

LPIC-1 101-400 – Lesson 23

104.7 Find system files and place files in the correct location




Filesystem Hierarchy Standard – FHS

- The Filesystem Hierarchy Standard – (FHS) is a standard that describes the hierarchy and purpose of directories in a Linux system
- All the main Linux distributions support it (with minor variations)
- Tries to set some order in the alternative hierarchies chaos of the early years of Linux and give some homogeneity between distributions
- The latest release is 3.0 of 2015



Directory Classification

- **Shareable – Unshareable**
 - **Shareable** directories are directories that are typically common in a number of systems and so they can share a common network filesystem
 - **Unshareable** are directories with different information in every system and so they must be local to the system and cannot be shared
 - **Variable – Static**
 - **Variable** are directories with variable data
 - **Static** are the directories that do not normally change. When they do change is usually after an admin task. Some examples are executable files, libraries and documentation files.
- 

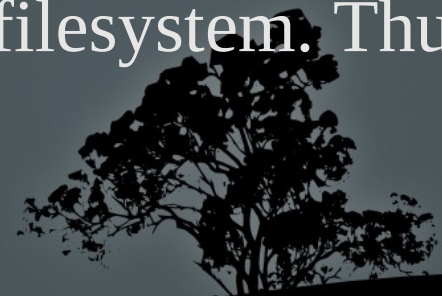
Directory Classification

	Shareable	Unshareable
Static	/usr	/etc
	/opt	/boot
Variable	/var/mail	/var/log
	/var/spool/news	/var/run
	/home	/proc

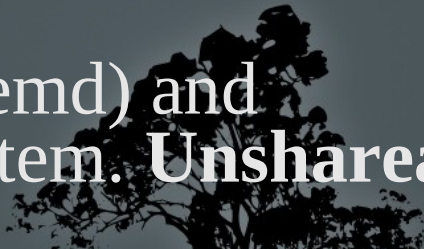


The *root* (/) filesystem


- It is the "root" of the directory hierarchy tree and contains all the basic configuration commands and libraries needed for the system to run. Because configuration is unique for each system this directory is **unshareable**
- It is desirable to be as small as possible. To do this safely the /boot, /usr, /home, /var and /tmp directories must be mounted to separate filesystems
- What remains in the root filesystems are the necessary applications for running and repairing the system, essential libraries and system configuration
- No application should be allowed to create variable data like files or directories under the root filesystem. Thus the root filesystem is **static**



Directories directly under the *root* filesystem

- **/bin**: contains essential system commands like bash, cp, ls, etc, intended for all users and these must be available during system recovery. **Unshareable/Static**
 - **/boot**: contains viable startup file like Linux kernel images (vmlinuz-*), ramdisks (initrd-*), boot loader files etc. It is usually a separate filesystem. **Unshareable/Static**
 - **/dev**: in legacy systems this directory contains statically created device files. In modern systems devices are created dynamically via **udev** and this is now a separate virtual **devtmpfs** filesystem. Device can be hot-plugged and unplugged so they are dynamically created. **Unshareable/Variable**
 - **/etc**: contains startup files (sysvinit, systemd) and configuration files unique for each system. **Unshareable/Static**
- 

Directories directly under the *root* filesystem

- **/home**: contains the home directories of normal users. It is usually a separate filesystem. **Shareable/Variable**
 - **/lib**: this directory contains essential system libraries and kernel modules necessary for the system applications to run. Sometimes it can be of the form **/lib32** or **/lib64**. **Unshareable/Static**
 - **/media**: usually used for hosting automatically created mount points after the mounting of removable media e.g. **/media/floppy**, **/media/cdrom/**, **/media/dvd** etc. **Unshareable/Variable**
 - **/mnt**: usually used for temporary, manual mounting of filesystems. **Unshareable/Static**
 - **/opt**: used for the installation of 3rd party applications, not part of the distributions repositories.
- 

Directories directly under the *root* filesystem


- **/proc**: virtual filesystem used for system and process information. **Unshareable/Variable**
- **/root**: the home directory of the **root** user. Lies on the root filesystem. **Unshareable/Static**
- **/sbin**: used for the essential recovery and maintenance applications and must be available during recovery. That's why it should always be hosted under the root filesystem. **Unshareable/Static**
- **/srv**: used for saving application read-only and writable data files, cgi scripts and other application specific data. **Unshareable/Variable**
- **/tmp**: used for saving temporary system and application data which are deleted on restart. Every user can read and write to it. For security reasons it should be placed on a separate filesystem. **Shareable/Variable**

Directories directly under the */usr* filesystem

- **/usr**: lives under the root directory and it is typically read only. It hosts user applications, non essential system applications, user libraries, static data, documentation etc. Usually hosted on a separate filesystem.
Shareable/Static
- **/usr/X11R6**: application and data for the X Window system graphical subsystem
- **/usr/bin**: hosts the user applications which come from the distributions packages
- **/usr/include**: contains header files for use by C and other programming languages
- **/usr/lib**: shared libraries for user programs



Directories directly under the */usr* filesystem

- ***/usr/local***: used for installing applications from the source code. These programs are not packaged by the distribution
 - ***/usr/sbin***: non essential system programs
 - ***/usr/share***: architecture independent shared information and data
 - ***/usr/share/<prog_name>***: initial, read-only application data
 - ***/usr/share/doc/<prog_name>***: application documentation
 - ***/usr/share/man/****: man pages
 - ***/usr/src***: source code for the system or the applications
- 

Directories directly under the `/var` filesystem

- `/var`: the `/var` directory contains variable files and data like log files, spools, caches, database data, etc. **Variable**
- `/var/cache`: host temporary or ephemeral data that do not need to be backed up, like package manager metadata and files
- `/var/lib`: hosts application variable data, e.g. databases
- `/var/lock`: lock files. These files declare that some system resources are locked by a process and cannot be used unless they are released by the process, or the process terminates. In modern systems it's just a symlink to `/run/lock`



Directories directly under the `/var` filesystem

- `/var/log`: hosts log files. They are used for documenting the operation of the system and applications, errors, debugging, etc.
- `/var/mail`: user mailboxes
- `/var/opt`: variable data for applications hosted in `/opt`
- `/var/run`: contains files that are depended on running processes like sockets and `.pid` files. The latter contain the **PID** of the processes of services. In modern systems it's simply a symlink to `/run`
- `/var/spool`: hosts spools for system like **cron**, **email** and the **CUPS** printing system
- `/var/tmp`: used for temporary data but unlike `/tmp` these remain after restart



Additional information about the *FHS*

- <http://refspecs.linuxfoundation.org/fhs.shtml>
The new page of the Filesystem Hierarchy Standard under the Linux Foundation
- http://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard
Wikipedia is always a valuable resource



The `find` command

- The **find** command is among the most useful system command because it let us search for files
- Has numerous options for search customization
- It can run commands against its results with **-exec**
- It does not index the results so each time we search the search performance is not improved



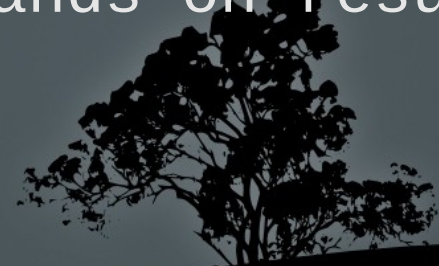
The `find` command

- `$ find . # show the absolute path of files under the working directory`
- `$ find /etc # find all files under /etc`
- `$ find / -name fstab # find all files named fstab`
- `$ find / -name "*.pl" # find all files ending in .pl`



Options of the `find` command

- **-atime** # access time
- **-mtime** # modified time
- **-ctime** # changed time
- **-empty** # empty files
- **-type** # define file type
- **-uid** # user id
- **-gid** # group id
- **-user** # user name
- **-group** # group name
- **-name** # file name
- **-inum** # inode id
- **-samefile** # search for files with the same inode (hard links)
- **-regex** # regular expression
- **-size** # file size
- **-delete** # delete files
- **-exec** # execute commands on results



The `locate` command


- The **locate** command is also used to find files on a system
- A basic difference with **find** is that it indexes the system files under `/var/lib/mlocate/mlocate.db`. This provides the advantage of yielding faster results than **find**
- To update the index the **updatedb** command is used
- It has fewer options than **find**



The `locate` command

- `$ locate fstab` # find all files with the name `fstab` in the path
- `$ locate "*.pl"` # find all files ending in `.pl`

Options:

- `-b` # match only the name, not the path
 - `-c` # print number of files
 - `-i` # ignore case
 - `--regex` # use regular expressions
- 

The `updatedb` command

- The **updatedb** create an index under **/var/lib/mlocate/mlocate.db** for use by **locate**
- Its configuration file is under **/etc/updatedb.conf** and here we can set exceptions for directories and filesystems
- ```
updatedb # update index
```
- ```
# updatedb -e "/tmp,/var/tmp,/proc" #  
update index, exempting the  
directories in the double quotes
```



The */etc/updatedb.conf* configuration file

- `$ cat /etc/updatedb.conf # show file contents`

```
PRUNE_BIND_MOUNTS="yes"
```

```
# PRUNENAMES=".git .bzip .hg .svn"
```

```
PRUNEPATHS="/tmp /var/spool /media  
/home/.ecryptfs"
```

```
PRUNEFS="NFS nfs nfs4 rpc_pipefs afs binfmt_misc  
proc smbfs autofs iso9660 ncpfs coda devpts  
ftps devfs mfs shfs sysfs cifs lustre_lite  
tmpfs usbfs udf fuse.glusterfs fuse.sshfs  
curlftpfs ecryptfs fusesmb devtmpfs"
```



Find the path of commands with `which`

- The **which** command looks into the **\$PATH** for command names
- `$ which locate # find the path of
locate
/usr/bin/locate`
- `$ which mkfs.ext2 mkfs.ext3 # find two
paths
/sbin/mkfs.ext2
/sbin/mkfs.ext3`
- `$ which type # no results since type is
a builtin command`



Find the path of commands and other info with `whereis`

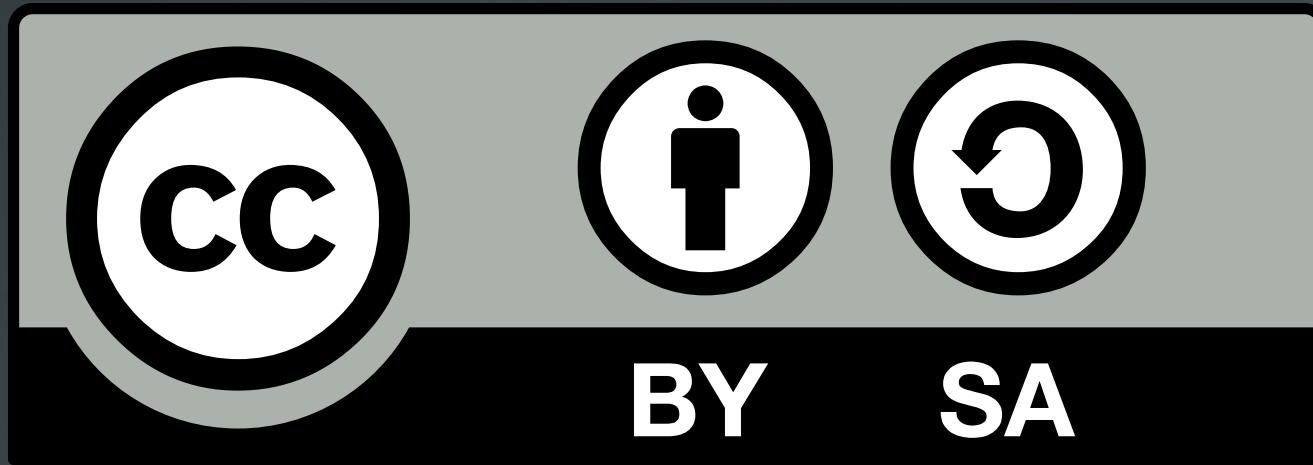
- The **whereis** provides additional information to **which** like man pages and source code
- `$ whereis pwd # show command path, source code files # and man pages`
- `pwd: /bin/pwd /usr/include/pwd.h /usr/share/man/man1/pwd.1.gz`

Options:

- `-b # search for command only`
- `-s # search for source code`
- `-m # search for man pages`



License



The work titled "LPIC-1 101-400 – Lesson 23" by Theodotos Andreou is distributed with the Creative Commons Attribution ShareAlike 4.0 International License.

