10.2.2 Switching Foreground and Background Processes

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

Switch Foreground and Background Processes 0:00-0:09

In this demonstration, we're going to practice switching processes between the foreground and the background.

Default Foreground Processes 0:10-0:53

Understand that whenever you run a command at the shell prompt, such as the 'gedit' command, below the gedit text editor, by default, its process is going to run in the foreground. Press Enter.

We see the gedit window loaded and notice here that because the gedit process is running in the foreground, I don't have access to the shell prompt and I won't be able to get access to the shell prompt again until this process closes out.

For example, I could come over here and click the X in this window to exit, or from the shell session itself I can press CTRL+c. Either one will do the same thing. The process is killed, the application is gone, but the shell prompt is back, so I can run another process.

Background Processing 0:54-3:10

There may be times when you need to run multiple programs from the shell prompt at the same time. In this situation, what you can do is run your processes in the background instead of in the foreground.

When you run a process in the background, it still runs normally; you can do everything you would normally do with it, but control will be returned to the shell prompt so that you can run other processes as well.

For example, if we wanted to run gedit in the background, I would use the 'gedit' command just like I did before, but I'm going to add a space and then I'm going to add an ampersand symbol (&) to the end of the command.

This tells the shell to load gedit but to run its process in the background instead of in the foreground. Hit Enter. You'll notice that the gedit window is displayed just like it was before, no difference there, but I have access to the shell prompt and I can do other things.

I can, for example, type the 'ls' command and view the contents of the current directory. I couldn't do that before.

When you run a process in the background, two values are displayed, as you see right here. This first value, the number 1, is the job ID number that was assigned to the background job when we created it.

This job number is very important because you can launch multiple programs into the background. Right now we have one only, but we could have launched three or four or however many we needed.

And if you need to manage those jobs, you have to be able to specify which one it is you need to work on, and you use the job ID number to do that. In addition, we also have the process ID number that has been assigned to that process as well.

You can view a list of jobs that are currently running in the background, using the jobs command. Just type 'jobs' at the shell prompt. Right now you can see that we have only one background job. Its status is displayed here. It's currently running and here's the name of the command that was used to run it.

If you have a process that is currently running in the foreground, you can switch that process from the foreground to the background, or vice versa. If you have a process that's currently running in the background, you can move it to the foreground if you need to.

In this example right here, we have the gedit process already running in the background. Let's say for some reason we decided that we don't want it in the background anymore, and we instead need to move that process to the foreground.

Process Switching 3:11-5:19

To do this, we type the 'fg' command (foreground), space, and then the job number of the background process that we want to move to the foreground. There's only one currently, so we just type the job number, number '1'. Press Enter.

Notice that as a result, nothing happened with my gedit window. It stayed running just like it was. I can still type in it and do whatever it is I need to do, but notice over here that we lost control of the shell prompt, just like if we had loaded the gedit process in the foreground in the first place.

We don't get access to the shell prompt again until we exit out of the gedit window, and now we're back to the shell prompt. So that's how you move a process from the background into the foreground.

You can do things in the opposite direction as well. You can take a process that's running in the foreground and then send it to the background. Let's go ahead and launch 'gedit' again, but this time note that we are not using an ampersand (&). So where are we going to run this process? We're going to run it in the foreground, the default.

So we press Enter and lose control of the shell prompt. The process is loaded and it's running in the foreground. Because we didn't use an ampersand (&) when we entered this command, notice that we did not assign a job ID to the process like we did before when we did use the ampersand (&).

Before we can move a process between the foreground and the background, we have to have a job ID number assigned so we can manage that job. To do this with a process that's running in the foreground, we press CTRL + z.

Notice that when I did that, the job number was assigned--job number 1 again. Because we killed our earlier job that was job number 1, that number became recycled and was used again for the current process.

Now that we've assigned a job number to the process, we can specify where we want it to run, background or foreground. In this case, it was already in the foreground, so let's just move it to the background by typing the 'bg' command followed again by the job ID number, '1'.

Press Enter and notice that nothing happened to the window up here. The process continued to run, but it was moved to the background. I have my shell prompt back and I can now use it to run other commands again.

Summary 5:20-5:37

That's it for this demonstration. In this demo, we talked about moving processes between the foreground and the background. We first talked about how processes are loaded in the foreground by default. We then showed you how to launch a process into the background. We then talked about how to move processes between the foreground and the background.
