**Permissions in Linux**

Linux File System Security restricts user to access the files and directories.

User require permissions to access files or directories.

“ls -l” or “ll” command can be used to check security of any file or directory.

Above command will display contents from directory along with its security details.

drwxr-xr-x 3 root root 4096 May 9 18:41 Linux

-rw-rw-r-- 1 root root 60582753 May 10 12:02 file1.txt

The first character = ‘- ‘ which means it’s a file, ‘d’ which means it’s a directory.

The next nine characters = (rw-r–r–) show the security

The number shows the symbolic link (1-file, others no.-directory)

The next column shows the owner of the file. (Here it is root)

The next column shows the group owner of the file. (Here it is root which has special access to these files)

The next column shows the size of the file in bytes.

The next column shows the date and time the file was last modified.

Last Column = File name or Directory name. (For example, here are: prac, snap, test, example)

**Types of files**

1. **Ordinary File/ Regular File (start with \_)**

* File which contains Data.
* Regular files are nothing but the everyday files used to store information such as text, or images.
* These files can be found in directories, which are yet another type of files.
* In Linux, regular files can exist with or without an extension.

1. **Directory File (Start with d)**

* Directory file is equal to folder, it contains files and folders.
* Directories are also Linux files. But rather than storing data, they store the location of other files. To achieve this, the directory uses directory entries.

1. **Link File (Start with l)**

* A link is a symbolic connection or pointer to a single file that allows us to access it from more than one directory.
* We can set up a link to a file in a restricted directory, allowing access to the file without providing access to the directory.

1. **Soft link**

* Link will be removed if original file removed.
* soft or symbolic link creates a pointer to the original file i.e. it creates shortcut of the file.

Syntax to create Softline,

ln –s <original\_file> <softlink\_name/path>

1. **Hard link**

* A hard link develops a mirror copy of the original file.
* Renaming, deleting or removing original file will not effect link.

Syntax to create hardlink,

ln <original\_file> <hardlink\_name/path>

1. **Named Pipe file (start with p)**

* used to establish two-way communications between two unrelated programs.
* Sends data from one process to another so that receiving process reads the data in FIFO manner.
* These files are present in /run folder.

1. **Socket file (start with s)**

* Special file to enable communications between two processes.
* These files are present in /run folder.

1. **Character file (start with c)**

* Reads and write data character by character.
* These files are present in /dev folder.
* A character device file is a hardware file that reads or writes data one character at a time in a file.

1. **Block Device file (start with b)**

* file that refers to device.
* a piece of hardware that provides data access in blocks

**Three types of permissions**

1. **Read (r):**

* Allows reading and viewing the contents of a file or directory.

1. **Write (w):**

* Grants the ability to modify, delete, or add to the contents of a file or directory.

1. **Execute (x):**

* Enables the execution of a file or allows access to a directory's contents if used with directory permissions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Octal** | **File** | **Directory** |
| r | 4 | Display the content | List the content |
| w | 2 | Modify the file | Create or remove directory |
| x | 1 | Execute the file | To enter into directory |

**These permissions are set for three categories of users:**

1. **Owner: (u)**

* The user who owns, create the file or directory.

1. **Group: (g)**

* Users who are members of the group associated with the file or directory.

1. **Others: (o)**

* All other users who are not the owner or part of the group.

1. **All: (a)**

* Permissions are represented by a series of symbols:

r: indicates read permission.

w: indicates write permission.

x: indicates execute permission.

-: indicates no permission.

**Symbols**

* + Add permissions
* - Remove permissions
* = Set the permissions to the specified values

**Umask**

* When a user creates a file or directory under Linux or UNIX, she makes it with default permissions.
* The user file-creation mode mask (umask) is used to determine the file permission for newly created files. It can be used to control the default file permission for new files.
* The default umask 002 used for normal user. With this mask default directory permissions are 775 and default file permissions are 664.
* The default umask for the root user is 022 result into default directory permissions are 755 and default file permissions are 644.
* For directories, the base permissions are (rwxrwxrwx) 0777 and for files they are 0666 (rw-rw-rw).
* It is a four-digit octal number. A umask can be set or expressed using:

1. Symbolic values
2. Octal values