10.2.3 Process Prioritization

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

Process Prioritization 0:00-1:37

In this lesson, we're going to talk about how you prioritize processes that are running on a Linux System. Remember, Linux is a multitasking operating system. Therefore, it rotates CPU time between each of the processes that are currently running on the system, creating an illusion, really, that all these processes are running concurrently.

Because Linux is a multitasking operating system, you can specify a priority level for each process running on a system. And doing this determines how much CPU time a particular process is going to get in relation to the other processes that are running on the system.

By default, Linux really doesn't do a whole lot of prioritization. It actually tries to equalize the amount of CPU time given to all the processes running on the system, with a very limited number of exceptions. However, there may be times when you need to adjust the priority assigned to a particular process.

Depending upon how the system is deployed, you may want one particular process to have a higher priority than the other processes running on the system. Maybe the system is being used for a web server. In which case, you'd want the web server process to have a higher priority than the other processes running on the system.

Well, you can do this using several Linux utilities. In this lesson, we're going to talk about how you can set priorities with the nice command, and how you can set the priority of a process that's already running with the renice command.

Let's begin by looking at how you can use the nice utility. The nice utility can be used to launch a new process on a Linux system, with a customized priority level.

PR and NI 1:38-4:13

Remember, both the top and the ps utilities can be used to display information about the processes running on a Linux system. And two of the values that will be displayed by ps and top are the PR value, which is the priority; and the nice value, NI.

Here's the key thing that you need to remember, and it actually is something that trips up a lot of new Linux system administrators. The higher the priority number right here, the lower the priority of the process. It's kind of counterintuitive.

In this example, we have most of these processes running with a priority value of 20, but there's another process down here that has a priority of 0. Because this one uses a lower number, it's actually a higher priority process than all of these other processes with a priority value of 20.

Just remember, with the priority column, the lower the number, the higher the priority of the process.

In addition to the priority, you need to be familiar with the nice value. This is displayed in the NI column. The NI value is the nice value of the process. It determines, basically, how nice the process is.

The nicer the process is, the less CPU time it's going to demand. The less nice a process is, the more CPU time it's going to demand.

The important thing to remember about the nice value is the fact that it's factored into the kernel's calculations, that will determine the overall priority of the process. The nice value for any Linux process can range between negative 20 to positive 19.

Notice in the output of the top command here that most of these processes have exactly the same nice value. They have the default value of zero. Which basically causes them to have the same priority on the system.

Notice, however, that there is one exception, and this is the kworker process. It has a nice value of negative 20, and as a result, it has a lower priority value than the other processes running on the system. Which means it has a higher priority level on the system.

The important thing to remember is that the lower the nice value is, then the lower the priority value will be, and, therefore; the higher the overall priority of the process on the system.

Using nice 4:14-4:49

Here's another key thing you need to understand. That is the fact that you cannot directly manipulate the priority of a process. But you can indirectly manipulate it by customizing a process's nice value. The easiest way to do this is to actually set the process's nice value when you initially load that process at the shell prompt.

You do this using the nice command. The syntax is shown here. You enter 'nice -n' followed by the nice level that you want to assign to the process, followed by the name of the actual command that you want to run.

Default Priority and Nice Levels 4:50-5:34

For example, let's suppose that I wanted to launch the gedit program from the shell prompt. Gedit is a graphical text editor that can be used on Linux systems. But I want to launch gedit with increased priority on the system. Because I take my text editing seriously.

By default, if I just run gedit from the shell prompt, it's going to receive a priority of 80 on the system, because it's going to be assigned a nice value of zero. That nice value, zero, is used by the kernel to calculate the overall priority of this process. And using its various calculations, that comes out to be 80. Because, as I said, I take my text editors very seriously.

Using nice 5:33-7:12

And I demand the best performance I can get when I'm editing text. I could adjust this process's priority level to a higher level by decreasing its nice value.

For example, here, I entered 'nice -n -15', followed by the 'gedit' command. Remember that by default, the nice value is 0. I'm cranking it down to negative 15. This will cause the priority number assigned to the process created by the command to decrease as well, which increases its overall priority on the system, and you can see that right here.

After running the command, my nice value reduced to negative 15 and my priority number reduced to 65 from 80, which increases the overall priority of that process on the system.

With that said, be aware that Linux is actually kind of hesitant to allow you to reduce the value of the nice parameter for processes running on the system. Because Linux is a true multi-user operating system, it is possible for multiple users on the same system to adjust the nice values of their own processes.

And now, human nature being what it is, naturally what is everyone going to do? Everyone is going to think, "Well my processes are more important than everybody else's." Every user on the system is going to try to crank down the nice value of their processes to increase their priority level on the system, and we're going to end up with a big tug-of-war.

Therefore, to keep this from happening, Linux won't let you adjust the nice value of a process below zero, unless you're logged in as root. Which means, if you aren't root, you won't be allowed to use a negative number with the nice command.

Using renice 7:13-9:58

The nice command works great for modifying the nice value when you're initially running a command at the shell prompt to start a process. But what do you do if the process that you want to modify is already running and you really don't want to, or can't, restart it?

Maybe it's running some mission critical application, and if you shut it off and restart it with a new nice value, you're going to have the entire company coming after you with pitchforks.

Well, in this situation you can't use nice. Instead, you can use the renice command. Instead of having to kill a process and restart it with nice to reset its nice value, you can use the renice command to adjust the nice value of the process that's already running, and the syntax is shown here.

We run renice -n, followed by the nice level we want to assign. And this time we reference the process ID number of the process whose nice value we want to change. So in this example, I've already got gedit running on the system. It has a nice value of negative 15, and its priority level is 65.

Let's say because I did that, I'm causing all kinds of havoc on the system, and I'm using way more system resources than I should be just to edit text files. In this situation, I want to increase the nice value of the gedit program; making it a nicer, more friendly, program, which will then lower its overall priority level on the system, freeing up resources for other users to use.

And because I do still take my text editing seriously, I don't want to actually exit out of gedit and reload it with nice. I instead want to change its nice value dynamically--while it's still running.

The first thing I do is run ps -elf to find out the PID number of the gedit process, which in this case is 4133. Then I can run the renice command, and I specify a nice value of 4.

Notice when I do--I run the ps -elf command a second time--and if I look at the priority and nice value of the gedit process now, it's much higher than it was before, making gedit a much nicer, friendlier program. The nice value is much higher than most of the other processes running on the system.

Therefore, my priority number is also higher, meaning that my priority level is lower on the system. I will get less CPU attention than the other processes running on the system. One quick thing to remember is that when you're using renice, you're still constrained by the same rule we talked about with nice.

And that is the fact that you have to be logged in as root if you want to adjust the nice level of a running process to a number that is less than zero--in other words, a negative number. If you're logged in as a regular user, you can set its value only to zero or higher.

Summary 9:59-10:02

That's it for this lesson. In this lesson, we reviewed how to manage process priorities using the nice and the renice commands.