3.1.1 System Design Part 1

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

System Design Part 1 0:00-1:08

To install Linux properly, you need to spend some time planning the installation before you actually start the installation process. Now, if you're working on a test system at home or in a lab environment, you can usually get away with implementing Linux without a lot of planning. There's probably no real risk in making a mistake. In fact, it can be a great learning experience, and I highly recommend it when time permits. You can really learn a lot.

However, if you're going to be deploying systems that will be used in your organization's production environment for day-to-day work, then this unplanned, haphazard approach is completely unacceptable. That's because mistakes on your part could lead to system outages, and outages cost your organization time and money and will probably also cost you your job.

So, instead of deploying systems in a haphazard, unstructured manner, you instead should develop a deployment plan before you start buying hardware and installing software. Doing this will prevent a lot of costly errors, and it will save your job.

So, in this lesson, we're going to go through the process of planning a Linux installation, and this process involves several key tasks.

System Design Process 1:09-1:50

First, conducting a needs assessment. Second, selecting a distribution. Third, verifying system requirements and hardware compatibility. Fourth, planning the file system. Fifth, selecting software packages. Sixth, identifying your user accounts. Seventh, gathering network information. And then, finally, selecting an installation source. Now, this process is fairly long, so we're going to break it up into two parts. In this first lesson, we're going to review the first three steps of the system design process, conducting a needs assessment through verifying system requirements and hardware compatibility. The remaining steps will be covered in a separate lesson. So, let's begin by looking at the first step: conducting a needs assessment.

Conducting a Needs Assessment 1:51-4:11

Now, conducting a needs assessment is one of the most important aspects of creating a Linux deployment plan, but it's also the most frequently skipped step. Unfortunately, even if someone does do it, it's usually done very poorly. So, what exactly is a needs assessment? Well, it's the process of determining why this Linux deployment is being undertaken. How many systems are involved? What outcomes are expected? When is it going to be done? Please be aware that completing this needs assessment is going to require you to change a few things. For example, you're going to have to take off your computer admin hat and put on a different hat that's labelled Project Manager. And in this role, you're going to need to meet with a lot of different individuals and gather data about this deployment, which means you're going to have to talk to other carbon-based lifeforms. I know it's difficult, but I believe that you can do it.

Your findings should be recorded in a document that can be easily distributed and reviewed by others. And when you're done, the needs assessment portion of the deployment plan should contain the information shown here. First of all, what are the goals of this project? The first thing you should find out is why this implementation is even being requested. What problem is it designed to fix? What will the final result be? What objectives will be met by this implementation?

Second, identify who the stakeholders are in this project. This is very important. You need to identify every single individual who is going to be impacted by this project in any way. For example, who requested this system? Who is going to use it after it's installed? Who has the authority to approve funds to pay for this project? Who has the authority to allocate your time to that project? Who has to get final approval before you can start? And a very important one is, who will actually maintain and support it after it's done?

Another key question you should ask is when this whole project needs to be done by. Before you can create a schedule for this project, you have got to know when the stakeholders expect it to be complete.

By gathering all of this information, you can define one of the most critical components in your deployment plan, and that's the project scope. The project scope identifies exactly what will be done, when it's going to be done, and who is going to do it.

Selecting a Distribution 4:12-6:28

So, with your project scope identified, you're ready to move on to the next part of the plan, and that is selecting a distribution to install. Now, as you know,

Linux is available in a wide variety of flavors called distributions. Therefore, a key part of your plan is picking which of these distributions you're going to use. Now, you can't just go out and pick whatever you want. The distribution you choose depends upon what you want the system to do. You can use these guidelines here to help guide your choice.

First of all, will the system function as a workstation or a server? One of the cool things about Linux is the fact that you can implement just about any distribution as either a workstation or a server, but you need to be aware of the fact that there are Linux distributions available that are specifically designed and optimized to function either as a server or as a workstation.

For example, Red Hat provides the Red Hat Enterprise Linux distribution, and it's designed specifically for providing network services for a large organization. However, Red Hat also provides other distributions that are specifically designed for use in a desktop environment, such as Red Hat Enterprise Linux Desktop.

You also need to see whether or not that distribution offers adequate support. This is very important because some vendors offer technical support for their distributions, whereas others offer very limited support or no support at all. And if this system's going to be used in a corporate environment, you need to use a well-supported distribution. That way, if a problem occurs at some point after installation, you will have what you need to be able to resolve the issue and get the system back into production very quickly.

In these situations, you don't have time to go searching through a bunch of forums on the internet, trying to figure out what's wrong and how to fix it. You need to be able to call somebody and get an answer fast. Now, this is something that kind of trips up a lot of folks, because they're used to everything being free when you're working with Linux. This is not true of support. Although the distribution itself may be free, or very nearly free, you will probably be required to pay for any support. And the price that they will charge you will vary from vendor to vendor, so it really pays to look around first.

In addition, you need to evaluate the software that you want to run after the installation and verify that it will actually run on the distribution that you're picking.

Hardware Support and System Requirements 6:29-8:44

Now, you should also verify that the distribution you want to use will run on the hardware that you plan to install it on.

Back in the early days of Linux, hardware compatibility was problematic, especially if you were trying to install Linux on a notebook system or some other system that had proprietary hardware. And the problem back then is there just weren't enough developers writing Linux drivers.

Today, this is much less of an issue. Most hardware vendors now offer drivers for Linux that support their hardware. In fact, most of the drivers for common PC hardware are available right within the distribution itself.

To be safe, however, it's still a very good idea to check your distribution's website and verify that your system's hardware is listed on your distribution's hardware compatibility list.

Now, there's one issue here that you have to be aware of: issues regarding UEFI. Now, most hardware that uses UEFI instead of a BIOS are configured with a feature called Secure Boot, which basically forces the operating system to be digitally signed before the motherboard will actually boot that operating system.

In the old days, when we had a BIOS on the motherboard, we could buy whatever systems we wanted, blow away whatever operating system came with it, and drop Linux on top of it, no sweat. That is not the case with UEFI. You need to do your research to find out whether or not you can boot Linux from that particular hardware manufacturer's implementation of UEFI.

My experience has been that some systems will boot Linux without any problems. Others will absolutely refuse to boot Linux, so do your homework first so you don't end up with a lot of hardware that doesn't work.

Now, you also need to make sure that you download the correct version of the distribution for your system's architecture.

For example, if you have a really ancient system that uses a 32-bit CPU, then you'll need to download the x86 32-bit version of your distribution-- typically not a problem anymore because 32-bit CPUs are dinosaurs. Most likely, your hardware's going to use a 64-bit CPU, in which case you need to make sure that you download an x86 64-bit version of the distribution. If you pick the wrong one, most Linux installers will pop up an error message and won't let you complete the install.

Summary 8:45-8:54

That is it for this lesson. In this lesson, we reviewed the first three steps in the system design process. We talked about conducting a needs assessment, selecting distribution, and verifying system requirements and hardware compatibility.