chown example: recursive update

chown command

chown is a basic Unix command, super useful and very powerful. I have provided a **chown example** earlier, but would like to show another common way it’s used.

chown: update ownership recursively

chown command uses -R option to apply changes recursively.

For example, if I have a **/Users/greys/unixtutorial** directory with the following layout:

```
file1
file2
dir1/file3
```

… then it’s very easy to show the difference between chown and chown -R commands.

Let’s go to the **/Users/greys/unixtutorial** directory:

```
greys@maverick:~ $ cd /Users/greys/unixtutorial
greys@maverick:~/unixtutorial $
```

Now, let’s look at the files with the **ls command**:

```
greys@maverick:~/unixtutorial $ ls -al *
 -rw-r--r--  1 greys  staff  0  8 Oct 22:55 file1
 -rw-r--r--  1 greys  staff  0  8 Oct 22:55 file2
dir1:
 total 0
 drwxr-xr-x  3 greys  staff   96  8 Oct 22:55 .
 drwxr-xr-x  5 greys  staff  160  8 Oct 22:55 ..
 -rw-r--r--  1 greys  staff    0  8 Oct 22:55 file3
```

Everything belongs to my own user, **greys** and my primary group: **staff**.

Time to change ownership of everything in the current directory:

```
greys@maverick:~/unixtutorial $ sudo chown root:wheel *
```

Checking files with ls again, I can see that immediate contents of the /Users/greys/unixtutorial directory (where I was at the time of running chown) got updated ownership: file1 and file2, along with dir1, now belong to root:wheel.

But **file3** was in a **dir1** subdirectory, so it stayed intact and still belongs to greys:staff:

```
greys@maverick:~/unixtutorial $ ls -al *

-rw-r--r--  1 root  wheel  0  8 Oct 22:55 file1

-rw-r--r--  1 root  wheel  0  8 Oct 22:55 file2
dir1:
 total 0
 drwxr-xr-x  3 root   wheel   96  8 Oct 22:55 .
 drwxr-xr-x  5 root   wheel  160  8 Oct 22:55 ..
 -rw-r--r--  1 greys  staff    0  8 Oct 22:55 file3
```

That's because without the -R (recursive) option, chown will only inspect and update files in the current directory, but not in any of its subdirectories (or their subdirectories, and so on).

Running the same command with -R option does the trick:

```
greys@maverick:~/unixtutorial $ sudo chown -R root:wheel *
greys@maverick:~/unixtutorial $ ls -al *
-rw-r--r--  1 root  wheel  0  8 Oct 22:55 file1
-rw-r--r--  1 root  wheel  0  8 Oct 22:55 file2
dir1:
total 0
drwxr-xr-x  3 root  wheel   96  8 Oct 22:55 .
drwxr-xr-x  5 root  wheel  160  8 Oct 22:55 ..
```

```
-rw-r--r--  1 root  wheel    0  8 Oct 22:55 file3
```

Hope you learned something new, come back for more!

See Also

- **Unix Basic Commands**
- **ls**
- chown command
- chmod vs chown
- Running chown with sudo