2.6.2 View Shell Configuration Files

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

Viewing Shell Configuration Files 0:00-0:39

In this demonstration, we're going to talk about managing our bash shell configuration files. Currently, we are in my user's /home directory. We can tell that by the tilde (~). If I do a pwd command, we see that the /home directory is actually /home/rtracy. If I run an ls command, we see a bunch of different files and directories within my /home directory. But notice that none of my bash shell configuration files are shown here. The reason for this is because your bash shell configuration files are hidden files. We have to use the ls command with the -a option. Now we see a whole bunch of hidden files within my /home directory.

.bashrc File 0:40-1:25

The three shell configuration files that we want to pay attention to here, in this demo, are the .bashrc file, the .bash_profile, and the .bash_logout file. Now, remember, any file in the Linux file system that begins with a period is a hidden file. And you have to include that period whenever you manipulate that file in a text editor or in a text viewing command because that is a part of the filename, and if you exclude it, the shell won't know what file you're talking about. The .bashrc file is used whenever you open a non-login shell, such as the shell session that I've currently opened up on this session. Because I was already authenticated to the system, I didn't have to log in to my shell session. I didn't have to authenticate before I could use this shell session. That's why it's a non-login shell.

.bash_profile Configuration File 1:26-2:10

As we'll talk about in just a minute, this configuration file is also used whenever you open up a login shell session as well. We'll talk about how that works in just a second. In addition, we have the .bash_profile configuration file. This file configures the way the shell will work whenever you launch a login shell on initial login. Now, you might be thinking, "Wait a minute. I boot my Linux system into a graphical environment, and I log in using the graphical user interface. I don't open up a shell session and log in like you would do on a server system that doesn't use a graphical environment." Well, it's important to note that this file, here, this configuration file, is used even if you were to log in to the Linux system using a graphical login. It's not limited to just a server system with no graphical user interface.

.bash_logout File 2:11-2:42

And, finally, we have the .bash_logout file down here, which is used whenever you log out from a shell session. Before we go on, it's important to note that these three files, .bashrc, .bash_profile, and .bash_logout, are not actually configuration files in the typical sense, like a configuration file you might see in the /etc directory. Instead, these files are actually script files, and they are run whenever a particular action occurs on the system--such as opening up a new shell session, or logging out, and so on.

Look at .bashrc 2:43-3:40

So, with that, let's take a look at the .bashrc file. To do this, I'll use the less command, and we'll take a look at .bashrc. As you can see, this is actually a script. There is even a control structure in here, an if control structure. This if statement tells the shell to go out and look and see if this file right here exists. If /etc/bashrc exists, that's what the -f parameter does. It's a test. It says, "Let's go out and see if this file exists. And if it does, then let's run it." 'run /etc/bashrc'. This version of the bashrc file, notice, does not reside in the /home directory. Instead, it resides in the /etc directory, and that's because it contains global configuration parameters that are applied to all user accounts. The .bashrc file we're working with here is stored in my /home directory and is only applied to my user account. I can make changes to it, and it will not affect any other user on the system.

Modifications to the Shell Environment 3:41-4:43

As a best practice, if you need to make modifications to your shell environment, you should make your customizations here, in this file in your home directory, not in the /etc/bashrc file, because, as we indicated earlier, any changes you make here will be applied to all the users in the system, not just your account, and that might tick somebody off. So, we have our first statement right here, that tests for the existence of this file and runs it if it's there. At the end of the file, we have an option down here to add user specific aliases and functions. You're not just limited to aliases and functions. You could do a variety of different tasks in this file, such as setting the value of an environment variable, and so on. You just add the commands to the end of the file. Notice, here, that I have put in a user-defined alias. I say alias lll="ls -al" command. And because of this, whenever I open up a shell session on the desktop, this alias will be defined every single time, and I can use this alias to run this command.

Look at .bash_profile 4:44-6:41

I'll press q to exit less. Now we want to look at the .bash_profile file. Essentially, this script is run whenever you log into the system, regardless of whether you're logging in from the shell prompt or if you're logging in from the graphical user interface. There's something very important here that I want to point out. Remember, earlier, we said that bashrc configuration file in your /home directory is used whenever you create a non-login shell, such as opening a terminal session like the one we have here. But we said it's also run whenever you initiate a login shell. Here's why. It's because of this file right here: .bash_profile. Notice that we have an if statement like we did before, but it looks for a different file. It's testing for the existence of the .bashrc file in the user's /home directory, and if it's there, it is run. Therefore, the .bashrc file in your /home directory is run whenever you initiate a non-login shell, or whenever you initiate a login shell because of this statement, right here.

Also, notice down here that the PATH environment variable is customized when this script is run. Notice, here, that it takes the current value of the path and retains it to whatever the PATH environment variable that was probably set in /etc/bashrc is. It will keep it here, and then it's going to concatenate to additional variables to the PATH environment variable. They're separated by these two colons, right here and right here. The first one says whatever the value of the /home directory is, $HOME points to the HOME environment variable. Let's add a directory to the path called .local (so it's a hidden directory) /bin. Basically, this allows us to put executable files in the hidden .local/bin directory in our /home directory and have them run without having to specify the full path to the file.

---

bin 6:42-8:06

In addition, we can have another directory in our /home directory that's not hidden called bin, and you can drop executables in it, as well, and run them from there. This statement adds two directories in our /home directory to the PATH environment variable, and then it's exported, so it will be persistent between shell sessions. By adding these commands here, our shell environment is automatically configured for us every time we log in to the system.

Look at .bash_logout

The last shell configuration file we need to look at here is the .bash_logout file. As we talked about earlier, this script is run whenever you close a shell session, whenever you log out, by either typing 'logout' or 'exit', depending on which shell environment you're in. When you run those commands, this script is executed. Notice that there is nothing in here by default. It's a blank, empty file. But you could add commands in here to customize how it works. For example, you may want to reset the value of the environment variable when you log out. Maybe we customized the PATH environment variable in one of the other shell configuration files, and we want to reset it back to its original value when we log out. Or maybe we have a temporary directory in our home directory that we put temporary files in, and we don't want that directory to get clogged up with a bunch of junk files that we don't need, so we can add commands here that would go in and clear out the contents of that temporary directory whenever we logged out so it would be nice and fresh when we logged back in the next time.

Summary 8:07-8:21

That's it for this demonstration. In this demo we talked about the shell configuration files that reside within your home directory: .bashrc, .bash_profile, and .bash_logout. And we also pointed out the global bash configuration file located in the /etc directory named bashrc.
