10.2.4 Prioritizing Processes

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

Prioritize Processes 0:00-0:27

In this demonstration we're going to discuss prioritizing processes on the Linux system. On Linux, you can't directly set the priority level of a process, but what you can do is set its nice value. The nice value is used by the kernel to assign a specific priority level to that process.

To manage the nice level of a process, you can use two different commands: you can use nice and renice. Let's look at nice first.

nice Command 0:28-4:51

Let's run the 'ps -elf' command. When we do, you see that we have two columns in the output. We have the priority column and we have the nice column. The nice value that's assigned to a process can range from a value of -20 up to 19.

Basically, the nicer a process is, the lower the priority that will be assigned to that process. There's a relationship between the nice value and the priority level. The lower the priority, the higher the nice value. In essence, the process becomes more polite, allowing other processes to access the CPU before it.

On the other hand, the less nice a process is--for example, this process right here--the higher the priority of that process on the system. For example, you can see that many of these processes were assigned a value of 0. That's usually the default for most processes, although you can see that there are exceptions to that rule.

With a nice value of 0, we have a priority level of 80. Understand that the lower the number in the priority column, the higher the priority in the system. Think of it as a ranking where number one priority has a ranking of one, and number two has a ranking of two, and so on. A high number in the priority column indicates lower priority.

You manipulate this value by customizing the nice value. You can see here that this process has a nice value of -20. Hence, its priority is lower than the process that has a nice value of 0 and a priority value of sixty and, hence, has a higher priority on the system.

Just remember that the lower the priority value, the higher the priority on the system. A lot of folks get confused about that.

Let's go ahead and break out of here. Let's launch a process on the system using nice. We'll run 'nice -n', and then we specify the nice value that we want to use. Let's launch the 'gedit' program, the text editor, and let's launch it with a slightly higher priority on the system by decreasing its nice value.

Let's specify a nice value of '-5' and then run 'gedit'. Be aware that only root can run a process with a nice value less than 0. If you're a standard user, you're limited to using nice values between 0 and 19. You can't go any lower.

Root can go lower than 0, which, as we said earlier, increases the priority of the process on the system. Go ahead and run it. Our gedit application is now running.

Let's open up a new terminal. Let's switch to root. Let's run 'ps -elf' again, and let's pipe the output to the 'grep' command and search for the gedit process that we just launched. Notice that there are two results displayed.

Don't get too uptight about that. One of these is the grep command itself that was searching for gedit. The other is the gedit process itself. That's the one we're concerned with.

Notice right here that the nice value is slightly lower than those that were used for other processes. On this system, if I had not used nice, gedit would have been loaded with a nice value of 0 by default, and so its priority would have been 80.

We customized that using the nice command and we set the nice value to -5. Hence, the priority value was decreased to 75, which actually increases the priority of this process on the system. As a result, the gedit process would actually have more access to the CPU. It basically is allowed to butt in line ahead of other processes that have a lower priority value.

nice works great when you launch a process from the command line, but what if you already have a process running and you want to dynamically adjust its nice value to either raise or lower its priority on the system?

Maybe that process is a really important one that other users are using, so you really don't have the option of killing it and then reloading it with nice.

You need to keep it running. You can do that using the renice command. Let's close our gedit window and let's launch it again this time without setting its nice value. We're just going to load it with the defaults.

renice Command 4:52-6:41

To do that, we simply run 'gedit' at the command prompt and now go over to this window. Let's run our same command again to find out what the effect was on the priority and the nice value. We'll run 'ps -elf' again and we'll pipe it to 'grep' just looking for the gedit process.

You can see, because we did not customize the nice value, it loaded with the default value of 0 and, hence, its priority was decreased on the system. The priority went from 75 to 80.

Again, remember, when you increase the value of the priority, you actually decrease that process's priority level. It got bumped back in line, essentially.

Let's say we're working on a very important document here. We don't want to close out the application, but we do need to change its priority level in the system to increase it. To do that, we can run 'renice' and then we specify the custom nice level that we want to assign to the process.

Let's take it down even more this time, '-10'. Then we have to specify the process ID assigned to the process.

If we look in the output of the ps command, we can see the process ID that's been assigned to the gedit process right here. Be aware that this is the parent process ID, the ID of the process that launched this process.

Don't get these two confused. You want the process ID of the process itself, not its parent. We enter '3433' to specify the gedit process. Hit Enter, and notice that the nice value has changed. It was 0, now it's -10.

We run the 'ps' command again. We can verify that the nice value did go down to -10 from 0, and notice that this increased the priority of the process on the system by decreasing its priority number. It went from 80 down to 70.

Summary 6:42-6:54

That's it for this demonstration. In this demo, we talked about prioritizing processes. We first looked at using the nice command to launch processes with a custom nice value, and then we ended this demonstration by talking about how you can use renice to modify the nice value of a running process.