

chown example



```
# chown user:group file
```

One of the most useful and powerful [basic Unix commands](#), [chown command](#) allows you to change ownership of specified files and directories – change user or group owner.

chown Must be Run as root

[chown](#) is one of these commands that must be run as root. Running it as a regular user will not work: one user can't change even its own files so that they belong to another user.

Basic chown Example

I'll start in my home directory `/home/greys`. Let's use [touch command](#) to create a file named `try` and then use [sudo command](#) to become root:

```
[greys@rhel8 ~]$ touch try
[greys@rhel8 ~]$ sudo -i
[sudo] password for greys:
[root@rhel8 ~]# cd /home/greys
```

```
[root@rhel8 /home/greys]# ls -ald try
-rw-rw-r--. 1 greys greys 0 Feb 20 06:44 try
```

As you can see, the file rightfully belongs to me and my group: **greys:greys**.

Let's change the owner and owner group to root:

```
[root@rhel8 /home/greys]# chown root:root try
[root@rhel8 /home/greys]# ls -al try
-rw-r--r--. 1 root root 0 Feb 20 06:44 try
```

chown with Verbose Reporting

If we use the `-v` command line option for [chown](#), it will confirm every action:

```
[root@rhel8 /home/greys]# chown -v greys:greys try
changed ownership of 'try' from root:root to greys:greys
```

chown Using Reference File

A really cool way of using [chown](#) and also a great gateway to shell scripting is making `chown` inspect a given (reference) file and then apply its ownership information to other specified files. So you're making [chown command](#) confirm owner and group of a file and then apply this to lots of other files – all without really knowing or specifying the actual ownership info. That's while such a file is called reference file.

For instance, look at the **chrony** config files in **/etc** directory. See how **/etc/chrony.keys** file belongs to **root:chrony**?

```
[root@rhel8 /home/greys]# ls -al /etc/chrony.*
-rw-r--r--. 1 root root 1083 Apr 4 2018 /etc/chrony.conf
-rw-r-----. 1 root chrony 481 Apr 4 2018 /etc/chrony.keys
```

Here's how you can make [chown](#) apply the same ownership details

to my /home/greys/try file:

```
[root@rhel8 /home/greys]# chown --reference=/etc/chrony.keys  
try
```

```
[root@rhel8 /home/greys]# ls -la try  
-rw-r--r--. 1 root chrony 0 Feb 20 06:44 try
```

Pretty cool, huh?

See Also

- [Different between chmod and chown](#)
- [Basic Unix commands](#)
- [How To: confirm file ownership details in Linux](#)
- [How To: change ownership of files and directories in Unix](#)