10.1.7 Viewing Process Information with pgrep

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

View Process Information with pgrep 0:00-0:45

In this demonstration we're going to talk about using the pgrep command. The ps command is very commonly used to view process information. It's very useful, it's very flexible, it's very powerful.

However, sometimes the output of the ps command can be overwhelming. If we were to type 'ps -elf' for example, there's a ton of information that's displayed. Sometimes there's too much information being displayed.

This is true especially if you're looking for just one or two specific processes. One option for managing the excessive output of the ps command is to pipe (|) the output from the ps command to the grep command to search for just the information that you need.

ps Output to the grep Command 0:34-1:07

For example, let's specify that we want to find any line that has the term 'rtracy' in it. And here you can see a list of all the processes that are owned by the rtracy user. That's one option for trimming down the amount of information that you get from the ps command.

Another option is to use the pgrep command.

pgrep Command 1:04-2:43

As its name implies, the pgrep command basically combines the functionality of the ps command and the grep command into one single utility, so you don't have to pipe anything.

When you run pgrep, you specify certain selection criteria that you want the command to look for. The command then searches through all the current running processes, and then just outputs a list of only those processes that match the criteria that you specify.

Let's take a look at the 'man' page for 'pgrep'. There are lots of different options that you can use, but there are a couple in particular that I think are very useful. For example, you can use the -f option to search for a specific process name.

Another option is -p, which allows you to search based on the parent process ID of the process. Another useful one is the -u option, which allows you to search based on the user that owns the process.

I also like the -l option right here, because by default, when you run pgrep, it will output only the process ID numbers of the processes that match the search criteria that you specify. If you use the -l option, it will display not only the process ID number, but also the name of the process as well. Makes it a lot easier to work with.

Let's 'exit' out of the man page, and let's do an example. Let's do the same thing that we did before with ps and grep, but this time with pgrep. Run 'pgrep' and we'll specify long output so we can see the process ID number and the process name.

Then let's specify that we want to see all the processes that are owned by the 'rtracy' user. When we do, a list of all the processes that match the search criteria that we specified is displayed.

Summary 2:44-2:55

That's it for this demonstration. In this demonstration we looked at using the pgrep command. We began this demonstration by looking at ways you can pipe the output from the ps command to the grep command, and then we talked about how you can accomplish the same thing by using the pgrep command.