

15 most used chown command in Linux [Cheat Sheet]

 golangcloud.com/chown-command-in-linux

August 31, 2021

Introduction to chown command in Linux

chown, which stands for **change owner**, is a command in Linux to change user or group ownership of a file, directory, or symbolic link. Every file or directory has a user or group ownership in the Linux systems. The Linux system can have multiple users and groups. They all have **unique** names and IDs. However, the name and ID of a user and a group can be the same.

Different examples to use chown command

chown command requires root permissions for its execution. You have to log in as a root user or use **sudo** in front of **chown** command.

Advertisement

To view the owner and group, you can use **ls -l** command. Here, test is the name of a file. The first home is owner and the second home is group.

```
$ ls -l test
-rw-rw-r-- 1 home home 42 Aug 29 21:58 test
```

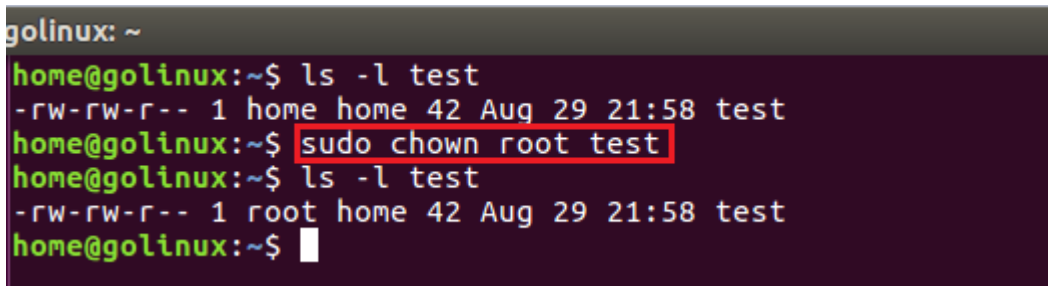
In this tutorial, you will learn the different ways to change owner and group of a file, directory, or symbolic link using **chown** command.

1. Change the owner of a file using chown command

This is a simple **chown** command that changes the owner/user of a file. It requires a new owner name and file name. The new owner must be a valid user.

```
$ sudo chown new_owner file_name
```

Sample Output:



```
golang: ~
home@golang:~$ ls -l test
-rw-rw-r-- 1 home home 42 Aug 29 21:58 test
home@golang:~$ sudo chown root test
home@golang:~$ ls -l test
-rw-rw-r-- 1 root home 42 Aug 29 21:58 test
home@golang:~$
```

2. Change the group of a file using chown command

chown command allows you to change the group of a file. You must use a colon in front of a new group name. Otherwise, it will be considered as a new owner.

```
$ sudo chown :new_group file_name
```

Sample Output:

```
golinux: ~/Folder
home@golinux:~/Folder$ ls -l test2
-rw-rw-r-- 1 elliot home 26 Aug 29 22:20 test2
home@golinux:~/Folder$ sudo chown :newgroup test2
home@golinux:~/Folder$ ls -l test2
-rw-rw-r-- 1 elliot newgroup 26 Aug 29 22:20 test2
home@golinux:~/Folder$
```

3. chown command to change the owner using user ID

`chown` command also changes the owner of a file with numeric user ID. But there should not be any other user having the same name as userID. You can use `id -u user_name` to print the user ID of a user.

```
$ sudo chown user_ID file_name
```

Sample Output:

```
golinux: ~
home@golinux:~$ id -u home
1000
home@golinux:~$ ls -l test
-rw-rw-r-- 1 root home 42 Aug 29 21:58 test
home@golinux:~$ sudo chown 1000 test
home@golinux:~$ ls -l test
-rw-rw-r-- 1 home home 42 Aug 29 21:58 test
home@golinux:~$
```

4. chown command to change the group using group ID

You can change the group of a file using group ID instead of group name. `id -g group_name` prints the group ID.

```
$ sudo chown :group_id file_name
```

Sample Output:

```
golinux: ~
home@golinux:~$ id -g root
0
home@golinux:~$ sudo chown :0 test
home@golinux:~$ ls -l test
-rw-rw-r-- 1 home root 42 Aug 29 21:58 test
home@golinux:~$
```

5. chown command to change owner of multiple files

This is a useful command that allows you to change owner of multiple files simultaneously. You have to separate file names with a space. It will assign same owner to all mentioned files.

```
$ sudo chown new_owner file_name1 file_name2 file_name3
```

Sample Output:

```
golinux: ~/Folder
home@golinux:~$ cd Folder
home@golinux:~/Folder$ ls -l
total 12
-rw-rw-r-- 1 home home 26 Aug 29 22:20 test2
-rw-rw-r-- 1 home home 25 Aug 29 22:20 test3
-rw-rw-r-- 1 home home 26 Aug 29 22:22 test4
home@golinux:~/Folder$ sudo chown root test2 test3 test4
home@golinux:~/Folder$ ls -l
total 12
-rw-rw-r-- 1 root home 26 Aug 29 22:20 test2
-rw-rw-r-- 1 root home 25 Aug 29 22:20 test3
-rw-rw-r-- 1 root home 26 Aug 29 22:22 test4
home@golinux:~/Folder$
```

6. Change owner and group name at the same time with chown command

chown command allows you to change owner and group name at the same time. You have to provide both new owner and group name in order to assign them in a file.

```
$ sudo chown new_owner:new_group file_name
```

Sample Output:

Advertisement

```
golinux: ~/Folder
home@golinux:~/Folder$ ls -l test3
-rw-rw-r-- 1 home home 25 Aug 29 22:20 test3
home@golinux:~/Folder$ sudo chown root:linux test3
home@golinux:~/Folder$ ls -l test3
-rw-rw-r-- 1 root linux 25 Aug 29 22:20 test3
home@golinux:~/Folder$
```

7. chown command to copy owner and group name from one file to another

This command copies the owner and group name from one file and assigns the same ownership to another file in the same directory.

```
$ sudo chown --reference=souce_file destination_file
```

Sample Output:

```
golinux: ~/Folder
home@golinux:~/Folder$ ls -l test2 test3
-rw-rw-r-- 1 elliot root 26 Aug 29 22:20 test2
-rw-rw-r-- 1 home   home 25 Aug 29 22:20 test3
home@golinux:~/Folder$ sudo chown --reference=test2 test3
home@golinux:~/Folder$ ls -l test2 test3
-rw-rw-r-- 1 elliot root 26 Aug 29 22:20 test2
-rw-rw-r-- 1 elliot root 25 Aug 29 22:20 test3
home@golinux:~/Folder$
```

8. Print the changes made by chown command

This command prints the changes made by `chown` command in detail. By default, `chown` command prints nothing.

```
$ sudo chown -v
```

Sample Output:

```
golinux: ~/Folder
home@golinux:~/Folder$ sudo chown -v elliot test2
changed ownership of 'test2' from root to elliot
home@golinux:~/Folder$ ls -l test2
-rw-rw-r-- 1 elliot home 26 Aug 29 22:20 test2
home@golinux:~/Folder$
```

It also prints the information when the owner or group name is **not changed**.

```
golinux: ~/Folder
home@golinux:~/Folder$ ls -l test2
-rw-rw-r-- 1 elliot home 26 Aug 29 22:20 test2
home@golinux:~/Folder$ sudo chown -v elliot:home test2
ownership of 'test2' retained as elliot:home
home@golinux:~/Folder$
```

9. chown command to print the detail only if changes are made

Unlike the previous command, it prints the information only if the changes are made in the owner or group name. It does not print anything if the changes are not made in the owner or group name.

```
$ sudo chown -c
```

Sample Output:

```
golinux: ~/Folder
home@golinux:~/Folder$ sudo chown -c home:linux test4
changed ownership of 'test4' from root:root to home:linux
home@golinux:~/Folder$ sudo chown -c home:linux test4
home@golinux:~/Folder$ prints nothing
```

10. chown command to change owner or group of a directory

Using the directory name instead of the file name, you can change the owner or group of a directory using `chown` command. You can also use **directory path**.

Advertisement

For changing the owner of a directory:

```
$ sudo chown new_owner directory_name
```

Sample Output:

```
golinux: ~
home@golinux:~$ ls -l
total 52
drwxr-xr-x 2 home home 4096 Jun 16 2020 Desktop
drwxr-xr-x 2 home home 4096 Aug 28 20:00 Documents
drwxr-xr-x 2 home home 4096 Jun 16 2020 Downloads
-rw-r--r-- 1 home home 8980 Jun 16 2020 examples.desktop
drwxrwxr-x 2 home home 4096 Aug 29 22:24 Folder
drwxr-xr-x 2 home home 4096 Jun 16 2020 Music
drwxr-xr-x 2 home home 4096 Jun 16 2020 Pictures
drwxr-xr-x 2 home home 4096 Jun 16 2020 Public
drwxr-xr-x 2 home home 4096 Jun 16 2020 Templates
-rw-rw-r-- 1 home root 42 Aug 29 21:58 test
drwxr-xr-x 2 home home 4096 Jun 16 2020 Videos
home@golinux:~$ sudo chown root Folder
home@golinux:~$ ls -l
total 52
drwxr-xr-x 2 home home 4096 Jun 16 2020 Desktop
drwxr-xr-x 2 home home 4096 Aug 28 20:00 Documents
drwxr-xr-x 2 home home 4096 Jun 16 2020 Downloads
-rw-r--r-- 1 home home 8980 Jun 16 2020 examples.desktop
drwxrwxr-x 2 root home 4096 Aug 29 22:24 Folder
drwxr-xr-x 2 home home 4096 Jun 16 2020 Music
drwxr-xr-x 2 home home 4096 Jun 16 2020 Pictures
drwxr-xr-x 2 home home 4096 Jun 16 2020 Public
drwxr-xr-x 2 home home 4096 Jun 16 2020 Templates
-rw-rw-r-- 1 home root 42 Aug 29 21:58 test
drwxr-xr-x 2 home home 4096 Jun 16 2020 Videos
home@golinux:~$
```

For changing the group of a directory:

```
$ sudo chown :new_group directory_name
```

Sample Output:

```

golinux: ~
home@golinux:~$ ls -l
total 52
drwxr-xr-x 2 home home 4096 Jun 16 2020 Desktop
drwxr-xr-x 2 home home 4096 Aug 28 20:00 Documents
drwxr-xr-x 2 home home 4096 Jun 16 2020 Downloads
-rw-r--r-- 1 home home 8980 Jun 16 2020 examples.desktop
drwxrwxr-x 2 root home 4096 Aug 29 22:24 Folder
drwxr-xr-x 2 home home 4096 Jun 16 2020 Music
drwxr-xr-x 2 home home 4096 Jun 16 2020 Pictures
drwxr-xr-x 2 home home 4096 Jun 16 2020 Public
drwxr-xr-x 2 home home 4096 Jun 16 2020 Templates
-rw-rw-r-- 1 home root 42 Aug 29 21:58 test
drwxr-xr-x 2 home home 4096 Jun 16 2020 Videos
home@golinux:~$ sudo chown :root Folder
home@golinux:~$ ls -l
total 52
drwxr-xr-x 2 home home 4096 Jun 16 2020 Desktop
drwxr-xr-x 2 home home 4096 Aug 28 20:00 Documents
drwxr-xr-x 2 home home 4096 Jun 16 2020 Downloads
-rw-r--r-- 1 home home 8980 Jun 16 2020 examples.desktop
drwxrwxr-x 2 root root 4096 Aug 29 22:24 Folder
drwxr-xr-x 2 home home 4096 Jun 16 2020 Music
drwxr-xr-x 2 home home 4096 Jun 16 2020 Pictures
drwxr-xr-x 2 home home 4096 Jun 16 2020 Public
drwxr-xr-x 2 home home 4096 Jun 16 2020 Templates
-rw-rw-r-- 1 home root 42 Aug 29 21:58 test
drwxr-xr-x 2 home home 4096 Jun 16 2020 Videos
home@golinux:~$

```

11. Change owner of a file using chown command if only the current owner matches

This command assigns the new owner to a file if only its current owner matches. `--from` option verifies the current owner.

```
$ sudo chown --from=current_owner new_owner file_name
```

Sample Output:

```

golinux: ~/Folder
home@golinux:~/Folder$ ls -l test4
-rw-rw-r-- 1 root home 26 Aug 29 22:22 test4
home@golinux:~/Folder$ sudo chown --from=root elliot test4
home@golinux:~/Folder$ ls -l test4
-rw-rw-r-- 1 elliot home 26 Aug 29 22:22 test4
home@golinux:~/Folder$

```

If the current owner does not match, then the file owner will remain unchanged.

Sample Output:

```
golinux: ~/Folder
home@golinux:~/Folder$ ls -l test4
-rw-rw-r-- 1 elliot home 26 Aug 29 22:22 test4
home@golinux:~/Folder$ sudo chown --from=root home test4
home@golinux:~/Folder$ ls -l test4
-rw-rw-r-- 1 elliot home 26 Aug 29 22:22 test4
home@golinux:~/Folder$
```

12. chown command to change group of a file only if the current group matches

Similar to the previous command, it only allows you to change the group of a file if the current group matches. `--from` option also verifies the current group of a file.

```
$ sudo chown --from=:current_group :new_group file_name
```

Sample Output:

```
golinux: ~/Folder
home@golinux:~/Folder$ ls -l test3
-rw-rw-r-- 1 home newgroup 25 Aug 29 22:20 test3
home@golinux:~/Folder$ sudo chown --from=:newgroup :home test3
home@golinux:~/Folder$ ls -l test3
-rw-rw-r-- 1 home home 25 Aug 29 22:20 test3
home@golinux:~/Folder$
```

You can also use `--from` option to verify the current owner and group and assign a new owner and group at the same time.

```
$ sudo chown --from current_owner:current_group new_owner:new_group file_name
```

Sample Output:

```
golinux: ~/Folder
home@golinux:~/Folder$ ls -l test2
-rw-rw-r-- 1 elliot root 26 Aug 29 22:20 test2
home@golinux:~/Folder$ sudo chown --from=elliot:root home:linux test2
home@golinux:~/Folder$ ls -l test2
-rw-rw-r-- 1 home linux 26 Aug 29 22:20 test2
home@golinux:~/Folder$
```

13. Change the owner or group of files using chown command

`-R` option allows you to change the owner or group of files and sub-directories recursively present in the same directory. It is very useful when there are large number of files that need same ownership.

```
sudo chown -R new_owner:new_group directory_name
```

Sample Output:

```
golinux: ~/Folder
home@golinux:~$ sudo chown -R root:root Folder
home@golinux:~$ cd Folder
home@golinux:~/Folder$ ls -l
total 12
-rw-rw-r-- 1 root root 26 Aug 29 22:20 test2
-rw-rw-r-- 1 root root 25 Aug 29 22:20 test3
-rw-rw-r-- 1 root root 26 Aug 29 22:22 test4
home@golinux:~/Folder$
```

14. chown command to change ownership of a symbolic link file

A symbolic link file is a type of file that points to another file. You can create a symbolic link by using:

```
$ ln -s source_file sym_file
```

To change the ownership of a symbolic link file using `chown` command, you have to use `-h` option to apply the changes. Otherwise, the changes will apply to the main file.

```
$ sudo -h new_owner:new_group sym_file
```

Sample Output:

```
golinux: ~
home@golinux:~$ ln -s test symtest
home@golinux:~$ ls -l test symtest
lrwxrwxrwx 1 home home 4 Aug 30 16:22 symtest -> test
-rw-rw-r-- 1 home home 42 Aug 29 21:58 test
home@golinux:~$ sudo chown -h elliot:linux symtest
home@golinux:~$ ls -l test symtest
lrwxrwxrwx 1 elliot linux 4 Aug 30 16:22 symtest -> test
-rw-rw-r-- 1 home home 42 Aug 29 21:58 test
home@golinux:~$
```

Without using `-h` option:

```
golinux: ~
home@golinux:~$ ls -l test symtest
lrwxrwxrwx 1 elliot linux 4 Aug 30 16:22 symtest -> test
-rw-rw-r-- 1 home home 42 Aug 29 21:58 test
home@golinux:~$ sudo chown root:root symtest
home@golinux:~$ ls -l test symtest
lrwxrwxrwx 1 elliot linux 4 Aug 30 16:22 symtest -> test
-rw-rw-r-- 1 root root 42 Aug 29 21:58 test
home@golinux:~$
```

As you can see in the above output, the ownership of 'test' is changed instead of 'symtest'.

15. Hide error message in the output with chown command

This command hides the error message in the output when the file or directory is not found.

```
$ sudo chown -f new_owner file_name
```

Sample Output:

```
golinux: ~/Folder
home@golinux:~/Folder$ sudo chown home test5
chown: cannot access 'test5': No such file or directory
home@golinux:~/Folder$ sudo chown -f home test5
home@golinux:~/Folder$ error not shown
```

It does not hide the error message printed due to **invalid user, group, or syntax**.

```
golinux: ~/Folder
home@golinux:~/Folder$ sudo chown -f happy test5
chown: invalid user: 'happy'
home@golinux:~/Folder$ sudo chown -f :friday test5
chown: invalid group: ':friday'
home@golinux:~/Folder$ sudo chown -f home
chown: missing operand after 'home'
Try 'chown --help' for more information.
home@golinux:~/Folder$
```

Conclusion

These are the most important `chown` commands in Linux. Now, you can easily change the user and group ownership of a file, directory, or symbolic link in the Linux systems.

What's Next

You can explore [20 most used grep command examples in Linux](#)

Further Reading

[man page for chown command](#)