Chapter 1

Preparing Your Environment

✔ **Objective 1.1: Explain basic Linux concepts**

Before beginning your journey to successfully pass the CompTIA Linux+ certification exam, you need a learning space. A learning space consists of Linux systems (virtual or physical), where you can actively try out, practice, and explore various Linux commands and utilities. Besides reading this book, having a private space to work freely will assist in your success.

You may already have experience working with Linux in your enterprise environment. However, most likely you are using only one Linux distribution. Training with more than one distribution is needed to pass the Linux+ certification exam.

In addition, your employer may frown upon any risky behavior on its systems. You need to feel free to try out Linux commands that may cause a system to crash. Your own learning space, containing various Linux distributions and their assorted tools, is a key factor in successfully passing the Linux+ exam.

This chapter begins by looking at a few items concerning the setup of your learning space environment. First is a quick explanation of the parts that make up Linux. Next, the chapter discusses how to run Linux in your learning space. A deep dive into the various Linux distributions for your learning space follows. At the chapter's end, I'll cover a method for accessing the Linux command line.

Exploring Linux

Before diving into how to set up a Linux learning environment, this section explores just what the pieces are that make up a Linux system.

What Is Linux?

If you've never worked with Linux before, you may be confused as to why there are so many different versions of it available. You've most likely heard terms such as kernel, distribution, and GNU when looking at Linux packages, and you may have been confused.

Although people usually refer to the Linux operating system as just "Linux," in reality, there are four parts that make up a complete Linux system:

- **The Linux kernel** interfaces between software and the computer hardware on the computer system, managing CPU usage, memory, and devices.

- **The GNU utilities** provide command-line programs for managing files and programs, created by the "GNU's Not Unix" project.
- **A user interface** provides a way to communicate with the operating system, usually by either using a graphical desktop environment or a command line.
- **Application software** provides desktop and server programs for such diverse things as editing documents and serving web pages.

Although each of the parts by itself isn't very useful, put together, they all create what people refer to as "Linux." A Linux *distribution* bundles each of these parts to meet the specific needs of a community, such as developers, artists, or hobbyists.

What Does Linux Run On?

Since the Linux kernel operates directly with the CPU, each kernel release must be compiled for a specific CPU environment. However, these days there is a multitude of different types of computer systems that Linux can operate on, from the small Raspberry Pi devices to the large IBM mainframe computers. When you download a Linux distribution, you must download one built for the CPU type that your computer system uses.

The Linux kernel has been compiled to run on several different platforms, but the ones stressed in the Linux+ exam are:

- **Intel/AMD x86 and x86_64:** Intel 32-bit and 64-bit CPUs used in most desktop and laptop computers.
- **AMD64:** Advanced Micro Devices (AMD) 64-bit CPUs.
- **ARM (aarch64):** The Advanced RISC Machines CPUs, developed by Arm Ltd. Corporation. These CPUs are used in small-scale systems such as the Raspberry Pi.
- **IBM Z (s390x):** IBM series of mainframes and minicomputers.
- **RISC-V:** The Reduced Instruction Set Computing, version 5 open standard architecture that can be used by any CPU manufacturer.

That's a wide range of CPUs that covers a lot of different types of computer platforms. Most likely there's a Linux distribution created for your learning environment.

What About Licensing?

With most software packages you must agree to an end user license agreement (EULA) that dictates how you can use the software. Most commercial software packages are released under a *proprietary license*, which restricts what you can and can't do with the software, usually forbidding you from making any modifications to the software itself.

Linux is released under the *open source* license model. The open source model allows you to obtain the actual program source code used to create the software. However, there are

different open source license models, based on how you are allowed to use the source code. There are two main license models, each with different submodels:

Copyleft Licenses Allow you to download and use the software source code, but restrictions apply on how you can use or distribute it. Any changes you make to the code inherit the license terms of the parent software. The two most popular Copyleft license models are:

GNU General Public License (GPL) Allows you to make changes to the source code, but you must release your changes to the public under the GPL license.

GNU Lesser General Public License (LGPL) Allows you to integrate the source code (or even just parts of it) into your own project without having to release it to the public.

Permissive Licenses Allow you to download and use the source code, but don't place any restrictions on the redistribution of it. Popular permissive license models include:

Apache License Allows you to create derivative software that uses a different license model.

MIT License Allows you to do anything you want with the source code, as long as the original copyright notice is included with all derivative work.

While the Linux kernel itself is licensed under the GPL version 2 license, not all Linux distributions use this license, as they include third-party software licensed under other license models.

The next section dives into what you'll need to set up an environment where you can experiment with Linux.

Setting Up a Learning Space

Your learning space needs to be an environment where you can freely explore Linux and its various distributions (called *distros* for short) and utilities. Whereas some companies may have a spare Linux server available for you to fully use, many of us are not so lucky. Even if you are a student, with a nice lab environment already set up and available for your use, you may want your own space, where you can explore without restrictions.

Though there are many different ways to set up your personal learning space, I will focus on only a few, such as setting up Linux on an old laptop, implementing a virtualized environment, and using the cloud. Hopefully the ideas here will spur you on to setting up a helpful exploration and study environment.

Using That Old Laptop

If you've got a spare or old laptop sitting around, repurposing it as your Linux learning space may work well for you. This is especially useful if you like to move your study

TABLE 1.1 Hardware requirements for using single distribution at a time

Resource	Minimum	Recommended
Memory	2 GB	>= 4 GB
Free disk space	25 GB	>= 30 GB
Processor	2 GHz dual core	> 2 GHz dual core

environment, such as, for example, moving to a different and quieter location in your home when things get a little loud and crazy. An old desktop will also work, but you will be less mobile.

Whatever system you choose, you need to ensure that it has enough capacity to handle the minimum hardware requirements for a learning space. If you plan on installing multiple Linux distributions on a single system, booting them individually, and not using a virtualized environment, then Table 1.1 will serve as your requirements guide.

Though you can use this learning space, it is certainly not ideal. In addition, you can expect this type of Linux learning environment to boot and operate slowly. This learning space environment should be used only if you have no other options.

Creating a Virtualized Environment

Creating a virtualized environment for your Linux learning space is ideal. This setting will allow you to boot multiple Linux distributions at the same time, enable you to move quickly between them, and provide compare and contrast experiences. In addition, you can explore networking utilities more thoroughly in such an environment.

If you are unfamiliar with a virtualized environment, do not despair. Not only are there many resources on the Internet that can get you up to speed, I also cover virtualization concepts in Chapter 30, "Inspecting Cloud and Virtualization Services."

There are several excellent and free virtualization products (called *hypervisors* or *virtual machine managers*) that you can install. They include the following.

Oracle VirtualBox This actively developed open source software is available at `www.virtualbox.org`. It can run on Linux, Windows, macOS, and even Solaris. You can use VirtualBox to run multiple Linux distributions at the same time, assuming your hardware has enough resources. The website is loaded with helpful documentation and has community forums to help you create your Linux learning space.

Microsoft Hyper-V Server 2019 This closed source virtualization product is available on many current Windows 64-bit versions, such as Windows 11 Professional and Enterprise. However, Windows 11 Home edition does not support it. You can use Hyper-V to run multiple Linux distributions at the same time, assuming your hardware has enough resources.

QEMU The Quick Emulator (QEMU) package is an open source emulator that is different in that it emulates a computer's CPU using dynamic binary translation (translating the CPU instructions from one CPU type to another). This enables you to run operating systems compiled for a completely different CPU type than the host system. You can find QEMU at `www.qemu.org`.

Please don't feel limited by this list. It includes only a few suggested hypervisors to investigate. If you have found a virtualization product that works better for your environment, use it for your learning space.

The most popular hypervisor for Linux host platforms is the Kernel-based Virtual Machine (KVM) software. While this is popular on Linux hosts, it's not overly useful as a learning space since you already need to have Linux installed and running to use it.

Before selecting and installing a particular hypervisor, determine if your laptop or chosen system has enough capacity to handle the entire learning space's minimum hardware requirements. If you plan on installing and running multiple Linux distributions at the same time, use Table 1.2 as a guide for your needed hardware resources. However, be aware that the virtualization products' websites may provide more detailed information.

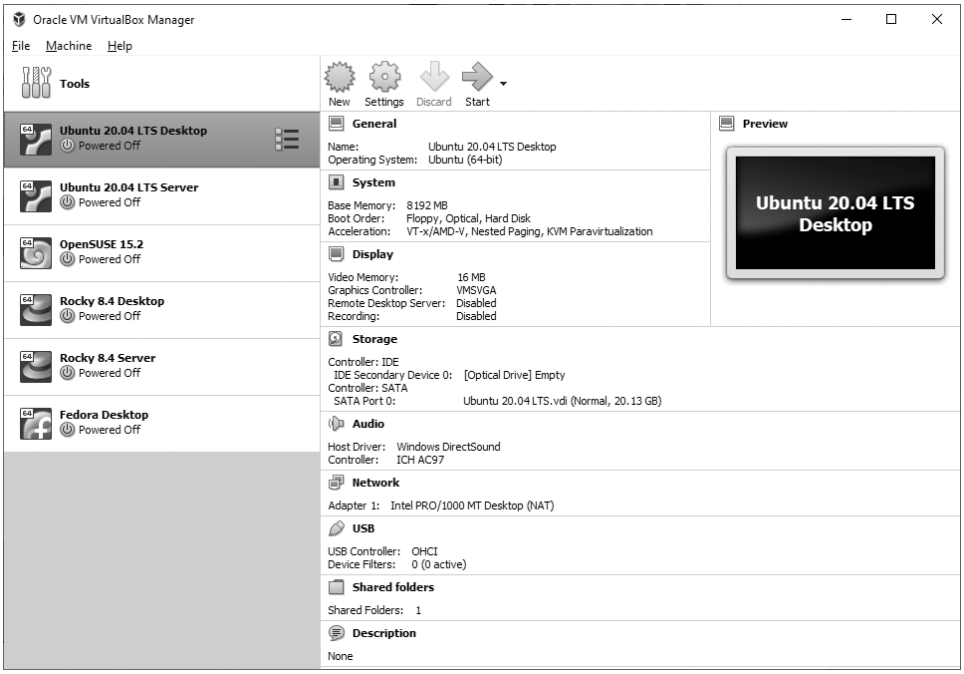
Using a virtualized learning space is very flexible. Figure 1.1 shows an example of this type of elastic learning space environment.

Notice in the learning space depicted in Figure 1.1 that there are two installations of both the Ubuntu and Rocky Linux distributions. These distributions provide the ability to install

TABLE 1.2 Hardware requirements for using a virtualization product

Resource	Minimum	Recommended
Memory	8 GB	>= 8 GB
Free disk space	70 GB	>= 100 GB
Processor	x86_64 2 GHz dual core	x86_64 > 2 GHz dual core

FIGURE 1.1 Learning space using Oracle VirtualBox



either a server-oriented environment or a graphical desktop-oriented environment. With VirtualBox you can easily install both environments and compare them!

Hopefully you are starting to gather some ideas of how you want to configure your private learning space. Before you do, there is one more platform category I need to explore.

Jumping to the Cloud

If you do not own a laptop or desktop with enough resources to provide a multiple Linux distribution learning space, consider the cloud. While cloud servers have become increasingly popular for large environments, they can also provide an easy way to run just a single Linux system.

There are many cloud service providers where you can start up various Linux distribution virtual machines, such as Amazon Web Services (AWS), Microsoft Azure, and DigitalOcean. Cloud services change rapidly, so you may not be able to find the Linux distribution versions you need. However, it is worth your time to take a look at the various offerings from cloud service providers. The cloud just might be a cheaper option for your learning space than a new computer.

If you choose to use a cloud service, the service may not allow you a way to explore certain CompTIA Linux+ objectives—for example, modifying how a Linux server boots, such as via BIOS versus UEFI. Keep this in mind as you explore your learning space venue.

Before you settle on the location for your learning space, consider the various recommended Linux distributions and their versions. These are additional components of your successful learning space environment.

Exploring Linux Distributions

The CompTIA Linux+ certification is vendor neutral. In practical terms, that means no particular Linux distribution is the focus of the exam. If you only have experience with Red Hat Enterprise Linux (RHEL), you need to learn more about utilities and features on Ubuntu and openSUSE distributions and vice versa.

It is tempting to think that Linux distributions are all the same and that few differences exist between them. Unfortunately, this is a fallacy. I like to compare the Linux kernel to a car's engine and a distribution to a car's features. If you have ever rented a car, the car's features are often rather different than the features of the car you normally drive. When you get into the rented car, you have to take a few minutes to adjust the seat, view the various car controls, and figure out how to use them before taking off onto the roadway. This is also true with learning new distributions. The good news is that if you have lots of previous experience with Linux, learning a new distribution is not that difficult.

Linux distributions are often based on other distributions or distribution forks. Two popular distribution groups, which contain distributions helpful to passing the Linux+ exam, are Red Hat–based and Debian-based (also called RPM-based and dpkg-based, identifying the software packaging method they both use). Differences between these two groups include software packages, names, and their management tools; configuration filenames and/or locations; software release schedules; firewall configuration utilities; and so on. Red Hat Inc. tends to focus on business and enterprise computing, while the Debian Project focuses on free software. Due to these various differences, it is necessary to use distributions from both groups in your learning space.

It is important to understand which Linux distros will help you in successfully passing the CompTIA Linux+ certification exam. In addition, you should know which particular distribution versions are helpful.

Looking at Red Hat Enterprise Linux

The original Red Hat Linux started life in 1995 as an open source project. It gained in popularity to the point where it was at one time the most popular Linux distribution, used in educational environments, corporate environments, and even by casual Linux hobbyists.

However, in 2003 Red Hat discontinued the Red Hat Linux project in favor of the Red Hat Enterprise Linux (RHEL) project. The RHEL project is primarily focused on business Linux environments. RHEL is a commercial package; thus under most situations you must purchase a license to use it. In return, Red Hat provides full customer support to help with setting up and troubleshooting the Linux system, unlike most other Linux distributions.

Fortunately for Linux hobbyists, there is an alternative way to run RHEL. Since Linux is an open source software package, Red Hat is required to release the source code for RHEL. A few other Linux distributions have popped up using the RHEL source code. The most popular had been the Community Enterprise Operating System (CentOS). It was nearly an exact duplicate of RHEL, and a great free study resource for the CompTIA Linux+ certification exam.

However, as is often the case in the fast-moving Linux world, things have changed. In 2014 CentOS joined Red Hat's Open Source and Standards team, and in 2020 Red Hat replaced the original CentOS project with a new development version called CentOS Stream. While you can still freely obtain CentOS Stream, it's no longer an exact duplicate of the current RHEL version, but rather a testing ground for new concepts, making it less beneficial as a study resource.

The Fedora Linux distribution is another Red Hat Linux testing ground, used mostly for the desktop environment.

But have no fear, the original developers of CentOS have started yet another distribution, named Rocky Linux. Rocky Linux has gone back to the origins of CentOS—it's an exact duplicate of the latest RHEL version. You can obtain a Rocky Linux distribution ISO from the Rocky website at `www.rockylinux.org`. Be aware that this distribution, like many others, comes in multiple flavors. I recommend you obtain Rocky version 9.* Default Image download package.

As time goes on, new Rocky distribution versions will be available. While it is always tempting to get the latest and greatest version, it is not beneficial to use it in your learning space. Remember that the CompTIA Linux+ objectives are static, until the next time the certification exam is updated. Therefore, it is wise to use the distribution versions that were available at the certification exam's creation time.

As you install Rocky Linux, you'll be prompted for the environment you want to install. For learning Linux, it's usually best to install a graphical desktop environment, because that provides the easiest way to access all of the Linux features, you'll need to learn about.

After you install your Rocky Linux version 9.* Default Image distribution, you should update the software packages. Do this by logging into the root account using the password you set up during installation and issuing the commands shown in Listing 1.1.

Listing 1.1 Updating software on Rocky Linux

```
# sudo dnf update
Loaded plugins: fastestmirror
[...]
Upgrade  3 Packages

Total download size: 1.3 M
Is this ok [y/d/N]: y
[...]
Complete!
#
```

Another package compiled using the RHEL source code is Alma Linux. While relatively newer than Rocky, it's quickly gaining in popularity.

While RHEL (and its derivatives) is a popular distro, you also need a distribution in the Debian camp. Next, I'll explore the Ubuntu distribution.

Looking at Ubuntu

The Ubuntu Linux distribution is managed by Canonical LTD and has been around since 2004. This free and popular Linux distro is based on the Debian distribution and is a must-have in your personal Linux learning space.

You can obtain the Ubuntu distro ISO from `www.ubuntu.com`. There are several flavors of Ubuntu, and if you'd like to ensure that you can follow the examples in this book, we recommend you download the Ubuntu Desktop version 24.04 LTS.

The LTS in the Ubuntu version name stands for Long-Term Support. This is an indicator Canonical uses to show it will provide maintenance and security updates for an extended time period. In the case of 24.04 LTS, you can count on free updates through April 2029, and you can purchase a subscription to receive updates until April 2034.

If you are unfamiliar with Ubuntu, you need to be aware of a few important items. By default, you cannot log into the root account. Instead, when you need to use super user privileges, log into the account you set up at installation and put the command **sudo** in front of your command-line commands. An example is shown in Listing 1.2.

Listing 1.2 Using `sudo` on Ubuntu

```
$ sudo grep root /etc/shadow
root:!:17737:0:99999:7:::
$
```

If you have never issued command-line commands in a terminal, it is recommended you read the entire chapter prior to attempting to do so. You will read more about terminals later in this chapter.

Another important item concerns installing Ubuntu. If you are connected to a network, you can automatically update the distribution's software when you install the distribution. You will see this option listed in the installation process as `Download updates during the installation` with a check box next to it. If you choose to not install updates during the installation, you can update the software via the command line later on by manually issuing the commands shown in Listing 1.3 in a terminal, using super user privileges.

Listing 1.3 Updating software on Ubuntu

```
$ sudo apt update
[sudo] password for rich:
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
[88.7 kB]
[...]
Fetched 1,053 kB in 2s (631 kB/s)
Reading package lists... Done
$
$ sudo apt dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
[...]
Do you want to continue? [Y/n] Y
[...]
$
```

If you have room for only two Linux distros, Rocky Linux and Ubuntu make fine choices. If you have additional resources, it would be worthwhile to add another distribution, openSUSE.

Looking at openSUSE

The openSUSE distro had its first release in 1994, under a different name, SUSE Linux. There have been many companies involved in supporting it, with the Germany-based company SUSE being the original.

This distro has a very loyal and solid following. Not only is the openSUSE distribution strongly supported by community developers, the openSUSE users love it as well. One of its unique and popular utilities is Yet another Setup Tool (YaST). YaST, which can be thought of as a command-center utility, allows you to control many system services from one interface.

You can obtain the openSUSE distribution ISO from `https://software.opensuse.org`. This distro comes in two primary flavors, Leap and Tumbleweed. We recommend you select openSUSE Leap version 15.*.

The openSUSE community changed its distribution's version numbering scheme in 2017. The version before 15.0 was 42.3. Be aware of this dramatic change when you go to obtain openSUSE Leap.

Once you have successfully installed openSUSE, it is a good idea to update all the software before exploring this distro. To update the software via the command line, manually issue the commands shown in Listing 1.4, in a terminal, using super user privileges.

Listing 1.4 Updating software on openSUSE

```
$ sudo zypper patch
[sudo] password for root:
Loading repository data...
Reading installed packages...
Resolving package dependencies...
[...]
    Note: System reboot required.
Continue? [y/n/...? shows all options] (y): y
[...]
Warning: One of the installed patches requires a
reboot of your machine. Reboot as soon as possible.
There are some running programs that might use files
deleted by recent upgrade. You may wish to check and
restart some of them. Run 'zypper ps -s' to list these programs.
$
```

You may have noticed that the last three distros use different commands for updating software. This is another reason you need to have access to multiple distributions in your learning space.

Locating a Terminal

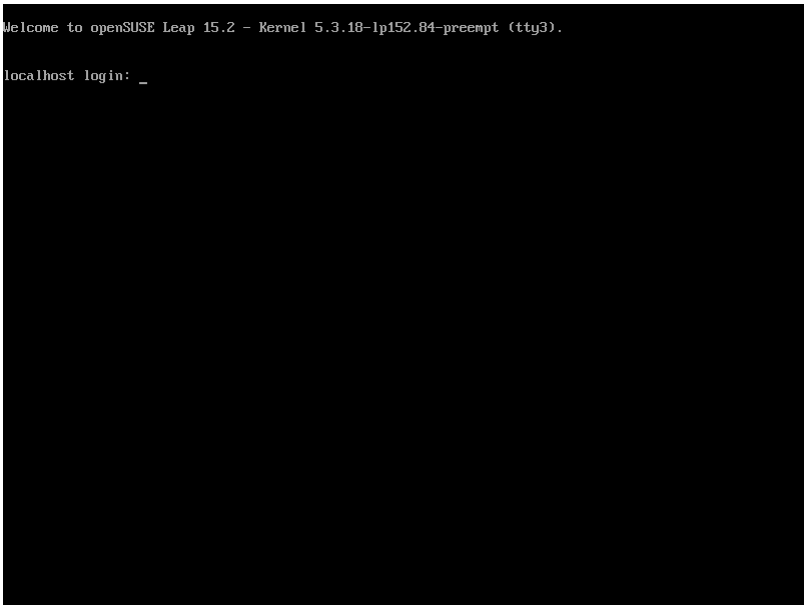
For exploring Linux and preparing to take the CompTIA Linux+ certification exam, you need to spend some time at the command line. The terminal is your gateway to the command line. Once you understand how to locate and use this terminal, you can start progressing through the rest of this book's contents.

Server-oriented distributions only provide command-line access, so that's not a problem. However desktop-oriented distributions don't show a command line by default. The simplest way to reach a terminal in those distributions is by pressing the key combination Ctrl+Alt plus one of the functions keys (usually F2 or F3) after the system boots. This will take you to a terminal named tty2. After entering the username and password you created during the Linux distribution's installation, you will be provided a prompt. Figure 1.2 shows a tty3 terminal on the openSUSE distribution.

At the terminal prompt, you can start entering commands. If you have newly installed the distro, go ahead and update its software as directed earlier in this chapter. To leave this terminal, simply type the command **exit**.

If you're using a graphical desktop environment, you can also access the command line by using a terminal application.

FIGURE 1.2 openSUSE tty3 terminal



Summary

Linux distributions bundle several pieces of software together to create a complete Linux system. The Linux kernel is the core of the system, but also needs the GNU utilities to allow you to interact with files and programs, as well as some type of user interface (either graphical or text-based). Finally, you'll want some application software to use on the Linux system.

A learning space is a virtual or physical Linux system where you can explore, practice, and try out various Linux commands and utilities. A private learning space is a necessity to be successful in passing the CompTIA Linux+ certification exam. You can set up a learning space on an old laptop, on a current laptop using a hypervisor, or within the cloud.

Having multiple Linux distributions in your learning space is also essential. Because the distributions have differences, it is important to have them readily available to explore those differences.

Once you have your Linux learning space set up, you can start to dive into the CompTIA Linux+ certification objectives. I'll begin covering those objectives in the next chapter.

Exam Essentials

Describe the different license models used by Linux distributions. There are two main license models used by open source software: Copyleft and Permissive. Copyleft licenses (such as GPL and LGPL) require that you license the derivative work using the same license model. Permissive licenses (such as Apache and MIT) allow you to release derivative work using a different license, or even no license at all.

Explain why there are so many different Linux distributions. Linux distributions bundle different parts of the Linux system together to make it easy to install a complete Linux system. The main parts include the Linux kernel, the GNU software utilities, a user interface (either a graphical desktop or a command-line interface), and finally, specific application software, based on your requirements.

Describe the different hardware platforms Linux can run on. The Linux kernel has been compiled to run on several different CPU architectures, including Intel and AMD x86 and x86_64, ARM aarch64, and the PowerPC architecture used in older Apple workstations.

Review Questions

You can find the answers to the review questions in Appendix A.

1. What license model requires that you release any derivative software using the same license?
   - **A.** MIT
   - **B.** GPL
   - **C.** Apache
   - **D.** LGPL
   - **E.** Proprietary
2. What CPU architecture is an open standard that can be used by any CPU manufacturer?
   - **A.** x86
   - **B.** x86_64
   - **C.** RISC-V
   - **D.** s390x
   - **E.** aarch64
3. Which distribution group uses the Debian-based model?
   - **A.** dpkg
   - **B.** aarch64
   - **C.** GPL
   - **D.** RPM
   - **E.** LGPL