chmod file permissions:

chmod stands for **change mode (**to change the permissions/mode of files and directories**)**.

In Linux, every file and directory has a set of permissions that determine who can read, write, and execute the file. These permissions are essential for maintaining security and controlling access to files and directories. The "**chmod**" command is used to change these permissions.

Permissions levels are denoted by:

u - (**current user**) yourself

g - **group** in which current **user belongs to**

o - **others**/public users

a - **all**

The **permission code** are represented by a **three-digit number or a combination of letters**. Each digit or letter corresponds to a specific set of permissions:

- **First Digit (Owner permissions):** Represents permissions for the **owner of the file/directory.**
- **Second Digit (Group permissions):** Represents permissions for the **group** associated/related **with the file/directory**.
- **Third Digit (Others/public permissions):** Represents permissions for all other users(i.e public users) who are **not the owner** and **not in the group**, **of that file**.

Three types of permissions & denoted by:

- **r (read): Allows reading the file** or **listing the directory contents,** if the permission is set.
- **w (write): Allows modifying/writing the file** or **adding/removing files in a directory,** if the permission is set.
- **x (execute): Allows executing a file** or **accessing/entering inside a directory,** if the permission is set.

Here are some examples of **permissions representations**:

- **Numeric representation:** Each digit represents the sum of the following values: 4 (read), 2 (write), and 1 (execute).

  - For example, **7 = 4 (read) + 2 (write) + 1 (execute)**, so, **7** means **full permissions**.

- **Symbolic representation:** Uses letters to represent permissions. For example, "**r - -**" represents **read**, "**- w -**" represents **write**, "**- - x**" represents **execute**, while "**rwx**" represents all **read**, **write** and **execute** which means **full permissions**.

Syntax of the "**chmod**" command is as follows:

⇒ **chmod [options] mode file(s)**

- "**chmod**" is the command itself.
- **[options]** are optional flags that modify the behavior of the command. Some common options include:
  - `-R`: Recursively change permissions for directories and their contents.
  - `-v`: Verbosely show the result of the operation.
  - `-c`: Similar to `-v`, but only shows output for files that have their permissions changed.
- "**mode**" specifies the new permissions you want to set. It can be represented either in numeric format or symbolic format.
  - Numeric format: A three-digit number representing the sum of permissions for the owner, group, and others. For example, 755 or 644.
  - Symbolic format: Uses letters to represent permissions. For example, "u+rwx" (add read, write, and execute permission for the owner).
- "**file(s)**" are the files or directories for which you want to change the permissions.

Note: The "**mode**" and "**file(s)**" are the **essential** components of the "chmod" command, while the **options** are **optional** and can be omitted if you only need the basic functionality.

Some basic chmod examples/usage:

How to view permissions ??

1. **To view the permissions of all files/folders in current directory:**

   ⇒ ls -l

   OR

   ⇒ ls -ltr **NOTE:** sorts the listing by modification time in reverse order (i.e., from the newest modification to the oldest).

How to change permissions ??

NOTE: A 'root' user can change the permission of any files/folders of any current user(owner).

NOTE: But, a normal 'user' cannot change the permission of files/folders of any other user(owner).

## In Symbolic representation:

- **For Adding:**

  Read permission to user ⇒ SYNTAX: chmod u+r file_name

  Read permission to user, group & others ⇒ SYNTAX: chmod ugo+r file_name

  OR, best way (**using 'a' instead of 'ugo'**)

  Read, write & execute permission to all ⇒ SYNTAX: chmod a+rwx file_name

- **For Removing:**

  Read permission from user ⇒ SYNTAX: chmod u-r file_name

  Read permission from user, group & others ⇒ SYNTAX: chmod ugo-r file_name

  OR, best way (**using 'a' instead of 'ugo'**)

  Read, write & execute permission from all ⇒ SYNTAX: chmod a-rwx file_name

## In Numeric representation:

NOTE: numeric permission code moves from right to left.

(That means, **if u only assign single number permission to any file**, then linux assigns, that number permission code for **other users,** which is on the left most side. Now, linux **by default delete remaining all user and group permission. So, be careful while using numeric representation**.

Likewise, **if u only assign two number permission to any file**, then linux assigns, that **right side number** permission code for **other users** and **left side number** permission code for **group**.Now, linux **by default delete remaining all user permission. So, be careful while using numeric representation**.)

SYNTAX: **chmod number number number file_name**

- **For:**

# Give Read permission to only user:

⇒ **SYNTAX**: **chmod number number number file_name**

eg: chmod 400 file_name

# Add Read permission to user:

⇒ **SYNTAX**: **chmod number prev_number prev_number file_name**

eg: chmod 477 file_name

# Give Read, write & execute permission to all:

⇒ **SYNTAX**: **chmod number number number file_name**

eg: chmod 777 file_name