UNIT 4: File Attributes

BCAN 601: UNIX and Shell Programming

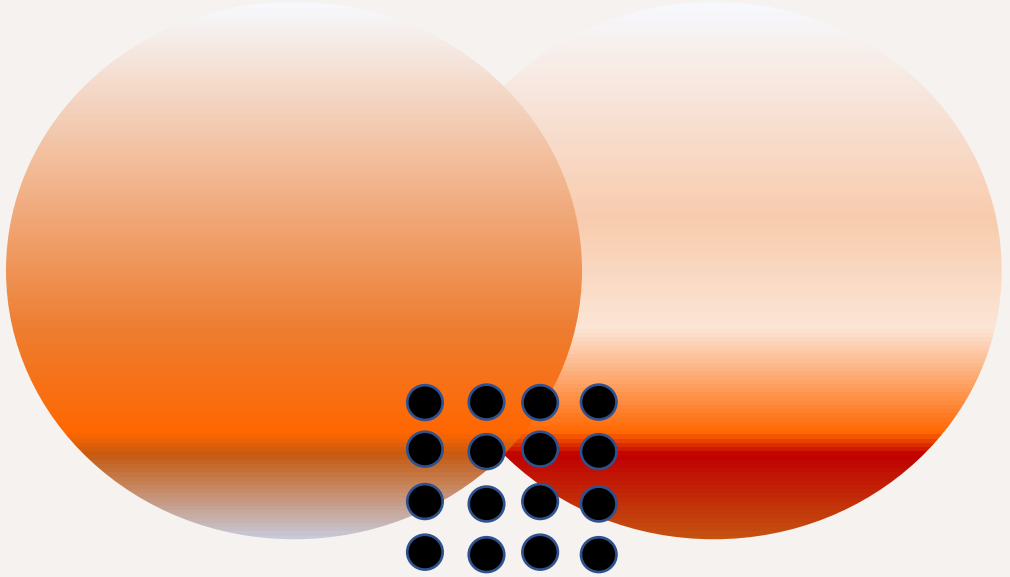

Table of Content

- File and directory attributes listing and very brief idea about the attributes, File ownership, File permissions,
- Changing file permissions – relative permission & absolute permission,
- Changing file ownership, Changing group ownership
- File system and inodes
- Significance of file attribute for directory
- Default permissions of file and directory and using umask
- Listing of modification and access time, Time stamp changing (touch), File locating (find)

File Attributes

- All the data stored in computer in a file.
- Every day we come across the file attributes.
- Find fine attributes are the characteristic of file itself.

File Attribute

- Name: file name is the name given to the file. A name usually a string of characters
- Identifiers: identifier is a unique number for a file. Identifies files within the file system
- Type: type is another attribute of a file which specifies the type of file such as archive(.zip), source file(.c,.java)
- Location: specifies the location of the file on the device. this attribute is a pointer to a device.

File Attribute

- Size: specifies the current size of the file(in kb, mb) and possibly the maximum allowed size of the file.
- Protection: specifies information about access control. Provides security to sensitive and private information

File Attribute

- Some other attributes may include attributes related to flags. Flags control or enable some specific property.
- Read only flag: 0 for read/write, 1 for read only
- Hidden flag: 0 for normal, 1 for do not display in listing of all files
- Archive flag: 0 for has been backed up, 1 for needs to be backed up
- ASCII/ binary flag: 0 for ASCII file, 1 for binary file
- Random access flag: 0 for sequential access, 1 for random access

File Attribute

- Temporary flag: 0 for normal, 1 for deleted file on process exit
- Lock flag: 0 for unlocked, non zero for locked

- File may posses different attributes as per the requirement. The attributes varies from system to system.
- Attributes are also stored in secondary storage. Attributes provide extra information about the files which can be useful.

File Ownership

- UNIX file system have three types of owner.
  - User: a user is one who created the file. By default , whosoever, creates the file become the owner of the file. A user can create, delete or modify the file.
  - Group: a group can contain multiple users. All users belonging to a group have same access permission for a file.
  - Other: any one who has access to the file other than user and group comes in the category of other.

File permission

- When many users are sharing one file system. It is important to be able to restrict access to certain file
- The system administrator want to prevent other users from changing important system files.
- Many users have private file that they want to restrict others from viewing
- File permissions are designed to address these needs.

File permission

- There are three classes of file permissions for the three classes of users
  - The owner of the file
  - the group of the file belong to
  - all the other users of the system
- The first three letters of the permission field as seen in the output of a long format ls refers to the owner's permission.
- The next three letters refers to the permissions for members of the file group, and the last three letters for other users of the system.

File Permission

- $ $ls\ -l\ note.txt$
- $-rwxr\_xr\_x\ note.txt$
- The first three letters rwx shows that owner of the file can read, write and execute
- The second three letters r_x shows that the permission for the group can read and execute, but can not write.
- The third three letters r_x shows that other users can read and execute, but can not execute.

File Permission

- If you have read permission you can view the content of the file
- Write permission means you can alter the file content
- Execute file permission means that you can run the program.

Special Permission

- There are few other codes that occasionally appear in the permission fields.
- The letter l appear in the place of an $r, w\ or\ x$. This means that the file will be locked when it is accessed, so that other users can not access it while it is being used.
- This s is relevant to programmers and administrator.

Permission for Directories

- For directories read permission allows users to list the content of the directory.
- Write permission allows user to create or remove files or directories inside that directory.
- Execute permission allows user to change this directory using cd command or use it as part of a path name.

$chmod$ command

- The UNIX system allows user to set the permission of each file you won. Only the owner of the file or superuser can alter the permission.
- You can independently manipulate each of the permission to allow or prevent reading, writing or executing by yourself, group or other users.
- $chmod$ is used to modify the file permission.

$chmod$ command

- First, show which set of permissions you are changing with u for user, g for group, and o for other users.
- Secondly, specify how they should be changed with $+$(add permission) or $-$(to remove permission).
- Third, list the permission to alter: r for read, w for write, x for execute
- Specify the file or files that the changes refer to

$chmod$ command

- Syntaxx of $chmod$:
- $\$\ chmod\ category\ operation\ permission\ filename$
- $chmod\ go\ -wx\ file.txt$

Setting Absolute Permission

- how to set file permission without knowing the existing one
- *chmod* can be used with three octal numbers
- Read permission- 4
- Write permission-2
- Execute permission-1
- *chmod* use three-digit string as the expression

Absolute Permission

Binary	Octal	Permission	Significance
000	0	---	No permission
001	1	--x	Executable only
010	2	-w-	Writeable only
011	3	-wx	Writeable and executable
100	4	r--	Readable only
101	5	r-x	Readable and executable
110	6	rw-	Readable and writable
111	7	rwx	Readable, writeable and executable

Change group

- The `chgrp` command in Linux is used to change the group ownership of a file or directory.
- All files in Linux belong to an owner and a group. You can set the owner by using “chown” command, and the group by the “chgrp” command.
- First we need to have administrator permission to add or delete groups. We can login as root for this purpose or use sudo. In order to add a new group
- $chgrp group_name file
- Change Group Ownership of Multiple Files
- Change Group Ownership of a Directory or Folder

Change owner

- file ownership is a crucial aspect of system security and user management.
- The `chown` command, short for “change owner,” is a powerful tool that allows users to change the owner of files and directories.
- This command is particularly useful in scenarios where administrators need to grant or revoke access to specific resources.
- Syntax of chown Command in Linux
- $ $chown\ [options]\ new\_owner[:new\_group]\ file(s)$

Change owner

Option	Description
$-c$	Report file changes
$-v$	showing detailed information for every processed file.
$-f$	suppress most error messages and forcefully or silently change ownership, even when not permitted.

- $ $chown : group1\ file1.txt$
- the command designates the user “master” as the new owner of the file `file1.txt`. This is particularly useful when transferring ownership of files between users.

Default file and directory permission

- create files and directories based on the default permission
- The UNIX system has following default permission for all files and directories
- $rw - rw - rw - \ (octal\ 666) for\ regular\ files$
- $rwxrwxrwx\ (octal\ 777) for\ directories$
- This default is transformed by subtracting the user mask from it to remove one or more permissions.
- $ $umask$ 022
- This is an octal number which has to be subtracted from the system default to obtain the actual default.

Modification and access time

- A UNIX file system has three time stamps associated with it.
- Time of last modification (show by $ls\ -l$)
- Time of last access (show by $ls\ -lu$)
- Time of last inode modification (show by $ls\ -lc$)

Changing the timestamp

- Sometimes it is required to set modification and access time
- The touch command changes the times in following manner
- $ $touch\ option\ expression\ filename$
- $touch$ without option set the access time to current time

Locating files

- Search a file in the system using find command
- $ $find\ path\_list\ selectionCriteria\ Action$
- $ $find\ .-name\ dataFile\ -print$