3.2.2 Installing Linux

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

Install Linux 0:00-0:10

In this demonstration we're going to walk through the installation of a new Linux system.

ISO Image From Fedora Website 0:11-1:56

On the system I want to install the Fedora distribution. I've gone out to getfedora.org, downloaded the necessary ISO image, and I have booted this system from that ISO image. Which meant I had to go into the BIOS of this system and tell it to boot from the optical drive first and not the hard disk drive, which is the default.

I now need to choose one of two different menu options. The first one is Start Fedora Live; the other one is Troubleshooting. We'll use this option here so we can start the installation process Start Fedora Live. Press Enter.

At this point, we have a decision to make. We want to actually physically install Linux on the hard disk drive in this system. I'm going to click that option.

The first thing I need to do is specify what language we want to use during the install. We'll go ahead and use the default selection of English and United States English and click Continue.

We need to configure several different installation parameters. The first one is which keyboard we want to use. We'll use the default of US English; that will work great. We also need to set the date and time and select our time zone.

By default, this Linux distribution has selected the Americas region and the city of Denver, which is essentially the Mountain Time zone. That's actually where I live, so I don't need to change anything.

If this was not the right time zone, then I would need to pick one by either clicking on it on the map or by going up here and selecting particular region and city from these two dropdown lists. I should come down here and make sure that the time display down here, as well as the date, are currently accurate. I'm going to click Done.

The next thing we need to do is design a partitioning layout.

Disk Partitions 1:57-2:53

You can use a default partitioning layout that the installer proposes for you or you can down here and say, "I want to configure my own partitioning scheme." Go ahead and click Done.

The first thing we need to do is determine what type of partitioning scheme we're going to use. LVM is selected by default. LVM is great, but it is a little more complex than a standard partitioning layout, and so that's what we're going to use here.

I've selected standard partitioning, so the first thing I need to do is define all the different partitions that I'm going to use on this particular system. Could you install everything in a single partition? Well, yeah probably. I don't think it would work very well. Instead, with Linux, we should create several different partitions.

The first one I'm going to create is actually a swap partition.

Swap Partition 2:54-4:03

The swap partition functions as virtual memory for this system. It allows the operating system to run more applications than it physically has memory to use. Essentially, it allows the system to use parts of your hard disk drive as virtual memory.

It allows the system to load more applications than it has enough physical RAM to support by using some pieces of your hard disk drive as if it were RAM. It's slower, but it provides some extra capacity.

We have to decide how much memory we want to allocate.

The general rule of thumb is that your swap partition should be about 1.5 times the size of your installed RAM. Is it the end of the world if you don't make it 1.5 times the size of your RAM? No. If you make it twice as a big, for example, it will still work great.

In fact, it will probably work wonderfully; just it's used up some of your hard disk space for swap partition that you probably didn't need to. But here's the key point: if you make it too small, your system performance will probably suffer.

This system has 2 GB of RAM in it, so I'm just going to make a 3 GB swap partition. That's 1.5 times the size of the physical RAM installed on the system. Add mount point. We've got our swap partition created.

Mount Point 4:04-4:51

The next thing we need to do is create a mount point for the root of our file system, '/'. This mount point is going to store the entire Linux operating system. It needs to be about around 10 GB in size, probably, at a minimum.

For me that seems a little small. I'd rather make it more like 12 GB. This is a demonstration system. Would I make a 12 GB root partition on a production system? Absolutely not. I would make it much larger, at least 10 times larger than what I'm setting it here, but for our purposes it will work.

Add the mount point. We've got a swap partition and we've got a root file system. I want to make two additional partitions and mount them at other locations in the file system.

/home 4:52-5:46

First, I want to create a separate partition for /home. This is where our end users' /home directories reside.

Creating a separate partition for /home has a lot of advantages. First of all, if your end users create huge files that consume all the space in the file system. By constraining the /home directory to its own partition, the users who create those gigantic files will not bring the system down by taking away the space needed by the operating system.

In addition, if we ever need to migrate these users to a different disk or we need to reinstall Linux, we can keep their data and preserve the /home partition during the installation process, and instantly have those accounts' data available when the system comes back up.

This is just a demonstration system, so we're not going to make it as big as we would on a production system--about 2 GB in size. Again, this would be way too small on a production system.

/var 5:47-9:21

I'm going to add another partition for /var. Now you might be asking, 'Why would you create a separate mount point for /var?" /var is where all of your log files go, and if something goes wrong with your log rotation configuration, such as those log files keep growing bigger and bigger, it can actually crash your system as it consumes so much space that there's none left for the operating system to use.

I've actually had this happen on one of my client's systems.

They called me up and said, 'I can't log in to my Linux system. It's horrible. Everything is blowing up.' I went and looked and I was able to discover that the log files had grown so big that they had consumed all the free space on the hard disk drive. That system did not have a separate partition for /var.

By putting /var on its own partition, oversized log files would be constrained to this partition and they will not affect our operating system files over here on the /file system. Once again, on a live production system I would not create a 4 GB partition for /var; I would make it much larger for a production system. Hit Done.

Here's a summary of changes that are going to be made as a result of our partitioning. Let's go ahead and accept changes. The last thing we need to look at over here is our networking. It's a really good idea to have this Linux system plugged into a network that has internet access during the installation process.

You can actually do this with either wired or wireless configuration.

If you use wired and you have a DHCP server running on your network, it's pretty straightforward because it just goes out and gets an address on the DHCP server and then is able to go out on the internet.

If you use a wireless connection, then you have a few more configuration steps to perform. They're not too tough. You basically have to just configure the appropriate SSID and key for your wireless network in order for it to connect.

The reason why this is a good idea is because during the installation process, the Linux installer can actually go out on the internet to the repositories for your Linux distribution and actually pull down updated versions of the operating system files as needed during the install. That way when you're done with the installation, you have a fully updated running system.

One of the things we need to do here is actually provide a hostname for the Linux system. We'll call this fs2.corpnet.com. We'll hit Done. That's all we need to do. We're ready to start the installation process. I'm going to click Begin and notice down here that the installation is actually starting in the background.

First, it starts creating all these partitions that we configured. It will take a few minutes to complete. Once that's done, it's going to actually start installing the operating system files on those partitions. However, as it does so in the background, there are a couple of other configuration tasks we need to complete.

First of all, we need to configure our root user's password. Remember, the root user is the super user on the system. Root can do anything it wants with any files and directories on the system. Therefore, it needs to be secured with a very strong password. It should have at least 8 characters, use upper and lowercase characters, and numbers or symbols.

Notice down here that there is a password strength meter right here that will tell you, as you type, how strong your password is. I'm going to go ahead and start typing my password here and I've created a fairly strong password. Passwords match, click Done.

User Accounts 9:22-11:59

The next thing I need to do is create a user, a standard user account, and this is very important. Could you get away without creating one? Yeah, but it's really a bad practice. A lot of Linux administrators get into a bad habit of always logging in as the root user and performing all their tasks on the system as the root user.

The problem is doing that creates a security vulnerability. If you're logged in to your system as root, you get up to go use the restroom or go to the break room or something like that and leave your system logged in as root. Somebody could very easily come sit down and do whatever they wanted with your system because they would have full access. We don't want that.

Instead what you should do is log in as a standard user account and then switch to your root user account whenever you need to perform root level tasks, and when you're done you come back to your standard user account. Entering my name here. I'll use the default user name that it creates for me. I do want to require a password to access this account and a strong password and we'll click Done.

At this point, we need to just let the operating system software be installed. We'll come back when we're closer to being done. Okay, the installation process is complete. It was successfully installed and we're ready to use the system. All we have to do is reboot it at this point.

We go up here, we click on the power button, and then we click on Restart. At this point I'm going to log in using the user account that I defined during the installation process. I put in the password that I defined during the installation process. We have a working, functioning Linux system.

Because this is the first time we've logged in to the system, there are a couple of additional configuration parameters that we need to make. Right here we are configuring the language that will be used in the Gnome environment. I'm going to go ahead and use English and we'll use the US English as listed down here.

Here we can configure our privacy settings. For example, we can turn location services off or on. As to whether or not we're going to allow our location to be used, let's just use the default settings for now. If we wanted to, we could connect this system to an online account either with Google, with Facebook, or even with our Windows online account, but we're not going to worry about that today.

Skip and we are ready to go start using Fedora, and a Gnome getting started guide is displayed to help you learn how to use the new Gnome environment. We now have a functioning Linux system.

Summary 12:00-12:15

That's it for this demonstration, in this demo we talked about how to install Linux. To do this, we first booted our system off of an ISO image that we downloaded from the Fedora website. We went through the installation process, we configured our disk partitions, we installed our operating system files, and then created our user accounts that we use to log in to the system.
