2.7.4 Use Piping

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

Using Piping 0:00-0:14

In this demonstration, we're going to look at piping with shell commands. This can be a very useful thing. You can use piping to send the output of one command to the input of another command for processing.

Single Pipe 0:15-2:35

Let's begin by switching to our root user account. Let's use the cat command alone just to view the contents of a very large file, /var/log/boot.log. This is a log file that's created as the system boots up. Each boot message that's generated during the boot-up process is written to this file so you can review it later; it can be a very useful tool for troubleshooting. It's also a very large file. If I press Enter and then start scrolling, you can see that this file goes on, and on, and on. It's a big file.

If we wanted to make thing a little bit more user-friendly, we could actually pause the output of the cat command one page at a time. To do this, we would take the output from the cat command and pipe it to a different command that would pause the output. In this case, it would be the more command. To do this, we would use the same command that we started with before, 'cat /var/log/boot.log', and then we add a pipe character (|). The pipe character tells the shell to take whatever output comes from this command and send it to the input of whatever command we specify here, in this case, the more command. cat will generate the same output that we saw here. But instead of writing it to the screen, it's going to redirect it to the input of the more command, and then the more command will process it. As you can see here, it pauses it one page at a time on the screen. Then I can press the spacebar or use the arrow keys to work my way through the file instead of having it all just fly by really fast.

Piping is commonly used in conjunction with the grep command to search for information within the output of another command. For example, we can take the cat command, and we could use it to write the information from boot.log, like we did before, pipe it to the input of grep, and then let grep search for specific text.

For example, let's say we want to verify that the firewall started correctly when this system was booted. Let's run the cat command again to view the boot.log file, and let's pipe it again. But this time, let's pipe it to grep, and then let's tell the grep command what text we want to search for. In this case, we want to search for the text 'firewalld'. These will be boot messages generated when the firewall daemon on the system was stared up. We'll hit Enter, and we can see here that the firewall was, indeed, started automatically when the system was booted up.

Create a Text Stream 2:36-3:44

You can also chain multiple pipes together, meaning we can take the output of one command, send it to the input of another command, and then take the output of the second command and send it on to a third command, and so on, as many times as we need to get the information that we need. This is called creating a text stream. For example, we could take the command that we worked with before, and let's change it a little bit. Let's use cat to display the contents of boot.log. Let's pipe it to grep, and then let's tell grep to find all instances of the word 'target'. Just like with firewalld, this will cause grep to just pick out the lines that contain the text we specify. This time, instead of firewalld, it's going to pick out all the lines of text in the output of the cat command that contain the word target. There might be a lot of them, and so we want to pause the output a page at a time. We'll pipe it again to the more command. We've made a chain.

We're going to take the output of cat and send it to the input of grep. We'll take the output of grep and send it to the input of more. Hit Enter, and the output is displayed. We can work our way through it using the spacebar and the more command.

Summary 3:45-3:57

That's it for this demonstration. In this demo, we talked about how to use piping at the shell prompt. We first talked about how to create a single pipe with an input and an output. Then we talked about how to chain multiple pipes together to create a text stream.