

Normal user:

group 1. by default a normal user has access to his own home directory location only. The files and folders that are created by an user, can be accessible only by the owner/user itself by default. Because when we add an user into the Linux operating system computer, the user will be created by default with a primary group as the user group itself. So no one other than the root and the owner itself can access his/her files or folders created.

From the above we can think in Linux operating system environment, each user is isolated from others on the machine as no one can see/access the files and folders of others. Even though we can think this as advantage interms of security, in another way it is an biggest limitation, because in a corporate or organization environment we wanted multiple users to collborate and work together in sharing the resources or files created by others.

To allow an Linux user to grant access permissions in allowing the other users of the linux operating system to access their files/folders, Linux has introduced File permissions.

How does the File permissions works in Linux? Let us say, Linux operating system allows: an user of the Linux operating system can grant access permissions for a File or folder of his own to other users of the Linux operating system individually.

A user might have several Files and Folders of his own, granting each of them individually to different linux users is:

- 1. very complex approach and takes lot of time in granting permissions for each file/folder to every other linux users whoever we want to share
- 2. keeping track of which user has been granted the permissions of which files/folders is very difficult.

3. changing the access permissions and revoking them is very difficult.

So to overcome the above problems in granting File/Folder permissions, the Linux operating system has introduced "Groups".

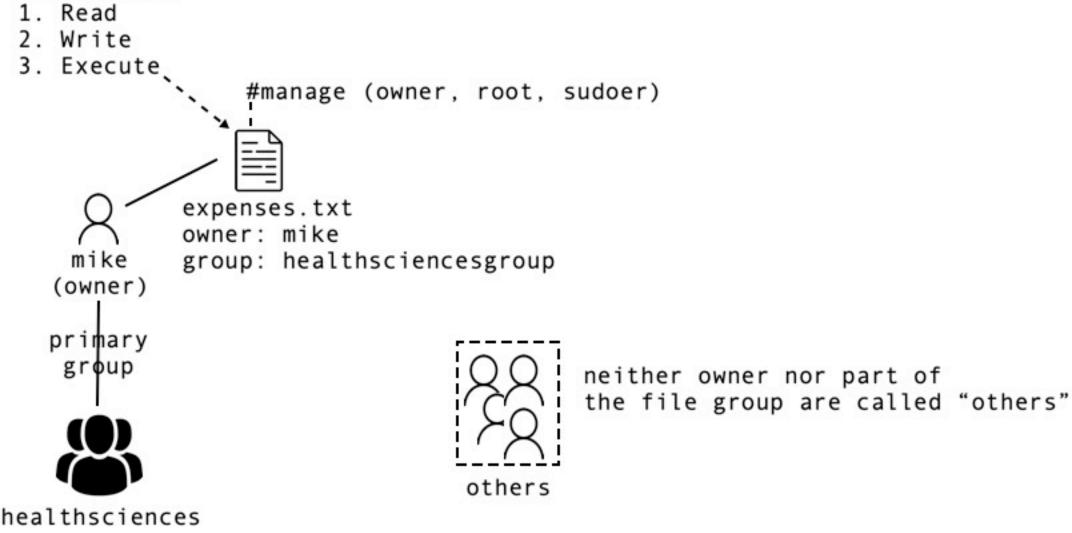
Group:

Permissions

group

A linux group is nothing but an logical entity used for combining a multiple users of the Linux operating system into one umberella.

Rather than granting permissions to a File or Folders to another user of the Linux operating system, Linux encourages us to grant permissions to a group of users (Linux Group).



Granting access to a File/Folder means providing access permissions on that file/ folder. Typically on a File/Folder we can manage #3 types of permissions

- 1. Read
- 2. Write
- 3. Execute

These permissions of a File/Folder can be granted to #3 different people

- owner
- 2. group (which the file/folder belongs to)
- others

What does these permissions meant to be on a File? For a file we can grant #3 types of permissions

1. Read = users can see the contents of the file. like cat filename, or we can open the file in nano or vi or any text editor to see its contents.

2. write = we can modify the contents of the file

3. execute = if the file contains programming instructions that can be the computer like .sh (shellscript file) etc

What does the permissions on a Folder?

On a Folder/directory also we can grant #3 types of permissions 1. Read = read permission on a folder means we can see the contents of the folder. which means we can run ls command on the folder. but we cannot go (cd) into the folder

ls folder (allowed)

cd folder (not allowed)

Write = write permission on a folder means we can create new files or folders executed, then execute permission let the people to run the program file on I inside the folder or we can delete existing files/folders inside that folder

> 3. Execute = Excecute permission on a folder means we can go into the folder by running cd command. If we dont have execute permission on a folder we cannot navigate into that

folder.

cd folder (allowed only when we have execute permission)

These permissions are represented or indicated in linux operating system using symbols as r = read

- w = writex = execute
- !- = no permission

ls -l filename

!typeownergroupothers owner group mike healthsciencesgroup 1 rw-rw-r-

symbolinks size filename 1024 expenses.txt