Files & Permissions

File, directory and inode

- **File:** A file represents a sequence of bytes.
  - Each file will have a name
  - Special characters are allowed but need to be used carefully
- **Directory:** A directory represents a list of files.
  - **A directory is also a file** which contains the list of files containing in it.

    Every directory and file will be listed in its parent directory
- **Inode:** An inode (Index Node) contains information about a file (metadata) File permissions, UID, GID, Size, Time Stamp etc.

Permissions

- **File Permissions:** There are 3 permissions for any file r, w, x.

1. Read (r) - Indicates that a given category of user can read a file.
2. Write (w) - Indicates that a given category of user can write to a file.
3. Execute (x)- Indicates that a given category of user can execute the file.

- **Directory permissions:**

1. Read (r) - The direcotry can be read.
2. Write (w) - The directory can be updated, renamed or deleted.
3. Execute (x)- Operations can be perfomred on the files of the directories. This bit is also called as search bit, it indicates whether you are permitted to search files under that directory

- **Categories of users:** All of these three permissions are assigned to three
- categories of users User (U), Group(G), Others(O)

File types

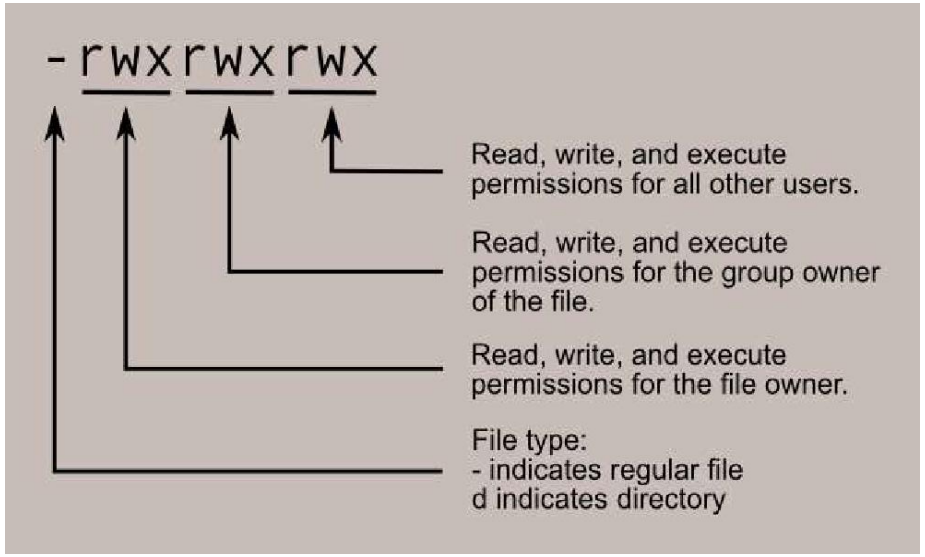
File Type	Symbol	Created by	Removed by
Regular file	-	Editors, cp, etc..	rm
Directory	d	mkdir	rmdir, rm -f
Character device file	c	mknod	rm
Block device file	b	mknod	rm
UNIX domain socket	s	socket(2)	rm
Named pipe	p	mknod	rm
Symbolic link	l	ln -s	rm

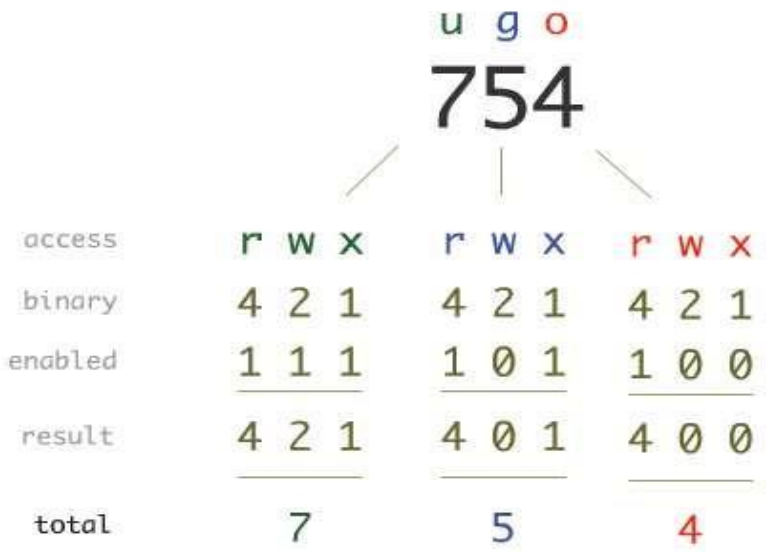
File Types Table

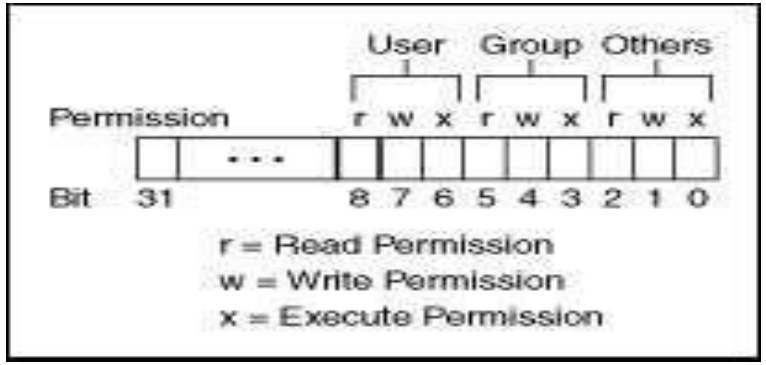
File Permissions

- Each file has a set of permissions that control who can mess with the file.
- There are three types of permissions:
  - read abbreviated **`r`**
  - write abbreviated **`w`**
  - execute abbreviated **`x`**
- There are 3 sets of permissions:
  1. user
  2. group
  3. other (the world, everybody else)

Access Permissions…







ls -l permissions

- **Type of file:**
- **-plain file d directory**
- **s symbolic link (others)**

r w x permissions

- Directory:
  - r – allowed fro listing
  - w – allowed for add or remove files
  - x – allowed to enter the directory
- Files:
  - R – allowed to read
  - W – allowed to write
  - X – allowed to execute

Change permissions

- “chmod” is the command to change the permissions for file and directory
  - Syntax : chmod xxx <filename>

Chmod with numeric value

- Consider for user
  - r – 4
  - w – 2
  - x - 1
- Consider for user, group and other
  - 755 – rwxrw_rw_
  - 611 – rw___x__x

chmod examples

- $ **chmod 700 CS571**
- $ **lso Personal**
- drwx------ 10 kschmidt 4096 Dec 19 2004 CS571/
- $ **chmod 755 public_html**
- $ **chmod 644 public_html/index.html**
- $ **lsao public_html**

chmod – symbolic modes

- Symbolic modes
  - U – user
  - G – group
  - O – other
  - A – all
  - + add permission
  - - remove permission
  - = set permission

File links

- Hard links
- Symbolic links

Hard links

- Hard link is a reference to the physical data on
- a file system
- All named files are hard links
- More than one name can be associated with
- the same physical data
- Hard links can only refer to data that exists on the ***same*** file system
- You can **not** create hard link to a directory

Soft links

- Also Known As Soft links or Symlinks
- A Symbolic Link is an indirect pointer to a file
- You can create a symbolic link to a directory
- A symbolic link can point to a file on a different file system
- A symbolic link can point to a nonexistent file