What Makes Up the Modern Linux OS?

In this Paper

Introduction

Most of us use Linux every day without even thinking about it. Linux is in our Android phones and our home routers. At the same time, Linux is also powering the world's largest datacenters and the Internet. But what makes up the modern Linux operating systems? And where did Linux come from?

Let's find out!

The History of Linux

For a moment, let's first take a step back in history. The creation of Linux starts with another operating system known as UNIX, which was first released in 1971. In 1983, the GNU Project (which stood for "GNU's not Unix") was started to create a complete UNIX-compatible operating system. Efforts stalled, and the project was missing a *kernel*. Around 1987, a UNIX-like operating system for students was released called MINIX, but its licensing prevented it from being distributed freely. Linus Torvalds at the University of Helsinki in Finland was frustrated by the licensing of MINIX and began working on his own operating system kernel. His kernel, released in 1991, when combined with the GNU applications and open-source licensing, became the Linux operating system we know today.

What Is a Kernel, and What Does It Do?

The *kernel* is the special piece of the operating system that controls the CPU hardware, allocates memory, accesses data, schedules processes, runs the applications, and protects them from each other. It is the first program loaded on the computer when the computer starts up. The most critical pieces of code in the kernel are loaded into protected areas of memory so that they can't be overwritten by other applications running in the operating system.

Since then, thousands of developers from around the world have contributed to enhancing the Linux kernel as well as the many pieces of software that make up the many different Linux distributions. Those developers include volunteers as well as developers from commercial companies. Today, the nonprofit Linux Foundation helps to create standards, awareness, and advancements across many different Linux projects. What Is an Operating System?

The short answer is that an *operating system,* or OS, is software that you load on your hardware to make it "do things." Without an operating system, most hardware is useless. For example, you might have a Dell computer that runs the Windows 10 operating system from which you run your applications. You might have an iPhone that runs the iOS operating system. You may also have an Apple MacBook that runs the Apple macOS operating system. The operating systems on these hardware platforms are what enable them to run applications, as shown in Figure 1.

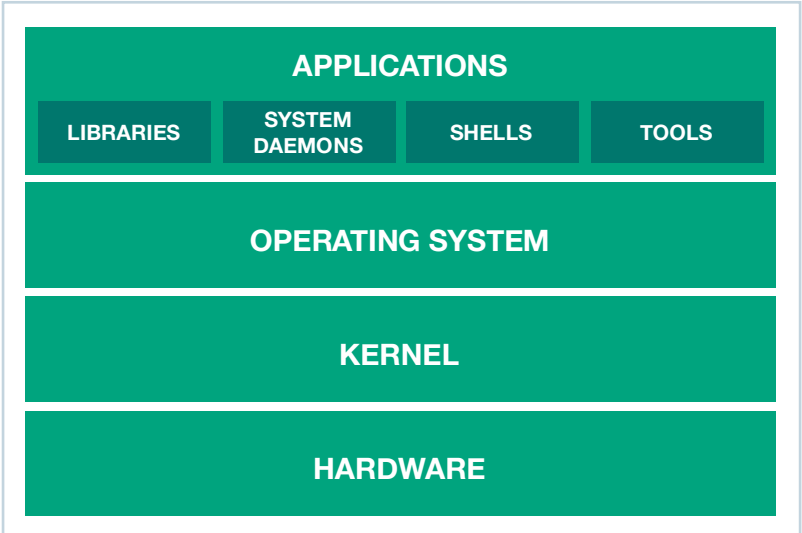

Figure 1. How an operating system works with hardware and applications

The Components that Comprise the Linux Operating System

Linux is an open-source OS that can be installed on a variety of different types of hardware to allow you to develop software, run applications, and more. At the heart of Linux is the *kernel*. Linux was developed in C and assembly language to run on i386 personal computers, but it has since been ported to more hardware than just about any other operating system in history. Today, Linux is the most installed operating system globally. In fact, the Space X Falcon 9 rocket and the International Space Station both use Linux!

Linux is typically administered from a command line interface (CLI), also known as a *shell*. Besides the kernel, which manages the hardware and software processes, Linux distributions include a collection of Linux software, such as device drivers for accessing and controlling hardware, shared libraries, applications, and system *daemons*, which run the in background and respond to network requests. Figure 2 shows an example of what a common Linux distribution might look like. Numerous programming languages are available for Linux, as well as more than 70,000 different applications. Applications are installed from *packages*, which contain the application itself and metadata about the application.

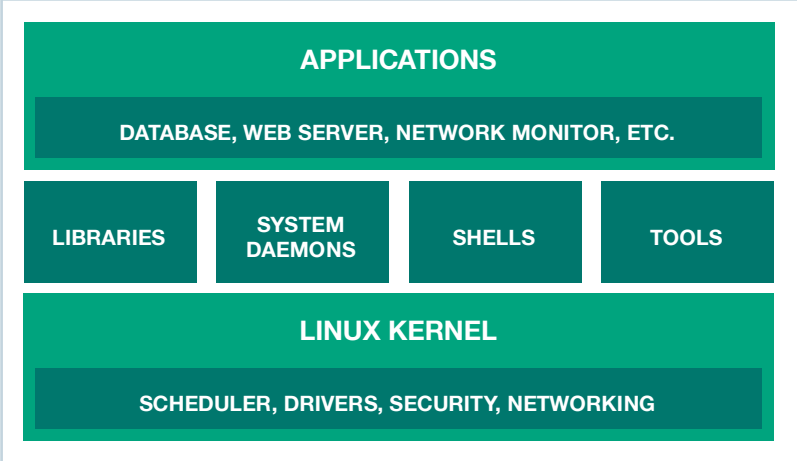

Figure 2. Example of a common Linux distribution

Definition: Metadata

Metadata is data about data. In essence, metadata describes the kind of information that an underlying data set will store. Take, for instance, a file system on a computer. When you view a directory listing, you see the file name, file size, create date, last modified date, and so forth. These are basic examples of metadata associated with each object in that directory.

What Is a Distribution?

Often called a "distro," a Linux *distribution* is the combination of specific versions of the Linux kernel with other libraries, system daemons, development tools, applications, packaging, and life-cycle management tools that are compatible with each other and tested for interoperability. The most common way that people acquire Linux today is by downloading one of the many different Linux distributions. Distributions are available not just for servers, desktop, and laptop computers, but also for a huge variety of more specialized devices that run Linux. Examples of Linux distributions are Ubuntu, Debian, Fedora, openSUSE, and Cumulus Linux.

Understanding User Space vs. Kernel Space

Operating systems all execute their kernel in protected and restricted memory that is called *kernel space* (see Figure 3) to prevent the kernel from terminating and crashing the system. When a user runs an application or tool, that application or tool executes in what is called *user space*. This distinction is critical. Applications can come from a variety of sources, may be poorly developed, or originate unknown sources. By running these applications separate from kernel space, they can't tamper with the kernel resources and cause the system to *panic* (crash).

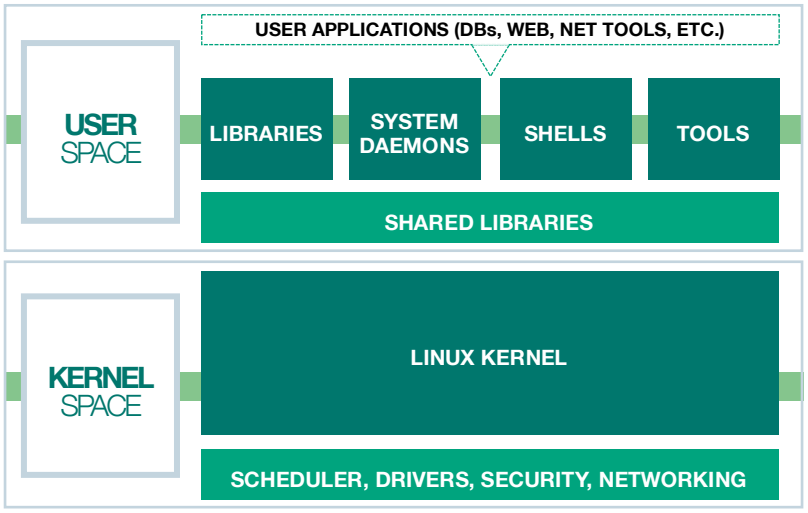

Figure 3. User Space and Kernel Space in the Linux kernel

All applications, even system daemon processes that perform critical operating system functions, must make what is called a "system call" to the kernel in the kernel space in order to access system resources such as memory or network devices. Every modern multi-user operating system has some type of user space versus kernel space design, which is intended to keep it secure, high-performing, and reliable.

In short, the separation between user space and kernel space is made to ensure that Linux is as reliable and secure an operating system as possible.

Benefits of Using Linux

Besides the fact that Linux is a great operating system, is continually being enhanced, and has a huge community following, Linux has gained such tremendous popularity because there are so many different benefits to using it. Some of these benefits include:

- **Consistent operating model**. No matter what version or distribution of Linux you use, whether you're on a supercomputer or a tiny embedded device, the general operation of Linux is the same no matter where you go. What this means is that, with some exceptions, the command line syntax is similar, process management is similar, basic network administration is similar, and applications can be (relatively) easily ported between distributions. The end result of this consistent operating model is a cost savings generated by greater staff efficiency and flexibility.
- **Scalability**. At this point, you already know that Linux is eminently scalable and is able to run on everything from wristwatches to supercomputers to globally distributed computing clusters. Of course, the benefit of this scalability isn't just the device mix, but also that its basic functionality — command line tools, configuration, automation, and code-compatibility — remains the same no matter where you're using it.

- **Open-source and community optimized**. With Linux's open-source, freely available nature, you might be concerned about future enhancements, bug fixes, and support. Fortunately, you can put those worries aside. If you look at the Linux kernel alone, with its 22 million lines of code, you'll find a strong community developing it behind the scenes. In 2016, one report said that over 5,000 individual developers representing 500 different corporations around the world contributed to enhancements in the Linux kernel, not to mention all the other surrounding applications and services. A staggering 13,500 developers from more than 1,300 companies have contributed to the Linux kernel since 2005. You might wonder why commercial entities contribute code to Linux. While many open-source advocates see the open-source nature of Linux as purely idealistic, commercial contribution of code is actually a strategic activity. In this sense, the for-profit companies who are dependent on Linux contribute their changes to the core to ensure that those changes carry forward into future distributions without having to maintain them indefinitely.
- **Full function networking**. Over the years, Linux has built up a strong set of networking capabilities, including networking tools for providing and managing routing, bridging, DNS, DHCP, network troubleshooting, virtual networking, and network monitoring.
- **Package management**. The Linux package management system allows you to easily install new services and applications with just a few simple commands. See the sidebar below for more information about Linux package management.

Linux Package Management

A Linux package management system is a tool that helps Linux administrators install and manage applications and extensions for the Linux operating system. Each Linux distribution carries its own package management capabilities. A Linux package includes all the bits necessary for a new application or service to operate. The package management system can also help an administrator address any dependencies that a package may have. A dependency is a software package necessary for another package to operate. By layering these dependencies, newly developed packages can then leverage the work of others without having to constantly reinvent the wheel. However, maintaining dependencies can be difficult, particularly as you continue to add packages. A good package management system will ensure that all dependencies are handled at the same time that you install new packages.

How Is Linux Used in the Enterprise?

Many modern ideas in data center computing have Linux underpinnings. Here are just a few examples:

- **Automation and orchestration.** Automation is used to perform a common task in a programmatic/scripted way, whereas orchestration is used to automate tasks across multiple systems in a data center. Linux is being used to automate and orchestrate just about every process in the enterprise data center.
- **Server virtualization.** Server virtualization is the ability to run a full operating system on top of an existing bare metal server. These virtual machines (VMs) can be used to increase server utilization, simplify server testing, or lower the cost of server redundancy. The software that allows VMs to function is called a *hypervisor.* Linux includes an excellent hypervisor called KVM.
- **Private cloud.** Another open-source project called OpenStack, which also runs on Linux, has become a leading cloud management platform for creating a private cloud. With private cloud, companies can leverage many of the same advantages of public cloud (scalability, self-service, multi-tenancy, and more) while running their own IT infrastructure on-premises.
- **Big data.** More and more companies are having to deal with exponentially increasing amounts of data in their data center, and because Linux offers such scalability and performance, it has become the go-to operating system for crunching big data via applications like Hadoop. Even Microsoft recently announced a big data solution based on Linux.
- **Containers.** Linux can also be used to run containerized applications, such as Docker containers, which are being used more and more by many companies. In fact, Linux is the foundation of the modern container movement; all container packaging and orchestration relies on Linux namespace and isolation mechanisms in order to operate.

In Summary

Linux has been around for over 25 years, is widely used on Earth and in space, and is a resilient operating system powered by the Linux kernel. Linux is made up of a number of components including the scheduler, drivers, libraries, system daemons, shells, tools, and applications. It's separated into the user and kernel space to ensure availability and security. The numerous benefits of using Linux are the consistent operating model, scalability, open source community, full-function networking, and package management. Because of its power, reliability, and flexibility, Linux has earned a dominate position for when enterprises need a solution for automation, orchestration, server virtualization, private cloud, big data or containers.

Last updated: December 2017