In this chapter, we will discuss in detail about file permission and access modes in Unix. File ownership is an important component of Unix that provides

**Ownership:**

Every Linux system have three types of owner:

1. **User:** A user is the one who created the file. By default, whosoever, creates the file becomes the owner of the file. A user can create, delete, or modify the file.
2. **Group:** A group can contain multiple users. All the users belonging to a group have same access permission for a file.
3. **Other:** Any one who has access to the file other than **user** and **group** comes in the category of**other**. Other has neither created the file nor is a group member.

Users and groups can be locally managed in **/etc/psswd** or **/etc/group.**

**File Access Modes:**

The permissions of a file are the first line of defence in the security of a UNIX scheme. The basic building blocks of Unix permissions are the read, write, and execute permissions, which have been describe below −

Read

Grants the capability to read, i.e., view the inside of the file.

Write

Grants the capability to modify, or remove the content of the file.

Execute

User with execute permissions can run a file as a program.

* Note that each permission has a numeric value associated with it:

|  |  |
| --- | --- |
| Value | Permission |
| 4 | read (r) |
| 2 | write (w) |
| 1 | execute (x) |

So to understand this concept in a simpler way, think of file permissions as a 3x3 matrix, where owners, groups, and others each have r, w, and x settings.

* The file’s *creator* (owner/user) has read and write permissions: -**rw**-r--r--.
* The file’s group creator (group) has read permissions: -rw-**r--**r--.
* Others have read permissions represented by the last bits: -rw-r--**r--**.

Now, let’s see the default permission values for a directory. Let's say the directory chmod\_directory was created with the default permissions of 755. Unlike files, a directory has files in it. In order for anyone other than the owner to 'cd' into the directory, it needs an execute permission, which in turn makes the directory:

* Readable, writable and executable by the owner (rwx is 7).
* Readable and executable by the group (r-x is 5).
* Readable and executable for others (r-x is 5).
* Note: The r-x designation does NOT mean r minus x, it means read and execute but missing write. The - is a placeholder for a permission.

**Using chmod in symbolic mode**

The problem with the absolute mode is that you should always provide three numbers for all the three owners even if you want to change the permission set for just one owner.

This is where you can use the symbolic mode with chmod command.

In symbolic mode, owners are denoted with the following symbols:

* u = user owner
* g = group owner
* = other
* a = all (user + group + other)

The symbolic mode uses mathematical operators to perform the permission changes:

* + for adding permissions
* - for removing permissions
* = for overriding existing permissions with new value

if you want to add execute permission for group owner, you can use chmod command like this:

chmod g+x agatha.txt