LINUX FILE PWERMISSIONS

Check Permissions in Command-line
*ls -l
*ls -l [file_name](for specific file/directory)

NOTE-:
above command "-l" means long list format

PERMISSION TYPES:
The three basic permission types found in Linux file and directory permissions are: read, write, and execute.

*Read. (r) The read permission allows users to view the contents of a file or list the contents of a directory.
*Write. (w) The write permission allows users to modify a file's contents or add, remove, or rename files within a dire
*Execute. (x) The execute permission allows users to execute a file or traverse (i.e., enter) a directory. For files, execute permission is required to run the file as a program or script. For directories, execute permission is required

PERMISSION GROUPS:

*Owner (the user who created the file/directory).
*Group (which the owner belongs to).
*Others (all other users)

How to Read Linux Permissions:

here i have mentioned a example below .

example:
ubuntu@ip-172-24-0-10:~ ls -l kishore.txt
-rw-r--r-- 1 ubuntu ubuntu 20 aug 27 2024 kishore.txt

*The first character indicates the file type - a regular file (-), directory (d), symbolic link (i), etc.
*The next three characters represent the user's (owner's) permissions.
*The three characters after that are the group's permissions.
*The final three characters are the permissions for all other users.

How to Change Permissions in Linux
There are two primary methods for changing permissions in Linux:

*Absolute mode or numeric mode
*Symbolic mode.

ABSOLUTE MODE OR NUMERIC MODE:

Absolute Mode
Another way to specify permission is by using the octal/numeric format.
This option is faster, as it requires less typing, although it is not as straightforward as the symbolic mode.

Instead of letters, the octal format represents privileges with numbers:

read has the value of 4.
write has the value of 2.
execute has the value of 1.
no permission has the value of 0.
The privileges are summed up and depicted by one number. Therefore, the possibilities are:

7 - for read, write, and execute permission.
6 - for read and write privileges.
5 - for read and execute privileges.

4 - for read privileges.
3 - for write and execute privileges

As you have to define permission for each category (user, group, owner), the command includes three numbers (eac

For example, let’s look at the kishore.txt file.

chmod 777 kishore.txt

i have given full permissions for user and group and others(read,write,execute).

SYMBOLIC MODE:

chmod u=rwx,g=rwx,o=rwx kishore.txt

i have given full permissions for user and group and others(read,write,execute).

Change File Ownership:

To change the file ownership, use the chown command. The syntax is:

chown [user_name] [file_name/directory]

example:
chown tarak kishore.txt

NOTE:
in above command tarak is user name and "kishore.txt" is file name.

Replace [user_name] with the name of the user you want to make the new owner of the file or directory.

Change Group Ownership:

To change the group ownership, use the chgrp command. The syntax is:

chgrp [group_name] [file_name/directory]

Ex:

example:
chown dev-team kishore.txt

Note:
in above command dev-team is group name and "kishore.txt" is file name.

********** LINUX FILE STRUCTURE**********

Linux File Structure
Linux organizes files and directories in a hierarchical structure, starting from the root directory /.
The root directory contains several subdirectories, each with a specific purpose and function.
Here are some of the common subdirectories and their descriptions:

/bin: Contains binary executable files that are essential for the system to run, such as ls, cp, mv, etc.
/boot: Contains files needed for booting the system, such as the kernel image and the boot loader.
/dev: Contains device files that represent hardware devices, such as disks, keyboards, mice, etc.
/etc: Contains configuration files for the system and various applications, such as /etc/passwd, /etc/hosts, etc.

/home: Contains the home directories of regular users, where they can store their personal files and settings.
/lib: Contains library files that are needed by the binary files in /bin and /sbin.
/media: Contains mount points for removable media, such as CDs, DVDs, USB drives, etc.
/opt: Contains optional software packages that are not part of the standard distribution.
/proc: Contains virtual files that provide information about the system processes and kernel parameters.
/root: Contains the home directory of the superuser or root user, who has complete control over the system.
/sbin: Contains binary executable files that are used for system administration, such as fdisk, ifconfig, mount, etc.
/tmp: Contains temporary files that are created and deleted by various programs.
/usr: Contains user-related programs and data, such as applications, games, documentation, etc.
/var: Contains variable data that changes frequently, such as logs, caches, spools, etc.