

## chown

chown means change of owner.

Only root users can perform this activity for all files and directories. For regular users only files owned by them can be changed.

Syntax: `chown [Options] UserName File1 [File2 ...]`

Example: `chown root demo.txt` → Now the owner of demo.txt is root

Options:

- 1) `c` → Reports when a file change is made.
- 2) `v` → It is used to show verbose information for every file processed.
- 3) `f` → It suppresses most of the error messages. When you are not permitted to change group permissions and show error, this option forcefully/silently changes the ownership.
- 4) `--from = (UserName)` → To change the owner from particular ownership only.
- 5) `--reference = (FileName)` → To copy ownership of one file to another.
- 6) `R` → Recursively changes the owner of a Folder.

## chgrp

The `chgrp` command in Linux is used to change the group ownership of a file or directory. All files in Linux belong to an owner and a group. You can set the owner by using the “chown” command, and the group by the “chgrp” command.

Syntax: `chgrp [OPTION]... GROUP FILE1 [FILE2 ...]`  
`chgrp [OPTION]... --reference=RFILE FILE...`

Example: `sudo chgrp CodingBlocks abc.txt`

Options:

- 1) `R` → To recursively change the group ownership of a folder and all of its contents.
- 2) `--reference = (FileName)` → To copy group ownership of one file to another.
- 3) `c` → To describe the action for each File whose group actually changes.
- 4) `f` → To suppress error messages.
- 5) `v` → To describe the action or non-action taken for every File.

## Swap Space:

Swap is a portion of hard drive storage that has been set aside for the operating system to temporarily store data that it can no longer hold in RAM. This lets you increase the amount of information that your server can keep in its working memory, with some caveats. The swap space on the hard drive will be used mainly when there is no longer sufficient space in RAM to hold in-use application data. The information written to disk will be significantly slower than information kept in RAM, but the operating systems will prefer to keep running application data in memory and use swap for the older data. Overall, having swap space as a fallback for when your system's RAM is depleted can be a good safety net against out-of-memory exceptions on systems with non-SSD storage available.

## Steps to create Swap Space:

- 1) Checking the system for swap information: It is possible to have multiple swap files or swap partitions, but generally one should be enough. We can see if the system has any configured swap by typing:
  - a) `sudo swapon --show`
  - b) `free -h`
- 2) Checking Available Space on the Hard Drive Partition: Before we create our swap file, we'll check our current disk usage to make sure we have enough space. Do this by entering:
  - a) `df -h`

The device with / in the Mounted on column is our disk in this case. Although there are many opinions about the appropriate size of a swap space, it really depends on your personal preferences and your application requirements. Generally, an amount equal to or double the amount of RAM on your system is a good starting point. Another good rule of thumb is that anything over 4G of swap is probably unnecessary if you are just using it as a RAM fallback.

- 3) Create a Swap File: The best way of creating a swap file is with the `fallocate` program. This command instantly creates a file of the specified size:
  - a) `sudo fallocate -l 2G /swapfile2`
  - b) `ls -lh /swapfile2` → to verify file is created.
- 4) Change Permission: We need to lock down the permissions of the file so that only users with root privileges can read the contents. This prevents normal users from being able to access the file, which would have significant security implications. Command is:
  - a) `sudo chmod 600 /swapfile2`
- 5) Make the file Swap: Now mark the file as swap space. Command is:
  - a) `sudo mkswap /swapfile2`
- 6) Enable swap space: After marking the file as swap space file we can enable the swap file. Command is:
  - a) `sudo swapon swapfile2`
- 7) Check the swap space using
  - a) `sudo swapon --show`
  - b) `free -h`
- 8) Make swap file permanent: Our recent changes have enabled the swap file for the current session. However, if we reboot, the server will not retain the swap settings automatically. We can change this by adding the swap file to our `/etc/fstab` file. Command is:
  - a) `echo '/swapfile2 none swap sw 0 0' | sudo tee -a /etc/fstab`

## Deleting Swap Space

If we want to delete swap space, Do the following:

- 1) `sudo swapoff /swapfile2`
- 2) `free -h`
- 3) remove `swapfile2` → `sudo rm /swapfile2`
- 4) Delete the entry from `/etc/fstab` file.