Filesystem Hierarchy and Permissions

Linux

Prepared by Steven Gordon on 19 April 2017
Common/Reports/linux-file-permissions.tex, r1417

Multiuser and Server Operating System

- Linux systems are commonly used as a multi-user system
  - E.g. multiple users have account on a shared computer
- Linux systems are commonly used as servers
  - Web, email, SSH, database servers
- How to ensure that authorized users can access only designated resources on a Linux system?
  - Understand filesystem organisation
  - Understand access control mechanisms on the filesystem

Contents

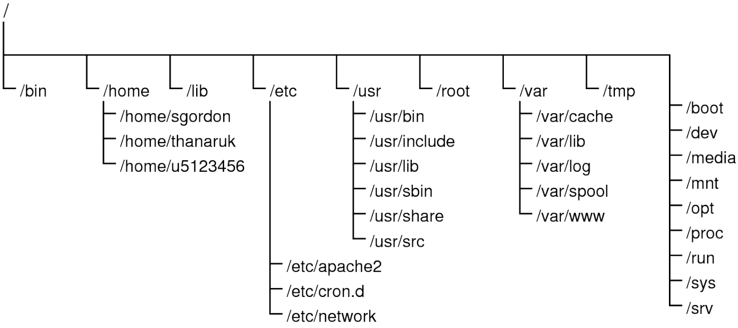
Linux Filesystem Hierarchy

Filesystem Organisation with inodes

Filesystem Access Control

Linux Filesystem Hierarchy

- Most UNIX and UNIX-like operating systems have *similar* filesystem hierarchies, e.g. Solaris, Ubuntu, RedHat, OSX, FreeBSD
- Directories and files
- Root directory is /
- An example Linux filesystem hierarchy (incomplete):



Linux Filesystem Hierarchy

/bin	essential binaries, e.g. `ls`, `cat`, `cp`
/boot	files needed to boot
/dev	devices
/etc	system configuration files
/home	users' home directories
/lib	libraries needed for binaries in `/bin` and `/sbin`
/media	mount points for USB, CDs etc.
/mnt	mount points for temporary filesystems
/opt	optional applications
/proc	information about running processes and kernel
/root	home directory of `root` user
/sbin	essential system binaries, i.e. requires `root` access
/srv	data made available by this system to others
/sys	information about devices
/usr	secondary hierarchy for install applications
/var	variable/temporary files, e.g. logs, inboxes, websites, caches

Where are applications installed?

Applications have files in multiple directories. Common naming scheme:

- bin binaries, i.e. executable applications (`sbin` for system binaries)
- lib libraries that applications use
- include header files, e.g. `.h`
- src source code, e.g. `.c`
- share documentation, template, data files of applications

Different locations for different types of applications:

- / for operating system applications
- /usr `usr` for installed applications
- /usr/local `usr/local` for installed applications specific to this computer

Which directories are important for new users?

Your files	`/home/username`
External drives	`/media`
OS configuration	`/etc`
Websites	`/var/www`
OS logs	`/var/log`

More advanced users …

Root user files	`/root`
OS processes	`/proc`
OS devices	`/dev` and `/sys`
Incoming email	`/var/mail`
App data	`/var/lib`

Contents

Linux Filesystem Hierarchy

Filesystem Organisation with inodes

Filesystem Access Control

inodes

- Files and directories administered by operating system using inodes
- inode is data structure that stores important information about a file or directory
  - mode
  - owner information
  - size
  - timestamps
  - pointers to data blocks (data blocks contain the actual file)
- OS maintains list of inodes in inode table
- Directories are a file that lists an entry for each file in that directory
  - inode number of file
  - length of name of file
  - name of file

inode Contents

mode 16 bits

- 12 protection bits: permissions
- 4 bit file type: regular file, directory, . . .

owner id 16 bit user ID

group id 16 bit group ID

size size of file in bytes

timestamps last time, in seconds since epoch:

- atime: inode accessed
- ctime: inode changed
- mtime: file data modified

and other fields ...

Contents

Linux Filesystem Hierarchy

Filesystem Organisation with inodes

Filesystem Access Control

Permissions and Users

Permissions

- read the file; list the contents of the directory
- write to the file; create and remove files in the directory
- execute the file; access files in the directory

Categories of Users

- user that owns the file
- users in the file's group
- other users
- (all users, i.e. the above three)

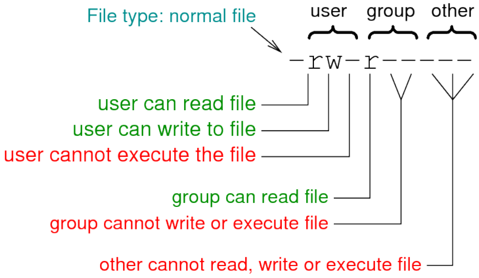
Permissions and Users

Special Permissions

- setuid bit: Set the process's effective user ID to that of the file
  - Directory: files created in that directory are given same user owner as the directory
- setgid bit: Set the process's effective group ID to that of the file
  - Directory: files created in that directory are given same group owner as the directory
- sticky bit: prevent users from removing or renaming a file unless they are user owner

Protection bits in an inode

- 12 bits in an inode are protection bits
  - First 9 bits indicate read, write, execute permissions for user, group and others
  - Last 3 bits indicate special permissions
- File type (regular or directory) and values of protection bits shown in user-friendly format
  - First letter indicates file type: directory; – is normal file
  - Next 9: Letter indicates the permission is set; – indicates the permission is not set



Useful Commands

Common Linux Commands

ls list directory contents, showing information about file (including permissions)

stat display file (or file system) status, including inode information

df report file system disk space usage

chmod change file mode bits, i.e. set permissions

Special Linux Commands

lsattr list special file attributes maintained by file system

chattr change special file attributes