ - C header files`
	`    |- kerberos/`
	`    |- lib/ - object files and libraries that are not designed to be directly utilized by users or shell scripts`
	`    |- libexec/`
	`    |- local/ - entry point for the Tertiary hierarchy - specific things for the current machine`
	`    |- sbin/ - system administration binaries (those that do not belong in the  /sbin/ directory)`
	`    |- share/ - files that are not architecture-specific`
	`    |- src/ - for source code`
	`    |- tmp -> ../var/tmp/`

/var	Variable files—files whose content is expected to continually change during normal operation of the system—such as logs, spool files, and temporary e-mail files.
	`/var`
	`    |- log/ - System log files`
	`    |- mail ->spool/mail/`
	`    |- run/`
	`    +- spool/ - for programs in which data files are stored`
	`        |- at/`
	`        |- cron/`
	`        |- mail/`
	`    |- tmp/`

The FHS can be seen on the following figure:



Note

The process of developing a standard file system hierarchy began in August 1993 with an effort to restructure the file and directory structure of Linux-based operating systems. The FSSTND (File system Standard), a file system hierarchy standard specific to the Linux operating system, was released on 14 February 1994. Subsequent revisions were released on 9 October 1994 and 28 March 1995.
In early 1996, the goal of developing a more comprehensive version of FSSTND to address not only Linux, but other Unix-like systems was adopted with the help of members of the BSD development community. As a result, a concerted effort was made to focus on issues that were general to Unix-like systems. In recognition of this widening of scope, the name of the standard was changed to File System Hierarchy Standard.
The last version (v2.3) of FHS was released on 2004 but its update is coming soon because v3.0 is under development. The biggest change in FHS 3.0 is probably the addition of a new top-level directory, /run. The /run directory essentially moves /var/run (which used to hold run-time data that appears only after booting up, such as process ID (PID) files) up one level, and consolidates a few other dynamically-generated, non-persistent miscellany — including /var/lock lock-files, /dev/shm temporary storage, and udev state. The goal is for /run to hold all system-state data that is not meaningful if persisted from one reboot to another.

Prev Up Next
File systems under UNIX Home Chapter 4. Working with files