10.1.6 Viewing Process Information with ps

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

View Process Information with ps 0:00-0:21

In this demonstration, we're going to practice using the ps utility to view information about processes currently running on the Linux system. Before running ps, I'm going to switch to my root user account. ps can be run in many different ways to display many different types of information.

---

ps without Options 0:21-2:28

The most basic way to use ps is to just simply type 'ps' itself with no options at the command prompt. I'll hit Enter and when I do, a list of all the processes currently associated with the current shell session are displayed, and that's very important to remember.

Without any options, all you're seeing are the processes associated with the current shell. Which you can see here.

First of all, we have the su utility, which I ran to elevate to root level privileges. We see bash here, which is the current bash shell session that I'm running in.

Then we also see the ps command itself, which we ran in order to generate the output that we see here. Because the ps command was run within the current shell session in order to generate the list of processes, its process had to be listed as well.

There's one thing that I want to emphasize here about ps. Unlike the top utility, which displays process information dynamically--it's constantly being updated--ps does not.

Instead, its output is static. It basically just displays a snapshot of the current processes running at the time the command was run. This could change. As you can see, you run the ps command, it writes its output out to the screen, and then it exits.

Anything that's changed since ps was run is not reflected up here in the output. If you need to view a dynamic display of processes and how they're changing all the time, then you should use the top utility instead.

When we run ps with no options, we see four columns of information. The PID column displays the ID number of each process. The TTY column displays the name of the terminal session, or the shell, that the process is running within.

The TIME column displays the amount of CPU time that's being used by the process. As we talked about a minute ago, the COMMAND column displays the name of the command that was entered, in order to create the process.

As we mentioned a minute ago, only three processes are listed here. This system actually has many, many processes actually running. Why didn't they show up in this list? It's because ps by default shows only the processes associated with this current shell session.

---

ps Options 2:28-3:34

To see all of the processes running on the system, you need to use the -e option with the ps command. Let's try entering that here: 'ps -e'. As you can see, the amount of output is dramatically different when we use the -e option with ps.

Instead of just displaying the processes associated with the current shell session, all of the processes running on the system were displayed. Although the number of columns is still the same, we still see the PID, TTY, TIME, and COMMAND columns like we saw with just the plain old ps command.

However, I want to point out an important difference here under TTY. Notice that almost all of these processes are enlisted with a question mark (?) in the TTY column. This is important, because it indicates that the associated process is a system process.

Remember that system processes--our daemons-- are loaded by the systemd process when the system initially boots. Because of this, they're not associated with any particular shell session. Because they're not associated with any particular shell session, we use a question mark (?) in the TTY column in the output of the ps command.

More Details 3:34-8:09

If you've ever used the top command to view information about processes, you'll note that top displays a lot more information about each process than the ps command does here. However, you can tell the ps command to display more details about each process running on the system.

To do that, we add the -f option to the command. As you can see here, you can mix the options together. In this case, I'm going to use '-e' first to display all the processes running on the system, and I'm going to use the 'f' option to display extended information about each of those processes.

I'll press Enter and now we see a lot more information about each process. Let's scroll to the top of the output here. Notice now we see the UID column has been added to the output. This displays the user ID of the process's owner. As you can see, most of the system processes are owned by root.

We also added the PPID column. PPID stands for Parent Process ID. This is the ID number that's assigned to the parent process of each of these processes. We also added the C column right here. The C column specifies the amount of processor time utilized by that process. We also added the STIME column, which specifies the time that the process originally started.

If you want to really crank up the amount of information that you see, you can add the 'l' option to the ps command. The -l option displays the long format of the ps output.

Let's go ahead and run it, and as you can see, even more information is displayed about each process. For example, we've added an F column right here. F specifies the flags that are associated with the process. For example, you can have a 1 in this column, which indicates the process forked but didn't execute; or you can have a 4, which indicates that root privileges were used for this process.

In addition, we also have a column labeled S added to the output. This identifies the state of the process. This is very useful information.

If there's a D in this column, it indicates that the process is uninterruptedly sleeping. An R in this column indicates that the process is running. An S in this column, which you see a lot of here, indicates that the process is interruptedly sleeping, versus uninterruptedly sleeping. A T in this column indicates that the process is stopped or traced. And a Z in this column indicates that we are dealing with a zombie process.

We also have the PRI column. This specifies the priority of the process. We have the NI column, which identifies the nice value assigned to the process. We have the ADDR column, which specifies the memory address of the process. We have SZ column, which specifies the size of the process.

We have the WCHAN column, which specifies the name of the kernel function in which the process is currently sleeping. If the process is not sleeping and is actually running, you won't see anything in this column; you'll just see a dash.

There are many other options that you can use with the ps command. If you want to see them, load the man page for ps. For example, you can use the -a option to select all processes, just like the -e option that we looked at earlier. You can use the -r option to restrict the selection to only processes that are currently running, and so on.

Based upon what we saw in the man page, one common way to use ps is to run ps aux. This will display all of the processes running on the system.

Before we actually execute that command, I want to load a problematic process into memory. I'm going to go ahead and open up a new session over here. Open terminal, and in my rtracy user's /home directory, I have a executable file called zombie. I compiled this little program, and all it does is load a process into memory and zombie it. Let's go ahead and run it.

The process has been loaded into memory, and it has been zombied. We can use the 'ps aux' command to identify that zombie process. Press Enter.

And notice down here in the output of the command we see a Z+. Scroll up so you can see the column header here. In the stack column, which is just the status of the process because it's Z, we know that that process is a zombie process. We going to have to do some cleanup to try to figure out what caused it and what process we need to kill in order to clean it up.

Summary 8:09-8:23

That's it for this demonstration. In this demo we talked about how to use the ps command to view process information. We first reviewed all of the different options you can use with the ps command to identify specific information. Then we used the ps command to identify a zombie process.
