Everything About Chmod and Chown

#Linux #Ubuntu

Introduction

The chmod and chown commands are used to manage permissions on Linux and Unix-like operating systems. They allow users to modify the access rights for files and directories, which means that you can give certain people or groups of people permission to access them. This article will explain everything about these two commands and their uses!

The Chmod and Chown Command

The chmod command is a powerful tool that can be used to change permissions and ownership of files and directories. It has many different uses, but the most common use of this command is changing the owner or group of a file or directory.

You may have noticed that some programs give you an option to add or remove permissions when they're first installed or updated with special tools (such as RPM). These programs usually do this because they don't want people from messing with their system too much! But if you want more control over how these commands work in your own computer then there are two ways:

- By using another command called chown (which stands for "change owner") instead of just using one word like "chmod". This works exactly like how we might expect it would based off its name alone - only difference being that instead of changing ownership levels on individual files/directories within their own directory structure respectively; we're changing them globally across all available paths within our entire filesystem just below whichever root directory was chosen earlier during installation/upgrade process(es).

Chmod Symbols

Chmod uses the following symbols to set permissions:

- u for user
- g for group (also known as a supplementary group)
- o for others, which are any users other than the owner and superuser. In practice, this means everyone on your system unless you have explicitly given them privileges in some way. For example, if you're using sudo , then everyone else will have read access but not write or execute permissions. However, if you've used chmod 700 , then only root can run commands with that permission set!

- a for all (or no permission). This means that files are readable by all users; they don't need to be executable by anyone at all—just readable!

Changing the Permissions of a Directory

- To make a directory writable by all users, use chmod -R 755.
- To make a directory read-only for everyone except the owner and root user, use chmod -R 777.
- The same thing can be done with fewer characters: just change the permissions on the last letter of each word (e.g., rwxrwxrwx becomes rwxr-x---).

Changing the Owner and Group of a File

The chown command allows you to change the owner and group of a file. You can do this by using the -R option, which will recursively change all owners, groups, and permissions on a file. For example:

- To change the owner of your current directory (the directory where you are right now) to "user":
- chmod 777 *
- chown user:user *

This will make both yourself as well as other users in your group "users" have full access over this directory. If there were any files owned by another user or group already present in this path then those files would also be changed with their new ownership information included (assuming no other conflicting permissions).

About Chmod and Chown

chmod and chown are two commands that you can use to manage permissions on Linux and Unix-like operating systems.

chmod is used to change the permissions of a file or directory, while chown is used to change the owner and group of a file.

Conclusion

As you can see, the chmod and chown commands are very useful tools to manage your Linux or Unix-like computer. By changing the owner and group of a file or directory, you can make sure that files are only readable by those who have permission to view them and can't be changed by anyone else on your system.