8.8.2 Managing Ownership

Click one of the buttons to take you to that part of the video.

Manage Ownership 0:00-0:09

In this demonstration, we're going to talk about managing ownership of files and directories in the Linux file system.

Manage New File Ownership 0:10-2:21

Be aware that by default, whenever a user creates a new file or creates a new folder, that user automatically becomes its owner. Also, the default group that's been assigned to that user automatically becomes the owning group of that file or folder.

Let's see how that works. I'm currently logged in as my rtracy user and I'm in my home directory. I'm going to just create a new file using the touch command.

Enter 'touch newfile'. All this does is create an empty blank file in the file system. Press Enter.

Now let's run the 'ls -l' command. When we do, we see the newfile file that I just created. It's 0 bytes, but here's the important thing I wanted you to see.

This first field here in the output of the ls -l command specifies the name of the user who owns this file. In this case, it is the rtracy user. This second field right here specifies the name of the group that owns this file, which is again rtracy.

This is a point of confusion for many new Linux administrators. Understand that when you create a new user account in some Linux distributions, such as the Fedora system that we're running here, it will automatically create a group with the same name as the user and assign that group to be that user's primary group.

In this case, when I created the rtracy user account on this system, a corresponding group named rtracy was also created and was assigned to the rtracy user as rtracy's primary group.

Hence, the owning user is rtracy for this file and because the rtracy group is the rtracy user's primary group, this group is also the owning group of that file.

That's not the case with all distributions. Some distributions will create one single group named Users and whenever you create a new user account in the system, that user is automatically made a member of the Users group, and Users becomes the primary group.

You have to check with your particular distribution to see which way it does this. More and more distributions are starting to do things this way; however, where we have a group with the same name as the user by default.

Change Owner of a File 2:22-5:11

Even though this user and this group was assigned to newfile when we created it, we're not stuck with that; you can change ownership. However, there are a few caveats that you have to be aware of.

First of all, only the root user can change which user owns a particular file or directory in the file system. No one else can.

However, if we want to change the owning group, we actually have two options. Root, of course, can change which group owns a particular file or directory.

Also, if you are the owning user--in this case, I'm logged in as rtracy and rtracy is the owning user--then that user can actually change the owning group. It can't change the user itself that owns the file, but it can change the owning group.

Let's go ahead and try doing that. Let's change the ownership of newfile.

To do this, I want to switch to my root user account first. What I'm going to do is actually copy this file from my user account's home directory to another user account's home directory. I'm going to do 'cp /home/rtracy/newfile' and let's copy it to the home directory of the ksanders user.

Let's go ahead and switch to that directory now. We'll run the 'ls -l' command and we see the file that was copied over, but notice something very important happened during the copy process. Note who is the owning user and owning group of this copy of the original file. It's been

changed to root.

The reason this happened is because when root executed the copy command, a new copy of that file was created in the new location in the file system. And who created that file with the copy command? It was root. Therefore, root becomes the owning user and root's default group, which is root, became the owning group.

Because this file is owned by root, the ksanders user will have very limited access to this file even though it exists in ksanders' home directory. Let's fix this. Let's change ownership of this copy of the file to the ksanders user so that she can access it in her home directory.

To do this, we run the 'chown' command for change owner. We then specify the name of the user that we want to change ownership to--ksanders--and then we specify the name of the file that we're modifying. In this case, it's 'newfile'. Press Enter.

Let's run the 'ls -l' command again, and when we do, we see that the user that's assigned as the owner of the file has been changed from root, up here, to ksanders. However, notice over here that the group that owns the file was not changed. The root group is still the owning group of newfile.

Change the Owning Group 5:12-7:14

More than likely, we're going to want to change the owning group for this file as well.

That can actually be done in two different ways. You can do it with the chown command. Enter 'chown' again and then we specify the name of the group that we want to change the ownership to for that file.

However, we have to specify that this is now a group and not a user that we're changing ownership for, so we enter a period first and then the name of the group, '.ksanders'.

This is very important, because remember the name of the user and the name of that user's primary group is the same. By putting a period before ksanders, we're telling chown that this is a group name, not a username.

Then we add a space and then enter the name of the file, 'newfile'. Now let's run 'ls -l' again and we see that ksanders is now the user who owns that file and the ksanders group also owns that file.

There's actually a second command you can use to change the group ownership of a file. It's called chgrp. As its name implies, that stands for change group and its syntax is the same as chown. We enter 'chgrp' and then the name of the group that we want to change ownership to, 'ksanders', and then the name of the file, 'newfile'.

Notice here that when I use chgrp, I don't have to put the period before the group name. That's because chgroup can change only group ownership. Therefore, there's no need to differentiate it from a user account.

Press Enter, and it changes the group ownership, which actually didn't really change, because it was already set to ksanders.

Before we end this demo, there's one other thing I want to show you and that is the fact that you can use chown to actually change both user and group ownership of a file at the same time.

We do this by first specifying the name of the user who we want to own the file or directory, a period, and then the name of the group. That would be ksanders again in this case and then the name of the file. When we do that, both the user and the group ownership is changed at the same time.

Summary 7:15-7:21

That's it for this demonstration. In this demo, we talked about how to manage ownership. We first talked about how to change the owner that owns a file and then we talked about how to change the owning group.
