Contents

Applied Linux

What is Linux?

Linux is a piece of software called a kernel. When combined with a set of accompanying tools known as utilities and daemons, these components form an operating system. This system enables users to run complex applications, ranging from Steam games to accounting software.

Running a Linux system is much like being part of a professional band. Each musician in the band has dedicated time and effort to mastering a specific instrument. Learning to play an instrument involves understanding not only how to produce sound but also how to tune and maintain it for optimal performance.

Similarly, to use a Linux system effectively, one doesn't need to be an expert in every aspect. Just as someone can join a band by playing simpler instruments like the tambourine or cowbell, a Linux user can start with basic tasks while gradually learning how the system components work together. First, it's essential to understand the overall composition, the role of each part, and how they fit into the system.

In a Linux system, the kernel acts as the band manager, coordinating the different parts of the system. The various utilities function like individual musicians, each contributing to the overall harmony. Some utilities work silently in the background, ensuring the system runs smoothly. Others provide support through a graphical user interface (GUI), offering visual tools for user interaction.

Additionally, there are specific applications designed to perform particular tasks, such as browsing the web or playing games like NetHack. There are also utilities dedicated to handling computer to computer communications, such as web servers, mail servers, or social media instances like Mastodon.

In essence, using Linux is akin to touring with a well-run band. The kernel plans the show, while each utility and application plays its part, contributing to the creation of a cohesive and powerful operating system. Whether you're a novice

or a seasoned expert, there's a place for you in this dynamic and collaborative environment.

Why Linux?

The same Linux operating system that powers 100% of the world's top 500 supercomputers, NASA's Mars rovers, and a wide range of drones is available to you and your boundless imagination. According to the TOP500 list, Linux now runs on every one of the top supercomputers globally, demonstrating its dominance in high-performance computing. NASA's Perseverance and Curiosity rovers both rely on Linux for key functionalities as they explore the surface of Mars, further exemplifying Linux's reliability in extreme environments. Linux is also the operating system of choice for many drone systems, including those using the PX4 flight stack and ArduPilot, thanks to its real-time capabilities and flexibility.

This powerful operating system is supported by over 21,000 developers who have contributed to the Linux kernel since its inception, with more than 1,200 companies involved in the development effort (Linux Foundation). Every year, thousands of individuals work on maintaining critical utilities, applications, and distributions, showcasing the strength of open-source collaborative development and the enduring success of the Linux ecosystem.

The Power of Open Source

Linux is open-source software, meaning its source code is freely available for anyone to view, modify, and distribute. This open nature fosters a collaborative environment where developers worldwide can contribute to its development. The benefits of open-source software include:

- Transparency: Users can inspect the source code to learn how the appliation works.
- Security: With many people besides just the developers reviewing the code, security issues can be identified and fixed quickly.
- Flexibility: Users can modify the software to suit their specific needs.
- Community Support: A large and active community provides support, documentation, and tutorials.

Supercomputers

Linux is the top choice for supercomputers, which are the most powerful computers in the world. According to the TOP500 ranking of these supercomputers, all of them have been running on Linux since 2017.

The choice of Linux for supercomputing is popular for a few important reasons:

1. Customization: Supercomputers built from varying hardware configurations need to handle huge amounts of data and perform very complex

calculations. Linux's flexibility allows it to be fine-tuned to get the best performance for these demanding jobs and varying configurations.
2. Stability: Supercomputers need to run continuously for long periods without crashing. An unstable operating system can cause a lot of problems and wasted time. Linux is known for being very stable and reliable.
3. Support for High-Performance Applications: There are many software programs and tools designed to work with Linux, especially for high-performance computing. This makes it easier for scientists and engineers to run their advanced research projects and simulations on Linux-based systems.

Space Exploration

Linux's use in space missions is a testament to it's adaptability, reliability, and robustness. NASA's Mars robers, including the famous Perserverance, run on different Linux variants. The open source nature allows NASA engineers to modify the software to meet the unique requirements of space exploration, ensuring that the rovers can operate in the harsh conditions of space.

Drones

Many drones, particularly those used in research and commercial applications, run on Linux. The operating system's flexibility allows developers to create sophisticated flight control software, integrate various sensors, and process data in real-time. Projects like the Dronecode Foundation leverage Linux to develop open-source software for drones, benefiting from the collective expertise of developers worldwide.

Personal Computers

Linux distributions (distros) such as Ubuntu, Rocky Linux, and Linux Mint provide user-friendly interfaces and a vast repository of software for personal use. These distributions are suitable for tasks like web browsing, office productivity, media consumption, and software development. Users appreciate the security, performance, and customizability that Linux offers.

Phones

Android, the most popular mobile operating system globally, is built on the Linux kernel. The Linux kernel provides Android with a stable and secure foundation, managing hardware resources efficiently. This allows Android devices to offer robust performance and security features, essential for modern smartphones that handle sensitive data and perform complex tasks.

Servers

Linux is the backbone of the internet, running a significant portion of web servers, databases, and cloud infrastructure. Popular web servers, like Apache and Nginx, are built on Linux, which provides the performance and stability required for high-traffic websites. Linux's powerful networking capabilities and robust security features make it the preferred choice for server environments.

Development

For software developers, Linux offers an unparalleled development environment. The operating system comes with a rich set of programming tools, compilers, and libraries. Package managers like APT (the "Advanced Package Tool"), YUM ("Yellowdog Updater Modified"), and DNF ("Dandified YUM") make it easy to install and update software. Integrated development environments (IDEs) such as Visual Studio Code, Eclipse, and JetBrains' suite of tools run smoothly on Linux, providing developers with everything they need to create software.

Community and Collaboration

The Linux community provides the system's greatest strengths. This global network of developers, users, and enthusiasts collaborates to improve the operating system, develop new software, and provide support. Communities exist around specific distributions, projects, and technologies, offering forums, mailing lists, and chat channels where members can share knowledge and help each other.

The **Free Software Foundation (FSF)** and the **GNU Project** are deeply interconnected, as the GNU Project was founded by **Richard Stallman** in 1983 under the principles of the FSF, which he founded in 1985. The FSF is the organization behind promoting the philosophy of free software and the development of the GNU Project, which is focused on creating a completely free Unix-like operating system. The relationship between the FSF and the GNU Project can be summarized as follows:

- **FSF** provides legal, philosophical, and infrastructural support for free software development, including advocacy for software freedom and maintaining the GNU General Public License (GPL), which is a key component of the GNU Project.
- **GNU Project** is the technical implementation of the FSF's mission, The creation of a free operating system (initially called GNU, standing for "GNU's Not Unix"). The GNU Project includes many tools and software components, such as the GNU Compiler Collection (GCC) and the GNU Core Utilities.

Twin Cities Linux User Group (TCLUG)

It was a festive atmosphere, alive with the hum of excited conversation and the clatter of keyboards. Enthusiasts of all ages and backgrounds filled the room,

darting between tables to get a closer look at the latest or strangest hardware someone had proudly hauled in. It was a ritual of sorts, gathering around devices like ancient storytellers around a fire, except here, the tales were of processors, ports, and possibilities.

Real-Time our ISP sponsor always set up their FTP server in a corner, a quiet yet crucial element of the event. The server acted like a beacon, constantly feeding CDs into the burner with the latest ISOs—fresh new operating systems ready to be tested and explored. The soft whir of burning discs punctuated the air as people lined up to grab one, eager to get their hands on the newest builds.

Knowledge flowed freely here. Everyone was both a student and a teacher, though some were more sought-after than others. Those with the answers—especially the hard-earned ones—were like sages. You just had to know who to ask. Someone might whisper, "Go find so-and-so, they've got that Parallel Port Iomega Zip drive running smooth as butter." And off you'd go, navigating the maze of people to track down the elusive expert, hoping they'd have the magic fix for your particular problem.

In another corner of the room, debates raged like a wildfire over the finer points of Linux. Veteran users, the seasoned soldiers of this open-source world, passionately argued the merits of their chosen distributions. They would preach to the wide-eyed newcomers, trying to win them over: "You *have* to install this one! It's the best for stability, security, and—well, everything!"

It wasn't just about the distros, though. The debates dug deep into technical details that only a room like this could fully appreciate. "You need at least *two times the RAM* for your swap space!" one person would proclaim with authority, arms crossed, as if this was the final word on the matter. A chorus of murmured agreement and raised eyebrows followed, though, of course, not everyone was convinced. "LVM or no LVM?" someone else would chime in, sparking another round of impassioned arguments. And then there was the question of encryption: Was it necessary for everyday use, or just for the paranoid? Everyone had an opinion, and no one was shy about sharing it.

What was most remarkable, though, was how the conversation transformed people. The socially awkward—the ones who might normally shuffle through life avoiding eye contact—suddenly became animated and alive when talk turned to code and systems. They would light up, speaking with the kind of confidence that only comes from mastering a domain few truly understand. It was as though the technical language itself was a key, unlocking a part of them that had been waiting to burst forth.

As the evening wore on, the crowd thinned. The once packed room became quieter, save for the hum of a few laptops still grinding through lengthy installs. Those left behind sat in patient anticipation, caught in the timeless dilemma of tech enthusiasts everywhere: "Do I pack up and try to finish this at home, risking something breaking along the way? Or do I wait it out here, where help is just a few steps away?"

Eventually, the last holdouts made their way to the door, fatigue in their steps but excitement still in their eyes. They talked of dinner plans, sharing one last laugh before parting ways, already looking forward to the next gathering. It wasn't just the love of Linux that brought them together—it was the sense of community, of belonging to something bigger than any one person or system.

This was the heart of the Linux hacker community: a place where ideas and arguments flew as fast as the bits on the wire, where everyone—no matter how awkward or unassuming—could find their voice, and where the spirit of curiosity and collaboration thrived, always pushing toward the next innovation, the next solution, the next great conversation.

Components and Related Technologies:

1. **GNU Hurd**:
   - The **Hurd** kernel developed by the GNU Project, works with the GNU utilities. Unlike more traditional monolithic kernels, Hurd is a collection of servers running on top of a microkernel (typically **Mach**). However, due to its complexity and slow development, it has not been widely adopted.
2. **Linux Kernel**:
   - The **Linux kernel**, developed by **Linus Torvalds** in 1991, became a critical piece in creating a fully functional operating system when combined with GNU tools. The combination of GNU software and the Linux kernel is colloquially referred to as "GNU/Linux." Although the Linux kernel is not part of the GNU Project, it is used as the kernel in most GNU-based operating systems today.
3. **musl libc**:
   - Popular in the embedded systems and minimalist distributions like Alpine Linux, **musl libc** is an alternative to the **GNU C Library (glibc)**, which is the default C standard library used in GNU systems, musl is known for being lightweight, fast, and designed with simplicity in mind.
4. **BusyBox**:
   - **BusyBox** is a collection of Unix utilities designed to provide much of the functionality of the GNU Core Utilities in a smaller, more lightweight package. It's often referred to as the "Swiss Army Knife of Embedded Linux" because it combines many common command-line tools into a single executable, which is useful for embedded or minimalistic environments.

Conferences and Events

Events like the Linux Foundation's Open Source Summit, FOSDEM, and the Technology and Security Conference, DEFCON, bring together thousands of Linux users and developers to share ideas, collaborate on projects, and discuss

the future of open-source software. These conferences feature talks, workshops, and hackathons, providing opportunities for learning and networking.

Why this book?

Clearly, journeying with Linux reveals a world of possibilities. This book will guide you through the essentials and beyond, equipping you with the knowledge and skills to master the Linux operating system. Here's a glimpse of the key concepts and topics we will explore in the upcoming chapters:

Logging In

We will start with the most basic of operations and build on them. To begin, we'll explore and practice **Logging In** to your Linux system. We'll use the Graphical User Interface (GUI), the Terminal, and Secure Shell (SSH) to interact with, configure, and use a Linux operating system.

Linux Networking

Networking is the backbone of modern computing. We will cover the configuration and management of network settings, ensuring you can connect and communicate effectively. You will also learn how to troubleshoot network issues, maintaining a seamless and reliable network environment.

Security

Security is paramount in today's digital world. We will discuss basic security measures, firewall configuration, and access control to safebuard your Linux system against unauthorized access and attacks.

Systems Operation and Maintenance

Linux administratorsi need a solid foundation to **managing system startup and services**. We will cover the intricacies of system operation and maintenance, focusing on monitoring system performance and ensuring your system runs efficiently and reliably.

Automation and Scripting

Efficiency and automation are key to effective system administration. You will learn how to write and run scripts, automating routine tasks to save time and reduce the potential for human error. This chapter will empower you to streamline your workflows and enhance productivity.

Hardware and System Configuration

We will deep dive into understanding **different hardware architectures**, essential for appreciating the flexibility and power of Linux. You will learn how

to install and configure Linux on various hardware setups, ensuring a smooth and optimized performance for your specific needs.

Storage, Monitoring and Troubleshooting

You can **identify and resolve common problems** swiftly with the right tools and knowledge. We will introduce you to various diagnostic tools and techniques for troubleshooting, helping you maintain a stable and robust Linux system.

As we proceed, each chapter will build on these foundational concepts, providing you with a comprehensive understanding of Linux. By the end of this book, you will be well-equipped to tackle complex Linux tasks with confidence and expertise. So, let's continue this journey together and unlock the full potential of Linux.

Getting Started

Installing and Configuring Linux

Although installing an operating system can be intimidating, we'll safely guide you through the process with the following stepwise instructions.

Before dificing in let's make sure you're well-prepared. Think of this as picking an instrument before your first lesson.

1. Hardware Compatibility

- Your computer system is the instrument you will be training on. We'll learn how to tune it and make sure it's ready for our lesson.
- Linux runs on hardware as old as the Intel 80486 from 1989. While unlikely to be incompatible, check that any custom or specialized hardware components are compatibleand supported with your chosen Linux distribution.
- Most modern distributions, like Ubuntu and Rocky, support a wide range of hardware. A quick search on your distribution's website or forums can save you from unexpected issues.

2. Backup Data:

- Just like safeguarding your sheet music and notes, it's crucial to back-up your important files when you install Linux on a system with existing data. Use an external drive or cloud storage to keep your files safe. This way, you won't lose any precious memories or important documents if an error occurs during the installation.

3. Choose a Distribution:

- Selecting a Linux distribution is like choosing a music genre to practice. Popular choices include Ubuntu, Rocky, Debian, Alpine, and Arch Linux. Each has its own cadence and caters to different needs. Whether you're