4.4.4 Runlevels

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

Runlevels 0:00-0:48

As a Linux administrator, you need to know what happens as a Linux computer starts and how to manage the startup process.

In this video, I'm going to explain how runlevels determine the operating state, or mode, your computer runs in.

I'll also show you how to manage runlevels on a SysVinit or init system by editing the inittab file and using commands such as telinit and runlevel.

Before we begin, I want to make sure you understand that a Linux distribution might use the init system or the systemd system. Although many of the concepts concerning runlevels are similar on both systems, they are managed differently.

So, for this video, we're going to focus on how runlevels work on an init system.

It's also important to point out that you might find slight differences in how runlevels function from distribution to distribution.

Linux Runlevels 0:49-3:34

A runlevel is the preset operating state, or mode, that a system uses when it starts.

In a Linux system, there are seven possible runlevels, which are numbered 0-6.

Since each level has a different function, choosing the correct one for an individual user or a special requirement is important.

For example, to run a system designed for many users with a graphical user interface (or GUI) requires a different runlevel than a system that is designed for many users and non-graphical user interface.

In addition, the ability to choose the runlevel lets you save system resources or start the system in a minimal configuration so you can fix problems.

So, what is each runlevel's purpose? Well, let's start with runlevel 0.

Although it's a runlevel, runlevel 0 doesn't actually change the mode the system runs in. It shuts down or halts the computer.

In other words, if your system is running at a different runlevel, such as 3 or 5, and you decide to change the runlevel to 0, your system will shut down.

Next, we have runlevel 1.

In runlevel 1, Linux operates in a single user mode, meaning that only one user is allowed to log in. Usually, it's the root user.

This runlevel uses a command line interface, since there is no GUI, and all networking is disabled.

Runlevel 1 is ideal for system administrators to perform system maintenance, repair activities, or troubleshooting.

In runlevel 2, a Linux system runs in multiuser mode, meaning that multiple users can log in at the same time. The networking is still disabled, and there is no graphical user interface.

Runlevel 3 is very similar to runlevel 2, except networking services are started. This runlevel is commonly used for server-based systems when running without a GUI.

Now, you might be asking, "Why wouldn't you use a GUI?" The answer is that many system administrators don't want their servers wasting processor cycles redrawing the screen when the processing cycles could be doing something else, such as providing a web server or an FTP server. This is one of the reasons you have to know the Linux command line so well.

There's also runlevel 4 on Linux, but this runlevel is undefined, or user definable.

Runlevel 4 can be configured to provide a custom boot state.

Next, we have runlevel 5.

Runlevel 5 is the most commonly used mode used for workstation or desktop systems because it provides access for multiple users and includes a graphical user interface and networking.

Lastly, we have runlevel 6.

Runlevel 6 is similar to runlevel 0. It is not a mode that a Linux computer would boot to or actively run. Instead, runlevel 6 is used to reboot or restart a system.

In other words, if you need to reboot a system and you're at the command prompt, using runlevel 6 is one of many ways to do it.

Configure the Default Runlevel 3:35-4:38

Now that you know each runlevel's purpose, let's configure Linux to use the desired runlevel.

On a day-to-day basis, your system's runlevel will typically stay the same and is, therefore, set and changed using the /etc/inittab file. However, if you would prefer to use a different default runlevel or if you need to change the runlevel for something, such as troubleshooting, you can edit this file to make the change.

In older distributions, the inittab file was used to configure many things beside runlevels. But today, the inittab file is only used to set the default runlevel.

The line used to set the default runlevel uses the syntax id, colon (:), runlevel, colon, and then an initdefault action.

In this example, the current runlevel, 3, is being used. Remember, 3 is a multi-user mode with networking, but no graphical user interface. If you want to take advantage of a graphical user interface, all you need to do is open the inittab file using a text editor, change the 3 to a 5, and save the changes to the file. Now, the next time you boot the system, it will use runlevel 5.

Change Runlevels Dynamically 4:39-5:31

As you just saw, you can use the inittab file to change your default runlevel so that when the system boots, it goes directly to that particular runlevel.

However, you can also change your runlevels dynamically while the system is running from a shell prompt using the telinit command.

As the root user, simply type telinit and the desired runlevel and press Enter.

For example, if you're currently in runlevel 5 and you want to change to runlevel 3, all you have to do is type ‘telinit 3',

and the graphical user interface will disappear. You'll be taken to a text-based interface, and you'll be in runlevel 3. No reboot is required.

Most systems also support the init command, which works the same way as telinit, meaning that all you need to do is run init, space, and the new runlevel, and your system will be changed to that level.

But keep in mind that this is only temporary, meaning that the next time the system is started, the runlevel configured in the inittab file will be used.

View Current Runlevels 5:32-6:13

Before we end this video, I want to show you two different ways you can determine which runlevel is currently being used without looking at the inittab file.

The first method is using a shell script command named runlevel.

This command has no options. When you run it, it reads the user's temporary file (known as the utmp) and displays the previous file (if known) and current runlevels. If the previous runlevel isn't known, then the letter N is displayed.

The current runlevel can also be displayed using the who -r command, although the output is slightly different.

The who command specifies the current runlevel and the last runlevel, as shown here. If the last runlevel is unknown, then only the current runlevel is shown.

Summary 6:14-6:31

And that's it for this lesson. In this lesson, we discussed how you can manage the runlevels on a SysVinit system by viewing and editing the /etc/inittab file.

We also saw how the current runlevel could be temporarily changed using the telinit and init commands.

And lastly, we showed you how to view the current and previous runlevels using the runlevel and who -r commands.
