Chapter: 5 Controlling Access to Files

Objectives:

- List the file system permissions on files and directories, and interpret the effect of those permissions on access by users and groups.
- Change the permissions and ownership of files using command-line tools.
- Control the default permissions of new files created by users, explain the effect of special permissions, and use special permissions and default permissions to set the group owner of files created in a particular directory.

LINUX FILE-SYSTEM PERMISSIONS

Effects of Permissions on Files and Directories

PERMISSION	EFFECT ON FILES	EFFECT ON DIRECTORIES
r (read)	Contents of the file can be read.	Contents of the directory (the file names) can be listed.
w (write)	Contents of the file can be changed.	Any file in the directory can be created or deleted.
x (execute)	Files can be executed as commands.	Contents of the directory can be accessed. (You can change into the directory, read information about its files, and access its files if the files' permissions allow it.)

VIEWING FILE AND DIRECTORY PERMISSIONS AND OWNERSHIP

For File:

ls –l filename

For Directory:

ls –ld directoryname

MANAGING FILE SYSTEM PERMISSIONS FROM THE COMMAND LINE

Objectives

After completing this section, you should be able to change the permissions and ownership of files using command-line tools.

CHANGING FILE AND DIRECTORY PERMISSIONS

The command used to change permissions from the command line is chmod, which means, "change mode".

Changing Permissions with the Symbolic Method

chmod WhoWhatWhich file|directory

- Who is u, g, o, a (for user, group, other, all)
- What is +, -, = (for add, remove, set exactly)
- Which is r, w, x (for read, write, execute)

Changing Permissions with the Numeric Method

chmod ### file|directory

Each digit represents permissions for an access level: user, group, other.

The digit is calculated by adding together numbers for each permission you want to add,

4 for read,

2 for write, and

1 for execute.

CHANGING FILE AND DIRECTORY USER OR GROUP OWNERSHIP

chown user_owner:group_owner file|directory

MANAGING DEFAULT PERMISSIONS AND FILE ACCESS

OBJECTIVES

After completing this section, students should be able to:

- Control the default permissions of new files created by users.
- Explain the effect of special permissions.

- Use special permissions and default permissions to set the group owner of files created in a particular directory.

SPECIAL PERMISSIONS

Special permissions constitute a fourth permission type in addition to the basic user, group, and other types. As the name implies, these permissions provide additional access-related features over and above what the basic permission types allow.

Effects of Special Permissions on Files and Directories

SPECIAL PERMISSION	EFFECT ON FILES	EFFECT ON DIRECTORIES
u+s (suid)	File executes as the user that owns the file, not the user that ran the file.	No effect.
g+s (sgid)	File executes as the group that owns the file.	Files newly created in the directory have their group owner set to match the group owner of the directory.
o+t (sticky)	No effect.	Users with write access to the directory can only remove files that they own; they cannot remove or force saves to files owned by other users.

Setting Special Permissions

Symbolically: setuid = u+s; setgid = g+s; sticky = o+t

Numerically (fourth preceding digit): setuid = 4; setgid = 2; sticky = 1

UMASK (User Mask or User file creation MASK)

The minimum and maximum UMASK value for a directory is 000 and 777

The minimum and maximum UMASK value for a file is 000 and 666